Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Masterarbeit

Lars Mählmann

Deliver who I mean, automatische Erstellung von Personenprofilen in großen Unternehmen

Fakultät Technik und Informatik
Studiendepartment Informatik

Faculty of Engineering and Computer Science
Department of Computer Science

Lars Mählmann

Deliver who I mean, automatische Erstellung von Personenprofilen in großen Unternehmen

Masterarbeit eingereicht im Rahmen der Masterprüfung
im Studiengang Master Informatik
am Studiendepartment Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. rer. nat. Kai von Luck
Zweitgutachter: Prof. Dr. Birgit Wendholt

Abgegeben am 23. März 2009

Lars Mählmann

Thema der Masterarbeit
Deliver Who I Mean - automatische Erstellung von Personenprofilen in großen Unternehmen
Stichworte
Social Networks , OSGI, Profiles, people search
Kurzzusammenfassung
Das Auffinden der wichtigen und relevanten Informationen ist seit jeher ein schwieriges Thema. Im Intranet werden die Datenmengen größer und die Strukturen komplexer, wodurch das Finden relevanter Daten schwieriger wird. Soziale Netzwerke innerhalb der Firmenstrukturen helfen den Mitarbeitern beim Auffinden von Information im Intranet. Untersucht werden soll, ob soziale Netzwerke mit Hilfe von Personenprofilen technisch im Intranet abgebildet werden können um die Suche nach Informationen zu verbessern. Im Rahmen dieser Arbeit wird eine Architektur vorgestellt und prototypisch implementiert, die Personenprofile automatisch auf Basis von Daten aus dem Intranet generiert und die Suche um eine Personensuche zu erweitern.

Lars Mählmann

Title of the paper
Deliver Who I Mean - Automatically Generated User Profiles for Larger Companies
Keywords
Social Networks, OSGI, Profiles, Search
Abstract
Finding important and relevant information has always been a difficult topic. In the Intranet, in particular, this difficulty increases as the amount of data and their structure become more complex. Social networks [are an example of a mechanism that can help] within a corporate structure to help employees locate relevant information. As part of this framework, I present an architecture and prototypical implementation that, in order to expand the search options, generates user profiles automatically by means of a people search on the intranet.

Danksagung

Viele Personen haben mich bei der Erstellung unterstützt; ihnen allen bin ich zu Dank verpflichtet.
Ein großer Dank geht an Inga und Sam, die viel Verständniss und Ermunterung für mich hatten. Für Korrektur- und Verbesserungsvorschläge danke ich insbesondere Heike, Cezar, Karsten, Raul und Sven. Für die Korrektur meiner unglaublichen Rechtschreibung und allen weiteren Korrekturen danke ich Christian und Christian. Ich möchte mich weiter für die geduldige und tolle Betreuung während des Studiums bei Kai von Luck bedanken.
An dieser Stelle noch vielen vielen Dank für die Unterstützung an meine Eltern.

Inhaltsverzeichnis

Tabellenverzeichnis

Abbildungsverzeichnis

Kapitel 1

Deliver who I mean

In Firmen besteht meistens ein Computernetzwerk, das unabhängig von Ort und Zeit sämtliche Abteilungen miteinander verbindet. Zugänglich ist ein solches Netzwerk für Mitarbeiter über das Intranet. Im Intranet werden Dokumente, Anleitungen, Projekte, Schulungsunterlagen oder Benutzerverzeichnisse zur Verfügung gestellt. Bei größeren Firmen werden die Strukturen eines Intranets schnell großer, unübersichtlicher und das Finden von Informationen schwierig.
Die im Intranet angebotenen Suchfunktionen sind eine Hilfe. Es gibt verschiedene Möglichkeiten Informationen zu suchen. Beispiele hierfür sind die Volltextsuchen, die von verschiedenen Herstellern angeboten werden, z. B., Enterprise FAST, xfriend, Yahoo!. Auch aus der Open Source Gemeinde gibt es verschiedene Systeme zur Suche, Beispiele wären: Beagle, Nutch Projekte, die aus aus Lucene entstanden, Strigi, Swish-e oder WebGlimpse. Die erzielten Ergebnisse sind von der Qualität eher durchschnittlich zu bewerten, weil die Ergebnisse nicht exakt genug sind. Die Ergebnisse müssen von dem Suchenden selektiert werden. Ein Hauptproblem ist es, dass viele Suchmaschinen Dokumente nach deren Relevanz zu dem Gewünschten auflisten. Es wird nicht der Begriff in Zusammenhang mit dem Inhaltes betrachtet. Weiterhin gibt es keine Aussage über die Personen, die Organisation oder den Zusammenhang zu diesen. Die Suche ist aufwendig und kostet viel Zeit. Eine andere Möglichkeit ist es Mitarbeiter direkt zu dieser Thematik zu befragen. Die Schwierigkeit ist das Finden der Personen. In den Unternehmen gibt es Mitarbeiterverzeichnisse die unterschiedlich gut dokumentiert sind. In vielen sind die Mitarbeiter mit ihren Fertigkeiten aufgelistet, so dass eine Suche über diese Verzeichnisse dem Benutzer die entsprechenden Mitarbeiter auflistet. Häufig sind diese Beschreibungen zu Personen zu oberflächlich oder werden nach der Erhebung nicht weiterführend gepflegt. Zu dem geschilderten Problem kommt hinzu, dass ein Unternehmen sich sehr dynamisch entwickelt. Firmen verändern ihre Struktur, Mitarbeiter wechseln die Abteilungen, Abteilungen und Fachbereiche werden umstrukturiert, die Firma eröffnet neue oder schliesst alte Standorte. Die Pflege der Dokumentation ist auf-

wendig und wird häufig vernachlässigt. Dadurch sind wertvolle Information oft nicht verfügbzw.. auffindbar oder indiziert.
Viele der Informationen sind schneller verfügbar, wenn man die richtigen Personen kennt, bzw. fragt. Die Herausforderung besteht darin, die richtige Person zu finden, die unsere Frage beantworten kann oder eine Hilfestellung geben kann. Diese Aufgabenstellung wird komplexer umso grösser die Firma ist und umso mehr Standorte diese besitzt. Es ist schwierig heraus zufinden, welches Wissen in der Firma vorhanden ist und wer über das Wissen verfügt.
Das Finden der richtigen Person basiert meistens auf unserem "Social Network" (Yu und Singh (2002)bzw. Nardi B.A und H. (2002)). Könnte man die normale Suche so erweitern, daß auch Personen gefunden werden, würde sich die Effektivität eines Intranets verbessern. Ein Beispiel dafür ist die Verbreitung von möglichen relevanten Informationen durch "Social Navigation". Es ermöglicht die Profitierung von Erfahrung anderer für den eigenen Zweck. Beispiele hierfür sind Amazon (".. Kunden die diesen Artikel gekauft haben, ...") oder das Austauschen von Bookmarks, wie bei del.icio.us social bookmarking System delicious. Andere Benutzer profitieren von den eigenen sortierten Bookmarks.
Es werden Hinweise für einen Benutzer hinterlassen, was andere in der selben Situation getan haben.
Ein anderes Beispiel ist das Nutzen von Semantic Web Aleman-Meza u. a. (2007) zum Aufbau eines "Social Networks" . Benutzer hinterlegen Informationen über sich in einem maschinen lesbaren Format. Diese Informationen werden analysiert und visuell aufbereitet zur Darstellung der Personen und deren Netzwerke. Beispiel für solche Netzwerke sind LinkedIn, Xing und Foafnaut von Ley. Das Ziel könnte es sein ein System zu entwickeln, das von den verschiedenen Ansätzen Gebrauch macht. Das Intranet soll als eine Plattform genutzt werden, wo man verschiedene Ansätze kombinieren bzw. ergänzen kann. Also Informationen automatisch zusammen trägt, die eine Person, ihr soziales Netz und die Fähigkeiten beschreibt und es in einen Kontext zu dem Gesamtsystem stellt. Daran anschliessend kann man die bekannten Suchfunktionen um diese Funktionalität erweitern und bei einer Suche Personen, die im Zusammenhang mit dem Gesuchten stehen auflisten. Über die Profilbeschreibung kann entschieden werden, ob dieser, der suchenden Person behilflich sein könnte.
Wie weit ist es möglich eine Suche aufzubauen, die Informationen über Mitarbeiter automatisch generiert an Hand der gefunden Informationen im Intranet generiert und in einer inteligenten Suche den Mitarbeitern im Intranet zur Verfügung stellt?
Diese Ausarbeitung beschreibt ein Konzept zur automatischen Erstellung von Profildaten an Hand von offen zugänglichen Informationen im Intranet. Es beschreibt wie man einzelne Konzepte und Technologien in einem Gesamtsystem nutzen kann, um diese Profile zu erstellen.
Der Aufbau der Arbeit ist wie folgt gegliedert: Es wird erklärt werden, was ein Social Network 4.5 ist, wie es für eine Suche genutzt werden kann und welche Probleme bestehen.

Nach Erläuterung der Hintergründe von Social Networks wird am Beispiel eines größeren Versicherungsunternehmens ein konkrete Problemstellung (siehe 1.3) beschrieben. Die grobe Problemstellung beschreibt dabei ein Skill Managment System, was genutzt werden soll um die Fähigkeiten der einzelnen Mitarbeiter zu beschreiben und diese mit Hilfe einer Suchmaschine zur Verfügung zu stellen. An Hand dieser Problemstellung wird eine Analyse (Kapitel 2) vorgenommen. Es wird die Firmenstruktur 2.1 , deren grobe IT-Architektur 2.1.1 und die bisherigen Suchoptionen 2.1.2 beschrieben. Mit Hilfe einer Istzustandsbeschreibung 2.2 werden die Forderung aus organisatorischer, personen bezogener und aus technischer Sicht analysiert und beschrieben. An Hand der vorgenommen Analyse werden verschiedene Arbeiten in Kapitel 3 vorgestellt die ähnliche Forderungen untersucht und unterschiedliche Konzepte zu diesen vorgestellt haben. Unter Berücksichtigung dieser Arbeiten und der Analyse wird ein Konzept entwickelt, welches als Basis für das Design und die Architektur dient (siehe 4). Es wird untersucht welche Daten zur Informationsgewinnung zur Verfügung stehen und wie diese Informationen extrahiert werden können. In einem weiteren Schritt wird untersucht, ob und wie die gewonnen Informationen weiter aufbereitet werden können. Es wird ein Design vorgestellt, wo die einzelnen Bestandteile zusammengefasst werden und in einer Architektur, 4.7, münden. Diese Architektur wird in einer Implementierung auf ihre Umsetzbarkeit geprüft und umgesetzt. Die Ergebnisse der Arbeit werden in Kapitel 4.8 zusammengefasst. Es wird ein Fazit erstellt und ein Ausblick auf weitere Ansätze und Möglichkeiten gegeben.
Um zu verstehen, wie es zu dem Aufbau eines Social Networks kommt oder wie man eine Person in diesem beschreibt, soll zu diesem Bereich eine Beschreibung und Einordnung gegeben werden. Im nächsten Kapitel wird beschrieben, wie diese in der Psychologie beschrieben werden und wie es in der Informatik definiert ist.

1.1 Verhaltensbasierte Profile aus der Psychologie

Eine Schwierigkeit, beschrieben von Ehrlich und Shami (2008), ist das soziale Verhalten einer Person. Mitarbeiter verhalten sich nach organisatorischen und sozialen Kontext unterschiedlich. Ein Mitarbeiter fragt nicht direkt Jemanden, der ihm unbekannt ist und zu dem keine Beziehung über Bekannte besteht, nach Hilfe. Ein Suchsystem, welches Personen als Antwort zu einer Frage liefert, muss zusätzlich auch dessen Umgebung, dessen Stellung und dessen Verantwortungsbereich liefern, damit der Suchende sich an diesen wendet. Gleiches gilt hierbei auch für den Befragten. Dieser bevorzugt es, einem “indirekten” Bekannten zu helfen, als einen Unbekannten. Daraus ergeben sich für den Aufbau eines “Expertise Locator Systems” mehrere Problemstellungen, die Beachtung finden sollten. Es gibt verschiedene Ansätze von Hilfestellungen, um diesen Forderungen gerecht zu werden (siehe Ehrlich u. a. (2007)).
Im Vergleich zum Internet ist die Suche im Intranet nicht so weit entwickelt. Eine Untersuchung von Feldman und Sherman (203) zitiert in Li u. a. (2005) ergab, dass Mitarbeiter 15% -

35% ihrer Arbeitszeit mit Suchen nach Informationen verbringen. Davon beklagen sich 40% der Mitarbeiter, dass sie die gesuchten Informationen im Intranet nicht finden.
Gesellschaftliche Einflüsse, Kultur und Normen sind wesentliche Elemente des Handelns einer Person. Handeln ist immer an bestimmte Gegebenheiten gebunden, d.h. es beinhaltet immer Konformität, damit ein Ziel erreicht werden kann. D.A. Wilder (Wilder (1986)) beschreibt es folgendermaßen:

> Als soziale Kategorisierung bezeichnen wir einen Prozess, bei dem wir unsere soziale Umwelt organisieren, indem wir uns und andere in Gruppen einordnen.

Die Einordnung in soziale Gruppen erfolgt nach dem eigenen Ermessen derjenigen Person, die diese Einordnungen vornimmt. Soziale Gruppen können demnach an Normen oder an eigenständige Kriterien gebunden sein. Soziale Gruppen bilden sich aus den verschiedenen Lebensräumen in denen man sich bewegt, z.B. Arbeit, Hobbies oder Internet. Solche Gruppen werden an das eigene Verhalten angelehnt, dabei wird immer nach Ähnlichkeiten zu sich selbst gesucht. Die Vor- und Nachteile einer Verbindung werden unbewußt geprüft und entschieden. Dabei ist die Ähnlichkeit zu sich selbst, oder dem was man sucht, wesentlich. Gruppenzugehörigkeit sichert heute das soziale Überleben. So kann es passieren, dass einige Menschen sich an Strukturen einer bestimmten Gruppe anpassen und dabei ihre eigenen Vorstellungen aufgeben. Die Solidarität in solchen Gruppierungen oder Kategorisierungen kann sich schnell entwicklen oder auch künstlich erzeugt werden. Für eine Informationssuche sind Gruppen oder soziale Kategorisierungen sehr wichtig. Die gewünschten Informationen können sich nicht nur im direkten Umfeld, der Gruppe, die in die Suche einbezogen worden ist, befinden, sondern ebenso in dessen Umfeld. Eine Gruppe kann sich in diesem Fall aus den direkten Kollegen oder aus dem Intranet zusammensetzen. Die Gemeinsamkeiten der Gruppe, bzw. das Arbeitsumfeld und der Arbeitsbereich liefern die Rahmenbedingungen für eine gelingende Suche, da diese durch gewisse Attribute bestimmt wird. Innerhalb dieser Attribute ergeben sich neue Strukturen in denen sich neue Möglichkeiten der Informationsgewinnung verbergen, z.B. durch weitere Kontakte der Gruppenmitglieder, Kollegen. Auf diese Weise baut sich ein gruppenübergreifendes Netzwerk auf, da jedes Mitglied weiteren Gruppen angehört und so weitere Kontakte bereitstellen kann. Wie zuvor festgestellt, ist dies überlebenswichtig, auch für eine schnelle Informationssuche. Hinzu kommt, dass viele Menschen Gruppen im Internet angehören, sei es beruflich oder privat. Daraus ergibt sich eine weitgestreute Suchmöglichkeit.
Die eingangs genannte Gefahr sich in einer Gruppe zu verlieren, gibt es auch bei der Informationssuche. Zu viele Details erschweren die Suche nach der relevanten Information und können dem Suchenden zu viele Irrwege präsentieren.

> Die einfachste und tiefgreifenste Form der Kategorisierung ist das Urteil selbst darüber, ob andere Menschen so sind wie wir selbst. Philip G. Zimbardo (1996)

Das Internet ist ein vielfältiger Raum, hier schließen sich diverse Menschen in Gruppen zusammen, die es sonst vielleicht nicht getan hätten. Die oben genannten Kategorisierungen ergeben sich im Internet aus Stichpunkten an Hand dessen Informationen gefunden werden können. Diese Kategorisierungen beinhalten Vorteile, da die gezielte Suche eine große Zeitersparnis mit sich bringt. Das ist vor allem von Vorteil, wenn man sich die Arbeit mit Kategorisierungen und Stichwörtern im Intranet größerer Firmen vorstellt. Die Kategoriserung wird nicht von den Nutzern eigenständig erstellt, sondern ist an Berufsfeld oder Arbeitsumfeld geknüpft. Diskrimminierung als Unterscheidungsmerkmal, ist in diesem Zusammenhang sehr wichtig.

1.2 Social Networking

Bei der Suche nach einer Person oder einer Personengruppe wird versucht Gemeinsamkeiten zwischen diesen Personen zu finden. Die Gemeinsamkeiten können verschiedene Relationen sein, wie Freundschaft, Arbeitskollegen, gleiche Interessensgebiete, Visionen, Geschäftsbeziehung, dieselbe Universität, Konferenzen, etc ... (siehe Matsuo u. a. (2006)).
Die Struktur wird in Graphen abgebildet, wobei Personen die Knoten darstellen und Beziehungen im Netzwerk als Kanten bezeichnet und beschrieben werden, Granovetter (Granovetter (1973)). Diese Darstellung wird in den verschiedenen vorgestellten Arbeiten benutzt und weiter ausgebaut.
Solche Beziehungen oder Verbindungen bezeichnet mal als “Social Networking”. Als Begründer des Begriffes “Soziale Netzwerke” wird J.S. Barnes (Barnes (1954)) betrachtet. Er führt in das Netzwerkkonzept soziale, produktive und individuelle Beziehungsfelder ein. Diese Beziehungsfelder sind lockere, indirekte, teilweise unüberschaubare Geflechte sozialer Beziehungen, an denen die Mitglieder sich mehrheitlich und wechselseitig beteiligen. Dabei können Netzwerke in Größe, Dichte, Stabilität und Qualität unterschiedlich sein. Innerhalb der Netzwerke ist der soziale Rückhalt (J.S. (1981)) durch Informationen instrumentiert und durch die faktische Verfügbarkeit von Hilfeleistung (Siegrist (1995)) gegeben.
Wie können diese Begriffe und Darstellung in einem Computersystem abgebildet und nutzbar gemacht werden? Die Untersuchung soll an dem konkreten Problem eines Versicherungsunternehmens untersucht werden. Es folgt die Problemstellung an Hand dessen die Analyse erstellt und ein Konzept entwickelt wird.

1.3 Problemstellung und Zielsetzung

Mit der stetig wachsenden Aufgaben-Komplexität in Versicherungsunternehmen wachsen auch die Anforderungen an die Mitarbeiter. Das führt immer häufiger zu Problemen, die nicht mehr von einem Mitarbeiter allein gelöst werden können. Eine übergreifende Zusammenarbeit von Mitarbeitern unterschiedlicher Bereiche wird unerläßlich. Damit müssen aber Fragen beantwortet werden wie: - Wer versteht die Problematik ? - Wer kennt die zu nutzenden Techniken und - Welche Mitarbeiter können Hilfestellung leisten oder gegebenenfalls mitarbeiten. Einfacher ausgedrückt: Wer kann mir bei meinem Problem direkt helfen? Je größer ein Unternehmen ist, desto schwieriger wird es für seine Mitarbeiter, alle Kollegen und deren Fähigkeiten zu kennen. Die Aufgabe ist also, eine umfassende Bestandsaufnahme des Wissenstandes und der Fertigkeiten der Mitarbeiter vorzunehmen (vergleiche Signal-Iduna (2008)).
Es sollen zwei Punkte verbessert werden:

Wissensaustausch - Die Mitarbeiter sollen sich über ihr soziales Netzwerk hinaus bei Problemstellungen besser austauschen können. Es soll möglich sein, an Hand der Problemstellung gezielt nach Mitarbeitern zu suchen.

- Es soll erfasst werden, welche Fähigkeiten und Fertigkeiten ein Mitarbeiter besitzt. Eine aktuelle, breite transparente Darstellung des Benutzers erfolgen.
- Es soll erkennbar sein, welcher Mitarbeiter zur Lösung konkreter Probleme beitragen kann.
- Es sollen relevante Informationen gefunden werden: wer weiss etwas über die Fragestellung oder das Problem.

Förderung/Weiterbildung - Die Weiterbildung kann an Hand des Wissenstandes des Mitarbeiters effizienter gestaltet werden.

- Wer erfüllt das Anforderungsprofil für ausgeschriebene Stellen?
- Welcher Qualifizierungsbedarf besteht aufgrund zukünftiger Anforderungen?
- Es sollen die Mitarbeiter, die für ihre Aufgaben qualifiziert sind, effizienter genutzt werden. Wenn bekannt ist, welche Mitarbeiter das meiste Wissen oder die meiste Erfahrung besitzen, können Aufgaben besser verteilt werden (z. B. wird die Aufgabenverteilung nicht mehr stur nach veralteten Abteilungsstrukturen vorgenommen, sondern abteilungsübergreifend).

Die Zielsetzung ist in Abb. 1.1 verdeutlicht. Es soll eine Suche für Personen installiert werden. Mit Hilfe dieser Suche soll es möglich sein, nach Personen zu suchen, die etwas zu dem Gesuchten beitragen können. Der zweite Punkt, die Förderung und Weiterbildung, wird in den weiteren Untersuchungen zurück gestellt und nicht genauer betrachtet.

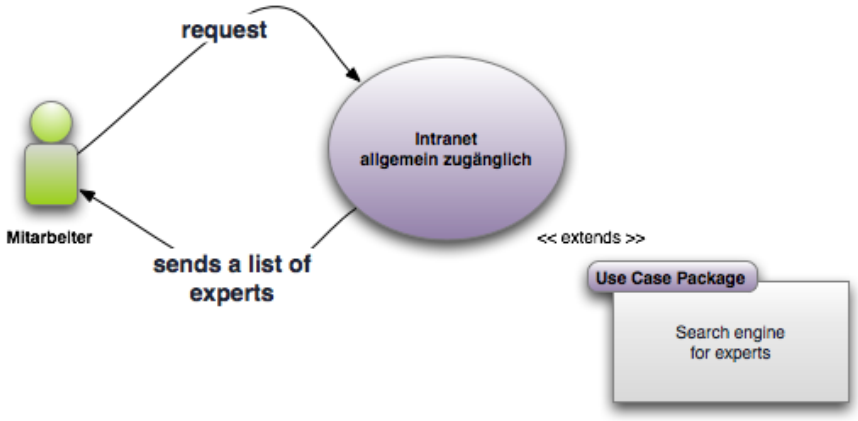

Abbildung 1.1: Darstellung der Benutzersuche aus Sicht des Anwenders

Es wurde die Problemstellung vorgestellt und eine Zielsetzung beschrieben. Im folgenden Kapitel 2, der Analyse, wird die Struktur der Firma und deren IT-Strktur vorgestellt sowie die daraus entstehenden Anforderungen an die hier erläuterte Zielsetzung.

Kapitel 2

Analyse des Versicherungsunternehmens

In diesem Kapitel soll die Zielsetzung einer Personensuche im Intranet detailliert erläutert werden und die Ausgangssituation am Beispiel eines Versicherungsunternehmens beleuchtet werden. Dazu wird die Struktur des Intranets einschließlich der bestehenden Suchoptionen und die Interaktion der Mitarbeiter mit dem System untersucht. Außerdem werden die vorhandene Infrastruktur und der Systemkontext vorgestellt. Daraus werden die Anforderungen fachlicher und technischer Art an die Personensuche abgeleitet, auf deren Basis die Konzeption und Implementierung aufsetzt.

2.1 Aufbauorganisation

Exemplarisch sollen eine Firma und deren Forderungen nach einer verbesserten Suche im Intranet untersucht werden. Diese Ausarbeitung findet im Umfeld eines Versicherungsunternehmens statt. Die Strukturen des Intranets und die getroffenen Annahmen sind auf Basis dieser speziellen Umgebung getroffen worden. Aus diesem Grund soll ein Einblick in die grobe Struktur der Firma gegeben werden. Die Versicherung ist ein international tätiges Unternehmen mit den Hauptsitzen in Dortmund und Hamburg. An diesen beiden Standorten der Hauptverwaltung sind über 2000 Mitarbeiter beschäftigt. Es gibt zusätzlich weitere Zweigstellen in Europa. Die Signal-Iduna besteht aus sieben verschiedenen Ressorts, die aus der Tabelle 2.1 ersichtlich sind.
Die verschiedenen Ressorts sind übergreifend auf die beiden Verwaltungen und kleine Zweigstellen verteilt. Es kommt zusätzlich vor, dass mehrere Abteilungen über die beiden Hauptverwaltungen verteilt sind. Die einzelnen Ressorts arbeiten in verschiedenen Projekten zusammen oder sind miteinander verbunden. Ressort 8 ist für die gesamte Softwareentwicklung zuständig. Sie arbeitet dabei mit den verschiedenen Fachabteilungen aus dem Kerngeschäft (Versicherungen) zusammen, um die fachlichen Anforderungen umsetzen zu

Ressort	Aufgabenbereich
Ressort 1	Lenkung und Koordination der Unternehmungsführung Hauptaufgaben sind die Gebiete Recht, Presse, PR, Personal, Revision und Konzernentwicklung im Ausland
Ressort 2	Vertriebssteuerung, Aussendienstorganisation, Marketing, Aus- und Weiterbildung, Personalentwicklung
Ressort 4	Krankenversicherung, Tarifentwicklung/-überwachung, Underwriting, Vertragswesen und Leistung
Ressort 5	Finanzen und Finanzprodukte, Finanztöchter und Beteiligungen, Immobilien, Darlehen, Unternehmensrechnung und Steuern
Ressort 6	Tarifentwicklung-/überwachung, Underwriting STHUK, Kredit- Kautionsversicherungen; Rückversicherungen, Planung und Controlling
Ressort 7	Lebensversicherung, Tarifentwicklung/-überwachung, Underwriting, Vertragswesen und Leistung
Ressort 8	Betriebsorganisation, Softwareentwicklung, Hardwarebetrieb der EDV, Haustechnik, Service Center, Qualitätsmanagment, Allgemeine Verwaltung, Inkasso und Datenschutz/Datensicherheit

Tabelle 2.1: Eine Auflistung der verschiedenen Aufgabenbereiche im Unternehmen

können. Dabei ist wichtig, die gesetzlichen Anforderungen bei Versicherungsverträgen umzusetzen. Das bedeutet eine starke Zusammenarbeit zwischen den einzelnen Ressorts. Das Problem besteht darin, herauszufinden, wer jeweils die nötigen Informationen besitzt und wie diese Personen an dem selben Standort zusammenarbeiten können.

2.1.1 Systemkontext

Zum besseren Verständniss soll eine Beschreibung der IT-Struktur erfolgen, um einen Gesamteindruck des Systems zu vermitteln. Die Grundzüge der zugehörigen Internet-Architektur sind in Abb. 2.1 dargestellt und werden nachfolgend beschrieben.

Client - Die Clients dienen der visuellen Darstellung zur Bearbeitung der Versicherungsleistungen. Es gibt verschiedene Ansätze um in der IT-Struktur zu arbeiten:

- Die Rich Client Anwendung: Diese Anwendungen basieren auf Java (swing, swt), C++ und Visual Basic.
- Die Thin Client Anwendungen: Diese Anwendungen werden intern von den Sachbearbeitern über das Intranet genutzt. Das Extranet bietet die Anwendungen für die externen Sachbearbeiter (Versicherungskaufleute).

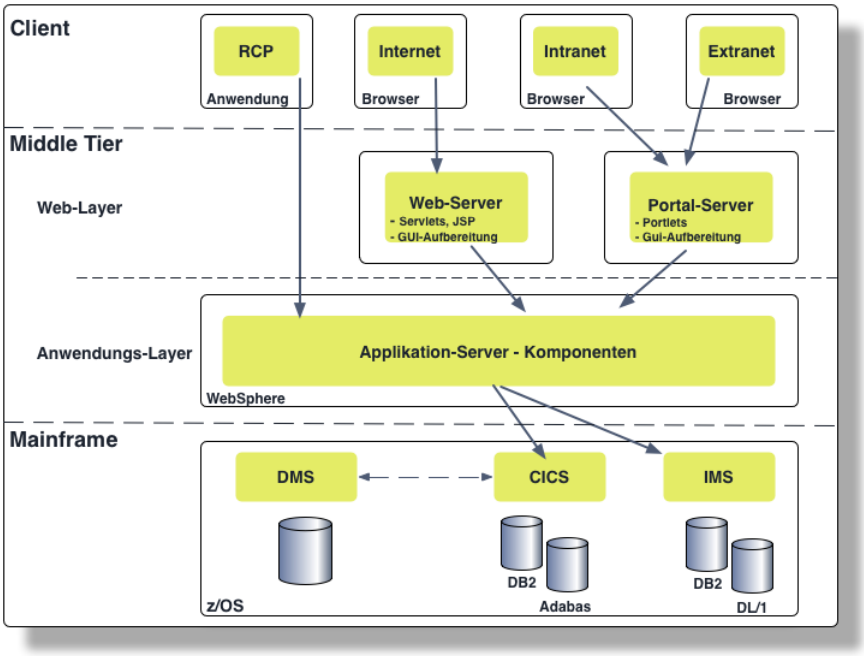

Abbildung 2.1: Darstellung der IT-Struktur

In der Grafik sind die Entwicklerumgebungen, welche für die Entwicklung und Pflege der IT-Struktur über spezielle Fat-Client PCs verfügen, nicht aufgeführt.

Middle Tier - Der Applikations-Server stellt verschieden Komponenten zur Verfügung:

- ... ist eine Integrationsplattform, die die Anbindung verschiedenartigster Systeme ermöglicht: Transaktionsmonitore (IMS, CICS), Datenbanken (DB2, Oracle), Verzeichnisstrukturen (LDAP), E-Mail und CTI-Systeme (Service Center bzw. Call Center), Außendienstsysteme etc.;
- ... ermöglicht Online-Zugriffe auf die operativen Datenbestände, so dass aufwändige Daten-Replikationen entfallen können;
- ... gewährleistet Transaktionsschutz, Sicherheit, Performance und Lastverteilung (Quality of Service), nicht nur für einzelne Komponenten, sondern über das Gesamtsystem (end-to-end);
- ... bietet detaillierte Möglichkeiten der Prozesssteuerung (z. B. Regeln, State Machine);
- ... stellt mit einem Web-Server und (optional) einem Portal-Server Komponenten zur Verfügung. Dies können so genannte Rich-Clients sein, die in Aussehen und Funktionalität den bekannten Windows-Oberflächen in nichts nachstehen,

oder Portale (wie das Intranet), die über Portlets individuelle Anpassungen einer Internet-Seite (Personalisierungen) erlauben.

Mainframe - Die Systeme der Mainframe-Schicht (Vertragsdienste, Leistungsdienste etc.) erlauben über die Transaktionsmonitore IMS und CICS einen gesicherten Zugriff auf die operativen Datenbestände und das Dokumentenmanagementsystem (DAISY). Sie sind in der Regel in Cobol oder PL/1 implementiert.

Die Transaktionsmonitore[1] IMS und CICS haben eine Verfügbarkeit von ca. 99,5 %, skalieren bis zu 4 Mio. Transaktionen pro Tag und haben bei 3270-Anwendungen, Terminal Emulationen, in der Regel performante Antwortzeiten[2].

Die Darstellung 2.1 ist nicht vollständig und stellt die IT-Struktur des Hauptgeschäftes des Unternehmens dar. Einzelne Bestandteile des Unternehmens pflegen Besonderheiten wie in Ressort 8 das Ticketsystem oder das Versioning Werkzeug, welche beide über das Intranet und Internet erreichbar sind.
Die Darstellung zeigt, wie die verschiedenen Strukturen zusammenarbeiten. Es werden die Komponenten dargestellt, die von allen Mitarbeitern genutzt werden bzw. genutzt werden können. Die Besonderheiten sind hier nicht detaillierter beschrieben, da diese speziell den Aufgaben der einzelnen Abteilung geschuldet sind.
Die weitere Analyse konzentriert sich auf das Intranet, da dort die wichtigsten Komponenten für die Mitarbeiter bereitgestellt werden. In der Abb. 2.1 wird das Intranet durch die Client Box “Intranet” dargestellt. Zusätzlich werden die Web-Server, auf denen die verschieden Anwendungen dargestellt werden, gezeigt. Die Anwendungen sind auf dem Applikation-Server hinterlegt und bestehen aus verschiedenen Komponenten. Diese Komponenten werden im folgenden Abschnitt genauer betrachtet und vorgestellt.

2.1.2 Beschreibung des Intranets

Die Struktur des Unternehmens wurde in Kapitel 2.1 beschrieben, skizziert und erläutert, wie die Abteilungen zusammenarbeiten. Im folgendem soll das Intranet des Unternehmens genauer betrachtet werden. Es besteht aus den verschiedensten Bestandteilen (s. o. Abb. 2.1.1). Hier folgt eine Auflistung und Beschreibung der einzelnen Teilbereiche des Intranets:

[1] Transaktionsmonitore werden heute üblicherweise in einer sogenannten Three-Tier-Konfiguration (Tier = Stufen) eingesetzt (Präsentation, Anwendungslogik, Datenhaltung) und decken normalerweise die folgenden Kernfunktionen ab: Message Queuing, Lock-Verwaltung, Logging, Roll-Back, Laststeuerung und Two-Phase Commit-Synchronisation, `http://www.bsi.de/gshb/deutsch/m/m02296.htm`

[2] DMS ist das “Datenbank Management System”

- Der Hauptbestandteil ist das Intranet selbst. Es werden aktuelle Nachrichten, offizielle Dokumentationen zu Werkzeugen und Arbeitsvorschriften, Berichte sowie Projektdokumentationen aus den unterschiedlichen Abteilungen veröffentlicht.
- In die Intranetseiten ist zusätzlich ein Mitarbeiterverzeichniss integriert. Das Verzeichniss enthält die Informationen über den Standort, Telefon, Fax, E-Mail, Abteilung und die Vefügbarkeit des Mitarbeiters zum aktuellen Zeitpunkt (ist der Mitarbeiter in dem System angemeldet).
- Das E-Mail-System basiert auf Lotus Notes. Die gesamte Verwaltung der Mitarbeiter wird über Lotus Notes abgewickelt. Die Informationen über Fortbildung, Urlaub, Krankheit und Arbeitszeitabrechnung wird zusätzlich zu den E-Mails verwaltet.
- Entwicklungwerkzeuge: Wie in Ressort 8 beschrieben wird die Anwendungssoftware im Haus selbst entwickelt. Dafür werden Werkzeuge zur Unterstützung bereitgestellt, die über das Intranet verfügbar sind.
  - Ticketsystem zum Verwalten der Anforderungen und des Entwicklungsworkflows
  - Versionierungswerkzeuge für die Softwareentwicklung
- Das Intranet beinhaltet zu den allgemeinen Dokumentationen zusätzlich Informationen, die von Mitarbeitern in einem Wiki hinterlegt und ohne Beschränkung erweitert, bearbeitet und ergänzt werden.
- Ein weiterer großer Bereich ist die Ablage in strukturierten Verzeichnissen auf Fileservern, die vor allem Protokolle, Projektunterlagen, Präsentationsunterlagen sowie abteilungsinterne Dokumentationen zu Werkzeugen enthalten.

2.1.3 Beschreibung der aktuellen Suchfunktionen

Bisher wurde erläutert, woraus sich das Intranet zusammensetzt. Es wurde beschrieben, welche Komponenten existieren. Weiterhin wurde erklärt, wie die einzelnen Komponenten benutzt werden. Welches sind nun die Möglichkeiten, Informationen aus dem System zu erhalten oder anders beschrieben, warum sind die bisherigen Dienste zur Suche nicht ausreichend für die Mitarbeiter? Die folgenden Punkte sollen beschreiben, wie die Suchfunktionen benutzt werden und welche Bereiche diese nicht abdecken.

- Das Intranet selbst beinhaltet wie beschrieben sämtliche offiziellen Dokumente und aktuelle Nachrichten. Diese Informationen lassen sich mit Hilfe einer Volltextsuche durchsuchen. Die Volltextsuche unterstützt die Operatoren *Und*, *Oder* sowie *Negationen*. Es ist möglich nach einem genauen Wortlaut oder Zitat zu suchen. Der Suchbereich lässt

sich begrenzen, in dem man die Startseite festlegt. Es wird “Top-Down” gesucht. Verschiedene Dokumente werden angezeigt, aber sind zugriffsbeschränkt, d.h. es kann nicht auf alle Dokumente von jedem Mitarbeiter zugegriffen werden.

- Das beschriebene Mitarbeiterverzeichniss lässt sich aus den Intranetseiten einfach durchsuchen nach: Durchwahl, Funktionsstelle, Nachname, Standort und Telefon.

- Die genannten Entwicklungswerkzeuge bieten die jeweiligen anwendungsseitigen Suchmöglichkeiten. Diese beschränken sich auf die jeweilige Applikation. Diese Suchmöglichkeiten sind darüber hinaus nicht erweitert worden und stellen die jeweilige Standardsuche dar.

  Hervorgehoben sei das Ticketsystem, welches eine sehr gute Suchfunktion bietet, mit der nach Anwendern und deren Rolle im System (z. B. Entwickler, Benutzer) gesucht werden kann. Weiterhin gibt es die Möglichkeit den Zeitraum sehr detailliert einzuschränken. Es gibt eine Volltextsuche, ähnlich der oben genannten, die sich mit den anderen Optionen kombinieren lässt.

- Das eingesetzte Wiki basiert auf Mediawiki. Es besitzt eine Suchfunktion, die man mit Hilfe von “Namespaces” einschränken kann. Die Suche ist wie bei der Intranetsuche auf die Suche nach Begriffen ausgelegt und liefert keine zuätzlichen Informationen, die in diesem Bezug stehen.

- Die Informationen auf den Fileservern sind nahezu unmöglich zu durchsuchen, da jede Abteilung die Dokumente in verschiedenen Strukturen vorhält und bei vielen Dokumenten oder Verzeichnissstrukturen der Zugriff verweigert wird. Zur Durchsuchung der Dokumente auf den Fileservern steht die Betriebssystem-Suche zur Verfügung (in dem Unternehmen, Windows XP).

2.1.4 Prozesse zur Personensuche

Im vorherigen Kapitel wurden die vorhandenen Möglichkeiten zur Suche beschrieben. Aber wie nutzt ein Mitarbeiter diese Möglichkeiten? Oftmals sucht ein Mitarbeiter nicht nur über die vorgestellte Technik, sondern nutzt andere Kommunikationskanäle. Die folgende Abb. 2.2 verdeutlicht das Vorgehen der Suche eines Mitarbeiters in der Organisation.
Der Mitarbeiter fragt bei einer Problemstellung als erstes seine Kollegen/Mitarbeiter im direkten Umfeld. Das Umfeld besteht aus den Personen, die am selben Projekt arbeiten, in räumlicher Nähe zu dem Mitarbeiter sitzen oder aus den Mitarbeitern der selben Abteilung. Diese Mitarbeiter haben in der Mehrheit einen ähnlichen Erfahrungsschatz (unterscheiden sich aber durch Spezialwissen). Die Personen sind einander bekannt. In dieser Situation kann es auch zur Kontaktaufnahme mit unbekannten Mitarbeitern kommen, vermittelt durch

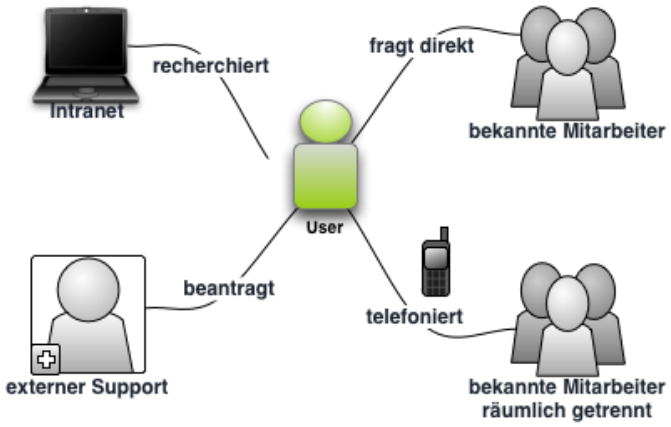

Abbildung 2.2: Die Informationssuche aus Sicht des Mitarbeiters

Kollegen. Das Netzwerk aus bekannten Mitarbeitern kann sich um deren Bekannte erweitern.
Die zweite Möglichkeit ist, bekannte Mitarbeiter in anderen Geschäftsstellen zu befragen. Durch die Projektarbeit und häufige Umstrukturierung von Abteilungen oder Bekanntschaften durch Arbeitsgruppen, Gremien, etc ..., ist das soziale Netzwerk ausserhalb der eigenen Abteilung gewachsen und es gibt eine Vielzahl von Mitarbeitern, die befragt werden können. Das Intranet bietet wie im Abschnitt vorher beschrieben verschiedene Suchmöglichkeiten. Sucht ein Mitarbeiter direkt nach Informationen zu einem Thema, kann der Suchende direkt in den einzelnen Ressorts bzw. Abteilungen, unterteilt nach deren Fachgebieten suchen, s. o. 2.1, oder im Wiki, welches aber nicht komplett von allen Abteilungen benutzt wird. Ein etwas umständlicheres Verfahren ist die Suche in dem Ticketsystem. Dort sind alle bestehenden Projekte mit deren Mitarbeitern aufgeführt. Es lässt sich so ermitteln, welche Mitarbeiter wie und wo mitarbeiten. Zusätzlich lassen sich die Fileserver nach Dokumenten durchsuchen, ob dort gesuchte Informationen und mögliche Ansprechpartner hinterlegt sind.
Die letzte und eher selten angewandte Möglichkeit ist die Anfrage bei externen Beratern. Die Organisation hat für Fremdsoftware Wartungsverträge und Berater, die zu den unterschiedlichen Problemen oder Schwierigkeiten angefordert werden können. Zu einigen Beratern besteht ein sehr enger Kontakt. Durch den direkten Kontakt zu Beratern kann eine direkte Hilfestellung gegeben werden oder bei bestehenden Wartungsverträgen Hilfe beantragt werden (je enger der soziale Kontakt zu den Beratern, desto besser die Kommunikation bzw. Hilfestellung).

Umfrage mit Hilfe der vorgestellten Grafik

Mit Hilfe der Grafik 2.2 wurden 23 Mitarbeiter befragt, in welcher Reihenfolge welche Suche bevorzugt wird. Diese 23 Mitarbeiter stammen aus verschieden Anwendungsentwicklungsabteilungen und sind mit verschiedenen Aufgabenbereichen betraut (Mainframeentwicklung, Projektmanagment, Anwendungsentwickler und Projektleiter). Zusätzlich wurde gefragt, welche Methoden außerdem zu Suche genutzt werden.

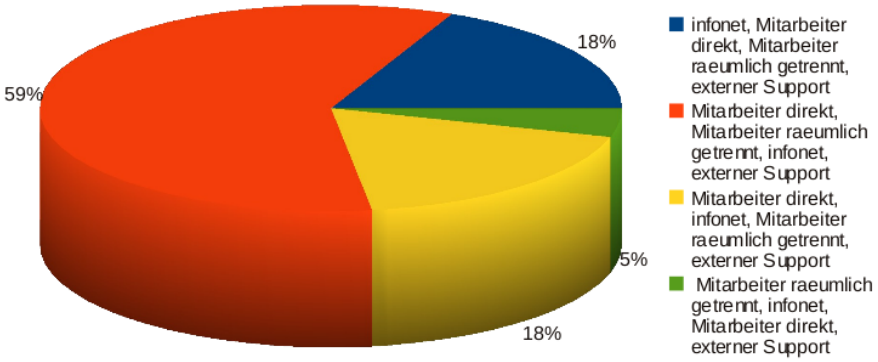

Abbildung 2.3: Darstellung des Suchverhaltens in dem Unternehmen

Es stellte sich heraus, dass 59% der Personen (13) an erster und zweiter Stelle bei Ihnen bekannten Kollegen um Rat fragen und erst an dritter Stelle das Intranet zur Suche benutzen (Abb. 2.3). Bemerkenswert ist, dass 36% der Mitarbeiter (8) im Intranet suchen, nachdem es eine Kommunikation mit anderen Mitarbeitern gab. Das bedeutet, es wird erst im Intranet gesucht, nachdem man erfahren hat, dass dort Informationen zu der Fragestellung vorliegen. Neun befragte Personen gaben an, bei technischen Fragen vorhandene Literatur zu verwenden. Vier Personen nutzen bei nicht zeitkritischen Fragen E-Mail an Mitarbeiter oder externen Support. Eine weitere Beschreibung für das Suchverhalten war die Kategorisierung zwischen technischen und fachlichen Fragen. Als Entwickler ist die eigene Suche bei technischen Problemen einfacher als bei Fragen zu den fachlichen Problemen (Versicherungen, Rechtsfragen, ...). Dadurch ist die Nachfrage bei einem Mitarbeiter die favorisierte Anfrage. Das Intranet wurde von sechs Personen als erste Suchoption genutzt, wenn diese wussten, dass das Gesuchte dort “irgendwo” vorhanden ist. Zwei Mitarbeiter gaben an, externen Support direkt zu erfragen, wenn für problematische Anwendungen ein Supportvertrag existiert.

2.2 Bewertung des “Ist-Zustandes”

An dieser Stelle folgt eine Zusammenfassung und Bewertung des “Ist-Zustandes”. Das Versicherungsunternehmen verfügt über eine Größe, bei der nicht mehr jeder jeden kennt oder

genau weiß, welche Abteilungen für welche Aufgaben zuständig sind oder welche Bereiche es genau gibt. Das Unternehmen ist auf mehrere Standorte verteilt, was zur Folge hat, dass einige Abteilungen ebenfalls aufgeteilt auf mehrere Standorte wurden. Die komplexen Strukturen der Firma machen die Zusammenarbeit und den Informationsausstausch schwer bis auf fast unmöglich.
Das Intranet soll den Mitarbeitern helfen, sich über Neuigkeiten, Anwendungen und Projekte zu informieren. Grundsätzlich werden alle Informationen auch zur Verfügung gestellt, nachdem sie durch ein vier Augen Prinzip auf ihre Korrektheit überprüft wurden. Als Ergebnis liegen statische, formell korrekte “steife” Dokumente vor, die dadurch viel an ihrer Informationsqualität einbüßen. Diese Dokumente sind darüber hinaus für Benutzer des Intranets schwer auffindbar, da sich die Struktur des Intranets an den Aufbau des Unternehmens anlehnt. Mitarbeiter werden daher selten über die Grenzen ihres Bereichs hinaus nach Dokumenten suchen. Die Suche im Intranet soll den Mitarbeiter zwar dabei unterstützen. Sie bietet aber nicht genug Optionen, um nach einem bestimmten Thema zu suchen. Die Suche ist eine reine Volltextsuche, die nur genau nach dem eingegebenen Begriff sucht. Die Suchergebnisse liefern keinen Zusammenhang zu Personen oder Abteilungen, die im Kontext zu den gefundenen Dokumenten stehen oder diese verfasst haben.
Eine Verbesserung wird durch das Wiki erreicht. Es bietet Vorteile, wie eine dynamisch, einfach zu benutzende Dokumentationsplattform. Es beinhaltet “Best Practice” Hilfestellungen, Infrastrukturbeschreibungen und Dokumentationen zu Anwendungen, die eingesetzt werden oder selbst entwickelt wurdenHasan und Pfaff (2006). Das Wiki bietet den Mitarbeitern eine einfache Art der Kommunikation und des Wissensaustausches. Das Problem bei dem Versicherungsunternehmen ist die noch nicht bestehende Aktzeptanz und die fehlende Verbreitung, die nötig ist, um einen erfolgreichen Wissensaustausch zu gewährleisten. Eine Schwierigkeit ähnlich wie bei Foren ist, dass die Mitarbeiter mit dem “Expertenwissen” nicht die nötige Zeit finden ihr Wissen niederzuschreiben. Die Bedienung ist für viele Sachbearbeiter ein weiteres Hindernis.
Das Ticketsystem ist ein System zur Fehlerverfolgung oder Dokumentation für zusätzliche Anforderungen. Es bietet keinerlei Informationen über Dokumentation, Expertenwissen oder Mitarbeiter. Die Projektmitarbeiter haben nur Zugriff auf Ihr eigenes Projekt. Es ist nicht möglich neue Informationen aus dem System über “Experten” zu extrahieren. Alle Personen in einem Projekt sind bekannt und bieten keine neuen Informationen an. Versionierungswerkzeuge dienen der Sicherung des Sourcecodes. Es bietet Informationen über den Stand des Sourcecodes und des Projektstandes, wie in Valetto u. a. (2007) beschrieben. Weiterhin haben nur Entwickler auf diese Informationen Zugriff.
Die Fileserver sind eine der wichtigsten Kommunikationsplattformen für die Mitarbeiter. Die Mehrzahl der Mitarbeiter arbeitet auf Thin Clients, somit werden die erstellten Dokumente alle auf den File Servern gespeichert. Jede Abteilung besitzt einen gemeinsamen Ordner, der nach den eigenen Vorstellungen der Abteilung gepflegt wird. Vielfach sind die Strukturen der Abteilungen völlig unterschiedlich, oftmals sogar selbst innerhalb der Abteilung.

Die dort gesammelten Informationen sind sehr detailliert, aktuell und häufig sehr reichhaltig. Sucht ein Mitarbeiter nach Informationen, steht er vor einer Vielzahl von Problemen. Er kennt weder alle Abteilungskürzel, wonach die Abteilungen gekennzeichnet sind, noch deren Bedeutung. In einem zweiten Schritt ist die Hierachie in den Strukturen der jeweiligen Abteilung nicht nachvollziehbar und die Dateinamen der Dokumente sind nicht verständlich, weil meistens Abkürzungen diese beschreiben. Der Schreibschutz, der auf den meisten Ordnern liegt, verhindert eine intensive Suche nach benötigen Informationen oder Unterstützung bei der Suche nach Mitarbeitern, die helfen könnten. Eine Suche mit der betriebssyteminternen Suche liefert aus den geschilderten Gründen nur bedingt Ergebnisse.
Ein Einwand könnte sein, dass ersichtlich ist, wer einen Artikel verfasst, ein Ticket eröffnet, Texte im Infonet hinterlegt oder Dokumente auf Fileservern hinterlegt hat. Diese Unterlagen müssen jedoch gefunden werden, um zu wissen, wer der Ansprechpartner sein kann. Die Forderung ist, dass der Suchende diese Information geliefert bekommt und nicht über Zwischenschritte suchen muss.
Zusammenfassend lassen die bisherigen technischen Möglichkeiten des Intranets keine Suche oder Abfrage nach dem Wissen und den Fähigkeiten einzelner Mitarbeiter, wie in der Problemstellung gefordert, zu. Jeder Mitarbeiter ist auf sein soziales Netzwerk angewiesen, das je nach Mitarbeiter mehr oder weniger weit über Unternehmensbereiche ausgeweitet ist. Die Folge ist, dass das Wissen innerhalb bestimmter Grenzen bleibt, z.B. Dokumente oder Links zu Informationen per E-Mail oder direkten Kontakt ausgetauscht werden.
Die bestehenden Funktionen im Intranet mssen daher erweitert werden, um allen Mitarbeitern die Informationen bereitzustellen, welcher Mitarbeiter über welches Wissen verfügt. Basierend auf dem bestehenden System werden in den nachfolgenden Abschnitten die Anforderungen an eine solche Erweiterung beschrieben.

2.3 Anforderungen an eine Personensuche

In den folgenden Beschreibungen und Szenarien wird sich auf die Ressort 8 beschränkt und die diversen Aufgaben in diesem Bereich. Das Ressort 8, ist wie oben aufgeführt, für alle Art von Softwareentwicklung und Netzwerkstruktur verantwortlich. Das bedeutet, dass es immer in irgendeiner Beziehung zu anderen Fachbereichen steht. Die Anforderungen unterteilen sich in organisatorische, personenbezogene und fachliche Anforderungen. Begonnen wird mit der organisatorischen Anforderungen.

2.3.1 Anforderungen aus organisatorischer Sicht

> Das immaterielle Vermögen des Unternehmens darf nicht brach liegen, sondern soll besser genutzt werden.
> Signal-Iduna (2008), Reinhold Schulte

Das Wissen der Mitarbeiter soll besser und effektiver genutzt werden. Es soll heraus gefunden werden, wo die Stärken und Schwächen der Mitarbeiter liegen. Die Stärken sollen gefördert werden und die Schwächen durch Weiterbildung behoben werden.

- Es soll die Rollenverteilung bei den Informationen beachtet werden. Es gibt Mitarbeiter, Abteilungsleiter oder Bereichsleiter. Diese verschiedenen Rollen haben unterschiedliche Sichten auf die verschiedenen Informationen. Es gibt Informationen, auf die ein Mitarbeiter keinen Zugang erhält. Bei der Suche nach Informationen, ist ein unterschiedlicher Abstraktionsgrad gefordert. Ein Abteilungsleiter bzw. ein Projektleiter muss und will nicht immer alle Details eines Themas, sondern nur relevante Informationen erhalten. Daher soll zwischen den Personen die eine Anfrage stellen unterschieden werden. Es gibt weiterhin eine Gruppe von externen Mitarbeitern, die im System enthalten sind, aber ihr Wissen, nach dem Ende eines Projektes oder deren Mitarbeit im Unternehmen, mitnehmen. Es soll vermerkt werden, wer diese Personen sind und in welchen Bereichen es relevant ist.

- Eine grosse Schwierigkeit ist es, den Datenschutz zu gewährleisten. Ziel ist es, die Fertigkeiten und Fähigkeiten eines Mitarbeiters zu erfassen, aber die Privatsphäre des Mitarbeiters muss gewährleistet werden. Die Ermittlung der Informationen muss transparent sein. Für Mitarbeiter muss die Einsicht in ihre Daten zu jeder Zeit gewährleistet sein. Es sollten die entsprechenden Datenschutzgesetze eingehalten werden. Die Einhaltung der geschilderten Punkte sollte durch den Datenschutzbeauftragten der Organisation kontolliert werden. Weitherhin sollte beachtet werden, welche Fertigkeiten und Fähigkeiten von dem Benutzer dokumentiert werden. Es ist möglich das ein Mitarbeiter bestimmte Qualifizierungen besitzt, diese aber nicht zur Verfügung stellen möchte (siehe Ackerman u. a. (1999), dort wurde untersucht, welche Aspekte bei der Suche nach personenbezogenen Daten wichtig sind). Mitarbeiter dürfen nicht dazu missbraucht werden, andere Mitarbeiter zu beurteilen und an Hand der Qualifizierung zu bestimmen, wie wichtig ein Mitarbeiter ist. Es gilt zu beachten, dass diese Beschreibung einer Person nur eine Auflistung von Qualifikationen ist und nicht dessen Persönlichleit und dessen Arbeit bewerten kann.

- Die Informationen eines Mitarbeiters sollen im Kontex von Zeiträumen betrachtet werden. Ein Mitarbeiter, der sich vor mehreren Jahren mit bestimmten Themen beschäftigt hat, gilt hier nicht als Garant für eine gute Hilfestellung. Gleiches gilt für jemanden, der sich erst seit kurzer Zeit in Themen einarbeitet. Ein Faktor zur Bewertung sollte aber die Erfahrung sein, wie lange ein Mitarbeiter in seiner Funktion arbeitet. Diese Informationen sollen dazu dienen, festzustellen, wann und in welcher Form eine Weiterbildung stattfinden soll.

- Das Verfahren soll Zeit und Aufwand einsparen. Mitarbeiter sollen effektiver miteinander arbeiten. Durch die Bereitstellung und Verwaltung der Informationen soll kein

Mehraufwand entstehen. Die Informationsgewinnung soll automatisiert stattfinden. Die Bereitstellung muss im Intranet erfolgen und immer verfügbar sein.

2.3.2 Benutzeranforderungen

Die Ressort 8 ist zuständig für Innovation und Entwicklung neuer Softwarearchitekturen. Aus dieser Sicht gibt es für einen Anwender verschiedene Anforderungen nach bestimmten Interessenschwerpunkten.
Die Anforderungen aus Nutzersicht sollen beschreiben, was ein Anwender von dem System erwartet und wie dieser damit umgehen kann. In einem ersten Schritt soll der typische Mitarbeiter vorgestellt werden. Um die Anforderungen eines Benutzers besser zu verstehen, soll geklärt werden, wonach Anwender suchen. Es wurden mehrere Untersuchungen veröffentlicht, die sich damit beschäftigt haben, was ein Anwender sucht und wie er sucht.
Dabei wurde in der Arbeit von Yiman-Seid und Kobsa (2003) herausgefunden, dass Personen eine Suche oftmals benutzen um Personen zu finden, wenn das eigene soziale Netzwerk nicht ausreicht. Ein Anwender benutzt die Suche zum Finden von Experten, als Quelle für fachspezifische Informationen oder aber um Kontakte oder Personen als Verantwortliche für eine bestimmte Funktion oder einen Posten zu finden.
Diese beiden Motive für eine Suche lassen sich nach Ehrlich und Shami (2008) noch weiter unterteilen in vier Punkte:

1. **Answer**: Die Suche nach einer direkten Antwort auf eine Frage, was unwichtig ist und wer diese Frage beantwortet.

2. **Person**: Die Suche nach einem Experten, der über spezielles Wissen verfügt und mit dem man seine Fragestellung diskutieren kann.

3. **Awareness**: Das Wissen über ein Thema oder über Problemstellungen im Allgemeinen, wo evaluiert werden kann, ob es interessant sein oder zukünftig eine Rolle spielen kann in dem eigenen Bereich.

4. **Providing information**: Die Suche nach einer Person oder Gruppe, die Interesse am Austausch von Informationen hat oder Interesse an dem Thema. Finde Jemanden mit dem man sein Wissen austauschen kann.

Diese verschiedenen Punkte sollen bei der Entwicklung des Konzeptes mit berücksichtigt werden. Weiterhin gibt es spezielle Erwartungen, die sich direkt aus der Arbeit in der Versicherung ergeben zugeschnitten. Die folgenden Fragestellungen sind sehr detailliert auf die Erwartungen und die Verbesserung der Kommunikation und der Zusammenarbeit der Mitarbeiter bei dem Versicherungsunternehmen.

- Durch die Globalität des Unternehmens ist das Wissen um Mehrsprachigkeit bei Mitarbeitern für die Entwicklung von grossem Wert.

- Es werden Vorgehenskonzepte und Standards für die computergestützte Arbeit entwickelt. Diese Arbeit erfordert eine enge Zusammenarbeit mit den Mitarbeitern innerhalb der Ressorts und zusätzlich mit den Anforderung und Neuerungen aus den Fachabteilung vertraut zu sein. Es stellt sich die Frage, wer kennt was und wer kann bei bestimmten Problemen eine Hilfestellung geben oder aber bei der Entwicklung helfen.

- Durch die Schnelllebigkeit in der Informatik wechseln die Techniken sehr schnell und es entsteht ein hoher Aufwand bei der Einarbeitung. Es besteht die Forderung in der Firma jemanden zu finden der einem eine Einleitung geben kann oder bei speziellen Problemen helfen kann.

- Die Suche soll einfach zu bedienen sein, sich ohne Beschränkung nutzen lassen und "öffentlich" zugänglich sein (z. B. ähnlich wie die jetzige Suche im Intranet).

2.3.3 Systemanforderungen

Zuvor wurden die direkten Wünsche der Mitarbeiter nach einer Verbesserung des Netzwerkes beschrieben. Wie kann eine Personenbeschreibung erfolgen?
Welche Anforderungen und Schwierigkeiten entstehen bei der Konzeption und der Umsetzung? Die Gewinnung von Informationen und die richtige Zuordnung ist die Hauptaufgabe bei der Erstellung einer Beschreibung. Die Auflistung beschreibt Fragestellungen, die bei der Beschreibung von Personen beachtet werden müssen.

- Zum Erstellen von Personenbeschreibungen benötigt man Informationen. Woher erhält man diese Daten und wieviele Daten werden benötigt, um eine vollständige Personenbeschreibung zu erstellen.

- Wenn geklärt ist, woher die Daten oder Informationen kommen, ist nicht erklärt, wie man Zugriff auf diese Daten erhält und in welcher Form die Daten vorliegen. Liegen die Daten in einem binären Format vor oder werden diese in einer Datenbank vorgehalten? Kann man über eine Query Sprache wie SQL auf die Daten zugreifen oder müssen spezielle Algorithmen entwickelt werden? Kann man vielleicht trotz des Wissens um die Daten nicht auf diese zugreifen?

- Wie erhält man aussagekräftige Informationen aus der Menge an Daten? Was sind aussagekräftige Daten? Müssen die Daten aufbereitet werden, wie z. B. beim Data Mining oder liegen diese Informationen direkt bereit?

- Welche Informationen sind relevant und verfügbar? Wie kann entschieden werden, welche Daten Informationen erhalten, die eine Beschreibung für eine Person liefern. Die Informationen müssen weiterverarbeitbar sein, dass heisst ein Algorithmus sollte

auf die Daten zugreifen und sie bewerten können. Die Daten sollten nicht dem Datenschutz oder Sicherheitsrichtlinien unterliegen (z. B. Gehaltsinformationen, etc ...).

- Wie nach der Gewinnung der Information und der Erstellung von Personenbeschreibungen kann überprüft werden, dass die Informationen korrekt sind.
- Die Informationen sollen automatisiert erstellt und immer aktuell gehalten werden. Die Daten sollen in regelmässigen Abständen wieder überprüft und dahingehend neu bewertet werden, ob die bisherigen Informationen über eine Person noch korrekt sind oder ergänzt bzw. neu erstellt werden müssen.

2.4 Zusammenfassung

Die Forderung aus der Problemstellung in Kapitel 1.3 war es, eine Suche nach Personen zu entwickeln, als Integration in das bestehende Intranet. Es soll aus dem Intranet für jeden Benutzer anwendbar sein und in die bestehende Suche eingebunden werden können.

Abbildung 2.4: Abbildung einer möglichen Suchmaske

Der Lösungsansatz könnte wie folgt aussehen: Der Mitarbeiter stellt über die bekannte Suchmaschine im Intranet eine Anfrage (z. B. über eine Technologie). Die Suchmaschine nutzt weiterhin die bekannten Algorithmen zur Suche nach passenden Informationen (verschiedene Typen von Dokumenten). Zusätzlich zu denen wird die Suche um die Suche nach Personen erweitert. Zu den gesuchten Begriffen wird zusätzlich ein Liste von Personen als Ergebnis geliefert, die in einer Beziehung zu dem Gesuchten stehen. Diese Anwendung könnte ungefähr wie in Abb. 2.4 aussehen.
An Hand der Forderungen wird ein Benutzerprofil über jeden Mitarbeiter angelegt; das Profil beschreibt einen einzelnen Mitarbeiter unter Berücksichtigung des Datenschutzes. In der Be-

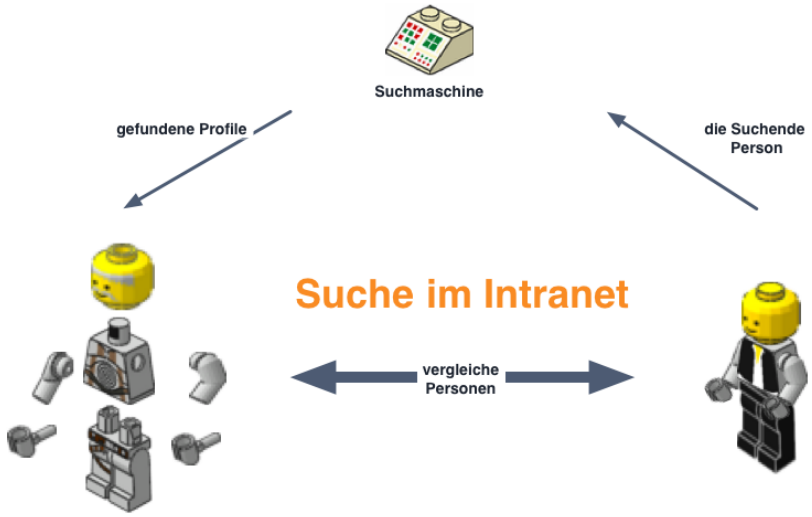

Abbildung 2.5: Die Anforderung für ein Profil

schreibung können Informationen festgehalten werden, z. B. über die Firmenstruktur, in der sich der Mitarbeiter befindet, sowie dem sozialen bzw. technischen Kontex. Die Informationen werden automatisiert erstellt und immer aktuell gehalten. Das Ergebnis soll, wie in Abb. 2.5 dargestellt eine Liste von Personen anzeigen, die in einer Beziehung zu dem Gesuchten stehen.
Es soll möglichst die Auswahl der Personen an Hand des Suchenden abgeglichen werden. Wie tiefgehend ist die Fragestellung des gesuchten und wie groß ist das Wissen des Suchenden? Es soll von den Mitarbeiter ein Profil erstellt werden, welches von einer Suchmaschine genutzt werden kann, und was die Person und deren Fähigkeiten beschreibt (Abbildung 2.5).

Kapitel 3

Vergleichbare Ansätze

In diesem Kapitel werden mehrere Arbeiten vorgestellt, die den Aufbau eines “Social Network” zeigen oder Untersuchungen in dem Kontext des “Social Networks” vorstellen.
Der Begriff “Social Networking” und die dadurch erfolgenden Hilfestellungen, wie bessere Hilfestellungen oder Verbesserung von Teamarbeit, wurden in Arbeiten von Borgatti und Cross (2003), Nardi B.A und H. (2002) beschrieben. Dabei wird gezeigt wie gross die Bedeutung von Sozialen Netzwerken ist. Dargestellt wird außerdem, das der Anwender auf der Suche nach Informationen häufig auf das persönliche Netzwerk zurück greift, um schnelle Hilfe zu erlangen und dabei Vertrauen zu den erlangten Informationen hat. Die Personensuche wurde mehrfach untersucht und dabei wurden verschiedene Facetten festgestellt. Zum Beispiel, ob man direkt nach einer bestimmten Person oder ob man nach einer beliebigen Person, die einem helfen kann suchtYiman-Seid und Kobsa (2003). Diese Beschreibung wurde in Ehrlich und Shami (2008) weiter unterteilt in: eine einfache Antwort auf eine direkte Fragestellung, die Suche nach einer beliebigen Person mit einer speziellen Qualifikation, die Suche nach vorhandenem Fachwissen ohne weitere Unterteilung nach Person oder Gruppe und Personen, die Suche nach Personen, die sich für ihr Fachgebiet interessieren oder an ähnlichen Dingen arbeiten beispielsweise: die einzige Person in der Abteilung die sich mit “Social Networkig” beschäftigt oder gibt es in der Firma noch weitere Personen die daran arbeiten?). In Li u. a. (2005) wird vorgestellt, dass man die Daten für die Suche in unterschiedliche Kategorien (“What is”, “Who is”, “Who knows about” und “Where is homepage”) aufteilen kann.

3.1 Searching for Experts in the Enterprise: Combining Text and Social Network Analysis

Die Ausführung von Ehrlich u. a. (2007) stellt eine “social-context-aware” Suchmaschine (SmallBlue) vor. Diese sucht nach Experten für eine Problemstellung oder ein Thema. Zusätzlich analysiert sie das soziale Netzwerk der gefunden Personen und versucht die soziale

Verbindung zu diesen Personen zu finden. Das Ziel von SmallBlue ist es Experten, “Communities” und soziale Netzwerke in großen Organisationen zu analysieren und verfügbar zu machen. Es werden verschiedene Techniken wie Data Mining, Information Retrieval und Social Network Analysis benutzt. Die Suchmaschine funktioniert wie eine “normale Suchmaschine” vergleichbar etwa zu Google. Es werden Suchbegriffe eingegeben und als Ergebnis eine Liste von N möglichen Experten zurück geliefert. SmallBlue unterteilt sich in vier Werkzeuge:

SmallBlue Ego: zeigt das eigene soziale Netzwerk (nicht in dieser Ausarbeitung beschrieben)

SmallBlue Find: zeigt eine Liste von Personen, die in Verbindung mit einem Schlüsselbegriffe stehen (mehrere Millionen Suchbegriffe umfassend)

SmallBlue Reach: schätzt mit Hilfe von Begriffen ab wie sehr eine Person als Experte für dieses Thema gelten kann. Es zeigt die Aufgabenbeschreibung der Person, einen möglichen Blog, Einträge in Foren, Gruppenzugehörigkeit und deren Projekt- bzw. Abteilungszugehörigkeit an.

SmallBlue Net: zeigt das Soziale Netzwerk von Personen oder “Communities” innerhalb einer Firma in Bezug auf einen gesuchten Begriff.

Die Kombination aus SmallBlue Find und Net bewertet und zeigt welche Experten einem am nächsten sind. Bei einer unbekannten Person wird zusätzlich angezeigt in welcher Verbindung man zu einer Person steht, um zu vermeiden eine völlig unbekannte Person zu kontaktieren.
Die Daten aus E-Mails und Chat Protokollen werden als Quelle der Informationen genutzt. Die lokalen Daten werden indiziert und ausgewertet. Die Entscheidung E-Mail und Chat als Informationsquelle zu benutzen hat mehrere Gründe:

a) vollständige Abdeckung, jeder benutzt E-Mail und chat.

b) Möglichkeit zur Wartung, es kommen immer neue Informationen in das System.

c) Benutzbarkeit, jeder weiss wie es benutzt wird.

Die bestehende Problematik des Datenschutzes wurde berücksichtigt indem die Nutzung der Anwendung freigestellt wurde. Es jeder Zeit möglich ist die Anwendung zu löschen und alle Daten zu entfernen. Das System wurde transparent gestaltet und dem Benutzer erläutert wie das System arbeitet. Mit wachsendem Datenbestand konnten die Regeln optimiert werden, die festlegen, welche Daten gesammelt, ausgewertet und dem Anwender zur Verfügung gestellt werden(siehe Lin u. a. (2007)).

Bei der Bewertung von SmallBlue stellte sich heraus, dass eine Auflistung mit dem Ziel Experten zu einem Thema zu finden eher Personen aufführt, die selbst keine Experten sind, jedoch Experten zu einer Fragestellung kennen. Die Anwender bevorzugten in SmallBlue Net die Personen aus ihrem eigenen Netzwerk. Es konnte mit SmallBlue das eigene Netzwerk um Gruppen mit Personen mit den selben Interessen erweitert werden. Eine Schwierigkeit ist die Ausbreitung von SmallBlue auf andere Abteilungen und geographisch getrennte Gruppen. Es muss versucht werden eine möglichst breite Abdeckung von SmallBlue zu erreichen, um das Soziale Netzwerk und deren Experten möglichst vollständig darzustellen. Es gab die Erkenntnis, dass es unklar ist wann Personen Kontakt zu unbekannten Personen aufnehmen und wonach in welcher Form gesucht wird.

3.2 Modeling and Predicting Personal Information Dissemination Behavior

In Song u. a. (2005) wird analysiert wie Information durch E-Mail und soziale Netzwerke verbreitet werden. Es wird ein personalisiertes Profil, CommunityNet, vorgestellt. Es wird durch Kontakte, Inhalt der E-Mails und Datumsinformationen aufgebaut. Es wird versucht folgende Fragestellungen zu beantworten:

1. Wer ist durch die Thematik einer E-Mail miteinander verbunden?
2. Wer versteht etwas von einem bestimmten Thema und ist möglicherweise involviert?
3. Wie werden Informationen verteilt bzw. weitergeleitet?
4. Wer wird benachrichtigt wenn eine Nachricht veröffentlich wird?

Das Vorgehen zur Informationsgewinnung und Aufbereitung wird wie in Abb. 3.1 realisiert.

1. Analyse des Sozialen Netzwerkes eines Anwenders an Hand der E-Mail.
2. Aufbau von CommunityNet mit Hilfe der Analyse aus Schritt 1.
3. Analyse des Benutzerverhaltens mit CommunityNet.
4. Anzeigen von Vorschlägen für das eigene persönliche Netzwerk und den Aufbau des Organisationsnetzwerkes.

Bei der Umsetzung wird CommunityNet hauptsächlich mit zwei Algorithmen vorgestellt, Personal Social Network und Content-Time-Relation. Diese basieren auf Cluster Analysis und Information-Retrieval. Durch die Kombination der beiden und einem zusätzlichen angepassten Latent Dirichlet Algorithmus, bei dem Begriffe aus einem Dokument (hier E-Mail Inhalt) einem Thema zugewiesen werden, wurden die oben genannten Punkte umgesetzt.

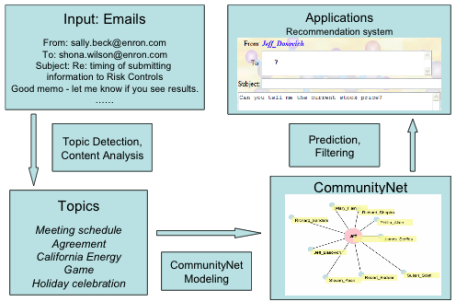

Abbildung 3.1: Schematische Ablauf der Informationsgewinnung und Aufbereitung der Daten

3.3 A New Approach to Intranet Search Based on Information Extraction

Es wird ein anderer Ansatz einer Intranetsuche vorgestellt und als “Information Desk” implementiert. Unter Intranetsuche versteht man die Suche in einer Firma bzw. Organisation. Es wird in Li u. a. (2005) angenommen, dass man die Suche nach Informationen in Kategorien unterteilen kann. Diese Studie wurde am Beispiel des Intranets von Microsoft vorgenommen. Es wurde die Historie der Intranetsuche an Hand folgender Punkte untersucht

- die Suchanfrage.
- die Dokumente, die als Antwort zurück geliefert wurden.
- die Dokumente, die vom Benutzer aufgerufen worden sind.

Das Ergebniss unterteilt sich in drei Bereiche: Information (was, wie, warum), Navigation (Suche nach bestimmten Personen, Internetseiten oder Organisationen) und eine Kombination aus Beiden. Mit Hilfe dieser Informationen wurden die Daten im Intranet analysiert. Es wurden die Metadaten der verschiedenen Dateien extrahiert und katalogisiert. Mit Hilfe von SVM (Support Vector Maschine) wurden die Dateien nach ihrem Inhalt und der Bedeutung bewertet. Diese Information wurden in einer Datenbank abgelegt. Es konnte der Inhalt der Dokumente ermittelt werden und es konnte an Hand der Metadaten und einer “Keyword extraction” festgestellt werden, wer ein Dokument erstellt hat und wer im Bezug zu diesem Dokument steht.

3.4 Polyphonet, An Advanced Social Network Extraction System from the Web

Polyphonet 3.2 ist ein Projekt aus dem ein Social Network aufgebaut werden kann. Das System findet Beziehungen zwischen Personen oder verschiedenen Gruppen und ordnet

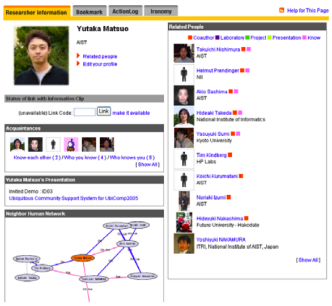

Abbildung 3.2: Screenshot von einer Person aus Polyphonet

Schlagwörter einer Person zu. Das Ergebnis des Projektes wurde als eine “Super Social Network Mining” Architektur vorgestellt und auf mehreren Konferenzen erfolgreich getestet (Matsuo u. a. (2006)). Zunächst wurden verschiedene Algorithmen entwickelt, um Informationen aus dem jeweiligen System zu extrahieren. Im zweiten Schritt wurden mehrere Data-Mining Algorithmen verwendet, die eine Einteilung von verschiedenen Relationen in Kategorien vornimmt. Durch die Skalierung der Daten verbesserte man die Möglichkeiten Beziehungen zwischen Personen und Begriffen zu erstellen. Es zeigt die Beziehung zwischen Personen in einzelnen Fachgebieten.
In Polyphonet werden verschiedene Algorithmen zum Extrahieren von Daten beschrieben, die Algorithmen basieren auf Google Hacks Calishain und Dornfest (2003) und wurden auf die Umgebung angepasst. Es wird auf die Skalierung der Daten eingegangen und erklärt wie man mit der Mnge von verschiedenen Beziehungen zwischen Personen und Dokumenten arbeitet und diese verarbeitet. Es werden weitere Algorithmen vorgestellt, mit denen die Metadaten von Personen an Hand von Personen - Wort - Matrizen besser sortiert werden können.

3.5 Augmenting employee profiles with people-tagging

Ziel der Arbeit war die Erweiterung von bestehenden Intranetprofilen 3.3. Es wurde versucht automatisch Inhalte aus anderen Systemen, z. B. “Web 2.0” Technologie (Blogs oder Social Bookmarks) in die Profile zu integrieren. Benutzer wurden befähigt an Hand von Schlüsselwörtern andere Benutzer zu “taggen” bzw. zu klassifizieren. Durch das “Taggen” mit “Social Bootmarks” wird ein Standartbenutzerprofil erweitert Farrell u. a. (2007).
Es wurde untersucht ,ob und wenn ja wie Benutzer diese Anwendung verwenden und ob es die Qualität der bestehenden Profile verbessert. Außderm wurde untersucht wie man diese Technologie des “taggens” auf mehrere Anwendungen erweitern kann.
Mit Hilfe von verschiedenen Algorithmen (Data mining, text processing) wurde ermittelt, womit eine Person “getaggt” wurde. Dabei wurde versucht ähnliche Begriffe zu einem zusam-

Abbildung 3.3: Abbildung eines Benutzerprofiles mit der Erweiterung von Fringe

menzufassen und gezählt wie häufig eine Person mit verschiedenen Thematiken in Zusammenhang gebracht wurde. Durch Algorithmen, ähnlich dem Google' PageRank, [3] wurde ermittelt welche Person zu den jeweiligen Begriffen die "beste Bewertung" hat.

3.6 Towards Effective Browsing of Large Scale Social Annotations

Es gibt viele "Services", die den unterschiedlichen Benutzern ermöglichen ihre favorisierten Daten mit anderen Interessierten zu teilen und zu verwalten. Durch die steigende Vielzahl der sozialen Annotationen ergibt sich das Problem der effizienten Suchgestaltung. In kleineren Suchgebieten helfen z. B. TagClouds bei der Verwaltung, bei steigender Größe ist dies jedoch kompliziert. Eine Lösung für dieses Problem der Anwender und Serviceanbieter stellt ELSABer Li u. a. (2007). ELSABer kann größere Datenmengen effektiv, hierarchisch und semantisch durchsuchen und Verbindungen herstellen.
Dies erfolgt durch

- Erstellen semantischer Beziehung von ähnlichen Begriffen (z. B. Film, Movie) und dazugehörigen Begriffen
- Erstellung von hierarchischen Beziehungen in einer geordneten Abfolge (z. B. Säugetiere, Katzen, Siamkatzen)
- Die Verbreitung "Sozialen Annotationen" wird für eine effektive Suche weiter untersucht.

[3] Ermittelt wird der PageRank eines Dokuments rekursiv anhand von Verweisen auf dieses Dokument. Je mehr solcher Verweise existieren und je höher jener PageRank der hierauf verweisenden Dokumente ist, desto höher fällt entsprechend der Wert des Dokuments aus, auf das verwiesen wird, `http://www.google.com/technology`

Effective Large Scale Annotation Browser (ELSABer) kann, durch Einbeziehung diverser Komponenten, wie eine Suchfunktion bzw. Suchmaschine hier zum Beispiel Lucene und Clusteranalyse [4], zu einer zeitlichen und personellen Suche erweitert werden. Der Nachteil bei dieser Anwendung ist, daß es keine vollständige Implementierung gibt und der Aufwand nicht genau beschrieben ist, da mit steigender Anzahl von "Tags" auch deren Sichtung mehr Aufwand bedeutet.

3.7 Weitere Ansätze

Es gibt weitere Arbeiten, wie "Learning Social Networks from Web Documents Using Support Vector Classifiers"von Makrehchi und Kamel (2006) oder "Data Mining Through Fuzzy Social Network Analysis" von Nair und Sarasamma (2007). Es wurde im erst genannten Artikel versucht mit Support Vector Maschinen halbstrukturierte Datenmengen zu klassifizieren und um die fehlenden Daten zu erweitern. Probleme bei dieser Technik entstehen durch die Voraussetzungen des Trainingsets und dadurch, daß Daten nicht vollständig automatisch bearbeitet werden können. Im zweiten Fall wurde versucht mit Hilfe von Fuzzy Logik ein Data Mining auf schon bestehenden Daten vorzunehmen und darüber ein "Social Network" aufzubauen.
Ein ähnlicher Ansatz wie Polyphonet von Matsuo u. a. (2006) wird in Petrelli u. a. (2006) dargestellt, wo ein Netzwerk für "Relationship-Oriented Computing (ROC)" aufgebaut wurde. Dieses Netzwerk wurde für Konferenzteilnehmer aufgebaut um einfacher in Kontakt mit anderen Teilnehmern treten zu können. Es wurden aus den einzelnen Konferenzdokumenten die Metadaten extrahiert und in einem Ressource Description Framework(RDF) Graph aufbereitet. Dieser wurde analysiert und ein ROC erstellt mit dessen Hilfe man die Beziehungen unter den Konferenzteilnehmern kenntlich machen konnte. Diese Arbeit basiert darauf, dass die Metadaten vollständig vorhanden sind. Das ist bei Konferenzen eher der Fall als bei "einfachen" Unterlagen in einer Organisation.
Ein einfach gehaltener Bericht von Matsumura u. a. (2005) untersucht, wie mit Hilfe eines "Message Boards" und Data-Mining ein "Social Network" aufgebaut und Aussagen über die Teilnehmer getroffen werden können. Es wurde ein Distance Vektor in der Form aufgebaut, dass gesagt wurde, umso mehr Nachrichten hin und her gingen und umso direkter die Nachrichten verschickt wurden, desto höher war die Gewichtung.

[4] Unter Clusteranalyse versteht man ein strukturentdeckendes, multivariates Analyseverfahren zur Ermittlung von Gruppen (Clustern) von Objekten, deren Eigenschaften oder ihre Ausprägungen bestimmte Ähnlichkeiten oder Unähnlichkeiten aufweisen, Jain u. a. (1999)

3.8 Kritische Würdigung der Ansätze

Die hier vorgestellten Arbeiten sind ein kleiner Ausschnitt aus der Forschung die in den letzten Jahren zu diesem Thema gemacht wurde. Die Schwierigkeiten damit, die aussagekräftigen Daten aus dem System herauszufiltern und richtig zu bewerten sind allen Arbeiten gemein. Die verschiedenen Berichte zeigen alle gute Ansätze, Information aus einem Netzwerk heraus zu filtern. Bei den Arbeiten (3.4, 3.6) war problematisch, dass die Dateien und Dokumente nicht bekannt waren und keine exakten Aussagen darüber getroffen werden konnten, mit welchen Datentypen man gerade arbeitet. Eine weitere Schwierigkeit ist die Menge an Daten und der Aufwand der damit verbunden ist. Bei der Arbeit 3.6 entwickelte sich der Vergleich von Eigenschaften und Personen mit einem $O(n^2)$. Bei der Towards Effective Browsing of Large Scale Social Annotations (3.6) konnte über den Aufwand keine Aussage gemacht werden, da nur exemplarisch ein Prototyp entwickelt wurde. Bei Fringe in 3.5 vertraute man mit dem Tag Mechanismus darauf, dass die Benutzer nur korrekte Angaben machen. Es gibt keine direkte Überprüfung der Eingaben. Eine Thematik die bei den Projekten nicht benannt wurde, ist der Zeipunkt oder Zeitraum, wann und wie lange die Informationen gültig sind oder welche wichtige Informationen enthalten sind. Bei Anwendungen wie SmallBlue von Ehrlich u. a. (2007) muss die Problematik des Datenschutzes beachtet werden, wodurch der Ansatz in einem Versicherungsunternehmen schwer umsetzbar ist.
Die in den vorgestellten Arbeiten aufgetretenen Probleme bei der Informationsgewinnung lassen sich wie folgt zusammenfassen: Die aufwendigen Vergleiche von Benutzerinformationen und die Bestimmung der Relationen bzw. Zuordnung von Personen, Gruppen und Schlüsselworten (Vgl. Matsuo u. a. (2006)). Bei Fringe 3.5 muß man den “taggen” der Benutzer vertrauen und hat nur eine beschränkte Kontrollstruktur, womit Profilbeschreibungen nicht vollständig oder einfach falsch sein können. Bei SmallBlue handelt es sich um ein voll funktionierende Suche nach Personen und Sozialen Netzwerken aber die Benutzung von E-Mail Daten ist nicht möglich in dem gegebenen Umfeld. Hinzu kommt das Chat in der Umgebung nicht vorhanden ist. Die Untersuchung von Song u. a. (2005) aus 3.2 ist durch die Analyse von E-Mails nicht direkt nutzbar, zeigt aber ein effizientes Verfahren zur Analyse von Dokumenten auf der Suche nach Experten. Weitere Arbeiten sind nur bedingt nutzbar, weil diese keine komplette Suche abbilden (siehe Matsumura u. a. (2005)) oder nicht automatisiert ausführbar sind (siehe Makrehchi und Kamel (2006)).
In dieser Arbeit soll ein Netzwerk aufgebaut werden und es sollen die speziellen Fähigkeiten von Mitarbeitern herausgefunden werden. Es muss der Datenschutz beachtet werden. Projekte wie Polynet von Matsuo u. a. (2006) untersuchen nur einen bestimmten Bereichen . Andere Arbeiten wie zum Beispiel Information Retrieval besitzen bereits einen bestimmten Satz an Informationen und besitzen eine Kenntniss über die Datenstruktur.
Die Analyse der Organisation und der IT-Struktur hatte ergeben, dass persönliche Daten, wie E-Mail nicht benutzt werden sollen. Eine vollständig, geordnete Struktur ist auf den Servern der Organisation nicht vorhanden. Die Informationen müssen aus den im Intranet vorhan-

denen Daten herausgearbeitet werden. Es müssen die Datenquelle/en evaluiert werden. Es muss versucht werden einen Algorithmus oder Algorithmen zu installieren, die die Informationen extrahieren. Die vorgestellten Arbeiten zeigen Möglichkeiten, Ansätze und Algorithmen, die beim Aufbau von einem Sozialen Netzwerkes helfen aber bieten keine vollständig Lösung.

Kapitel 4

Design

In diesem Kapitel werden das Konzept und die Architektur beschrieben. Aus der Analyse in Kapitel 2 wird ein Konzept erstellt, welches als Basis zur Entwicklung eines Designs und der daraus resultierenden Architektur dient.

4.1 Einleitung

Die grundsätzliche Idee ist es, die Daten aus dem Intranet und dessen verschiedenen Bereichen zu extrahieren, aufzubereiten, zu analysieren, und in einem Personenprofil maschinenlesbar und visuell, als Ergebnis einer Suche, wie in Kapitel 2.4 darzustellen.
(siehe dazu Abbildung 4.1). Es müssen die verschiedenen Datenquellen ausgewählt werden, die zur Informationsgewinnung dienen. Die benötigten Informationen sollen in einem eigenem Persistenzmodell abgelegt werden.
In dem eigenen Modell sind die Daten zu analysieren, unötige Informationen zu entfernen und Datenredundanzen, wenn nicht gewollt, zu löschen. Die Daten sind in einer bekannten Struktur abzulegen, womit das Ergebnis zur weiteren Verwendung zur Verfügung steht. Die Ergebnisse sind in einem eigenen Format abzuspeichern. Die visuelle Darstellung dient zur Betrachtung der Ergebnisses.
Die Idee dieses Vorgehens ist an das Datawarehousprinzip, wie es in von Maur Robert Winter (2002) und H. Mucksch (2000) beschrieben wurde, angelehnt. Es sollen alle Datenbestände in einer Datenbank gesammelt und analysiert werden. Die Daten werden direkt interpretiert, d.h. die Information werden direkt aus den gesammelten Daten extrahiert und weiter verwendet. Im zweiten Schritt soll neues Wissen aus den Daten gewonnen werden. Der genaue Ablauf wird in Kapitel 4.2.1 erläutert.

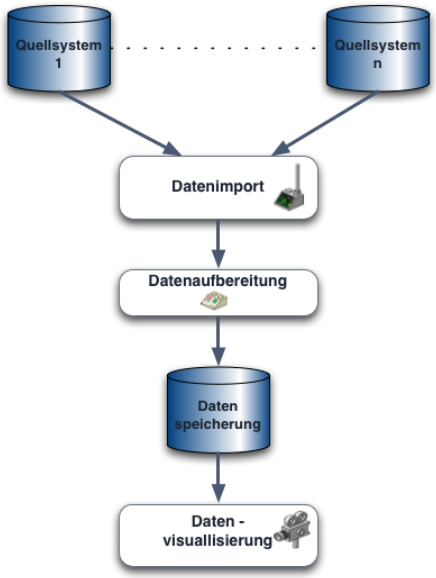

Abbildung 4.1: Das Data Warehouse-Konzept als Basis einer unternehmensweiten Informationslogistik, in H. Mucksch (2000)

4.1.1 Das Finden von relevanten Daten

Die Algorithmen zur Auswertung bestehen aus zwei verschiedenen Ansätzen. Die Daten werden in den Filtern direkt vorsortiert und in die Datenbank geschrieben.
Die Vorgehensweise zum extrahieren und aufbereiten der Daten ist in jedem Filter verschieden und lässt kein allgemeines Vorgehen zu. Es gibt allgemeine Ansätze von Suchmaschinen deren Algorithmen aufgegriffen werden, wie z. B. in dem Buch “Google Hacks” Calishain und Dornfest (2003). Es wird beschrieben, wie eine Beziehung zwischen einzelnen Objekten hergestellt werden kann. Die im Kapitel 3 vorgestellten Arbeiten haben verschiedene Algorithmen entwickelt und beschrieben.
Die gewonnen Informationen aus den verschiedenen Filtern können in einem weiteren Verfahren weiter untersucht werden, um Beziehungen zwischen verschiedenen Informationen herzustellen und ein Personenprofil damit zu erweitern.
In den folgenden Kapiteln werden die einzelnen Bestandteile aus der Abb. 4.1 weiter ausgeführt, begründet und genauer beschrieben.

4.2 Konzept zur Informationsgewinnung

Die Aufgabenstellung ist ein Mitarbeiterprofil zu erstellen, was den Mitarbeiter, seine Fertigkeiten und seine Aufgaben möglichst detalliert beschreibt.
Die Informationen für ein Profil werden, aus dem Intranet herausgesucht. Die Schwierigkeit ist die Zusammenstellung der Daten, woher kann ich die Daten nehmen und welche Daten sind relevant. Es sollte möglichst eine zeitliche Bewertung der Informationen vorgenommen werden (z. B. eine Person hat vor 5 Jahren Wissen erworben und verfügt heute nicht mehr über das aktuelle benötigte Wissen.).
Das hier betrachtete Intranet setzt sich aus verschiedenen Bereichen zusammen. Es gibt Personalinformation (Adressen, Telefonnummern, etc ...), Projektunterlagen, Repositories, Dokumente (z. B. Publikationen, Dokumentationen, Online Bücher, etc ...), Web 2.0 Anwendungen (Blog, Wiki) und Firmeninterna wie Schulungsunterlagen und Informationen über Mitarbeiter Projekte, die nicht direkt abrufbar sind. Diese Bereiche werden in der Abbildung 4.19, als Quellen bezeichnet und bilden den Ursprung der Informationen. Eine genauere Beschreibung der einzelnen folgt in Kapitel 4.2.1. Es müssen zwei weitere Ausprägungen gesondert betrachtet werden. Der erste Punkt ist die Sichtbarkeit von Informationen. Es gibt unterschiedliche Benutzergruppen und damit verschiedene Sichtbarkeiten. Ein “normaler” Mitarbeiter hat nicht den selben Zugang, wie ein Projektleiter oder Manager zu Informationen. Der zweite Punkt ist der Datenschutz, weil personenbezogenen Daten gesammelt werden, die nicht ohne vorheriger Absprache veröffentlich werden dürfen.
Nach Festlegung der zu verwendenen Quellen müssen die Daten aus dem System extrahiert werden. Umso mehr mögliche Quellen identifiziert werden, desto mehr Daten können gesammelt werden. Daraus ergibt sich ein detaillierteres und lückenloseres Profil. Es ist aber anzumerken, das Quantität nicht gleich Qualität bedeutet und es sich herausstellen kann das viele Quellen eher hinderlich und nutzlos sind für eine weitere Analyse.
Es gibt verschiedene Datentypen, die unterschiedlich behandelt werden müssen. Die Daten können strukturiert vorhanden sein, wie z. B. in einer Datenbank. Sie können aber, vollständig unstrukturiert sein, wie auf Fileserver Ebene. In Dateiverzeichnissen werden die Daten von einer Person, in einer für diese ersichtlichen Struktur, abgelegt. Diese Form der “Ablage” ist nur bedingt erkennbar für andere. Eine andere Möglichkeit ist die Informationsbeschaffung “zu Fuss”. Es können Daten in der Art hinterlegt sein, daß kein direkter Zugang zu den Daten möglich ist, also keine computergestützte Bearbeitung, und man sich selbst indirekt den Zugang beschaffen muss, wie z.B. bei Firmeninterna, die nur unter bestimmten Bedingungen herausgegeben werden. Es kann sich um Personalinformationen handeln, die nur einen gewissen Grad an Informationen enthalten darf (z. B. eine Art der Zensur auf Grund des Datenschutzes).
Aufgrund der Unterschiedlichkeit der Quellen/Datentypen müssen zur Extraktion der Daten auch einzelne Anwendungen, jeweils passend zur Quelle, implementiert werden. Diese Anwendungen werden im weiteren als Filter bezeichnet und sind in 4.19 als Adapter bezeichnet.

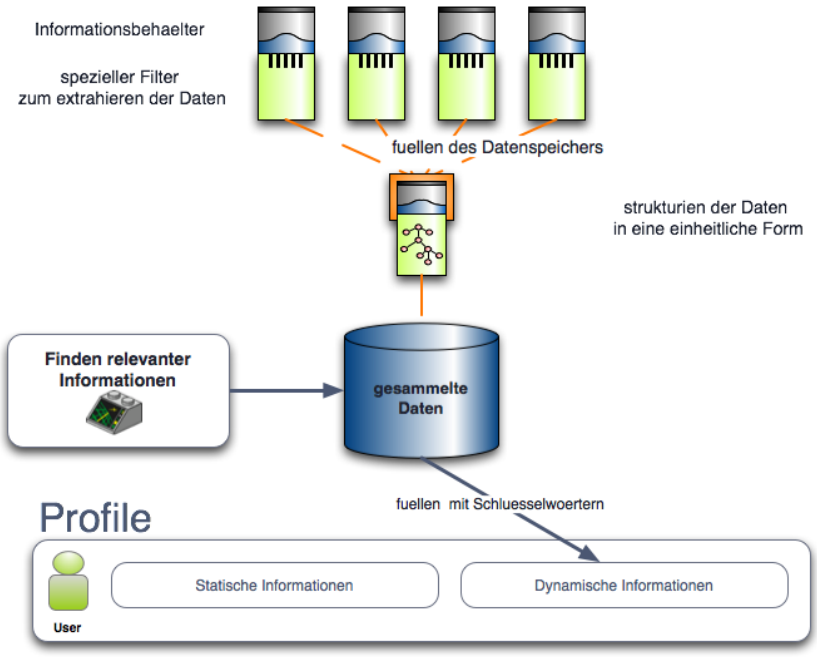

Abbildung 4.2: Konzept zur Gewinnung von Informationen

Mit Hilfe der Adapter werden die extrahierten Daten in eine einheitliche Form gebracht und in einer Persistenzschicht abgespeichert.
Die Strukturierung der Daten erfolgt, im Filter selbst, weil einzelne Filter mit unterschiedlichen Informationen arbeiten und in diesem beurteilt werden wie Daten zu behandeln sein sollen. Die Struktur soll mit der Persistenzschicht festgelegt werden. Die Aufbereitung, siehe Abb. 4.19, ist als Füllen des Datenspeichers beschrieben. Die gesammlten und aufgearbeiteten Informationen sollen in einer Datenbank gesammelt werden. Die Speicherung in einer Datenbank geschieht auf Grund der großen Menge an Informationen, die zu bearbeiten sind, und die Vielzahl an Möglichkeiten zur Analyse der Daten. Das genauere Vorgehen wird im Kapitel 4.1.1 zur Datengewinnung genauer erörtert. Eine Aufgabe beim Design des Datenbankschemas ist die Berücksichtigung der Zuordnung von Informationen zu den richtigen Personen. Eine Vielzahl von Begriffen und Unterlagen muss immer einer Person (eine Person wird an Hand der Personalnummer geführt) oder mehreren Personen zugeordnet sein. Ohne die Zuordnung wäre die Erstellung eines Profils nahezu unmöglich. Die Festlegung der Relevanz von Informationen und deren Korrektheit muss berücksichtigt werden, um zu vermeiden ein Profil mit falschen und unwichtigen Informationen zu überfrachten. Die Erstellung des Datenbankschemas wird in Abschnitt 4.3 beschrieben.
Die Informationen über eine Person, das Profil zwei geteilt, siehe Abbildung 4.19 unten, den statischen Teil, die strukturelle, Informationen, wie der Name, Telefonnummer, etc ...

abgelegt. Der statische Anteil des Profiles ist dadurch kennzeichnet, dass es sich um Informationen handelt, die sich eher selten ändern. Es sind Personenbeschreibungen, direkt aus der Personalabteilung, die dem Profil ein Rahmen geben, der durch die dynamischen Information angereichert und verdichtet wird (siehe Kapitel 4.5).
Der dynamische Anteil wird mit gefunden, und ausgewerteten Informationen aus den verschieden Quellen gefüllt. Diese Informationen sind die Ergebnisse der Auswertung der Daten in den Filtern und der weiteren Analyse aus der Datenbank. Der dynamische Anteil ist gekennzeichnet durch die Interpretation von Daten und die ständige Veränderung der Daten in den Ursprungsquellen, die die Auswertung immer wieder verändern. Die Informationen können für jedes Profil unterschiedlich sein, abhängig von den gefundenen Daten. Die Quellen liefern unterschiedliche Mengen an Informationen zu Personen,weil die Quellen nur in bestimmten Bereichen Daten über Personen zur Verfügung stellen können (z. B. ein Versicherungsangestellter wird nicht in Softwareprojekten direkt auftauchen), siehe hierzu Kapitel 4.5. Es muss im folgenden ein geeignetes Format für das Profil, zur Weiterverarbeitung, im Intranet gefunden werden.
Die verschiedenen Komponenten des Konzeptes werden im weiteren Verlauf des Kapitels genauer beschrieben und am Schluss in einer Architektur abschließend dargestellt.

4.2.1 Informationsquellen

Nachfolgend wird die Aufteilung der einzelnen Bereiche beschrieben, die sich aus der Analyse der Umgebung, in Kapitel 2, ergeben hat.
Die gewählte Aufteilung lässt sich vereinfacht aus der in Tang u. a. (2007a) beschriebenen Graphik entnehmen 4.3. Der Bereich “Company Data” in der Abbildung 4.3 steht für die Personalabteilung und deren Unterlagen über die Angestellten. Die Daten können in unterschiedlichen Systemen verwaltet werden, ein Beispiel hierfür wäre SAP. Es lässt sich so eine Liste aller Angestellten erstellen mit mehr oder weniger vielen Informationen über einen Mitarbeiter. Beispiele wären hierfür Eigenschaften, wie Name, Geburtstag, Abteilung, Position, Beruf, besuchte Weiterbildungsmassnahmen oder Tätigkeitsbereich. Dabei ist entscheidend, wie die Daten gepflegt sind [5]. Eine Beschreibung von einzelnen Abteilungen, des Aufgabengebietes und der Mitarbeitern enthält Informationen, die aus der Personalverwaltung zur Verfügung gestellt werden können. Der Datenschutz ist hierbei nicht direkt betroffen, da es eine allgemeine Beschreibung der Abteilung ist und nicht direkt einer Person.
Wesentlich sind die Fragestellungen, welche Erfahrungen bislang vorliegen, wer gerade woran arbeitet und in welchen Projekten Mitarbeiter involviert sind. Projektarbeit besitzt eine Menge an diesen Informationen. Durch die Projekte kann beschrieben werde in welchen Positionen Mitarbeiter arbeiten, welche Techniken und Werkzeuge eingesetzt werden und welche Aufgaben jemand ausübt. Bei vielen Projekten werden Werkzeuge zur Untersützung der

[5] Der Datenschutz wird hier vernachlässigt

Arbeit eingesetzt, wie z.B. Mircosoft Office Project Mircosoft, IBM Lotus Notes Partner, Web 2.0 Technologien wie Wikis oder Ticketsysteme wie Bugzilla und Jira Ltd.. Diese verschiedenen Werkzeuge bieten eine Vielzahl an Metainformationen zur Gewinnung an Informationen über ein Projekt und Mitarbeiter (siehe dazu Abschnitt 4.2.2). Es werden häufig Informationen zu Projekten umstrukturiert und auf allgemein zugänglichen Fileservers freigegeben. Dort legen jeweils alle Beteiligten die Dokumentation, Berichte und Gesprächsprotokolle ab. Diese Informationsquellen können beliebig viele oder wenig Informationen enthalten. Es ist möglich keine Struktur vorzufinden und auf Grund von einer Menge nicht benutzbarer Daten keine Aussagen zu tätigen. Bei einer klaren Struktur lassen sich an Hand der Metadaten der einzelnen Dateien verschiedene Ausagen über Inhalt, Autor und Zeitraum der Bearbeitung treffen siehe dazu Tang u. a. (2007b).
Ist der Anwender an Dokumentationen in verschiedenen Sprachen beteiligt, siehe Abb. 4.3? Eine Beteiligung an einer Dokumentation bedeutet, dass er Berichte über Projekttreffen angefertigt haben kann. Der Anwender kann im Kontext von Dokumentation zu einem bestimmten Thema, einem Arbeitsbereich, einem Berichte oder einer Präsentationen für den internen Gebrauch auftauchen. Bei Firmen die europaweit arbeiten ist es möglich, in bestimmten Bereichen festzustellen, wer der Autor von Dokumenten ist um festzustellen wer Fremdsprachen beherrscht [6].
Eine weitere Fragestellung ist, ob es an Universitäten oder in Laboren zusäzliche (Konferenz-)Veröffentlichungen gibt. Durch die Informationen lässt sich ableiten, an welchen Themen intensiv gearbeitet wird und welche Personen als Experten gelten können. Durch Co-Autoren lassen sich Verbindungen zu anderen Personen herstellen und es lässt sich feststellen wer sich mit der Thematik auseinandersetzt (siehe Matsuo u. a. (2006) und Farrell u. a. (2005)). Die Literaturreferenzen in Veröffentlichungen bieten weitere Anhaltspunkte für die Eigenschaften und Kenntnisse eines Autors oder zu dem Besitzer des Dokumentes und den Beziehungen zu anderen Personen, vergleichbar mit der " The ACM Digital Library" for Computing Machinery.
Informationen wie Internetlesezeichen und Hotlist bieten viele Anhaltspunkte zu dem Verhalten des Besitzers oder zu dessen Interessensschwerpunkten. Kalendereinträge geben Auskunft über die aktuelle Arbeit des Anwenders, bzw. die Projektbesprechungen oder Schulungs- und Weiterbildungen. Der E-Mailaccount gibt einem Informationen zu den Themen und Kontakten über die der Anwender mit anderen kommuniziert. Diese Informationen dürfen aber nicht für die Profilerstellung benutzt werden, da dieses dem Datenschutz in der Firma wiedersprechen würde. Diese Möglichkeiten sind aber mit aufgefürt, da es sich hier um mein Fallbeispiel handelt und dieses nicht allgemein gültig ist.
Interessant ist die Frage, ob die Daten, die sich ergeben, in diesen "Containern" unter einander eine Beziehung haben. Ein Beispiel hierfür wäre, daß eine Dokumentation für ein

[6] Aus welchen Land oder welchen der Teil der Firma der Angestellte kommt, lässt sich an Hand der Personaldaten feststellen und wird hier nicht weiter betrachtet

bestimmtes Projekt geschrieben wird. Daraus ergibt sich, dass dieser Anwender in diesem Projekt arbeitet. Eine weitere Möglichkeit ist, dass jemand zu einem Thema viele Publikationen besitzt oder beispielsweise promoviert hat. Damit verfügt er über dementsprechend weitreichende Erfahrung in diesen Bereichen. Die unterschiedlichen Beziehungen könnten verschiedene Interpretationen zulassen. Beispielsweise in der Dokumentation, in der die Grösse der Dokumentation und die Anzahl der Beteiligten etwas über den Umfang des Projektes, so wie den Schwerpunkt des Anwenders aussagt. über diese Interpretation erfährt man etwas über die Gruppenaufteilung und das Verhalten anderer Benutzer. Die einzelnen Quellen können sich gegenseitig überschneiden und somit einzelne Informationen bestätigen und verifizieren das diese Information einen grossen Wahrheitsgehalt besitzen. Ebenso ist es möglich das einzelne Informationen sich widersprechen und man diese Information als “Datenmüll” in der weiteren Analyse und Verarbeitung nicht berücksichtigt. Das Problem ist aber die Erkennung des selbigen, wie in den oberen Kapiteln bereits erwähnt.

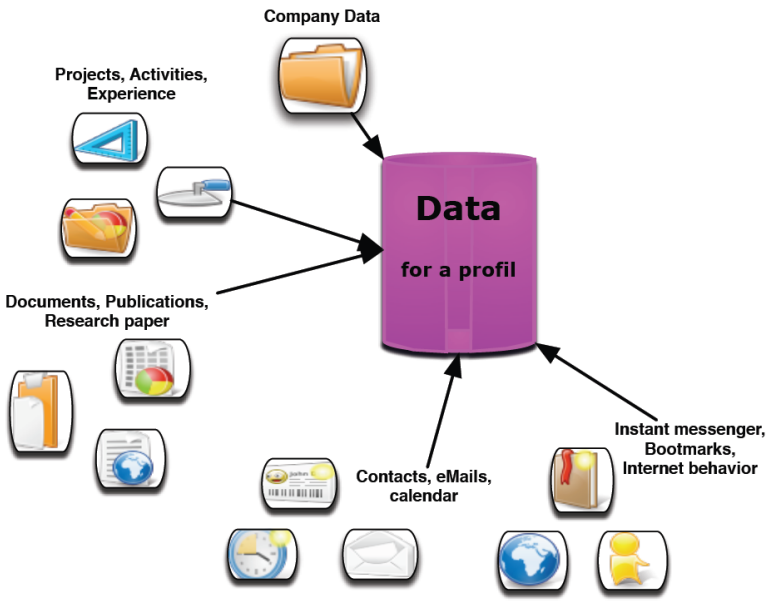

Abbildung 4.3: Darstellung der verschiedenen Kategorien der Daten

4.2.2 Filter

Es ergibt sich die Frage, wie man die verschieden in Abb. 4.3, dargestellten Daten erhält. In der Abbildung 4.19 werden Filter (Komponenten) dargestellt. Es sollen verschiedene Quellen aus Kapitel 4.2.1 betrachtet und erläutert werden, wie man aus diesen Quellen Informationen extrahieren kann.

Personalinformationen

Personalinformationen sind im ersten Schritt statisch und beinhalten Informationen wie Name, Telefonnummer, E-Mail, Adresse, Abteilung, die Anschrift (Firmenadresse und Raumnummer), Position und deren aktuellen Status wie z. B. im Urlaub, Anwesend oder extern tätig.
Um Personenprofile generieren zu können, muss man eine Liste aller angestellten Personen in dem Intranet besitzen. Mit Hilfe einer Anwendung (Komponente)werden die Informationen über eine Person zur Verfügung gestellt. Zusätzlich kann versucht werden weitere Information zu extrahieren. Informationen zu den Qualifikationen einer Person, die Weiterbildungsmassnahmen oder deren thematischen Arbeitsschwerpunkt. Diese Informationen unterliegen sehr stark dem Datenschutz und es muss geprüft werden, ob die Informationen zugänglich sind. Die Daten könnten wie folgt aussehen: Abb. 4.4(z. B. Name, Personalnr, Adresse). Diese Informationen werden von der Personalverwaltung gepflegt und können im Laufe eines Zeitraumes von der Anwendung bzw. dem Filter immer wieder abgeglichen werden (durch den direkten Zugang zu den Daten aus der Personalabteilung ist die Vertrauenswürdigkeit gegeben).

firma	SIGNAL IDUNA Gruppe
fkt	aems
fktn	97440
fkts	AE Methoden/Standards
givenname	Lars
ibm-appuuid	a71d3b00-e1e6-11db-9779-82814612558e
postaladdress	Hamburg
postalcode	20351
standort	Hamburg
standorts	20354
street	
telephonenumber	040/4124-3729

Abbildung 4.4: Ein Beispiel für die Personeninformationen

Das LDAP Verzeichniss des Unternehmens

Das Unternehmen verfügt über ein LDAP Verzeichniss für die Mitarbeiterverwaltung in dem Intranet. Es wird benutzt zur Authentifizierung an den verschiedenen Anwendungen im System (z. B. Intranet, Arbeitszeitkonto, Ticketsystem oder Wiki). Das Verzeichniss ist im Intranet für jeden Mitarbeiter offen lesbar und steht zur freien Verfügung zur Implementierung in Anwendungen. Das LDAP Verzeichniss stellt die Informationen wie oben in Abb. 4.4 zur Verfügung.

Für die Implementierung können zwei verschiedene Bibliotheken eingesetzt werden:

- Es gibt von Sun eine Bibliothek, die für eine einfache Implementierung alle Funktionalitäten für das Lesen von LDAP Verzeichnissen bietet, siehe Java.
- eine weitere Bibliothek wird von Open LDAP und Novell angeboten. JLDAP bietet die volle Funktionalität um einen LDAP Verzeichniss zu administrieren.

In dieser Implementierung wird aber nur die Lesefunktionalität benötigt und die Sun Implementierung gewählt.

Metainformationen aus Dokumenten im Intranet

Das Intranet der Firma stellt zusätzlich zu den verschiedenen Intranetwebseiten mehrere Netzlaufwerke zur Verfügung, auf denen verschiedene Arten von Dokumenten von den eizelnen Abteilungen und Mitarbeitern abgelegt sind. Die Dokumente enthalten verschiedene Thematiken und Berichte. Es handelt sich, wie aus der Tabelle 4.3 ersichtlich, um Office-, Pdf- und Multimediadateien. Diese Dokumente haben verschiedene Dateiattribute, wie Eigentümer, Datumsinformationen, Dateinamen und den Dateityp wie in Tabelle 4.1 bzw. die Metainformationen [7] in Tabelle 4.2.

Attribute	**Erklärung**
Dateiname	eindeutige Zuweisung einer Datei auf einem Datenträger
Datum	Erstellungsdatum der Datei auf einem Datenträger
Besitzer	Eigentümer der Datei
Dateityp	Ein Dateiformat definiert die Syntax und Semantik von Daten innerhalb einer Datei (aus Wotsit).

Tabelle 4.1: Ausgewählte Attribute und deren Bedeutung

Ist der Dateityp dem System bekannt, kann man prüfen, inwieweit die Datei bzw. das Dokument noch weitere Informationen enthalten kann. Die möglichen Informationen sind in der Tabelle 4.3 abgebildet. Es lassen sich Informationen entnehmen, wie den genauen Titel, welche Autoren und Schlüsselwörter, die das Dokument beschreiben.
Im Idealfall, wenn alle Felder angegeben sind, lassen sich die Informationen direkt weiterverarbeiten. Findet man die Autoren der Dokumente in dem Personenverzeichnis des Intranets wieder, können die Schlüsselwörter direkt zugeordnet werden. Gibt es Co-Authoren kann man diese als direkte Arbeitskollegen vermerken und dem Profil zuätzlich die Dokumente als eine Art von Veröffentlichung zuweisen. Die selbe Vorgehensweise gilt für andere Dateien, wie z. B. Multimedia.

[7] Als Metadaten oder Metainformationen bezeichnet man allgemein Daten, die Informationen über andere Daten enthalten (z. B. in einer Datei der Autor).

Metainformationen	**Erklärung**
Titel	Der Titel des Dokumentes
Autor	Ersteller des Inhaltes, kann eine oder können mehrere Personen sein
Schlüsselwörter	Wörter die den Inhalt des Dokumentes wiedergeben
Erstellungsdatum	Das Datum an dem das Dokument erstellt wurde, ist nicht automatisch gleich dem Datum der Datei auf dem Datenträger.

Tabelle 4.2: Ausgewählte Metainformation und deren Bedeutung

Anwendung	Titel	Autor	Schlüsselwörter	Erstellungsdatum
Microsoft Office	✓	✓	✓	✓
Open Office	✓	✓	✓	✓
Acrobat Pdf	✓	✓	✓	✓
Multimedia	✓	✓	**(✓)**	✓

Tabelle 4.3: Eine Auflistung der wichtigsten Dateitypen und deren Metainformationen

Apache Tika

Mit Apache Tika gibt es ein Bibliothek, die unterschiedlichen Dateitypen erkennt und deren Metadaten extrahiert. Weiterhin extrahierte es den Inhalt der Dokumente mit bestehenden Parsern in einfachen Text oder in XHTML. Tika ist Bestandteil von Apache Lucene.
Apache Tika wird in ein eigenes Bundle eingebunden und stellt einen Service zum extrahieren von Metadaten zur Verfügung. Zusätzlich werden die Informationen mit Hilfe des Datenbankservices in die Datenbank geschrieben, wo die Daten, wenn vorhanden mit der Person verknüpft werden, die Besitzer oder Autor der Datei ist.

Begriffsfindung in Dokumenten

Um die Personenprofilen mit Begriffen aus deren Interessensbereich anzureichen, und um die Informationen aus dem vorherigen Filter 4.2.2 zu bestätigen und zu erweitern, wird Text Mining genutzt. Texte in Dokumenten, Dateien werden mit Hilfe von Text Mining Algorithmen analysiert und Begriffe, die am häufigsten auftauchen, genutzt um das Dokument zu beschreiben. Es gibt verschiedene Werkzeuge zum untersuchen von Dokumenten, wie z. B. Gate in Cunningham (2005).

Abbildung 4.5: Eine vereinfachte Darstellung des Ablaufes zur Schlüsselwort suche in Dokumentenordnern

Für die Erstellung des Personenprofils werden die Begriffe aus dem Dokument extrahiert und mit der Information aus welcher diese kommt abgespeichert. Dieses Vorgehen lässt sich auf beliebige Dokumente anwenden (z. B. Dokumente im Intranet, auf Fileservern oder Texte in einem Wiki). Die extrahierten Begriffe oder *Keyphrases* können mit Hilfe der Dateiinformationen (Autor, Dateibesitzer) einer Person zugeordnet werden (siehe Abb. 4.5). Es lassen sich weiterhin verschiedene Data Mining Verfahren (z. B. Clustering) verwenden um die Keyphrases besser Einordnen zu können, weitere Informationen zu erhalten oder die Keyphrases zu bereinigen.

Werkzeuge zur Informationsgewinnung

Es gibt verschiedene Anwendungen, die Data Mining und verschiedene Implementierungen von Alogrithmen zur Informationsgewinnung, als Bibliothek, anbieten. In dieser Implementierung sollen zwei Werkzeuge genutzt werden.

Weka - ist eine Sammlung von "Machine Learning Algorithms" für Data Mining Aufgaben (siehe Kapitel 4.4).

Gate - General Architecture for Text Engineering, ist eine Architektur, Framework und Entwicklungsumgebung zum Entwickeln, Testen und Einbinden von Textkategorisierung oder -klassifikation (siehe Cunningham (2005)).

Beide Anwendungen Bibliotheken sind frei verfügbar (unter GNU) nutzbar und besitzen eine Java Implementierung sowie eine Vielzahl von Bibliotheken, die die Verwendung in der eigenen Applikation einfach gestalten. Zusätzlich verfügen beide über eine große Verbreitung und eine detailierte Dokumentation sowie viel Beispiele und Erweiterungen für den praktischen Einsatz.
Mit Hilfe von Gate und einer Erweiterung von Schutz (2008) sollen die Dateien auf einem Fileserver analysiert werden. Es sollen die verschiedenen Dokumentformen, wie txt, pdf oder Office Dokumente durchsucht werden. Zu jedem Text wird eine Liste mit Worten extrahiert, die den Inhalt des Dokumentes wiedergeben. Die Anwendung wird als eigenständige Komponente, dem eine einfache URL übergeben wird, eingebunden.

Versioning

In speziellen Bereichen wie der Softwareentwicklung wird mit Version Control Systemen gearbeitet, die den Sourcecode und deren Entwicklung während eines Projektes dokumentieren (zwei Beispiele sind CVS und SVN). Es werden zusätzlich Dokumentationen oder kurze Beschreibungen zu Werkzeugen, Libaries/Bibliotheken mit abgelegt (zB. ReadMe Dateien oder Lizenzen). Diese Daten sind in einer bestimmten Struktur abgelegt und können mit Hilfe von Werkzeugen die ein Control Versioning Werkzeug mitliefern, ausgelesen und bewertet werden (siehe Tabelle 4.4, siehe Valetto u. a. (2007) dort wird beschrieben wie man mit Hilfe von Software Versionierungswerkzeugen soziale Netzwerke erarbeiten kann).
Die versionierten Dateien beinhalten die schon bekannten Informationen, wie in den oberen Schritten beschrieben. Zusätzlich wird aber eine Historie zu den einzelnen Dateien angelegt, die aufzeigt, wer, wie häufig und zu welchem Zweck Dokumente bearbeitet oder benutzt. Damit lässt sich direkt erschliessen, woran und wie intensiv eine Person arbeitet. Durch die Metainformationen lassen sich Rückschlüsse ziehen, welches "Expertenwissen" vorhanden ist. Zusätzlich lässst sich über die Kommentare sagen, welche für Änderungen eingeflossen sind und welchen Einfluss diese auf das Projekt haben. Es ist anzumerken, dass dies stark abhängig von der Qualität eines Kommentares ist.

Befehl	Verwendung
list	Der Befehl listet Dateilisten im Repository auf, ähnlich dem `dir` oder `ls`
log	Der Befehl gibt die Log-Messages von Dateien und Verzeichnissen aus (nur die bei denen sich was geändert hat). Eine Besonderheit ist der Export in eine XML Datei.
proplist	Es dient zur Anzeige der vorhanden Propertynamen einer Datei, Verzeichnisse oder einer Revision. Es lassen sich zusätzlich die zugehörigen Werte anzeigen.
status	Der Status zeigt Zustandsinformationen von Dateien, Verzeichnissen und Properties auf lokalen Arbeitskopien an. Es gibt Informationen inwieweit die Informationen nicht berücksichtigt werden, weil sie automatisch erstellt werden.

Tabelle 4.4: Eine Auflistung der wichtigsten Informationen aus dem Versionswerkzeug Subversion, Budszuhn (2005)

Bugtracking- bzw Ticketsysteme

In den vorherigen Filtern sind Personen in Verbindung mit Schlüsselwörtern gesetzt worden. Eine weitere Möglichkeit ist Projekte zu analysieren, wie z. B. in dem SVN-Filter 4.2.2 angedeutet. In einem Ticketsystem werden Informationen, Anforderungen, Probleme und verschiedene Lösungen dokumentiert. Diese Tickets bestehen aus einem Freitextteil und verschiedenen Informationen die über Auswahlfelder vordefiniert sind und die Problemstellung bzw. Anforderungen genauer beschreiben sollen. Es werden die beteiligten Personen, das Datum, die Priorität von Problemen festgehalten (siehe Tabelle 4.5). Die Informationen sind in einer Datenbank hinterlegt und können mit verschiedenen Techniken ausgewertet werden (z. B. Data Mining). Es lassen sich so einzelnen Projekte analysieren und feststellen wer an den einzelnen Projekten beteiligt ist und welche Aufgabenbereiche ihnen zugeteilt werden. Mit diesem Filter ist es möglich, zeitlich bestimmte Artikel und Informationen einzugrenzen. Durch die Projektstruktur lassen sich Verbindungen zwischen einzelnen Mitarbeitern aufbauen und dokumentieren.

Wiki

Durch die als Web 2.0 bezeichneten Technologien haben Wikis in Intranets stark an Bedeutung gewonnen. Wikis sind einfach zu pflegen und jeder kann an diesem Ort sein Wissen mit anderen Mitarbeitern teilen. Oftmals wird es zu Dokumentationszwecken in einzelnen Projekten eingesetzt oder als Veröffentlichsort für Neuigkeiten und Kommunikation genutzt. Es

Auflistung	**Verwendung**
Projekt	Der Name des Projektes für das der Eintrag erstellt wird
Autor	Die Person, die den Eintrag erstellt
Verantwortlicher Entwickler	Der aktuelle Bearbeiter
Typ	Ist das eine Anforderung, ein Fehler oder eine Nachbesserung?
Status	gibt den Bearbeitungsstatus wieder
Gewichtung/Priorität	die Beschreibung wie wichtig die Anforderung oder wie dringend ein Problem ist
betrifft Version	die Anwendung hat verschiedene Versionen und dieses Feld gibt wieder für welche der Eintrag erstellt wurde
Beschreibung	Das Freitextfeld in dem genauer beschrieben wird, worum es sich handelt
Kommentare	Kommentare zu den Beschreibungen
Änderungshistorie	zeigt wie oft ein Eintrag gändert wurde
Transitions	gibt die verschiedenen Stadien wieder, in denen sich der Eintrag befand
Subversion	zeigt den dazugehörigen Code aus dem Control Versiontool an (Subversion).
Commits	Wer wann was verändert hat.

Tabelle 4.5: Eine Aufstellung der verfügbaren Informationen im Ticketsystem Jira

löst die in dem Beispiel des Versicherungsunternehmen starren Strukturen einer Intranetseite, auf der alles doppelt geprüft werden muss ab. Es ermöglicht eine schnelle und flexible Möglichkeit sich über das Intranet auszutauschen. Ein Wiki besteht aus zwei verschiedenen Teilen: Ein Teil besteht aus den Texten die im Wiki abgelegt sind und der zweite Anteil sind die verschiedenen Informationen die mit dem Text mitgeneriert bzw. angelegt werden, siehe Tabelle 4.6.

Information	**Beschreibung**	**Verwendung**
Author	Die Person die einen Artikel schreibt, kann sich ändern wenn weitere Personen den Artikel ändern oder erweitern	Es lässt sich verwenden, um Schlüsselwörter einer Person zuzuweisen
Date	Es gibt verschiedene Datumsanzeigen: wann wurde es erstellt, wann geändert	es lässt Rückschlüsse auf die Aktualität eines Artikels und den Stand der Bearbeitung zu
History	Die Historie listet auf wieviele Änderungen an einem Artikel vorgenommen worden sind.	zeigt, ob sich eine Priorität für bestimmte Bereiche ableiten lässt
Comments	Kommentare werden angelegt wenn ein Artikel verfügbar gemacht wird oder eine Information über Änderungen in einem Artikel auftaucht	Eine Untersuchung nach Begriffen und Schlüsselwörtern nach Gewichtung für den Artikel oder die Änderung

Tabelle 4.6: Eine Auflistung von Metainformationen im Wiki

Die Informationsgewinnung kann auf verschiedenen Wegen erfolgen. Die Daten sind in einer Datenbank hinterlegt und können darüber ausgewertet werden. Die Texte können als Text exportiert werden und wie in der Volltextsuche 4.2.2 beschrieben und nach Schlüsselwörtern ausgewertet werden. Die zusätzlichen Metainformationen werden in ein XML Format exportiert und über XML Parser ausgewertet. Man kann somit wieder eine Relation zwischen Topics und Personen herstellen, etwas über den Zeitraum aussagen und mit Hilfe der Artikel und deren Korrektur feststellen, welche Personen an den gleichen Projekten, Fachbereichen, etc, ... arbeiten.

Zusammenfassung

Es wurden einzelne Filter (Komponenten) beschrieben, mit denen Daten aus dem Intranet für die Erstellung eines Profils gesammelt werden. Es gilt dabei, um so mehr Daten umso besser für die Profilerstellung. Durch die unterschiedlichen Quellen erhält man verschiede-

ne Information und kann somit ein möglichst breites Sprektum an Informationen über eine Person abdecken.
Die Gewinnung der Daten reicht nicht aus für die Erstellung eines Personenprofils; die Daten müssen eine Relevanz für die Erstellung des Personenprofils besitzen. Die Filter dienen zur Gewinnung der Informationen, aber diese müssen zusätzlich aufbereitet werden und in einer strukturierten Form gesichert werden. Die Informationen in der extrahierten Form sind wenig aussagekräftig, da Felder nicht belegt sind oder eine fehlerhafte Formatierung besitzen. Die Felder sind möglicherweise nicht zuzuordnen oder doppeldeutig, daher müssen die Filter/Komponenten eine Vorsortierung vornehmen. Die Filter selbst sind die einzigen Komponenten, die die Datenquelle kennen und wissen, welche für Informationen vorliegen. Die Filter müssen die Informationen sortieren, bevor diese mit anderen Informationen aus anderen Filtern gemeinsam gespeichert werden. Die Sortierung muss nach einem einheitlichen Muster vorgenommen werden, da man sonst beim Speichern der Daten nicht sagen kann, welche für Daten gespeichert werden. Die Semantik würde den Daten verloren gehen. In diesem Moment sind die Daten nicht mehr benutzbar.
Im nächsten Kapitel soll die Struktur vorgestellt werden und gezeigt werden in welcher Form diese Daten weiter interpretiert werden. Das Problem des Datenbankmodells besteht aus den vielen unterschiedlichen Daten, die von den unterschiedlichen Komponenten angeliefert werden. Die Forderung ist diese Informationen in einem gesamten einheitlichen Datenmodell zusammen zu führen. Ein erster Entwurf wird im folgendem Kapitel beschrieben.

4.3 Datenpersistenz

Durch die oben beschriebene Verwendung variabler Filter und der daraus möglicherweise resultierenden Veränderung der Datenstrukturen ist die Entwicklung eines Datenbankmodelles schwierig. Die Forderung ist es, gewonnene Informationen in einem einheitlichen Datenmodell zusammenzuführen ohne eine Erweiterung oder Veränderung der Struktur zu verbieten. Auf Grund dieser Anforderung wird das Datenbankmodell als ein Cache für die gewonnen Informationen, aus den einzelnen Filtern, verstanden. Die Informationen werden dort gesammelt um daraus im weiteren Verlauf Profile der Mitarbeiter zu generieren. Der Cache kann immer wieder gelöscht, neu angelegt und verändern werden. Das Vorgehen wird nötig durch den Austausch der verschiedenen Filter, die Informationen bereitstellen. Wird ein Filter verändert oder kommt ein neuer Filter hinzu wird somit auch das Datenbankmodell verändert.
Es ergibt sich daraus für die Datenstruktur ein vereinfachtes Modell, dass in den folgenden Abschnitten erläutert wird. Zusätzlich zu der flexiblen Datenstruktur ist die zweite Anforderung eine Verwendung der Datenstruktur durch Data Mining. Die Darstellung der Nutzung folgt in Kapitel 4.4.
Die Darstellung 4.6 ist ein vereinfachtes Datenmodell. Es dient der Speicherung der in den Filtern gesammelten Informationen. Es besteht aus zwei Entitäten, Person und Attribut, die,

wie in der Grafik dargestellt, in einer n:m Beziehung stehen. Entität Person enthält die statischen Informationen aus dem Filter 4.2.2. Entität Attribut beinhaltet dynamische, aus den anderen Filtern gesammelte Informationen. Das Attribut ist als Oberbegriff für eine Tabelle, die Informationen aus einem Filter speichert, dargestellt. In diesem Modell ist immer darstellbar, wem die einzelnen gewonnen Informationen aus den einzelnen Filtern zu zuordnen ist. Die n:m Beziehung stellt sich folgt dar: eine Person kann über verschiedene Attribute verfügen und ein Attribut kann mehreren Personen zugeordnet werden. Eine Eigenschaft muss immer eine Person beschreiben. Die Informationen besitzen, ohne die Zuordnung zu einer Person, keine direkte Bedeutung in diesem ERM - Model.

Abbildung 4.6: Eine `n:m` Beziehung zwischen Person und Attribut

Sieht man sich die Grafik 4.4 genauer an, sollte das Entity Person in weitere Teile mit Hilfe der Normalisierungsregel aufgeschlüsselt werden. Diese Aufschlüsselung wird exemplarisch mit dem Abteilungskürzel dargestellt in Abb. 4.7. Eine Abteilung besteht immer aus mehr als einer Person. Die Abteilung ist eindeutig definiert mit dem Abteilungskürzel. Es gibt mehrere Attribute in dem Schema, die normalisiert werden sollten. Es wird im weiteren Verlauf nicht aufgeführt, falls es für das Design nicht entscheidend ist.

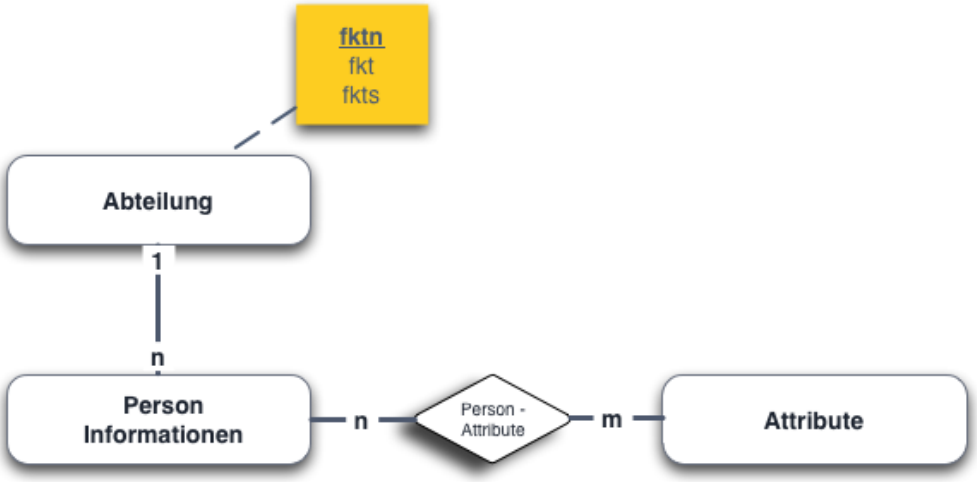

Abbildung 4.7: Normalisierung von dem Entity Person

Erweiterung der Datenspeicherung

Dieses Datenbankmodell dient als erster allgemeiner Evaluierungsschritt. Er beschreibt allgemein eine Verbindung zwischen Person und Attribut. In Abb. 4.8 wird das Attribut aufgeschlüsselt. Beispielhaft sind die drei Filter, TicketSystem, Wiki und Control Versioning, betrachtet worden, die jeweils eine Entität darstellen. Die Filter sind über die eindeutige Benutzer ID mit der Entität Person als n:m Beziehung verbunden.
Es reicht nur bedingt aus, um Informationen über eine Person zu sammeln. Die Attribute werden “blind” einer Person zugeordnet, ohne eine direkte Aussage über den Informationsgehalt zu treffen. Die Abb. 4.9 zeigt eine allgemeine Darstellung der Herkunft der Attribute. Diese kann nur bedingt etwas über die Information aussagen. Sie erklärt nicht den Aussage der Information. In 4.2.1 wurde erläutert, dass mehrere Attribute aus einem Filter gewonnen werden (z. B. Wiki, Ticketsystem, etc ...). Wie können zusätzliche Informationen, die nicht direkt ersichtlich sind, ermittelt werden. Eine Möglichkeit wäre, festzustellen, wie häufig bestimmte Informationen im Zusammenhang zu einer Person auftauchen. Beispiele dafür können Attribute in einer Projektdokumentation sein. Verschiedene Personen besitzen die selben Eigenschaften. In den Projektdaten finden sich Informationen über die Abteilungen, die daran beteiligt sind (Abbildung 4.8).

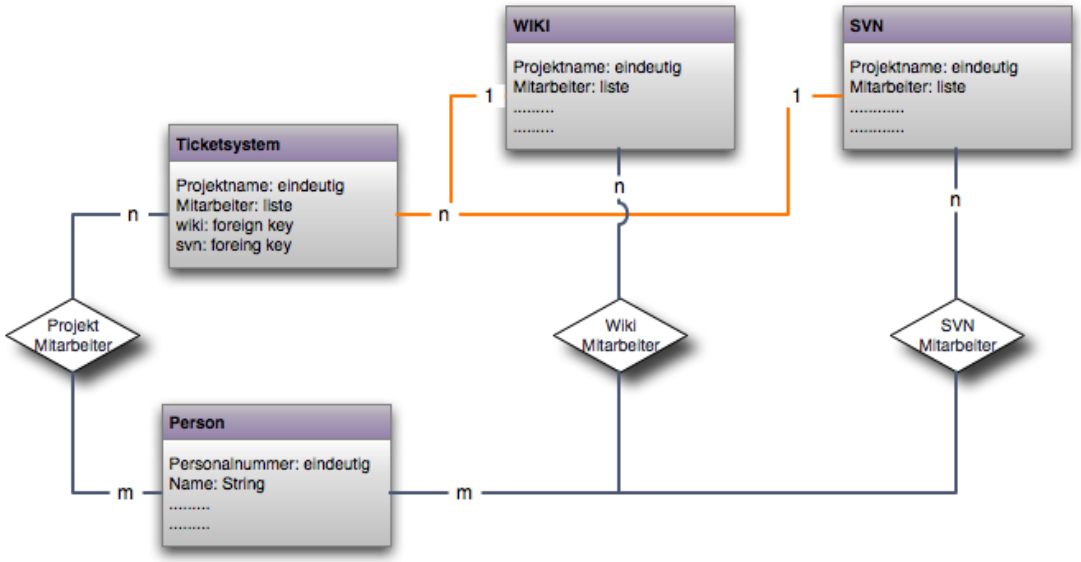

Abbildung 4.8: Darstellung der Zusammenhänge zwischen Ticketsystem, Wiki und Versioning

Diese Daten können benutzt werden, um die Profile mit den Informationen über das Projekt und die Beziehung der Mitarbeiter zueinander zu definieren (Es wird davon ausgegangen,

dass ein Projekt abteilungsübergreifend ist). In Projektattribute kann ermittelt werden, wie oft eine Person in einem Projekt namentlich auftaucht, um zu vermuten, welche Rolle die Person in dem Projekt spielt. Je häufiger Begriffe in Verbindung zu einer Person auftreten, desto mehr ist die Person mit diesen verbunden (siehe z. B. Farrell u. a. (2007), dort wird die Häufigkeit gezählt mit der eine Person zu einem bestimmten Thema verbunden wird).
Eine andere Erweiterung ist der Aufbau eines sozialen Netzwerkes. Welche Personen kennen sich und wie genau hängen Personen miteinander zusammen. Durch die Abteilungszugehörigkeit und gemeinsame Projekte lässt sich ein einfacher Graph aufbauen, der das soziale Netzwerk zwischen den einzelnen Mitarbeitern zeigt (vergleiche J. Chen C.-H. Chen-Ritzo C. A. Chess und Topol (2008), Stanley Wasserman (1994) dort werden die Zusammenhänge und Algorithmen detailliert beschrieben und erklärt).

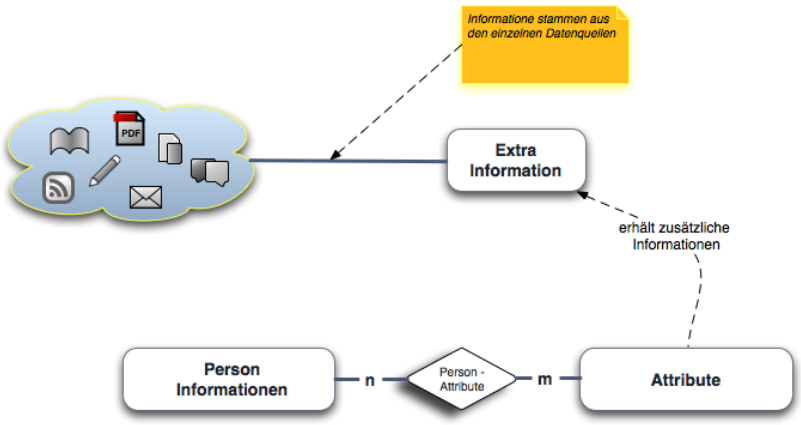

Abbildung 4.9: Eine `n:m` Beziehung zwischen Person,Schlüsselwort und zusätzlichen Informationen

Komplexere Strukturen bei der Datenspeicherung

Im vorherigen Abschnitt 4.3 wurde beschrieben, wie das ERM-Modell mit verschiedenen Interpretationen erweitert werden kann. Mit dem Speichern der Zeiträume von Projekten, dem Datum von Schulungen, Konferenzen oder wann etwas erstellt bzw. verändert wurde, lässt sich eine qualitativ bessere Aussage über die Aktualität der Informationen geben.

- Die Aktualität von Information kann besser bewertet werden.
- Sind die Informationen über eine Person älteren Datums, ist dessen Wissenstandes zu einem bestimmten Thema nur rudimentär oder veraltet.
- Personen lassen sich besser herausfiltern, falls Informationen zu einem bestimmten Zeitraum benötigt werden.

- Durch häufige Zugriffe auf Daten in einem bestimmten Zeitraum wird verdeutlicht, wie intensiv eine Person an bestimmten Thematiken arbeitet.
- Anhand der Zeitspanne, lässt sich erkennen, wie kontinuierlich eine Person an einem Thema arbeitet oder ob sich deren Aufgabenbereich häufig ändert.

Diese Erweiterung verändert nicht direkt das ERM-Modell, sondern es werden einzelne Entitäten um den Wert Zeit erweitert. Es werden zusätzlich Rückschlüsse auf die Person und deren Eigenschaften ermöglicht.
Trotz dieser beschriebenen Erweiterung ist das ERM- Model nicht komplex genug um alle Information aus den beschrieben Filtern aufzuarbeiten. Es dient einer einfachen Informationsselektion ohne einer genauerern Interpretation des Inhaltes. Es fehlt, wie in Kapitel 4.3 beschrieben, die genaue Bedeutung der einzelnen Attribute und deren Aussage. Im folgendem Abschnitt soll diese Problematik genauer thematisiert werden.

4.4 KDD - Knowledge Discovery in Databases

Die Schwierigkeit besteht darin, Informationen zu finden, die nicht direkt enthalten sind. Es ist nicht eindeutig wie eine Aussage über das Wissen eines Mitarbeiters getroffen werden kann oder wie dessen spezielles Fachwissen ihn von den anderen unterscheidet. Damit Attribute eine Person beschreiben, muss man wissen was die Attribute aussagen und in welchem Kontext diese zu einzelnen Themen stehen. Es geht darum, Daten zu extrahieren und Informationen zu interpretieren, die nicht direkt auffindbar sind. Es sollen neue, bisher unbekannte Erkenntnisse gewonnen werden. Dies erfordert einen Abgleich von gefunden Informationen und bisher bekanntem Wissen. Dieses Wissen lässt sich nicht direkt aus den vorhanden Daten extrahieren. An dieser Stelle soll ein *Knowledge Discovery in Databases(kurz KDD)* Prozess benutzt werden, um neues Wissen zu gewinnen.
Ein KDD-Prozess besteht aus einzelnen Schritten. Diese Schritte sind in der Abbildung 4.7 nach dem Vorgehen von Beierle und Kern-Isberner (2000) aufgelistet. Die Umsetzung geschieht in dieser Arbeit analog dazu. Diese Tabelle beschreibt das weitere Vorgehen der Informationsgewinnung zur Beschreibung der Attribute einer Person.
Es soll benutzt werden, um Eigenschaften und besondere Schwerpunkte einer Person zu extrahieren. Welche Schlüsselworte treten besonders häufig in Kombination mit einer Person auf?
Es wurde in dem vorhergehenden Kapitel beschrieben, dass man einer Person Attribute bzw. Schlüsselworte zuweisen kann, aber nicht wie man diese interpretiert. Welche Bedeutung haben diese Schlüsselworte und was sagen diese Worte über eine Person aus?
Diese Informationen müssen aus den Daten interpretiert werden.
Der erste Schritt, die Beschreibung und die Zielsetzung, ist oben schon definiert worden. Es sollen Attribute und Schlüsselbegriffe sortiert und mit einem Oberbegriff versehen werden.

allgemein KDD	Beschreibung	Personenprofilbeschreibung
Hintergrundwissen und Zielsetzung	relevantes Wissen bereitstellen und die Ziele/Ergebnisse definieren	Es sollen Verbindungen von Personen zu einzelnen Thematiken anhand von Attributen gefunden werden.
Datenauswahl	Eine Menge von Daten zur Untersuchung wird festgelegt	Die Informationsquellen aus Kapitel 4.2.1 werden nach deren Nutzen ausgewählt
Datenbereinigung	Ausreisser müssen entfernt werden, Rauscheffekte gefiltert werden	Überflüssige Informationen (die Filter haben nicht die erwarteten Informationen geliefert) entfernen.
Datenreduktion und -projektion	Die bereits behandelten Daten sollen nochmals verbessert und komprimiert werden.	Die einzelnen Attribute können nach Ähnlichkeiten und Oberbegriffen untersucht werden.
Modelfunktionalität	Entscheidung über das Einsatzgebiet, z. B. Klassifikation, Clustering u.a.m	Die Attribute sollen klassifiziert werden.
Verfahrenswahl	Bestimmung eines Data-Mining-Verfahrens, passend zu dem dem KDD-Prozess.	Auswahl eines passenden Algorithmus, unter Beachtung von Aufwand und Komplexität
Interpretation	Aufbereitung der gewonnenen Informationen	die Zuweisung von Intressensgebieten zu einem Personenprofil

Tabelle 4.7: Darstellung der Schritte eines KDD-Prozesses und analog am Beispiel dieser Anwendung

Filter	Attribute	Beschreibung
Metainformationen aus Dokumenten im Intranet	“Keywords”	“Keywords”/Schlüsselworte beschreiben bei Office Dokumenten den Inhalt der Datei
Volltextsuche	Index, Suchmaschine	Das Ergebnis der Indizierung, alle Schlüsselworte des Index
Bugtracking- bzw. Ticketsystem	Beschreibung	Die Beschreibung eines Ticket
Wiki	Freitexte	Die Artikel im Wiki

Tabelle 4.8: Die Auflistung der Filter und deren Attribute zur weiteren Verwendung

An Hand dieser Zuordnung soll eine Person als “Experte” für das Thema gelten. Durch die Anzahl gefundener Begriffe, die ein Thema beschreiben, welches einer Person zugeordnet ist, soll gesagt werden, wie gut das Wissen der Person zu diesem Thema ist.
Der zweite Schritt, die Datenauswahl, ist im Kapitel 4.2.1 ausführlich diskutiert worden. Die Speicherung der ausgewählten Daten wird in Kapitel 4.3 geschildert. Das nächste Kapitel beschreibt die Datenbereinigung.

Datenauswahl

Bei der Datenbereinigung soll als erstes überlegt werden, welche Filter sich für die weitere Untersuchung eignen. Die in Abschnitt 4.2.1 aufgeführten Filter besitzen nicht alle eine Relevanz für das weitere Vorgehen. Es werden die Filter gesucht, die Schlagwörter oder Begriffe enthalten. Die aufgeführte Tabelle 4.8 stellt die Filter und Felder da, die in Frage kommen.

Datenbereinigung

Ein weiteres Problem sind die Schwierigkeiten mit Datenbeständen im allgemeinen, beschrieben in H. Mucksch (2000)

- Unvollständigkeit der Daten - Eines der Hauptprobleme bei dem Herausfinden von Informationen, ist das Fehlen der selbigen. Die Möglichkeit, dass bei den riesigen Datenmengen die relevanten Informationen fehlen, ist sehr groß.
- Datenschmutz - Bei den Daten aus den einzelnen Komponenten/Filtern lässt sich nicht sagen, wie hoch der Ausschuss an fehlerhaften oder falschen Daten ist. Es ist wichtig, die Filter immer wieder zu überprüfen und die Qualität der einzelnen Filter zu verbessern, damit der Datenschmutz verringert wird.

- Irrelevante Informationen - Bei der Datenmustererkennung sind zahlreiche Felder überflüssig, aber es ist oftmals nicht zu sagen welche dies sind. Es ist auch nicht möglich auf diese zu verzichten ohne einen direkten Informationsverlust zu erleiden.

Die Untersuchung nach den obengenanten Problemen soll für jeden Filter in einem eigenständigen Schritt durchgeführt werden. Durch das Wissen über den Inhalt sowie über die Struktur der Daten, sollten die Informationen besser auf ihre Vollständigkeit untersucht werden können. Liefert ein Filter nicht die zu erwartenden Informationen, muss er aus dem Prozess entfernt oder verbessert werden.
Datenschmutz sollte bei den Daten nicht “direkt” auftreten, weil die einzelnen Filter speziell auf die Quellen zugeschnitten sind und es vermieden werden sollte, fehlerhafte oder falsche Daten zu extrahieren. Dies gilt ebenfalls für irrelevante Informationen. Es kann aber vorkommen, dass Informationen sich im weiteren Verlauf der Untersuchung als nutzlos erweisen und diese dann entfernt werden.

Datenreduktion und -projektion

Bei den Filtern, dem Ticketsystem und dem Wiki soll der Inhalt aus freien Texten genutzt werden. Bei diesen Feldern muß ein zusätzlicher Aufwand betrieben werden um Begrifflichkeiten zu extrahieren.

1. Erkennen des Dokumentenformates und der verwandten Sprache.
2. Die Texte müssen in ihre einzelnen Worte zerlegt werden.
3. Die Worte sollen nach Stoppwörtern und ähnlichem analysiert werden.
4. Duplikate sollen entfernt werden.

Eine Möglichkeit zur Durchführung ist der Einsatz von Textmining Werkzeugen. Es gibt verschiedene Werkzeuge, die zur Extraktion der Informationen dienen können, siehe Kapitel 4.2.2.
In Li u. a. (2007) wurde beschrieben, wie man Informationen abstrahieren kann und verschiedene Informationen zusammenfasst. Es soll versucht werden, dieses Vorgehen als eine Vorstufe des Data Mining Prozesses anzuwenden, die Begriffe zu komprimieren und zu abstrahieren (Abb. 4.11, Feature Selection). Ein weiterer Schritt wäre die ebenfalls in Li u. a. (2007) vorgestellte Methode von *Similarity Measurement*, d.h. Begriffe die immer in Kombination auftreten zu finden und aufzubereiten.
Bei Anwendung dieser Aggregationen werden die Originaldaten unter Oberbegriffen komprimiert und zusammengefasst. Dies bedeutet, dass die ursprünglichen Daten nicht mehr verfügbar sind. Die Daten sind für die spezielle Fragestellung der Zuordnung von Begriffen

zu einer Kategorie aufbereitet worden. Sollten in der Zukunft weitere Fragestellungen hinzukommen, müssen die Daten aus den einzelnen Datenquellen extrahiert werden oder es muss auf die Daten aus der Persistenz zurück gegriffen werden. Der Vorgang der Datenauswahl, Datenbereinigung und Datenreduktion muss wiederholt werden.

Modellfunktionalität

Nach dem die Daten aufbereitet wurden, soll jetzt die Analyse der Daten geschehen. Es soll versucht werden, die Informationen zu klassifizieren. Das bedeutet, die einzelnen Informationen werden bestimmten Gruppen zugeordnet. Durch die Einteilung in verschiedene Gruppen bekommt man unterschiedliche Mengen, die mit einem Oberbegriff beschreiben, worum es sich bei den Informationen handelt. Umso weiter Begriffe vom Zentrum der Menge entfernt sind, siehe Abbildung 4.10, desto weniger beschreibt ein Begriff diese Menge. Daraus lässt sich schlussfolgern, wie intensiv eine Person mit dem jeweiligen Thema beschäftigt ist oder über wieviel Wissen diese Person verfügt. Es sollte vermieden werden, dass Überschneidungen der einzelnen Mengen entstehen. Die Informationen sollen nicht zu verschiedenen Mengen gleichzeitig gehören. Die Begriffe dürfen daher nicht zu weit vom “Zentrum” entfernt liegen. Das würde bedeuten, ein Begriff wird vielleicht einer anderen Menge zugeordnet, hat aber keinen direkten Bezug ausser zufälligen syntaktischen Übereinstimmungen (z. B. Java - Programmiersprache, Java - Kaffee).

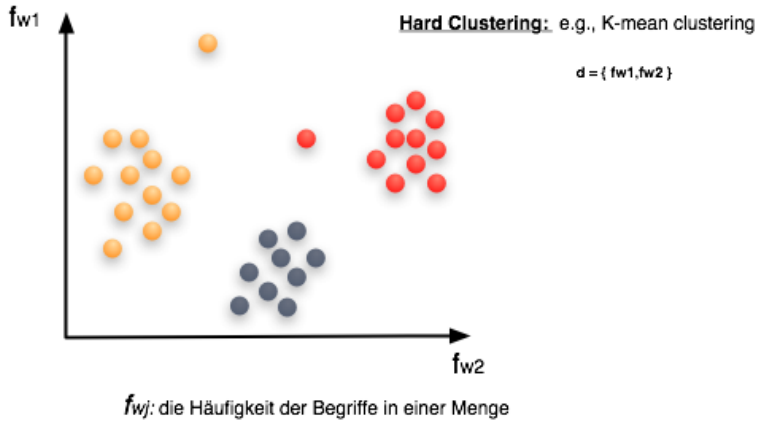

Abbildung 4.10: Beispiel eines zweidimensionalen Clusterings

Das beschriebene Verfahren ist ein Clusteringverfahren (Jain u. a. (1999) und Beierle und Kern-Isberner (2000)) was, die verschieden Kategorien bzw. Bereiche ermittelt. Dabei werden die Begriffe einer oder mehrern Klassen zugeordnet. An Hand der Begriffe bzw. der Daten werden die Gruppierungen aus den Daten selbst gebildet. Abschließend sind in Abb. 4.11 die einzelnen Komponenten im Zusammenhang dargestellt.

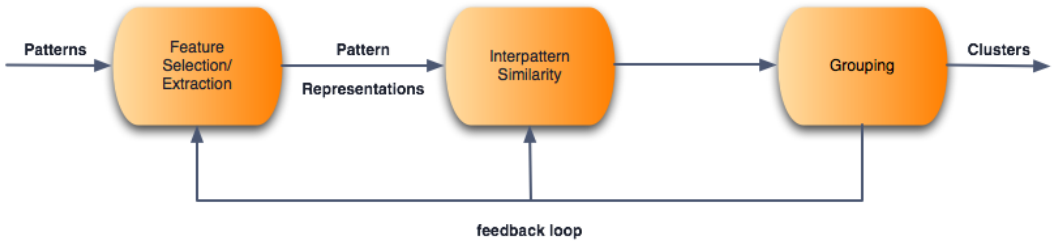

Abbildung 4.11: Die einzelnen Schritte des Clustering Verfahrens, Jain u. a. (1999)

Die *Feature Selection* kann als ein “Prefilter” betrachtet werden, der aus den vorhandenen Informationen eine Untermenge extrahiert. Die Vorselektierung verbessert das spätere Clustering. *Feature Extraction* untersucht die zu sortierenden Daten auf weitere Informationen die vorsortiert werden können und neue Information enthalten. Die beiden geschilderten Vorgehensweisen sind nicht zwingend erforderlich, aber helfen bei der Vorsortierung der vorhanden Informationen, wie bei den Algorithmen in Kapitel 4.4 beschrieben. Die *Pattern Representation* beschreibt die Anzahl von Vorgängen und Skalierungen von Daten, die für das Clustering verfügbar sind. *Interpattern Similarity* beschreibt die Auswahl des Distance Vectors, der beschreibt, wie weit Pattern voneinander entfernt liegen dürfen, um nicht mehr dem selben Cluster anzugehören. Das *Grouping* beschreibt den eigentlichen Clustervorgang, der mit verschiedenen Algorithmen durchgeführt werden kann. Mit dem Ergebnis kann der Vorgang so oft wiederholt werden bis ein optimales Ergebnis erreicht wurde. Bei der Wiederholung können verschiedene Algorithmen oder Stellwerte verändert werden, um das Ergebnis zu verbessern. Im nächsten Abschnitt soll ein mögliches Vorgehen für das Clustering beschrieben werden.

Verfahrenswahl

Das gewählte Verfahren zum Clustering ist der K-means Algorithmus. K-Means ist bewährt und oft verbessert worden siehe Arthur und Vassilvitskii (2006) oder Lu u. a. (2004). Der Ablauf ist wie folgt:

1. Angabe, wieviele Cluster (k) man bilden möchte
2. Es werden beliebige Attribute(k) als Zentrum der Cluster gewählt.
3. Es werden die anderen Attribute den Clustern zugeordnet.
4. Mit Hilfe der vorgenommenen Zuordnungen bestimmt man ein neues Zentrum.
5. Jetzt wird Schritt drei wiederholt, bis es keine Veränderung bei den Zuweisungen mehr gibt und die Cluster feststehen.

Die Bestimmung, wie weit ein Attribut von dem Zentrum entfernt liegt, wird mit Hilfe der Distanzmessung durchgeführt.

Berechnung der Distanz zwischen einzelnen Clustern

Die Distanz wird mit einem “Hilfsalgorithmus”, der die Distanz zwischen dem Zentrum des Clusters bis zu der äussersten Grenze des Clusters oder zum nächten Clusterzentrum bestimmt, ermittelt. Die Bestimmung der Metrik wird häufig mit der *Euclidean distance* durchgeführt.

$$d_2(x_i,x_j) = \left(\sum_{k=1}^{d}(x_{i,k}-x_{j,k})^2\right)^{\frac{1}{2}} = \| x_i - x_j \|_2$$

Die Beschreibung der einzelnen Parameter:

- *d* beschreibt die Dimension in der die feature Vectoren liegen.
- *x* ist ein Feature (oder Attribute). *x* ist ein einzelner Wert, der beim Clustering verwendet wird.
- *feature vector* enthält eine Menge von Werten für *d* in folgender Darstellung: $\mathbf{x} = (x_1,\ldots,x_d)$

Der Algorithmus funktioniert besonders gut, wenn die einzelnen Cluster sich klar voneinander abgrenzen, wie in Grafik 4.10 dargestellt. Eine Alternative ist die Repräsentation mit *Strings* oder *tree structures* (siehe Knuth (1998)). Das Erkennen von ähnlichen Begriffen wurde in Frakes und Baeza-Yates (1992) beschrieben. Mit diesen Algorithmen soll bestimmt werden, wie weit ein Begriff vom Zentrum des Clusters entfernt liegen darf, um noch dem Cluster anzugehören. Die Schwierigkeit eine sinnhafte Clusterbildung zu bestimmen, soll mit den beschriebenen Verfahren verringert werden.
Gelingt es mit diesen Verfahren nicht, sinnvolle Cluster zu erhalten, kann das Verfahren wiederholt werden. Alternativ kann die Datenprojektion oder die Datenauswahl verbessert werden. Die konkrete Implementierung ist in vielen Bibliotheken umgesetzt. Eine mögliche Implementierung wird im folgenden Abschnitt vorgestellt.

Clusterbildung mit Hilfe der Begriffe

Die Daten sollen mit Hilfe eine K-means Algorithmuses klassifiziert werden. Das Verfahren wird von einer Bibliothek aus dem Weka Projekt zur Verfügung gestellt.
Weka ist eine Sammlung von “Machine Learning Algorithms” für Data Mining Aufgaben. Man kann es als Anwendung zur direkten Untersuchung von Datenbeständen benutzen

oder in eine eigene Java-Anwendung einbetten. Weka beinhaltet Werkzeuge für “Data Pre-Processing”, Klassifizierung, Regressionsanalyse, Clusteranalyse, Assoziationsanalyse und visuelle Darstellung. Es ist zusätzlich geeignet für die Entwicklung von neuen Verfahren für “Machine Learning” (siehe Witten und Frank (2005)).
Es wird wie bei der “Keyphrase Extraction” eine eigene Komponente implementiert, die einen zusätzlichen Service zur Verfügung stellt. Der Service liefert eine Liste mit Clustern zurück. Mit Hilfe der in Kapitel 4.2.2 vorgestellten “Keyphrase Extraction” sollen gesammelte Begriffe mit dem k-means sortiert werden, um Interessensschwerpunkte oder Tätigkeitsbeschreibungen für Mitarbeiter zu ermitteln.
Die beschriebenen Vorgehen können ebenfalls als Voruntersuchungen oder -sortierungen implementiert werden, die dann als Services zur Verfügung stehen für eine weitere Verarbeitung.

Ausblick auf mögliche Erweiterungen

Findet man eine sinnvolle Einteilung durch das Clustering, können verschieden Kategorien für Fachbereiche angelegt werden. Diese Kategorien können mit sogenannten “Feature Vectoren” definiert werden. Die Vektoren enthalten die kategorisierten Begriffe, die eine Klasse beschreiben. Anhand dieser Informationen besitzt man ein sogenanntes Trainingsset mit dessen Hilfe man weitere ungeordnete Daten hinzuordnen kann. Es wird hierbei davon ausgegangen, dass die Klassifizierung korrekt ist. Eine vollständige Beschreibung des Vorgehens wird in Makrehchi und Kamel (2006) beschrieben.
Ein weiteres Vorgehen ist das Content Clustering Song u. a. (2005), bei dem eine Kombination von “Semantic content classification” Methoden aus dem Gebiet der Spracherkennung und “Machine learning” genutzt werden. Das geschilderte Vorgehen oder die oben genannten Verfahren lassen sich auf einzelne Informationsquellen anwenden oder als “Vorstufe” für die Informationsgewinnung mit einem anderen Verfahren nutzen. Die in Kapitel 4.4 beschriebene Bibliothek Weka hat eine Vielzahl von diesen Verfahren implementiert und unterstützt eine flexible Umsetzung für den Prozess der Informationsgewinnung.
Die gewonnen Informationen sollen eine Person beschreiben in Form eines Profils. Wie eine solche Beschreibung aussehen könnte wird im nächsten Schritt beschrieben.

4.5 Profile

Eine Person soll in einem Profil beschrieben werden. Aber wie beschreibt man eine Person? Eine Person hat verschiedene Eigenschaften/Attribute, die in Leunziger (1996) als “qualifizierter Faktortyp” beschrieben werden. Der Faktortyp ist eine Person und ein qualifizierter Faktortyp:

ist eine beliebig genau definierte Untermenge eines Faktortyps. Er stellt die kleinste Einheit dar, die als Teil betrachtet wird.

Leunziger (1996)

Ein qualifizierter Faktortype kann als komplexe Struktur betrachtet werden (siehe Abbildung 4.12). Die Darstellung durch den Faktortyp und die Merkmale gebildet. Die Struktur eines Profils muss gleichermassen die fachlichen Anforderungen erfüllen und für die Verarbeitung in der Informatik genügent einfach sein für die Verwendung.

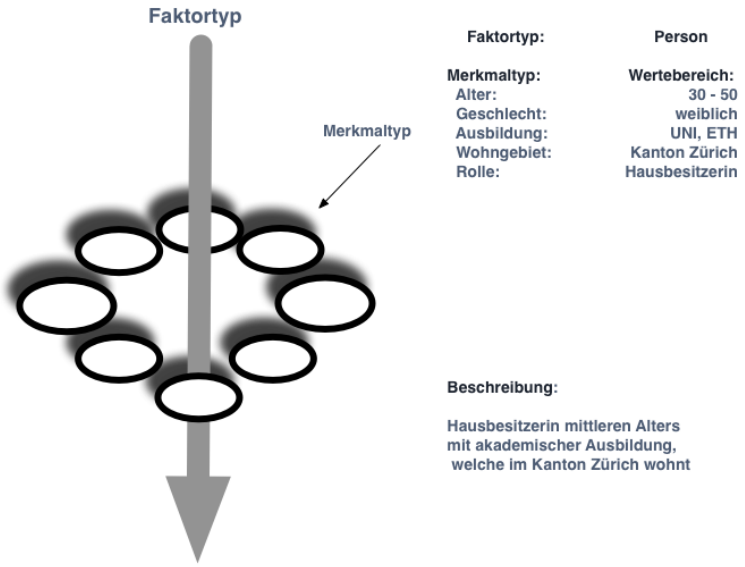

Abbildung 4.12: Beispiel eines qualifizierten Faktortyps nach Leunziger (1996)

Diese Beschreibung lässt sich direkt auf ein Profil abbilden. Aber wie beschreibt man ein Profil? In Wikipedia [8] wird es wie folgt beschrieben:

Configuration settings and other data associated with an individual user or with a group.

Profile, werden häufig, in unterschiedlichen Kontexten (z. B. UML, Instant Messenger, Internetbrowser, etc, ...) genutzt. Das Profil muss von einem Computer lesbar sein, d.h. ein Programm kann automatisiert diese Profile auslesen und interpretieren. Die Struktur dieses Profils muss komplex genug sein, um sämtliche qualifizierten Faktortypen aufzunehmen. Zusätzlich muss das Profil erweiterbar sein. Ist ein Profil über eine Person angelegt, muss es möglich sein, die Information automatisch zu ändern oder weitere Attribute hinzuzufügen. Die Form des Profil sollte einen Standard erfüllen. Das hier beschriebene Profil ist wie in

[8] http://en.wikipedia.org

Kapitel 4.2 beschrieben zweigeteilt. Im nächsten Abschnitt sollen der dynamische Anteil und der statische genauer betrachtet werden.

statische Informationen

Der statische Anteil an einem Profil ist ein Grundgerüst von Informationen, die sich nicht sehr häufig ändern und Informationen enthalten, wie den Namen, Abteilung, Telefonnummer oder dem Standort. Diese Informationen werden aus der Komponente 4.2.2 genommen. Die regulären Personalinformation können mit dieser Komponente direkt aufgebaut werden (siehe 4.9). Alle weiteren Informationen werden dynamisch aufgebaut und beziehen sich direkt auf diese Basisinformationen. Das Beispiel aus Abb. 4.4 für ein Profilgrundgerüst könnte wie folgt aussehen:

Information	**Beschreibung**
firma	Firmennamen
fkt	Abteilungskürzel
fktn	Abteilungsnummer
fkts	Abteilungsnamen
givenname	Vorname des Mitarbeiters
standort	Name der Stadt
telephonenumber	Telefonnummer des Mitarbeiters

Tabelle 4.9: Eine Beispielauswahl für statische Information

Die Tabelle 4.9 stellt einen kleinen Ausschnitt der Daten da die abgespeichert werden können, somit sind in einem ersten Schritt zur Profilerstellung festgelegt wer Zugriff auf das Internet hat und wem im dynamischen Teil Informationen zugeortnet werden können und in welchen Verhältnis Personen zu einander stehen können. Der Vorteil bei der Erstellung des statischen Anteils ist die Gewährleistung der Korrektheit der Informationen.

dynamische Informationen

Die dynamischen Informationen sind mit den Filtern extrahiert und in der Datenbank abgelegten Informationen. Die jeweiligen Ergebnisse varieren stark nach den einzelnen Aktivitäten der Mitarbeiter. Die Sammlung der Informationen werden in einem bestimmten Zeitraum wiederholt und der dynamische Anteil des Profiles neuerstellt. In einem weiteren Schritt sollen die vorhanden Informationen mit den aktuellen Daten verglichen werden. Es lässt sich so ersehen wo sich Schwerpunkte verlagert haben oder die Aufgabenbereiche sich verschieben. Welche Personen auf längeren Zeitraum kontinuierlich auf bestimmten Fachgebieten arbeiten und als Experte gelten können. Die dynamischen Informationen können mit verschieden

Algorithmen und der Datenbank ausgewertet werden (z. B. wie in Kapitel 4.4 beschrieben) und in einer aufgearbeiteten Form für das Profil zur Verfügung gestellt. Die folgende Tabelle 4.10 zeigt Informationen zur Profilerstellung:

Information	**Beschreibung**
Group	Mitglied in Arbeitsgroupen, Übergreifende Fachgroupen
Organization	Mitglied eines Vereines wie z. B. ACM
currentProject	eine Auflistung der aktuellen Projekten
pastProject	an welchen Projekten wurde mitgearbeitet
primaryTopic	Arbeitsschwerpunkte (z. B.. Programmierung)
topic interest	Intressensschwerpunkte
publications	Veröfentlichungen im Intranet, Zeitschriften oder Konferenzen
Document/Image	Erstellte Dokumente, die im Intranet zur Verfügung stehen (Projektdokumentation, Schulungsunterlagen)

Tabelle 4.10: Eine Auswahl für dynamische Informationen,siehe FOAF Spezifikation Miller (2007)

Ein weiterer Punkt, oben bereits erwähnt, ist die häufige Veränderung der Daten über einen zeitlichen Raum, was bedeutet die Profile müssen in einem bestimmten Zeitraum überarbeitet und gegebenenfalls angepasst werden. Einige Beispiel dafür sind: Mitarbeiter verlassen die Firma neue Mitarbeier kommen hinzu. Der Schwerpunkt verschiebt sich, die Firmenstruktur ändert sich. Ein anderer Fall könnte sein, wenn man Schwerpunkte setzen möchte, in zeitlichen Abständen, mit denen sich verschiedene Mitarbeiter auseinander setzen und somit höher priorisiert werden bei ihrer Arbeit.
Es werden verschiedenen Möglichkeiten/Typen von Profilen dargelegt und erläutert wie diese aussehen. Es gilt die im Design festgelegten Kriterien zu erfüllen und zusätzlich eine einfach zu implementierende Lösung zu finden.
Zu den im Design beschrieben Aufgaben und Forderungen kommen in der Implementierung noch technische Erfordernisse hinzu:

- Die ausgewerteten Daten aus dem Datawarehouse müßen in eine Struktur gebracht werden, die einfach generierbar und weiterverarbeitbar sein.
- Die Struktur dieses Profils muss komplex genug sein, um sämtliche Beschreibungen aufnehmen zu können.
- Das Profil muss erweiterbar sein, d.h. falls neue Attribute im Intranet zu einer Person gefunden werden, müssen diese hinzufügbar sein.
- Ist ein Profil über eine Person angelegt, muss es möglich sein, die Information automatisch zu ändern. Der Zeitraum spielt dabei eine Rolle.

- Das Profil muss von einem Computer lesbar sein, d.h. ein Programm kann automatisiert diese Profile auslesen und interpretieren. Die gesamte Anwendung muss auf Grund der Menge der Informationen und deren Komplexität automatisch umsetzbar sein.

Mit Hilfe dieser Kriterien sollen verschiedene Techniken untersucht werden und auf ihre Verwendbarkeit geprüft werden. Die folgenden Profilbeschreibung stammen aus einer der vorherigen Arbeit von Mählmann (2008).

V-Cards

Eine Möglichkeit wäre der Einsatz von V-Cards. Eine VCard ist eine elektronische Visitenkarte, die vom Internet Mail Consortium (Consortium (2008)) standardisiert wurde und als RFC 2426 Dawson und Howes (1998) bereitgestellt wird. Die VCards lassen sich meistens mit verschiedenen E-Mailprogrammen erstellen und in andere Programme übertragen. Eine VCard wird als einfache unformatierte ASCII-Datei gespeichert.Die einzelnen Eigenschaften werden über vordefinierte Parameter gekennzeichnet.

```
BEGIN:VCARD
VERSION:3.0
N:Mustermann;Hans
FN:Hans Mustermann
ORG:Wikipedia
URL:http://de.wikipedia.org/
EMAIL;TYPE=INTERNET:hans.mustermann@example.org
END:VCARD
```

Die Vorteile der VCARD ist:

- Es ist ein Standard und weit verbreitet auf verschiedenen Plattformen
- Es stellt Attribute bereit, um Personen zu beschreiben
- Es ist von Suchmaschinen auf Grund des einfachen ASCII Formates einfach auslesbar

Die Nachteile schliessen die Benutzung aus, weil die Attribute nur Adressinformationen unterstützen und keine weiteren Attribute zur Beschreibung von Projekten, Dokumente, etc. . . . unterstützen. Man kann eine VCARD mit Schlagwörtern anreichern, allerdings reicht dies nicht aus, um eine Person genauer zu beschreiben. Zusätzlich können bei VCARD Sonderzeichen, z. B. Umlaute, verloren gehen, zudem ist die Implementierung von VCards trotz des Standards nicht einheitlich.

XML VCARD

Eine weitere Möglichkeit ist die Beschreibungen in einem XML Dokument abzulegen.

- XML Dokumente sind einfach erstellbar
- Durch Suchmaschinen lesbar, dar XML Dokumente in einfachen Textdateien ablegbar sind.
- XML Formate lassen beliebig viele Atribute zu, die man beliebiig verschachteln kann.

Die Beschreibung der Attribute ist komplex, weil man in einem ersten Schritt einen Namespace definieren muss, um die Einsetzbarkeit der Attribute zu erklären. Ein zweites Problem ist die Festlegung von Attributen. Welche Attribute werden benötigt und wie können diese beschrieben werden. Zusätzlich könnte die Erweiterung von Profilen erschwert sein, da man zusätzliche Attribute erst einführen muss. Es ist an dieser Stelle besser einen Standard zu nutzen der bereits existiert und möglichst alle Attribute direkt abbildet. Der Standard hätte zusätzlich den Vorteil auf bestehende Erfahrungen und bewährte Techniken setzen zu können, die im Hinblick auf Suchmaschinen Vorteile bringen können, z. B. bei der Indizierung durch die Suchmaschine, wenn man keine properitären Lösungen benutzt.
Das W3C hat einen auf RDF basierte Standard zur Implementierung von VCARD veröffentlicht (Iannella (2001)). Dieser integriert VCARD in Semantic Web und ermöglicht die Interpretierbarkeit für Programme. Zusätzlich werden die Probleme der Sonderzeichen gelöst. Dieser Standard erweitert die Attribute und damit die Verwendbarkeit zur Profilgenerierung (siehe Beispiel aus Iannella (2001)).

```
<?xml version="1.0"?>
  <myns:myElement xmlns:myns = "http://www.qqqfoo.com/my-namespace#"
           xmlns:vCard = "http://www.w3.org/2001/vcard-rdf/3.0#" >

      <vCard:FN> Corky Crystal </vCard:FN>
      <vCard:N>
        <vCard:Family> Crystal </vCard:Family>
        <vCard:Given>  Corky </vCard:Given>
      </vCard:N>
      <vCard:EMAIL vcard:TYPE="http://www.w3.org/2001/vcard-rdf/3.0#internet">
           corky@qqqfoo.com       </vCard:EMAIL>
      <vCard:ORG>
        <vCard:Orgname> qqqfoo.com Pty Ltd </vCard:Orgname>
        <vCard:Orgunit> Commercialisation Division </vCard:Orgunit>
        <vCard:Orgunit> Engineering Office </vCard:Orgunit>
        <vCard:Orgunit> Java Unit </vCard:Orgunit>
      </vCard:ORG>
  </myns:myElement >
```

Durch die Erweiterung von VCard mit XML und RDF erhält man zu dem “einfachen” Format ausserdem noch eine semantische Beschreibung, die eine automatische Abarbeitung der Profile möglich macht. Dies war im VCARD Standard und XML selbst nicht möglich. Der Nachteil ist, daß nicht genügenden Attribute zur genauerern Beschreibung einer Person bestehen bleiben. Eine Erweiterung wäre den Standard nutzen und mit eigenen zusätzlichen Attributen anzureichern.
Basierend auf RDF und der Idee von VCARD soll im nächsten Abschnitt ein weiter Standard von W3C betrachtet werden (Für eine genauere Erklärung zu RDF siehe Anhang A).

FOAF

Friend of a friend, Miller (2007), ist eine RDF Beschreibung für Personen und Gruppen im Social Networking. Ziel ist es (beschrieben bei Dumbill (2002)):

- E-Mail durchsuchen und Freunde, Kollegen priorisieren an Hand der Profile
- Unterstützung bei dem Einstieg in neue Gruppen (Community)
- Personen zu finden, die die selben Interessen haben

also die Beziehungen zwischen einzelnen Personen oder Gruppen darzustellen. Dabei sind die Kanten, die die Relationen zwischen Personen bzw. Gruppen beschreiben als Attribute in einem RDF Format abgespeichert. Weiterhin werden die Personen mit deren allgemeinen Eigenschaften in diesem Format beschrieben (ein Beispiel ist LiVEJOURNAL (2008)).
FOAF bietet eine Vielzahl von Attributen, die eine Person beschreiben können. Somit verfügt es über eine großere Anzahl an Beschreibungen als eine VCARD. Die folgende Abbildung 4.13 zeigt einen Überblick der Tags zur Darstellung einer Person. Zusätzlich zu den aufgeführten Punkten wurde FOAF bereits erfolgreich in einer Umgebung eingesetzt, siehe hierzu die Veröffentlichung von Farrell u. a. (2005).
Im folgendem Abschnitt wird eine Bibliothek vorgestellt mit der die beschriebenen Anforderungen umgesetzt werden sollen.

Automatische FOAF Generierung

In Farrell u. a. (2005) wurde ein Framework Jena (Programme (2009)) zum Erstellen von FOAF Profilen vorgestellt. Das in Java geschrieben Framework Jena wird in der Implementation als Komponente zur Profilerstellung benutzt. Es ist von Hewlett-Packard zum Bau von Semantic Web Anwendungen entwickelt worden. Es stellt eine Umgebung für RDF, RDFS und OWL zur Verfügung. Desweiteren enthält es einen regelbasierten SPARQL Interpreter. Das Framework enthält folgende Bibliotheken:

- eine RDF API

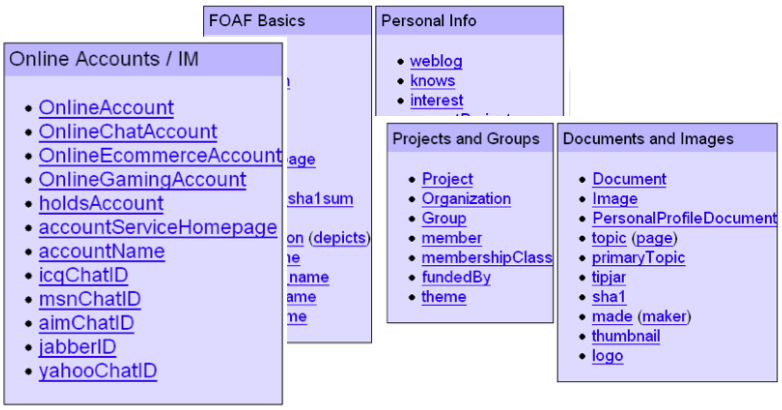

Abbildung 4.13: Attributdarstellung von FOAF

- Lesen und schreiben von RDF in RDF/XML, N3 und N-Triples
- Eine OWL API
- Ein Speichermechanismus
- Eine SPARQl Query Engine

FOAF ist eine Spezialisierung von RDF. Jena kann mit dem Namespace von Miller (2007) ein "Vocabulary" erstellen. Mit dessen Hilfe wird ein FOAF Profil gelesen und geschrieben. Jena bietet mit der RDF API eine einfache Schnittstelle zum generieren der einzelnen N3-Triples aus denen eine Friend of a Friend Datei besteht.
Die Bibliothek wird in der Implementierung (Kapitel 5) als eigene Komponente eingebunden und erhält mit Hilfe der Datenbankservice die benötigten Informationen um ein FOAF Profil zu generieren.

4.6 Framework für die Komponentenstruktur

In den einzelnen Abschnitten wurden bisher die einzelnen Komponenten - wie man Informationen extrahiert, die Daten in einer Datenbank ablegt und woraus ein Profil besteht - beschrieben. In diesem Abschnitt soll dargelegt werden, wie die Komponenten mit einander interagieren bzw. beschrieben werden, was bei den Schnittstellen zur Kommunikation beachtet werden muss. Die einzelen Teilbereiche wurden als Komponenten beschrieben. Was bedeutet dieser Begriff in der Architektur?

> Eine Software-Komponente ist ein Software-Element, das konform zu einem Komponentenmodell ist und gemäßeinem Composition Standard ohne Ände-

> rungen mit anderen Komponenten verknüpft und ausgeführt werden kann. William T. Councill (2001)

Eine Komponente arbeitet unabhängig von anderen Bereichen der Anwendung. Was bedeutet das für ein Framework, welches dazu dienen soll diese stark unabhänigen Komponenten zu verbinden? Was sind weitere Anforderungen und wie sollen diese realisiert werden?
Die einzelnen Quellen die von der Anwendung mit Hilfe der Filter abgefragt werden sollen, sind vollkommen unabhängig von einander und können sich im Laufe der Zeit ändern oder nicht mehr verfügbar sein. Durch die Veränderung in den verschiedenen Quellen zur Datengewinnung müssen die Filter selbst sehr dynamisch veränderbar sein. Bei Veränderungen der einzelnen Filter darf nicht das ganze System beinflusst werden durch Veränderung in einem.
Daraus folgt: Die Komponenten sollen sich gegenseitig nutzen ohne direkt von einander abhängig zu sein. Die Komponenten sollen dynamisch in das System eingebunden werden. Es soll zur Laufzeit möglich sein, neue Komponenten einzubinden oder alte zu entfernen ohne das komplette System neu zu starten oder neu kompilieren zu müssen.
Die Datenbankkomponente darf nicht von den Filtern abhängen und muss eine einheitliche Schnittstelle für die Anwendung bieten, es muss auch erweiterbar sein auf mehrere Datenbanken.
Da diese Module viele personenbezogene Daten aus dem Intranet extrahieren, sollte der Sicherheitsaspekt mit berücksichtigt werden. Es sollte möglich sein, ein Sicherheitskonzept zum signieren der Daten sowie eine Zugriffsverwaltung zu berücksichtigen. Diese verschiedenen Anforderungen werden weiterführend in dem Bereich Verteilte Systeme behandelt und genauer beschrieben (siehe hierzu Tanenbaum und van Steen (2006)).
Man kann verschiedene Programming Frameworks betrachten, wie CORBA Component Model (CCM), Microsoft Common Object Model (COM) oder Enterprise Java Beans (EJB), die alle in Verteilten Systemen eingesetzt werden können. Diese basieren aber auf einem sehr komplexen Modell zur Programmierung und schränken den Programmierer bei der Implementierung stark ein. Ein wichtiger Punkt ist der nicht dynamische Aktualisierungsmechanismus. Eine weitere Möglichkeit ist Service-Oriented Architekture (SOA), welche aber einen zusätzlichen overhead durch Webservices oder andere Kommunikationsmechanismen mit sich bringt.
Das hier verwendete Framework ist OSGI Alliance (2007). Es erüllt alle in diesem Abschnitt beschriebenen Anforderungen und findet in verschieden Architekturen wie z. B. Eclipse Verwendung. Eine genaue Beschreibung und eine Erläuterung der Arbeitsweise von OSGI befindet sich im nachfolgendem Kapitel.

4.6.1 Was ist OSGI

Die *Open Services Gateway Initiative (OSGI)* Alliance wurde 1999 gegründet. Die OSGI Alliance hat das Ziel eine auf dynamischen Komponenten basierende Java Plattform zu schaffen bzw. schon zu sein . In der Spezifikation Alliance (2007) sind bisher folgende Anforderungen umgesetzt (vgl. Alliance (2009)):

Reduced Complexity - In OSGI werden die einzelnen Komponenten als Module bzw. *Bundle* entwickelt. Die einzelnen Bundles verstecken ihren inneren Aufbau voreinander und kommunizieren über vorgegebene Schnittstellen. Es reduziert nicht direkt die Komplexität, aber es gibt einem Programmierer die Freiheit die Bundles intern zu verändern ohne das andere Bestandteile außerhalb des Bundles betroffen sind.

Reuse - Die OSGI Komponenten ermöglichen es einfach Drittanbieter-Komponenten einzubinden. Es gibt verschiedene Anbieter die Ihre Anwendung in einem OSGI Bundle zur Verfügung stellen.

Real World - Das OSGI Framework ist ein dynamisches Framework. Es ermöglicht das Ändern, Aktualisieren und Neuinstallieren in der laufenden Umgebung. Das OSGI Framework stellt Services zur Verfügung an denen die einzelnen Bundles registriert und abgemeldet werden.

Easy Deployment - Die Spezifikation des OSGI Framework gibt vor, wie Komponenten installiert und verwaltet werden sollen. Diese API bietet eine Vielzahl von *managment agents* (z. B. TR-69 management protocol driver, OMA DM protocol driver, a cloud computing interface für Amazon's EC2 oder ein IBM Tivoli Management System) die eine Anbindung an andere Systeme vereinfachen.

Dynamic Updates - Das OSGI Komponenten-Modell ist ein dynamisches Modul. Bundles können während der Laufzeit installiert, gestartet, angehalten, aktualisiert und deinstalliert werden. Durch diese Funktionalität muss das Gesamtsystem nicht neu konfiguriert oder neu gestartet werden.

Adaptive - Das Framework erlaubt es unterschiedlichen Komponenten mit einander zu kommunizieren und sich gegenseitig ihre Dienste über einen Service Manager anzubieten (in einer heterogenen Struktur).

Lazy - OSGI benutzt *Lazy Initialize* (vgl. Gamma u. a. (1995)). Einzelne Bundles werden nur gestartet wenn diese erforderlich sind.

Simple - Die API ist sehr klein und einfach gehalten. Es entsteht kein overhead durch die Verwaltung der Bundles und deren Kommunikation. OSGI arbeitet mit Simple Objects ohne Deployment Diskriptoren (POJO).

Small - Das Framework ist darauf ausgelegt auf “Embedded Devices” zu arbeiten. Das System kommt in der kleinsten Version mit 300kb aus.

Secure - OSGI benutzt die “Java Secure API”. Die einzelnen Bundles können mit einer Signatur versehen werden um das Einspielen von “falschen” Bundles zu verhindern (z. B. “Man in the Middle Attack”).

Versioning - OSGI verwaltet die Bundles intern und informiert über Versionsprobleme zwischen den Bundles.

In der OSGI Alliance sind viele große Firmen vertreten (z. B. Oracle, IBM, Samsung, Nokia, IONA, Motorola, NTT, Siemens, Hitachi, Deutsche Telekom, Redhat, Ericsson, etc. ...). Dadurch ist die Akzeptanz und die Verbreitung von OSGI gewährleistet. Zusätzlich entstehen unterschiedliche Bundles die bei der Implementierung von Anwendungen den Aufwand verringern.
Die beschrieben Eigenschaften von OSGI werden durch die Struktur des Frameworkes definiert. Der Hauptbestandteil ist die *Service Registry*. In den folgenden Abschnitten soll die Service Registry und die wichtigsten Bestandteile von OSGI genauer beschrieben und erklärt werden.

4.6.2 OSGI Service Registry

Eines der Merkmale von OSGI ist die oben bereits erwähnte Service Schnittstelle. Mit der Service Registry werden die einzelnen installierten Bundles im System verwaltet. Über die Service Registry kann eine Komponente eigene Service anbieten (Schritt 1 in Abb. 4.14) oder andere Services nutzen(Schritt 2). Ist ein Serviceangebot verfügbar, kann ein Bundle den fremden Service nutzen (Schritt 3) (siehe Tavares und Valente (2008)). Ein besonderer Punkt ist die Dynamik des Frameworks, sobald ein Bundle `A` einen Service anbietet oder ein anderes Bundle `B` sucht, werden die beiden Bundles mit Hilfe von der Service Registry verbunden.

Whiteboard Pattern

Die Service Registry ist ein Schlüsselelement des OSGI Framework. Das Whiteboard Pattern beschreibt das Konzept der Service Registry und wird in Kriens und Hargrave (2004) behandelt. Das Pattern beschreibt wie die Bundles in dem System registriert werden und ihre Dienste zur Verfügung stellen.
Das Whiteboard Pattern basiert auf dem Listener Patter was wiederum Bestandteil des Observer Patterns ist (Gamma u. a. (1995)). Bei dem Whiteboard Pattern registriert man sich nicht mehr direkt an dem Obserbable, wie beim Listener Pattern, sondern an der Registry. Der Observer muss nicht mehr das zu beachtende Objekt kennen. Dadurch gewinnt man

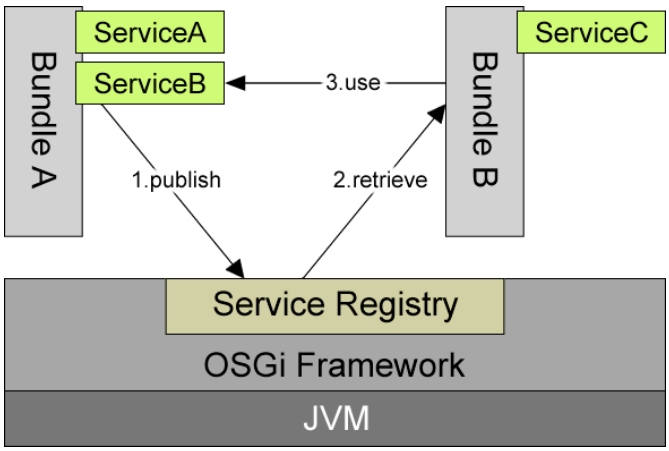

Abbildung 4.14: OSGI Service Registry (aus J.S. Rellermeyer und Roscoe (2007)

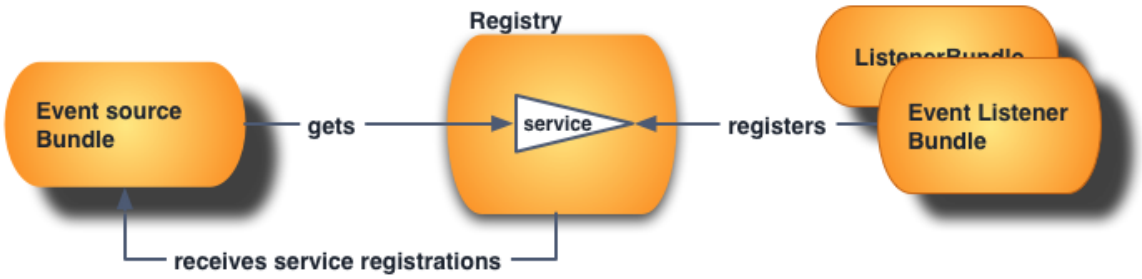

Abbildung 4.15: Whiteboard Akteure im OSGI Framework (vgl. Kriens und Hargrave (2004)

die Dynamik wenn ein passende Obserable Objekt in das System kommt, wird man automatisch registriert. Das OSGI Framework hat eigene *Event Listener*, die als eigener Service Bundle fungieren. Ein Bundle meldet sich bei der Service Registry, wenn es einen Service benutzen möchte, register Listener Bundle in Abb. 4.15. Ein Bundle registriert sich an der Registry, wenn es informiert werden will, sobald ein bestimmtes Ergeignis eintritt. Durch die Verwaltung der Services und der Bundle, wird ein *Event source Bundle*(Abb. 4.15) nicht an der Registry angemeldet. Der *Event source Bundle* bekommt mit dem *get* diesen Service zugewiesen. Es wird in der Service Registry vermerkt, wer den Service anbietet.
Vergleicht man das Listener Pattern aus Java mit dem Whiteboard Pattern besteht der Unterschied zwischen diesen beiden in der Kopplung zwischen den zu beobachtenden Objekt (Observable) und dem Beobachter (Observer). Bei dem Listener gibt es eine lose Kopplung vom zu beobachtenden Objekt zu dem Observer. Das Objekt ist egal wer alles benachrichtigt wird. Der Observer muss das zu beobachtende Objekt kennen um auf Änderungen zu reagieren. Es entsteht eine enge Kopplung die einen hohen Aufwand erfordert, weil die *Event Listener* selber implementiert werden müssen. Bei dem Whiteboard Pattern entfällt diese Kopplung. Die Service Registry implementiert diese Listener als ein eigenständiges Bundle. Durch die Kontrolle der *Event Listener* bietet OSGI weitere Vorteile:

Debugging : Die *Event Listener* sind sichtbar in der Registry und können durch Framework Werkzeuge eingesehen werden. Die Abhängigkeiten zwischen den Bundles sind transparenter.

Security : Das OSGI Framework hat Kontrolle darüber, wer einen Service nutzt. Es bestimmt darüber wer Zugriff auf ein *Event source Bundle* erhält.

Properties : Die Registry bietet die Möglichkeit des Anlegens von Eigenschaften für ein Service. Es lassen sich dann Untegruppen von *Listeners* anlegen für ein Service.

4.6.3 Implementierung eines Bundles

In diesem Abschnitt wird am Beispiel eines konkreten Bundles erklärt, was ein Bundle ist und was die Bestandteile eines Bundles sind.

> Ein Bundle ist eine fachlich oder technisch zusammenhängende Einheit von Klassen und Ressourcen, die eigenständig im Framwork installiert und deinstalliert werden kann. Wütherich u. a. (2008)

Fachlich ist ein Bundle eine einzelne Komponente. Durch die Verwendung von Komponenten bleiben die einzelnen Bestandteile einer Anwendung wiederverwendbar und austauschbar. Beispiele dafür sind die GUI-Elemente, z. B. swt ui, web ui oder eine Konsolen Komponente.

In der Abb. 4.16 ist ein solches Bundle dargestellt. Technisch besteht es aus einem Java Archiv die Implementierung, der Funktionalität beinhaltet. Zusätzlich sind die dafür benötigten Bibliotheken mit enthalten. Zusätzlich ist ein *Bundle Manifest* beigefügt (MANIFEST.MF), indem die verschiedenen Informationen, wie dessen Name oder die deklarativ beschriebenen Packages beschrieben werden. Um ein Bundle nutzbar zu machen für andere Bundles, müssen dessen Klassen und Packages explizit exportiert werden. Diese Angabe erfolgt deklarativ in der Manifest-Datei. Das selbe Vorgehen wird, als Import genutzt, um Klassen und Ressourcen aus anderen Bundles zu nutzen. Die Java Klasse *Activator*, siehe Abb. 4.16, als Minimum die zwei Methoden *start, stop*. Die *start* Methode wird aufgerufen wenn das Bundle aktiviert wird und *stop* wenn es gestoppt wird. Mit Hilfe des Aktivators kann ein Bundle beim Start und beim Stopp beliebige Operationen ausführen. Spezifiziert wird der Aktivator über *Bundle-Activator: classname* in der MANIFEST.MF.
Das OSGI-Framework sorgt zur Laufzeit dafür das jedem Bundle, die benötigten Packages zur Verfügung gestellt werden. Eine weitere Möglichkeit der Entkopplung können Bundles Services verwenden. Ein Service ist ein Java-Objekt. Es wird über ein Interface-Namen systemweit zur Verfügung gestellt.

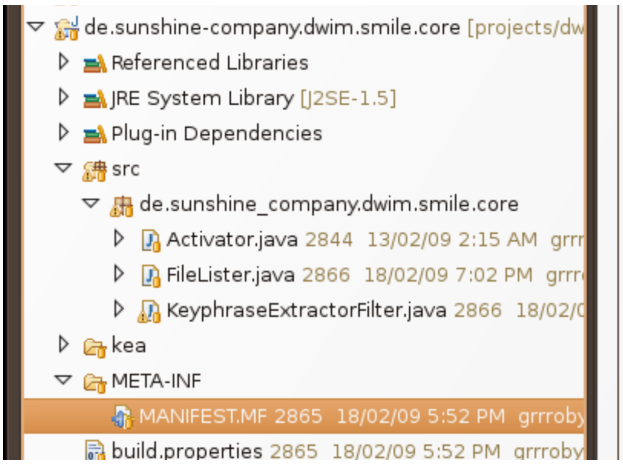

Abbildung 4.16: Ein Bundle mit Bundle Manifest

Angemeldet und verwaltet werden die Services über die in Kapitel 4.6.2 bereits erwähnte Service Registry. Die Service Registry ist ebenfalls ein Bundle, dass zentral und bundleübergreifend bereitsteht.
An dieser können die angemeldeten Dienste von jedem beliebigen Bundle abgefragt werden wie in Abb. 4.17 dargestellt. Ein benötigter Service kann von einem Bundle an der Service Registry angefragt werden, ohne konkret zu wissen wer diesen Service bereitstellt oder wie dieser Implementiert ist. Mit der Dynamik der OSGI Service Plattform können Service “kommen und gehen” Die einzelnen Bundles/Komponenten müssen sicherstellen, dass ein Service, der angeboten werden soll im System bereitsteht oder nicht. Die Verwaltung der

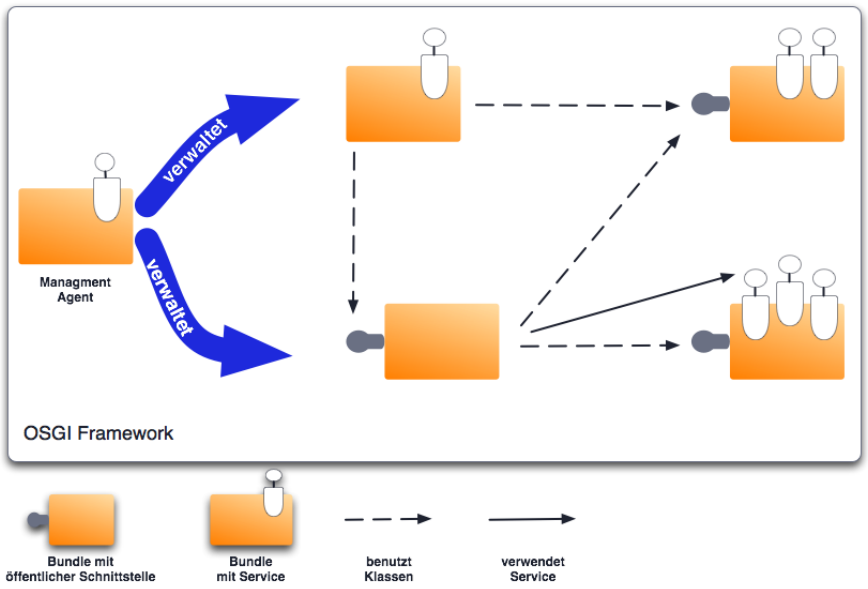

Abbildung 4.17: Basiskomponenten OSGI Framework aus Wütherich u. a. (2008)

Bundles erfolgt über ein *Management Agent*. Dieser Management Agent organisiert das Installieren, Deinstallieren, Starten und Stoppen von Bundles. Er ist gewöhnlich als ein Bundle im Framework implementiert.

Logische Schichten in OSGI

Das OSGI-Framework ist die Basiskomponente der OSGI Service Plattform. In diesem werden alle Bundles und deren Services installiert und ausgeführt. Das Framework ist in verschiedene Schichten gegliedert, vgl. Abb. 4.18. Hier sollen kurz diese Schichten vorgestellt und erklärt werden.

Module-Schicht - Es definiert das Modulkonzept des Frameworks und legt fest das ein Bundle eine eigene Einheit ist die eigenständig in- bzw. deinstalliert werden kann. Alle Bestandteile des Frameworks werden als Module betrachtet.

Lifecyle-Managment-Schicht - Die Schicht beschreibt die dynamischen Bestandteile eines Bundles. Es legt fest welche Zustände während eines Lebenszykluses durchlaufen werden können. Der oben beschriebene Management Agent implementiert dabei ein Schnittstelle mit der ein Lifecyle gesteuert werden kann. Diese Schnittstelle wird von Alliance (2007) Die Implementierung ist abhängig von der jeweiligen Umsetzung des OSGI-Frameworks.

Service Schicht - Die Service Schicht legt fest wie die einzelnen Objekte, der Bundles verwendet werden. Die Services werden über einen eindeutigen Namen an der Service Registry angemeldet. Damit können dann andere Bundles den Service abfragen und die Objekte nutzen.Durch die Dynamik des Systems kann es passieren, dass Service plötzlich nicht mehr erreichbar sind. Um das Arbeiten mit den Services zu vereinfachen sind in der Spezifikation drei Möglichkeiten (*Service Listener*, *Service Tracker* und *Declarative Services*) beschrieben mit dem ein Bundle auf diese Veränderung reagieren kann.

Security-Schicht - Die Security Schicht ermöglicht und spezifiziert das Verwenden von signierten Bundles und die Verwendung von Ausführungsberechtigungen für einzelne Bundles.

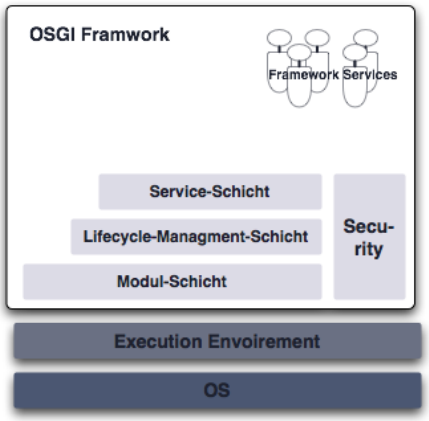

Abbildung 4.18: Logische Schichten im OSGI Framework aus Wütherich u. a. (2008)

4.7 Architekturansicht

Das in Kapitel 4.2 vorgestellte Konzept wurde hier in einer fertigen Architektur umgesetzt.
Die gesamte Architektur wurde auf Basis einer dreischichtigen Architektur konzipiert (siehe Abbildung 4.19) und ist an die Architektur von Balzert (1998) angelehnt. Beschrieben und umgesetzt wurde sie in Tavares und Valente (2008).
Es gibt eine Schicht für die Komponenten, d.h. die Filter und die Komponente zur Erstellung der Profile. Es gibt eine Kommunikationsschicht, die die einzelnen Komponenten miteinander verbindet und als letzte Schicht die Datenbankschicht und deren Data Access Objekts

(DAO). In der dargestellten Architektur ist die Datenbankschicht unabhängig von allen anderen Komponenten und ist, wie in Abb. 4.19 dargestellt, nur über die Kommunikationsschicht erreichbar.
Die Kommunikationsschicht ist die Verbindung zu den einzelnen Ressourcen. Es ist vergleichbar mit einem *Enterprise Service Bus* der dazu dient unterschiedliche Dienste zur Verfüng zu stellen.Sämtliche Zugriffe auf die unterschiedlichen Komponenten geschehen über die Kommunikationsebene. Sie ist verantwortlich für die Zugriffe auf die DAO und das Organisieren der einzelnen Komponenten in dem Gesamtsystem. Die Kommunikationsschicht stellt Schnittstellen zu den einzelnen Komponenten zur Verfügung. Die Komponenten können ohne Neustart oder Neukompilierung in das System eingebunden oder entfernt werden. Die Komponentenschicht besteht aus vielen einzelnen Komponenten/Anwendungen, die vollständig unabhängig von einander arbeiten. Die einzelnen Anwendung haben kein Wissen über andere Teile des Systems und werden durch die Kommunikationsschicht geleitet. Die einzelnen Teilbereiche einer Komponente in Abb. 4.19 stellen die einzelnen Bestandteile der Anwendung bzw. Filters dar. Diese einzelnen Filter, sind mit Hilfe des Model View Controller Pattern, aus Gamma u. a. (1995), konzipiert und bestehen wieder aus Teilkomponenten. Die in der Abbildung dargestellten Adapter sind die jeweiligen Interfaces, mit denen die Komponente sich an der Kommunikationsschicht anmeldet. Ohne diese Schnittstelle ist die Komponente eine einzelne, unabhängig von dem System funktionierende, Anwendung.

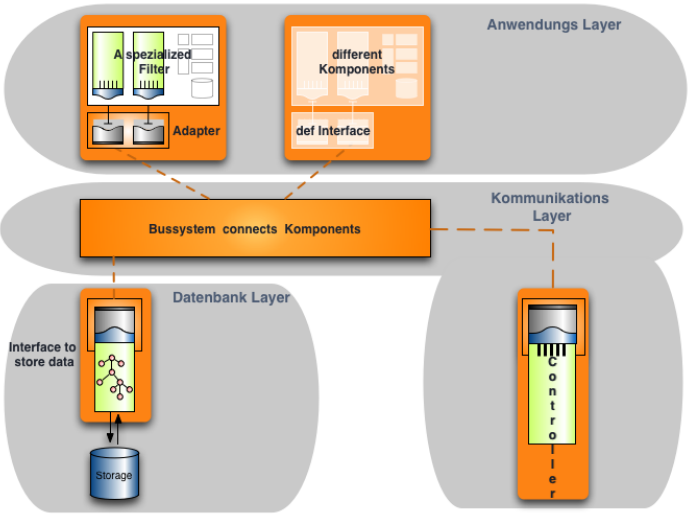

Abbildung 4.19: Überblick über die Gesamtarchitektur

4.7.1 Konkrete Architektur

Die geschilderte Architektur wird wie in Abb. 4.20 umgesetzt. Die einzelnen Bestandteil wer-

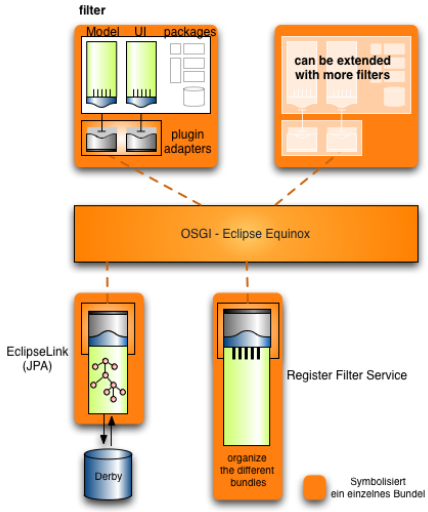

Abbildung 4.20: Umsetzung der abstrakten Architektur in einer Konkreten

den kurz vorgestellt, für eine genauere Betrachtung der einzelnen Komponenten wird auf Kapitel 5 verwiesen.

OSGI - Eclipse Equinox

Das Kommunikations-Layer wird, wie in Kapitel 4.6 beschrieben, mit Hilfe von OSGI realisiert. Als Plattform wurde Equinox von Eclipse gewählt, weil im Unternehmen Eclipse die Standardentwicklungsumgebung ist. Es stellt einen Kommunikationsbus zur Verfügung und eine Service Registry (Abb. 4.20 der Service Register Filter), bei der die einzelnen Bundles mit deren Services angemeldet werden. Das OSGI Framework sichert die Kommunikation zwischen den einzelnen Bundles und macht die Vorgaben für die Kommunikation.

Anwendungen bzw. Bundles

Das System besteht aus verschiedenen Anwendungen, die als Bundles in dem System angemeldet werden. Eine Anwendung ist ein spezieller Filter für die verschiedenen Informationsquellen. Ein Anwendung besteht aus zwei Komponenten. Ein Filter soll immer mit Hilfe des Model-View-Controller Pattern (MVC, siehe Gamma u. a. (1995)) realisiert werden.

- Eine Komponente ist jeweils das Modell und der Controller. Die beiden Bestandteile wurden zusammengefasst um den Aufwand innerhalb des Bundles zu verringern. Das Modell ist auf die jeweilige Informationsquelle angepasst. Es extrahiert die vorgegebenen Daten, bereitet diese auf und schreibt diese in die Datenbank (s. u. Kapitel 4.7.1).

- Die zweite Komponente ist die graphische Oberfläche, mit der die einzelnen Filter konfiguriert werden. Ein Beispiel wäre die Pfadangabe für eine Volltextsuche auf einem Server. Ein weiteres Beispiel ist die Service Registry. Es wird angezeigt welche Bundles bzw. Filter im System angemeldet und verfügbar sind.

Die Datenbank

Die Datenbank soll zur Implementierung eines Prototypen genutzt werden und sollte besonders einfach zu installieren sowie veränderbar sein. Es ist bei der ersten Implementierung nicht gefordert eine produktive Umgebung zu schaffen. Die Datenbank soll sich einfach verändern und gut in die Umgebung einbinden lassen. Apache Derby erfüllt diese Eigenschaften und hat weitere Eigenschaften, durch die sie gut für eine prototypische Implementierung geeignet ist:

- Derby ist sehr klein (ca. 2MB) als Basis System und mit enthaltenen JDBC Treibern.
- Derby erfüllt die Standards von Java, JDBC und SQL.
- Derby kann als Bibliothek in die Anwendung als Einzelplatz direkt oder im Server Modus eingebunden werden.
- Derby ist einfach zu installieren und einfach zu bedienen.

Apache ist ein Apache DB Teilprojekt. Es ist eine relationale Datenbank, die komplett in Java entwickelt wurde und unter der Apache License verfügbar ist.

O/R Mapper

Der Zugriff auf die Datenbank soll nicht direkt erfolgen, sondern über eine Zwischenschicht, die eine Trennung zwischen der Anwendungsschicht und dem relationalen Datenbankmodell vornimmt. Es gibt dafür die Spezifikation von Sun: Java Persistence API. Eine mögliche Implementierung lässt sich mit EclipseLink realisieren. Eclipselink ist der Zusammenschluss mehrerer Firmen unter der Führung von Oracle, die Ihr eigene JPA Entwicklung (Oracle), für eine Umsetzung der JPA Spezifikation unter der Eclipse Lizenz eingebracht haben.
Das Eclipse Persistence Services Project (EclipseLink) stellt ein komplettes Persistence Framework zur Verfügung. Es ist in Java entwickelt worden. Es liest und schreibt Objekte in jede mögliche Persistenz Umgebung: rationale Datenbanken, XML oder EIS Systeme (Enterprise Information Systems).
Die Entwicklung findet anhand der führenden Spezifikationen für Persistenzen statt: die Java Persistence API (JPA) für relationale Datenbanken, Java Architecture for XML Binding (JAXB), J2EE Connector Architekture (JCA) für EIS, ältere System in diesem Kontext und Service Data Objects (SDO).

Die Nutzung der Eclipselink Umgebung als eigenständige Komponente ermöglicht den Zugriff und das Arbeiten mit einer relationalen Datenbank, ohne die Schnittstelle selber spezifizieren oder implementieren zu müssen. Die komplett in Java geschriebene Implementierung funktioniert ohne Einschränkungen mit einer Derby Datenbank. Eclipselink verfügt über verschiedene Bibliotheken zur Einbindung in eine Anwendung, als Java Bibliothek oder als OSGI Bundle.

4.8 Zusammenfassung

In diesem Kapitel wurde ein Konzept zum Erstellen von Personen bezogenen Profilen entwickelt. Der Grundgedanke wurde in in Anlehnung an das Datawarehouseskonzept beschrieben. Daraus wurde ein Konzept skizziert. Es wurde beschrieben, woher die Informationen für ein Profil bezogen werden sollen. Die einzelnen Methoden zur Informationsgewinnung wurden detailliert beschrieben. Die Datenspeicherung und deren Datenbankmodell sind beschrieben worden. Es wurde erläutert, wie dieses ERM entwickelt wurde und wie aus diesem Modell weitere Informationen gewonnen werden können. Das ERM-Modell dient als Grundgerüst und ist nicht vollständig wiedergegeben worden. Das Modell ist an dieser Stelle unvollständig, weil nicht genau geklärt werden kann, wie die Daten aussehen und wie die Verarbeitung genau gestaltet sein wird. Im Anschluss wurden die allgemeinen Konstruktionen und Algorithmen aufgefürt, die zur Informationsgewinnung in den Filtern und in der Datenbank dienen sollen. Es folgten kurze Erklärungen und Begründungen zu den einzelnen Vorgehensweisen. Im nächsten Abschnitt wurde eine Betrachtung eines Profiles und die Definition im Kontext zu dem Konzept formuliert. Es wurden die Anforderungen aus der Analyse (Kapitel 2) zur Beschreibung der einzelnen Elemente für ein Profil erklärt. Die Beschreibung der Komponenten des Entwurfes sind in einem Konzept zur Verbindung der einzelnen Bereiche beschrieben worden. Es sind einzelne Konzepte diskutiert worden und das Framework OSGI ausgewählt worden. Nach dem Konzept und dem Designentwurf ist abschliessend eine Architektur vorgestellt worden.
Die beschriebene Idee zu diesem Konzept besteht aus verschiedenen Komponenten, die in Kapitel 3 in verschiedenen Projekten dargelegt worden sind. Die einzelnen Arbeiten werden benutzt, um in dem Design zu einer komplexeren Architektur zusammengebunden zu werden. Die Komponenten sind sehr speziell und die Hauptaufgabe besteht darin, diese in einem einheitliche Entwurf zu verbinden. Das Design der Datenbank muss alle unterschiedlichen Daten in einem gemeinsamen großen Model speichern und zur Verfügung stellen. Es muss in iterativen Schritten untersucht werden, wie hoch der Gehalt an Informationen in den einzelnen Daten ist. Gegebenenfalls muss dann eine Verfeinerung der Komponenten oder des Datenbankmodelles vorgenommen werden. Daher ist es wichtig, die Architektur so flexibel zu halten, dass die einzelnen Komponenten immer wieder ausgetauscht oder verändert werden können, ohne die gesamte Architektur zu behindern. Mit der Implementierung

im nächsten Kapitel soll untersucht werden, ob dieser Entwurf und die Architektur diesen Anforderungen gerecht werden.

Kapitel 5

Implementierung

Dieses Kapitel befasst sich mit der Implementierung eines Prototypen für die im Designkapitel vorgestellte Architektur. Es wird erläutert wie die Architektur umgesetzt, ein Basissystem aufgebaut und die einzelnen Komponenten aus dem Design implementiert und eingebunden werden.
Zum Abschluss folgt ein Fazit aus der Implementierung.

5.1 Architekturbeschreibung und deren Umsetzung

Die Basis der Architektur bildet OSGI (siehe Kapitel 4.6.1). Die einzelnen Filter werden als *Bundle* eingehängt und werden über eine vordefinierte Schnittstelle beim Framework angemeldet. Die Schnittstelle bietet den Vorteil, daß es möglich ist, beliebige Filter (auch im laufenden System) in die Applikation einzubinden. In 4.6.2 wurde die Service Registry als die zentrale Stelle des Frameworks beschrieben. In Abb. 5.1 ist die Service Registry für diese Anwendung dargestellt.
Die Service Registry ist eine Klasse, die mehrer Interfaces besitzt. Die Interfaces repräsentieren die einzelnen Services, die im System implementiert und angemeldet wurden. Es wird zwischen drei verschiedenen Typen von Servicen unterschieden.

Datenbankservice - Die verschiedenen Interfaces, die unter IDAO zusammen gefasst sind, beschreiben den Zugriff auf die Datenbank. Der Zugriff erfolgt nicht direkt auf die Datenbank, sondern mit Hilfe eines O/R Mappers (mit Hilfe von Eclipselink JPA).

IDAO - In IDAO sind die Services(Methoden), die mit `getConfigViewID` und `getFilterTyp` den Service beschreiben und mit `startTransaction` und `closeTransaction` den Datenbankzugriff initiieren oder beenden.

IDAOKeyphraseEx - schreibt mit `save(KeyphraseExtractionContainer)` die Informationen aus einem Keyphrase Bean in die Datenbank.

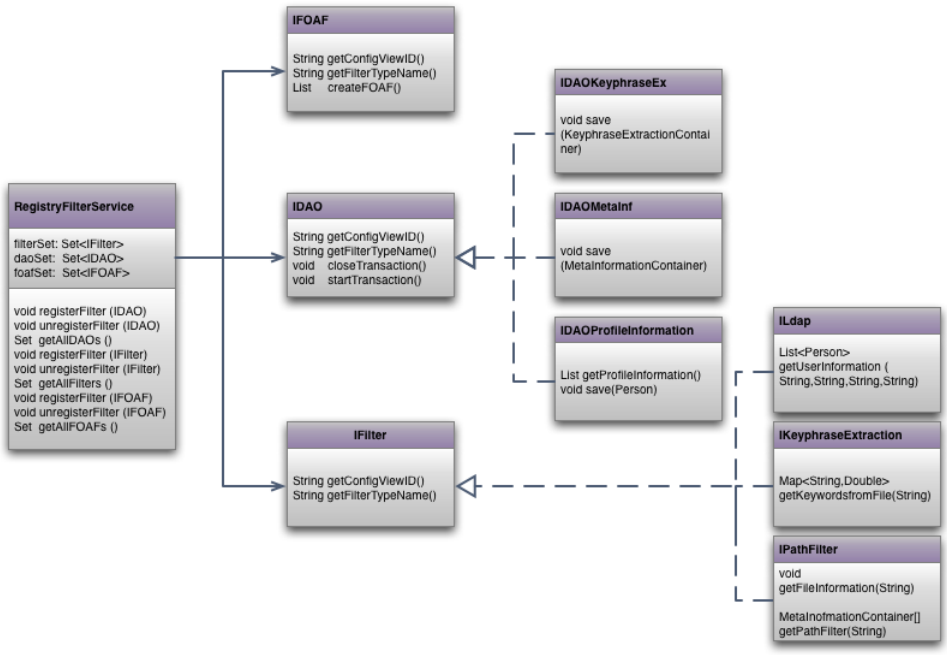

Abbildung 5.1: Die Service Registry mit den verschiedenen Services, die als Interface dargestellt und implementiert sind.

IDAOInf - speichert die gewonnenen Metadaten mit `save(MetaInformationContainer)` in die Datenbank.

IDAOProfileInformation - speichert mit `save(Person)` in die Datenbank und liest mit `getProfileInformation` die Personen aus.

IFilter - IFilter fasst die Filter unter sich zusammen. An dieser Stelle können Erweiterungen für die Informationsbeschaffung eingebunden werden. Man erbt von dem IFilter und bietet sein Service als IFilter-Service an. Soll ein Filter ausgetauscht werden, muss nur dafür gesorgt werden, dass das Interface weiter implementiert wird. Komponenten die den Service nutzen, bekommen diese Veränderung nicht mit, da diese nur den Service nutzen.

ILdap - mit `getUserInformation` wird ein LDAP Verzeichnis abgefragt, in dem alle Mitarbeiter der Firma mit Ihren Informationen abgelegt sind. Die Parameter können dabei sein: Name, E-Mail, Abteilung und Telefonnummer.

IKeyphraseExtraction - Mit Hilfe von `getKeywordsFromFile` wird dem Information Retrieval System ein Pfad übergeben, in dem die einzelnen Dateien analysiert werden sollen. Als Ergebnis bekommt man eine Liste von Begriffen, die nach ihrer Häufigkeit gegliedert sind (maximale Anzahl von Begriffen ist 25).

IPathFilter - Es wird die Apache Tika Libary benutzt, um die Metainformationen sowie eine kurz Beschreibung aus verschiedenen Dokumententypen auszulesen. Es wird ähnlich wie bei IKeyphraseExtraction ein Dateipfad mit `getFileInformation` angegeben, von dem aus die Komponente die Dateien analysiert.

IFOAF - Dieser Filter generiert die Profile mit den Informationen aus der Datenbank. Mit `createFOAF` wird der Prozess angestossen, der mit Hilfe von der Bibliothek Jena die Profile in Form von FOAF anlegt. Der Service `getFOAF(String uid)` ist nicht implementiert. Die Methode soll ein bestimmtes Profil zurück liefern.

Die Interfaces lassen sich beliebig erweitern. Es muss nur dafür gesorgt werden, dass die Services in der Klasse RegistryFilterService angebunden werden. Ein Beispiel wäre eine andere Ausgabe der Profile. Man kann z. B. VCards benutzen an Stelle von FOAF. Es wird ein neues Interface definiert `IVCard` mit dem benötigten Methoden wie `creatVCard` und man entwickelt dann eine Komponente, die diese Generierung mit den Interfacemethoden implementiert.
Im nächsten Abschnitt wird beschrieben, wie die verschiedenen Komponenten im System implementiert wurden und wie die Einbindung an die Service Registry abläuft.

5.2 Beschreibung der Implementierung der einzelnen Komponenten

Komponenten beschreiben Bestandteile des Systemes, die dessen Funktionalität erweitern. Komponenten implementieren die einzelnen Services wie sie in der RegistryFilterService Klasse definiert wurden. Die Komponenten sind eigenständige Anwendungen auf einer Two-Tier Architektur (siehe Abb. 5.2).

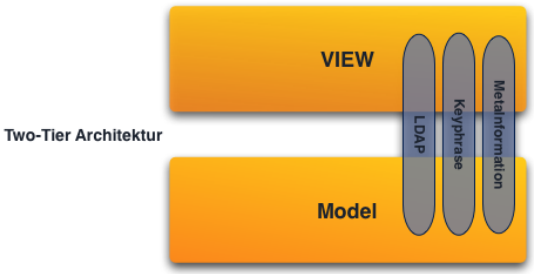

Abbildung 5.2: Darstellung der Bundles in einer two-tier Architektur

Eine Komponente besteht aus zwei Bundles: die Anwendung selbst und ein graphisches Bundle. Das graphische Bundle dient zur Darstellung des Status und der Konfiguration durch

den Anwender. Die graphische Benutzerschnittstelle ist abhängig von dem Modell (dem in der Implementierung als *core-Bundle* bezeichneten Paket). Die beiden Bundles beschreiben eine Komponente.

An- und Abmelden an der Service Registry

Ein Bundle selbst besteht aus einem oder mehreren Java Paketen. Es besitzt eine *Activator* Klasse und die Komponente implementiert jeweils das Interface, welches wiederum den geforderten Service implementiert. Die Bundles werden mit Hilfe des Registry Services und einer Activator Klasse in dem System an- bzw. abgemeldet. Ein schematischer Ablauf ist in Abb. 5.3 dargestellt.

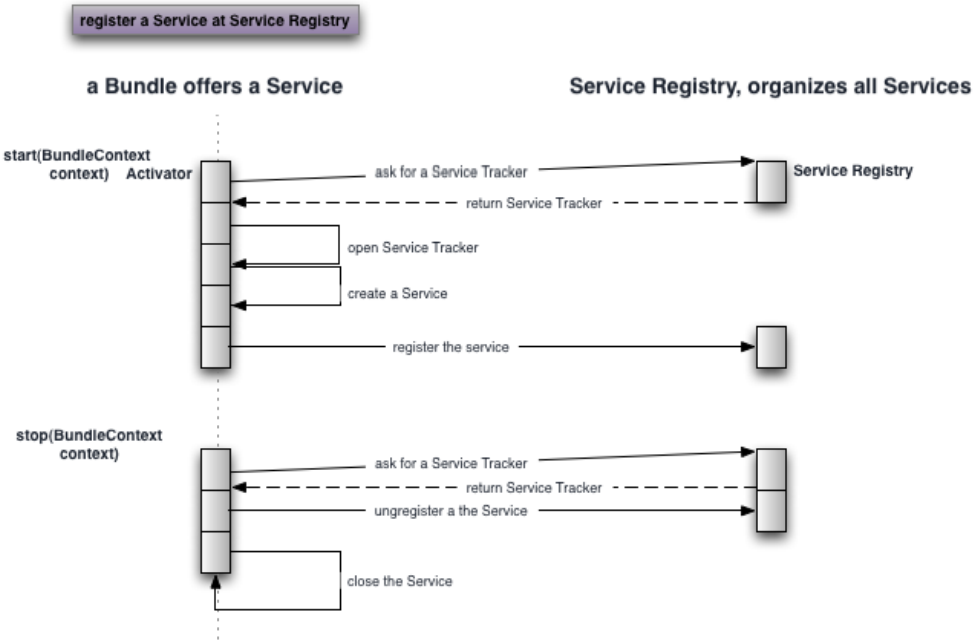

Abbildung 5.3: Schematischer Ablauf zur Registrierung eines Services an der Service Registry

Soll ein Bundle an der Service Registry einen Service anmelden, wird beim Start des Bundles in der Aktivator Klasse die Methode `start(BundleContext context)` aufgerufen (Die Aktivator Klasse wurde in Kapitel 4.6.3 beschrieben). Das BundleContext Objekt wird immer benötigt um Services zu registrieren und neue Bundles im Framework zu installieren Alliance (2007). Es wird bei der Service Registry nach einem Service Tracker angefragt. Der Service Tracker vereinfacht das Benutzen der Service Registry. Der Service Tracker übernimmt alle Einzelheiten für das An- und Abmelden von Services beim Eintreten eines Service Events, Alliance (2007). Mit Hilfe des Service Tracker wird unser Service in die Service

Registry aufgenommen. Es wird eine Instanz unserer Anwendung, die den Service anbietet, generiert. Diese Instanz wird an den Service Tracker übergeben. Mit Hilfe des Service Trackers wird das Bundle in der Service Registry aufgenommen.
Beim Beenden bzw. bei Stoppen eines Bundles wird im Aktivator die Methode `stop(BundleContext context)` aufgerufen (siehe 5.3 unterer Teil). Es wird wie beim Start ein Service Tracker angefordert. Am Service Tracker wird die beim Start angegebene Instanz abgemeldet und der Service Tracker geschlossen. Im letzten Schritt wird die Instanz gelöscht und der Aktivator geschlossen.

Nutzen eines Services

Sind Services angemeldet kann ein anderes Bundle auf diese zugreifen und diese nutzen. Der Ablauf, wie ein Service benutzt werden kann, ist in Abb. 5.4 dargestellt.

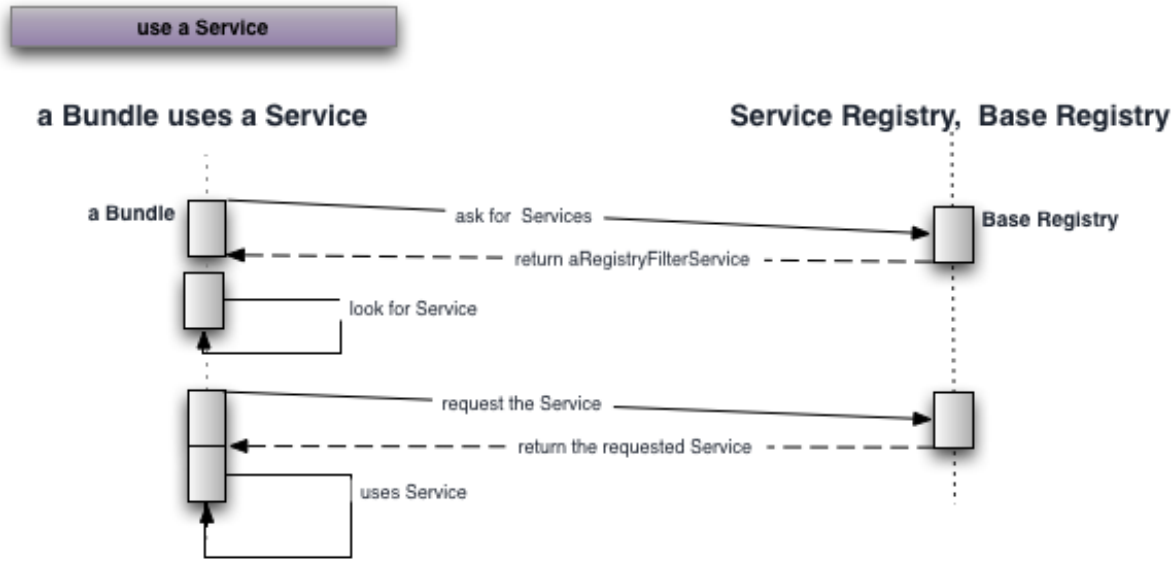

Abbildung 5.4: Schematischer Ablauf bei der Benutzung eines Services

Ein Bundle fragt bei der Klasse Base Registry (Das Bundle der Service Registry) nach der Service Registry, die Service Registry wird an das Bundle zurück geliefert (siehe Abb. 5.1). Das Bundle sucht sich den passenden Service und ruft mit dessen Signatur den Service auf und verwendet ihn im eigenen Bundle.
Die hier beschrieben Abläufe und Beschreibungen gelten für alle Implementierungen in dieser Arbeit und finden sich so im beigelegten Sourcecode wieder. Im folgenden werden die Einzelnen Filter bzw. Komponenten vorgestellt.

5.3 Fazit

Auf Basis der bisher beschriebenen Techniken und Bibliotheken wurde ein erster Prototyp entwickelt. Die in den vorherigen Kapiteln vorgestellten Komponenten wurden mit Hilfe der beschrieben Bibliotheken implementiert und mit einer graphischen Oberfäche versehen, (Abb. 5.5) um die Komponenten zu konfigurieren. Die OSGI Schnittstellen wurden wie im voherigen Abschnitt 5.1 beschrieben implementiert. Die Abb. 5.5 zeigt die einzelnen Komponenten links im Fenster. Die einzelnen Komponenten besitzen jeweils die Services, die mit OSGI verwaltet werden. Die einzelnen Filter speichern die extrahierten Daten jeweils in einem Bean. Über das Bean wird mit dem Service zum Speichern das jeweilige Objekt in die Derby Datenbank geschrieben. Der Service zum Speichern der Daten wird von dem EclipseLink Plugin zur Verfügung gestellt. Die einzelnen gespeicherten Objekte besitzen jeweils eine n-m Beziehung zu der Person die als Author oder als Dateibesitzer ermittelt wurden.

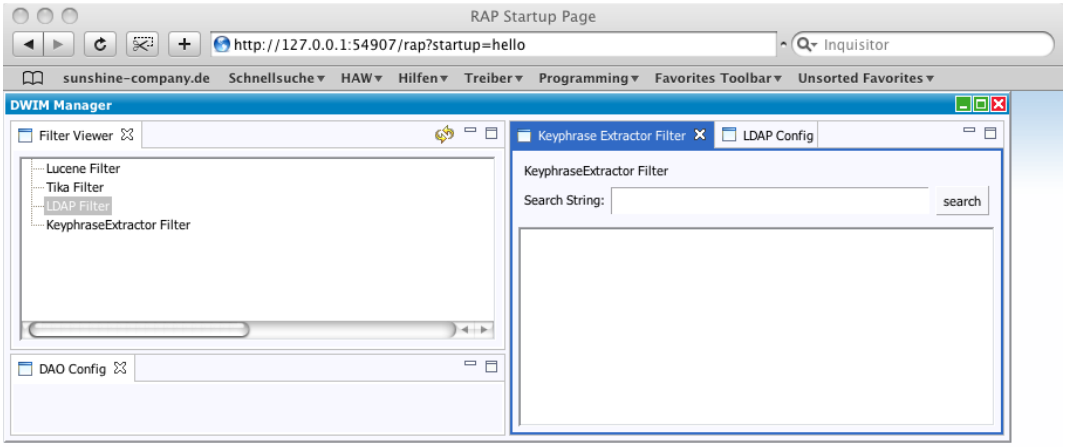

Abbildung 5.5: Eine graphische Weboberfläche zur Keyphrase Extraction

In diesem Prototyp wurde die im Designkapitel 4 vorgestellte Architektur umgesetzt. Die einzelnen Komponenten agieren unabhängig von einander und können während der Laufzeit ausgetauscht oder verändert werden. Die Implementierung der Schnittstelle zur Datenbank war mit großem Aufwand verbunden. Der bereitgestellte JPA O/R Mapper aus dem EclipseLink Projekt ist noch nicht vollständig ausgereift und ließ verschiedenste Datenbankkonfigurationen nicht zu. Es ist ein erheblicher Aufwand nötig gewesen, um die notwendigen Einstellungen bereitzustellen. Um den Aufwand zu reduzieren, könnte es vielleicht besser sein ein eigenes OSGI Bundle für den O/R Mapper zu implementieren. Ein Erfahrungswert bei dem Testen des Prototypen war die Performance der einzelnen Anwendungen. Beim Scannen und Extrahieren von einzelnen kleinen Testverzeichnissen und Dokumenten war die Geschwindigkeit des Vorganges noch ausreichend. Beim Einsatz auf einem ganzen Abteilungsverzeichnis (1̃0 GB) brach die Anwendung häufig ab oder blieb im laufenden Betrieb

"stehen". Es muss bei einem produktiven Einsatz mehr Aufwand für die Geschwindigkeit und die Stabilität der einzelnen Komponenten eingerechnet werden. Vom Vorteil bei dem Prototypen erwies sich OSGI als Plattform. Trotz des Ausfalles einzelner Komponenten blieb das Gesamtsystem unbeeinflusst von diesen Störfällen.
Die einzelnen Filter erfüllen ihre Aufgabe. Das Ergebnis ist dabei aber unterschiedlich erfolgreich. Es stellte sich bei der Verarbeitung der verschieden Informationsquellen heraus, dass die Informationen nicht immer in der Form vorhanden waren, wie bei der Entwicklung argumentiert. Viele Informationsquellen enthielten unvollständige oder gar keine Informationen.
Das Extrahieren von Personen aus dem LDAP-Verzeichnis funktioniert problemlos.
Betrachtet man die Komponente zur Extraktion von Metainformationen ist das Ergebnis unbefriedigend. Die eingesetzte Bibliothek arbeitet auf einem grossen Fileserver zu langsam. Die Metainformationen sind bei den internen Dokumenten nicht zu verwenden, da weder Informationen wie Autor oder Title bekannt gegeben werden noch die Datumsangabe oder weitere Angaben möglich waren.
Die Keyphrase Extraction funktioniert angewendet auf Textdokumente sehr gut. Die Komponente arbeitet aber sehr langsam und benötigt für durchschnittliche Dokumente mehrere Sekunden. An dieser Stelle muss die Komponente performanter werden, um diese auf mehrere tausend Dokumente anzusetzen.
Es ließ sich aus den verschiedenen Komponenten kein direktes Spezialwissen für die einzelnen Benutzer ableiten. Eine Unterscheidung zwischen zwei Benutzern hinsichtlich der Qualität ihres Fachwissens war bei den bisherigen Implementierungen nicht möglich. Die Entwicklung von Komponenten zur Auswertung von Teilbereichen des Intranets ist aufwendig und muss häufig wiederholt werden. Es muss klar definiert sein, wonach gesucht wird. Nimmt man als Beispiel ein Dateiverzeichnis mit Java Quellcode Dateien, sagt es nichts darüber aus, welche Java -Techniken (z. B. JSF, OSGI, SOA oder Spring) verwandt wurden und ob der Besitzer als Experte gelten kann. Die Qualität des Quelltextes kann mit den vorgestellten Methoden nicht analysiert und bewertet werden. Eine Aussage die getätigt werden kann, ist: Der Mitarbeiter ist ein Softwareentwickler und arbeitet an folgenden Projekt.
Abschliessend kann die Implementierung positiv bewertet werden. Die Basisstruktur des Prototyps konnte mit geringem Aufwand implementiert werden. Die Anforderungen wurden aus konzeptioneller und technischer Sicht umgesetzt, mit der Einschränkung des oben erwähnten O/R Mappers. OSGI erwies sich als ein einfach zu implementierendes Framework. Es ist stabil und bietet eine hohe Skalierbarkeit. Der Ansatz verschiedene "kleine" Anwendungen zu bauen, die auf bestehenden, vielfach getesteten und bereits genutzten Bibliotheken aufbauen, war ein Erfolg. Die fachliche Umsetzung der einzelnen Filter dagegen muss genauer spezifiziert werden und besser an die einzelnen Datenquellen angepasst werden. Für einen produktiven Einsatz müssen die verschieden Problematiken wie Performance und Stabilität bei den einzelnen Filtern und der Datenbankschnittstelle berücksichtigt werden.

Kapitel 6

Zusammenfassung

In dieser Arbeit wurde untersucht wie Benutzerprofile automatische generierten werden können. Das Ziel war eine Verbesserung der Suche in einem Intranet. Es sollten Personenprofile erstellt werden an Hand denen man Aussagen über die Fähigkeiten des Benutzers treffen kann. Zusätzlich sollte ein Social Network erarbeitet werden, mit dessen Hilfe die jeweiligen Experten in Kontext zu ihrem Umfeld gestellt wurden.
Es wurde als erstes beschrieben, was ein Social Network ist und wie es verwendet wird. An Hand von einem Versicherungsunternehmen wurde beschrieben, was von einem Social Network und deren Benutzerinformationen für das eigene Intranet erwartet wird. In der Analyse des Unternehmens wurde gezeigt, was die Schwierigkeiten sind. Es wurde gezeigt, daß es verschiedene Faktoren, wie die Mitarbeiter, die IT-Struktur und die Aufgabenbereiche gibt, die berücksichtigt werden mussten. Es wurden die verschiedenen Anforderung dieser Faktoren herausgearbeitet und bei dem folgendem Konzept mit berücksichtigt. Mit diesen Informationen wurden bisherige Arbeiten und Ideen verglichen und auf deren Verwendbarkeit in dieser Aufgabenstellung untersucht. Die Schwierigkeit beim Entwickeln des Konzeptes war herauszufinden, woher die Informationen für eine Personenbeschreibung kommen können. Es wurden verschiedene Möglichkeiten erläutert und deren technische Umsetzung diskutiert. Mit dem Wissen über Datenquellen für die benötigten Informationen, wurde eine Architektur entwickelt. Die Architektur musste verschiedene Aufgaben erfüllen: Es sollte ein offenes System sein, bei dem Datenquellen hinzugefügt, verändert oder entfernt werden können, ohne die anderen Bestandteile des Systems zu beeinflussen oder zu verändern. Eine Schwierigkeit, die sich daraus ergibt unterschiedliche Informationsquellen zu benutzen, die sich verändern können, ist die Persistenzschicht. Diese sollte die gewonnen Daten speichern und zusätzlich eine einheitliche Struktur vorhalten. Es musste möglich sein, durch eine KDD Prozess weitere Informationen aus den Daten gewinnen zu können. Die Datenbank wurde als ein Cache implementiert, der bei jedem neuen Prozess der Informationsgewinnung neu aufgebaut wird. Aus den gewonnen Informationen wurden danach Profile erstellt, die jeweils eine Person beschreiben. Das Format dieser Profile zeigt zusätzlich die Beziehungen zwischen den

einzelnen Personen. Mit dessen Hilfe ein Social Network innerhalb des Intranets aufgebaut werden kann. Diese Architektur und deren Komponenten wurden in einem Prototypen implementiert. Die einzelnen Komponenten wurden als eigenständige Anwendung betrachtet, die Schnittstellen zur Verfügung stellen durch die diese mit anderen Komponenten kommunizieren konnten. Das Framework für diese Kommunikation wurde von dem OSGI Framework zur Verfügung gestellt. Es wurden mehrer Komponenten entwickelt: eine Persistenzschicht (O/R Mapper und Datenbank), eine Komponente zum Extrahieren der Mitarbeiterinformationen aus der Benutzerverwaltung des Intranets, eine Komponenten zum Extrahieren von Beschreibungen aus einzelnen Dokumenten, eine weitere zum Sammeln von Metainformationen und eine Komponente zum Generieren von Benutzerprofilen aus den ausgewerteten Informationen.

Diese Arbeit zeigt, dass es möglich ist automatisch ein Benutzerprofil zu generieren mit Hilfe der bestehenden Informationen aus dem Intranet. Die Vorgehensweise sich nicht, wie in verschiedenen vorherigen Arbeiten (siehe Kapitel 3), auf nur auf einen bestimmten Bereich konzentriert, sondern für jede mögliche Datenquelle eine eigene Anwendung entwickelt, die Informationen aus den Bereich hat sich bewährt extrahiert. Durch dieses Vorgehen behält man eine bessere Kontrolle über die Daten die untersucht werden. Ein weiterer Vorteil ist, daß die Anwendungen speziell auf die Datenquellen angepasst werden können. Die komplexen Strukturen eines Intranets und die sozialen Netzwerke ließen sich in einfache Teilbereiche herunterbrechen und in einer übersichtlichen Umgebung zu einem stabilen Fundament entwickeln. Die Skalierbarkeit der Architektur und des OSGI Frameworks erlauben es einfach, unterschiedliche Komponenten hinzuzufügen, bestehende auszutauschen oder zu verbessern.

Vorgestellte Arbeiten aus Data Mining oder Semantic Informations ließen sich in das System integrieren. Bei dem aktuellen Stand vom Data Mining ist die Prozedur zum Aufarbeiten von Informationen sehr weitentwickelt und es lassen sich diese komplexen Anwendungen und Algorithmen aus vorhergehenden Arbeiten, als eigene Komponente, direkt verwenden. Es lassen sich unterschiedliche Data Mining Verfahren “einfach” implementieren und auf deren Effizienz untersuchen. Die unterschiedlichen Komponenten können mit Erweiterungen verbessert werden und rudimentäre Komponenten können in einzelnen kleinen Schritten weiter entwickelt und angepasst werden.

Die entwickelten Komponenten sind Beispiele für mögliche Implementationen. In einer Weiterentwicklung sollte der Schwerpunkt auf einer genaueren Analyse der einzelnen Filter liegen, um bessere Ergebnisse zu erzielen. Ein Schwerpunkt sollte die Verbesserung des KDD Prozesses sein. Es gibt verschiedene Möglichkeiten wie in Kapitel 4.4 erwähnt. Einige dieser Verfahren wurden in Kapitel 4.4 vorgestellt (siehe dazu Ehrlich u. a. (2007), Li u. a. (2005) und Makrehchi und Kamel (2006)). Die bereitstehenden Bibliotheken erlauben eine Vielzahl von Verbesserungen für die Suche. Der Schwerpunkt sollte aber auf genauen Untersuchung der Datenquellen liegen.

Dieses Ausarbeitung stellt eine funktionierende Implementierung zum Aufbau eines auto-

matisch generierten Benutzerprofiles dar. Die eingesetzte Architektur aufsetzend auf dem OSGI Framework, ist skalierbar, stabil und stellt, wie im Design gefordert, eine offene Basis, die sich beliebig erweitern lässt und auf verschieden Plattformen verfügbar ist. Die Implementierung war durch das einfache Einbinden von bereits bestehenden Bibliotheken und Anwendungen mit geringen eigene Aufwand möglich. Die Plattform lässt sich ohne Probleme in das bestehende Intranet einbinden. Die verschiedenen bereitstehenden Werkzeuge helfen bei der Extrahierung der Informationen, lassen aber keine direkte Interpretation zu. Das bedeutet, dass beim Entwickeln von Filtern überlegt werden sollte, was genau gesucht und erwartet wird.

6.1 Ausblick

Die beschriebenen Vorgehensweisen sind technischer Art. Die Akzeptanz bei den Mitarbeitern sollte dabei ein wesentlicher Bestandteil sein. Eine wesentliche Aufgabe sollte es sein, ein möglichst transparentes Vorgehen zu entwickeln und die Mitarbeiter einzubinden. Das System muss Vertrauen schaffen, dass es sich dabei nicht um eine Kontrolle, sondern um eine Hilfe für die Mitarbeiter handelt. Der Datenschutz spielt bei der gesamten Entwicklung eine wichtige Rolle. Es dürfen keine Information verwendet werden, die unter den Datenschutz fallen oder bei denen ein Mitarbeiter Einwände erhebt. Es wären sicherlich wesentlich mehr Beziehungen zwischen Personen oder Informationen zu den einzelnen Mitarbeitern zu finden, würde deren Internetverhalten mit ausgewertet werden. Durch die Nutzung dieser Daten würde die Akzeptanz der Mitarbeiter verloren gehen und die Benutzung eines solchen Systemes eher verhindert als verbessert werden. In dieser Ausarbeitung wurden Verfahren benutzt, die bestehendes Wissen aus dem Intranet extrahieren. Es wird nur bekanntes Wissen extrahiert. Wissen, das der Mitarbeiter aus seiner Vergangenheit vor der Anstellung in dem Unternehmen besitzt, ist nicht zugänglich. Weitere Fertigkeiten, wie eine Fremdsprache, sind ebenso nicht abfragbar. Es sollte in der Zukunft überlegt werden, wie ein Mitarbeiter seine eigenes Profil erweitern und verbessern kann. Bei diesem Vorgehen könnte der Mitarbeiter ebenfalls sein Profil überprüfen und Fehler in dem generierten Profil ausbessern.

Anhang A

RDF - Ressource Description Framework

In dem Kapitel 4.5 wurde bereits über Resource Description Framework (RDF) und Semantic Web gesprochen. An dieser Stelle folgt eine genauere Erläuterung.
Semantic Web ist entstanden aus einer Idee von Tim Berneers-Lee. Das Internet ist für Computer zwar lesbar, aber nicht interpretierbar, mit der Idee vom Semantic Web soll eine Plattform geschaffen werden, die es erlaubt das Maschinen untereinander kommunizieren können und die Inhalte interpretierbar sind. Das Resource Description Framework ist einer der Techniken mit der die Anforderung umgesetzt werden soll [9]. RDF ist die Basis zum Auslesen von Metadaten zur Beschreibung von Daten zum automatischen Abarbeiten von Informationen. RDF wird zum Auffinden von Informationen in Datenbeständen benutzt, um Suchmaschinen bessere Ergebnisse zu liefern, bei deren Indizierung lassen sich zusätzlich Informationen mit RDF katalogisieren.
Eine RDF Beschreibung besteht aus einem Triple:

- Resources, ist eine Internetresource, was eine Webseite sein kann oder auch mehrere. Es besteht immer aus einer URI.
- Properties, beschreiben immer eine bestimmte Eigenschaft oder einen bestimmten Aspect der Ressources. Die Properties sind in der RDF Spezifikation festgelegt.
- Statements, die bestimmte Ressources mit den Property und dessen Value sind ein RDF Statement.

Man spricht dabei auch von Subjekt, Prädikat und Objekt, wie bei einem normalen Satzbau. Ein Beispiel dafür ist:

"Ora Lassila is the creator of the resource http://www.w3.org/Home/Lassila"

[9] `http://www.w3.org/TR/PR-rdf-syntax/`

Subject (Ressource)	http://www.w3.org/Home/Lassila
Predicate (Property)	Creator
Object (literal)	“Ora Lassila”

Eine textuelle Darstellung im Source Code sieht folgendermassen aus,

```
<rdf:RDF>
  <rdf:Description about="http://www.w3.org/Home/Lassila">
    <s:Creator>Ora Lassila</s:Creator>
  </rdf:Description>
</rdf:RDF>
```

Literaturverzeichnis

[Ackerman u. a. 1999] ACKERMAN, Mark S. ; CRANOR, Lorrie F. ; REAGLE, Joseph: Privacy in e-commerce: examining user scenarios and privacy preferences. In: *EC '99: Proceedings of the 1st ACM conference on Electronic commerce*. New York, NY, USA : ACM, 1999, S. 1–8. – ISBN 1-58113-176-3

[Ackerman u. a. 2002] ACKERMAN, Mark S. ; WULF, Volker ; PIPEK, Volkmar: *Sharing Expertise: Beyond Knowledge Management*. Cambridge, MA, USA : MIT Press, 2002. – ISBN 0262011956

[Aleman-Meza u. a. 2007] ALEMAN-MEZA, Boanerges ; HAKIMPOUR, Farshad ; BUDAK ARPINAR, I. ; SHETH, Amit P.: SwetoDblp ontology of Computer Science publications. In: *Web Semant.* 5 (2007), Nr. 3, S. 151–155. – ISSN 1570-8268

[Aleman-Meza u. a. 2008] ALEMAN-MEZA, Boanerges ; NAGARAJAN, Meenakshi ; DING, Li ; SHETH, Amit ; ARPINAR, I. B. ; JOSHI, Anupam ; FININ, Tim: Scalable semantic analytics on social networks for addressing the problem of conflict of interest detection. In: *ACM Trans. Web* 2 (2008), Nr. 1, S. 1–29. – ISSN 1559-1131

[Alliance 2007] ALLIANCE, Osgi: *OSGi Service Platform Core Specification 4*. Ca, US: , 2007

[Alliance 2009] ALLIANCE, OSGI: *Benefits of Using OSGi*. Feb 2009. – URL `http://www.osgi.org/About/WhyOSGi`

[Amazon] AMAZON: *Amazone*. – URL `http://amazon.de/`. – 2009.03.03

[Arthur und Vassilvitskii 2006] ARTHUR, David ; VASSILVITSKII, Sergei: How slow is the k-means method? In: *SCG '06: Proceedings of the twenty-second annual symposium on Computational geometry*. New York, NY, USA : ACM, 2006, S. 144–153. – ISBN 1-59593-340-9

[Balzert 1998] BALZERT, Helmut: *Lehrbuch der Software-Technik - Software Management, Software-Qualitätssicherung, Unternehmensmodellierung*. Spektrum Akademischer Verlag, 1998

[Barnes 1954] BARNES, J.: Class and Comittetees. In: *Human Relations* (1954)

[Beagle] BEAGLE: *Beagle Desktop Search*. – URL `http://beagle-project.org/Main_Page`. – 2009.03.03

[Beierle und Kern-Isberner 2000] BEIERLE, Christoph ; KERN-ISBERNER, Gabriele: *Methoden wissensbasierter Systeme - Grundlagen, Algorithmen, Anwendungen*. vieweg, 2000

[Borgatti und Cross 2003] BORGATTI, Stephen P. ; CROSS, Rob: A Relational View of Information Seeking and Learning in Social Networks. In: *Manage. Sci.* 49 (2003), Nr. 4, S. 432–445. – ISSN 0025-1909

[Budszuhn 2005] BUDSZUHN, Frank: *Subversion*. Galileo Computing, 2005

[Bugzilla] BUGZILLA, Mozilla: *Bugzilla is server software designed to help you manage software development*. – URL `http://www.bugzilla.org/`. – 2009.03.03

[Calishain und Dornfest 2003] CALISHAIN, Tara ; DORNFEST, Rael: *Google Hacks: 100 Industrial-Strength Tips and Tools*. Sebastopol, CA, USA : O'Reilly & Associates, Inc., 2003. – ISBN 0596004478

[Carmel u. a. 2003] CARMEL, David ; MAAREK, Yoelle S. ; MANDELBROD, Matan ; MASS, Yosi ; SOFFER, Aya: Searching XML documents via XML fragments. In: *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*. New York, NY, USA : ACM, 2003, S. 151–158. – ISBN 1-58113-646-3

[Chang und Hsu 2005] CHANG, Hsi-Cheng ; HSU, Chiun-Chieh: Using Topic Keyword Clusters for Automatic Document Clustering. In: *IEICE - Trans. Inf. Syst.* E88-D (2005), Nr. 8, S. 1852–1860. – ISSN 0916-8532

[for Computing Machinery] COMPUTING MACHINERY, Association for: *The ACM Digital Library*. – URL `http://portal.acm.org/dl.cfm?coll=portal&dl=ACM&CFID=15980938&CFTOKEN=94338736`

[Consortium 2008] CONSORTIUM, Internet M.: *vCard: Your Electronic Business Card*. 2008. – URL `http://www.imc.org/pdi/vcardoverview.html`

[Costa u. a. 2008] COSTA, Ricardo A. ; OLIVEIRA, Robson Y. S. ; SILVA, Edeilson M. ; MEIRA, Silvio R. L.: A.M.I.G.O.S: knowledge management and social networks. In: *SIGDOC '08: Proceedings of the 26th annual ACM international conference on Design of communication*. New York, NY, USA : ACM, 2008, S. 235–242. – ISBN 978-1-60558-083-8

[Cunningham 2005] CUNNINGHAM, H.: Information Extraction, Automatic. In: *Encyclopedia of Language and Linguistics, 2nd Edition* (2005)

[Dawson und Howes 1998] DAWSON, F. ; HOWES, T.: *vCard MIME Directory Profile*. 1998. – http://tools.ietf.org/html/rfc2426

[delicious] DELICIOUS: *delicious, social bookmarks*. – URL http://delicious.com/. – 2009.03.03

[Dumbill 2002] DUMBILL, Edd: *Representing vCard Objects in RDF/XML*. 2002. – http://www.ibm.com/developerworks/xml/library/x-foaf.html

[Durkheim 1977] DURKHEIM, Emile: *Über die Teilung der sozialen Arbeit*. 1977

[EclipseLink] ECLIPSELINK: *EclipseLink*. – URL http://www.eclipse.org/eclipselink/. – 2009.02.03

[Ehrlich u. a. 2007] EHRLICH, Kate ; LIN, Ching-Yung ; GRIFFITHS-FISHER, Vicky: Searching for experts in the enterprise: combining text and social network analysis. In: *GROUP '07: Proceedings of the 2007 international ACM conference on Supporting group work*. New York, NY, USA : ACM, 2007, S. 117–126. – ISBN 978-1-59593-845-9

[Ehrlich und Shami 2008] EHRLICH, Kate ; SHAMI, N. S.: Searching for expertise. In: *CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*. New York, NY, USA : ACM, 2008, S. 1093–1096. – ISBN 978-1-60558-011-1

[Enterprise] ENTERPRISE, Google: *Productivity Solutions for your Business*. – URL http://www.google.com/enterprise/. – 2009.03.03

[Farrell u. a. 2005] FARRELL, Stephen ; CAMPBELL, Christopher ; MYAGMAR, Suvda: Relescope: an experiment in accelerating relationships. In: *CHI '05: CHI '05 extended abstracts on Human factors in computing systems*. New York, NY, USA : ACM, 2005, S. 1363–1366. – ISBN d1-59593-002-7

[Farrell u. a. 2007] FARRELL, Stephen ; LAU, Tessa ; NUSSER, Stefan ; WILCOX, Eric ; MULLER, Michael: Socially augmenting employee profiles with people-tagging. In: *UIST '07: Proceedings of the 20th annual ACM symposium on User interface software and technology*. New York, NY, USA : ACM, 2007, S. 91–100. – ISBN 978-1-59593-679-2

[FAST] FAST, A MicrosoftÂ® S.: *FAST, A MicrosoftÂ® Subsidiary*. – URL http://www.fastsearch.com/. – 2009.03.03

[Feldman und Sherman 203] FELDMAN, S. ; SHERMAN, C.: The high cost of not finding information / IDC. April 203. – Forschungsbericht

[Frakes und Baeza-Yates 1992] FRAKES, William B. (Hrsg.) ; BAEZA-YATES, Ricardo (Hrsg.): *Information retrieval: data structures and algorithms*. Upper Saddle River, NJ, USA : Prentice-Hall, Inc., 1992. – ISBN 0-13-463837-9

[Friedman 2003] FRIEDMAN, Nat: Dashboard. In: *In Proceedings of the Linux Symposium Conference on the Linux kernel and major OS infrastructure and research projects*, 2003, S. 1 – 10

[Gamma u. a. 1995] GAMMA, Erich ; HELM, Richard ; JOHNSON, Ralph ; VLISSIDES, John: *Design patterns: elements of reusable object-oriented software*. Boston, MA, USA : Addison-Wesley Longman Publishing Co., Inc., 1995. – ISBN 0-201-63361-2

[Geoffray u. a. 2008] GEOFFRAY, Nicolas ; THOMAS, Gaël ; FOLLIOT, Bertil ; CLÉMENT, Charles: Towards a new isolation abstraction for OSGi. In: *IIES '08: Proceedings of the 1st workshop on Isolation and integration in embedded systems*. New York, NY, USA : ACM, 2008, S. 41–45. – ISBN 978-1-60558-126-2

[Granovetter 1973] GRANOVETTER, Mark: The strength of weak ties. In: *American Journal of Sociology* (1973)

[H. Mucksch 2000] H. MUCKSCH, Behme: *Das Data Warehouse-Konzept*. Gabler, 2000

[Hasan und Pfaff 2006] HASAN, Helen ; PFAFF, Charmaine C.: The Wiki: an environment to revolutionise employees' interaction with corporate knowledge. In: *OZCHI '06: Proceedings of the 18th Australia conference on Computer-Human Interaction*. New York, NY, USA : ACM, 2006, S. 377–380. – ISBN 1-59593-545-2

[Hotlist] HOTLIST, ICQ: *Hot list, share your style*. – URL `http://hotlists.hotornot.com/Style-g35027955-Icq.html`. – 2009.03.03

[Iannella 2001] IANNELLA, Renato: *Representing vCard Objects in RDF/XML*. 2001. – `http://www.w3.org/TR/vcard-rdf`

[Iofciu u. a. 2005] IOFCIU, Tereza ; KOHLUETTER, Christian ; PAIU, Raluca ; NEJDL, Wolfgang: Keywords and RDF Fragments: Integrating Metadata and Full-Text Search in Beagle++. In: *Workshop on The Semantic Desktop - Next Generation Personal Information Management and Collaboration Infrastructure at the International Semantic Web Conference*, 2005, S. 1–10

[J. Chen C.-H. Chen-Ritzo C. A. Chess und Topol 2008] J. CHEN C.-H. CHEN-RITZO C. A. CHESS, K. Ehrlich M. Eleftheriou M. E. Helander C. Lasser S. C. McAllister S. A. Medeiros K. Penchuk J. L. Snowdon M. L. S. ; TOPOL, A.: Enhanced Professional Networking and its Impact on Personal Development and Business Success / IBM Watson Research Center. IBM Watson Research Center, May 2008. – Forschungsbericht. – URL `http://domino.watson.ibm.com/cambridge/research.nsf/58bac2a2a6b05a1285256b30005b3953/705fba7f6b06c8ac852574c00057e23c?OpenDocument`. IBM Watson Research Center

[Jain u. a. 1999] JAIN, A. K. ; MURTY, M. N. ; FLYNN, P. J.: Data clustering: a review. In: *ACM Comput. Surv.* 31 (1999), Nr. 31, S. 264–323. – ISSN 0360-0300

[Java] JAVA, Sun: *Package javax.naming.ldap*. – URL `http://java.sun.com/javase/6/docs/api/javax/naming/ldap/package-summary.html`. – 2009.03.03

[JLDAP] JLDAP, Novell: *Java LDAP*. – URL `http://www.openldap.org/jldap/`. – 2009.03.03

[J.S. 1981] J.S., House: *Work Stress and social support*. Reading, Mass : Addison-Wesley, 1981

[J.S. Rellermeyer und Roscoe 2007] J.S. RELLERMEYER, G. A. ; ROSCOE, T.: R-OSGi: Distributed applications through software modularization. In: *Lecture Notes in Computer Science* Bd. 4834, Springer, 2007, S. 1 – 20

[Knuth 1998] KNUTH, Donald E.: *The art of computer programming, volume 3: (2nd ed.) sorting and searching*. Redwood City, CA, USA : Addison Wesley Longman Publishing Co., Inc., 1998. – ISBN 0-201-89685-0

[Kriens und Hargrave 2004] KRIENS, P. ; HARGRAVE, B.: Listeners Considered Harmful: The "Whiteboard"Pattern. / OSGi Alliance. August 2004. – Technical whitepaper. osgi, whiteboard pattern

[Leunziger 1996] LEUNZIGER, Paul Schönsleben Ruth: *Innovative Gestaltung von Versicherungsprodukten - Flexible Industriekonzepte in der Assekuranz*. Gabler, 1996

[Ley] LEY, Jim: *FOAFnaut is a viewer of peoples relationships*. – URL `http://jibbering.com/nauts/htmlnaut/`. – 2009.03.03

[Li u. a. 2005] LI, Hang ; CAO, Yunbo ; XU, Jun ; HU, Yunhua ; LI, Shenjie ; MEYERZON, Dmitriy: A new approach to intranet search based on information extraction. In: *CIKM*

'05: Proceedings of the 14th ACM international conference on Information and knowledge management. New York, NY, USA : ACM, 2005, S. 460–468. – ISBN 1-59593-140-6

[Li u. a. 2007] LI, Rui ; BAO, Shenghua ; YU, Yong ; FEI, Ben ; SU, Zhong: Towards effective browsing of large scale social annotations. In: *WWW '07: Proceedings of the 16th international conference on World Wide Web*. New York, NY, USA : ACM, 2007, S. 943–952. – ISBN 978-1-59593-654-7

[Lin u. a. 2007] LIN, C.-Y. ; GRIFFITHS-FISHER, V. ; EHRLICH, K. ; DESFORGES, C.: SmallBlue: People Mining for Expertise Search and Social Network Analysis. In: *IEEE Multimedia Magazine* (2007), Oct.–Dec.

[LinkedIn] LINKEDIN: *LinkedIN*. – URL `http://www.linkedin.com/`. – 2009.03.03

[LiVEJOURNAL 2008] LIVEJOURNAL: *LiveJournal lets you express yourself, share your life, and connect with friends online*. 2008. – `http://www.livejournal.com/`

[Ltd.] LTD., Atlassian Software Systems P.: *Track your issues & tasks*. – URL `http://www.jira.com`. – 2009.03.03

[Lu u. a. 2004] LU, Yi ; LU, Shiyong ; FOTOUHI, Farshad ; DENG, Youping ; BROWN, Susan J.: FGKA: a Fast Genetic K-means Clustering Algorithm. In: *SAC '04: Proceedings of the 2004 ACM symposium on Applied computing*. New York, NY, USA : ACM, 2004, S. 622–623. – ISBN 1-58113-812-1

[Lucene] LUCENE, Apache: *Java-based indexing and search technology*. – URL `http://lucene.apache.org/`. – 2009.03.03

[Mählmann 2008] MÄHLMANN, Lars: *Deliver Who I Mean*. Feb 2008. – URL `http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master07-08-aw/maehlmann/bericht.pdf`. – Project Summary, describes a profile which is indexable and readable by Apache Lucene

[Makrehchi und Kamel 2006] MAKREHCHI, Masoud ; KAMEL, Mohamed S.: Learning Social Networks from Web Documents Using Support Vector Classifiers. In: *WI '06: Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*. Washington, DC, USA : IEEE Computer Society, 2006, S. 88–94. – ISBN 0-7695-2747-7

[Matos und Sousa 2008] MATOS, Miguel ; SOUSA, António: Dependable distributed OSGi environment. In: *Companion '08: Proceedings of the ACM/IFIP/USENIX Middleware '08 Conference Companion*. New York, NY, USA : ACM, 2008, S. 104–106. – ISBN 978-1-60558-369-3

[Matsumura u. a. 2005] MATSUMURA, Naohiro ; GOLDBERG, David E. ; LLORÀ, Xavier: Mining directed social network from message board. In: *WWW '05: Special interest tracks and posters of the 14th international conference on World Wide Web*. New York, NY, USA : ACM, 2005, S. 1092–1093. – ISBN 1-59593-051-5

[Matsuo u. a. 2006] MATSUO, Yutaka ; MORI, Junichiro ; HAMASAKI, Masahiro ; ISHIDA, Keisuke ; NISHIMURA, Takuichi ; TAKEDA, Hideaki ; HASIDA, Koiti ; ISHIZUKA, Mitsuru: POLYPHONET: an advanced social network extraction system from the web. In: *WWW '06: Proceedings of the 15th international conference on World Wide Web*. New York, NY, USA : ACM, 2006, S. 397–406. – ISBN 1-59593-323-9

[von Maur Robert Winter 2002] MAUR ROBERT WINTER, Eitel von: *Vom Data Warehouse zum Corporate Knowledge Center*. Physica-Verlag, 2002

[Mediawiki] MEDIAWIKI: *MediaWiki is a free software wiki package*. – URL `http://www.mediawiki.org/wiki/MediaWiki`. – 2009.03.03

[Miller 2007] MILLER, Dan Brickley L.: *FOAF Vocabulary Specification 0.91*. November 2007. – URL `http://xmlns.com/foaf/spec/`

[Mircosoft] MIRCOSOFT: *Microsoft Project 2007*. – URL `http://office.microsoft.com/en-us/project/FX100487771033.aspx`. – 2009.03.03

[Nair und Sarasamma 2007] NAIR, Premchand S. ; SARASAMMA, Suseela T.: Data Mining Through Fuzzy Social Network Analysis. In: *North American Fuzzy Information Processing Society*, 2007, S. 251 – 255. – ISBN 1-4244-1214-5

[Nardi B.A und H. 2002] NARDI B.A, Whittaker S. ; H., Schwarz: It's not what you know, it's who you know: Work in the information age. (2002)

[Notes] NOTES, IBM L.: *Lotus Notes Homepage*. – URL `http://www-01.ibm.com/software/lotus/`. – 2009.03.03

[Nutch] NUTCH, Apache: *Nutch, is a open source web-search software*. – URL `http://lucene.apache.org/nutch/index.html`. – 2009.03.03

[Oracle] ORACLE: *Oracle TopLink*. – URL `http://www.oracle.com/technology/products/ias/toplink/index.html`. – 2009.03.03

[Partner] PARTNER, IBM B.: *Simplified Project Management through Lotus Notes*. – URL `http://www.trackersuite.com/`. – 2009.03.03

[Petrelli u. a. 2006] PETRELLI, Daniela ; LANFRANCHI, Vitaveska ; MOORE, Phil ; CIRAVEGNA, Fabio ; CADNAS, Colin: Oh my, where is the end of the context?: dealing with information in a highly complex environment. In: *IIiX: Proceedings of the 1st international conference on Information interaction in context*. New York, NY, USA : ACM, 2006, S. 37–41. – ISBN 1-59593-482-0

[Philip G. Zimbardo 1996] PHILIP G. ZIMBARDO, Richard J. G.: *Psychologie*. Springer Verlag, 1996

[Programme 2009] PROGRAMME, HP Labs Semantic W.: *Jena - A Semantic Web Framework for Java*. January 2009. – URL `http://jena.sourceforge.net/`

[SAP] SAP: *SAP*. – URL `http://www.sap.com/index.epx`. – 2009.03.03

[Schutz 2008] SCHUTZ, Alexander T.: *Keyphrase Extraction from Single Documents in the Open Domain Exploiting Linguistic and Statistical Methods*, Digital Enterprise Research Institute, National University of Ireland, Galway, Diplomarbeit, August 2008

[Siegrist 1995] SIEGRIST, Johannes: *Medizinische Soziologie : mit 13 Tabellen*. München, Wien, Baltimore : Urban und Schwarzenberg, 1995. – ISBN 3-541-06385-8

[Signal-Iduna 2008] SIGNAL-IDUNA: Skill Management - Fertigkeiten kennen und nutzen. In: *KONtour* 37 (2008), Dezember, Nr. 37, S. 8,9

[Song u. a. 2005] SONG, Xiaodan ; LIN, Ching-Yung ; TSENG, Belle L. ; SUN, Ming-Ting: Modeling and predicting personal information dissemination behavior. In: *KDD '05: Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*. New York, NY, USA : ACM, 2005, S. 479–488. – ISBN 1-59593-135-X

[Stanley Wasserman 1994] STANLEY WASSERMAN, Katherine F.: *Social Network Analysis: Methods and Applications (Structural Analysis in the Social Sciences)*. Bd. 1. Cambridge University Press, November 1994. – 857 S

[Strigi] STRIGI: *Strigi, the fastest and smallest desktop searching programm*. – URL `http://strigi.sourceforge.net/`. – 2009.03.03

[Swish-e] SWISH-E: *Simple Web indexing System for Humans - Enhanced*. – URL `http://swish-e.org/`. – 2009.03.03

[Tanenbaum und van Steen 2006] TANENBAUM, Andrew S. ; STEEN, Maarten van: *Distributed Systems: Principles and Paradigms (2nd Edition)*. Upper Saddle River, NJ, USA : Prentice-Hall, Inc., 2006. – ISBN 0132392275

[Tang u. a. 2007a] TANG, John C. ; DREWS, Clemens ; SMITH, Mark ; WU, Fei ; SUE, Alison ; LAU, Tessa: Exploring patterns of social commonality among file directories at work. In: *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*. New York, NY, USA : ACM, 2007, S. 951–960. – ISBN 978-1-59593-593-9

[Tang u. a. 2007b] TANG, John C. ; LIN, James ; PIERCE, Jeffrey ; WHITTAKER, Steve ; DREWS, Clemens: Recent shortcuts: using recent interactions to support shared activities. In: *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*. New York, NY, USA : ACM, 2007, S. 1263–1272. – ISBN 978-1-59593-593-9

[Tavares und Valente 2008] TAVARES, Andre L. ; VALENTE, Marco T.: A gentle introduction to OSGi. In: *SIGSOFT Softw. Eng. Notes* 33 (2008), Nr. 5, S. 1–5. – ISSN 0163-5948

[Valetto u. a. 2007] VALETTO, Giuseppe ; HELANDER, Mary ; EHRLICH, Kate ; CHULANI, Sunita ; WEGMAN, Mark ; WILLIAMS, Clay: Using Software Repositories to Investigate Socio-technical Congruence in Development Projects. In: *MSR '07: Proceedings of the Fourth International Workshop on Mining Software Repositories*. Washington, DC, USA : IEEE Computer Society, 2007, S. 25. – ISBN 0-7695-2950-X

[Wang u. a. 2002] WANG, Jidong ; CHEN, Zheng ; TAO, Li ; MA, Wei-Ying ; WENYIN, Liu: Ranking user's relevance to a topic through link analysis on web logs. In: *WIDM '02: Proceedings of the 4th international workshop on Web information and data management*. New York, NY, USA : ACM, 2002, S. 49–54. – ISBN 1-58113-593-9

[Wang und Kitsuregawa 2002] WANG, Yitong ; KITSUREGAWA, Masaru: Evaluating contents-link coupled web page clustering for web search results. In: *CIKM '02: Proceedings of the eleventh international conference on Information and knowledge management*. New York, NY, USA : ACM, 2002, S. 499–506. – ISBN 1-58113-492-4

[WebGlimpse] WEBGLIMPSE: *WebGlimpse, Allow users to search your website without installing any software*. – URL `http://webglimpse.net/subsupport/remoteindexing.html`. – 2009.03.03

[Wilder 1986] WILDER, D.A.: Social Categization: Implications for creation and reduction of intergroup bias. In: *Advances in Experimental Social Psychology*. Springer Verlag, 1986

[William T. Councill 2001] WILLIAM T. COUNCILL, George T. H.: *Component-Based Software Engineering*. Addison-Wesley, 2001

[Witten und Frank 2005] WITTEN, Ian H. ; FRANK, Eibe: *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition (Morgan Kaufmann Series in Data Management Systems)*. San Francisco, CA, USA : Morgan Kaufmann Publishers Inc., 2005. – ISBN 0120884070

[Wotsit] WOTSIT: *Wotsit, the programmer's file and data format resource*. – URL `http://www.wotsit.org/`. – 2009.03.03

[Wütherich u. a. 2008] WÜTHERICH, Gerd ; HARTMANN, Nils ; KOLB, Bernd ; LÜBKEN, Matthias: *Die OSGI Service Platform-Eine Einführung mit Eclipse Equinox*. Dpunkt Verlag, April 2008 (1). – 468 S

[xfriend] XFRIEND: *xfriend*. – URL `http://www.x-friend.de/`. – 2009.03.03

[Xing] XING: *XING Global networking for professionals*. – URL `https://www.xing.com/`. – 2009.03.03

[Yahoo!] YAHOO!: *Yahoo! Desktop Suche*. – URL `http://de.docs.yahoo.com/search/desktop/`. – 2009.03.03

[Yiman-Seid und Kobsa 2003] YIMAN-SEID, D. ; KOBSA, A: *Sharing Expertise: Beyond Knowledge Management*. Kap. Expert-finding systems for organizations: Problem and domain analysis and the DEMOIR approach, S. 328–358. In: *Sharing Expertise: Beyond Knowledge Management*, MIT Press, 2003

[Yu und Singh 2002] YU, Bin ; SINGH, Munindar P.: Searching Social Networks. Raleigh, NC, USA : North Carolina State University at Raleigh, 2002. – Forschungsbericht. social networks

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach §22(4) bzw.§24(4) ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 23. März 2009 Lars Mählmann