Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorarbeit

Stefan Wiese

Suche in Peer-to-Peer-Netzwerken - Ein ontologiebasierter Ansatz

Fakultät Technik und Informatik
Department Informatik

Faculty of Engineering and Computer Science
Department of Computer Science

Stefan Wiese

Suche in Peer-to-Peer-Netzwerken - Ein ontologiebasierter Ansatz

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung
im Studiengang Angewandte Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer : Prof. Dr. Olaf Zukunft
Zweitgutachter : Prof. Dr.-Ing. Martin Hübner

Abgegeben am 30. April 2008

Stefan Wiese

Thema der Bachelorarbeit

Suche in Peer-to-Peer-Netzwerken - Ein ontologiebasierter Ansatz

Stichworte

Netzwerk, Peer-to-Peer, Chord, Ontologie, Suche

Kurzzusammenfassung

Peer-to-Peer-Netzwerke, die auf verteilten Hash-Tabellen basieren, bestechen durch ihre Robustheit und Selbstorganisation. Die dezentralen Strukturen können aber die Lokalisierung von Ressourcen erschweren. Diese Arbeit befasst sich mit der Idee, das Wissen aus Ontologien für eine Suchoptimierung heranzuziehen. Ausgangspunkt ist dabei der Lösungsansatz des Papers „Semantic Driven Hashing (SDH):An Ontology-based Search Scheme for the Semantic Aware Network (SA Net)“ [Sangpachatanaruk und Znati (2004)]. Er wird einer Analyse unterzogen und ist Basis für einen Prototyp, der die Verwendung von Ontologien realisiert, ohne das Peer-to-Peer-Paradigma zu verletzen.

Stefan Wiese

Title of the paper

Searching in peer-to-peer networks - An ontology-based approach

Keywords

Network, Peer-to-Peer, Chord, Ontology, Search

Abstract

The appeal of peer-to-peer-networks based on distributed hash-tables lies in their robustness and self-organization. As the localization of resources can be hindered by their decentralized structures this thesis follows the idea of using ontology-based knowledge for an improvement of existing search algorithms. Referring to initial findings of Sangpachatanaruk/Znati during the Proceedings of the Fourth International Conference on Peer-to-Peer Computing [Sangpachatanaruk und Znati (2004)] the author of this thesis develops and implements a prototype which allows the usage of ontologies without compromising the peer-to-peer-paradigm.

Inhaltsverzeichnis

Abbildungsverzeichnis

1. Einführung

1.1. Motivation

Peer-to-Peer-Netzwerke nehmen auf dem Informatikgebiet der Rechnernetze und im Speziellen der Netzwerkarchitekturen seit ihrer Einführung eine Sonderstellung ein [Mahlmann und Schindelhauer (2007)]. Sie bieten Eigenschaften, die dem klassischen Client-Server-Modell fehlen. Besonders attraktiv macht sie ein Minimum an zentralen Strukturen. Ein *Single-Point-of-Failure* wird vermieden und Angriffspunkte verringert.
Bei der Thematik „Suche nach Inhalten“ haben die Entwickler von Peer-to-Peer-Netzwerken jedoch rasch realisiert, dass sich dieses Minimum an zentralen Strukturen negativ auswirkt. Daher greifen zahlreiche Peer-to-Peer-Netzwerke auf zentralisierte Indexsysteme zurück.
Ontologien werden seit ihrer Einführung auf Informatikteilgebieten wie der „Verarbeitung natürlicher Sprache“, „Wissensmanagement“ oder dem „Semantic Web“ eingesetzt. Sie dienen als Mittel zur Repräsentation eines Wissensbereiches. Eine standardisierte Terminologie macht sie für Automaten les- und verwertbar.[Gómez-Pérez u. a. (2004)]

Ausgangspunkt dieser Arbeit ist die Idee, **Ontologien** als Mittel zur Suchoptimierung in **Peer-to-Peer-Netzwerken** heranzuziehen - ohne die Verwendung von zentralen Strukturen. Bei der Literaturrecherche nach bereits existierenden Lösungsvorschlägen geriet das Paper Sangpachatanaruk und Znati (2004) in den Fokus (siehe Anhang A). Auf zwei Seiten wird dort ein Vorschlag zur Nutzung von Ontologien zur Suchoptimierung beschrieben.
Die Arbeit analysiert den Vorschlag auf Machbarkeit und stellt hierauf basierend einen experimentellen Prototyp vor. Detaillierte Informationen folgen im Kapitel 3 *Vision*.

1.2. Aufbau der Arbeit

Die Arbeit gliedert sich in sieben Kapitel. Nach einer **Einführung** folgt das Kapitel **Grundlagen**, in dem Hintergrundinformationen zu den zwei zentralen Bausteinen der Arbeit, Ontologien und Peer-to-Peer-Netzwerk, präsentiert werden.
Die **Vision** beschreibt die Grundproblematik und weist auf Gründe für die Anforderung nach einer Suchoptimierung in Peer-to-Peer-Netzwerken hin. Daran angeknüpft werden die Ziele

der Arbeit formuliert.
Das Kapitel **Semantic Aware Net** widmet sich dem Ausgangspunkt der Arbeit, dem Paper Sangpachatanaruk und Znati (2004), nachzuschlagen im Anhang A. Der Lösungsvorschlag wird einer eingehenden Analyse unterzogen und die Machbarkeit überprüft.
In den beiden Kapiteln **Prototyp: DHTPlusO** und **Evaluation** wird ein Prototyp als Proof-Of-Concept vorgestellt. Er setzt die Idee um, das Wissen aus Ontologien für eine Verbesserung der Suche zu nutzen.
Den Abschluss bildet das Kapitel **Zusammenfassung**, es listet die Ergebnisse auf und reflektiert sie in den Abschnitten Hürden und Ausblick.

2. Grundlagen

2.1. Ontologien

In der Informationsgesellschaft von heute werden das Erstellen, die Speicherung und der Transfer von Informationen durch stetig sinkende Festplattenpreise, kostengünstige Flatrates und wachsende Transferraten immer attraktiver. Die Flut an Informationen macht aber die Vernetzung und das Auffinden von Informationen immer schwieriger, ein systematisches Durchsuchen stellt eine große Hürde dar.
Ein zweiter Punkt betrifft die Nutzung von Informationen. Menschen erzeugen neues Wissen, d.h. Begriffe und Zusammenhänge eines Wissensbereiches, indem sie auf ihr Grund- und Kontextwissen zugreifen und mit neuen Informationen verknüpfen. Automaten benötigen dafür eine formale Repräsentation dieser Begriffe und Zusammenhänge und damit eine Eindeutigkeit und Maschinenlesbarkeit [Hesse (2002)].
Hier setzt das Konzept der Ontologie an.

2.1.1. Definition

Ontologien helfen, Erkanntes und Erdachtes eindeutig zu definieren und maschinenlesbar zu machen. Sie bieten eine Repräsentation der Konzepte und Zusammenhänge eines Wissensbereiches.
Der wohl bekannteste Definitionsversuch (siehe Gómez-Pérez u. a. (2004), Hesse (2002), Gruninger und Lee (2002)) stammt von T. R. Gruber [Gruber (1993)], wonach Ontologien eine

> „[...] explizite formale Spezifikation einer gemeinsamen Konzeptualisierung“

sind.
Oder, um Ehrig und Studer (2006) zu zitieren: Ontologien sind

> „[...] formale Modelle eines Anwendungsbereiches, [die] die Kommunikation zwischen menschlichen und/oder maschinellen Akteuren unterstützen und das Teilen des Wissens erleichtern.“

Die entscheidenden Stichwörter im Zusammenhang mit Ontologien sind somit „Konzeptualisierung“ und „Spezifikation“.
Konzeptualisierung steht für den Versuch, ein Phänomen der realen Welt auf eine abstrakte Art und Weise darzustellen, wobei nur der für das Problem relevante Kern des Phänomens betrachtet wird. *Spezifizierung* ist Stellvertreter für die Anforderung, dieses „Phänomen der realen Welt“ eindeutig sowie für Maschinen les- und auswertbar zur definieren (nach Staab (2002) und Gómez-Pérez u. a. (2004)). Diese Bedingungen ziehen folgenden Aufbau von Ontologien nach sich.

2.1.2. Struktur und Aufbau

Grundsätzlich lässt sich eine Ontologie nach Sure (2004) in vier Bestandteile unterteilen:

- Vokabular/Lexikon
- Konzepte
- Semantische Relationen
- Regelhafte Zusammenhänge

Lexikon Enthält eine Menge von Worten (lexikalische Einträge, Symbole), wobei die Worte *Konzepte* und *semantische Relationen* instanziieren und für ein *Konzept* mehrere Worte (= Synonyme) existieren können.

Konzepte Stehen für die relevanten Begrifflichkeiten einer Domäne und sind damit als Ergebnis einer Einigung innerhalb einer Gruppe („gemeinsame Konzeptualisierung“) anzusehen. Sure verwendet das Wort „Begriffe“ anstelle von Konzepten.

Semantische Relationen *Konzepte* werden durch *Semantische Relationen* miteinander verbunden. Beispielsweise setzt die *is-a* Relation den Oberbegriff „carnivore“ mit dem Unterbegriff „dog“ in Beziehung. Auch anwendungsspezifische Relationen können weitere Verbindungen definieren.

Regelhafte Zusammenhänge Erfassen zusätzliche Bedeutungsinhalte von *Konzepten* und *semantischen Relationen*. So impliziert die Relation „arbeitetIn“ die Relationen „hatMitarbeiter“. Dieser Zusammenhang wird durch eine Regel maschinenlesbar gemacht:
WENN ANGESTELLTER A1 IN PROJEKT P1 ARBEITET, DANN HAT DAS PROJEKT P1 DEN ANGESTELLTEN A1 ALS MITARBEITER.
Dieser Bestandteil bleibt aber im Rahmen der Arbeit außen vor, lediglich die *Semantischen Relationen* finden Gebrauch.

2.1.3. Kategorisierung

Eine Kategorisierung von Ontologien erfolgt nach Guarino (1998) durch ihren Allgemeinheitsgrad. Es ergeben sich vier Ebenen:

- top-level ontology
- domain-level ontology
- task ontology
- application ontology

Top-level ontology Beinhaltet allgemeine Konzepte wie Raum, Zeit, Materie (z.B. „physikalische Entität“) oder Aktionen.

Domain-level ontology beschreibt das Vokabular eines Problembereichs, indem allgemeine Konzepte der Top-level-Ontologie spezifiziert werden (z.B. „Tiere“). Diese Art von Ontologien kann als domainspezifische Wissensbasis dienen.

Task ontology Charakterisieren das Vokabular für eine generische Aufgabe oder Aktivität (z.B. „fortpflanzen“).

Application ontology Liefert eine Definitionen zur Modellierung des Wissens für eine bestimmte Anwendung.

2.1.4. Fazit

Mit Hilfe von Ontologien lassen sich die Wörter und deren Zusammenhänge innerhalb einer Domäne beschreiben. Eine Kategorisierung und die damit verbundenen Abstraktionsebenen helfen bei der Wiederverwendung. Die Erstellung von Ontologien erfordert jedoch die Erfassung des Wesentlichen einer Domäne, was einen Konsens der Anwender über die Zusammensetzung einer Ontologie erforderlich macht und nicht immer garantiert werden kann. Für eine effektive Nutzung ist die Gewährleistung der Integrität und Konsistenz aber unabdingbar. Zudem muss für einen weitreichenden Einsatz eine Automatisierung der Erzeugung von Ontologien unter Wahrung der Konsistenz möglich sein [Guarino (1998) und Staab (2002)].

2.2. WordNet

WordNet ist eine lexikalische Datenbank der englischen Sprache, entwickelt an der Universität Princeton [Miller (2008)] und frei verfügbar.
Sie gruppiert Nomen, Verben, Adjektive und Adverben in *synsets* - Listen von Synonymen, stellvertretend für ein Konzept. Dabei wird jedes *synset* durch die beinhalteten Wörter spezifiziert. Über die Hälfte der Wörter sind mit einem kurzen Erklärungssatz versehen (Gómez-Pérez u. a., 2004, Abschnitt 2.3.1). Die Wörter in einem *synset* sind im gleichen Kontext austauschbar. Taucht ein Wort in mehreren *synsets* auf, so besitzt es mehrere Bedeutungen, in WordNet *sense* genannt (siehe Abbildung 2.1).

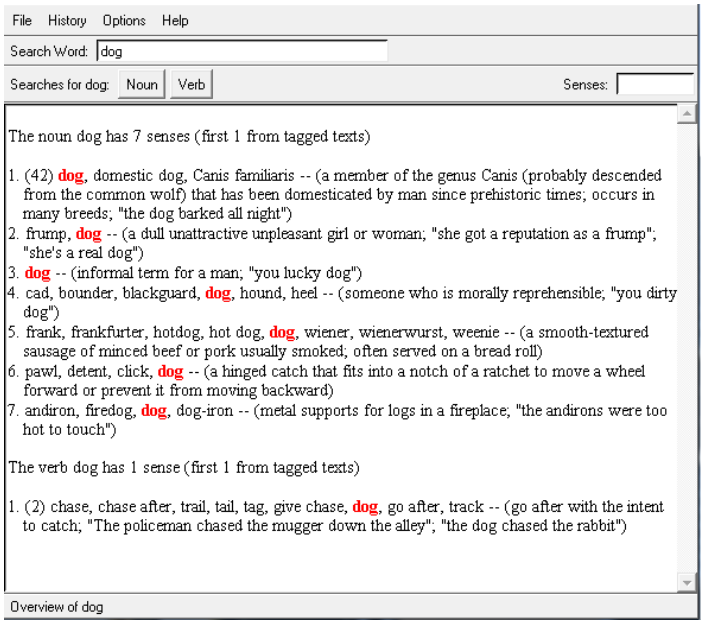

Abbildung 2.1.: WordNet-Suche nach „dog“

Die *synsets* beziehungsweise die Wörter in den *synsets* sind untereinander durch lexikalische und konzeptuelle Relationen verknüpft (siehe Abschnitt 2.2.1).
Durch das Netzwerk von *synsets* und Relationen kann mittels Browser navigiert werden, die Version 3.0 vereint 155.287 Wörter, davon 117.798 Nomen und 117.659 *synsets*.
Hervorzuheben ist, dass WordNet die Bedeutung eines Wortes in den Vordergrund stellt und nicht die Form. Eine Suche nach „dog“ liefert das gleiche Ergebnis wie eine Suche nach „dogs“.

2.2.1. Relationen

Die Arbeit konzentriert sich zunächst auf Substantive als Suchanfrage, sie machen ca. 75% der Einträge aus. Daher folgt ein genauerer Blick auf die Relationen zwischen Substantiven. WordNet unterstützt auf Substantivebene vier relevante Typen semantischer Relationen, die direkt aus der Linguistik übernommen wurden und alle unter konzeptuellen Relationen einzuordnen sind [Gangemi u. a. (2003)]. Zum einen die *Hyperonymie* und die inverse Relation *Hyponymie*, der Oberbegriff bzw. der Unterbegriff eines gegebenen Wortes. So ist das Wort „carnivore“ eine *Hyperonym* von „dog“ und „dog“ ein *Hyponym* von „carnivore“. (siehe Abbildung 2.2)

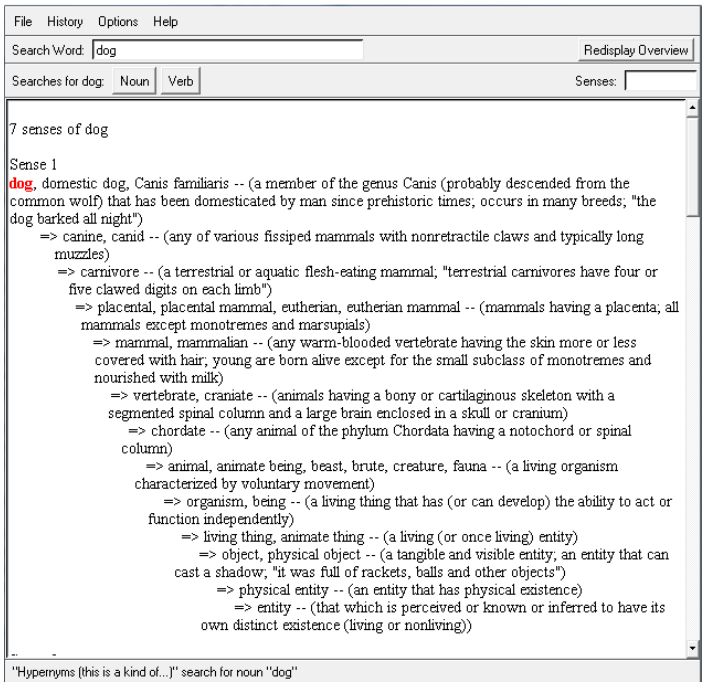

Abbildung 2.2.: Hyperonyme zu „dog“

Das zweite Paar sind die partitive Relation *Meronymie* und deren Umkehrung, die *Holonymie*. Sie spezifizieren eine „Teil-Ganzes-Beziehung“ . „Hand“ ist beispielsweise ein Meronym von „Arm“. Neben diesen Relationen führt WordNet noch die lexikalische Relation *Antonymie* als inverse Relation zur *Synonymie*.

2.2.2. Ontologiesicht auf WordNet

Wurde WordNet zu Anfang lediglich als lexikalische Datenbank verwendet, hat es sich im Laufe der Zeit auch als Ontologie-Quelle etabliert. So geben die unterstützen Relationen wie *Hyperonymie* und *Hyponymie* nicht nur Auskunft über die Sprache, sondern auch über die zugrundeliegende Domäne. Sie korrespondieren direkt mit der subsumption-relation (*is-a*) der Ontologiewelt [Oltramari u. a. (2008)]. In Bezug auf die in Abschnitt 2.1.3 eingeführten Kategorien beinhaltet die WordNet-Datenbank sowohl *top-level*-Ontologien mit Konzepte wie „physical entity“ als auch *domain-level*-Ontologien mit „canis“ oder „dog“ aus der Domain „animal“.
In einer Analyse haben die Autoren in Oltramari u. a. (2008) jedoch festgestellt, dass eine vollständige und korrekte Abbildung der WordNet-Datenbank auf Ontologien nicht ohne eine

Anpassung erfolgen kann. So besteht unter anderem eine Inkonsistenz bei der Verwendung von Konzepte und Instanzen, wenn beispielsweise „kung fu“ als *Hyponym* des Konzeptes „martial art“ geführt wird. Es handelt sich hierbei nicht um eine Relation zwischen Konzepte, sondern zwischen einem Konzept und einer konkreten Instanz.
Eine Bereinigung der WordNet-Datenbank hat sich das *OntoWordNet Projekt* [Gangemi u. a. (2003)] zur Aufgabe gemacht, es möchte automatisiert diverse Inkonsistenzen beheben.

2.2.3. Fazit

WordNet bietet mit seinen *synsets* und semantischen Relationen die Voraussetzung für eine Verwendung als Ontologie-Quelle. Obwohl die Datenbank nicht frei von Inkonsistenzen ist, kann sie dank ihres großen Vorrates an *synsets* zu einem gegebenen Wort zahlreiche verwandte Wörter liefern.

2.3. Peer-to-Peer

Das Peer-to-Peer-Modell stellt einen Gegenentwurf zum Client-Server-Modell dar. Es ist immer häufiger die Netzwerkarchitektur der Wahl, wenn es um das Design von Diensten wie „File-Sharing“, „High-Performance Multimediaanwendung“ oder „Collaborative Work“ geht. Laut *ipoque Internetstudie 2007* [ipoque (2007)] erzeugten alle Peer-to-Peer-Protokolle zusammengenommen ~70% des Internetverkehrs in Deutschland, das HTTP-Protokoll hingegen nur ~10%.
Charakteristisch für ein Peer-to-Peer-System ist die Tatsache, dass jeder Teilnehmer sowohl die Server- als auch Clientrolle annimmt und Teile seiner Ressourcen den anderen Teilnehmern zur Verfügung stellt. Eine klare Unterteilung zwischen Client und Server ist also nicht mehr vorhanden.
Nachfolgend werden das Peer-to-Peer-Modell und die Eigenschaften sowie Typen von Peer-to-Peer-Systemen vorgestellt - mit einer genaueren Betrachtung des *DHT*-basierten Overlay-Netzwerk *Chord* [PDOS und MIT (2008)].

2.3.1. Peer-to-Peer-Modell

Das hinter Peer-to-Peer-Systemen stehende architektonische Modell wird in Dustdar u. a. (2003) erläutert. Demnach existieren vier Abstraktionsschichten:

- Netzwerkebene
- Datenzugriffsebene

- Dienstebene
- Benutzerebene

Netzwerkebene Stellt anwendungsunabhängig elementare Routing-Dienste zur Verfügung.

Datenzugriffsebene Bietet anwendungsabhängig Dienste wie die Suche und das Ändern von Ressourcen.

Dienstebene Erweitert die Funktionalität der *Datenzugriffsebene* um Dienste wie beispielsweise Filesharing.

Benutzerebene Auf dieser Ebene sammeln sich Dienste zur Benutzer-Interaktion und zum Collaborative Work.

Wichtig ist nach Dustdar u. a. (2003) der Fakt, dass in allen vier Ebenen das *Peer-to-Peer-Paradigma* (siehe Abschnitt 2.3.2 *Eigenschaften*) mehr oder minder stark Einzug halten kann und die Kombination der Ebenen ein Architekturexemplar und damit ein Peer-to-Peer-System ausmacht.

2.3.2. Eigenschaften

Das *Peer-to-Peer-Paradigma* umfasst nach Mahlmann und Schindelhauer (2007) ein Netzwerk, das

> „ein *Overlay-Netzwerk* des Internets ist, in dem Rechner ebenbürtig ohne zentrale Koordination kommunizieren und gemeinsam eine Anwendung zur Verfügung stellen."

Damit besitzt eine Peer-to-Peer-System folgende Haupteigenschaften, an denen es sich messen lassen muss:

- Dezentrales Modell
- Skalierbarkeit
- Fehlertoleranz
- Selbstorganisation

Diese Eigenschaften stehen für ein vollkommen dezentralisiertes Modell ohne globale Kontrollmechanismen.

2.3.3. Typen

Eine Umsetzung dieser Eigenschaften aus Abschnitt 2.3.2 und die Konstruktion von Peer-to-Peer-Systemen wurden in der Vergangenheit mehr oder weniger streng gehandhabt. Die existierenden Systeme teilen sich in zwei Typen, die im nächsten Abschnitt beschrieben werden.

2.3.3.1. Unstrukturierte Peer-to-Peer-Systeme

Die auch als „Erste Generation“ bezeichneten unstrukturierten Peer-to-Peer-Systeme besitzen noch Client-Server oder auch hybride Netzwerkstrukturen und können nur mit Teilen der in Abschnitt 2.3.2 eingeführten Eigenschaften aufwarten.
Das Erste als ein solches bezeichnete Peer-to-Peer-Netzwerk **Napster** verwendet beispielsweise einen Server, der einen Index über die im Netzwerk verfügbaren Daten vorhält. Damit ist die Eigenschaft *Dezentralen Modell* nicht erfüllt.
Das sehr erfolgreiche **Gnutella** besitzt keine solchen Client-Server-Strukturen mehr, ihm mangelt es aber an der geforderten *Skalierbarkeit*. Die Suche geschieht durch sogenanntes *Flooding* und ist nicht deterministisch. Eine wachsende Teilnehmerzahl und der damit verbundene Anstieg des Nachrichtenaufkommens gehen zulasten der Bandbreite. In der Bachelorarbeit Pötter (2006) wird das Gnutella-Protokoll einer detaillierten Sicherheits- und Performance-Analyse unterzogen.

Als Fazit kann festgehalten werden, dass sich die „Erste Generation“ von ihrer Struktur her noch nicht vollständig von dem Client-Server-Modell loslösen kann. Bei der Organisation und Suche muss auf Hilfskonstrukte wie Flooding, Random Walk und Heuristiken zurückgegriffen werden.

2.3.3.2. Struktierte Peer-to-Peer-Systeme

Die im oberen Abschnitt aufgeführten Nachteile wie zentrale Kontrollmechanismen oder eine fehlende Skalierbarkeit werden durch den zweiten Typus, die strukturierten oder echten *Peer-to-Peer-Systeme*, behoben. Dieser Typ von Peer-to-Peer-Systemen ist auf der in Abschnitt 2.3.1 beschriebenen *Datenzugriffsebene* angesiedelt. Bei unstrukturierten Peer-to-Peer-Systemen wie Napster gehört nur die Suche zu dieser Ebene, die Dateiübertragung aber zur Dienstebene (Dustdar u. a., 2003, Abschnitt 7.3).

Ein Instrument, mit welchem die in Abschnitt 2.3.2 erläuterten Eigenschaften erfolgreich umgesetzt werden, sind verteilte Hash-Tabellen (engl. „Distributed Hash Tables“, kurz DHT). Zahlreiche Peer-to-Peer-Systeme wie CAN [Ratnasamy u. a. (2001)], Chord [Stoica u. a. (2001)] oder Pastry [Rowstron und Druschel (2001)] realisieren ein Overlay-Netzwerk auf Basis von DHT. Alle Implementationen haben eine kompakte Schnittstelle (u.a. put(key, data) und get(k)) und eine globale Hash-Funktion zur Abbildung von Peers und Daten auf einen Schlüsselraum gemein. Sie unterscheiden sich aber stark in puncto Routing oder der Organisation des Schlüsselraumes.
Einer näheren Betrachtung soll das Overlay-Netzwerk *Chord* unterzogen werden.

2.3.4. Chord

Das von Stoica u. a. (2001) vorgestellte strukturierte Peer-to-Peer-Netzwerk *Chord* basiert auf verteilten Hash-Tabellen und ist „... effizient, skalierbar und auf die Suche spezialisiert“ [Schindelhauer (2008)].
Da *Chord* die Basis für den Prototyp in Kapitel 5 sein wird, sollen die Eigenschaften und die Suche des Overlay-Netzwerkes an dieser Stelle hervorgehoben werden.

2.3.4.1. Eigenschaften

Die Netzwerktopologie von *Chord* folgt einer Ringstruktur. Daten und Peers werden via globaler Hash-Funktion mit einem m-bit Identifier - *key* genannt - assoziert und in den selben Schlüsselraum gehasht. Der Identifier befindet sich in einem Intervall zwischen 0 und $2^{(m-1)}$.
Zur Organisation der Ringstruktur besitzt jeder Peer im Netzwerk Informationen über seinen Vorgänger und Nachfolger. Hinzu kommt eine sogenannte Finger-Tabelle, die maximal *m* Einträge auf Folgeknoten besitzt und nach der Vorschrift $n + 2^{i-1}$ mit 1 <= i <= m konstruiert wird.
Die Abbildung 2.3 (Steinmetz und Wehrle, 2005, Abschnitt 8.1.1, S.96) illustriert ein *Chord*-Netzwerk und eine exemplarische Finger-Tabelle. Die gestrichelten Linien machen deutlich, welcher Peer für ein Datum zuständig ist, die schwarzen Linien zeigen die „Finger“ von Peer N8.

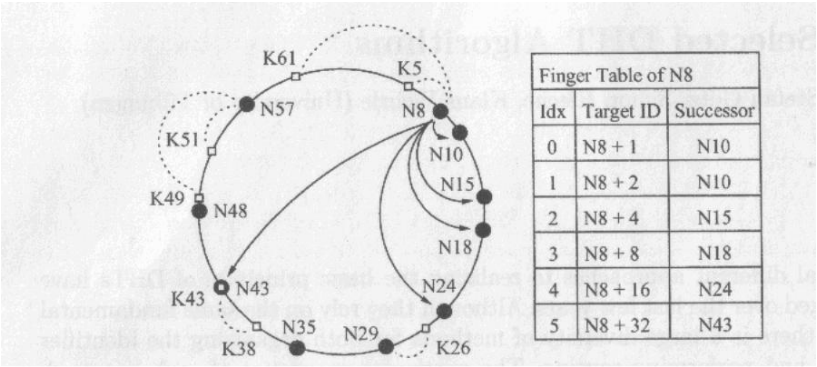

Abbildung 2.3.: Chord-Netzwerk mit 10 Peers und 7 Ressourcen

Damit ist für einen gegebenen *key* in O(log n) der zuständige Peer ermittelbar und eine Lokalisierung von Daten mit einer Komplexität vergleichbar der Binären Suche möglich.
Desweiteren existieren im *Chord*-Protokoll für die elementaren Operationen Einfügen, Entfernen und Suchen die Funktionen put()/join(), remove() und get().

2.3.4.2. Suche in Chord

Die Suche in *Chord* ist *key*-basiert. Der *key* ist das Ergebnis der Anwendung der globalen Hash-Funktion auf den zu suchenden Ressourcennamen bzw. der IP-Adresse des Peers. In Bezug auf *Chord* hat sich anstelle von *key* der Begriff *id* durchgesetzt. Wie in Abschnitt 2.3.4.1 *Eigenschaften* angedeutet, ist *Chord* auf die Suche ausgelegt und bietet durch Finger-Tabelle eine Suche mit O(log n) Sprüngen.
Folgendes Listing 2.1 illustriert den Pseudo-Code Algorithmus für die Suche in *Chord* (Schindelhauer, 2008, S. 88)):

```
1   //finde ausgehend von Peer p den für id verantwortlichen Peer
2   p.suche(id){
3       if id in (h(p),p.Nachfolger]
4           return p.Nachfolger
5       else
6           //wähle größten Finger-Zeiger von p, der noch vor id liegt
7         for i <- 1 to m
8               if p.finger[i] in [h(p),id)
9                       p' <- p.Finger[i]
10        return p'.suche(id)
11  }
```

Listing 2.1: Pseudo-Code Algorithmus der Suche in *Chord*

Die Suche ist auf der Stelle erfolgreich abgeschlossen, wenn p oder p's direkte Nachbarn für die *id* zuständig sind. In allen anderen Fällen wird mit dem größten Zeiger gestartet und evaluiert, ob er die *id* übersprungen hat.
Diese Vorgehensweise ist möglich, weil Ressourcen und Peers in den gleichen Schlüsselraum gehasht werden. Wurde die *id* nicht übersprungen, wird der Algorithmus rekursiv auf diesen Finger aufgerufen. Dies geschieht, bis der für das Datum zuständige Peer ermittelt ist.
Es bleibt zu konstatieren, dass der Abstand zum Ziel höchstens 2^m beträgt und das mit jedem Sprung die Entfernung zum Ziel halbiert wird, was die Suchkomplexität von O(log n) erklärt.
Abbildung 2.4 gibt einen schematischen Überblick zu den einzelnen Abschnitten der Suche.

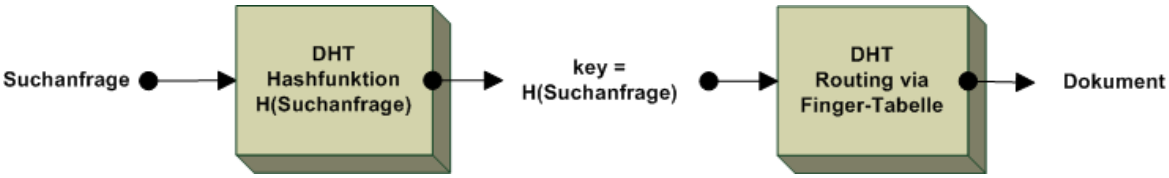

Abbildung 2.4.: Chord-Suche

2.3.5. Fazit

Peer-to-Peer-Systeme bieten interessante Eigenschaften wie Skalierbarkeit, Selbstorganisation oder dezentrale Strukturen, was sie gegenüber dem klassischen Client-Server-Modell abhebt.
Den hohen Anteil von Peer-to-Peer-Protokollen am Verkehrsvolumen haben Provider in jüngster Zeit sogar durch eine Drosselung der Bandbreite für Peer-to-Peer-Protokolle wie BitTorrent zu mindern versucht.
Im Fall von *Chord* wurde eine effiziente Suche in einem Overlay-Netzwerk vorgestellt, die aber noch nichts über die semantische Qualität der in „Rekordzeit" lokalisierten Ressource aussagt.

2.4. Suchoptimierung - Verwandte Arbeiten

Seit Einführung der strukturierten Peer-to-Peer-Systeme basierend auf verteilten Hasht-Tabellen wurden zahlreiche Vorschläge zur Suchoptimierung präsentiert.

Einige Optimierungen setzen dabei auch Ontologien ein. Dieser Abschnitt stellt eine Auswahl von Lösungsansätzen vor. Betrachtet werden die zugrundeliegende Idee, die Behebung der spezifischen Problematik und eventuelle Hürden.
Zunächst wird jedoch versucht, die zu lösenden Problematiken bei der Suche in *DHT*-basierten Peer-to-Peer-Netzwerken in Kategorien zu unterteilen.

2.4.1. Problemkategorien

Bei der Auseinandersetzung mit der Suchoptimierung in Peer-to-Peer-Netzwerken wurden die Autoren mit immer wiederkehrenden Problemen konfrontiert. Darauf basierend lassen sich die Gründe für eine erfolglose Suche in Problemkategorien unterteilen, die im Folgenden kurz beschrieben werden sollen.

2.4.1.1. Perfect-Matching

Perfect-Matching nach Mahlmann und Schindelhauer (2007) und Zeinalipour-Yazti u. a. (2004, S. 5, 3. Abschnitt) beschreibt den Umstand, dass in einem DHT-basierten Overlay-Netzwerk mit globaler Hash-Funktion nur solche Ressourcen gefunden werden können, dessen Objektname exakt mit dem der im Netzwerk gespeicherten Ressource übereinstimmt. Der Ressourcenname ist Eingabeparameter der globalen Hash-Funktion und bei kleinsten Abweichungen in der Schreibweise (beispielsweise durch orthografische Fehler) ist der erzeugte *key* ein anderer. In der *Chord* Implementierung des später noch näher beschriebenen *Overlay Weaver*-Framework (siehe Abschnitt 5.3.1) erzeugt die Hash-Funktion folgende völlig unterschiedliche *keys*.

Ressourcenname	**key**
dog	e49512524f47b4138d850c9d9d85972927281da0
dogg	9c6c927190d86d7e828b6587b574ab7f88a05dd2

2.4.1.2. Multiple-Keys

Die Suche in *DHT*-basierten Peer-to-Peer-Systemen funktioniert klassisch nach dem *Single-Key* Prinzip (Sangpachatanaruk und Znati, 2004, Kapitel 1, 2.Absatz). Für die Suche wird ein einzelner *key* als Parameter für die *get()*-Funktion herangezogen, gewonnen durch die Anwendung der globalen Hash-Funktion auf den Ressourcennamen.
Um die Suche zu verbessern, wurde die *Multiple-Keys*-Methode (Sangpachatanaruk und

Znati, 2004, Kapitel 1, 2.Absatz) entwickelt. Danach werden zur Platzierung und Lokalisierung von Ressourcen im Overlay-Netzwerk mehrere *keys* herangezogen, was eine Anpassung dieser Verfahren nach sich zieht.

2.4.1.3. Ähnlichkeitssuche

Die *Ähnlichkeitssuche* nach Schindelhauer (2008) ist die Reaktion auf die Anforderung nach einer Suche von Ressourcen, die semantisch zur Gesuchten passen. So sollten über die *Multiple-Keys* Methode hinaus Ressourcen lokalisiert werden, die thematisch zur Suchanfrage passen, ohne jedoch mit der Suchanfrage exakt übereinzustimmen zu müssen.

2.4.1.4. Datenzusammenhang

Schindelhauer (2008) weist in Verbindung mit der Verwendung einer globalen Hash-Funktion auf die Zerstörung des Datenzusammenhanges hin. Demnach verlieren Peers mit verschiedenen Ressourcen zu einem Themengebiet bei der Platzierung im Netzwerk die Zuständigkeit (siehe Abschnitt 2.4.1.1 *Perfect-Matching*).
Ein Peer im Netzwerk ist bedingt durch die Anwendung der Hash-Funktion auf die Ressourcen-Bezeichner für thematisch unabhängige Ressourcen zuständig. Es ist daher kaum möglich, nach der Lokalisierung einer Ressource bei einem Peer dort auch Ähnliche zu finden.

2.4.2. pSearch

Idee Einer der frühesten Ansätze zur Optimierung der Suche in strukturierten, DHT-basierten Peer-to-Peer-Netzwerken stellt *pSeach* aus dem Jahre 2003 dar, vorgestellt in dem Paper Fox u. a. (1991). Zentrale Idee ist die Repräsentation von Ressourcen und Suchanfragen als Attribut-Vektor. Die Ähnlichkeit der Ressourcen ergibt sich aus dem Kosinus zweier Vektoren. Der Vektor dient zudem als *key* für die Platzierung im Netzwerk. So werden semantisch verwandte Ressourcen „nahe“ zueinander gespeichert.

Problemkategorie Mit dem Attribut-Vektor besitzt *pSearch Multiple-Keys*. Das Speichern verwandter Ressourcen in der Nachbarschaft ermöglicht eine einfache *Ähnlichkeitssuche* und einen Erhalt des *Datenzusammenhanges*.

Hürden Es wird nicht beschrieben, wie eine konsistente Verwendung von Attributen gewährleistet wird. Denn für die Nutzung des *Semantic Vector* ist es essenziell, dass die Peers global gültige Attribute für ihre Ressourcen verwenden.
Das *Perfect-Matching* Problem wird nicht gelöst, nur auf die Attribute im Vektor verlagert.

2.4.3. Semantic Small World

Idee *Semantic Small World*, kurz *SSW* wurde 2004 von Li et. al. in dem Paper Li u. a. (2004) eingeführt. Peers, die „semantisch ähnlichen Daten“ lokal speichern, werden in Bereiche unterteilt und organisiert. Die semantische Ähnlichkeit bezieht sich wie bei *pSearch* auf einen Vektor aus Attributen (= Metadaten) zu jeder Ressource, dem *Semantic Vector*. Die Ähnlichkeit wird durch die euklidische Distanz zwischen zwei Vektoren repräsentiert. *SSW* leistet als Verbesserung gegenüber pSearch außerdem eine Reduzierung der durch die Größe des Attributvektors resultierenden Mehrdimensionalität des Schlüsselraumes. So soll der Verwaltungsoverhead verringert und die Performance bei der Platzierung und Lokalisierung von Daten verbessert werden.

Problemkategorie Der Einsatz des *Semantic Vector* entspricht der Verwendung von *Multiple-Keys*, das Clustering ermöglicht eine *Ähnlichkeitssuche* und das Beibehalten des *Datenzusammenhanges*.

Hürden Das *Perfect-Matching* Problem wird nicht behoben, durch die Verwendung mehrerer Attribute als *key* aber verringert. Auch ist wie bei *pSearch* nicht geklärt, wie global gültige Attribute sichergestellt werden.

2.4.4. Taxonomy-Based Routing

Idee Der Vorschlag *Towards Taxonomy-Based Routing in P2P Networks* aus Löser u. a. (2004) nutzt Taxonomien, um Ressourcen zu klassifizieren. Bei einer Suchanfrage werden nur solche Peers kontaktiert, die verwandte Ressourcen vorhalten. Dabei kommen sogenannte *Super-Peers* zum Einsatz. Sie halten die global gültigen Taxonomien vor und dienen zur strukturierten Platzierung und Lokalisierung von Ressourcen.

Problemkategorie Durch die Verwendung von Taxonomien wird eine *Ähnlichkeitssuche* erreicht und der *Datenzusammenhang* bewahrt. *Perfect-Matching* kann durch globale „Keyword Kataloge“ sichergestellt werden.

Hürden Die *Super-Peers* stellen einen *Single-Point-of-Failure* dar, ein gezielter Angriff würde die Funktionsfähigkeit des gesamten Netzwerkes stören.

2.4.5. Ontology-Based P2P Infrastructure

Idee Teil des Papers Schlosser u. a. (2002) ist die Einführung eines Peer-to-Peer-Netzwerkes basierend auf einem Hypercube. Es verwendet globale Ontologien, um die Netzwerktopologie in Bereiche mit verwandten Konzepte zu unterteilen. Tritt ein neuer Peer dem Netzwerk bei, so ordnet er seinen Ressourcen Konzepte aus „global verfügbaren “ (Schlosser u. a., 2002, Abschnitt 4.3) Ontologien zu. Die Struktur des Hypercube's ermöglicht eine effiziente Navigation im Netzwerk und damit eine günstige Platzierung neuer Peers in dem semantisch verwandten Bereich.

Problemkategorie Die „Bereiche mit verwandten Konzepten“ sind für eine *Ähnlichkeitssuche* geeignet und schaffen eine Basis für den *Datenzusammenhang*.

Hürden Kritikpunkt sind die „global verfügbaren“ Ontologien. Deren Erreichbarkeit sollte unter allen Umständen sichergestellt sein, sei es lokal als Replik bei jedem Peer oder zentral in einem Repository. Müssen sie von einem Server bezogen werden, droht ein *Single-Point-of-Failure*.

2.4.6. Fazit

Die vier vorgestellten Ansätze zur Suchoptimierung nutzen alle Ontologien. Die Bewertung hat gezeigt, dass kein Vorschlag ohne eine Verletzung wichtiger Peer-to-Peer-Eigenschaften (siehe Abschnitt 2.3.2) auskommt. Auch können nicht alle Probleme der in Abschnitt 2.4.1 eingeführten Kategorien behoben werden.

3. Vision

Overlay-Netzwerke wie CAN, Chord, Tapestry und Pastry nutzen verwaltungsarme und fehlertolerante verteilte Hash-Tabellen (*DHT*) als Datenstruktur. Die Suche in diesen strukturierten Peer-to-Peer-Netzwerken besticht durch ihre Einfachheit. Eine globale Hash-Funktion wird auf den zu suchenden Ressourcennamen angewendet und es steht der für die Suche erforderliche *key* meist in Form einer n-bit ID zur Verfügung (siehe Abschnitt 2.3 *Peer-to-Peer*). Im Anschluss sorgt der Routing-Algorithmus des Overlay-Netzwerkes für die Navigation zu dem für den *key* zuständigen Peer, im Falle von dem auf die Suche spezialisierten *Chord* sogar in O(log n) [Mahlmann und Schindelhauer (2007)].
Diese Einfachheit zieht aber gewichtige Nachteile mit sich. Die globale Hash-Funktion bewirkt, dass kleinste Fehler oder Abweichungen im Ressourcennamen es unmöglich machen, die gewünschte Ressource zu lokalisieren. Zudem ist eine *Ähnlichkeitssuche* aussichtslos, wieder bedingt durch die Hash-Funktion, die den *Datenzusammenhang* (siehe Abschnitt 2.4.1) zerstört.
Es kann also nicht sichergestellt werden, dass alle für die Suchanfrage relevanten Ressourcen auch gefunden werden. Es fließen solche nicht mit in das Ergebnis ein, die vom exakten Objektnamen der Suchanfrage abweichen [Tang u. a. (2003)].
Man stelle sich ein Peer-to-Peer-Netzwerk vor, in dem Tierfreunde sich ihre Dokumente untereinander zugänglich machen. Peer A bietet beispielsweise zahlreiche Artikel über sein Lieblingstier, den *dog*, an. Peer B möchte genau zu diesem Themengebiet Informationen sammeln. Dementsprechend wird er mit dem Suchwort *dog* eine Anfrage an das Netzwerk stellen. Die einzigen Dokumente, die er auf seine Anfrage erhalten wird, sind jedoch diejenigen, die den Dokumentennamen *dog* vorweisen. Alle anderen Dokumente, auch wenn sie zum Themengebiet *dog* zählen, wie beispielsweise *collie* oder *carnivore*, bleiben Peer B verwehrt.

Dieser Problematik haben sich Chatree Sangpachatanaruk und Taieb Znati angenommen und in dem Paper Sangpachatanaruk und Znati (2004) einen Lösungsansatz vorgestellt.
Die Autoren führen das *Semantic Aware Net* (*SA Net*) ein, dessen Basis ein strukturiertes Peer-to-Peer-System wie beispielsweise *Chord* [Stoica u. a. (2001)] ist, erweitert um den Suchalgorithmus *Semantic Driven Hashing (SDH)* und das Ontologie Ontology Semantic Net *(OSN)*, dessen Topologie den Konzepte und Relationen von Ontologien nachempfunden ist.

Zur Steigerung der Qualität und Anzahl der Suchergebnisse soll nicht nur der Ressourcenname allein zur Suche im Netzwerk herangezogen werden, sondern Ontologien. Die Ontologien, also die Begriffswelt zur gesuchten Ressource, sollen durch das Konsultieren des *OSN* gefunden und so die Suchanfrage um weitere Wörter ergänzt werden - mit dem Ziel der *Ähnlichkeitssuche*. Das Verwenden von globalen OSN_ids, die zu jedem Konzept gehören und als Eingabe in die *DHT* Hash-Funktion dienen, soll die *Perfect-Matching* Problematik 2.4.1 angehen.

3.1. Ziele

Ziel der Arbeit ist es, die Anforderungen des oben genannten Lösungsansatzes zu formulieren und deren Machbarkeit zu evaluieren. Kann er eine Suchoptimierung leisten und Problematiken wie *Ähnlichkeitssuche*, *Perfect-Matching* und *Single-Key*/*Multiple-Keys* (siehe Abschnitt 2.4.1 *Suchoptimierung - Verwandte Arbeiten*) bewältigen. Ohne dabei die Effizienz der Suche in *DHT*-basierten Overlay- Netzwerken durch zu viel Overhead zu beeinträchtigen und ohne das *Peer-to-Peer-Paradigma* zu verletzen (siehe Abschnitt 2.3.2).
Die auf zwei Seiten formulierten Vorschläge und Konzepte bilden die Basis. Da die Autoren aber nur ein grobes Konzept liefern und sich auf einen Überblick zur Funktionalität des *SA Net* beschränken, fließen eigene Ideen und Anforderung zur Optimierung der Suche ein.
Es soll ein Prototyp entwickelt werden, der den Anforderungen des Lösungsvorschlages in Sangpachatanaruk und Znati (2004) genügt und den Ansatz realisiert, Ontologien zur Verbesserung der Suche heranzuziehen.

4. Semantic Aware Net

Das *Semantic Aware Net*, kurz *SA Net*, soll als „structured peer-to-peer overlay architecture“ [Sangpachatanaruk und Znati (2004)] eine semantisch angereicherte Suche garantieren.
In diesem Kapitel werden zunächst die Anforderungen an das System aufgestellt, gefolgt von einer Beschreibung der Komponenten der vorgeschlagenen Architektur. Zum Abschluss wird in einem Analyseteil die Umsetzbarkeit evaluiert.

4.1. Anforderungen

Die Ermittlung der Anforderungen an das System *SA Net* dient der Bewertung des Lösungsvorschlages und dessen Anspruch. Mittel ist der klassische SE Ansatz, nach dem durch Interviews mit dem Anwender und der Analyse des Anwendungsbereiches Anforderungen dokumentiert werden. Es ist jedoch darauf hinzuweisen, dass im Rahmen dieser Arbeit die Rolle des Anwenders als auch die des Entwicklers beim Verfasser liegt.
Als wichtigste Quelle zur Ermittlung von Anforderungen an das System (siehe Kapitel *Vision* 3) dient dabei das Paper Sangpachatanaruk und Znati (2004). Daneben werden die Problemkategorien und Eigenschaften von Peer-to-Peer-Netzwerken aus den Abschnitten 2.4.1 und 2.3.2 herangezogen.

4.1.1. Funktionale Anforderungen

Die Dokumentation der funktionalen Anforderungen soll den Leistungsumfang des Systems beschreiben. Was ist erforderlich, um die in Kapitel 3 beschrieben Eigenschaften zu ermöglichen und eine semantisch angereicherte Suche in einem *DHT*-basierten Peer-to-Peer-Netzwerk sicherzustellen. Weg vom Ressourcennamen als Suchanfrage hin zur konzeptbasierten Suche.

4.1.1.1. Einsatz von Ontologien

Die in Kapitel 3 *Vision* und den folgenden Abschnitten beschriebene semantische Anreicherung der Suchanfrage soll durch den Gebrauch von Ontologien geschehen. Es wird somit nach einer Möglichkeit gesucht, für die *Ähnlichkeitssuche* das Wissen aus Ontologien zu nutzen und so die Suche zu optimieren. Dazu gehören die Konzepte und Relationen, die eine Ontologie bilden (siehe Abschnitt 2.1.2 *Ontologien*).
Als Ontologien sollen die in WordNet verwendeten lexikalischen Ontologien mit ihren Konzepte und Relationen zum Einsatz kommen. Sie bieten zum einen die Möglichkeit, auf lexikalischer Ebene Konzepte zu finden, die mit dem Suchwort verwandt sind - was eine gewisse Domain-Unabhänigkeit bedeutet. Auf der anderen Seite wird mit der Nutzung von WordNet dem Vorschlag aus Sangpachatanaruk und Znati (2004) Rechnung getragen.
Als Konzepte sollen zunächst nur Nomen genutzt werden, da sie 75% der Wortarten in der WordNet Datenbank ausmachen. Für sie existieren die meisten Relationstypen, nämlich die in Abschnitt 2.2 *WordNet* beschriebenen *Hyperonymie* und *Hyponymie* (Oberbegriff/Unterbegriff) sowie *Holonymie* und *Meronymie* (Teil-Ganzes Beziehung).

4.1.1.2. Non-Perfect-Matching

Mit *Non-Perfect-Matching* ist die Möglichkeit verbunden, bei der Suche

- **ähnliche Schreibweisen** heranzuziehen und
- **orthografische Fehler** zu korrigieren

In Abschnitt 2.4.1.1 wird die *Perfect-Matching* Problematik detailliert beschrieben. Das SA Net soll *Non-Perfect-Matching* leisten und damit einen Schritt zur Suchoptimierung beitragen.

4.1.1.3. Multiple-Keys

Zur Suchoptimierung ist es unerlässlich, die *SA Net*-Suchfunktion um die Möglichkeit zu ergänzen, mehrere Suchwörter verwenden zu können. Es steht die Umsetzung der Kategorie *Multiple-Keys* aus Abschnitt 2.4.1.2 im Vordergrund.
Wie an dieser Stelle und in Abschnitt 2.3.4.2 *Suche in Chord* aufgeführt bieten strukturierte Peer-to-Peer-Netzwerke auf Basis von verteilten Hash-Tabellen lediglich die Suche mit einem einzigen *key*.
Diese Anforderung bezieht sich auf die in Abschnitt 2.3.1 *Peer-to-Peer-Modell* dargestellte *Dienstebene*.

4.1.1.4. Ähnlichkeitssuche

Elementare Anforderung ist die Bereitstellung einer *Ähnlichkeitssuche*. Es muss den Peers möglich seinen, Ressourcen zu finden, die mit dem der Suchanfrage semantisch in Verbindung stehen, ohne dem Suchwort direkt zu entsprechen. Sucht ein Peer nach „dog“, soll er als Antwort nicht nur „dog.doc“ erhalten, sondern auch „puppy.doc“ oder „carnivore.doc“. Abschnitt 2.4.1.3 *Ähnlichkeitssuche* nennt in diesem Zusammenhang die zentralen Ansatzpunkte.

4.1.2. Nicht-funktionale Anforderungen

Nach den funktionalen Anforderungen gilt es im nächsten Schritt die nicht-funktionalen Anforderungen aufzustellen. Im Vordergrund stehen dabei die Zuverlässigkeit des Systems.

4.1.2.1. Dezentrales Modell

In Verbindung mit Abschnitt 2.3.2 *Peer-to-Peer-Eigenschaften* steht die Anforderung, im *SA Net* einen *Single-Point-of-Failure* zu vermeiden. Es sollen Angriffspunkte vermieden werden, bei dem das Ausschalten einer Komponente die Funktionsfähigkeit des kompletten Netzwerkes stört.
Eine einzige Instanz darf keine Informationen für andere Peers vorhält, wie beispielsweise einen Vorrat an Ontologien.

4.1.2.2. Peer-to-Peer-Netzwerktyp

Um Anforderung 4.1.2.1 ermöglichen zu können, muss die Suche in einem strukturierten Peer-to-Peer-Netzwerk erweitert werden. Diese wesentliche Anforderung an das *SA Net* macht ein Peer-to-Peer-Netzwerk vom Typus nötig, der in Abschnitt 2.3.3.2 *Strukturierte Peer-to-Peer-Systeme* vorgestellt wurde. Die Suchfunktionalität des *SA Net* baut auf dessen Suchfunktion auf.
Das in Abschnitt 2.3 eingeführte *Chord* [Stoica u. a. (2001)] ist hierfür prädestiniert. Pluspunkt ist die Spezialisierung auf eine effiziente Suche [Schindelhauer (2008)], außerdem existieren zahlreiche Implementationen.

4.1.2.3. Ontologiebasierte Topologie

Die semantische Anreicherung der Suchanfrage soll durch Konsultierung eines Overlay-Netzwerkes geschehen. Das Netzwerk ist derart aufgebaut, dass ein Peer für ein Konzept und eine Verbindung zwischen zwei Peers für eine Relation zwischen den beiden Konzepten steht. Die Topologie dieses Netzwerkes soll Ontologien nachempfunden sein.
Die funktionale Anforderung in Abschnitt 4.1.1.1 legt als Typ die WordNet-Ontologien fest.
Ist der für eine Suchanfrage zuständige Peer lokalisiert, so können durch die Ermittlung der Nachbarn verwandte Konzepte gefunden werden.

4.1.2.4. Fehlertoleranz

Es soll verhindert werden, dass bei Ausfall eines Peers die Funktionalität des Netzwerkes beeinträchtigt wird. Diese Anforderung gilt für Komponente *OSN* und betrifft die Speicherung von gleichen Konzepten bei unterschiedlichen Peers. Verlässt ein Peer das Netzwerk, so muss ein anderer Peer die Zuständigkeit für dessen verwaltete Konzepte erhalten.

4.1.3. Vorbedingungen

Die Sicherheit des Peer-to-Peer-Netzwerk, in Forschung (Mahlmann und Schindelhauer, 2007, Kapitel 10) und Wirtschaft von stetig wachsender Relevanz, wird zunächst vernachlässigt. Es soll die Annahme gelten, dass alle Peers als vertrauenswürdig eingestuft sind. Punkte wie Vertraulichkeit und Integrität in Peer-to-Peer-Systemen sind nicht Gegenstand dieser Arbeit.

4.2. Komponenten

Nachfolgend werden die *SA Net*-Komponenten auf Basis von Sangpachatanaruk und Znati (2004) näher beschrieben.

Das *SA Net* besteht auf **Datenzugriffsebene** (siehe Abschnitt 2.3.1 *Peer-to-Peer-Modell*) aus zwei Overlay-Netzwerken:

- Ontology Semantic Net
- Peer-to-Peer-Netzwerk

Auf **Dienstebene** organisiert der *Semantic Driven Hashing* (*SDH*) Algorithmus das Abrufen von *Keys* aus dem *OSN*. Das *OSN* ist wie in Abschnitt 4.1.2.3 beschrieben direkt dem Aufbau von Ontologien nachempfunden.

4.2.1. Ontology Semantic Net

Das Ontology Semantic Net ist zentrale Komponente der vorgeschlagenen Architektur. Nachfolgend wird es einer genaueren Betrachtung unterzogen.
Als begleitender Anwendungsfall soll ein *SA Net* dienen, mit dessen Hilfe Hundefreunde Dokumente anderen Teilnehmern zu Verfügung stellen können.

4.2.1.1. Topologie

Das OSN soll eine Organisation der Knoten nach dem Vorbild von Ontologien leisten. Das Peer-to-Peer-Netzwerk stellt eine Abbildung der WordNet-Ontologien mit Peers als Stellvertreter für Konzepte und den Verbindungen zwischen diesen Peers als semantische Relationen dar. Ein Knoten im OSN repräsentiert ein Konzept, eine Kante eine Relation zwischen zwei *Konzepten*.
Abbildung 4.1 illustriert den logischen Aufbau des *OSN* am Beispiel des „Tierfreunde"-Netzwerkes und der *Hyperonymie*/*Hyponymie* Relation (Ober-/Unterbegriff).

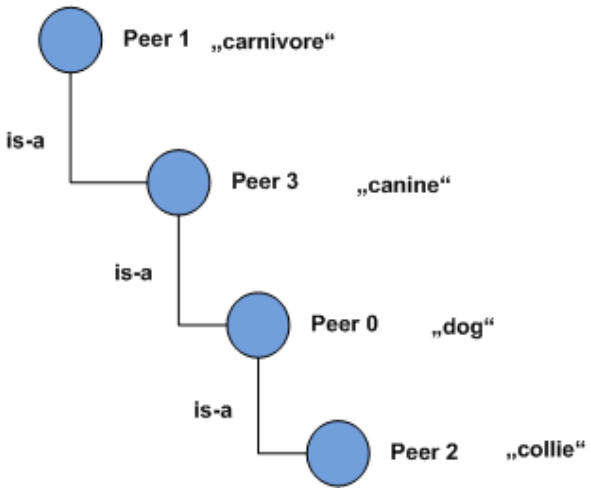

Abbildung 4.1.: OSN-Topologie

Abbildung 4.2 zeigt eine stilisierte WordNet-Ontologie (siehe auch Abschnitt *WordNet* 2.2). Es werden zu dem Wort *dog* in Beziehung stehende Begriffe dargestellt, genauer das *Hyperonym carnivore* und das *Hyponym collie*. Zu beachten ist, dass je nach Blickwinkel der Relationstyp wechselt (siehe Abschnitt 2.2.2). So ist *dog Hyponym* von *canine* und *Hyperonym* von *collie*.

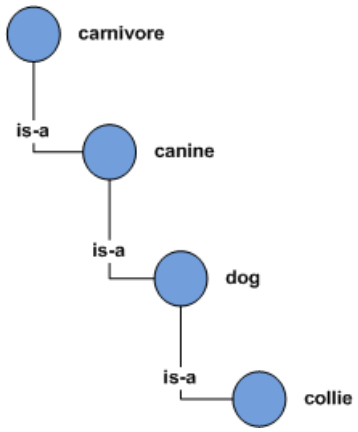

Abbildung 4.2.: WordNet-Ontologien

4.2.1.2. Suche

Ziel ist es, die Suchanfrage vor der Suche in der Komponente *Peer-to-Peer-Netzwerk* mit Semantik aus den Ontologien anzureichern und so eine *Ähnlichkeitssuche* zu ermöglichen. Jeder Knoten und damit jedes Konzept besitzt einen eindeutigen Bezeichner. Dieser Bezeichner wird durch die Ontologien vorgegeben und ermöglicht so eine eindeutige Zuweisung von Konzept und Knoten. Wird mit einem *keyword* die Suche im *OSN* angetreten, so ist zunächst der für das Konzept verantwortliche Peer zu lokalisieren. Ist er gefunden, können durch das Kontaktieren der Nachbarknoten verwandte Begriffe gesammelt werden, da die Netzwerktopologie der Struktur von Ontologie folgt.
Der suchende Peer erhält somit als Antwort auf die Anfrage beliebige verwandte Begriffe, mit anderen Worten die **rekonstruierte Ontologie**.

4.2.2. Peer-to-Peer-Netzwerk

Die *SA Net*-Komponente *Peer-to-Peer-Netzwerk* kann nach Sangpachatanaruk und Znati (2004) ein beliebiges strukturiertes Peer-to-Peer-Netzwerk sein. Die Anforderung 4.1.2.2

Peer-to-Peer-Netzwerktyp spezifiziert Chord als Overlay-Netzwerk. In ihm werden die Ressourcen und Peers organisiert.
Abbildung 4.3 zeigt den logischen Aufbau dieses Overlay-Netzwerkes. Er gleicht im Gegensatz zum *OSN* einer Ringstruktur (siehe Abschnitt 2.3.4 *Chord*).

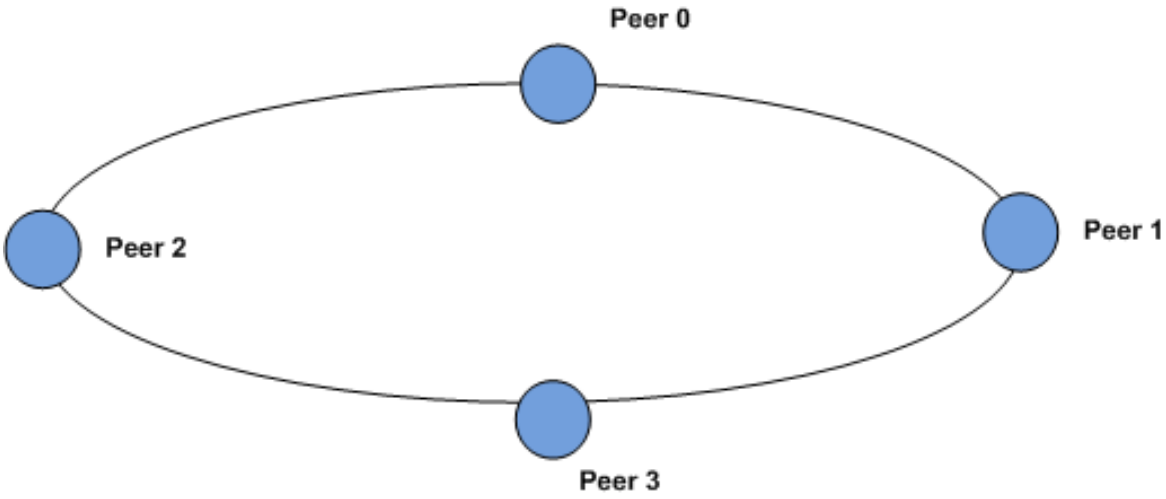

Abbildung 4.3.: Chord-Topologie

4.2.3. Semantic Driven Hashing

Zur semantischen Anreicherung der Suche dient dem *SA Net* der *Semantic Driven Hashing* Algorithmus, kurz *SDH*. Dessen Kernaufgabe ist die Konsultierung des *OSN*, um verwandte Begriffe zu einer Suchanfrage zu erhalten.

4.2.3.1. Der Algorithmus

Der *SDH*-Algorithmus besteht im Wesentlichen aus zwei Teilen:

1. Ermittelung verwandter Konzepte (Zeile 2)
2. Erzeugung der OSN_keys (Zeile 6)

```
1   SDH(Keyword K){
2     OSN_id[] <- Ontology.discovery(K)
3     //retrieve ontology IDs from the keyword using OSN
4     OSN_keys[][]
5     for i = 0 to OSN_id.length do
6           {obtain DHT key for each OSN_id}
7             for j =0 to NUMBER_REPLICAS do
8                 {one concept has several replikas for fault-
                      tolerance}
9               OSN_keys [i][j]<-Hash(OSN[i]+j)
10            end for
11        end for
12        return OSN keys;
13    }
```

Listing 4.1: SDH Pseudo-Algorithmus

Die Suchanfrage, bestehend aus einem String, wird als Parameter für die *Ontology.discovery()*-Funktion des *OSN* übergeben. Das *OSN* (siehe Abschnitt 4.2.1) liefert dann die global gültige OSN_ids der verwandten Konzepte.
Die Hash-Funktion des *SA Net* erzeugt im nächsten Schritt aus den ermittelten OSN_id die entsprechenden OSN_keys, die als Parameter für die get()-Funktion des zweiten Overlay-Netzwerkes dienen. Der Schlüsselraum der Hash-Funktion gleicht dabei dem des Overlay Netzwerkes in Abschnitt 4.2.2. Die einzelnen Phasen des *SDH*-Algorithmus verdeutlicht Abbildung 4.4.
Die Zweidimensionalität des OSN_keys-Arrays erklärt sich durch die Verwendung von Repliken. Wählt man nach Anforderung *Peer-to-Peer-Netzwerktyp* aus Abschnitt 4.1.2.2 *Chord* als Peer-to-Peer-Netzwerk- Komponente des *SA Net*, so werden die OSN_keys als Parameter *id* der suche()-Funktion (siehe Abschnitt 2.3.4.2 *Suche in Chord*) übergeben.

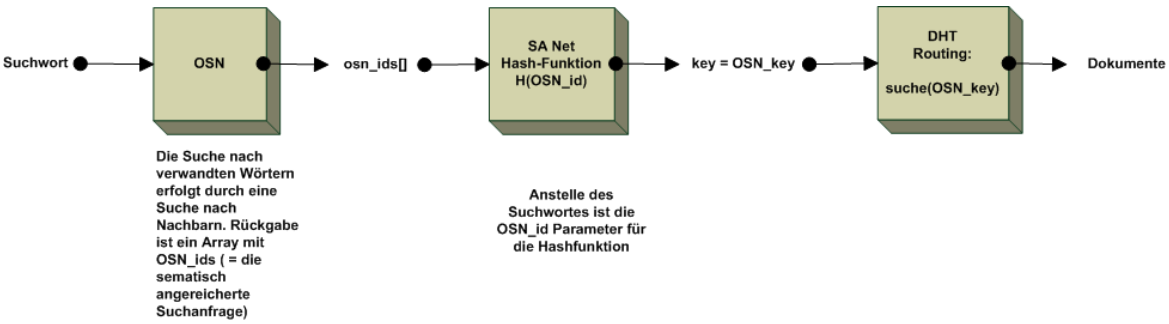

Abbildung 4.4.: Semantic Driven Hashing

4.3. Analyse

Das in Abschnitt 4.2.1 eingeführte Peer-to-Peer-System *SA Net* soll ausgehend von den Anforderungen aus Abschnitt 4.1 und den Komponenten aus Abschnitt 4.2 einer näheren Analyse unterzogen werden. Es gilt, die Anforderungen hinsichtlich ihrer Umsetzbarkeit und Gültigkeit gegenüber denen in Abschnitt 2.3.2 aufgestellten *Peer-to-Peer-Eigenschaften* zu bewerten.
Zum Abschluss tritt die Frage nach der Machbarkeit des *SA Net*-Konzeptes in den Vordergrund.

4.3.1. Bewertung der funktionalen Anforderungen

Einsatz von Ontologien Ontologien kommen wie gefordert zum Einsatz, da die *OSN*-Topologie auf ihnen basiert. Die Konzepte und Relation aus WordNet bilden einen „globally defined ontology map“ (Sangpachatanaruk und Znati, 2004, Kapitel 2, Absatz 3).
Details zur Umsetzung und etwaige Hürden folgen im Abschnitt 4.3.2.

Non Perfect-Matching / Multiple-Keys/ Ähnlichkeitssuche Das Ermitteln der OSN_id zu einem Suchwort allein stellt nicht sicher, dass auch Konzepte gefunden werden, die wie in Abschnitt 4.1.1.2 *Perfect-Matching* beschrieben vom Suchwort abweichen. Stellt der Anwender eine Anfrage, die Rechtschreibfehler enthält oder die von der Wortform her nicht mit den verwalteten Konzepten im *OSN* übereinstimmt, schlägt die Suche schon an dieser Stelle fehl.
Die Nutzung von *Multiple-Keys* und eine *Ähnlichkeitssuche* finden statt. Der *SDH*-Algorithmus ermittelt im *OSN* zu einer Suchanfrage verwandte Konzepte und hinterlegt sie im *OSN_id*-Array . (siehe Abschnitt *Semantic Driven Hashing* 4.2.3).

4.3.2. Bewertung der nicht-funktionalen Anforderungen

Dezentrales Modell / Peer-to-Peer-Netzwerktyp Der *SA Net*-Entwurf sieht als Basis ein *DHT*-basiertes Overlay-Netzwerk vor (Sangpachatanaruk und Znati, 2004, Kapitel 1, Absatz 3). Die Organisation der Daten ist durch die Verwendung von *Chord* sichergestellt.
Der kritische Punkt ist die Art des Overlay-Netzwerkes, auf dem das *OSN* basieren soll. Hier setzt der folgende Paragraf an.

Ontologiebasierte Topologie Es ist offensichtlich, dass eine andere Art der Organisation der Peers als in Peer-to-Peer-Netzwerken mit verteilten Hash-Tabellen zum Einsatz kommen muss. Die Position der Peers im Overlay-Netzwerk ist lediglich von der Anwendung der Hash-Funktion über die IP-Adresse abhängig. Nachbar eines Peers ist derjenige, dessen Hashwert der IP-Adresse linear folgt. Das *OSN* erfordert demgegenüber eine Zuweisung von:

- Peer zu Konzepte
- Zeigern auf Peers mit verwandten Konzepte (semantische Relation)

Die Zuweisung muss dabei nach dem Vorbild der verwendeten Ontologien erfolgen.

Fehlertoleranz Die Organisation des *OSN* sieht vor, dass ein Konzept bei mehreren Peers hinterlegt ist. Ist ein Peer nicht mehr erreichbar, kann die zu einem Konzept gehörige OSN_id laut Lösungsvorschlag von einem zweiten Peer bezogen werden.

4.3.3. Machbarkeit

Zur Umsetzung des *OSN* musst jeder Peer eine Liste mit Einträgen der Art (Wort, Relationstyp, OSN_id) besitzen. Diese Liste enthält Konzepte, die mit dem knoteneigenen Konzept in Beziehung stehen. Sie bildet die Ontologien ab. Der Peer mit Zuständigkeit für das Konzept „dog“ speichert somit als relevantes Wissen seine OSN_id und Zeiger auf seine Nachbarn.

Konzept	Relationstyp	OSN_id
carnivore	Hyperonym	1234
collie	Hyponym	5678

Um diese Liste zu füllen und Konzepte wie Relation auf Peers und Zeigern abzubilden, ist eine zentrale Instanz, einer Manager, nötig. Dieser besitzt einen Vorrat an Ontologien und organisieren nach deren Vorbild das *OSN*. Bei Eintritt in das SA Net weist dieser Manager

dem neuen Teilnehmer ein Konzept zu und richtet die Zeiger auf Peers mit verwandten Konzepte.
Ein anderer Ansatz ist, die Ontologien bei allen Peers im OSN zu hinterlegen. Tritt ein neuer Peer dem *OSN* bei und stellt den initialen Kontakt zu einem schon im Netzwerk befindlichen Peer her, wird diesem der Vorrat an Ontologien übertragen.
Ungelöst bleibt die Frage der Organisation. Es muss entschieden werden, für welches Konzept der neue Peer zuständig ist und wie er den Kontakt zu den Peers herstellt, die für die verwandten Konzepte verantwortlich sind.

4.4. Fazit

Die Analyse hat verdeutlicht, dass das *SA Net* und insbesondere das *OSN* die aufgestellten Anforderungen nicht erfüllen können. In der Form, wie es in dem Paper Sangpachatanaruk und Znati (2004) vorgeschlagen wird, ist das *SA Net* nicht zu realisieren.
Insbesondere die in Abschnitt 4.3.1 erläuterte Notwendigkeit einer zentralen Instanz steht im starken Gegensatz zu der elementaren Anforderung in Abschnitt 4.1.2.1. Das folgende Kapitel stellt ein Prototyp vor, der die oben genannten Anforderungen realisiert und ein dezentrales Modell bietet.

5. Prototyp: DHTPlusO

Bei der Entwicklung des Prototyps soll die Erhaltung der Einfachheit von *DHT*-basierten Peer-to-Peer-Netzwerken im Vordergrund stehen. Dabei gilt es, die konzeptionelle Schwäche des *OSN* (siehe Abschnitt 4.3 *Analyse*) zu umgehen und einen *Single-Point-of-Failure* zu vermeiden.
Im folgenden Kapitel stehen Architektur und Design des Prototyps im Mittelpunkt.

5.1. Anforderungen

In diesem Abschnitt werden Anforderungen aufgestellt, die für den Prototyp im Vordergrund stehen und in dessen Entwicklung einfließen. Sie ergänzen die *Anforderungen* aus Abschnitt 4.1. Zudem macht die *Analyse* aus Abschnitt 4.3 und der Zwang nach einem zum *SA Net* abgewandelten Prototyp die Dokumentation weiterer *Anforderungen* nötig.

5.1.1. Funktionale Anforderungen

5.1.1.1. Art der Suchanfrage

Die vom Anwender gestellte Suchanfrage soll aus einem einzigen Substantiv bestehen. So ist es zunächst nicht vorgesehen, auf *Benutzerebene* (siehe Abschnitt 2.3.1) ganze Sätze bzw. mehrere beliebige Wörter als Anfrage verwenden zu können. Erst auf *Dienstebene* geschieht mittels Suchanfrage und den Ontologien die Nutzung von *Multiple-Keys* (siehe Abschnitt 4.1.1.3).

5.1.1.2. Konfigurierbarkeit

Den Umfang der Ontologie-Nutzung soll der Anwender nach

- Art der Relationen und
- der zur Anreicherung der Suche herangezogene Tiefe

beeinflussen können.
Die *Art* der Relation richtet sich nach denen in der Anforderung 4.1.1.1 *Einsatz von Ontologien* spezifizierten Typen.
Mit *Tiefe* ist gemeint, wie weit die Suche nach verwandten Begriffen auch auf die zuvor gefundenen Wörter fortgesetzt wird.

5.1.2. Nicht-funktionale Anforderungen

5.1.2.1. Ressourcentyp

Als Ressourcentyp sind zunächst Dokumente angedacht, aber konzeptionell soll darauf geachtet werden, dass die Suchfunktionalität nicht vom Ressourcentyp abhängt.

5.1.2.2. Erweiterbarkeit

Die Anforderung 4.1.2.3 *Ontologiebasierte Topologie* beschreibt die Verwendung von WordNet-Ontologien. Diese ermöglichen die Nutzung von *Konzepten*, die lexikalisch in Beziehung zum Suchwort stehenden.
Denkbar ist aber auch, andere Ontologien zu gebrauchen. Am Beispiel der Tierfreunde im Kapitel 3 *Vision* verdeutlicht scheinen Ontologien aus der Domain *animal* naheliegend. Der Prototyp *DHTPlusO* soll dahin gehend erweiterbar sein, dass die WordNet Ontologien durch andere Ontologien ersetzt werden können.

5.2. Marktanalyse

Um die Anforderungen aus den Abschnitten 4.1 und 5.1 umsetzen zu können, sind Frameworks und Bibliotheken nötig.
Der Prototyp soll zum einen auf WordNet-Ontologien zugreifen können und zum anderen ein Peer-to-Peer-Netzwerk erweitern. Als obligatorisch gelten Open Source Software und eine Java-Implementierung.

5.2.1. Overlay-Framework

Die Marktanalyse zeigt eine Auswahl der verfügbaren Overlay-Frameworks nach PDOS und MIT (2008). Beachtet werden folgende Punkte:

- Evaluationsfunktionalität
- Dokumentation und Community

5.2.1.1. Open Chord

OpenChord der Universität Bamberg [Loesing und Kaffille (2008)] besticht durch seine Kompaktheit. Es fehlen eine Evaluationsmöglichkeit und umfangreiche Dokumentation.

5.2.1.2. Overlay Weaver

Großes Plus des *Overlay Weaver*-Frameworks vom „National Institute of Advanced Industrial Science and Technology“ (AIST) [Shudo (2008)] sind die Evaluationsfunktionen wie die Messung von versendeten Nachrichten oder der „Distributed Environment Emulator“. Zudem existiert zu diesem Projekt eine aktive Community und umfangreiche Dokumentation.

5.2.2. WordNet API

Um auf die WordNet-Datenbank zugreifen zu können, ist eine API nötig. Miller (2008) listet hierzu eine Reihe von Projekten auf. An dieser Stelle wird eine Auswahl auf folgende Merkmale untersucht:

- Zugriff auf Konzepte **und** Relationen
- Dokumentation und Community

5.2.2.1. Java API for WordNet Searching (JAWS)

Die *Java API for WordNet Searching (JAWS)* [Spell (2008)] ist eine sehr schlanke WordNet-API, die bei der Suche nach verwandten Wörtern nicht alle WordNet Relationen nutzt. Die Dokumentation ist umfangreich.

5.2.2.2. Java WordNet Library (JWNL)

Die gut dokumentierte *Java WordNet Library (JWNL)* [JWNL Project (2008)] ermöglicht eine effektive und vollständige Gewinnung von verwandten Konzepten und Relationen. Sie ist damit erste Wahl.

5.3. Architektur

Der folgende Abschnitt stellt die Architektur des *Overlay Weaver*-Framework sowie der *Java WordNet Library* dar und gibt einen detaillierten Einblick in deren Nutzung.

5.3.1. Overlay Weaver-Framework

5.3.1.1. Komponenten

Das *Overlay Weaver*-Framework versteht sich laut Entwickler als „[...] overlay construction toolkit“ [Shudo (2008)]. Es vereint lose gekoppelt Komponenten wie „Routing algorithm, „Message service“ oder „Directory service“. Die Abbildung 5.1 verdeutlicht die modulare Architektur.

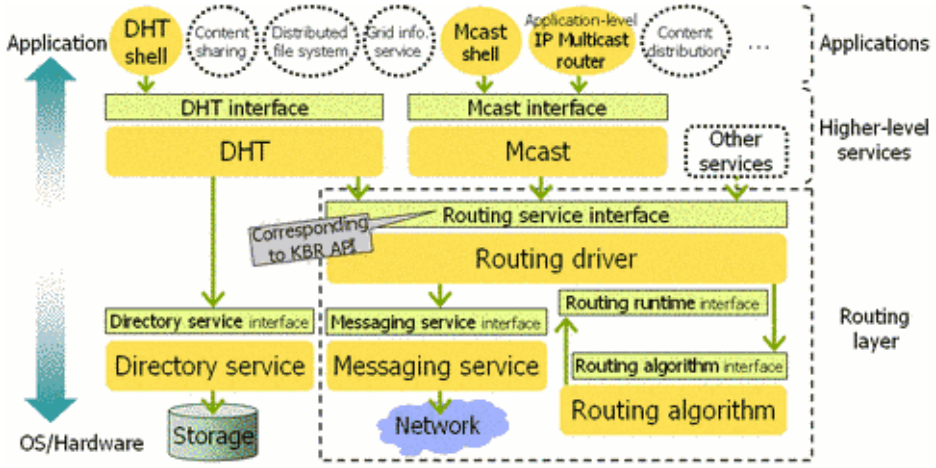

Abbildung 5.1.: Komponenten zur Organisation der Laufzeitumgebung des Overlay Weaver

Sie ist in drei Schichten Unterteil:

Routing-Schicht Besteht aus den Diensten *Routing driver*, *Messaging service* und *Routing algorithm*. Sie bietet eine Auswahlmöglichkeiten zwischen `iterativem` oder `rekursivem` Routing, den Internetprotokollen `UDP` oder `TCP` und Routing-Algorithmen von `Chord` bis hin zu `Tapestry`.

Higher-level service-Schicht Stellt Dienste wie **Multicast** oder **DHT** zur Verfügung.

Applications-Schicht Offeriert passend zur *Higher-level service*-Schicht Anwendungen wie beispielsweise die `DHT Shell`. Mit ihr lässt sich via Terminal ein *DHT*-basiertes Peer-to-Peer-Netzwerk betreiben.
Die modulare Architektur bietet Programmierern die Möglichkeit, in allen drei Schichten eigene Dienste zu implementieren. Hinzu kommt die Option, ein Overlay-Netzwerk mit zahlreichen Peers zu emulieren und evaluieren.

5.3.2. JWNL

Die *Java WordNet Library (JWNL)* managt via API Funktionen den Zugriff auf eine WordNet-Datenbank. Dabei ist eine getrennte Suche nach Wörtern oder Relationen möglich. Die Suchergebnisse werden als verkettete Listen repräsentiert.
Die Bibliothek bietet zahlreiche Konfigurationsmöglichkeiten. Die WordNet-Daten können via Datei, Datenbank oder „in-memory-map“ zur Verfügung gestellt werden.

5.3.3. Fazit

Der Abschnitt hat die Architektur des *Overlay Weaver*-Frameworks und die Möglichkeiten der *Java WordNet Library (JWNL)* vorgestellt. Damit ist Voraussetzung zum Verständnis des Abschnittes 5.4 *Design* geschaffen.

5.4. Design

Der Prototyp *DHTPlusO* umfasst Erweiterungen auf der *Higher-level service*- und der *Applications*- Schicht des *Overlay Weaver*-Frameworks. Abbildung 5.2 gibt in Verbindung mit dem Komponentendiagramm aus 5.1 einen ersten Überblick.

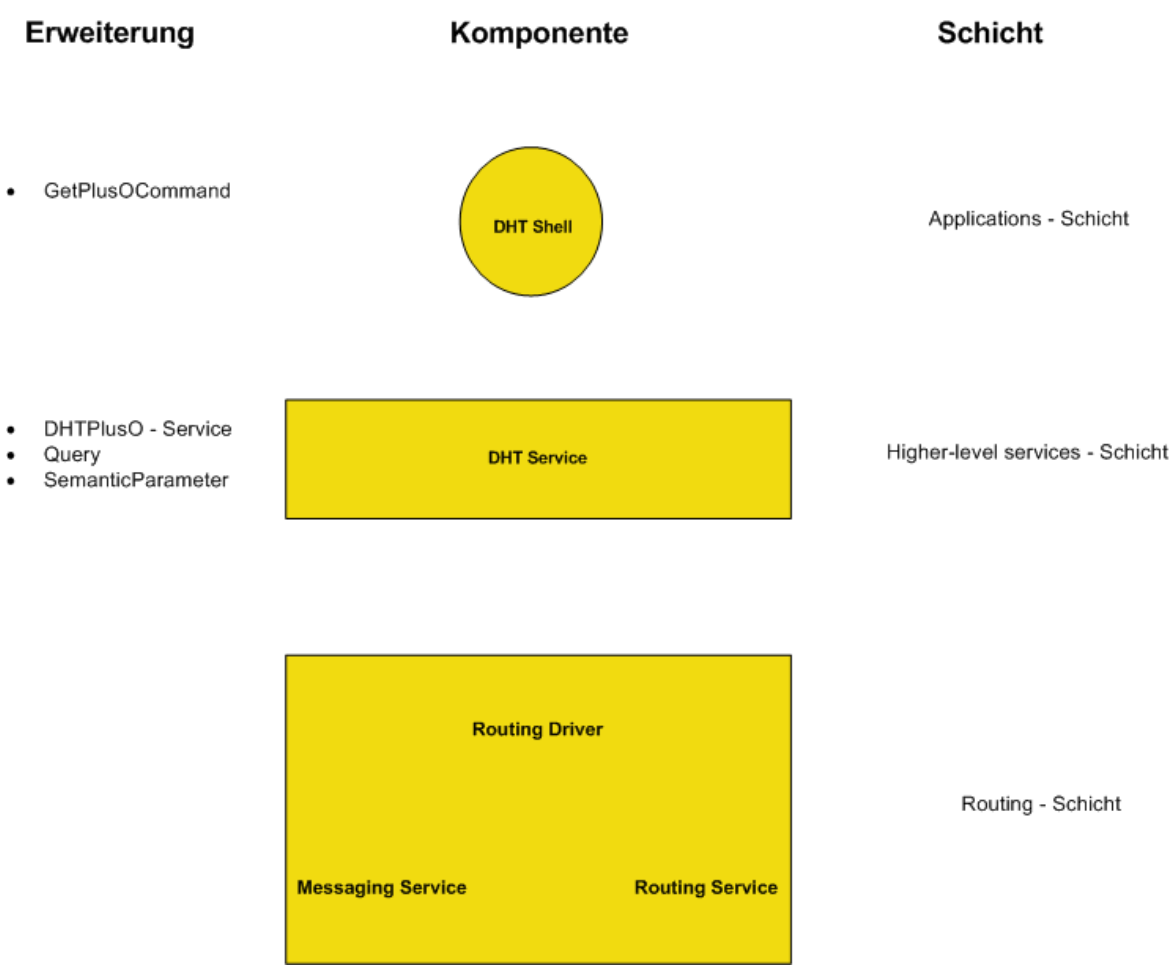

Abbildung 5.2.: Overlay Weaver-Erweiterungen

Folgend werden Funktionsweise und Aufgaben der einzelnen Komponenten näher beschrieben.

5.4.1. Higher-level services - Schicht

5.4.1.1. DHTPlusO-Service

Der *DHTPlusO-Service* fungiert als Wrapper für die vom Framework bereitgestellte *DHT*-Implementation. Die Konfiguration des Service richtet sich nach der festgelegten Anforderung in Abschnitt 4.1.2.2. Der eingesetzte Routing-Algorithmus ist daher *Chord*.

Um die Anforderungen aus Abschnitt 4.1.1 zu realisieren, wurde die get()-Funktion der DHT-Imlplementation erweitert. Die unmodifizierte Variante ermittelt Dokumente, dessen Dateinamen gemäß *Perfect-Matching* exakt mit der Suchanfrage übereinstimmen.
An dieser Stelle setzt *DHTPlusO* an. Der Prototyp ermittelt vor der Suchanfrage an das Netzwerk mittels WordNet-Zugriff (siehe Abschnitt 5.4.1.2) verwandte Wörter. Zu diesen wird dann ebenfalls eine Suchanfrage gestellt. Als Ergebnis stehen - soweit im Netzwerk vorhanden - Dokumente zu allen Begriffen zur Verfügung.

5.4.1.2. WordNet-Zugriff

Unter Beachtung der Anforderung in Abschnitt 4.1.2.1 besitzt jeder Peer eine lokale WordNet-Datenbank. So wird durch Replikation ein *Single-Point-of-Failure* vermieden.
Der Zugriff geschieht über die *OntologyFactory*. Diese ermöglicht die Nutzung verschiedener WordNet-Provider, beispielsweise auch *JAWS* aus dem Abschnitt 5.2 *Marktanalyse*. Abbildung 5.3 zeigt ein Klassendiagramm zu der Verbindung zwischen *OntologyFactory* und der *Java WordNet Library (JWNL)*-Kapselung.

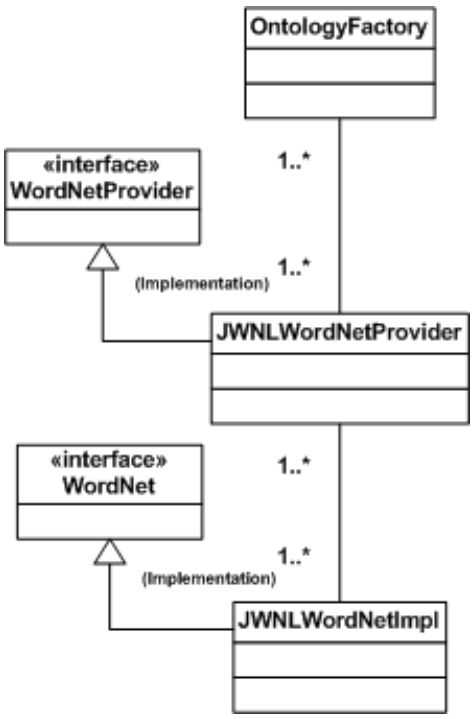

Abbildung 5.3.: OntologyFactory

Nach der Spezifikation in Abschnitt 5.3.2 kommt die JWNL-API zum Einsatz. In der Klasse *JWNLWordNetImpl* wird der Zugriff gestaltet. Die in 5.1.2.2 geforderte Erweiterbarkeit ist durch die Verwendung der *OntologieFactory* sichergestellt, die Kopplung des *DHTPlusO Service* mit der WordNet-Datenbank ist lose und kann durch andere „Ontologie-Quellen“ ersetzt werden.

5.4.1.3. SemanticParameter

Die Klasse *SemanticParameter* ist eine fachliche Klasse, die Parameter für den Zugriff auf die Ontologien sammelt. Im Fall des Prototyps beziehen sie sich auf die WordNet-Ontologien und setzen Anforderung 5.1.1.2 um.
Parameter sind:

- die Tiefe der Relationen ≙ rel_depth
- der Relationstyp ≙ rel_type

Der Parameter *rel_depth* besitzt folgenden Wertebereich:

Wert	Relationstyp	Bemerkung
1	Hyperonyme/Hyponyme	Ober-/Unterbegriffe
2	Holonyme/Meronyme	Teil-Ganzes
3	Hypero-/Hypo-/Holo-/Meronyme	Ober-/Unterbegriffe, Teil-Ganzes

Es besteht für den Anwender die Möglichkeit, bei der Suche nach verwandten Konzepten die Art sowie Tiefe der Relationen festzulegen. Unter Tiefe der Relationen ist zu verstehen, inwieweit die Suche nach verwandten Konzepten auf die Treffer fortgesetzt wird.
Die Auswahl macht es zum einen möglich, beide Relationspaare -also *Hyperonymie* und *Hyponymie* sowie *Holonymie* und *Meronymie*- separat zu nutzen. Auf der anderen Seite können auch alle vier Relationen zusammen für die Suchanfrage herangezogen werden.

5.4.1.4. Query

Die DHT-Implementation aus dem *Overlay Weaver*-Framework nutzt als Datenbehälter das sogenannte *ValueInfo*-Objekt. Es repräsentiert ein im Netzwerk gespeichertes Datum.
Die DHTShell wandelt auf Anwendungsebene die Suchanfrage (z.B. „dog“) via kryptografischer Hash-Funktion in eine *ID*-Objekt um. Die DHT-Implementationen nutzen dann in der *Higher-level service*-Schicht diese *ID* als Parameter für die get(key)-Funktion. Von der *ID* her lassen sich damit keine Rückschlüsse auf den Suchstring machen.
Damit auf dieser Ebene unabhängig von jeglichen Anwendungen verwandte Konzepte für

die Suche herangezogen werden können, muss das *ValueInfo*-Objekt um das zugehörige Suchwort im Klartext erweitert werden. Hierzu dient das *Query*-Objekt. Es speichert die Suchanfrage als String und ist Teil des *ValueInfo*-Objektes.

5.4.2. Applications - Schicht

5.4.2.1. GetPlusO-Befehl

Der *GetPlusO*-Befehl (GetPlusOCommand) ermöglicht dem Anwender durch die oben genannten Erweiterungen die semantisch angereicherte Suche.
Vor der Suche im Overlay-Netzwerk werden zu einem Suchwort aus der WordNet Datenbank je nach semantischen Parametern verwandte Konzepte ausgelesen. Erst dann folgt die Suche im Peer-to-Peer-Netzwerk.
Dabei wird die bestechende Einfachheit der Organisation und Suche des *Chord* Overlay-Netzwerkes nicht aufgebrochen.

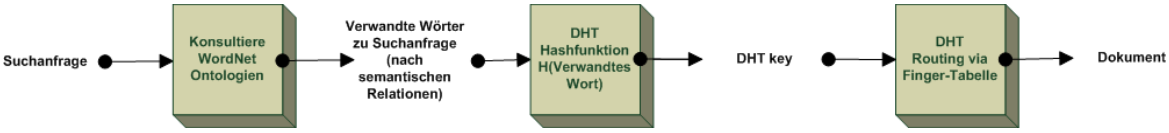

Abbildung 5.4.: Suche mit *DHTPlusO*

5.4.3. Sequenzdiagramm

Das *Overlay Weaver*-Framework ist auf zwei Schichten um *DHTPlusO*-Funktionalität ergänzt worden. Die Erweiterungen dienen der Realisierung der Anforderungen in Abschnitt 4.1.
Das Sequenzdiagramm in Abbildung 5.5 zeigt detailliert die Interaktion zwischen den Objekten.
In der *Applications*-Schicht nimmt die DHTShell die Anfrage der Form `getpluso dog 10 3` entgegen und wandelt die Eingabe in *Concept*- und *SemanticParameter*-Objekte um. Diese dienen auf der *Higher-level service*-Schicht der *getDHTPlusO()*-Funktion des *DHTPlusO*-Services als Parameter. Im weiteren Verlauf findet der via *OntologyFactory* und *JWNL* die Konsultierung der WordNet-Datenbank statt. Als Ergebnis werden die ermittelten Wörter an die *Applications*-Schicht weitergereicht.

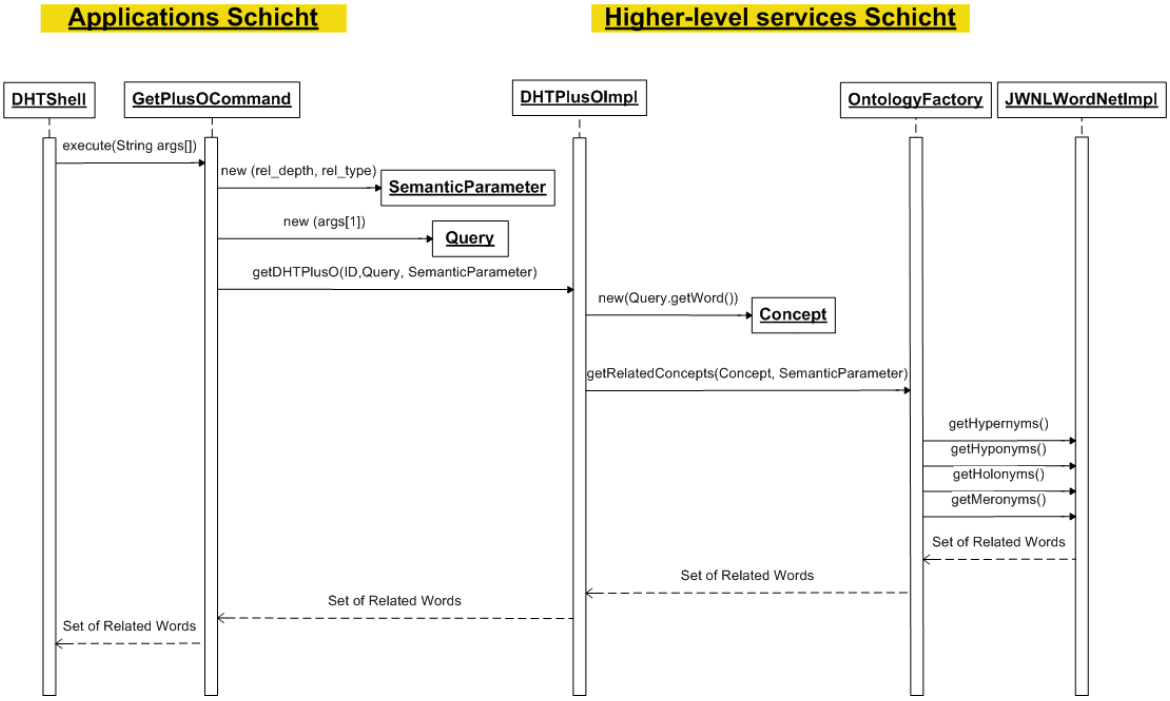

Abbildung 5.5.: Sequenzdiagramm

5.5. Anwendungsbeispiel

Um die Funktionsweise von *DHTPlusO* an einem Beispiel zu verdeutlichen, wird das Szenario aus der Vision in Kapitel 3 aufgegriffen. Peer A und Peer B möchten Dokumente zum Thema *dog* austauschen. Zunächst werden beide Peers initialisiert.

```
DHT configuration:
  hostname:port:     stefanPC/192.168.178.20:3997
  transport type:    UDP
  routing algorithm: Chord
  routing style:     Iterative
  directory type:    VolatileMap
  working directory: .
A DHT started.
Ready.
```

```
DHT configuration:
  hostname:port:     stefanPC/192.168.178.20:5998
  transport type:    UDP
  routing algorithm: Chord
  routing style:     Iterative
  directory type:    VolatileMap
  working directory: .
  initial contact:   3997
A DHT started.
Ready.
```

Abbildung 5.6.: Initialisierung von Peer A und B

Im nächsten Schritt platzieren die Peers via *put*-Befehl Dokumente im Netzwerk. Die Zahl 60000 steht dabei für die Time-to-Live (TTL) des Key-Value Paares.

```
Peer A:  put dog dog.doc 60000
         Ready.

Peer B:  put carnivore carnivore.doc 60000
         Ready.
```

Abbildung 5.7.: Platzierung der Dokumente im Netzwerk

Peer B nutzt zur Suche für die Anfrage *dog* zunächst den *get*-Befehl. Als Ergebnis erhält er wie erwartet das Dokument *dog.doc*.
Beim zweiten Versuch werden für den *getpluso*-Befehl die Parameter rel_depth=1 und rel_type=1 (*Hyperonymie*/*Hyponymie*) verwendet.

```
Peer B:  get dog
         key:   e49512524f47b4138d850c9d9d85972927281da0
         value: dog.doc 59977
         Ready.

         getpluso dog 1 1
         key:   e49512524f47b4138d850c9d9d85972927281da0
         value: dog.doc 59935
         Ready.

         getpluso dog 10 3
         key:   e49512524f47b4138d850c9d9d85972927281da0
         value: carnivore.doc 59927
         value: dog.doc 59909
         Ready.
```

Abbildung 5.8.: Suche mit dem getpluso-Befehl

Die Informationen aus der WordNet-Datenbank reichen noch nicht aus, um verwandte Dokumente im Netzwerk zu lokalisieren. Erst mit der Verwendung der Parameter `10 3` (Hypero-/Hypo-/Holo-/Meronyme) und der damit verbundenen höheren Anzahl an verwandten Wörtern erhält der Peer auch das Dokument zum verwandten Suchwort „carnivore“. Abbildung 2.2 aus den Grundlagen zeigt, dass „carnivore“ keine direktes *Hyperonym* zu „dog“ ist, sondern erst auf zweiter Stufe folgt (rel_depth=2).

6. Evaluation

Die Evaluation des Prototyps gliedert sich in drei Abschnitte. Zunächst wird das von Sangpachatanaruk und Znati (2004) vorgeschlagenen *SA Net* dem Prototyp gegenübergestellt. Ermittelt wird, in welchem Umfang die aufgestellten Anforderungen erfüllt werden.
Zum anderen soll ausgewertet werden, welche Auswirkung *DHTPlusO* auf Parameter wie die Nachrichtenanzahl oder die Zugriffszeit hat.
Abschließend erfolgt in Abschnitt *TREC Korpus* ein Testlauf mit einem Netzwerk, das 100 Peers und gut 1500 Dokumente umfasst.

6.1. DHTPlusO versus SA Net

Folgende Tabelle stellt das *SA Net* und den Prototyp *DHTPlusO* gegenüber. Im Mittelpunkt steht die Umsetzung der aufgestellten Anforderungen aus Abschnitt 4.1.

Anforderung	**SA Net**	**DHTPlusO**
Funktionale:		
Einsatz von Ontologien	*OSN*	WordNet
Non-*Perfect-Matching*	-	DHTPlusO-Service
Multiple-Keys	*SDH*	DHTPlusO-Service
Ähnlichkeitssuche	*SDH*	GetPlusO-Befehl
Nicht-funktionale:		
Dezentrales Modell	nein: OSN Manager	ja: WordNet-Repliken
Peer-to-Peer-Netzwerktyp	*Chord*	*Chord*
Ontologiebasierte Topologie	*OSN*	-
Fehlertoleranz	Konzept-Repliken	*DHT*

Die funktionalen Anforderungen erfüllen das *SA Net* und *DHTPlusO* im gleichen Maße. Bis auf das *Non-Perfect-Matching*. Der Prototyp *DHTPlusO* ist im Vorteil, da er via API direkt auf die WordNet-Datenbank zugreift und damit die Wortform nicht relevant ist (siehe Abschnitt 2.2).
Die wesentliche nicht-funktionale Anforderung *Dezentrales Modell* erfüllt nur *DHTPlusO*

durch die Verwendung von WordNet-Repliken. Das *OSN* setzt wie in Abschnitt 4.3.3 erläutert eine zentrale Instanz voraus.
Eine *Ontologiebasierte Topologie* und *Fehlertoleranz* wie in Abschnitt 4.1.1 erläutert besitzt der Prototyp *DHTPlusO* konzeptionell bedingt nicht. Was die Fehlertoleranz betrifft, kann der Prototyp aber auf Mechanismen von *Chord* zurückgreifen. Es sieht eine mehrfache Speicherung von Dokumenten vor (siehe Shudo (2008)).

6.2. Overlay Weaver-Statistics

Das *Overlay Weaver*-Framework besitzt eine Reihe von Evaluationstools, die Wichtigsten sind

- Message Counter (owmsgcounter)
- Distributed Environment Emulator (owemu)
- Emulation Scenario Generator (owscenariogen)

Message Counter Sammelt Informationen über alle im Netzwerk versendeten Nachrichten. Sei es zu *DHT*-spezifischen wie *Get* oder *Put* oder auf *Chord*-bezogene wie *Update_Finger_Table*. Jede Nachricht ist durch Tags mit Metainformationen angereichert.

Distributed Environment Emulator Ermöglicht den Betrieb von zahlreichen Peers auf einem Rechner. Befehle in einer Scenario-Datei steuern den Emulator.

Emulation Scenario Generator Dient der automatisierten Erzeugung von Overlay-Netzwerken mit Hunderten von Peers.

Für die Evaluierung ist relevant, wie groß der Overhead für die semantische Anreicherung ist. Beobachtet werden können das Nachrichtenaufkommen oder die Anfragedauer bei verschiedener Parameterwahl.
Alle Messungen wurden 10-mal wiederholt und der Median gewählt, um eine Robustheit gegen stark abweichende Werte sicherzustellen.

6.2.1. Parameterwahl

Bei allen Testläufen kommen identische Paramaterkombinationen zum Einsatz:

Relationstiefe [rel_depth]	Relationstyp [rel_type]	Erläuterung
1/5/10	1	*Hyperonymie*/*Hyponymie*
1/5/10	2	*Holonymie*/*Meronymie*
1/5/10	3	Hypero-/Hypo-/Holo-/Meronymie

Die Paramaterwahl für die Relationstiefe (rel_depth) richtet sich nach den Analysen in Gangemi u. a. (2003) und Oltramari u. a. (2008). Demnach beträgt sie im Durchschnitt nicht mehr als 10. Mit den zwei weiteren Werten 1 und 5 soll geprüft werden, inwiefern die Anzahl und Güte der Ergebnisse tatsächlich von der Relationstiefe abhängt.
Die Wahl der drei unterschiedlichen Kombinationen von Relationstypen liegt in der Frage begründet, wie Anzahl und Güte der Ergebnisse vom Relationstyp beeinflusst werden.

6.2.2. Ausführungszeit

Mit der Ausführungszeit ist die Zeitspanne gemeint, die vom Aufruf der *getpluso()*-Funktion des *DHTPlusO* Services bis zum Erhalt der Ergebnisse vergeht.
Das Ergebnis besteht aus einem symbolischen Dokument, welches mit dem Befehl *put <suchanfrage> <suchanfrage>.doc* im Netzwerk platziert wurde. In die Ausführungszeit fließt damit die eventuell nicht unerhebliche Zeit für die Übertragung eines großen Dokumentes nicht mit ein.
Beobachtet wird die Auswirkung in Bezug auf die Wahl der semantischen Parameter. Als Referenz dient die normale *get()*-Funktion von Chord.
Suchanfrage: **„dog“**
Referenz: **4 ms**

Relationstiefe	Relationstyp	Zeit [ms]	Ermittelte verwandte Wörter
1	1	975	75
1	2	1180	8
1	3	1437	81
5	1	2242	358
5	2	1662	48
5	3	2770	403
10	1	3096	368
10	2	2295	51
10	3	3396	416

Suchanfrage: **„ear“**
Referenz: **7 ms**

Relationstiefe	Relationstyp	Zeit [ms]	Ermittelte verwandte Wörter
1	1	975	19
1	2	1040	25
1	3	1277	44
5	1	1011	61
5	2	1220	34
5	3	1906	95
10	1	1065	69
10	2	1277	38
10	3	2362	104

Suchanfrage: **„employment“**
Referenz: **9 ms**

Relationstiefe	Relationstyp	Zeit [ms]	Ermittelte verwandte Wörter
1	1	913	44
1	2	779	-
1	3	1075	44
5	1	1356	113
5	2	756	-
5	3	2088	113
10	1	1267	113
10	2	802	-
10	3	2203	113

Erwartungsgemäß steigt die Ausführungszeit des *getpluso*-Befehls gegenüber dem Referenzwert, da auf die lokale WordNet-Datenbank zugegriffen wird. Sie ist aber mit einem Höchstwert von gut 3s bei 416 verwandten Wörtern vertretbar.
Eine genauere Betrachtung zeigt, dass die Dauer der Anfrage stark mit der Anzahl und dem Typ der genutzten Relation korreliert, nicht so sehr mit der Relationstiefe. Die Evaluation mit dem Suchwort „employment“ macht dies besonders deutlich. Die Verwendung der Parameter *5 3* und *10 3* schlägt beide Male mit gut 2s Ausführungsdauer zu Buche.
Einen weiteren Referenzwert liefert die Messung von 10000 aufeinanderfolgenden *get()*-Anfragen in einem klassischen Chord Overlay-Netzwerk. Sie beträgt knapp 9 s.

6.2.3. Nachrichtenanzahl

Folgende Tabelle gibt einen Überblick über die im Netzwerk verschickten Nachrichten in Abhängigkeit zu den verwendeten semantischen Parametern. Der Versuchsaufbau besteht aus n-Peers und einem Netzwerk mit einem Dokument, was für die Messung der Nachrichtenanzahl in Bezug auf *DHTPlusO* genügt.

Peers	Befehl	Nachrichtenanzahl
10	get	14
	getpluso 1 1	230
	getpluso 5 2	1528
	getpluso 10 3	1848
25	get	20
	getpluso 1 1	358
	getpluso 5 2	2658
	getpluso 10 3	2720
50	get	82
	getpluso 1 1	418
	getpluso 5 2	3426
	getpluso 10 3	3540
100	get	154
	getpluso 1 1	582
	getplus 5 2	4404
	getplus 10 3	4612
1000	get	2321
	getpluso 1 1	4457
	getplus 5 2	9044
	getplus 10 3	10685

Die Verwendung des *getpluso*-Befehls führt zu einer erhöhten Nachrichtenzahl, da nicht nur Ressourcen passend zum Suchwort lokalisiert werden, sondern auch Suchanfragen mit den verwandten Wörtern propagiert werden. Relativ zum *get*-Befehl wird durch den *getpluso*-Befehl aber kein übermäßiges Nachrichtenaufkommen erzeugt, in dem Netzwerk mit 1000 Teilnehmern beträgt der Faktor ~5.

6.3. TREC-Korpus

6.3.1. Hintergrund

Die TREC-Korpora [NIST (2008)] des National Institute of Standards and Technology (NIST) sind eine Datensammlung bestehend aus Dokumenten und sogenannten Topics.
Sie dienen als standardisierte Testdaten für Anwendungen auf Gebieten wie „Information Retrieval (IR)“, „Natural Language Processing“ und „Machine Learning Systems“. Ein Korpus beinhaltet zum Beispiel Blogeinträge oder auch Posts aus Newsgroups.

Bei der Evaluierung des Prototyps *DHTPlusO* wird der „TREC Terabyte Track“ [NIST (2006)] genutzt. Er liefert sogenannte „Topic Streams“, die als Ressourcenname dienen sollen und helfen, ein realitätsnahes Netzwerk mit verschiedenen Dokumenten zu simulieren.
Zu beachten ist, dass es sich damit nicht um ein domainspezifisches Netzwerk, wie im Kapitel 3 Vision beschrieben, handelt.

6.3.2. Versuchsaufbau

Um die Daten für das *DHTPlusO*-Netzwerk und den Emulator des *Overlay Weaver*-Frameworks nutzen zu können, wurde ein Programm entwickelt. Es liest automatisiert die Daten aus dem Korpus und schreibt sie in eine Datei mit Steuerungsanweisungen für den Emulator. Jeder Eintrag im Korpus besteht aus einem Topic mit mehreren Wörtern.

DHTPlusO unterstützt auf *Benutzerebene* vorerst nur *Single-Keys* (siehe Abschnitt 5.1.1.1 *Art der Suchanfrage*). Deshalb wird jedes Wort eines Topics einem Peer zugewiesen. Für die Evaluation des Prototyps mittels *TREC-Korpus* wird ein Netzwerk mit 100 Peers eingerichtet und gut 1500 Dokumente platziert. Drei Begriffe aus dem Korpus dienen als Suchanfrage.

In der Ergebnisübersicht gibt die Spalte „verwandte Wörter“ einen Überblick über die Anzahl der ermittelten Wörter aus der WordNet-Datenbank.
Kursiv hervorgehobene Dokumente stellen neue Treffer bezüglich der vorherigen Parameterkombination dar. Bei der Wahl `rel_type = 3` wird die Summe der Ergebnisse für `rel_type = 1` und `rel_type = 2` dargestellt. Informationen zu den Parametern befinden sich im Abschnitt 5.4.1.3 *Semantic Paramter*.

6.3.3. Ergebnisse

Suchanfrage: **„dog“**

Parameter [rel_depth rel_type]	**verwandte Wörter**	**Dokumente**
1 1	75	support.doc stop.doc dog.doc
1 2	-	dog.doc
1 3	75	Summe: 4
5 1	358	support.doc stop.doc dog.doc *being.doc food.doc beagle.doc feist.doc meat.doc substance.doc*
5 2	48	dog.doc *food.doc cell.doc personality.doc form.doc section.doc*
5 3	403	Summe: 13
10 1	368	support.doc stop.doc dog.doc being.doc food.doc beagle.doc feist.doc meat.doc substance.doc
10 2	51	dog.doc food.doc cell.doc personality.doc form.doc section.doc *head.doc*
10 3	416	Summe: 14

Suchanfrage: **„ear“**

Parameter [rel_depth rel_type]	**verwandte Wörter**	**Dokumente**
1 1	19	-
1 2	25	head.doc
1 3	44	Summe: 1
5 1	61	*power.doc struktur.doc sensitivity.doc*
5 2	34	head.doc *being.doc plant.doc tragus.doc*
5 3	95	Summe: 7
10 1	69	power.doc struktur.doc sensitivity.doc
10 2	38	head.doc being.doc plant.doc tragus.doc *section.doc*
10 3	104	Summe: 8

Suchanfrage: **„employment“**

Parameter [rel_depth rel_type]	**verwandte Wörter**	**Dokumente**
1 1	44	employment.doc state.doc services.doc use.doc abuse.doc job.doc service.doc exercise.doc business.doc application.doc
1 2	-	employment.doc
1 3	44	Summe: 10
5 1	113	employment.doc state.doc services.doc use.doc abuse.doc job.doc service.doc exercise.doc business.doc application.doc *act.doc engineering.doc technology.doc*
5 2	-	employment.doc
5 3	113	Summe: 13
10 1	113	employment.doc state.doc services.doc use.doc abuse.doc job.doc service.doc exercise.doc business.doc application.doc act.doc engineering.doc technology.doc
10 2	-	employment.doc
10 3	113	Summe: 13

Der Test mit dem *TREC-Korpus* zeigt, dass der Prototyp *DHTPlusO* eine *Ähnlichkeitssuche* wie in Abschnitt 4.1.1.4 gefordert leisten kann.
So liefert die Suche nach „employment“ Treffer wie *job.doc*, *business.doc* oder *services.doc*, die Suchanfrage „dog“ beispielsweise die Dokumente *beagle.doc* und *meat.doc*.
Was die Anzahl der gefundenen Dokumente betrifft, hat die Erhöhnung der Relationstiefe von 5 und 10 kaum mehr Ergebnisse zutage befördert.
Bei der Art der verwendeten Relation gibt es keine eindeutige Tendenz. Bei dem Suchwort „ear“ liefern *Meronymie*/*Holonymie* (rel_depth = 2) mehr Treffer, bei der Suche nach „dog„ und „employment“ *Hyperonymie*/*Hyponymie* (rel_depth = 1).

6.4. Fazit

Der Evaluation hat bewiesen, dass der Prototyp in der Lage ist, mithilfe „verwandter Wörter“ weitere Dokumente aus dem Netzwerk zu beziehen. Dabei hält sich der Overhead bezüglich der *Ausführungszeit* und des *Nachrichtenaufkommens* im vertretbaren Rahmen. Bei der Weiterentwicklung des Prototyps ist zu überlegen, die Parameterwahl auf rel_depth = 10 und rel_type = 3 festzusetzen, da die Ausführungszeit keine Hürde ist und bei dieser Kombination die meisten Treffer zu verbuchen sind.
Die Evaluation mittels *TREC-Korpus* auch deutlich gemacht, wie elementar Güte und Konsistenz der Ontologien mit ihren Konzepten und Relationen für eine zufriedenstellende Suche sind. Je mehr Relationen genutzt werden und je tiefer nach Wörtern gesucht wird, desto höher ist die semantische Abweichung von der eigentlichen Suchanfrage. Die gefundenen Treffer stehen nur noch sehr entfernt im Zusammenhang mit der Suchanfrage. Bei einer Relationstiefe von 10 liefert *DHTPlusO* für das Suchwort „ear“ beispielsweise das Dokument *section.doc*.
Grundsätzlich ist die Anzahl der Treffer vom konkreten Netzwerk mit seinen verwalteten Dokumenten und der Güte der Ontologie abhängig.

7. Zusammenfassung

7.1. Ergebnisse

Die Arbeit hat verdeutlicht, dass der Lösungsansatz zur Suchoptimierung, dass *SA Net* und im Speziellen das *OSN*, nicht ohne die Verletzung wichtiger Eigenschaften von Peer-to-Peer-Netzwerk auskommen kann (siehe Abschnitt 2.3.2 Peer-to-Peer-*Eigenschaften*). Durch die in Abschnitt 4.3.3 beschriebene Notwendigkeit eines Managers kann die Forderung nach einem dezentralen Modell 4.1.2.1 nicht erfüllt werden. Die Organisation des *OSN* als Overlay-Netzwerk nach Form der WordNet-Ontologien ginge auf Kosten eines *Single-Point-of-Failure*.

Die Idee der Verknüpfung von Ontologien und Peer-to-Peer-Netzwerk ohne Verletzung des Peer-to-Peer-Paradigmas (siehe Abschnitt 2.3.2) leistet der Prototyp *DHTPlusO*. Die lokalen Repliken der WordNet-Datenbank können der Anforderung, die Suchanfrage semantisch anzureichern, gerecht werden. Kapitel 6 und nachdrücklich der Abschnitt 6.3 weisen dies nach.

7.2. Hürden

Ein Hindernis ist die Tatsache, dass die Güte der Suchoptimierung von der Güte der verwendeten Ontologien abhängt. Liefern diese nicht die von den Anwendern verwendeten Konzepte oder wird eine Domain nicht hinreichend beschrieben, leiden Qualität und Quantität der Suchergebnisse. Der Abschnitt 2.2.2 *Ontologiesicht auf WordNet* weist in diesem Zusammenhang auf die Grenzen der WordNet-Ontologien hin.
Zudem hat die Evaluation in Kapitel 6 demonstriert, dass eine erfolgreiche Suche maßgeblich von der Übereinstimmung von Konzepten und Ressourcennamen - besser der Dokumentennamen - abhängt. Der in Abschnitt 2.4.1 beschriebenen *Perfect-Matching*-Problematik kann der Prototyp nicht vollständig entgegenwirken. Er ist aber Dank WordNet (siehe Abschnitt 2.2) in der Lage, unabhängig von der Wortform eine Suche durchzuführen.

Ein weiterer Punkt betrifft die Synchronisation der Ontologie respektive der WordNet-Datenbank-Repliken. Hier muss eine Strategie entworfen werden, um Änderungen und Erweiterungen an alle Netzwerkteilnehmer zu propagieren.
In Abschnitt 4.1.3 wurde darauf hingewiesen, dass die Annahme gelten soll, dass alle Peers vertrauenswürdig sind. Durch die Verwendung von *Query*-Objekten als Parameter der get()-Funktion wird im Netzwerk die Suchanfrage im Klartext verschickt. Dadurch wird die Informationsvertraulichkeit verletzt. Bei der Standardfunktion geschieht dies in Form einer ID, die aus der Anwendung einer globalen, kryptografischen Hash-Funktion gewonnen wird.

7.3. Ausblick

Denkt man an die Suche und eine Weiterentwicklung des Prototyps, so sind ein Ansatzpunkt sicherlich die verwendeten Ontologien. Wie in Abschnitt 4.1.2.1 erläutert, zeichnen sich die WordNet-Ontologien zwar durch ihre Domain-Unabhänigkeit aus. Doch gibt es Anwendungsszenarien, bei denen eine exakte und möglichst vollständige Abbildung der betreffenden Domäne gewünscht ist. Die Suche im Peer-to-Peer-Netzwerk der Tierfreunde aus Kapitel 3 würde durch die Nutzung domainspezifischer Ontologien an Güte gewinnen, da mehr potenziell verwandte Konzepte zur Verfügung stünden.

Ein zweiter Punkt ist der Ansatz, die Ontologien nicht nur bei der Suche nach verwandten Ressourcen heranzuziehen, sondern schon bei der Platzierung im Netzwerk und bei der Angabe eines Suchwortes. Möchte ein Peer eine Ressource im Netzwerk bereitstellen, so wählt er als Ressourcenname ein Wort aus den gemeinsam verfügbaren Ontologien. Ebenso bei der Suche. Natürlich werden dem Anwender dadurch gewissen Schranken auferlegt, die Konsistenz des ganzen Peer-to-Peer-Systems würde aber erhöht werden.

Weiter hat der Abschnitt 5.1.1.1 *Art der Suchanfrage* aufgedeckt, dass eine zukünftige Version des *DHTPlusO* Systems schon auf *Benutzerebene* ein *Multiple-Keys*-Mapping unterstützen sollte. So kann bereits der Anwender mehr als ein Wort als Suchanfrage angeben. Sie werden dann zusammen mit den Wörtern genutzt, die aus den Ontologien ermittelten werden.

Literaturverzeichnis

[Dustdar u. a. 2003] DUSTDAR, Schahram ; GALL, Harald ; HAUSWIRTH, Manfred: *Software-Architekturen für verteilte Systeme*. Berlin and Heidelberg : Springer, 2003. – URL http://lsirpeople.epfl.ch/hauswirth/SA-VS/

[Ehrig und Studer 2006] EHRIG, Marc ; STUDER, Rudi: *Wissensvernetzung durch Ontologien*. Kap. Methoden und Technische Infrastruktur, S. 469–484. In: *Semantic Web. Wege zur vernetzten Wissensgesellschaft*, Springer, 2006

[Fox u. a. 1991] FOX, Edward A. ; CHEN, Qi F. ; DAOUD, Amjad M. ; HEATH, Lenwood S.: Order Preserving Minimal Perfect Hash Functions and Information Retrieval. In: *ACM Transactions on Information Systems* 9 (1991), Juli, Nr. 3, S. 281–308

[Gangemi u. a. 2003] GANGEMI, Aldo ; NAVIGLI, Roberto ; VELARDI, Paola: The OntoWordNet Project: Extension and Axiomatization of Conceptual Relations in WordNet. In: MEERSMAN, Robert (Hrsg.) ; TARI, Zahir (Hrsg.) ; SCHMIDT, Douglas C. (Hrsg.): *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE - OTM Confederated International Conferences, CoopIS, DOA, and ODBASE 2003, Catania, Sicily, Italy, November 3-7, 2003* Bd. 2888, Springer, 2003, S. 820–838. – URL http://springerlink.metapress.com/openurl.asp?genre=article&issn=0302-9743&volume=2888&spage=820. – ISBN 3-540-20498-9

[Gómez-Pérez u. a. 2004] GÓMEZ-PÉREZ, Asunción ; FERNÁNDEZ-LÓPEZ, Mariano ; CORCHO, Oscar: *Ontological Engineering*. Springer, 2004

[Gruber 1993] GRUBER, T. R.: A translation approach to portable ontologies. In: *Knowledge Acquisition* 5 (1993), Nr. 2, S. 199–220

[Gruninger und Lee 2002] GRUNINGER, Michael ; LEE, Jintae: Ontology: applications and design. In: *Communications of the ACM* 45 (2002), Februar, Nr. 2, S. 39–41

[Guarino 1998] GUARINO, Nicola (Hrsg.): *Formal Ontology in Information Systems: Proceedings of the First International Conference*. Amsterdam : IOS Press, 1998

[Hesse 2002] HESSE, Wolfgang: Ontologie(n) - Aktuelles Schlagwort. In: *Informatik Spektrum* 25 (2002), Nr. 6, S. 477–480. – URL http://link.springer.de/link/service/journals/00287/bibs/2025006/20250477.htm

[ipoque 2007] IPOQUE: *Internet Study 2007*. 2007. – URL http://www.ipoque.com/news_&_events/internet_studies/internet_study_2007

[JWNL Project 2008] JWNL PROJECT: *JWNL (Java WordNet Library)*. 2008. – URL http://sourceforge.net/projects/jwordnet/

[Klemm u. a. 2007] KLEMM, Fabius ; GIRDZIJAUSKAS, Sarunas ; BOUDEC, Jean-Yves L. ; ABERER, Karl: On Routing in Distributed Hash Tables. In: *Seventh IEEE International Conference on Peer-to-Peer Computing (P2P'07)*, 2007, S. 113–122

[Li u. a. 2004] LI, Mei ; LEE, Wang-Chien ; SIVASUBRAMANIAM, Anand: Semantic Small World: An Overlay Network for Peer-to-Peer Search. In: *ICNP*, IEEE Computer Society, 2004, S. 228–238. – URL http://csdl.computer.org/comp/proceedings/icnp/2004/2161/00/21610228abs.htm. – ISBN 0-7695-2161-4

[Loesing und Kaffille 2008] LOESING, Karsten ; KAFFILLE, Sven: *Open Chord*. 2008. – URL http://www.uni-bamberg.de/en/fakultaeten/wiai/faecher/informatik/lspi/bereich/research/software_projects/openchord/

[Löser u. a. 2004] LÖSER, Alexander ; SCHUBERT, Kai ; ZIMMER, Frederik: Taxonomy-Based Routing Overlays in P2P Networks. In: *IDEAS*, IEEE Computer Society, 2004, S. 407–412. – URL http://doi.ieeecomputersociety.org/10.1109/IDEAS.2004.60. – ISBN 0-7695-2168-1

[Mahlmann und Schindelhauer 2007] MAHLMANN, Peter ; SCHINDELHAUER, Christian: *Peer-to-Peer-Netzwerke: Algorithmen und Methoden*. Springer-Verlag, 2007

[Miller 2008] MILLER, George A.: *WordNet - a lexical database for the English language*. 2008. – URL http://wordnet.princeton.edu/

[Nagel 2003] NAGEL, Karin: *Ontologien in der stichwortbasierten Suche*, HAW Hamburg, Diplomarbeit, 2003

[NIST 2006] NIST: *TREC 2006 Terabyte Track*. 2006. – URL http://trec.nist.gov/data/terabyte/06/06.efficiency_topics.tar.gz

[NIST 2008] NIST: *Text REtrieval Conference (TREC)*. 2008. – URL http://trec.nist.gov/

[Oltramari u. a. 2008] OLTRAMARI, Alessandro ; GANGEMI, Aldo ; GUARINO, Nicola: *Restructuring WordNet's Top-Level: The OntoClean approach*. 2008. – URL http://citeseer.ist.psu.edu/539629.html;http://www-dii.ing.unisi.it/aiia2002/paper/NLP/oltramari-aiia02.pdf

[Oram 2001] ORAM, Andy: *Peer-to-peer: Harnessing the Power of Disruptive Technologies*. Sebastopol, California : O'Reilly, 2001

[PDOS und MIT 2008] PDOS ; MIT: *The Chord/DHash Project*. 2008. – URL http://pdos.csail.mit.edu/chord/

[Pötter 2006] PÖTTER, Philipp: *Sicherheits- und Performance-Analyse des Gnutella Protokolls*, HAW Hamburg, Diplomarbeit, 2006

[Ratnasamy u. a. 2001] RATNASAMY, Sylvia ; FRANCIS, Paul ; HANDLEY, Mark ; KARP, Richard M. ; SHENKER, Scott: A scalable content-addressable network. In: *SIGCOMM*, 2001, S. 161–172

[Rowstron und Druschel 2001] ROWSTRON, Antony I. T. ; DRUSCHEL, Peter: Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems. In: GUERRAOUI, Rachid (Hrsg.): *Middleware 2001, IFIP/ACM International Conference on Distributed Systems Platforms Heidelberg, Germany, November 12-16, 2001, Proceedings* Bd. 2218, Springer, 2001, S. 329–350

[Sangpachatanaruk und Znati 2004] SANGPACHATANARUK, Chatree ; ZNATI, Taieb: Semantic Driven Hashing (SDH):An Ontology-based Search Scheme for the Semantic Aware Network (SA Net). In: *Fourth IEEE International Conference on Peer-to-Peer Computing (P2P'05)*, 2004, S. 270–271

[Schindelhauer 2008] SCHINDELHAUER, Christian: *Peer-to-Peer Netzwerke*. 2008. – URL http://cone.informatik.uni-freiburg.de/teaching/vorlesung/peer-to-peer-s06/

[Schlosser u. a. 2002] SCHLOSSER, Mario T. ; SINTEK, Michael ; DECKER, Stefan ; NEJDL, Wolfgang: A Scalable and Ontology-Based P2P Infrastructure for Semantic Web Services. In: GRAHAM, Ross L. (Hrsg.) ; SHAHMEHRI, Nahid (Hrsg.): *Peer-to-Peer Computing*, IEEE Computer Society, 2002, S. 104–111. – URL http://csdl.computer.org/comp/proceedings/p2p/2002/1810/00/18100104abs.htm. – ISBN 0-7695-1810-9

[Shudo 2008] SHUDO, Kazuyuki: *Overlay Weaver - An Overlay Construction Toolkit*. 2008. – URL http://overlayweaver.sourceforge.net/

[Spell 2008] SPELL, Brett: *Java API for WordNet Searching (JAWS)*. 2008. – URL http://engr.smu.edu/~tspell/

[Staab 2002] STAAB, Steffen: Wissensmanagement mit Ontologien und Metadaten. In: *Informatik Spektrum* 25 (2002), Nr. 3, S. 194–209. – URL http://link.springer.de/link/service/journals/00287/bibs/2025003/20250194.htm

[Steinmetz und Wehrle 2005] STEINMETZ, Ralf ; WEHRLE, Klaus: *Peer-toPeer Systems and Applications*. Berlin and Heidelberg : Springer, 2005

[Stoica u. a. 2001] STOICA, Ion ; MORRIS, Robert ; KARGER, David ; KAASHOEK, Frans ; BALAKRISHNAN, Hari: Chord: A Scalable Peer-To-Peer Lookup Service for Internet Applications. In: *Proceedings of the 2001 ACM SIGCOMM Conference*, ACM, 2001, S. 149–160. – URL citeseer.ist.psu.edu/stoica01chord.html

[Sure 2004] SURE, Yorke: *Automatische Wissensintegration mit Ontologien*. 2004. – URL http://www.aifb.uni-karlsruhe.de/WBS/ysu/publications/2006_modellierung_mapping.pdf

[Tang u. a. 2003] TANG, Chunqiang ; XU, Zhichen ; MAHALINGAM, Mallik: pSearch: information retrieval in structured overlays. In: *Computer Communication Review* 33 (2003), Nr. 1, S. 89–94. – URL http://doi.acm.org/10.1145/774763.774777

[Zeinalipour-Yazti u. a. 2004] ZEINALIPOUR-YAZTI, Demetrios ; KALOGERAKI, Vana ; GUNOPULOS, Dimitrios: Information Retrieval Techniques for Peer-to-Peer Networks. In: *Computing in Science and Engineering* 6 (2004), Nr. 4, S. 20–26

A. Paper

Anbei das Paper Sangpachatanaruk und Znati (2004) - es dient als Ausgangspunkt der Arbeit.

Semantic Driven Hashing (SDH): An Ontology-based Search Scheme for the Semantic Aware Network (SA Net)

Chatree Sangpachatanaruk[†] Taieb Znati[*,†]
[†]Department of Information Science and Telecommunications [*]Department of Computer Science
University of Pittsburgh, Pittsburgh, PA, 15260

Abstract

The success of personalized resource discovery depends on its ability to allow users to discover, extract and integrate information of interest from heterogeneous sources, and its ability to provide these users with efficient tools to manipulate and convert the discovered information into knowledge. We propose Semantic Aware Network (SA Net), a structured peer-to-peer (P2P) overlay architecture to support the basic functionalities of a personalized resource discovery. The SA Net uses an ontology-based representation of the resources to enable a semantic resource discovery and access that reflects the interest of the user. In this paper, we describe the the SA Net search scheme "Semantic-driven Hashing" (SDH), which uses lexical-based ontology to provide a foundation for indexing and search in structured P2P overlay infrastructure.

1. Introduction

Structured P2P overlay networks have gained attention as a viable solution to effectively facilitate resource discovery and sharing in large scale networks. The basic tenet of this technology is a distributed hash table (DHT), which is used to infer a structure that regulates location and access to a distributed set of resources. Using the DHT information, overlay nodes, which currently manage the resource, can be identified deterministically and in a finite number of steps, thereby reducing considerably the searching overhead due to communication and routing. The properties of structured overlay networks make them a viable solution to support the design of a resource discovery for and enable an effective management of a distributed resource indexes of interests to different communities. The basic DHT-based overlay structure, however, exhibit several shortcomings that need to be addressed in order for such a structure to meet the performance requirements of a personalized resource discovery.

One critical limitation of DHT-based structure, with respect to resource discovery, is its lack of support for location-based queries that go beyond perfect matching. Resource distribution in a structured P2P overlay network is based on a hash table, which is used for looking up a specific item based on a single key. This key is mapped to only one location in the overlay network. As a result, each resource is uniquely handled by a specific node in the network. Resource indexing, on the other hand, requires a flexible scheme to handle multiple-key mapping. The scheme must support resource distribution, whereby resources are spreaded among overlay nodes based on their multiple semantic keys.

To address the limitation of DHT-based structures and meet the design requirement of multiple key-mapping resource discovery scheme, we propose a novel Semantic Aware Network architecture, referred to as SA Net, for personalized Internet resource discovery. Within this structure, we also propose an ontology-based search scheme for a DHT overlay architecture, referred to as "Semantic Driven Hashing (SDH)". In the following, the concept of the SDH, its components, and main algorithms, will be described.

2. SA Net Search Scheme

SA Net is an agent-based system which achieves its semantic richness through the use of explicit ontologies to represent resources. SA Net further enhances the DHT-based resource distribution scheme by using the unique identifier assigned to each ontology as a key to locate the overlay node responsible for maintaining the resource index associated with the underlying ontology. In other words, the ontology-based hashing scheme, thereof referred to as ***"Semantic-Driven Hashing (SDH)"***, utilizes ontologies, instead of resource names, as the hash input to generate the key necessary to distribute the resource among overlay nodes.

To further explain the SDH-based resource discovery scheme, we consider an ontology that describes the concept of *"telecommunication"*, as depicted in Figure 1. Based on the above ontology, a given Internet user can express personal interest in communication by using the keyword *"telecommunication"*. When this keyword is issued as part

Proceedings of the Fourth International Conference on Peer-to-Peer Computing (P2P'04)
0-7695-2156-8/04 $20.00 © 2004 **IEEE**

Abbildung A.1.: Paper Seite 1/2

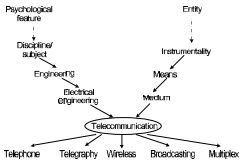

Figure 1. Ontology example.

of a search, the resulting query returns information of all resources which are directly related to the associated concept, namely *telephone, telegraphy, wireless, broadcasting, and multiplex*. When used in conjunction with the keyword of "*wireless*", however, the ontology can be scoped down to the resources that are associated with the new concepts (i.e. telecommunication that is wireless), thereby capturing more specifically the user's interests. In addition, when semantic relations are stored with the concepts among the overlay nodes, networks of resources for a user can be proactively formed to reflect the user interest.

To ensure global consistency in semantic classification, however, SA Net must support a globally defined ontology map which represents the well-known concepts and which can be used to guide an SDH-based search query. As such, given a keyword, SA Net can consistently identify which concept it is related to. To achieve this goal, SA Net uses an Wordnet lexical network, referred to as ***Ontology Semantic Network (OSN)***, where a node represents a concept as described by a set of similar words, and where an edge between two nodes represents the relation between these concepts[1]. Each node in OSN is assigned a numerical ID that is used as the hash input for an SDH-based search.

When a query to locate or advertise a resource is issued, the SDH scheme consults the OSN to obtain the ID of the concept associated with the resource specified in the query. This ID is then hashed to obtain the key, in the DHT space, of the overlay node where the metadata resource index is maintained. This key is then used to route the request to the identified overlay node. The pseudo code of the SDH algorithm is shown in Algorithm 1.

Algorithm 1 SDH algorithm: Retrieve DHT keys for keyword K.

```
1:  SDH(Keyword K){
2:  OSN_id[] ⇐ Ontology_discovery(K)
3:  {retrieve ontology IDs from the keyword using OSN}
4:  OSN_keys[][]
5:  for i = 0 to OSN_id.length do
6:     {obtain DHT key for each OSN ID}
7:     for j = 0 to NUMBER_REPLICAS do
8:        {one concept key has several replicas for fault-tolerance}
9:        OSN_keys[i][j] ⇐ Hash(OSN[i] + j)
10:    end for
11: end for
12: return OSN_keys;
```

3. Related Works

The recent trend to improve resource discovery for the structured P2P networks is to enable semantic-based data location and search by associating semantics into the structure to regulate resource distribution. Recent frameworks propose to use information retrieval techniques, namely vector space model (VSM), and latent semantic indexing (LSI), to obtain semantic representation from the textual information appearing in the resource and queries [4]. Another approach uses XML techniques to generate semantic representation from metadata attributes [2]. The most similar work to the SDH proposed scheme is the approach that uses a concept of global catalog to provide a standard for semantic classification [3]. While the first approach does not work well with non-textual resources, the second approach is likely to provide only perfect-matched result for a search, because of the indexing and searches that depend on keyword attributes. The catalog approach even though provides a better scheme for semantic-based resource classification, its naming still depends on the keywords. SA Net, on the other hand, uses global classification, but contrary to the catalog-based approach, the classification does not depends on word appearances. Instead, it relies on the ontologies that capture the keywords' meanings and semantic relations, which can be used to infer related resources more intuitively.

4. Conclusion

This paper presents the search scheme of SA Net, a semantic aware network to enable personalized resource discovery on the Internet. The SA Net search scheme enhances DHT search by using ontology, a semantic representation, as the hash input instead of filename or simple keywords. The meanings and relations derived from ontologies will allow personalization to evolve effectively. As such, a personalized view of the resources can reflect user's interest in a transparent manner. Several parts of the SA Net are currently being developed. These include personalization built upon this search scheme and mechanisms to provide security and fault tolerance for the SA Net architecture.

References

[1] C. Fellbaum, editor. *WordNet: An Electronic Lexical Database*. The MIT Press, May 1998.

[2] L. Garces-Erice, P. Felber, E. Biersack, K. Ross, and G. Urvoy-Keller. Data indexing in dht peer-to-peer network. In *Proceedings of the 24th International Conference on Distributed Computing Systems (ICDCS-04)*, March 2004.

[3] A. Loser. Towards taxonomy-based routing in p2p networks. In *Proceedings of Workshop on Semantics in Peer-to-Peer and Grid Computing*, pages 35–50, Newyork, USA, May 2004.

[4] C. Tang, Z. Xu, and S. Dwarkadas. Peer-to-peer information retrieval using self-organizing semantic overlay networks. In *ACM SIGCOMM 2003*, pages 175–186, August 2003.

Abbildung A.2.: Paper Seite 2/2

B. CD-ROM Inhalt

Auf der CD-ROM befinden sich in drei Ordnern

- die Bachelorarbeit als PDF-Datei
- der Prototyp *DHTPlusO* als Eclipse-Projekt
- Frameworks + Bibliotheken

Weitere Informationen auch zur Installation und Verwendung des Prototyps bitte der READ-ME.txt auf der CD-ROM entnehmen.

Glossar

DHT Distributed Hash Table - verteilte skalierbare Datenstruktur.

Holonym Teil-Ganzes Beziehung.

Hyperonym Oberbegriff eines geg. Begriffs.

Hyponym Unterbegriff eines geg. Begriffs.

Key Gehashte Suchanfrage.

Konzept Steht für eine Begrifflichkeit einer Domäne und ist Teil einer Ontologie. In WordNet existiert ein Konzept nicht als solches und wird durch ein synset mit Wörtern (Synonymen) repräsentiert.

Meronym Umkehrung des Holonyms.

Suchanfrage Suchstring/Suchwort - entspricht dem gesuchten Ressourcennamen.

Abkürzungsverzeichnis

AIST National Institute of Advanced Industrial Science and Technology (Japan).

DHT Distributed Hash Table - verteilte Hash-Tabelle.

DHTPlusO Distributed Hash Table Plus Ontologien.

JAWS Java API for WordNet Searching.

JWNL Java WordNet Library.

OSN Ontology Semantic Net.

OW Overlay Weaver.

P2P Peer-to-Peer.

SA Net Semantic Aware Network.

SDH Semantic Driven Hashing.

TREC Text REtrieval Conference.

Index

Versicherung über Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach §24(5) ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 30. April 2008	
Ort, Datum	Unterschrift