



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Anwendung 2

Lars Mählmann

Deliever who I mean

Lars Mählmann
Deliever who I mean

Anwendung 2 eingereicht im Rahmen der Diplomprüfung
im Studiengang Informatik
Studienrichtung Softwaretechnik
am Fachbereich Elektrotechnik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer :
Zweitgutachter : Prof. Dr. Kai von Luck

Abgegeben am 6. März 2008

Inhaltsverzeichnis

1	Einleitung	3
2	Deliever who I mean . . .	4
2.1	Social Networking	4
2.2	Konzept der Suche	4
3	Problembeschreibung	5
3.1	Anwendungsbeschreibung	5
3.2	Anforderungen	6
4	Technologie-Evaluation	7
4.1	V-Cards	7
4.2	XML VCARD	8
4.3	FOAF - Friend of a friend	9
4.4	Suchmaschinen	10
4.4.1	Lucene - apache.org	10
5	Implementierung eines Frontends	11
6	Zusammenfassung	12
6.1	Fazit	12
6.2	Ausblick	12
	Literatur	14
.1	RDF - Ressource Description Framework	15

1 Einleitung

Dieses Dokument ist ein Projektbericht über die Implementierung der Suche nach Benutzerprofilen [Mählmann \(2008b\)](#). Die Ausarbeitung besteht aus zwei Teilen, im ersten Teil wird untersucht wie Informationen über eine Person strukturiert abgelegt werden können. Im zweiten Teil ist eine Evaluation von Suchmaschinen zur Erweiterung der Suche nach Personen.

Die Gliederung der Ausarbeitung ist wie folgt: Im ersten Abschnitt [2](#) wird die Idee für eine Suche und die Idee von Social Networking vorgestellt. Im Teil [3](#) wird eine Problembeschreibung gegeben. Anhand der Problembeschreibung werden in Kapitel [4](#) verschiedene Alternativen

betrachtet. Darauf folgt die Implementierung eines Prototyps in Abschnitt 5. Am Ende in Kapitel 6 wird das Ergebnis diskutiert und ein Ausblick auf noch offene Punkte der Ausarbeitung gegeben.

2 **Deliever who I mean . . .**

In diesem Artikel soll die Implementierung einer Personensuche für das Intranet evaluiert werden. Die Umsetzung besteht aus zwei Bereichen (Frontend and Backend), die in den Arbeiten [Mählmann \(2008a\)](#) und [Mählmann \(2008b\)](#) beschrieben wurden. Hier soll noch einmal kurz die Idee des Frontends erläutert werden, um die Problemstellung zu verstehen. Die Idee für die Suche nach Personen basiert auf dem Konzept von Social Navigation. Was ist Social Navigation und wie wird dieses in einem Computerprogramm abgebildet.

2.1 **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, Geschäftsbeziehungen, dieselbe Universität, Konferenzen, etc . . . [Matsuo u. a. \(2006\)](#).

Die Struktur wird in Graphen abgebildet, wobei Personen die Knoten darstellen und Beziehungen im Netzwerk als Kanten bezeichnet ¹und beschrieben werden (siehe [Granovetter \(1973\)](#)).

2.2 **Konzept der Suche**

Die Suche nach Personen findet in zwei Schritten statt. Im ersten Schritt werden Daten aus dem Intranet extrahiert und in einem Datawarehouse abgelegt. Die Daten werden mit verschiedenen Techniken aus dem Intranet extrahiert, unterschieden wird dabei nach dem Typ von Datenquelle. Es gibt strukturierte wie Datenbanken oder LDAP oder unstrukturierte wie Code Repositories. Je nach Datenquelle werden unterschiedliche Techniken benutzt, um zusätzlich semantische Informationen mit dem Wissen über die jeweilige Quelle gewinnen zu können.

¹Vgl. <http://de.wikipedia.org/wiki/Netzwerktheorie>

Im zweiten Schritt werden die Daten ausgewertet, um die gesuchten Informationen über Personen zu finden, die zur Beschreibung dieser relevant sind. Die Auswertung wird mit Hilfe von Data Mining und Text Mining durchgeführt.

Im zweiten Schritt werden die Daten in einem Profil abgespeichert, welches eine Person abbildet und mit dessen Eigenschaften beschreibt. Das bedeutet, daß die Suchmaschine die Profile indizieren und interpretieren kann (siehe Abbildung 1).

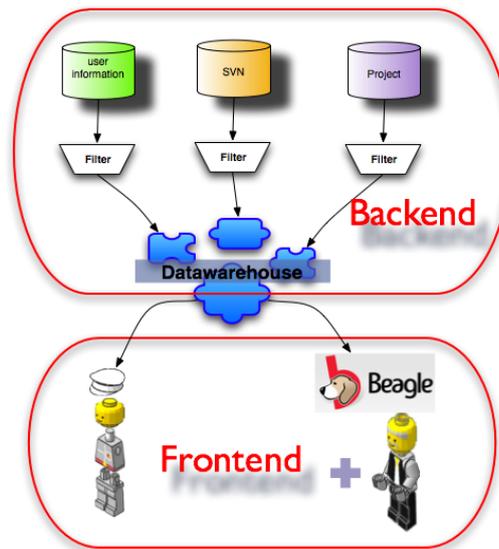


Abbildung 1: Abbildung eines Konzeptes zur Personensuchen

Das hier beschriebene Konzept stellt die gesamte Implementierung der Suche dar, in dieser Arbeit wird nur der zweite Teil, das Frontend, betrachtet. Im nächsten Schritt werden die einzelnen Bereiche des Frontends und die zu lösenden Probleme genauer erläutert.

3 Problembeschreibung

3.1 Anwendungsbeschreibung

Diese Abbildung 2 stellt auf der rechten Seite die Person dar, die mit Hilfe einer Suchmaschine nach Personen sucht, die aus dem Datawarehouse generiert worden sind. Die linke Seite zeigt ein erstelltes Profil, das aus verschiedenen Informationen zu einer Person zusammengestellt ist.



Abbildung 2: Abbildung des Frontends zur Personensuchen

Die Suchmaschine "scant" die Personenprofile und bietet dem Suchenden die Möglichkeit mit verschiedenen Suchbegriffen Übereinstimmungen in den Profilen zu finden und diese Person als Suchergebnis, mit Informationen, wie Telefonnummer, Adresse oder die Position in der Firma, darzustellen.

3.2 Anforderungen

Wie könnte ein Profil aussehen und was für eine Suchmaschine benötigt man zum Auslesen der Profile? Es gilt verschiedene Dinge zu beachten bei der Profilgenerierung:

- Die ausgewerteten Daten aus dem Datawarehouse müssen in eine Struktur gebracht werden, die einfach generierbar und zudem von einer Suchmaschine indizierbar ist.
- 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 hinzufügbare 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 automatisch diese Profile auslesen und interpretieren. Die gesamte Anwendung muss auf Grund der Menge der Informationen und deren Komplexität automatisch umsetzbar sein.

Ist ein Profil erstellt und erfüllt die oben genannten Kriterien muss die Suchmaschine diese durchsuchen können. Ist es möglich eine bereits existierende Suchmaschine zu nutzen oder muss eine eigene Suche implementiert werden? Soll die Suchmaschine auf diese Profile angepasst werden oder verändert man die Profile und deren Kriterien soweit, dass man eine Standardsuchmaschine benutzen kann. Den Aufwand für die Möglichkeiten gilt es abzuschätzen. Eine weitere Option ist es eine Suchmaschine um eine Profilsuche zu erweitern.

Es hat den Vorteil keine eigene Suche direkt zu implementieren, sondern eine Bestehende nutzen zu können und die Profile so anzuwenden, wie oben beschrieben.

Im nächsten Kapitel sollen verschiedene Möglichkeiten für Profildarstellungen aufgezeigt werden. Danach wird untersucht, welche Suchmaschine geeignet sein kann, um nach Profilen zu suchen.

4 Technologie-Evaluation

Im Folgenden sollen die verschiedenen Möglichkeiten zur Generierung eines Profiles beschrieben werden. Es gilt die vorher festgelegten Kriterien zu erfüllen und zusätzlich eine einfach zu implementierende Lösung zu finden.

4.1 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 Emailprogrammen 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 Standarts nicht einheitlich.

4.2 XML VCARD

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

- XML Dokumente sind einfach erstellbar
- Durch Suchmaschinen lesbar, dar auch XML Dokumente in einfachen Textdateien ablegbar sind.
- XML Formate lassen beliebig viele Attribute zu, die man beliebig 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 Standart 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 proprietären Lösungen benutzt.

Das W3C hat einen auf RDF basierte Standart 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 Standart 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 Standart und XML selbst nicht möglich. Der Nachteil ist, daß nicht genügend Attribute zur genaueren Beschreibung einer Person bestehen bleiben. Eine Erweiterung wäre den Standart nutzen und mit eigenen zusätzlichen Attributen anzureichern.

Basierend auf RDF und der Idee von VCARD soll im nächsten Abschnitt ein weiter Standart von W3C betrachtet werden (Für eine genauere Erklärung zu RDF siehe Anhang .1).

4.3 FOAF - Friend of a friend

Friend of a friend [Brickley und Miller \(2008\)](#) 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 größere Anzahl an Beschreibungen als eine VCARD. Die folgende Abbildung 3 zeigt einen Überblick der Tags zur Darstellung einer Person.

Im nächsten Kapitel soll eine Implementierung der FOAF Profile getestet werden. Im nächsten Abschnitt werden Suchmaschinen auf die Möglichkeiten zur Personensuche hin diskutiert. Die Suchmaschine sollte entweder FOAF unterstützen oder die Möglichkeit bieten diese um FOAF zu erweitern.

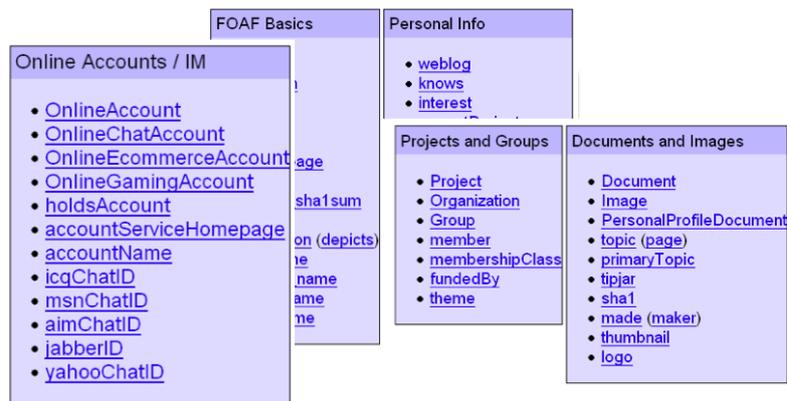


Abbildung 3: Attributdarstellung von FOAF

4.4 Suchmaschinen

Es gibt eine Reihe von verschiedenen frei verfügbaren Suchmaschinen. Verschiedenen wie Doodle ², Swish-e ³, Xapian ⁴ und Pinot ⁵ unterstützen eine Volltextsuche auf die verschiedenen Dokumententypen. Die einzelnen Suchmaschine verfügen über dieselbe Funktionalität. Keine dieser Suchmaschinen unterstützt zum jetzigen Zeitpunkt eine RDF Suche oder Interpretation. Eine weitere Schwierigkeit ist die jeweilige API, die Zeit und Aufwand benötigt, um eine Erweiterung zu implementieren. Eine Alternative ist Lucene. Lucene wird von verschiedenen Suchmaschine wie Nutch, Scalix Strigi oder Beagle (Love und Kubasik (2008a)) als Basis genutzt.

4.4.1 Lucene - apache.org

Lucene (Cutting (2008)) ist aktuell eine der meist genutzteste Suchmaschine. Wobei Lucene keine Suchmaschine im herkömmlichen Sinne ist, sondern ein Suchmaschinen-Framework. Lucene stellt Klassen und Methoden bereit, um Dokumente zu analysieren, zu indizieren und zu speichern. Die Bibliothek setzt sich aus zwei Hauptkomponenten zusammen,

- Eine Teilkomponente erzeugt den Index
- Eine Query Engine durchsucht den Index

²<http://gnunet.org/doodle/>

³<http://swish-e.org/>

⁴<http://www.xapian.org/>

⁵<http://pinot.berlios.de/index.html>

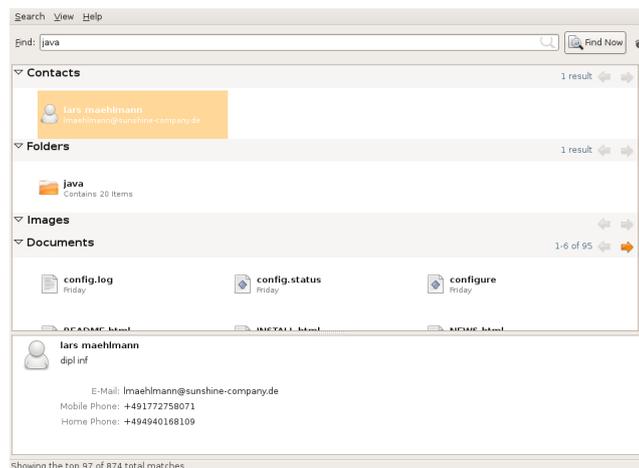


Abbildung 4: Abbildung des Frontends zur Personensuchen

Verschiedene Implementierungen beruhen auf diesem Framework. Die bekanntesten sind: Lucene Java, Nutch, Lucy, Lucene .NET, Tika. Die Vorteile der Suchmaschine sind eine hohe Performance, gute Skalierbarkeit, gute Dokumentation und zahlreiche Erweiterungen um die Suchmaschine an die eigenen Bedürfnisse anzupassen. In den nächsten Kapitel wird versucht Lucene zur Profilsuche einzusetzen.

5 Implementierung eines Frontends

Es folgt eine Implementierung des Prototyps mit Hilfe der beschriebenen Technologien. Die Suchmaschine Beagle ist eine Desktop Suchmaschine basierend auf dem vorgestellten Framework Lucene. Beagle ist ein Open Source Project (Love und Kubasik (2008a)), es durchsucht die angegebenen Verzeichnisse nach Dokumenten, emails, Adressbücher, Quellcode, Bilder, Multimedia, komprimierte Dateien sowie Anwendungen. Beagle realisiert diese Suche mit verschiedenen Filtern, auf die jeweiligen Datentypen angewandt. Dieser modularer Aufbau ermöglicht somit eine Erweiterung für einen FOAF Filtern. In einem ersten Test wurde eine VCard aus dem Adressbuch mit Schlagwörtern, wie z. B. Java, Eclipse angereichert. Bei der der Suche mit Beagle wurde dann die beschriebene Person als Suchergebniss angezeigt (4).

Dieses Beispiel zeigt, dass es möglich ist eine Personensuche mit Beagle zu realisieren. In einem zweiten Schritt wurde ein FOAF Profil manuell erstellt mit Hilfe von ⁶. Mit Hilfe von RDF Parsern lässt sich dieses Profil dann auslesen. Zwei Parser die hier zum Einsatz kamen waren ein Kommandozeilentool (Beckett (2008)), und eine Semantic Web/RDF Library for

⁶<http://www.ldodds.com/foaf/foaf-a-matic>

C#.NET Library [Tauberer \(2008\)](#) zum Einbinden in andere Anwendungen, in diesem Fall in einem Beagle Filter. Beide Anwendungen lieferten die erwarteten Ergebnisse. Im letzten Schritt wurde an Hand des Tutorial ([Love und Kubasik \(2008b\)](#)) ein Filter zum Suchen nach Personen implementiert. Der Filter kann als externen Plugin in die Beaglesuchmaschine implementiert werden und liefert das erstellte Profil als Suchergebnis zurück.

Die Implementierung eines Frontends auf Basis der bisher vorgestellten Technologien verlief erfolgreich und ermöglicht eine Personensuche, wie in dem Konzept zu einem Frontend vorgestellt.

6 Zusammenfassung

Dieser Projektbericht steht im direkten Zusammenhang mit den Ausarbeitungen [Mählmann \(2008a\)](#) und [Mählmann \(2008b\)](#) die sich ebenfalls mit den Konzept zu einer Personensuche beschäftigen. Der Schwerpunkt dieser Ausarbeitung liegt im Vergleich zu den anderen auf der Implementation einer Personensuche, ein Teilbereich des Gesamtkonzeptes. Es wurde im ersten Kapitel [2](#) beschrieben, wie das Konzept aussieht. Im zweiten Abschnitt wurden die damit verbunden Probleme erläutert, um in den folgenden Abschnitten verschiedene Ansätze zu diskutieren und eine Implementation zur Personensuche zu demonstrieren.

6.1 Fazit

Abschließend kann die Umsetzung des Frontendbereiches für eine Personensuche als erfolgreich angesehen werden. Die Schwierigkeiten bei der Umsetzung des Frontendbereiches bestanden hauptsächlich in der Entwicklung eines Konzeptes und in der Untersuchung der verschiedenen Technologien: den Profilbeschreibungen und den unterschiedlichen Suchmaschinen. Die Einarbeitung in Mono und den verschiedenen RDF Parser benötigte gegen Ende mehr Zeit als erwartet. Einer der Gründe war die teilweise eine nicht ausreichende Dokumentation und eine einfache gehaltene Entwicklungsumgebung, die die Entwicklung umständlicher gestaltete.

6.2 Ausblick

In der anstehenden Masterarbeit soll dieser Prototyp verwendet und weiter ausgebaut werden. Es ist im Moment nicht möglich über die GUI von Beagle den Filter zur Personensuche zu verwenden. Mit Hilfe von FOAF sind weitere Verfeinerungen möglich. Desweiteren muss untersucht werden, wie gut die Indizierung von neuen Profilen und dynamische Änderungen

an bestehenden Profilen funktioniert. Für einen anstehenden Prototypen in der Masterarbeit zur Personensuche ist die bisherige Implementierung ausreichend und der Schwerpunkt kann auf die Informationsgewinnung gelegt werden.

Literatur

- [Beckett 2008] BECKETT, Dave: *Raptor RDF Parser Library - Raptor RDF parser utility*. 2008. – <http://librdf.org/raptor/rapper.html>
- [Berners-Lee 2008] BERNERS-LEE, Tim: *W3C, Semantic Web*. 2008. – <http://www.w3.org/2001/sw/pub>
- [Brickley und Miller 2008] BRICKLEY, Dan ; MILLER, Libby: *FOAF Vocabulary Specification 0.91*. 2008. – <http://xmlns.com/foaf/spec/>
- [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
- [Consortium 2008] CONSORTIUM, Internet M.: *vCard: Your Electronic Business Card*. 2008. – <http://www.imc.org/pdi/vcardoverview.html>
- [Cutting 2008] CUTTING, Doug: *Apache Lucene*. 2008. – <http://lucene.apache.org>
- [Dawson und Howes 1998] DAWSON, F. ; HOWES, T.: *vCard MIME Directory Profile*. 1998. – <http://tools.ietf.org/html/rfc2426>
- [Dumbill 2002] DUMBILL, Edd: *Representing vCard Objects in RDF/XML*. 2002. – <http://www.ibm.com/developerworks/xml/library/x-foaf.html>
- [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
- [Granovetter 1973] GRANOVETTER, Mark: The strength of weak ties. In: *American Journal of Sociology* (1973)
- [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
- [LIVEJOURNAL 2008] LIVEJOURNAL: *LiveJournal lets you express yourself, share your life, and connect with friends online*. 2008. – <http://www.livejournal.com/>

- [Love und Kubasik 2008a] LOVE, Robert ; KUBASIK, Kevin: *Beagle Project*. 2008. – http://beagle-project.org/Main_Page
- [Love und Kubasik 2008b] LOVE, Robert ; KUBASIK, Kevin: *Beagle Project, Hello World Filter*. 2008. – http://beagle-project.org/Filter_Tutorial
- [Mählmann 2008a] MÄHLMANN, Lars: *Deliever who I mean*, 2008
- [Mählmann 2008b] MÄHLMANN, Lars: *Deliever who I mean, continues*, 2008
- [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
- [Sinha und Karger 2005] SINHA, Vineet ; KARGER, David R.: Magnet: supporting navigation in semistructured data environments. In: *SIGMOD '05: Proceedings of the 2005 ACM SIGMOD international conference on Management of data*. New York, NY, USA : ACM, 2005, S. 97–106. – ISBN 1-59593-060-4
- [Tang u. a. 2007] 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
- [Tauberer 2008] TAUBERER, Joshua: *Semantic Web/RDF Library for C#/NET*. 2008. – <http://razor.occams.info/code/semweb/>
- [Yu und Singh 2002] YU, Bin ; SINGH, Munindar P.: *Searching Social Networks*. Raleigh, NC, USA : North Carolina State University at Raleigh, 2002. – Forschungsbericht

.1 RDF - Ressource Description Framework

In dem oberen Kapitel 4.2 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 Berners-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 ⁷. 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 Ressourcen. Die Properties sind in der RDF Spezifikation festgelegt.
- Statements, die bestimmte Ressourcen 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>”

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

Eine textuelle Darstellung im Source Code sieht folgendermassen aus,

```
\begin{figure} [h]
  \centering
  \includegraphics [totalheight=0.5\textwidth] {graphic/concept.png}
  \caption{Abbildung eines Konzeptes zur Personensuchen}
  \label{fig:concept}
\end{figure}
```

⁷<http://www.w3.org/TR/PR-rdf-syntax/>