FernUniversität in Hagen

Department of Mathematics and Computer Science

Modellierung im Kontext: Ontologie-basiertes Information Retrieval

Jan Werrmann 2011

Research Report 1/2011
ISSN 1865-3944

Editor:	Dean of the Department of Mathematics and Computer Science
Type and Print:	FernUniversität in Hagen
Distribution:	http://deposit.fernuni-hagen.de/view/departments/miresearchreports.html

Modellierung im Kontext: Ontologie-basiertes Information Retrieval

Jan Werrmann*
Daimler AG, GSP/ODI

May, 2011

Abstract

Heterogeneous document landscapes in companies hold knowledge in the form of potential linkage between domain-specific documents of various document systems. To access this (hidden) knowledge, we developed a design pattern for an ontology to model a homogeneous access structure on top of the heterogeneous document landscape.
Furthermore, we describe an **Advanced Ontology-based Information Retrieval System** (AIRS) that includes this ontology to generate a retrieval strategy and to find (unknown) document relationships. With the help of the AIRS, it should be possible to access different document systems in different application contexts.

Zusammenfassung

Heterogene Dokumentenlandschaften in Unternehmen bergen Wissen in Form von Verknüpfungspotentialen zwischen domänenspezifischen Dokumenten verschiedener Dokumentensysteme. Um dieses teils verborgene Wissen ableiten oder generieren zu können, entwickeln wir ein Entwurfsmuster für eine **Ontologie**, die eine homogene Zugriffsstruktur über einer heterogenen Dokumentenlandschaft etabliert.
Weiterhin beschreiben wir ein **Advanced Ontology-based Information Retrieval** (AIRS) Verfahren, mit dessen Hilfe diese Meta-Ontologie zur Generierung von Anfragestrategien an Dokumentensysteme und für die Dokumentenrecherche in verschiedenen Anwendungskontexten genutzt werden können.

Stichworte

Ontology, Concepts, Relations, Context of use, Information Retrieval

*jan.werrmann@daimler.com

1 Motivation

In Unternehmen fallen in verschiedenen Bereichen große Mengen unterschiedlicher Daten an. Diese Daten entstehen aus geschäftsprozessorientierten und domänenspezifischen Arbeitsabläufen und dienen je nach Anwendungsszenario als Grundlage weiterer Prozesse.
Die so in der Historie und in verschiedenen Unternehmensbereichen entstandenen Datenhaltungssysteme agieren oft unabhängig voneinander.
Neue Geschäftsprozesse erfordern jedoch eine Gesamtbetrachtung dieser Daten, um verborgenes, erst durch eine semantische Zusammenführung ableitbares Wissen erheben und in verschiedenen Anwendungsszenarien nutzen zu können.

Betrachtet sei folgendes Beispiel für eine heterogene Dokumentenlandschaft. In der fachspezifischen Domäne des After-Sales Managements von Mercedes-Benz Cars finden sich im Werkstattprozess beispielsweise folgende Dokumentensysteme:

- *Ein Diagnose-Daten-System, in welchem fehlercodespezifische sowie symptombasierte Prüfungen für Fahrzeug-Diagnose-Prozesse verwaltet werden,*
- *ein Werkstatt-Informations-System, in welchem Wekrstatt-Literatur-Dokumente zur Unterstützung von Wartungen, Reparaturen und Diagnose von Fahrzeugen verwaltet werden,*
- *ein Werkstatt-Hilfe-System, in welchem Dokumente verwaltet werden, die über aktuelle Abhilfen zu technischen Beanstandungen im Kundenbetrieb von Fahrzeugen informieren,*
- *verschiedene domänenspezifische Wörterbücher*[1] *sowie*
- *elektronische Kataloge für beispielsweise Ersatzteile, Arbeitswerte und Richtzeiten.*

Diese Systeme werden von unterschiedlichen Bereichen des After-Sales von Mercedes-Benz Cars betreut und unterliegen somit verschiedenen Autoren– sowie Veröffentlichungsprozessen.
Im gesamten Werkstattprozess müssen für bestimmte Anwendungsszenarien diese Systeme allerdings übergreifend betrachtet werden:
Beispielsweise werden für den Fahrzeug-Diagnose-Prozess und für den Fahrzeug-Reparatur-Prozess (unter anderem) Dokumente des Werkstatt-Informations-Systems und des Werkstatt-Hilfe-Systems benötigt.
Im optimierten Fahrzeug-Annahme-Prozess sollen (durch Kunden erlebte)

[1] Zum Beispiel Thesauri zur Modellierung von Begrifflichkeiten oder Synonymlexika für den Anwendungsfall eines kontrollierten Vokabulars.

Symptome standardisiert erfasst und mögliche Teile einer Reparatur sowie hilfreiche Dokumente für eine Reparatur der Fahrzeuge zusammengestellt werden.
Motiviert durch das Vernetzungspotential der Service-Betreiber[2] *mit den, die verschiedenen Dokumentensysteme betreuenden Bereichen des After-Sales von Mercedes-Benz Cars besteht die Herausforderung darin, das Wissen der Systeme zu bündeln, um den Werkstattprozess zu optimieren.*

In diesem Rahmen sei also angenommen, dass Datenverknüpfungen zur Qualitätssicherung und Qualitätsoptimierung in verschiedenen Anwendungskontexten beitragen können.

Aus dieser Annahme ergeben sich Informationswünsche nach Ausschöpfung dieser Verküpfungspotentiale. Jene resultieren aus den Möglichkeiten der Ableitung, dem Lernen und dem Verknüpfen von bisher unbekanntem Wissen über die Daten der domänenspezifischen Datenbasis. Aus der Ausformulierung dieser Informationswünsche –und unter der Prämisse, dass man mit „unterschiedlichen Daten" eine heterogene Dokumentenlandschaft adressiert– ergeben sich folgende Fragestellungen:

1. Wie soll man die heterogene Dokumentenlandschaft –inklusive der Verknüpfungspotentiale der Dokumente– modellieren, um eine homogene Zugriffsstruktur etablieren zu können?

2. Wie soll man Verbindungen zwischen (nicht offensichtlich) semantisch korrelierenden Dokumenten erkennen, generieren und abbilden?

3. Wie kann man diese Verbindungen in einem Anwendungskontext nutzen?

Der hier vorgestellte Ansatz sieht vor, eine Ontologie als Meta-Ebene über der heterogenen Dokumentenlandschaft zu etablieren, um Dokumente verschiedener Dokumentensysteme (Quellen) miteinander in Beziehung setzen zu können.
So ist es möglich, eine Verbindung zwischen semantisch korrelierenden Dokumenten abzubilden. Diese Verbindungen können dann beispielsweise in einer quellenübergreifenden Dokumentenrecherche genutzt werden.

Im folgenden Abschnitt 1.1 werden Datenpotentiale allgemein und anhand der schon eingeführten Domäne[3] dargelegt. Im Abschnitt 2 werden verschiedene Aspekte der ontologiebasierten Wissensrepräsentation vorgestellt.
Anschließend wird in Abschnitt 3 ein Vorschlag für eine die heterogene Dokumentenlandschaft abbildende Ontologie gegeben. Im Abschnitt 4 wird

[2]Das sind sowohl Niederlassungen als auch Vertragswerkstätten.
[3]Werkstattprozess eines Automobilherstellers.

nachfolgend geschildert, wie sich diese Ontologie im Falle eines Information Retrieval Systems nutzen lässt und im letzten Abschnitt 5 werden die weiter notwendigen Arbeitsschritte beschrieben.

1.1 Ableitbares Wissen heterogener Dokumentenlandschaften

Wie oben beschrieben, bergen domänenspezifische heterogene Dokumentenlandschaften den Wunsch das Wissen der Verknüpfungspotentiale von Dokumenten und Quellen auszuschöpfen.

Als **Wissen**[4] versteht man hier das Kombinieren von Informationen zum Zweck der Problemlösung oder der Gewinnung neuer, bisher unbekannter Informationen. Ein **Dokument** sei ein durch Zeichen repräsentierter Informationsträger und eine **Quelle** eine Menge gleichartig attribuierter Dokumente.
So kann man Dokumente als mit „Kontext angereicherte Daten“ betrachten. Das Potential einer heterogenen Dokumentenlandschaft könnte man interpretieren als Möglichkeit, bisher unbekannte Verbindungen zwischen Dokumenten aufzudecken.

Durch die Verbindung heterogener Datenquellen wird es möglich, ein Informationsbedürfnis[5], welches an eine Quelle gerichtet ist, über semantische Verbindungen zwischen den Dokumenten an andere Quellen weiter zu reichen. So kann man Dokumente finden, die aus subjektiver Sicht eigentlich *nicht (offensichtlich)* dem Informationsbedürfnis zuzuordnen sind.

Abbildung 1 verdeutlicht dies: Ein bestimmtes Informationsbedürfnis lässt sich exklusiv und (mehr oder minder) eindeutig einem Dokument der *Quelle X* zuordnen. Da dieses Dokument (auf welche Art auch immer) mit bestimmten Dokumenten der *Quelle Y* in Beziehung steht und diese wiederum mit Dokumenten der *Quelle C* verbunden sind, ließe sich das Informationsbedürfnis von Quelle zu Quelle und von Dokument zu Dokument „tragen“[6].

Bei der Betrachtung der im Abschnitt 1 eingeführten Domäne des After-Sales Bereiches von Mercedes-Benz Cars könnte sich beispielsweise folgendes Szenario darstellen: Hier sei ein durch den Kunden erlebtes Symptom

[4] Es existieren verschiedene Definitionen von Wissen und dessen Zusammenhang mit Daten und Informationen. Ein thematischer Überblick ist unter anderem in [21] und [26] gegeben.

[5] Unter einem Informationsbedürfnis versteht man den Wunsch nach Informationen, die aus subjektiver Sicht relevant sind.

[6] Hier ist im Besonderen hervorzuheben, dass auf diese Art Dokumente einer Quelle selbst dann gefunden werden können, wenn die Quelle *nicht* mit dem Informationsbedürfnis verknüpft werden *kann*.

im Betrieb seines Fahrzeuges Gegenstand des Informationsbedürfnisses[7] *im Annahme-Prozess*[8] *der Werkstatt.*
Die Aufgabe der Service-Fachkraft für die Fahrzeugannahme soll nun darin bestehen, relevante Werkstatt-Dokumente sowie eventuell benötigte Ersatzteile zusammenzustellen, um (beispielsweise dem Kunden gegenüber) potente Aussagen über nötige Reparaturen tätigen zu können[9]*.*

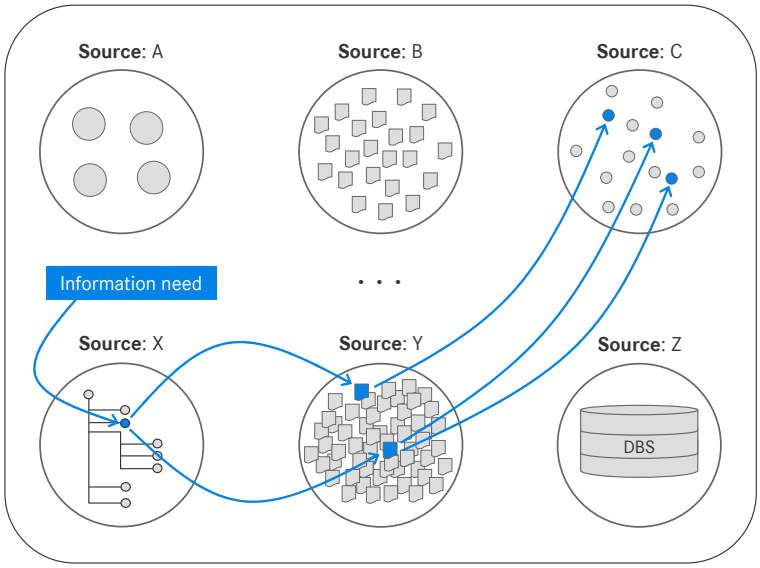

Abbildung 1: Informationsbedürfnis und heterogene Dokumentenwelt

Die Abbildung 1 betrachtend lässt sich im Hinblick auf ausgeschöpfte Dokumentenverknüpfungspotentiale folgende Erkenntnis gewinnen: Hier könnte sich eine Symptombeschreibung den taxonomisch erfassten Symptomen zuordnen lassen (Quelle X) und dieses taxonomisch erfasste Symptom könnte wiederum mit zwei Werkstatt-Hilfe-System Dokumenten in Verbindung stehen (Quelle Y). Da weiterhin jene Werkstatt-Hilfe-Dokumente auf Ersatzteile eines Teilekatalogs verweisen könnten (Quelle C), lässt sich auf diese Weise schon bei bekanntem Symptom eine Aussage über Ersatzteile und informative Werkstatt-Hilfe-Dokumente einer nötigen Reparatur tätigen.

Die Potentiale der heterogenen Dokumentenlandschaft lassen sich demnach im jeweiligen Anwendungskontext nutzen.

[7] Gemeint ist das Bedürfnis, Dokumente zu finden, die aus der Sicht des Werkstattprozesses und im Hinblick auf das durch den Kunden erlebte Symptom für eine Reparatur relevant sind.
[8] Der Annahme-Prozess ist dabei Teil des (gesamten) Werkstattprozesses.
[9] Dies könnten beispielsweise Informationen über Dauer und Kosten aber auch erste Einschätzungen von Ursachen der Beanstandung sein.

Für eine gewinnbringende Nutzung dieser Dokumentenverbindungen postulieren wir die Modellierungsform einer, die heterogene Dokumentenlandschaft beschreibenden, Ontologie, die in einem erweiterten ontologiebasierten Infomation Retrieval System genutzt werden kann. Diese (Ontologie) ermöglicht den Zugriff auf das Verknüpfungspotential der Dokumente.

2 Wissensrepräsentation im Kontext

What is a knowledge representation? In [6] wird die Frage durch fünf verschiedene Rollen, die eine Wissensrepräsentation spielt, versucht zu beantworten. So sei eine *knowledge representation a surrogate*, *a set of ontological commitments*, *a fragmentary theory of intelligent reasoning*, *a medium for efficient computation* und *a medium of human expression*. In [3] definiert man *knowledge representation* als *„... the field of concerned with using formal symbols to present a collections of propositions believed by some putative agent. “*.

Man kann argumentieren, dass eine Wissensrepräsentation eine Abbildung von Wissen darstellt. Darüber hinaus sollte sich diese Abbildung in verschiedenen Anwendungskontexten und zum Lösen bestimmter Probleme effizient nutzen lassen.
So verstehen wir unter einer *Wissensrepräsentation* eine Modellierung der Anwendungsdomäne (hier: heterogene Dokumentenlandschaft) zum Zweck der Gewinnung neuer, bisher unbekannter Informationen. Das Modell ist eine ontologische Beschreibung der Domäne.

Eine *Ontologie* ist frei nach Gruber [11] eine explizite und formale Spezifikation einer (gemeinsamen) Konzeptualisierung, wobei „Konzeptualisierung“ ein Domänenmodell ist, bestehend aus Begriffen und semantischen Relationen. Eine explizite sowie formale Spezifikation adressiert eine vollständige und maschineninterpretierbare Konzeptualisierung.
Darüber hinaus ist es möglich, Ontologien nach Formalisierungsgrad und Expressivität[10] sowie Domänenreichweite[11] zu klassifizieren. Hierbei ist zu beachten, dass die Eingliederung in diese Klassifikationen von der Definition des Begriffes der Ontologie selbst abhängt[12].

Als weiteres und signifikantes Unterscheidungsmerkmal von Ontologien soll-

[10] Die Spannweite reicht von den informalen *Lightweight Ontologies* bis hin zu den formalen *Heavyweight Ontologies* (siehe [10] und [5]).

[11] Die Reichweite erstreckt sich von den auf ein Anwendungsgebiet beschränkten *Application Ontologies* bis hin zu den eine allumfassende Domäne betrachtenden *Top-Level-Ontologies* (siehe [13] und [12]).

[12] So erfüllt eine informale und sehr beschränkte Domäne beschreibende Ontologie den Anspruch des in diesem Artikel eingeführten Ontologiebegriffs wohl nicht und würde nicht als solche betrachtet werden.

te allerdings, so finden wir, der *Anwendungskontext* angesehen werden.

Unter einem **Anwendungskontext** verstehen wir eine anwendungsorientierte Modellierungsform der Ontologie.
So kann man unter anderem zwischen den folgenden Anwendungskontexten unterscheiden:

- Eine **Welt von Entitäten**, die Beziehungen zueinander aufweisen dient der Modellierung von domänenspezifischem Fachwissen (siehe beispielsweise OpenCyc[13]).
- Eine **lexikalische Wissensbasis**, aufgebaut aus Begriffen, Benennungen und semantischen Relationen, modelliert ein Fachvokabular, beispielsweise in einer intelligenten Suche (unter anderem in [8]) oder einer konzeptbasierten Klassifikation (beispielsweise in [7]).
- **Modellierung einer Dokumentenwelt**[14] zur ontologiebasierten Recherche. Diese Ontologieform basiert *nicht* auf Begrifflichkeiten, sondern auf Dokumenten. Die Dokumente werden als ontologische Individuen modelliert und die Dokumentensysteme als Quellen[15] abgebildet. Existierende Beziehungen zwischen Dokumenten und Quellen werden hier als Relationen zwischen den ontologischen Individuen abgebildet.

Anzumerken ist, dass die Anwendungskontexte sich durchaus überschneiden und vermischen können. Eine Auswirkung auf die Modellierung der Ontologie sollte dann im Detail betrachtet werden.

2.1 Elemente der Ontologie

Zusammenfassend lässt sich sagen, dass eine Ontologie aus drei elementaren Bestandteilen bestehen sollte[16].

1. Einem **Anwendungskontext**,
2. der **Konzeptualisierung** (das Modellieren von *Klassen*, *Individuen* und *semantischen Relationen*) und
3. einem **Regelwerk**.

[13] Siehe [9] und http://www.opencyc.org/, Version vom 28.03.2011.
[14] Entspricht der in diesem Artikel modellierten Ontologie.
[15] Eine Quelle kann sowohl als ontologische Klasse oder als Individuum abgebildet werden.
[16] Andere Ansätze nennen ebenfalls Grundbausteine von Ontologien. So wird in [22] postuliert, dass eine Ontologie strukturell gesehen aus den vier Bestandteilen *Lexikon* (terminologische Abbildung der Domäne), *Begriffen*, *semantische Relationen* und *regelhaften Zusammenhängen* bestehen sollte. Der in dieser Arbeit vorgestellte Ansatz berücksichtigt hingegen zusätzlich den *Anwendungskontext* als Bestandteil von Ontologien.

Klassen, Individuen und semantische Relationen bilden den Anspruch der Konzeptualisierung ab[17]. Das Regelwerk ist der Anspruch an Formalisierungsgrad und Inferenz, um Wissen ableiten zu können. Der Anwendungskontext hingegen beeinflusst sowohl die Konzeptualisierung, den Formalisierungsgrad als auch den expliziten Anspruch der Spezifikation der Ontologie.

3 Design und Modellierung einer Ontologie

Für das Erstellen von Ontologien gibt es verschiedene Empfehlungen. So werden in [24] fünf Stadien (*Indentify Purpose*, *Building the Ontology*, *Evaluation* und *Documentation*) propagiert. In [4] wir das Model der *Ontologie-Reifung* in den fünf Phasen *Emergence of ideas*, *Consolidation in Communities*, *Formalization* und *Axiomatization* beschrieben. In [16] hingegen werden fünf Ansätze des *ontology-design* (*Inspiration*, *Induction*, *Deduction*, *Synthesis* und *Collaboration*) benannt und ein Ansatz für gemeinschaftliches Erstellen einer Ontologie gegeben.

Die Modellierung der Elemente der in diesem Artikel erstellten Ontologie folgt den im Abschnitt 2 vorgeschlagenen Ansatz und gliedert sich dementsprechend in die drei Bestandteile *Anwendungskontext*, *Konzeptualisierung* und *Regelwerk*.

3.1 Der Anwendungskontext

Um den Anwendungskontext zu erfassen, muss die zu beschreibende Domäne betrachtet werden. Im Fokus stehen hier Fragestellungen, sich aus einer *top-down* Anforderungsanalyse ergeben. Beispielsweise:

- Existiert ein Anwendungsszenario? Hier: Warum sollte das Potential der heterogenen Dokumentenlandschaft ausgeschöpft werden?
- Ist es möglich, Relationen zu generieren[18]? Hier: Wie kann man aus dem vorhandenen Dokumentensystemen Verknüpfungspotentiale ableiten?
- Wie müssen die Entitäten und Relationen abgebildet werden? Hier: Wie sollen Dokumentensysteme, Dokumente und Verknüpfungen abgebildet werden?

[17] Das Modellieren von Begrifflichkeiten sollte bei der Betrachtung von ontologischen Elementen keine Rolle spielen, denn das ist eine Designfrage. So könnten *Begriffe* als Klassen, *Benennungen* als Individuen und eine *Synonymierelation* als Relation zwischen Begriffen und/ oder Benennungen abgebildet werden. Es sind aber ebenso andere Modellierungsformen denkbar.

[18] Relationen sind Voraussetzungen für eine gewinnbringende Modellierung von Ontologien.

- Gibt es Einflussfaktoren? Hier: Gibt es Beschränkungen der Gültigkeit von Dokumenten und Verknüpfungen?

Basierend auf dem Anwendungskontext propagieren wir einen rein dokumentbasierten Modellierungsansatz der Ontologie. Dieser findet sich in der Konzeptualisierung wieder.

3.2 Konzeptualisierung

Im Hinblick auf den Anwendungskontext und durch eine Sondierung der heterogenen Dokumentenlandschaft haben wir folgende Erkenntnisse gewonnen:

- In Dokumentensystemen (egal welcher Art) werden Dokumente verwaltet (oder lassen sich darin finden).
- Diese Dokumente können in ihrem Dokumentensystemen attribuiert/ segmentiert vorliegen.
- Dokumente einer Quelle sind stets gleichartig attribuiert.
- Dokumente können auf verschiedene Art und Weise mit anderen Dokumenten (auch aus der gleichen Quelle) in Verbindung stehen[19].
- Es existieren verschiedene Arten von Relationen: *Den After-Sales Bereich von Mercedes-Benz Cars betrachtend können beispielsweise standardisierte Symptome mit Werkstatt-Hilfe-Dokumenten in Relation stehen (is-related-to Relation). Symptome hingegen können auch untereinander in Relation stehen (Taxonomy-Relation).*
- Relationen und Dokumente können über kontextuelle Gültigkeiten verfügen[20].
- Die kontextuellen Gültigkeiten korrelieren mit dem Informationsbedürfnis des Suchenden (Suche unter der Bedingung, dass ...[21]).

[19] *Im After-Sales Bereich von Mercedes-Benz Cars liegen beispielsweise standardisierte Symptome und Symptomorte taxonomisch erfasst vor. Bei der Betrachtung von Taxonomiekonzepten als Dokumente existiert zwischen den Dokumenten (Taxonomiekonzepten) eine taxonomische Ordnung (also eine Taxonomie-Relation).*

[20] *Beispielsweise: Das baureihenunabhängige Ersatzteil x ist für eine Reparatur nur dann relevant, wenn das Werkstatt-Hilfe-Dokument, durch welches es gefunden wurde, im Kontext einer bestimmten Baureihe steht. Andernfalls würden durch die Service-Fahrzeugannahmefachkraft Ersatzteile gewählt werden, die nicht zum Kundenfahrzeug passen würden.*

[21] *Beispielsweise: Suche nach relevanten Ersatzteilen unter der Bedingung einer bestimmten Baureihe.*

Im Hinblick darauf muss also die Möglichkeit bestehen, Dokumente und deren Quellen durch ontologische Entitäten abzubilden. Weiterhin müssen Verbindungen zwischen den Entitäten darstellbar sein. Darüber hinaus muss auch der Attribuierung der Dokumente Rechnung getragen werden[22]. Dies beinhaltet ebenso das Vorhandensein von Document-Attributes (auf Dokument-Ebene und Quellen-Ebene), wie die Möglichkeit von attribuierten Dokumentenverbindungen.
Um gültige *Wege* auf Dokumentenebene gehen zu können (also mehreren Dokumentenverbindungen zu folgen, siehe Abbildung 1) müssen Relationen und (beste) Wege zwischen Dokumenten definiert werden (Abschnitt 3.3). Um das Verknüpfungspotential zwischen Quellen zu *erkennen*, müssen ebenfalls Relationen und (beste) Wege zwischen Quellen definiert werden[23]. Weiterhin müssen Gültigkeiten von Dokumenten und Relationen durch die Modellierung, beziehungsweise durch das Regelwerk aufgefangen werden.

Im Weiteren geben wir eine Empfehlung für eine die heterogene Dokumetnenlandschaft abbildende Ontologie. Die modellierten Bestandteile der Ontologie haben wir in Abbildung 2 zusammengefasst.

3.2.1 Ontologie-Elemente

Unter **Source** ist das Dokumentensystem zu verstehen. **Document** repräsentiert eine strukturierte Menge inhaltlich zusammengehöriger Daten, eben ein Dokument, welches in einem Dokumentensystem zu finden ist. Bei der Modellierung werden die Dokumente den Quellen über eine **document-of-source** Relation zugeordnet[24] (Abbildung 3 A).

Document-Attributes werden Quellen zugeordnet (Abbildung 3 B) und können von Dokumenten „verwendet“ werden[25] (Abbildung 3 C). So geben die den Quellen zugeordneten Document-Attributes Aufschluss darüber, welche Attribute die Dokumente der Quelle besitzen können aber nicht müssen (Abbildung 8).

Relationen sind zweistellig und existieren zwischen unterschiedlichen ontologischen Entitäten. So können zwischen Quellen *unbestimmte Relationen*[26]

[22] Eine Attribuierung findet sich unter anderem dann, wenn Informationen in einem Dokument feldorientiert vorliegen. Felder, wie beispielsweise Spalten in einer RDBS-Tabelle oder Felder eines Such-Index, können als Document-Attributes aufgefasst werden.
[23] Um dies zu erreichen, muss der Zusammenhang zwischen Dokumentenverbindungen und Quellenverbindungen beschrieben werden.
[24] Diese könnte auch als Instanziierung modelliert werden.
[25] Die Idee besteht darin, Attribut-Werte *nicht* in der Ontologie zu verwalten, sondern nur zu hinterlegen, welche Attribute ein Dokument/ eine Quelle besitzt. Die Attribut-Werte werden im Suchindex verwaltet, siehe 4.4.
[26] Mit „unbestimmte Relationen“ sind noch nicht näher definierte Relationen gemeint. Für das *ontology-design* ist es hier nicht notwendig, die Relationen genau zu benennen, da

bestehen (Abbildung 3 D und F). Zwischen Quellen und zugeordneten Dokumenten besteht die *document-of-source* Relation und zwischen Quellen und Document-Attributes eine **attribute-of-source** Relation (Abbildung 3 B).

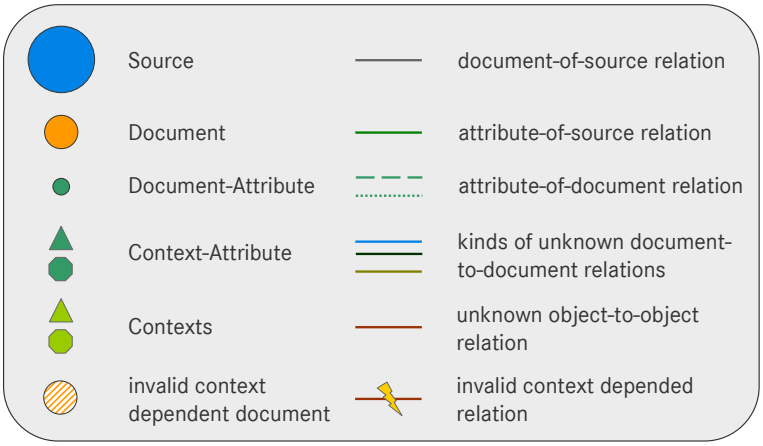

Abbildung 2: Ontologieelemente

Zwischen Dokumenten (egal welchen Quellen sie angehören) können ebenso *unbestimmte Relationen* (Abbildung 3 D-F) bestehen[27] und neben der schon genannten *document-of-source* Relation verfügen Dokumente über die **attribute-of-document** Relation (Abbildung 3 C), die Aufschluss darüber gibt, über welche Document-Attributes sie verfügen.
Document-Attributes können ebenfalls *unbestimmte Relationen* zu anderen Document-Attributes besitzen. Diese geben darüber Aufschluss, ob und wie Document-Attributes zueinander semantische Abhängigkeiten aufweisen (Abbildung 3 D-F).

Unter einem **Kontext** versteht man die Gültigkeit von Dokumenten und unter einem **Kontext-Attribut** den Definitionsbereich von Kontexten.
Wie schon weiter oben beschrieben, können Dokumente eines Dokumentensystems einem Informationsbedürfnis zugeordnet werden und in bestimmten Anwendungsszenarien noch durch Gültigkeiten beschränkt sein. Diese wirken direkt auf das Informationsbedürfnis, ändern dadurch die Relevanz der Dokumente signifikant und beeinflussen somit auch mögliche Verbindungen zu anderen Dokumenten. Der Kontext ordnet sich stets dem Informationsbedürfnis sowie dem Anwendungsszenario zu. Dokumente und Relationen

sich diese aus der Art der Verknüpfung der Dokumente ergeben. Wichtig ist, hier zu erkennen, dass diese Relationen existieren.

[27] In Abbildung 3 E sind beispielsweise attribuierte Relationen zwischen Dokumenten unterschiedlicher Dokumentensysteme zu sehen.

besitzen also eine Kontextabhängigkeit. Dennoch verfügen nicht alle Dokumente über die gleiche Abhängigkeit. Das muss bei der Modellierung der Ontologie beachtet werden.

Hier sei wieder auf die Domäne des After-Sales Bereiches von Mercedes-Benz Cars verwiesen. Im Abschnitt 1.1 wurde gezeigt, dass ein fahrzeugbezogenes Symptom durchaus mit Ersatzteilen einer möglichen Reparatur verknüpft werden kann. Falls ein standardisiertes Symptom auf zwei Werkstatt-Hilfe-Dokumente verweist, aber die jeweiligen Dokumente einem bestimmten Kontext zugeordnet sind, könnte sich folgendes darstellen:
Das Informationsbedürfnis steht im Kontext einer bestimmten Fahrzeugbaureihe. Von den beiden für relevant erachteten Werkstatt-Hilfe-Dokumenten besitzt aber nur das erste den gleichen Kontext (das zweite ist einer anderen Fahrzeugbaureihe zugeordnet). Somit entsprechen nicht mehr alle von den Werkstatt-Hilfe-Dokumenten referenzierten Ersatzteile dem Kontext des Informationsbedürfnisses. Context-Attributes „verhalten“ sich vermutlich ähnlich wie Document-Attri-butes: Beide sind Attribute und werden Quellen zugeordnet. Document-Attributes werden darüber hinaus auch Dokumenten zugeordnet, Context-Attributes hingegen nicht. Dokumenten werden *Instanzen* der Context-Attri-butes zugewiesen: Die Kontexte.

Anders als Document-Attributes beeinflussen Context-Attributes direkt die Gültigkeit der Dokumente. Die Unterscheidung zwischen *Attribut* und *Kontext* ist jedoch nicht trivial. Attribute können darüber Aufschluss geben, *wie* Dokumente (oder Quellen) miteinander verknüpft sind. Kontexte hingegen geben Auskunft darüber, *wann* (diese) Verknüpfungen *gültig* sind.

A sei die Menge aller Attribute, A_D die Menge aller Document-Attributes und A_K die Menge aller Context-Attributes, dann gilt:

$$A_D \subseteq A \text{ und } A_K \subseteq A$$

Im allgemeinen können zwischen Document– und Context-Attributes Überschneidungen existieren:

$$A_D \cap A_K \neq \emptyset$$

Je nach Anwendungskontext kann das aber auch ausgeschlossen werden.
Kontexte stehen also in anderer Art mit Dokumenten in Beziehung als Document-Attributes. Während Document-Attributes *nur* ausdrücken sollen, dass ein Dokument über ein Attribut verfügt, weisen die Kontexte dem Dokument einen konkreten Wert zu.

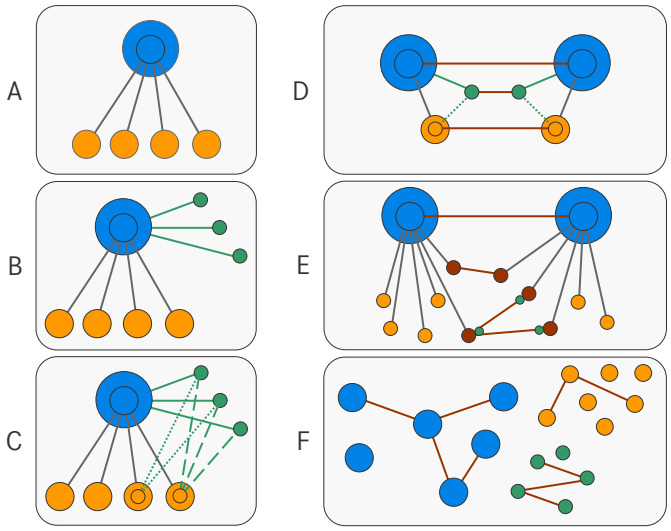

Abbildung 3: Ontologieelemente

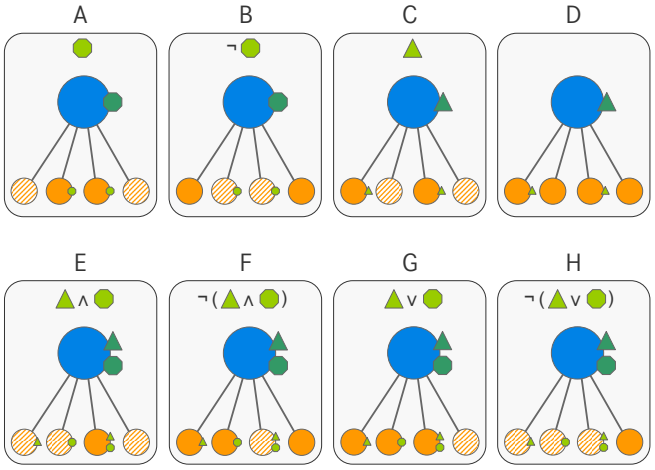

Abbildung 4: Quellen, Dokumente und Kontext

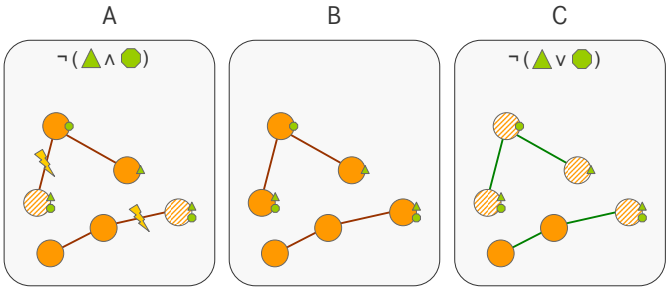

Abbildung 5: Relationen und Kontext

In Abbildung 4 A-H sind zwei Kontexte und deren Einfluss auf die Gültigkeit von Dokumenten gegeben. Bei Abbildung 4 A-C und 4 E-H sind kontextuelle Einschränkungen[28] auf Quellen und deren Wirken auf die jeweiligen Dokumente gegeben. Ist diese kontextuelle Einschränkung nicht gegeben (Abbildung 4 D), so beeinflusst der Kontext die Gültigkeit von Dokumenten *nicht*.

Auf Relationen zwischen Dokumenten besitzen Kontexte ebenfalls einen Einfluss. Dieser wird durch die Gültigkeit von Dokumenten direkt in die Relationen *getragen*.
Darüber hinaus existieren zwei Klassen von Relationen. Jene, die vom Kontext beeinflusst werden (Abbildung 5 A-B) und solche, die kontextunabhängig agieren (Abbildung 5 C).

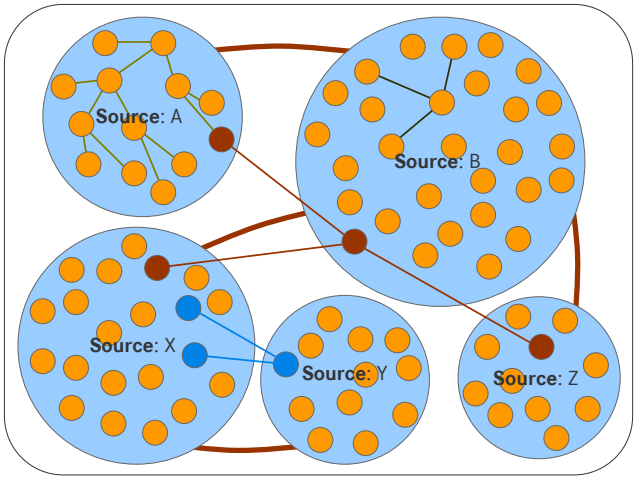

Abbildung 6: Ontologische Abbildung der heterogenen Dokumentenwelt (ohne Darstellung von Attributen und Kontexten). Dargestellt sind Quellen (die blauen Cluster *Source A*, *Source B*, *Source X*, *Source Y* und *Source Z*), Dokumente (meist orange Kreise in den Quellen) sowie Relationen zwischen Quellen (dicke rote Kanten zwischen Quellen) und typisierte Relationen zwischen Dokumenten (verschieden farbige Verbindungen zwischen Dokumenten). Bei quellenübergreifenden Dokumentenverbindungen wurden die beteiligten Dokumente in den Farben der Verbindungen dargestellt.

Im Beispiel der Abbildung 6 ist eine Ontologie mit fünf verschiedenen Dokumentensystemen und vier Relationsarten, die eine Navigation durch die Ontologie ermöglichen, skizziert.
Aus der in diesen Artikel vorgestellten Modellierung soll nun folgendes hervorgehen:

[28] Kontextuelle Einschränkungen korrelieren dabei direkt mit dem Informationsbedürfnis.

- Relationen zwischen Quellen geben Aufschluss darüber, welche *möglichen* semantischen Zusammenhänge es zwischen Dokumenten der Quellen gibt.
- Relationen zwischen Dokumenten geben darüber Aufschluss, welche der möglichen semantischen Zusammenhänge zwischen Quellen (oder der gleichen Quelle) wirklich existieren.
- Kontexte geben Aufschluss darüber, welche dieser wirklich existierenden Dokumentenverbindungen wann gültig sind.
- Relationen zwischen Attributen geben Aufschluss darüber, ob und wie Relationen zwischen Dokumenten attribuiert sind (Abbildung 3 E).

Über Zusammenhang, Abhängigkeiten sowie Gültigkeiten von Relationen zwischen Entitäten gibt das Regelwerk einen genaueren Einblick.

3.3 Regelwerk und formale Definitionen

Um die Ontologie im Anwendungsszenario nutzen zu können, sollte mit Hilfe eines Regelwerkes definiert werden, in welchem Rahmen sich Wissen generieren lässt. Ebenso sollten Bedingungen für das Erstellen der ontologischen Entitäten gegeben sein.

Um Relationen bewerten zu können und um gültige (und beste) Wege in der Ontologie zu „finden“, müssen *Wege* definiert und *Gewichte* zur Bewertung dieser Wege festgelegt werden.
In folgenden Herleitungen geben wir einen Vorschlag, wie Relationen, Gewichtungsfunktionen und Wege definiert sein können. Das Ziel ist es aufzuzeigen, wie man *beste Wege durch die Ontologie* beschreiben kann und welchen Einfluss Kontexte auf diese Wege besitzen.
Die genauen Bezeichnungen von Relationen und die Beschreibung von Gewichtungsfunktionen für Relationen und Wege sind in diesem Ansatz noch nicht gegeben und müssen in weiteren Arbeiten gefunden werden.

Seien $Q_1, Q_2, \ldots, Q_i, \ldots, Q_n$ Quellen und sei $\mathbf{d_i} \in \mathbf{Q_i}$ Dokument der Quelle $\mathbf{Q_i}$[29]. Ein Dokument sollte immer genau einer Quelle zugeordnet sein:

$$d \in Q_p \wedge d \in Q_q \rightarrow p = q$$

Sei $\mathbf{a} \in \mathbf{A_D}$ ein Attribut der Document-Attribute Menge A_D.

Sei $\mathbf{ka} \in \mathbf{A_K}$ ein Context-Attribute der Context-Attribute Menge A_K und $\mathbf{k}$ der Kontext des Context-Attribute ka. Für eine Quelle Q_p gelte:

[29] $d_i \in Q_i$ bezeichne hier die *document-of-source* Relation zwischen dem Dokument d_i und der Quelle Q_i.

$$Q_{kp} \subseteq Q_{kap} \subseteq Q_p$$

Sei $\mathbf{R_{Source}(Q_p, Q_q)}$ eine zweistellige Relation zwischen den Quellen Q_p und Q_q, mit $p \neq q$[30].

Sei $\mathbf{R(d_p, d_q)}$ eine zweistellige Relation[31] zwischen den Dokumenten d_p und d_q für $d_p \neq d_q$. Es gelte, falls zwei Dokumente verschiedener Quellen zueinander in Relation stehen, dann stehen auch die beiden Quellen zueinander in Relation.

$$R(d_p, d_q) \wedge p \neq q \rightarrow R_{Source}(Q_p, Q_q)$$

Sei $\mathbf{R_{attribute-of-source}(a, Q_p)}$ eine zweistellige Relation zwischen dem Attribut $a \in A_D$ und der Quelle Q_p. Es gelte, dass ein Document-Attribute nur einer Quelle zugeordnet wird:

$$R_{attribute-of-source}(a, Q_p) \wedge R_{attribute-of-source}(a, Q_q) \rightarrow p = q$$

Sei $\mathbf{R_{attribute-of-document}(a, d_p)}$ eine zweistellige Relation zwischen den Document-Attribute $a \in A_D$ und dem Dokument d_p der Quelle Q_p. Es gelte, wenn ein Document-Attribute einem Dokument einer Quelle zugeordnet wird, so wird es auch der Quelle des Dokumentes zugeordnet:

$$R_{attribute-of-document}(a, d_p) \rightarrow R_{attribute-of-source}(a, Q_p)$$

Sei $\mathbf{R_{Attribute}(a_1, a_2)}$ eine zweistellige Relation zwischen den Document-Attributes a_1 und a_2, mit $a_1, a_2 \in A_D$ und $a_1 \neq a_2$. Es gelte, falls eine Relation zwischen Document-Attributes zweier Dokumente besteht, dann stehen auch die beiden Dokumente zueinander in Relation:

$$R_{Attribute}(a_1, a_2) \wedge R_{attribute-of-document}(a_1, d_p) \wedge \\ R_{attribute-of-document}(a_2, d_q) \rightarrow R(d_p, d_q), \text{ für } d_p \neq d_q$$

Sei **f** eine Gewichtungsfunktion für Relationen zwischen Dokumenten, mit

$$f(R(d_p, d_q)) = \{x | 0 \leq x \leq 1, x \in \mathbb{R}\}$$

Sei **g** eine Gewichtungsfunktion für Relationen zwischen Quellen, mit

$$g(R_{Source}(Q_p, Q_q)) = \{x | 0 \leq x \leq 1, x \in \mathbb{R}\}$$

Sei **h** eine Gewichtungsfunktion für Relationen zwischen Document-Attributes, mit

$$h(R_{Attribute}(a_1, a_2)) = \{x | 0 \leq x \leq 1, x \in \mathbb{R}\}$$

[30] Hier sind die *noch unbenannten* Relationen zwischen Quellen gemeint.
[31] Das sind die *noch nicht benannten* Relationen zwischen Dokumenten.

$\mathbf{R_k(d_p, d_q)}$ heiße **Kontext-sensitive Relation** von d_p nach d_q für $d_p \neq d_q$, mit

$$R_k(d_p, d_q) := R(d_p, d_q)_{d_p \in Q_{kp}, d_q \in Q_{kq}}$$

$\mathbf{f_k(R_k(d_p, d_q))}$ heiße **Gewicht** einer **Kontext-sensitiven Relation**, mit

$$f_k(R_k(d_p, d_q)) = \{x | 0 \leq x \leq 1, x \in \mathbb{R}\}$$

$\mathbf{w_k(d_1, d_n)}$ heiße **Kontext-sensitiver Weg** von d_1 nach d_n, für $1 \neq n$ und $\oplus$ sei ein noch nicht weiter bestimmter Operator, mit

$$w_k(d_1, d_n) := R_k(d_1, d_2) \oplus \ldots \oplus R_k(d_{n-2}, d_{n-1}) \oplus R_k(d_{n-1}, d_n)$$

$\mathbf{w_{kshort}(d_1, d_n)}$ heiße **kürzester Kontext-sensitiver Weg** von d_1 nach d_n, für $1 \neq n$, mit

$$w_{kshort}(d_1, d_n) := R_k(d_1, d_n)$$

$\mathbf{f_k(w_k(d_1, d_n))}$ heiße **Gewicht** des **Kontext-sensitiven Weges** $w_k(d_1, d_n)$ für $1 \neq n$ und $\otimes$ sei ein nicht weiter bestimmter Operator, mit

$$f_k(w_k(d_1, d_n)) := f_k(R_k(d_1, d_2)) \otimes \ldots \otimes f_k(R_k(d_{n-2}, d_{n-1})) \otimes f_k(R_k(d_{n-1}, d_n))$$

Sei $\mathbf{W_k(d_1, d_n)}$ die **Menge aller Kontext-sensitiven Wege** $w_k(d_1, d_n)$ von d_1 nach d_n.

$\mathbf{f_{kmin}(w_k(d_1, d_n))}$ heiße **günstigster Weg** von d_1 nach d_n, mit

$$f_{kmin}(w_k(d_1, d_n)) := \\ min(\{x | x = f_k(w_k(d_1, d_n)), \forall w_k(d_1, d_n) \in W_k(d_1, d_n)\})$$

$\mathbf{w_{kmin}(d_1, d_n)}$ heiße **bester Kontext-sensitiver Weg** von d_1 nach d_n, mit

$$w_{kmin}(d_1, d_n) := f_{kmin}(w_k(d_1, d_n))$$

$\mathbf{w_{kmin}(d_p, Q_{kq})}$ heiße **beste Kontext-sensitive Wege** von d_p nach Q_q, mit

$$w_{kmin}(d_p, Q_{kq})_{d_p \in Q_{kp}} := \{w_{kmin}(d_p, d_q) | \forall d_q \in Q_{kq},\, mit\, w_k(d_p, d_q)\}$$

$\mathbf{R_{Context-Source}(ka, Q_p)}$ heiße **Kontext-Source-Relation** von Kontext-Attribut ka zu Quelle Q_p. $\mathbf{Q_{kap}}$ sei die durch Kontext-Attribute ka eingeschränkte Quelle $Q_{kap} \subseteq Q_p$, mit

$$Q_{kap} := R_{kaSource}(ka, Q_p)$$

$\mathbf{R_{kaSource}(Q_p, Q_q)}$ heiße **Kontext-sensitive Source-Relation** von Q_p nach Q_q für $p \neq q$, mit

$$R_{_{ka}Source}(Q_p, Q_q) := R_{Source}(Q_{_{ka}p}, Q_{_{ka}q})$$

$\mathbf{g_{ka}(R_{_{ka}Source}(Q_p, Q_q))}$ heiße **Gewicht** einer **Kontext-sensitiven Source-Relation**, mit

$$g_{ka}(R_{_{ka}Source}(Q_p, Q_q)) = \{x|0 \leq x \leq 1, x \in \mathbb{R}\}$$

$\mathbf{w_{_{ka}Source}(Q_1, Q_n)}$ heiße **Kontext-sensitiver Weg** von Q_1 nach Q_n für $1 \neq n$ und $\oplus$ sei ein noch nicht weiter bestimmter Operator, mit

$$w_{_{ka}Source}(Q_1, Q_n) := \\ R_{_{ka}Source}(Q_1, Q_2) \oplus \ldots \oplus R_{_{ka}Source}(Q_{n-2}, Q_{n-1}) \oplus R_{_{ka}Source}(Q_{n-1}, Q_n)$$

$\mathbf{w_{_{ka^{short}}Source}(Q_1, Q_n)}$ heiße **kürzester Kontext-sensitiver Weg** von Q_1 nach Q_n für $1 \neq n$, mit

$$w_{_{ka^{short}}Source}(Q_1, Q_n) := R_{_{ka}Source}(Q_1, Q_n)$$

$\mathbf{g_{ka}(w_{_{ka}Source}(Q_1, Q_n))}$ heiße **Gewicht** des **Kontext-sensitiver Weges** $w_{_{ka}Source}(Q_1, Q_n)$ von Q_1 nach Q_n für $1 \neq n$ und $\otimes$ sei ein noch nicht weiter bestimmter Operator, mit

$$g_{ka}(w_{_{ka}Source}(Q_1, Q_n)) := \\ g_{ka}(R_{_{ka}Source}(Q_1, Q_2)) \otimes \ldots \otimes g_{ka}(R_{_{ka}Source}(Q_{n-2}, Q_{n-1})) \otimes \\ g_{ka}(R_{_{ka}Source}(Q_{n-1}, Q_n))$$

Sei $\mathbf{W_{_{ka}Source}(Q_p, Q_q)}$ die **Menge aller Kontext-sensitiven Wege** $w_{_{ka}Source}(Q_p, Q_q)$ von Q_p nach Q_q.

$\mathbf{w_{_{ka}Source^{min}}(Q_p, Q_q)}$ heiße **bester Kontext-sensitiver Weg** von Q_p nach Q_q für $p \neq q$, mit

$$w_{_{ka^{Source}min}}(Q_p, Q_q) := \\ min(\{x|x = g_{ka}(w_{_{ka}Source}(Q_p, Q_q)), \forall w_{_{ka}Source}(Q_p, Q_q) \in \\ W_{_{ka}Source}(Q_p, Q_q)\})$$

$\mathbf{w_{Attribute}(a_1, a_n)}$ heiße **Attribut-Weg** von Attribut a_1 zu Attribut a_n für $a_1 \neq a_n$ und $\oplus$ sei ein noch nicht weiter bestimmter Operator, mit

$$w_{Attribute}(a_1, a_n) := \\ R_{Attribute}(a_1, a_2) \oplus \ldots \oplus R_{Attribute}(a_{n-2}, a_{n-1}) \oplus R_{Attribute}(a_{n-1}, a_n)$$

$\mathbf{w_{short\,Attribute}(a_1, a_n)}$ heiße **kürzester Attribut-Weg** von Attribut a_1 zu Attribut a_n für $a_1 \neq a_n$, mit

$$w_{Attribute}(a_1, a_n) := R_{Attribute}(a_1, a_n)$$

Sei $\mathbf{W_{Attribute}(a_1, a_n)}$ die **Menge aller Attribut-Wege** $w_{Attribute}(a_1, a_n)$ von Attribut a_1 zu Attribut a_n.

$\mathbf{h(w_{Attribute}(a_1, a_n))}$ heiße **Gewicht** des **Attribut-Weges** $w_{Attribute}(a_1, a_n)$ und $\otimes$ sei ein noch nicht weiter bestimmter Operator, mit

$$h(w_{Attribute}(a_1, a_n)) := h(R_{Attribute}(a_1, a_2)) \otimes \ldots \otimes h(R_{Attribute}(a_{n-2}, a_{n-1})) \otimes h(R_{Attribute}(a_{n-1}, a_n))$$

$\mathbf{w_{minAttribute}(a_1, a_n)}$ heiße **bester Attribut-Weg** von Attribut a_1 zu Attribut a_n für $a_1 \neq a_n$, mit

$$w_{minAttribute}(a_1, a_n) := min(\{x | x = h(w_{Attribute}(a_1, a_n)), \forall w_{Attribute}(a_1, a_n) \in W_{Attribute}(a_1, a_n)\})$$

4 Advanced Ontology-based Information Retrieval System (AIRS)

Unter Information Retrieval versteht man frei nach [17] das Finden von Informationen (welche ein bestimmtes Informationsbedürfnis erfüllen) in einer (großen) unstrukturierten Menge.

Unter einer unstrukturierten Menge verstehen wir eine durch ein Dokumentensystem repräsentierte Dokumentenkollektionen.
Finden bedeutet in diesem Zusammenhang das auf ein Retrieval-Modell beruhende und kontextabhängige Verknüpfen von Informationsbedürfnis mit einer Untermenge der Dokumentenkollektion. Ein *Infromation Retrieval System* (IRS) ist demnach ein System, welches das *Finden* von Dokumenten unterstützt.

Wie in Abschnitt 1 beschrieben, existiert in Unternehmen meist eine heterogene Dokumentenlandschaft mit unterschiedlichen Dokumentensystemen. Dies limitiert das *Finden* von Dokumenten mit Hilfe herkömmlicher Dokumentenretrievalsysteme, da nur „relevante“, nicht aber „in Relation stehende“ Dokumente gefunden werden[32].

Im Rahmen dieses Artikels sei deswegen eine Erweiterung eines IRS um ein ontologisches Modell der heterogenen Dokumentenlandschaft vorgeschlagen[33].

In diesem Sinn verstehen wir unter einem **Advanced Ontology-based**

[32] Zusätzlich kann eine Suche meist nur über *einem* Dokumentensystem durchgeführt werden.

[33] Andere Arbeiten befassen sich ebenfalls mit dem ontologiebasierten oder konzeptbasierten Information Retrieval. So finden sich unter anderem Ansätze in [23] und in [25].

Information Retrieval System ein IRS, welches um eine Ontologiekomponente zur Retrievalsteuerung und Dokumentenverknüpfung erweitert ist. Abbildung 7 zeigt den prinzipiellen Aufbau eines solchen Systems und illustriert seine Arbeitsweise.

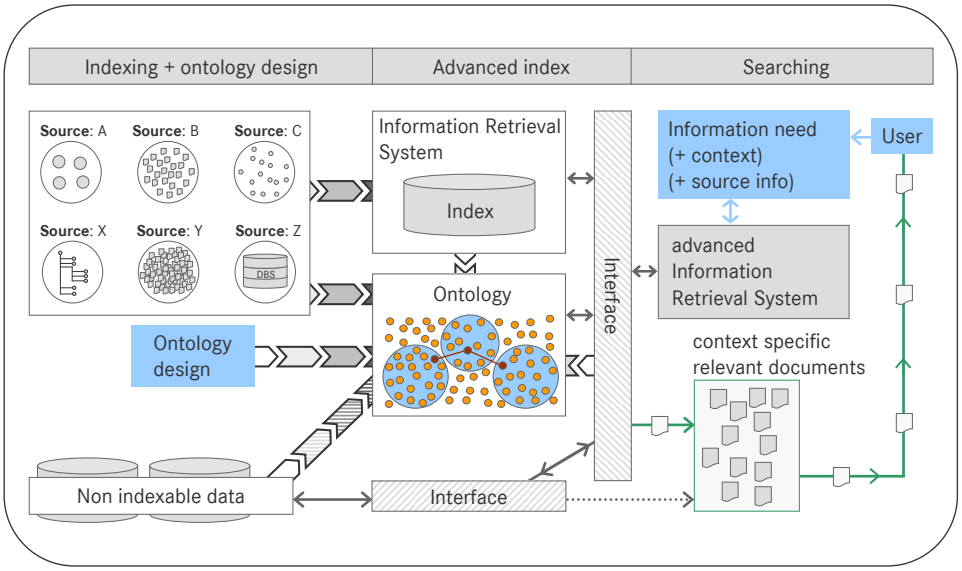

Abbildung 7: Aufbau und prinzipielle Arbeitsweise eines AIRS. Im Bereich „Indexing + ontology design“ (siehe Abschnitt 4.1) werden im Indexierungsprozess die Dokumentensysteme (soweit möglich) indexiert und im *Ontology design* eine der heterogenen Dokumentenlandschaft entsprechende Ontologie generiert. Der erweiterte Index („Andavced index“) kann danach dem Prozess der ontologiegetriggerten Suche („Searching“, siehe Abschnitt 4.2) zugeführt werden.

Wie bei herkömmlichen IRS unterscheidet man beim AIRS zwischen zwei voneinander getrennten Workflows: Der Indexierung und der Suche.
Die entscheidende Frage hierbei ist, wie Ontologiekomponente und IRS zusammenwirken.

4.1 AIRS: Workflow Indexierung

Der Indexierungsvorgang besteht aus zwei Bestandteilen: Zum Einen müssen die einzelnen Dokumentensysteme indexiert (soweit dies möglich ist) und zum Anderen muss eine, die heterogene Dokumentenlandschaft beschreibende, Ontologie erstellt werden. So müssen im *Ontology design* (Abbildung 7, blauer Kasten, Mitte links) sowohl die indexierbaren als auch die nichtindexierbaren Quellen in der Ontologie modelliert und durch einen Doku-

mentenverarbeitungsprozess[34] befüllt werden.

Dieser Prozess muss auch das Finden von Dokumentenverbindungen beinhalten, um auf Ontologieebene Relationen beschreiben zu können.

Ein weiterer wichtiger Arbeitsschritt sowohl des Indexierungsvorganges als auch des Ontologieerstellungsprozesses besteht darin, Indexdokumente mit zugehörigen Ontologieindividuen zu verbinden. Dies ist nötig, um ein Zusammenspiel von IRS und Ontologie zu ermöglichen.

4.2 AIRS: Workflow Suche

Die Abbildung 7 betrachtend ergibt sich folgender Such-Workflow: Ein *User* spezifiziert durch eine Anfrage sein Informationsbedürfnis. Optional legt er ebenso den *Kontext*[35] und die *Quelleninformationen* fest.
Der Kontext wirkt sich auf die Antwort des AIRS dahingehend aus, dass er *nur* (für den Kontext) als gültig erachtete Dokumente in der Antwortmenge zulässt[36]. Die Quelleninformation beeinflusst hingegen die Berechnung der Anfragestrategie.

So können durch die Quelleninformation **Start– und Zielquelle** definiert werden. In diesem Fall *weiß* das AIRS, dass das Informationsbedürfnis des Users mit der Startquelle zu verknüpfen ist und als Ergebnis Dokumente der Zielquelle ausgegeben werden sollen. Die Aufgabe der Ontologiekomponente ist in diesem Fall das *Finden möglicher Verknüpfungen auf Quellenebene*[37].
Die Aufgabe des IRS besteht darin, Dokumente der Startquelle zu liefern. Jene Dokumente resultieren aus der Verknüpfung von Informationsbedürfnis mit den indexierten Dokumenten der Startquelle.
Die so gefundenen Dokumente wären dann Ausgangspunkt für einen weiteren Arbeitsschritt[38], der[39] gültige und beste Wege auf Dokumentenebene – hin zur Zielquelle – geht. Dies wäre wiederum eine Aufgabe der Ontologiekompnente. Letztendlich werden die so *gefundenen* Dokumente durch das AIRS dem User als Antwortmenge übergeben.

Ist die **Startquelle**, nicht aber die Zielquelle gegeben, dann *muss* das Informationsbedürfnis mit der Startquellen verknüpft werden (IRS Komponente)

[34] Der Prozess sollte möglichst automatisiert, zumindest semi-automatisiert werden.
[35] Genau genommen könnte der Kontext auch durch das System selbst bestimmt werden: Zum Beispiel durch Erhebung von Umgebungsdaten.
[36] Das Entfallen des Kontextes bewirkt dementsprechend *keine* Beeinflussung der Antwortmenge.
[37] Es müsste also der *beste* Weg von Quelle X zur Quelle Y gefunden werden.
[38] Welche der gefundenen Dokumente für weitere Arbeitsschritte *relevant* sind, könnte durch das AIRS oder durch den User selbst entschieden werden.
[39] Ausgehend von den durch die Ontologiekomponente gefundenen Wegen auf Quellenebene.

und als Ergebnis könnten als relevant erachtete Dokumente (IRS Komponente), mit *ausgewählten* Verbindungen zu anderen Dokumenten (Ontologiekomponente), präsentiert werden.

Wenn aber nur die **Zielquelle** gegeben ist, so muss das AIRS selbst eruieren, mit welcher Quelle das Informationsbedürfnis verknüpft werden kann (IRS Komponente), um anschließend die günstigsten Wege hin zur Zielquelle zu finden (Ontologiekomponente).

Ist **weder** die **Start- noch Zielquelle** gegeben, werden (entsprechend den vorherigen beiden Fällen) durch das AIRS die Quellen bestimmt, mit denen das Informaionsbedürfnis verknüpft werden kann (IRS Komponente). Als Ergebnis werden alle als relevant erachteten Dokumente (IRS Komponente) mit ihren Verbindungen zu anderen Dokumenten (Ontologiekomponente) ausgegeben.

Im nächsten Abschnitt werden die Aufgaben der beiden Bestandteile (IRS Komponente und Ontologiekomponente) zusammengefasst.

4.3 AIRS: Ontologie-Komponente

Finden möglicher Verknüpfungen auf Quellenebene.
Nach Abschnitt 3.3 besteht diese Aufgabe (formal beschrieben) aus dem Finden des *besten Kontext-sensitiven Weges* $w_{_{ka}Source min}(Q_p, Q_q)$ von Quelle Q_p zur Quelle Q_q.
Dafür ist es nötig, Relationen zwischen Quellen zu erkennen und diese (Relationen) zur Wegfindung zu nutzen. Anschließend müssen die so gefundenen Wege mit einer Gewichtungsfunktion $g_{ka}(w_{_{ka}Source}(Q_1, Q_n))$ bewertet und der *beste* Weg (falls überhaupt vorhanden) gewählt werden.

Gehen gültiger Wege auf Dokumentenebene.
Diese Aufgabe lässt sich ebenfalls nach Abschnitt 3.3 formal beschreiben und besteht im Finden des *besten Kontext-sensitiven Weges* $w_{_{k}min}(d_1, d_n)$ von Dokument d_1 zu Dokument d_n oder im Bestimmen der *besten Kontext-sensitiven Wege* $w_{_{k}min}(d_p, Q_{_{k}q})$ von Dokument d_p zu Dokumenten der Quelle $Q_{_{k}q}$.
So ist es möglich, anhand bestimmter Relationen und Kontexte durch die gegebene Dokumentenwelt zu navigieren[40].

Erfassen nicht-indexierbarer Quellen und Dokumente.
In Unternehmen existieren nicht nur Dokumentensysteme, deren Dokumente man in einen IRS erfassen kann. Einige Dokumente können *nur* über

[40] Relationen zwischen Dokumenten vorausgesetzt.

definierte Schnittstellen aus anderen Systemen abgefragt werden. Verknüpfungen dieser Dokumente zu anderen (im Index des IRS gehaltenen Dokumenten) müssen in der Ontologie erfasst werden. Auf diese Weise können die nur über die Schnittstellen abzufragenden Systeme in den AIRS-Workflow integriert werden.

***Lernen* von Relationen.**
Nutzer-Feedback könnte durch das AIRS erfasst und in die Ontologie integriert werden. *Erkennt* ein Nutzer des Systems im Rahmen einer Dokumentenrecherche eine Verbindung zwischen zwei beliebigen Dokumenten, welche noch nicht im System erfasst ist, so sollte diese Verbindung in der Ontologie als Relation zwischen den beiden Dokumenten hinterlegt und bei späteren Abfragen berücksichtigt werden.
Die Aufgabe besteht in diesem Fall darin, die Relation $R(d_p, d_q)$ zwischen den beiden Dokumenten d_p und d_q zu benennen und ein Gewicht $f(R(d_p, d_q))$ zu bestimmen.

4.4 AIRS: IRS-Komponente

Verknüpfen von Informationsbedürfnis und Dokumenten.
Die eigentliche Aufgabe des IRS besteht in der Funktion der Suche über textreichen Dokumentenkollektionen und Attributmengen von Dokumenten. Wenn die Anfragestrategie durch das AIRS und mit Hilfe der Ontologiekomponente bestimmt wurde, kann das IRS dazu benutzt werden, um in den indexierten Daten eines bestimmten Dokumentensystems nach relevanten Dokumenten zu suchen.
Die *relevantesten* dieser Dokumente können dann für weitere Arbeitsabläufe verwendet werden.

5 Zusammenfassung und Ausblick

In diesem Artikel haben wir unseren Ansatz des AIRS vorgestellt. Es wurde beschrieben, wie die beiden AIRS-Komponenten (IRS-Komponente und Ontologiekomponente) zusammenwirken müssen, um eine dokumentensystemübergreifende intelligente Suche realisieren zu können. Dazu wurden theoretische Aspekte der Ontologiemodellierung betrachtet und auf die dazu nötige Beachtung des Anwendungskontextes eingegangen.
Für weitere Schritte ist wichtig, welche speziellen Herausforderungen an eine Ontologie und für einen gewinnbringenden Einsatz dieser im Ansatz des AIRS existieren. Darüber hinaus müssen Anforderungen und Forschungsschwerpunkte zur Realisierung des AIRS-Konzeptes benannt werden. Im Folgenden beschreiben wir offene Fragestellungen der nachgelagerten Forschungsschwerpunkte *Ontologiemodellierung* und *AIRS-Umsetzung*.

5.1 Ontologiemodellierung

Forschungsschwerpunkte der Ontologiemodellierung existieren im Umfeld der Ontologierepräsentation. Andere ergeben sich aus (bisher) unbeantworteten Fragestellungen bezüglich des Regelwerkes und nicht benannter Ontologieelemente. Darüber hinaus stellt sich die Frage nach der Möglichkeit, ontologische Strukturen aus Dokumentensystemen abzuleiten.

5.1.1 Ontologiesprachen

Ontologien können auf verschiedene Arten und durch verschiedene Sprachen beschrieben werden.
Denkbar wären auch proprietäre Ansätze auf programmatischer Ebene für spezifische Anwendungen.
Daneben stellen Graph-Datenbanken[41] einen guten Kompromiss zwischen Modellierung von (und Zugang zu) Ontologien dar.
Geht es hingegen um Austauschbarkeit und Formalisierung, sei besonders auf die im *Semantic Web* Umfeld entstandenen Ontologiesprachen verwiesen.
Zwei Beispiele sind die XML-Derivate der Ontologiesprachen RDF/RDFS [2] und OWL [1][42]. Eine Übersicht über weitere Ontologiesprachen des *Semantic Web* Umfeldes findet sich in [19] und [14].
Darüber hinaus existiert mit dem Ansatz der Topic-Maps [18] eine Art der Wissensrepräsentation, die den Anspruch einer subjektorientierten Modellierung von Wissen besitzt und durch ihr Konzept der Quellenreferenzen durchaus zur Modellierung von heterogenen Dokumentenlandschaften geeignet ist.

Welcher Ansatz der Wissensrepräsentation für die in diesem Artikel designete Ontologie am Besten geeignet ist, müssen wir in nachgelagerten Schritten erst noch eruieren.

5.1.2 Ontologieelemente und ontologisches Regelwerk

Durch die im Abschnitt 3.2 beschriebenen Ontologieelemente und das im Abschnitt 3.3 vorgeschlagene Regelwerk lässt sich der Anspruch ableiten, dass stets die besten Kontext-sensitiven Wege auf Dokumentenebene und die kostengünstigsten Wege auf Quellenebene zu finden sind. Der Anspruch der Kontextabhängigkeit findet sich in der Beschreibung der Regeln wieder. Allerdings bleiben einige Fragestellungen offen, die in weiteren Untersuchungen beantwortet werden sollten:

[41] Wie beispielsweise Neo4J (siehe http://neo4j.org/, Version vom 22.03.2011).

[42] So findet sich unter anderem in [20] ein Ansatz zur Modellierung von Ontologien in RDF/RDFS. In [15] wird beschrieben, wie sich RDF/RDFS oder OWL Ontologien aus (unter anderem) inkonsistenten Taxonomien erstellen lassen.

1. Dokumente stehen dann in starker Relation zueinander, wenn die Relation (zwischen ihnen) anhand einer festzulegenden Gewichtungsfunktion maximal wird, der Abstand zwischen ihnen aber minimal ist. Dabei kann sich eine Relation zwischen Individuen über mehrere Ebenen erstrecken. Festzulegen und zu evaluieren sind Maße und Gewichtungsfunktionen. Dabei stellt sich auch die Frage, wie ein lernendes System Gewichte für Beziehungen beeinflussen kann (und sollte)?

2. Wo liegt die Trennung zwischen Kontext und Attribut?

3. Kontexte wirken auf die Gültigkeiten von Dokumenten und Relationen. Es wurde auch vermutet, dass Kontexte mit Attributen in Verbindung stehen. Unklar ist jedoch, wie Kontexte zu modellieren sind[43].

4. Fragen nach Einfluss auf die Gewichtungsfunktion und nach der Gültigkeit von Kontexten selbst (Kontext existiert, hat aber keinen Einfluss oder Kontext beeinflusst Kontext) müssen beantwortet werden.

5.1.3 Ableiten ontologischer Strukturen aus verschiedenen Dokumentensystemen

Aus der Modellierung der heterogenen Dokumentenlandschaft ergibt sich nun die Frage, wie eigentlich genau die einzelnen Dokumentensysteme ontologisch abzubilden sind. Hier soll ein kurzer Einblick gegeben werden, wie dies geschehen könnte.

In Abbildung 8 A ist ein Beispiel gegeben, wie aus einer Indexstruktur[44] Quellen, Dokumente und Attribute extrahiert werden können. Dokumente können direkt aus der Indexstruktur abgeleitet werden, indem man sie den Indexdokumenten gleichsetzt. Da Indexfelder in der Regel attribuiert vorliegen, müssen diese Felder auch als Attribute modelliert werden. Dennoch sollten nicht alle Attribute direkt übernommen werden, sondern nur solche, die ein Verknüpfungspotential aufweisen. Welche das genau sind, muss im Rahmen des Anwendungskontextes entschieden werden[45].

In Abbildung 8 B wird gezeigt, wie eine taxonomische Struktur in der On-

[43] Kontexte bilden eine noch nicht näher bestimmte Menge von Gültigkeitseinschränkungen. Unklar ist unter anderem, über wie viele Elemente diese Menge verfügt. Je nach Anwendungskontext könnten es durchaus hunderte oder tausende sein. Sollten der Kontexte modelliert werden, muss dies als Attributmenge geschehen, die eine Menge von Dokumenten aufgrund bestimmter Kriterien segmentiert. Diese Untermengen wirken sich vor allem in der Art auf Relationen aus, dass sie diese entweder egalisieren oder verschieden gewichten.

[44] Die Struktur könnte auch ein RDBS sein.

[45] Zu klären wäre unter anderem noch, ob die Attribute kontextuelle Abhängigkeiten der Dokumente offenbaren.

tologie abgebildet werden kann[46]. Wieder müssen im Dokumentensystem[47] Dokumente und Attribute lokalisiert werden.
Der Dokumentendefinition folgend (siehe Abschnitt 1.1) stellen die Taxonomiekonzepte informationstragende Einheiten dar und werden dementsprechend als Dokumente modelliert.
Dokumente verfügen über Attribute. Bei der Taxonomie könnten Attribute aus Eigenschaften der Konzepte abgeleitet (oder über diesen generiert) werden.

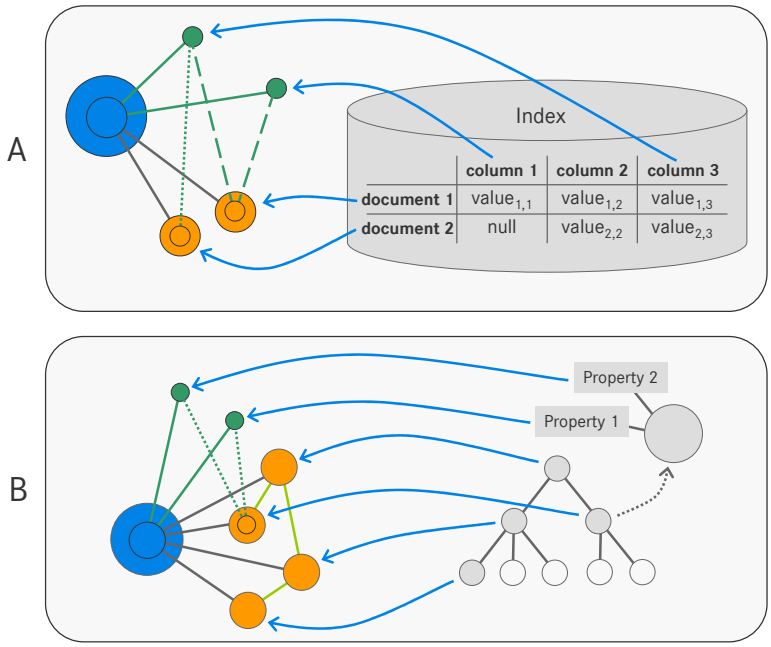

Abbildung 8: Ableiten ontologischer Strukturen aus verschiedenen Datenquellen.

Auf diese Weise ist es möglich, über den einzelnen Dokumentensystemen ein einheitliches ontologisches Modell zu generieren, welches dann für verschiedene Anwendungsszenarien Verwendung finden kann[48] – beispielsweise in einer erweiterten ontologiebasierten Dokumentenrecherche.
Wie allerdings Relationen extrahiert und Kontexte abgeleitet werden können, muss in weiteren Untersuchungen eruiert werden.

[46] Genau genommen ist eine Taxonomie selbst eine ontologische Struktur. Jedoch geht es darum, diese in die hier modellierte Ontologie zu integrieren.
[47] Hier: eine taxonomische Ordnung von Konzepten.
[48] Allerdings ist es nicht trivial diesen Prozess zu automatisieren. Hier müssen verschiedene Verfahren validiert werden, um Verbindungen zwischen den Dokumenten zu erkennen.

Finden, Benennen und Gewichten von Relationen zwischen Dokumenten.
Die Herausforderung besteht darin, Verfahren und Methoden zu evaluieren, mit denen es möglich ist (semi-) automatisiert Verbindungen zwischen (beliebigen) Dokumenten von (verschiedenen) Dokumentensystemen zu finden. Anhand dieser Verbindungen müssen semantische Relationen benannt werden. Im weiteren Schritt müssen diese Relationen im Hinblick auf ihre Ausdrucksstärke bewertet und eine Gewichtungsfunktion $f(R(d_p, d_q))$ bestimmt werden[49]. Darüber hinaus müssen Fragen beantwortet werden, die Relationsfindung direkt betreffen:

- Sind Quellen überhaupt potent in den Informationsfluss involviert und sind Informationsflüsse attribuiert (zu Attribuieren)?
- Welches Wissen benötigt man eigentlich, um von A nach B zu gelangen und welche Einflussfaktoren existieren (auch zwischen den Quellen)?
- Wie ist dieses Wissen durch Relationen zu benennen und wie sind Relationen zu gewichten?
- Durch welche Methoden kann man Verbindungen zwischen Dokumenten (semi-) automatisiert und in welcher Qualität finden?

5.2 AIRS-Umsetzung

In weiteren Untersuchungen müssen verschiedene Ansätze des ontologiebasierten Information Retrievals evaluiert und ein Architekturvorschlag erarbeitet werden. In späteren Arbeiten sollten dann Referenzimplementierungen angestrebt werden. Dazu ist es unter anderem nötig, folgende Aufgabenstellungen zu bewältigt:

Wegfindung
Wege[50] können durch Gewichtungsfunktionen *bewertet* werden. Die Anforderung besteht darin, *beste* Wege zu finden. Zu eruieren sind in diesem Fall Gewichtungsfunktionen zur Berechnung von Wegen durch die Ontologie. Wie genau funktioniert Wegfindung in der Praxis?

Konsistenz der AIRS-Komponenten
Wie lassen sich Index und Ontologie in einem konsistenten Zustand halten? Es muss ein Prozess erarbeitet werden, mit dessen Hilfe es möglich ist, Indexdokumente und die zugehörigen modellierten Individuen der Ontologie (im Hinblick auf die Ursprungsdaten) zu versionieren.

[49] Weiterhin ist zu überprüfen, ob die Relationen kontextabhängig agieren.
[50] Das können Wege durch die Ontologie auf Dokumentebene, Attributebene oder Quellenebene sein.

Literatur

[1] Sean Bechhofer, Frank van Harmelen, Jim Hendler, Ian Horrocks, Deborah McGuinness, Peter Patel-Schneijder, and Lynn Andrea Stein. Owl web ontology language reference. Recommendation, World Wide Web Consortium (W3C), February10 2004. See `http://www.w3.org/TR/owl-ref/`.

[2] Dave Beckett. Rdf/xml syntax specification (revised). W3C Recommendation, 10 February 2004. `http://www.w3.org/TR/2004/REC-rdf-syntax-grammar-20040210/`.

[3] Ronald Brachman and Hector Levesque. *Knowledge Representation and Reasoning*. Morgan Kaufmann, Amsterdam, 2004.

[4] S. Braun, A. Schmidt, and A. Walter. Ontology maturing: a collaborative web 2.0 approach to ontology engineering. In *Proceedings of the WWW Workshop on Social and Collaborative Construction of Structured Knowledge*, Banff, Canada, 2007.

[5] O. Corcho. Ontology-based document annotation: Trends and open research problems. *International Journal on Metadata*, Semantics and Ontologies(Volume 1):Issue 1, 2006.

[6] Randall Davis, Howard E. Shrobe, and Peter Szolovits. What is a knowledge representation? *AI Magazine*, 14(1):17–33, 1993.

[7] Gerard de Melo and Stefan Siersdorfer. Multilingual text classification using ontologies. In Giambattista Amati, Claudio Carpineto, and Giovanni Romano, editors, *ECIR*, volume 4425 of *Lecture Notes in Computer Science*, pages 541–548. Springer, 2007.

[8] M. Farhoodi, M. Mahmoudi, A.M.Z. Bidoki, A. Yari, and M. Azadnia. Query Expansion Using Persian Ontology Derived from Wikipedia. *World Applied Sciences Journal*, 7(4):410–417, 2009.

[9] D.H. Fischer. Ein Lehrbeispiel fur eine Ontologie: OpenCyc. *Information Wissenschaft und Praxis*, 55(3):139–142, 2004.

[10] F. Giunchiglia and I. Zaihrayeu. Lightweight ontologies. Technical Report DIT-07-071, University of Trento, 2007.

[11] Thomas R. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220, 1993.

[12] N. Guarino. Understanding, building and using ontologies. *Int. Journal Human-Computer Studies*, 45(2/3), 1997.

[13] Nicola Guarino. Formal ontology and information systems. pages 3–15. IOS Press, 1998.

[14] Jorge Rafael Gutierrez-Pulido, M. A. G. Ruiz, Roberto Herrera, E. Cabello, Steve Legrand, and Dave Elliman. Ontology languages for the semantic web: A never completely updated review. *Knowl.-Based Syst.*, 19(7):489–497, 2006.

[15] Martin Hepp and Jos de Bruijn. Gentax: A generic methodology for deriving owl and rdf-s ontologies from hierarchical classifications, thesauri, and inconsistent taxonomies. In Enrico Franconi, Michael Kifer, and Wolfgang May, editors, *ESWC*, volume 4519 of *Lecture Notes in Computer Science*, pages 129–144. Springer, 2007.

[16] Clyde W. Holsapple and K. D. Joshi. A collaborative approach to ontology design. *Commun. ACM*, 45(2):42–47, 2002.

[17] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, Cambridge, UK, 2008.

[18] Steve Pepper. The TAO of Topic Maps: finding the way in the age of infoglut. In *Proceedings of XML Europe*, 2000.

[19] Asuncion G. Perez and Oscar Corcho. Ontology Languages for the Semantic Web. *IEEE Intelligent Systems*, Jan/Feb:55–61, 2002.

[20] S. Staab, M. Erdmann, A. Mädche, and S. Decker. An Extensible Approach for Modeling Ontologies in RDF(S). In *First Workshop on the Semantic Web at the Fourth European Conference on Digital Libraries, Lisbon, Portugal*, September 18–20, 2000.

[21] D. Stenmark. The relationship between information and knowledge. In *Proceedings of IRIS*, volume 24, pages 11–14. Citeseer, 2001.

[22] York Sure, Marc Ehrig, and Rudi Studer. Automatische wissensintegration mit ontologien. In *Modellierung für Wissensmanagement*. Institut AIFB, Ulrich Reimer and Knut Hinkelmann, 2006.

[23] Stein Tomassen. Research on ontology-driven information retrieval. In Robert Meersman, Zahir Tari, and Pilar Herrero, editors, *On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops*, volume 4278 of *Lecture Notes in Computer Science*, pages 1460–1468. Springer Berlin / Heidelberg, 2006.

[24] M. Uschold and M. King. Towards a methodology for building ontologies. In *Workshop on Basic Ontological Issues in Knowledge Sharing, held in conjunction with IJCAI-95*, Montreal, Canada, 1995.

[25] D. Vallet, M. Fernandez, and P. Castells. An ontology-based information retrieval model. In *Proceedings of the European Semantic Web Conference (ESWC)*, Crete, Greece, 2005.

[26] Chaim Zins. Conceptual approaches for defining data, information, and knowledge. *JASIST*, 58(4):479–493, 2007.