Problemseminar:

Peer-to-Peer Data-Management

Thema 8:

Content- und Knowledge-Management in Peer-to-Peer-Systemen

Bearbeiter:

Christian Helmchen

Betreuer:

Andreas Thor

Inhaltsverzeichnis

1.Einleitung

1.1.Motivation und Beispiel

Was ist Content- bzw. Knowledge-Management und was bringt es mir bzw. meiner Firma ?
Sehen wir uns hierzu ein kleines Beispiel an:
Ein Caddy arbeitet seit mehreren Jahren in einem Golfklub. In dieser Zeit hat er viel Erfahrung gesammelt und kann somit den Golfern nützliche Tips an dem einen oder anderen Loch geben. Dies führt indirekt wiederum zu einer leichten Umsatzsteigerung des Golfklubs, da der Golfer bessere Ergebnisse erzielt und den Klub öfter besucht oder vielleicht sogar ein paar Freunde mitbringt. Somit profitieren alle Beteiligten vom Wissen des Caddys. Bei einem einzelnen Caddy kann man zwar noch nicht von Knowlegde-Management sprechen, wenn aber alle Caddys des Golfklubs ihr Wissen untereinander auf eine festgelegte Art und Weise (ein Schwarzes Brett oder eine regelmäßige Besprechung) austauschen, dann kann man sagen, dass durch dieses Knowlegde-Management der Klub, die Caddys sowie auch die Golfer gleichermaßen *profitieren*. [CXO03]

Nachfolgend werde ich näher auf Content- und Knowledge-Management-Systeme eingehen und insbesondere die Wechselnden Anforderungen auf Grund des Vormarsches der Peer-to-Peer-Systeme aufzeigen. Anschließend werde ich einige Ansätze und Architekturen charakterisieren.

1.2.Definitionen

Knowledge

- Wissen sind die Fakten, Gefühle und Erfahrungen einer Person oder Gruppe von Personen. Wissen wird aus Informationen abgeleitet, ist dabei jedoch reichhaltiger und aussagekräftiger. Es schließt Vertrautheit, Kenntnis und Verständnis – gewonnen aus Erfahrung oder durch Studium – sowie Ergebnisse aus Vergleichen, Ziehen von Schlüssen und Knüpfen von Verbindungen ein. [NELH01]
- Wissen wird auch als Menge von Modellen zur Beschreibung von Eigenschaften und Verhalten innerhalb eines (Fach)Bereiches definiert. Es kann im Gehirn des Einzelnen (Know How) oder in

organisatorischen Prozessen, Produkten, Einrichtungen, Systemen und Dokumenten gespeichert werden.

- In [UT98] wird Wissen auch als die Ideen oder das Verständnis einer Entität, das zum effektiven Erreichen der Ziele dieser Entität genutzt wird, bezeichnet. Ferner ist Wissen oftmals die beste oder sogar einzige Möglichkeit eines Unternehmens, sich von Konkurrenten *abzuheben*.

Knowledge-Management (KM)

- Die Bereitstellung und Betreuung einer Umgebung, die es ermöglicht, Wissen zu erzeugen, zu teilen, zu erlernen, zu verbessern, zu organisieren, zu analysieren und zum Wohle der Organisation und ihrer Kunden zu nutzen, nennt man Knowlegde-Management. [NELH01]
- Knowlegde-Management bewahrt geistige Werte vor dem Verfall (das Rad muss nicht neu erfunden werden) und erhöht die Flexibilität eines Unternehmens sowie die Motivation und die Zusammenarbeit der Mitarbeiter untereinander (durch gezielte Förderung und Belohnungen). Es unterstützt Unternehmen bei der Entscheidungsfindung und Problemlösung und steigert Effektivität sowie Produktivität. [Ha01], [UT98]

Content-Management (CM)

- Inhalt bezieht sich in diesem Zusammenhang meist auf computerbasierte Informationen wie der Inhalt einer Webseite oder einer Datenbank.
- Content-Management beschreibt alle Maßnahmen zur Sicherstellung der Relevanz, Aktualität, Genauigkeit, einfachen Zugänglichkeit, guten Organisierung usw. der entsprechenden Inhalte mit dem Ziel, dem Nutzer qualitativ hochwertige Informationen möglichst schnell zur Verfügung zu stellen. [NELH01]
- Die zum Erreichen dieser Ziele zum Einsatz kommenden Content-Management-Systeme (CMS) können darüber hinaus als Basis für Knowledge-Management dienen. [Ha01]

2. Content-Management

Laut einer Studie ([GC01]) lassen sich im Bereich des CM gegenwärtig drei Trends beobachten: die „Explosion“ unstrukturierter Daten, der dringende Bedarf Inhalte zu managen sowie die wachsende Nutzung des Internet bei der Arbeit innerhalb eines Unternehmens und zwischen verschiedenen Unternehmen. Diese Trends führen zu zwei wichtigen Anforderungen an CMS: die Ermöglichung kollaborativer Arbeit und das Erschaffen besserer Onlineerlebnisse für die User. In diesem Kapitel werde ich erläutern, warum Unternehmen in Zukunft nicht mehr um verteilte Content-Management-Systeme herumkommen werden und welche Rolle Peer-to-Peer-Systeme dabei spielen.

2.1. Verteilte CMS

Explosion unstrukturierter Daten

Strukturierte Daten werden normalerweise in Data Warehouses, Data Marts oder Datenbanken gespeichert. Bei unstrukturierten Daten ist dies allerdings nicht so ohne weiteres möglich. Ihre wachsende Menge erschwert Sharing, Management oder Filtern für Unternehmen wie auch Nutzer erheblich. Daher müssen verteilte CMS (DCMS) in der Lage sein, auf Inhalte beliebiger Formate an beliebigen Orten ohne Umwege zuzugreifen, was gleichzeitig die Reichweite und Einbeziehung der Dateninhaber erhöht. Dies gibt ihnen im Gegensatz zum zentralen Repository der zentralisierten CMS die Möglichkeit zur Erschaffung eines virtuellen Repositorys, wodurch eine Strukturierung der Inhalte nicht mehr zwingend notwendig ist.

Charakterisierung

DCMS ermöglichen den Echtzeitzugriff auf Inhalte wo immer sie sich befinden. Im Zusammenspiel mit dem virtuellen Repository können sie damit die Basis beliebiger Anwendungen bilden, die einen effizienten Datenzugriff in Echtzeit benötigen. Desweiteren ergänzen DCMS Unternehmensportale im Bereich Supply Chain Management (SCM), Customer Relationship Management (CRM) und E-Commerce. Außerdem sind DCMS besser skalierbar und zuverlässiger als zentralisierte Architekturen. Seine Inhalte verwaltet ein DCMS mittels dynamischer Inhaltstabellen. Die Suche

auf den verteilten Ressourcen erfolgt für die Nutzer nach außen genauso wie auf einem Server. Im Gegensatz zu zentralisierten CMS behalten die Dateninhaber jedoch die Kontrolle über ihre Inhalte und können Updates in Echtzeit durchführen. Die Zugriffsberechtigungen werden dabei auf der Basis der Rechte am Ursprungsort der Daten erstellt. Zu deren Durchsetzung erfolgt eine Authentifizierung der Nutzer. Die Darstellung der Ergebnisse von Suchanfragen ist unabhängig vom ursprünglichen Format der Inhalte (konsistent) und kann mittels Profilen personalisiert werden. Die Administration von DCMS ist von einem einzelnen Punkt aus möglich und daher sehr komfortabel.

Auswirkungen

Peer-to-Peer-Systeme erschließen in Zusammenarbeit mit CM völlig neue Dimensionen des Wettbewerbs durch (Echtzeit)Integration vorher unerreichbarer Inhalte und Quellen. Durch die immer stärkere Zusammenarbeit verwischen Unternehmensgrenzen zusehends, Produkte werden verbessert und der Endverbraucher profitiert letztendlich davon. Auch haben die Nutzer neue Möglichkeiten bei der Suche nach Informationen, doch dazu später mehr.

2.2. Anwendungen

Web Content Management

Web-Content-Management-Systeme (WCMS) wurden speziell für das Management von Webseiten entworfen. Sie ermöglichen den direkten Zugriff auf verteilte Inhalte und Quellen bei gleichzeitiger Nutzung eines zentralen Repositorys. Dabei sind die Quellen der Inhalte durch einen Workflow verbunden, der die Indexierung und den Zugriff auf die Daten durchführt. Daher gelten gewisse Einschränkungen in Bezug auf Zugriff und Aktualisierungen. Es kann zum Beispiel eine Weile dauern, neue Daten auf den Peers mit dem zentralen Repository zu synchronisieren, besonders wenn der Workflowprozess überlastet ist. Dadurch ist es auch oft nicht möglich, Suchanfragen in Echtzeit zu beantworten. Auch kann es schwierig sein bestimmte Arbeitsumgebungen in den Workflow zu integrieren, Peers außerhalb der Firewall eines Unternehmens müssen unter Umständen ebenfalls außen vor bleiben. Ein weiteres Problem ist, dass einige wichtige Informationen, aus welchen

Gründen auch immer, nicht im zentralen Repository enthalten sein können. Zielstellung von WCMS ist die Trennung von Inhalt und Design, Flexibilität, Geschwindigkeit und Skalierbarkeit. Ein verteiltes CMS kann ein virtuelles Repository als Basis des WCMS bereitstellen.

Globales WCM

Große internationale Firmen haben bei ihren Webseiten oft das Problem, dass die Inhalte in verschiedenen Versionen bezüglich Sprache, Kultur oder Wirtschaft vorliegen. Dies impliziert die Verwendung mehrerer redundanter Repositorys an verschiedenen Standorten, was jedoch durch die Bereitstellung eines globalen virtuellen Repository durch ein verteiltes CMS umgangen werden kann. Dabei müssen lediglich die Prozesse zur Lokalisierung beachtet werden.

P2P Content Networks

P2P gilt weithin als nächste „Killerapplikation“ für das Internet und insbesondere auch für CMS, stellt die Entwickler aber auch vor neue Probleme, wie zum Beispiel Sicherheitsfragen. Dabei kann man Peer-to-Peer-Systeme grundsätzlich in zwei Gruppen einteilen: *formale* und *informale* Systeme. Bei formalen Systemen erfolgt eine strenge Zugriffskontrolle auf Client- und Serverebene, sie basieren auf einfach zu erstellenden und zu überwachenden Netzwerkprotokollen und alle Anwendungen folgen einem vorgegebenen Workflow. Bei informalen Systemen ist dagegen ein unkontrollierter Zugriff auf sämtliche Ressourcen möglich. Natürlich eignen sich nur höchst formale Systeme für CM. Weiterhin lassen sich fünf Modelle für Peer-to-Peer-Systeme unterscheiden:

- Das *atomare Modell* umfasst die reinen Peer-to-Peer-Systeme ohne zentrale Komponente.
- Das *nutzerzentrierte Modell* verwendet ein Nutzerverzeichnis (Directory), um zu erfassen wer online ist. Dieses Directory kann auf einem zentralen Server liegen oder auf den Peers verteilt sein. Wer den Server mit dem Verzeichnis bereitstellt, hat damit auch einen Überblick wer wann wie lange online ist (und könnte diese Informationen irgendwie verwerten).
- Das *datenzentrierte Modell* ist besonders für die Suche und den Zugriff auf Daten anderer Peers geeignet und wird als das vielversprechendste Modell betrachtet. Ein Index aller verfügbaren Ressourcen befindet sich auf einem zentralen Server, über den auch die Suche abläuft. Dadurch

ist eine einfache Kontrolle über Inhalte (zum Beispiel Dateitypen) und Zugriffe möglich. Es wird prognostiziert, dass dieses Modell in den meisten CMS zum Einsatz kommen wird.

- Das *Web Mk 2* Modell bildet eine Mischung der drei vorigen Modelle unter Nutzung der gegenwärtigen Webarchitekturen und Infrastruktur. Browser werden sich zu extrem flexiblen, konfigurierbaren Arbeitsumgebungen entwickeln, die sämtliche Funktionalität der bisherigen Modelle in sich vereinigen. Mehrere Verzeichnisdienste werden Nutzer ad-hoc parallel mit anderen Nutzern verbinden und so den gleichzeitigen Zugriff auf verschiedenste Inhalte erlauben. Bei informalen Systemen reichen heutige Browser teilweise schon für einige Aufgaben aus, die formalen Systeme sprengen jedoch deren Fähigkeiten. Web Mk 2 Applikationen haben das Potenzial die aktuellen Onlinedienste, Portale, Internetshops usw. zu revolutionieren, einige Experten sind der Meinung der Umstieg wäre noch drastischer als die Einführung des Internet.
- Das *Compute Centered Model* (verteilte Berechnungen) kann vor allem zur verteilten Indexierung von Inhalten eingesetzt werden, um den Indexserver zu entlasten. Dies ist auch in mehreren Hierarchiestufen denkbar. Zur verteilten Datenverarbeitung eignet sich dieses Modell wegen den unter Umständen vertraulichen Informationen nicht.

Am ehesten eignen sich das datenzentrierte Modell und das Web Mk 2 Modell für CMS, es gibt jedoch noch einige Herausforderungen und Entwicklungen zu meistern:

- Die *Vermischung interner und externer Inhalte* (Newsgroups, Foren, Chats) wird erforderlich, da Unternehmen sich keine Isolation mehr leisten können. Dazu können Indexserver innerhalb und außerhalb des Unternehmens anhand gewisser Suchkriterien nach Informationen suchen. Diese Variante ist leicht umzusetzen jedoch schlecht skalierbar. Die zweite Möglichkeit ist die Verwendung der Peers als private Indexserver. Diese können auf vorhandene Dienste (Yahoo, AltaVista) zurückgreifen und bieten gute Skalierbarkeit, benötigen jedoch eine nicht unerhebliche Bandbreite.
- *Persistente Suchanfragen* werden auf Indexserverebene in Echtzeit oder mittels Stapelverarbeitung durchgeführt und ermöglichen somit eine stärkere Kontrolle. Sobald neue Ergebnisse vorliegen kann der Nutzer auf verschiedenen Wegen benachrichtigt werden, zum Beispiel über ein beliebiges netzwerkfähiges Gerät, das ins Peer-to-Peer-Netz eingeklinkt ist. Persistente Suche basiert auf dem Web Mk 2 Modell, und da es durchaus mehrere Indexserver geben kann, ist deren Management von besonderer Bedeutung. Ein Beispiel für einen ersten

Ansatz sind automatische Benachrichtigungen über neue E-Mails.

- *Personal Indices* erleichtern den Durchblick bei der Vielzahl der, teilweise überflüssigen, Informationen, die so gefiltert werden. Zunächst werden diese Indices auf dem Server abgelegt, das datenzentrierte Modell legt jedoch eine Speicherung auf den Peers nahe. Dadurch sind die Indices auch offline verfügbar und der Server wird entlastet, die Synchronisation mit der persistenten Suche erfolgt bei der nächsten Verbindung zum Server.
- *Content Communities* werden gebildet, wenn sich mehrere Peers einen persönlichen Index teilen, weil sie gemeinsame Suchkriterien haben (Mitarbeiter derselben Abteilung oder Nutzer mit gleichen Interessen). Dadurch werden viele Duplikate bei der Suche vermieden. Denkbar ist eine Art Subscription in öffentliche Indices.
- Bei *Content Consolidation* werden die Ergebnisse von Suchanfragen auf verschiedenen Indices verglichen und zusammengefasst. Dies wird schon auf Serverebene praktiziert, die Verlagerung auf die Peerebene würde dem Nutzer mehr Kontrolle ermöglichen. Ein großes Problem ergibt sich hier bei der Vermischung von geschützten Inhalten.
- *Lebende Dokumente* sind Dokumente, die Verweise auf verschiedenste Personen enthalten, die mit dem Dokument zu tun haben. Dazu zählen Autor, Eigentümer oder Personen die das Dokument bearbeitet, gelesen bzw. Vorträge und Diskussionen darüber gehalten haben.

Folgerungen

DCM und WCM ermöglichen den Zugriff auf nahezu sämtliche Inhalte innerhalb eines Unternehmens und zwischen Unternehmen, vorausgesetzt alle Peers sind entsprechend vernetzt. DCMS sind hervorragend zur Ergänzung von WCMS geeignet, dabei sind WCMS strikter beim Management und der Darstellung der Inhalte, während DCMS alle Funktionen in Echtzeit durchführen können. Peer-to-Peer-Systeme werden die CMS weiter beeinflussen, insbesondere P2P Content Networks können Unternehmen (auch im Zusammenspiel mit WCM) erhebliche Vorteile bringen. Wenn es vorrangig um den Zugriff auf verteilte, unternehmenskritische Inhalte ohne deren zentrale Verwaltung geht, eignen sich DCMS am besten. Wenn jedoch der Echtzeitzugriff von geografisch verteilten Standorten wichtig ist, sind P2P Content Networks die erste Wahl. Ein ideales CMS gibt es nicht, verschiedene Systeme erfüllen unterschiedliche Anforderungen. In Zukunft werden sie sich aber stärker überschneiden (siehe Abbildung 1).

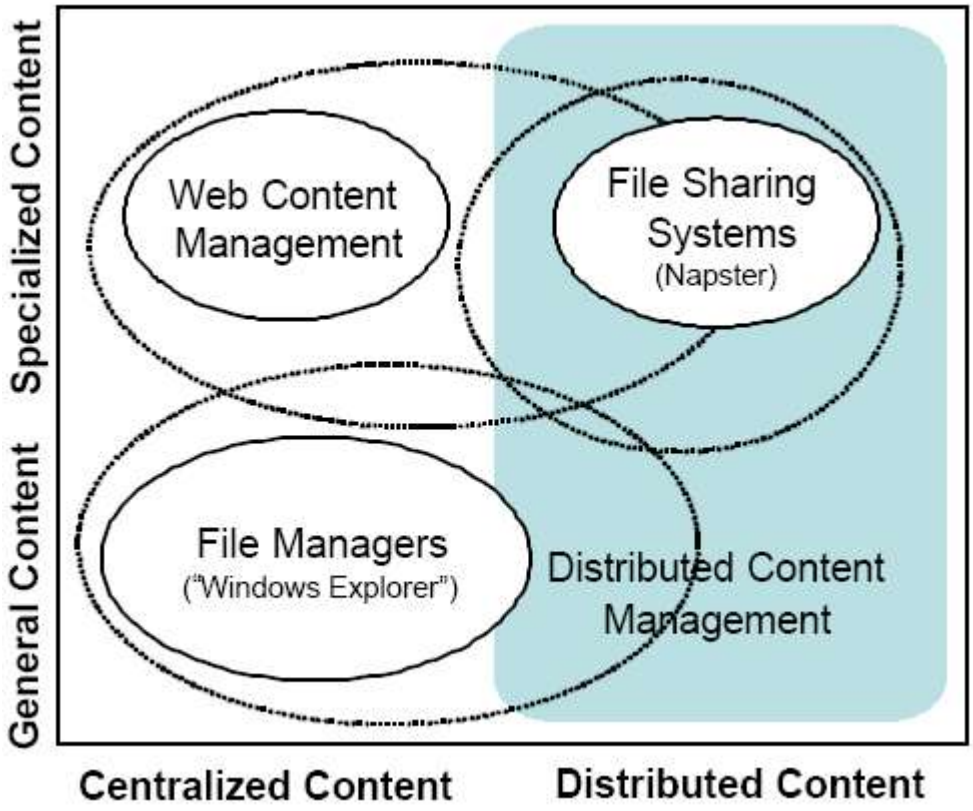

Abbildung 1: Das ideale CMS ? (aus [HS03])

3. Knowledge-Management

Aufgrund der wachsenden Globalisierung, Expansion und Vernetzung entschließen sich immer mehr Unternehmen und Organisationen zum Einsatz von Knowledge-Management-Systemen (KMS), um ihre Ziele effektiver zu Erreichen und sich von den Wettbewerbern abzuheben. Sie erkennen langsam, dass dem individuellen Wissen einzelner Mitarbeiter und Abteilungen eine größere Bedeutung beigemessen werden muss, um sich im harten Konkurrenzkampf zu behaupten. Das größte Problem der heute gebräuchlichen KMS ist jedoch ihre *zentralisierte* Architektur, wohingegen der Prozess der Wissensgenerierung durch die wachsende Bedeutung der *sozialen* Aspekte weitgehend *verteilt* ist. Ebenso spielen Religion und kulturelle Eigenheiten an bestimmten Standorten eine nicht zu vernachlässigende Rolle. Somit wird klar, dass die technologische und die soziale Architektur wechselseitig konsistent sein müssen, und das Wissen oft nicht mehr in ein zentrales und starres Schema gezwungen werden kann. Dies hängt vor allem auch davon ab, wie stark der Mensch am Prozess der Wissensgenerierung beteiligt ist.
In diesem Kapitel werde ich daher auf die Unterschiede zwischen zentralen und verteilten KMS eingehen und das Zusammenspiel mit Peer-to-Peer-Systemen darstellen. Näheres dazu wird in [BCM+02] beschrieben.

3.1. Zentralisierte KMS

Charakterisierung

Zentralisierte KMS sind für gewöhnlich computerbasierte Systeme bestehend aus einer *Knowledge Base* (KB) und einem *Enterprise Knowledge Portal* (EKP), die zum Zwecke der Kommunikation und des Austausches von Wissen geschaffen wurden. Sie speichern das Wissen an einem zentralen Ort in einem großen *Repository*, welches im einfachsten Fall ein Dateisystem ist, aber auch eine Datenbank sein kann. Das gesamte Wissen wird nur anhand einer einzigen Metastruktur, zum Beispiel einer Ontologie, einer Knowledge Map, einer Kategorisierung, Klassifizierung oder einer speziellen Sprache, erfasst. Diese legt fest, wie das Wissen zu interpretieren und zu deuten ist, und ermöglicht auf einfache Weise den Austausch des Wissens innerhalb der gesamten Organisation. Aufgrund der zentralen Architektur muss vorausgesetzt werden, dass alle kontextsensitiven,

subjektiven und sozialen Aspekte des Wissens durch eine einzige objektive, generelle Sichtweise ersetzt werden können. Dies steht allerdings im Gegensatz zu den verbreiteten Wissenstheorien, nach denen Wissen das Ergebnis einer subjektiven Interpretation eines Ausschnittes der Welt durch Einzelpersonen oder Gruppen infolge sozialer Interaktionen ist.
An die dem KMS zugrundeliegende Technologie wird ebenfalls eine Reihe von Anforderungen gestellt, um das Sammeln des „peripheren" Wissens von den Knoten optimal zu unterstützen. Dazu gehören unter anderem zentrale Kontrolle über das Wissen, Standardisierung, hohe Kapazitäten und Performance sowie Sicherheit und Robustheit.

Arbeitsweise

Einer der ersten Schritte ist die Erstellung der Metastruktur (zum Beispiel eine der oben genannten) zur Erfassung des Wissens. Diese kann je nach Situation sehr komplex aber auch restriktiv werden, was zu einem weiteren Problem führt: die Nutzer verstehen das Schema nicht oder finden es unpassend für ihre Bedürfnisse und lehnen es schlichtweg ab, das heißt sie nutzen das Wissen und die Möglichkeiten des KMS einfach nicht. Desweiteren muss die Kommunikation innerhalb von organisatorischen Einheiten des Unternehmens wie Gruppen oder Communities ermöglicht werden. Dabei interagieren Individuen direkt (durch praktische Zusammenarbeit) oder über eine Reihe von Tools und Mechanismen (Virtual communities, Groupware) miteinander und generieren dabei sogenanntes „Rohwissen". Dieses „Rohwissen" wird nun von allen Beteiligten mittels Dokumentenmanagementsystemen, Datamining oder Information Retrieval in die KB eingebracht wo es automatisch oder manuell (von entsprechenden Experten) bewertet und anhand des zentralen Schemas in das Repository aufgenommen wird. Als letzter wichtiger Schritt bleibt die Erschaffung eines EKP, um den diversen organisatorischen Einheiten einen einheitlichen sowie gleichzeitig einfachen Zugriff auf das Wissen zu ermöglichen. Dies kann durch Profile, Chats, Sichten usw. geschehen, jedoch auch in Form von neuen Handbüchern, Fortbildungsveranstaltungen und Arbeitsrichtlinien.

Folgerungen

Aufgrund der Funktionsweise von zentralisierten KMS eignen sie sich vorwiegend für den Einsatz in Client-Server-Architekturen. In reinen Peer-to-Peer-Systemen können sie wegen der fehlenden

zentralen Komponente nicht eingesetzt werden, in hybriden Peer-to-Peer-Systemen zumindest in abgewandelter Form oder mit Einschränkungen. Dazu müsste es zum Einen möglich sein, ein einheitliches Schema für das Wissen zu finden, und zum Anderen würde auf den in hybriden Systemen vorkommenden Superpeers lediglich das Wissen über den Ort einer benötigten Ressource gespeichert werden, während die Ressource selbst auf einem Peer verbleibt.
Ein bekanntes Beispiel hierfür wäre Napster (in der ersten Version). Der zentrale Server hält eine ständig aktuelle Liste aller im Netz vorhandenen Dateien zur Verarbeitung von Anfragen bereit, die Dateien befinden sich jedoch auf den einzelnen Peers und werden bei Bedarf direkt zwischen diesen transferiert. Auch gibt es ein recht einfaches Schema zur Repräsentation des Wissens, da dieses auf MP3- und WAV-Dateien beschränkt ist. Eine Bewertung erfolgt dabei automatisch durch entsprechende Filterung.

3.2. Verteilte KMS

Um die im vorigen Abschnitt geschilderten Probleme eines komplexen Schemas und der starken Verteilung von Gruppen und Individuen zu lösen, wurde ein alternativer Ansatz von Knowledge-Management entwickelt: Distributed Knowledge-Management (DKM). Dabei sollte das Wissen als System von lokalen Wissensbasen dargestellt werden, welches fortlaufend von den einzelnen Gruppen und Individuen gepflegt wird. Außerdem sollte DKM auf Unterstützung und Ausbalancierung allgemeiner organisatorischer Prinzipien aufbauen:

- **Prinzip der Autonomie:** jede Community ist unabhängig bezüglich ihres lokalen KM
- **Prinzip der Koordination:** die Communitys tauschen Wissen aus, jedoch nicht durch Adaption eines einzelnen Schemas, sondern durch gewisse Übersetzungsmechanismen zwischen verschiedenen interpretierbaren Kontexten – sogenannte semantische Interoperation

Charakterisierung

Verteilte Knowledge-Management-Systeme (DKMS) ermöglichen die Integration verschiedener unabhängiger KMS sowie die Koordination des Wissensaustausches zwischen diesen (siehe dazu auch Abbildung 2).

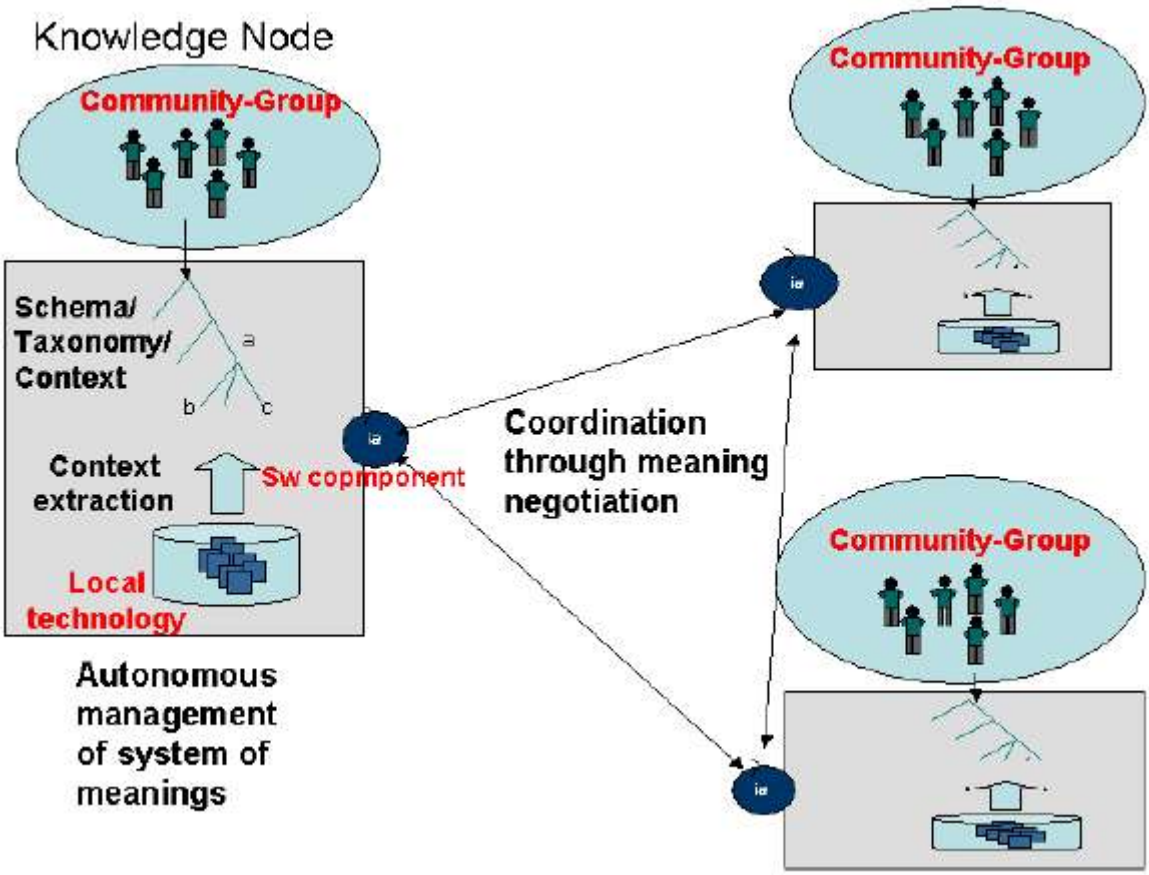

Abbildung 2: DKM Architektur nach [BCM+02]

Desweiteren werden bei DKM Subjektivität und soziale Aspekte als unverzichtbare Werte betrachtet und nicht als Probleme, die es zu lösen gilt. DKMS legen kein zentrales Repository an – das Wissen verbleibt bei den sogenannten Knowledge Nodes (KN). Diese repräsentieren Individuen, Gruppen und Communitys, aus denen sich Unternehmen und Organisationen (aus sozialer Sicht) zusammensetzen, und bilden die Grundbausteine von DKMS. Ob ein Unternehmen vorrangig auf Communitys oder auf Individuen als KN setzt, hängt im Einzelfall auch von den jeweiligen technischen Gegebenheiten ab. KN besitzen einen gewissen Grad an semantischer Unabhängigkeit, was ihnen die Fähigkeit gibt, eigene Schemata zum Management ihres lokalen Wissens zu erstellen (meist Klassifizierungen). Gruppen und Communitys haben außerdem die Möglichkeit der Erstellung gemeinsam genutzter lokaler Repositorys. Das lokale KM bedient sich Mechanismen wie Validierung des Wissens und muss Prozesse zum erfassen von neuem Wissen implementieren. Die Koordination und der Wissensaustausch zwischen verschiedenen KN erfolgt hauptsächlich mit Hilfe zweier logischer Elemente: Kontext und Verbindungstabelle (Connection Table). Kontext dient dazu, anderen KN das eigene Schema zur Interpretation des Wissens

verständlich zu machen, während die Verbindungstabelle auf einem KN das Wissen einem geeigneten Kontext zuordnet. Dieser Prozess der Einschätzung und des Vergleichs von Bedeutungen heißt auch „meaning negotiation process“. Um den Kommunikationsaufwand zu minimieren und gleichzeitig die Effektivität zu maximieren sind noch einige zusätzliche organisatorische Fähigkeiten nötig:

- **Föderalismus:** Mehrere KN mit gemeinsamen Interessen schließen sich zu einer Gruppe zusammen. So können sie leichter von anderen KN gefunden werden, ihr Wissen mittels Zugriffsberechtigungen schützen oder die Qualität des Wissens durch Aufnahmebeschränkungen erhöhen. Außerdem wird die Anzahl der Zugriffe von anderen KN vermindert.
- **Entdeckung:** KN sollten jeden anderen KN finden können, um herauszufinden wer (Peer) und was (Wissen, Service) verfügbar ist, und ob und wie eine Verbindung aufgebaut werden kann. Dieses Wissen an sich muss wiederum auf die KN verteilt sein.
- **Propagierung:** KN sind in der Lage Anfragen an benachbarte Knoten weiterzuleiten, damit der anfragende Knoten nicht mit jedem KN des Netzwerks direkt kommunizieren muss und eine Anfrage nach „relativ“ wenigen Schritten ein Ergebnis liefert.
- **Lernen:** Wenn einmal ein meaning negotiation process zwischen zwei KN stattgefunden hat, muss dieser nicht ein zweites Mal durchgeführt werden. Dieses Wissen kann auch zur Weiterleitung von Anfragen an geeignete Nachbarknoten genutzt werden.

Zur Durchführung sämtlicher geschilderter Aufgaben benötigt ein DKMS eine Reihe komplexer Protokolle und Algorithmen, die teilweise nur mit menschlicher Beteiligung ablaufen. Dazu gehören unter anderem Information Retrieval, Sprachverarbeitung, deduktives und induktives Schließen.

Folgerungen

Offenbar sind Peer-to-Peer-Systeme wie geschaffen für die Umsetzung von DKM. Sie sind dynamisch, dezentralisiert und alle Peers sind gleich und unabhängig. Die weit verbreiteten Filesharingsysteme erfüllen viele der Anforderungen an DKM, jedoch in unterschiedlichem Maße und oft mit starken Vereinfachungen. So verwenden alle Peers dasselbe Schema, was den meaning negotiation process überflüssig macht, und auch das lokale KM beschränkt sich auf die manuelle

Freigabe bestimmter Dateien. Ebenso spielen soziale Aspekte keine Rolle. Darüberhinaus ist bei Suchanfragen nicht bei jedem System sichergestellt, dass alle Ressourcen auch gefunden werden. Auf ein spezielles Beispiel für DKM, das Edutella-Projekt, werde ich später ausführlicher eingehen.

3.3. Entwicklung von DKMS

In diesem Abschnitt werde ich den Ansatz eines Framework zur Entwicklung von Knowledge-Management Anwendungen aus [BBM+02] erläutern, der im Wesentlichen auf der Tropos-Methodik aufbaut. Die Tropos-Methodik adaptiert einen agentenorientierten Ansatz der Softwareentwicklung von der ersten Spezifikation bis hin zur Implementierung. Den Kern bildet die konzeptionelle Modellierung mittels einer visuellen Sprache, welche intentionale und soziale Konzepte wie Akteur, Rolle, Anschauung, Ziel, Plan, Ressource und Abhängigkeit unterstützt. Man unterscheidet zusätzlich in Hard- und Softgoals. Zur Veranschaulichung der Funktionsweise werde ich die Ergebnisse der Tropos-Analyse für Gnutella und Napster darstellen.

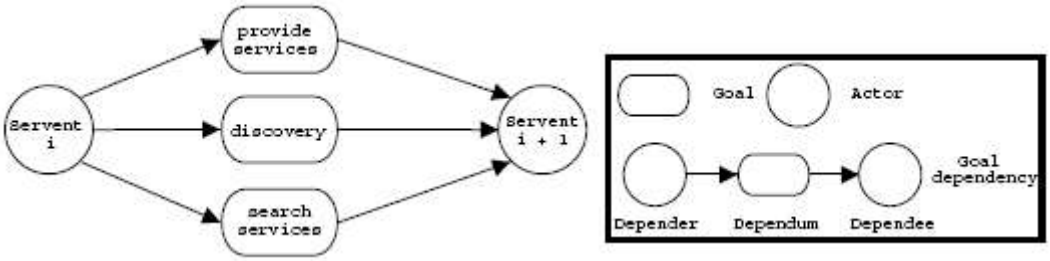

Abbildung 3: Gnutella – Architektur nach Tropos ([BBM+02])

Servent ist die Gnutella-Bezeichnung für Peer. Jeder Servent hängt von jedem anderen ab, um die dargestellten Ziele zu erreichen. Durch *discovery* bleibt das Netzwerk dynamisch, da ein Peer kommen und gehen kann wann er will, und sich nirgends registrieren muss. Es gibt keine zentralen Dienste und jeder Peer bietet die gleichen Dienste an.

Bei Napster haben die Peers ähnliche Ziele wie bei Gnutella, statt *discovery* muss sich jedoch jeder Peer mitsamt seinen Dateien bei einem zentralen Server registrieren und über diesen auch Suchanfragen stellen. Somit hängen die Peers nicht nur von anderen Peers ab, sondern auch vom Server, und dieser wiederum hängt von den Peers ab, da er ohne sie „arbeitslos“ ist.

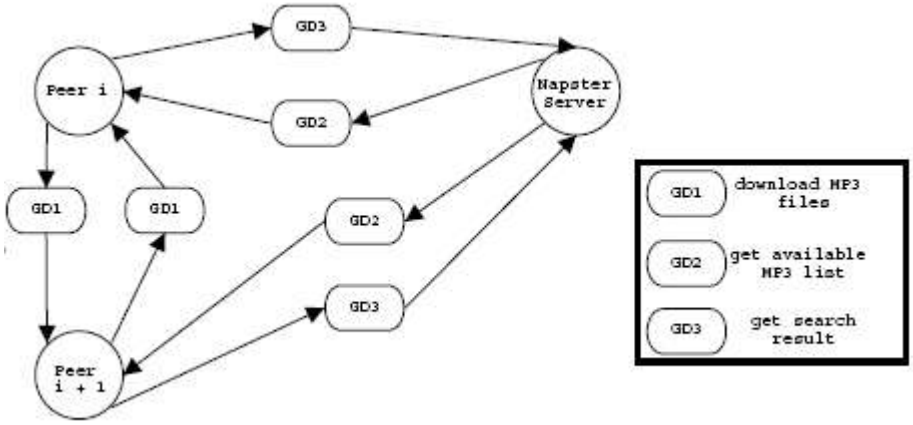

Abbildung 4: Napster - Architektur nach Tropos ([BBM+02])

In beiden Fällen können die Peers ihre Ziele erreichen, indem sie Peer-Communitys bilden (bei Napster zusätzlich koordiniert durch einen Server). Abstrahiert man diese und andere Modelle erhält man folgendes Modell einer virtuellen P2P-Community:

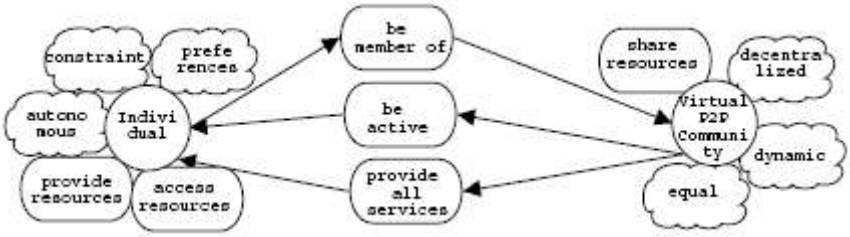

Abbildung 5: P2P virtual community ([BBM+02])

Jedes Individuum hat mindestens zwei Ziele: Ressourcen anbieten und nutzen. Dazu kommen noch Softgoals wie Autonomie, die das Verhalten des Individuums beschreiben (Kontrolle darüber, *welche* Ressourcen *wann* genutzt oder angeboten werden). Die virtuelle P2P-Community hat das Hauptziel, seine Mitglieder Ressourcen austauschen zu lassen, und als Softgoals Dynamik, Dezentralisierung und Gleichheit aller Peers. Die Peers können ihre Ziele nur in der Community erreichen, die Community ist nur aktiv wenn Peers vorhanden sind und nur dann können alle die gleichen Dienste anbieten, was man an den Abhängigkeiten erkennen kann.
Betrachtet man nun dieses Modell, dann erfüllt Gnutella alle diese Anforderungen perfekt, während Napster sich wegen des zentralen Servers disqualifiziert. Man kann sagen, dass die Erzeugung von Peergroups sich aus der Notwendigkeit herleitet, eine Menge von Peers zu definieren, die in der Lage sind über gemeinsame Fähigkeiten und Protokolle zu kommunizieren und zu interagieren, und welche sich aus Sicherheitsgründen zusammenschließen (Schutz vor Einmischung fremder Peers).

4. Das Edutella-Projekt

Metadaten zur Beschreibung von Ressourcen sind von essentieller Bedeutung für Peer-to-Peer-Systeme. Dies ist für einige Spezialfälle wie Filesharingsysteme relativ simpel, für universell einsetzbare Anwendungen jedoch nicht, weshalb die meisten aktuellen Peer-to-Peer-Systeme eher Nischenprodukte sind, die kaum für zukünftige Aufgaben genutzt werden können. Genau hier setzt Edutella an: ursprünglich zum Austausch von Lehr- und Lernmaterialien entwickelt, ist es inzwischen ein Projekt zur Spezifikation und Implementierung einer auf RDF basierenden Metadaten-Infrastruktur für Peer-to-Peer-Systeme, welche ihrerseits auf JXTA aufbauen, geworden. Die wichtigsten Services von Edutella sind Query, Replication, Mapping, Mediation und Annotation Services, die die Interoperabilität verschiedener auf JXTA basierender Anwendungen ermöglichen sollen. Daher wollen die Entwickler neben dem ursprünglichen Zweck später weitere Applikationen entwickeln. In diesem Kapitel werde ich auf die Grundlagen von Edutella und die wichtigsten Services eingehen, wie sie in [NWQ+02] beschrieben sind.

4.1. JXTA

Beim JXTA Framework handelt es sich um ein Open Source Projekt von Sun Microsystems, welches eine Reihe von Protokollen auf XML-Basis definiert, die alle wichtigen Funktionen von Peer-to-Peer-Systemen abdecken. Dazu gehören neben dem Zugriff auf entfernte Ressourcen auch Peer Discovery, Peer Groups, Peer Pipes und Peer Monitoring. Sie bilden den Kern von JXTA.

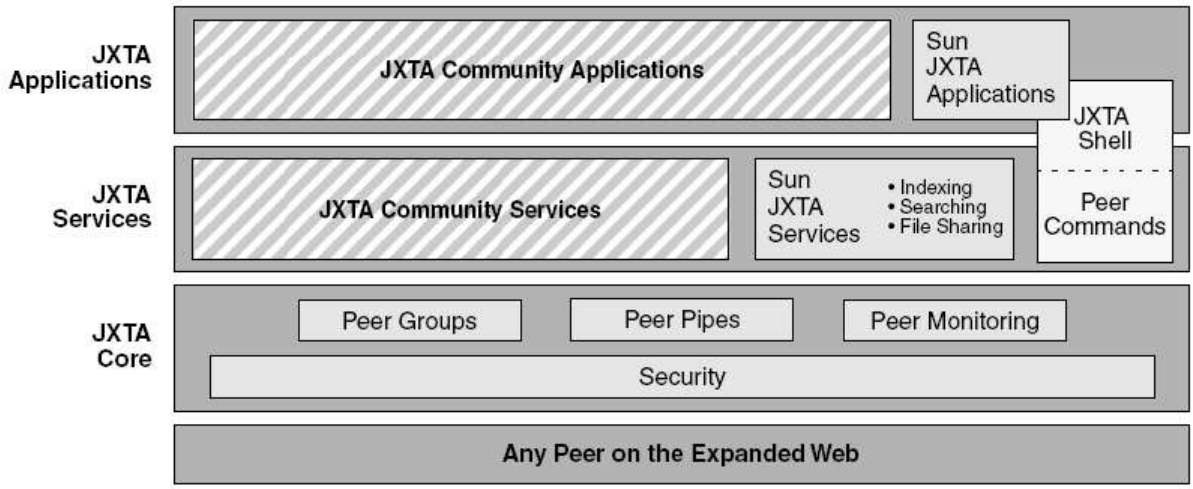

Abbildung 6: Das JXTA Framework nach [NWQ+02]

Wie in Abbildung 6 ersichtlich, ist das JXTA Framework aus mehreren Schichten aufgebaut und passt daher sehr gut zu Edutella: die Edutella Services ergänzen die JXTA Service Schicht während die Edutella Peers sich auf der Anwendungsschicht bewegen und sowohl Edutella Services als auch andere JXTA Services nutzen können. In der Edutella Service Schicht werden dazu Datenformate und Protokolle definiert, die beschreiben wie Querys, Queryergebnisse oder andere Metadaten zwischen Peers ausgetauscht werden.

4.2. Motivation

Viele Universitäten oder ähnliche Einrichtungen besitzen große Mengen an Wissen und Daten, die auch anderen Universitäten nutzen würden, wenn sie Zugriff darauf hätten. Die Bereitstellung von Servern zu diesem Zweck ist jedoch sehr kostspielig, sodass die Meisten davor zurückschrecken. Die scheinbar beste Lösung hierfür ist ein Peer-to-Peer-System. Die weit verbreiteten Filesharingsysteme verwenden jedoch eine recht einfache *Meta-Auszeichnungssprache*, mit der lediglich Dateiattribute (Name, Pfad, Typ, Größe usw.) erfasst werden können. Auch sind sie zueinander inkompatibel. Edutella hingegen verwendet eine sehr komplexe Auszeichnungssprache aufgrund der Vielfältigkeit der Ressourcen, was hohe Anforderungen an das System und die Technologie stellt. Außerdem soll das System leicht anpassbar sein und unterschiedlichste Peers (bezüglich Performance, Kapazität, Funktionalität, Anzahl der Nutzer, Verfügbarkeit usw.) mit eigenen Anfragesprachen und Funktionen sowie verschiedene Metadatenschemata integrieren können. Dazu ist es wichtig, dass alle Ressourcen in RDF beschrieben werden können, und alle Funktionen mittels RDF-Anfragen ausgeführt werden können. Für die Nutzer bleibt das Netzwerk dennoch durchschaubar und sie können verschiedene Clientsoftware verwenden. Diese muss lediglich einige Basisdienste bereitstellen und kann darüberhinaus zusätzliche Dienste optional anbieten.

4.3. Services

Jeder Peer im Edutella Netzwerk wird durch seine implementierten Services charakterisiert. Der wichtigste Service ist der *Query Service*, der für standardisierte Querys und Ergebnisse zuständig ist. Näheres dazu folgt später in diesem Kapitel. Der *Replication Service* dupliziert Metadaten in anderen Peers, um Persistenz, Verfügbarkeit und Loadbalancing zu gewährleisten, während

Datenintegrität und Konsistenz erhalten bleiben. Das Duplizieren der eigentlichen Daten geschieht vorerst nicht, weil dadurch auch Updatevorgänge schwieriger zu synchronisieren wären. Der *Mapping Service* dient zur Übersetzung von Querys zwischen verschiedenen Schemata und der Interoperabilität von RDF- und XML-Repositorys. Der *Mediation Service* koordiniert die Vereinigung von Daten aus verschiedenen Quellen und der *Clustering Service* ist in der Lage semantisches Routen durchzuführen und semantische Cluster zu bilden. Der *Annotation Service* dient schließlich der Ergänzung von Metadaten zu Dokumenten. Diese können einfache Formularfelder mit Werten sein (ohne erkennbaren Zusammenhang für maschinelle Verarbeitung), Auszüge aus dem Dokument oder Verweise auf bestimmte Dokumentteile (mit XPointer).

4.4. Der Query Service

Der Edutella Query Service stellt ein standardisiertes Interface zum Austausch von Querys für RDF-Metadaten bereit. Er unterstützt sowohl Querys auf RDF-Repositorys (beschreibt Metadaten anhand beliebiger RDF-Schemata mittels RDF-Statements) einzelner Peers als auch verteilte Querys auf RDF-Repositorys verschiedener Peers. Ziel ist es, mit der Edutella Query Exchange Language (QEL) und dem Edutella Common Data Model (ECDM) einen über bekannten Anfragesprachen wie SQL stehenden Standard zu schaffen, um über vollkommen heterogene Peers und Repositorys auf beliebige RDF-Metadaten zuzugreifen. Dazu verwendet Edutella das *RDF-QEL-i*-Format, welches seinerseits im RDF/XML-Format übertragen wird. Hierzu ein Beispiel:

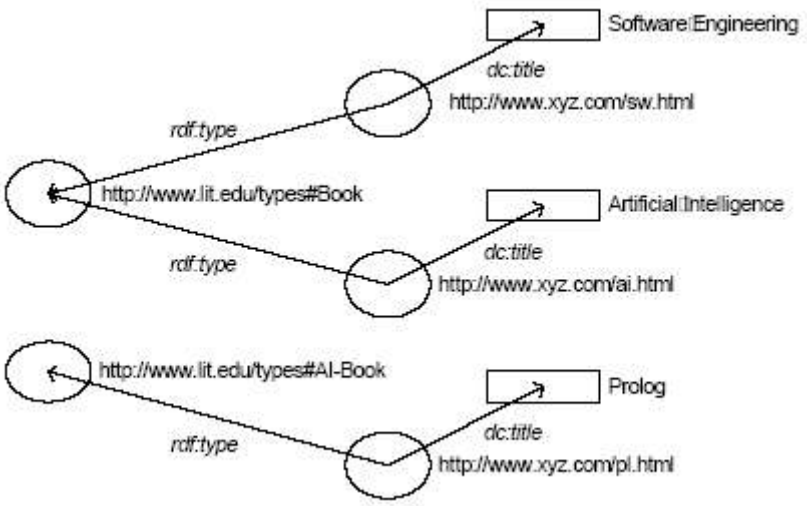

Abbildung 7: KB als RDF-Graph (aus [NWQ+02])

Im RDF/XML-Format sieht diese KB dann wie folgt aus:

```
<lib:Book about="http://www.xyz.com/sw.html">
     <dc:title>Software Engineering</dc:title>
</lib:Book>
<lib:Book about="http://www.xyz.com/ai.html">
     <dc:title>Artificial Intelligence</dc:title>
</lib:Book>
<lib:AI-Book about="http://www.xyz.com/pl.html">
     <dc:title>Prolog</dc:title>
</lib:AI-Book>
```

Wertet man nun folgende Query aus: „Finde alle Bücher mit dem Titel 'Artificial Intelligence' und alle AI-Bücher“, so erhält man folgendes Ergebnis:

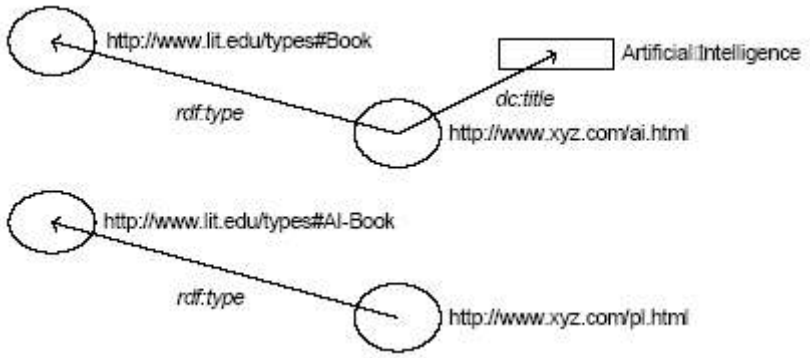

Abbildung 8: Ergebnisse der Query als RDF-Graph (aus [NWQ+02])

Die Umwandlung der Querys und Ergebnisse in das lokale Format eines jeden Peers und wieder zurück übernehmen die Edutella *Wrapper*. Sie haben auch die Aufgabe, den Peer mit dem Netzwerk auf Basis von JXTA-Funktionen zu verbinden. Intern werden sämtliche Querys und Ergebnisse durch ein auf *Datalog* basierendes Modell repräsentiert. Jede Query kann damit als logische Formel angesehen werden, die es mit Hilfe der KB zu beweisen gilt. Damit sieht das obige Beispiel so aus:

```
title(http://www.xyz.com/ai.html,'Artificial Intelligence').
type(http://www.xyz.com/ai.html,Book).
title(http://www.xyz.com/sw.html,'Software Engineering').
type(http://www.xyz.com/sw.html,Book).
title(http://www.xyz.com/pl.html,'Prolog').
type(http://www.xyz.com/pl.html,AI-Book).
```

Die Query hat nun folgende Gestalt:

```
aibook(X) :- title(X, 'Artificial Intelligence'),type(X, Book).
aibook(X) :- type(X, AI-Book).
?- aibook(X).
```

Und das Ergebnis sieht wie folgt aus:

```
aibook(http://www.xyz.com/ai.html)
aibook(http://www.xyz.com/pl.html)
```

Wie schon erwähnt verwendet Edutella das RDF-QEL-i-Format zum Austausch von Querys und Ergebnissen. Dieses Format besteht aus mehreren Ebenen, welche die Art der Querys bestimmen, die ein Peer verarbeiten kann. Dazu werden eine Reihe von Anforderungen an die Sprache gestellt:

- **Standardsemantik:** Eine einfache Standardsemantik ist wichtig um bei der Überführung von Querys durch die Wrapper die ursprüngliche Bedeutung zu erhalten. Dazu gesellt sich eine möglichst stimmige Codierung ins RDF/XML-Format.
- **Ausdrucksstärke:** Es soll sowohl möglich sein, einfach gestrickte Querys (zum Beispiel mit grafischen Tools) zu erstellen, als auch die Mächtigkeit anderer Sprachen wie SQL voll auszuschöpfen.
- **Anpassbarkeit:** Die Sprache soll neutral gegenüber allen Semantiken sein und jedes Prädikat verarbeiten können, dass eine Semantik besitzt. Es soll einfach möglich sein, Verbindungen zu RDFS-Repositorys, relationalen und objektorientierten Datenbanken aufbauen zu können.
- **Transformierbarkeit:** Die QEL soll möglichst einfach in verschiedenste andere Sprachen transformiert werden können, um die Arbeit bzw. Implementierung der Wrapper zu erleichtern.

Mit Hilfe der QEL lassen sich jedoch nicht nur Informationen über Ressourcen gewinnen, sondern auch Informationen über das verwendete RDF-Schema selbst. Dies unterscheidet sich nicht von den anderen Querys, da das RDF-Modell keinen Unterschied zwischen Schema und Daten macht. Es ist lediglich erforderlich, die Syntax der Querys auf ein Triple-Format umzustellen. So wird aus „title (X,'Artificial Intelligence')“ nun „s(X,title,'Artificial Intelligence')“.
Ergebnisse auf Anfragen können auf verschiedene Weise dargestellt werden. Bestehen sie aus Tupeln von Variablen und deren Belegungen, reicht eine einfache RDF/XML-Struktur aus. Es ist aber auch möglich, die Ergebnisse als Objekte zurückzuliefern, wie zum Beispiel einen RDF-Graph. Das besondere an einem solchen Graph ist, dass er wieder als Query abgeschickt werden kann, und dann dieselben Ergebnisse liefert wie die ursprüngliche Anfrage.

4.5. Der Edutella Prototyp

Es gibt bereits eine Reihe von Beispielapplikationen zum Edutella-Projekt, unter anderem:

- **Edutella Dublin Core Consumer:** Dies ist ein einfaches Programm zur Suche von Ressourcen anhand von Titel, Autor oder Fachgebiet. Es verwendet das Dublin Core Schema, Querys werden an alle vorhandenen Peers verschickt.
- **Edutella Swing Consumer:** Hier kann man Querys wie im Beispiel weiter vorn eingeben, entweder in einem Datalog-Format oder in XML. Die Suche wird solange fortgesetzt, bis man eine andere Suche startet. Die Ergebnisse werden in XML angezeigt.
- **Edutella SWEBOK Demonstrator:** SWEBOK steht für Software Engineering Body of Knowledge und enthält hauptsächlich Ressourcen zum Thema Software Engineering. Hier kann man die Suche mittels dreier verschiedener Kategorisierungen durchführen. Man kann verschiedene Themen des SWE wählen (Tools, Methoden, Design, Management usw.), oder aber nach der Art der Ressourcen einteilen (Vorlesung, Seminar, Beispiel, Video, Dokument usw.) und letztendlich eine Sprache wählen.

Außerdem unterstützt Edutella bereits eine ganze Reihe von Peers mit unterschiedlicher Funktionalität:

- *OLR Repository peer:* basierend auf IMS/LOM, kann Querys zwischen RDF-QEL-3 und SQL transformieren
- *DbXML peer:* basiert ebenfalls auf IMS/LOM und kann Querys zwischen RDF-QEL-1 und Xpath transformieren
- *Simple query and registration hub:* verteilt Querys anhand von Informationen über Schema und Fähigkeiten
- *Graphical query interface peer:* basiert auf Conzilla, kann RDF-Graphen in Querys übersetzen und die Ergebnisse dann grafisch darstellen
- *Annotation peer:* basiert auf Ont-O-Mat, kann Ergebnisse von Querys mit Metadaten versehen und wieder ins Repository zurückschreiben
- *File based repository peer:* basiert auf dem JENA Toolkit, verwendet die Sprache RDQL und speichert die RDF-Daten in normalen Dateien

5. Zusammenfassung

Das Wissen einzelner Personen, Personengruppen oder ganzer Abteilungen ist ein nicht mehr zu vernachlässigender Wirtschaftsfaktor. Viele große Unternehmen verwenden bereits Systeme zur Verwaltung von Inhalten und Wissen und viele weitere werden folgen. Dabei stoßen zentralisierte Architekturen aufgrund der Globalisierung und steigenden Komplexität der Informationen langsam an ihre Grenzen. Somit werden es die Peer-to-Peer-Systeme sein, die zukünftige Content- und Knowledge-Management-Systeme prägen und deren Infraatruktur bilden, denn sie erfüllen die wichtigsten der stetig wachsenden Anforderungen an solche Systeme. Die Nutzer werden mit immer neuen Informationen geradezu bombardiert, sie können sich mit beliebigen netzwerkfähigen Geräten in die P2P-Netze einklinken und Informationen nahezu überall und jederzeit abrufen. Die heutigen Browser werden sich zu multifunktionalen Arbeitsumgebungen entwickeln und dem Nutzer völlig neue Möglichkeiten, Kontrolle aber auch Verantwortung geben. Die Entwickler haben jedoch auch noch einige Aufgaben zu bewältigen, wie zum Beispiel die Einführung effektiver Sicherheitsmechanismen, besonders im Zusammenhang mit urheberrechtlich geschützen Daten oder geheimen Informationen.

Literaturverzeichnis

[BBM+02] D. Bertolini, P. Busetta, A. Molani, M. Nori and A. Perini, Peer-to-Peer, agents and knowledge management: a design framework, 2002

[BCM+02] Matteo Bonifacio, Roberta Cuel, Gianluca Mameli and Michele Nori, A Peer-to-Peer Architecture for Distributed Knowledge Management, 2002

[CXO03] CXO Media Inc., Knowledge Management, Januar 2003, http://www.cio.com/summaries/enterprise/knowledge/index.html

[GC01] Gartner Consulting, The Emergence of Distributed Content Management and Peer-to-Peer Content Networks, Januar 2001

[Ha01] Brandon Hall, New Technology Definitions, Oktober 2001, http://www.brandonhall.com/public/glossary/glossary.html

[HS03] David Hausheer, Burkhard Stiller, Design of a Distributed P2P-based Content Management Middleware, 2003

[NELH01] National Electronic Library for Health, Knowledge management glossary, 2001, http://www.nelh.nhs.uk/knowledge_management/glossary/glossary.asp

[NWQ+02] Wolfgang Nejdl, Boris Wolf, Changtao Qu, Stefan Decker, Michael Sintek, Ambjörn Naeve, Mikael Nilsson, Matthias Palmér and Tore Risch, EDUTELLA: A P2P Networking Infrastructure Based on RDF (Edutella Whitepaper), Mai 2002

[UT98] University of Texas, Knowledge Management FAQ, Februar 1998, http://www.mccombs.utexas.edu/kman/answers.htm