

Seminar

Informationsverwaltung in Netzen

Thema :
RDF – Beschreibungen und Peer – To – Peer Routing

Veranstalter : Prof. Heinz Schweppe

Betreuer : Valerie Bönström

Verfasser: Jan Kretzschmar

1. EINLEITUNG:

Ein wichtiges Kriterium für die Anwendungsmöglichkeiten von Peer – To – Peer – Systemen, ist die Art und Weise, in der es gelingt die erwarteten Eigenschaften, wie Autonomie der verschiedenen Teilnehmer, einer dezentralen Struktur, der Nutzung von verteilten Ressourcen und mit gewissen Einschränkungen Portabilität , den speziellen Erfordernissen der jeweiligen Anwendungen anzupassen.

In diesem Zusammenhang sind Besonders die Richtlinien zu nennen, die verteilten Ressourcen zielgenau und effektiv zu finden und Nachrichten zwischen den einzelnen Instanzen des Netzwerks effektiv, d.h. möglichst unter Vermeidung von Broadcasting und Redundanz weiterzuleiten. Bei der Planung eines solchen Systems ist also zu überlegen welche Strategien verwendet werden bzw. wie der Aufbau des Systems diesen Aspekten gerecht werden kann, z.B. durch eine spezielle Topologie.

Natürlich wird sich hierbei ein gewisser Konflikt zwischen einer optimalen Effizienz und einer entsprechenden Autonomie, Robustheit und Sicherheit nicht vermeiden lassen. Eine Lösung wird also immer ein Kompromiss mit einer Tendenz in die eine oder andere Richtung sein. Es gibt verschiedene Ansätze ein Peer – To – Peer –System (P2P – System) zu realisieren: Die Topologie kann völlig willkürlich sein wie z.B. bei Gnutella oder zentralisiert und eingeschränkt wie z.B. bei Napster, es gibt verschiedene Routing – Strategien vom „dummen“ Fluten bis zu raffinierteren Protokollen wie bei CAN oder Chord.

Andere Systeme fassen einzelne Peers zu gesonderten Einheiten zusammen und verwenden Super-Peers wie z.B. Kazaa, Grokster oder Morpheus. Es ist einzusehen, das die konkreten Anwendungen , aufgrund der meist speziellen Erfordernisse , zu eigenständigen Lösungen animieren. Diese stehen aber dem Anspruch gegenüber eine allgemeine Verfügbarkeit von Ressourcen zu ermöglichen sowie der Philosophie von P2P – Systemen, nach größtmöglicher Autonomie der Teilnehmer, welche inhomogene Systeme nicht ausschließt.

Wichtige Impulse für die Entwicklung von P2P – Systemen gehen von den Wissenssystemen aus. Die Idee wissenschaftliche oder allgemein Ausbildungs (engl.: „Education“) - Ressourcen im Sinne der P2P – Philosophie unabhängig und verteilt in einem Netzwerk zur Verfügung zu stellen ist nicht nur sehr reizbar, sondern erfordert in erster Linie die Möglichkeit diese meist sehr komplexen und uneinheitlichen, also nicht allgemein standardisierbaren Daten , effektiv und sicher zu adressieren und abzufragen. Um mit diesem Netzwerk wissenschaftlich arbeiten zu können werden natürlich auch andere Anforderungen an die Fähigkeiten des Systems „echtes“ und „vollständiges“ Information – Retrieval zu ermöglichen gestellt. Interessant ist hierbei die Nutzung von schon vorhandenen Internet-Standards, wie zum Beispiel das Ressource Description Framework (RDF) ,die verteilte, flexible und erweiterbare Ressourcenbeschreibungen unterstützen, was außerdem den angenehmen Nebeneffekt hat der Verwässerung von allgemeingültigen Standards durch konkurrierende Technologien entgegenzuwirken. Eine vorläufige Antwort und die neueste Entwicklung im Bereich der P2P – Systeme sind die sogenannten Schema- basierten P2P – Systeme. Eine besondere Instanz dieser, sind die schon erwähnten RDF – basierten P2P – Systeme, mit denen sich dieser Arbeit auseinandersetzt

Als Praktische Referenz ist abschließend die Edutella – Infrastruktur, ein Open - Source - Projekt (<http://edutella.jxta.org>) zu nennen, welches auf dem JXTA – Framework von Sun Microsystems basiert.

2. RDF als Beschreibung für W3-Ressourcen:

Das Resource Description Framework (RDF) welches vom World Wide Web Consortium (W3C) zur Modellierung von Metadaten über Ressourcen des Netzes vorgeschlagen wurde, wird benutzt um sogenannte „vocabularies“ oder Schemas zu definieren um damit Ressourcen zu beschreiben und bildet somit die Grundlage eines speziellen Typs von Schema – basierten P2P – Netzwerken den RDF – basierten P2P – Netzwerken.

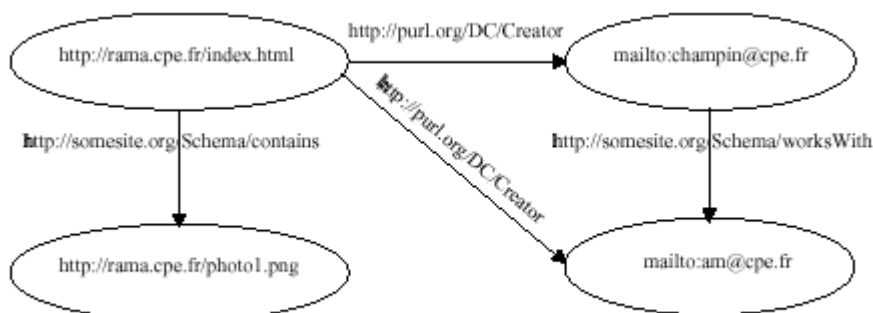
RDF identifiziert Ressourcen mit Hilfe von qualifizierten URI's (Uniform Resource Identifier), das sind URI's mit einem optionalen fragmentierten Identifier , von der URI durch ein „#“ getrennt. RDF betrachtet jede qualifizierte URI als eine eigene Ressource. Das Basis – Element des RDF – Modells (der Syntax) ist ein sogenannter Tripel: Eine Ressource (das Subjekt) ist mit einer anderen Ressource (dem Objekt) durch einen beschrifteten Pfeil , der dritten Ressource (dem Prädikat) verbunden. Man liest: das <Subjekt> besitzt ein Property <Prädikat> von der Ausprägung <Objekt>. Ein Tripel wird auch als Statement bezeichnet.

Ein Tripel:(Ein Statement) („Champin ist der Erzeuger von index.html“)



Alle Tripel ergeben einen gerichteten Graph, wie man an dem folgenden Beispiel erkennt, kann eine Ressource mehr als einen Wert für ein gegebenes Property besitzen.

Ein RDF – Graph



RDF benutzt die XML – Namespace – Notation für die eindeutige Auszeichnung von verteilten Ressourcen. Ein RDF – Dokument ist eine Liste von Beschreibungen. Jede Beschreibung wird gewöhnlich auf genau eine Ressource angewendet, und enthält eine Liste von Properties. Properties sind entweder URI´s , Literale oder andere Beschreibungen. Die fundamentalen Konzepte wären also rdf:Ressource, rdf:Property und rdf:Statement. In XML sind die RDF – Metadaten in ein Element namens rdf:RDF eingebettet. Dieses Element enthält eine Reihe von Elementen mit der Bezeichnung rdf:Description , welches eines der folgenden zwei Attribute rdf:about oder rdf:ID , aber nicht beide haben kann.

- rdf:about wird benutzt um eine beliebige Ressource zu beschreiben, dessen Wert entweder eine absolute bzw. eine relative URI ist.

Bsp.: `<rdf:Description about = „http://rama.cpe.fr/index.html“>`

`</rdf:Description>`

- rdf:ID wird für die Definition einer Ressource benutzt, sein Wert ist entweder ein fragmentierter Identifier
- (ohne „#“), um zur XML- Dokument URI hinzugefügt werden.

Bsp.: `<rdf:Description ID=“foo“>`

`</rdf:Description>`

Eine Beschreibung ohne rdf:about oder rdf:ID , beschreiben eine anonyme Ressource. Ein Element rdf:Descriptions besteht einer Sequenz von XML – Elementen. Diese Elemente werden als Propertie interpretiert, dessen Prädikat-URI´s zu einem Name des Elements erweitert wird. In Schemen können neue Ressourcen als Spezialisierung von alten definiert werden, welches implizite Tripel ermöglicht. Schemen stellen außerdem den Kontext sicher, in welchem die definierten Ressourcen benutzt werden können (Schema – Validität) Als erweiterte Schemenkonzepte von RDF können folgende angesehen werden:

rdfs:subproperty
 $(\forall s,p1,o,p2, \tau (s,p1,o) \wedge \tau (p1, rdfs: subPropertyOf, p2) \Rightarrow \tau (s,p2,o))$

Dieses Konzept ist folgendermaßen zu verstehen: Wenn ein Tripel ein Subproperty als Prädikat besitzt so muss er auch im Besitz des gesamten Property sein. Klassen sind Ressourcen, die eine Menge von Ressourcen(die Property rdf:type) beschreiben. Natürlich können diese hierarchisch angeordnet werden. Rdfs:Class, rdfs:subClassOf($\forall i,c1,c2, \tau (i,rdf:type,c1) \wedge \tau (c1, rdfs: subClassOf, c2) \Rightarrow \tau (i,rdf:type,c2)$)($\forall c1,c2,c3 \tau (c1,rdf:subClassOf,c2) \wedge \tau (c2, rdfs: subClassOf, c3) \Rightarrow \tau (c1,rdf:subClassOf,c3)$) Um die Eindeutigkeit von verteilten Ressourcen sicherzustellen gibt es noch zwei weitere Konzepte: Rdfs: domain und rdfs:range($\forall p,r1,r2, \tau (p,rdfs:range,r1) \wedge (r1 \neq r2) \Rightarrow \neg \tau (p,rdfs:range,r2)$)

RDF – QEL : RDF Query Exchange Language Nachdem die Daten mit Hilfe von RDF – Repositories ausreichend beschrieben worden sind, bleibt noch die Frage wie auf effektive Art und Weise diese (Meta)daten abgefragt werden können. Es gibt verschiedene Standards dieser Sprache (Wie diese zum Beispiel in Edutella eingesetzt werden):Die verschiedenen RDF – QEL –i Language Level:

RDF QEL1: für konjunktive Anfragen

RDF QEL2 : RDF QEL1 + disjunktive Anfragen

RDF QEL3 : RDF QEL2 + Negation (SQL92)

RDF QEL4 : RDF QEL3 + Transitives Schließen (SQL93)

RDF QEL5 : RDF QEL4 + Allgemeine Rekursion

Diese Sprachen basieren auf dem Datalog – Modell und sind deshalb mit den relational – vollständigen Abfragesprachen kompatibel.

Aufgrund dieser Eigenschaft ist es möglich die Repositories auch mit anderen Abfragesprachen wie z.B. simplen SQL mittels Wrapper abzufragen.

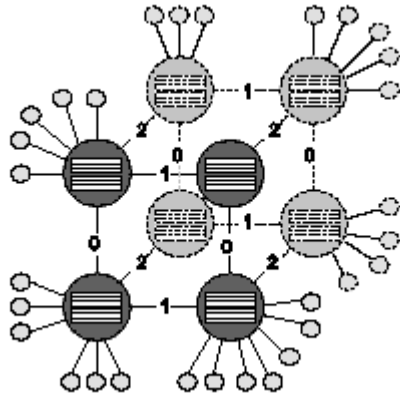
3. Allgemeine Prinzipien

In dieser Arbeit wird eine spezielle Instanz der Schema-basierten Netzwerke, den sogenannten RDF – basierten P2P – Netzwerken diskutiert werden. Aus diversen , später angeführten Gründen wird dafür eine Super-Peer-Topologie verwendet. Einer der Hauptgründe ist die bessere Skalierbarkeit im Verhältnis zu willkürlichen Strukturen, welches in vielen Fällen, den Einsatz von primitiven Broadcasting vermeiden hilft. Wie schon in der Einführung erwähnt wurde, erfordern Anwendungen wie verteilte und unabhängige Wissenssysteme unter Verwendung eines P2P – Netzwerks, die Möglichkeit mit den sehr unterschiedlichen Daten umzugehen. Diese müssen natürlich auch effektiv durch Metadaten beschrieben werden können. In RDF sind Ressourcen im Allgemeinen nichts anderes als eine URI, deshalb kann man mit Hilfe von RDF alle möglichen verschiedene Arten von Daten beschreiben. Die Entwicklung von Wissenssystemen ist schon ein wenig fortgeschritten, und es haben sich schon Standards für die Beschreibung solcher Ressourcen entwickelt, wie z.B. der aus 15 verschiedenen Auszeichnungen (Elementen) bestehende Dublin Core oder z.B. der Standard für Learning Object Metadata (LOM). Es ist eigentlich nur logisch, das diese bewährten Standards auch weiterhin für solche Systeme eingesetzt werden. Die Konsequenz ist natürlich das ein praktikables System mit solchen inhomogenen Teilnehmern, also den inhomogenen Schema-Beschreibungen umgehen muss. Es ist also notwendig zwischen verschiedenen Schemen zu vermitteln. Dafür gibt es feste Regeln, die sogenannten Zugehörigkeiten .Um ein „intelligenteres“ Routing zu ermöglichen, im Gegensatz zu den bisher verwendeten Systemen, sollen die ohnehin zur Verfügung stehende Informationen der verteilten Daten verwendet werden. Dies ist mit der Fähigkeit von RDF Metadaten – Schemen eindeutig zu identifizieren ohne weiteres möglich. Die Metadaten werden in sogenannte RDF – Repositories geschrieben. Diese können dann, wie z.B. Im Edutella – Netzwerk mit Hilfe von RDF – QEL abgefragt werden. RDF – QEL basiert auf der Datalog – Semantik und ist eine Erweiterung der gewöhnlichen relational- vollständigen Abfragesprachen. Zwei weitere interessante Ansätze sind diejenigen, die von Bernstein et. al : Locational Relational Model (LRM) und das von Aberer et. al. : dem Konzept der lokalen Translationsformeln um zwischen den einzelnen Schemen zu übersetzen. Weil sich die Metadaten höchstwahrscheinlich regelmäßig ändern, muß dies natürlich bei den verwendeten Routingstrategien Entsprechung finden. Für den Aufbau der Routing – Indizes, werden z.B. die sogenannten „dynamic frequency counting „ – Algorithmen benutzt. Dieses Konzept, ist der Art und Weise geschuldet, wie in einem Super – Peer – Netzwerk die einzelnen Indizes verwaltet werden, für die jeweiligen Schemen gibt es Referenzen zum benachbarten Super – Peer (die Anzahl ist abhängig von der Topologie), wenn eine Abfrage auf ein Schema abgesetzt wird, welches von keinem der benachbarten Super- Peers unterstützt wird, bleibt nichts anderes übrig als diese per Broadcasting weiterzuleiten. In einem völlig zufällig angeordneten Netzwerk könnte dies aber dazu führen, das die angedachte Ausnahme zum Regelfall wird. Deshalb ist man bemüht die Netzwerk – Topologie so zu gestalten das Peers mit „ähnlichen“ Eigenschaften in nahen Regionen angeordnet sind, die Bestimmung der Ähnlichkeit könnte z.B. nach dem wichtigsten Property erfolgen.

Nicht zu verschweigen ist allerdings, das dies ein praktisch nicht unproblematischer Ansatz ist, den die Bestimmung der Wichtigkeit einzelner Properties , bzw. was „Ähnlichkeit“ bedeutet ist nicht ganz einfach.

4. Super – Peer – Topologie:

Die einzelnen Peers werden zu einem Super-Peer-„Backbone“ verbunden.



Ein großer Vorteil der Super-Peer- Topologie im Zusammenhang mit Schema- basierten P2P- Netzwerken und Abfragen der entsprechenden Metadaten besteht darin, dass diese zentralisiert in den Super-Peers gehandhabt wird und damit nicht an jedem Peer Rechenleistung und Bandbreite im großen Maße , aufgrund der Abfragen und ihrer möglichen Antworten beansprucht wird. Jeder Peer ist mit seinem speziellen Super-Peer verbunden.

Ein weiterer Vorteil der Informationskonzentration ist die Möglichkeit effektiver zu routen.(Dies ergibt sich schon alleine aus der Möglichkeit Broadcasting weitgehend zu vermeiden). Aus Gründen, die später angeführt werden, ist das Netzwerk in einer HyperCuP – Topologie angeordnet. Eine solche erlaubt b^d Knoten in einem d-dimensionalen b-Hypercube. Wenn einige Knoten fehlen sollten, dann kann ein einzelner Knoten mehr als nur eine Stelle im Hypercube besetzen. Wenn neue Knoten dem Netzwerk beitreten, dann werden die fehlenden Lösscher ausgefüllt und gegebenenfalls die Dimension erweitert.

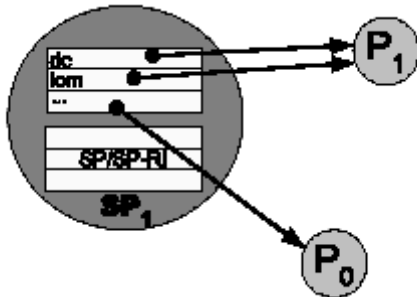
Ein neuer Super-Peer kann dem Netzwerk beitreten, wenn er einen anderen, bereits integrierten Knoten fragt und dann das Integrationsprotokoll ausführt.#

Die benachbarten Super-Peers und ihre relevante Schemainformation wird in die Indizes mit aufgenommen.Mit einem Aufwand von $O(\log N)$ kann ein neuer Knoten integriert und die Topologie beibehalten werden. Die Peers melden sich bei ihrem jeweiligen Super-Peer durch Abgabe einer Merkmals-Schablone (valid advertisement) an.

Bei Abmeldung der Peers/Super-Peers muss ein Update der relevanten Indizes und gegebenenfalls eine Bereinigung der Topologie erfolgen.

5. Super-Peer / Peer – Routing

Beispiel für einen Super-Peer//Peer Routing Index (SP/P-RI)



Die erste Art eines Index in diesem Netzwerk sind die sogenannten Super-Peer/Peer – Routing Indizes (SP/P-RI). In diesen Indizes schreibt jeder Super-Peer seine Metadaten-Informationen über jeden seiner Peers. Diese erhält er bei der Anmeldung, wo er von jedem Peer eine Merkmals-Schablone (advertisement) erhält. Um sicherzustellen, daß die Indizes immer auf dem aktuellen Stand sind, benachrichtigen die Peers ihren Super-Peer, sobald sich bei ihnen eine Veränderung vollzieht.

Jeder Peer teilt seinem jeweiligen Super-Peer sein bevorzugtes Schema mit, welches er für die Beantwortung von Abfragen benötigt.

Weil jeder Peer ein unabhängiges Schema benutzen kann z.B. Dublin Core oder LOM um seine Ressourcen zu beschreiben, sie aber dennoch semantisch gleichen Inhalts sein können, wird in unserem System zwischen diesen vermittelt.

An jedem Super-Peer wird eine etwaige Abfrage gegen die SP/P-RI´s gematched.

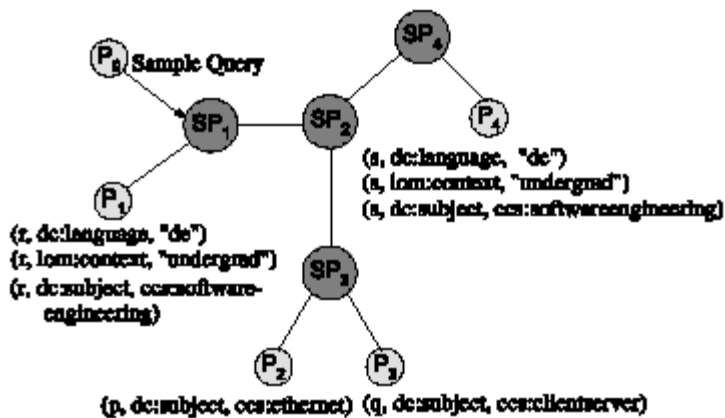
Damit wird herausgefunden, ob ein Peer in der Lage ist auf diese Abfrage zu antworten, was nicht heißt, das kein leeres Antwortset möglich ist.

Die Indizes können Informationen über die Peers in verschiedenen Granularitäten enthalten. (Schema- Index, Schema- Property, Property-Wert ..). Diese sind auch gesondert adressierbar.

Beispiel für die verschiedene Granularität einer Anfrage:

Granularity	Query	
Schema	dc, lom	
Property	dc:subject, dc:language, lom:context	
Property Value Range	dc:subject	ccs:sw engineering
Property Value	lom:context	"undergrad"
	dc:language	"de"

Eine Beispiel- Anfrage:



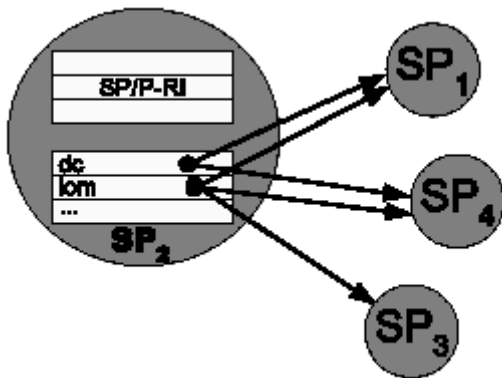
Es ist folgende Anfrage gegeben:

Finde jede Ressource, wobei das Property dc:subject = css:softwareengineering, dc:language = de und lom:context = „undergrad“.

Diese Abfrage bezieht sich auf die Abbildung darüber.

Peer P₀ sendet die Anfrage zu seinem Super-Peer. In diesem Beispiel kann die Abfrage von den Peers P₁ und P₄ beantwortet werden, welche zu SP₁ respektive zu SP₄ gehören. Diese enthalten die Metadaten-Ressourcen r und s, welche auf die Abfrage matchen.

Beispiel für einen Super-Peer / Super-Peer – Routing Index (SP/SP-RI)



Super-Peer/Super-Peer – Routing Indexe enthalten die selben Informationen wie alle Super-Peer/Peers – Indexe des jeweiligen Super-Peers zusammen und enthalten zusätzlich noch einen Zeiger auf die benachbarten Super-Peers.

6. Ein Beispiel für die Vermittlung von unterschiedlichen Schemen

Schema:

- 1.) lectures:identifier = dc:title
lectures:language = dc:lang
lectures:subject = dc:subject
 - 2.) lectures:identifier = lom:general.identifier
lectures:language = lom:general.language
lectures:context = lom:educational.context
-

Nun werden die Sichten der Peer-spezifischen Schemen erstellt:

1. lecturesViewDC(lectures:identifier, lectures:language, lectures:subject)
← DC(dc:title, dc:lang, dc:subject)
 2. lecturesViewLOM(lectures:identifier, lectures:language, lectures:context)
← LOM(lom:general.identifier, lom:general.language,
lom:educational.context)
-

Es folgt die Beschreibung, welche Attribute des Super-Peer lecture-Schemas durch lokale Peer-Schemen beantwortet werden können

- lectures(lectures:identifier, lectures:language, lectures:subject,-)
← lecturesViewDC(lectures:identifier,lectures:language,lectures:subject)
- lectures(lecture:identifier,lecture:language,-,lecture:context)
← lecturesViewLOM(lectures: identifier, lectures: language,
lectures:context)

Es ergeben sich folgende resultierende Zugehörigkeiten

Peer 1:Correspondence1 lectures(lectures:identifier,lectures:language,-
,lectures:educationalcontext)
← v(lectures:identifier,lectures:language,lectures:educationalcontext)
← LOM(lom:general.identifier,lom:general.language,
lom:educational.context)

Peer 2:Correspondence2
lectures(lectures:identifier,lectures:language,lectures:subject,-)
← v(lectures:identifier,lectures:language,lectures:subject)
← DC(dc:title, dc:subject, dc:lang)

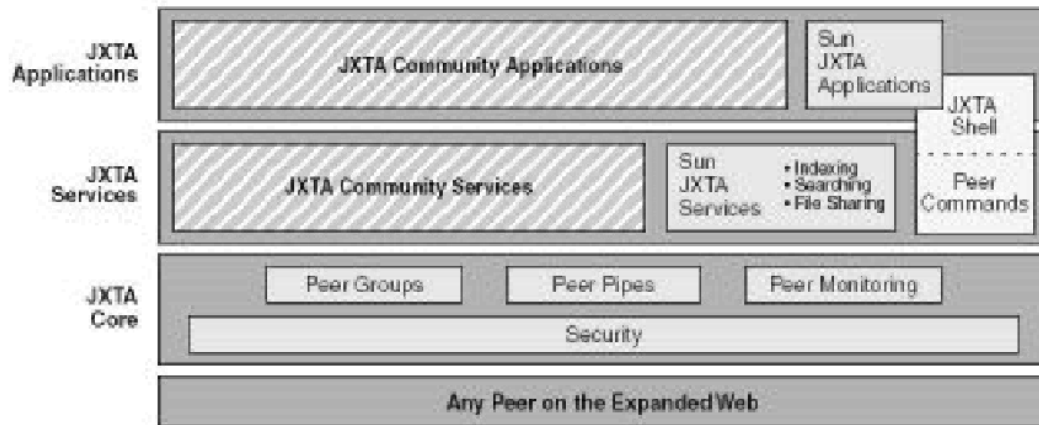
7.Dynamische Indizes

Eine wichtige Technik sind dynamische Indizes: Die Idee dabei ist die Indizes um zusätzliche (dynamische) Informationen. Diese „frequency information“ oder qualitative Informationen dienen zur Anpassung der Netzwerk-Topologie. (z.B. zur Vermeidung von Broadcasting)

Eine Ähnliche Idee ist das Clustern von Peers, d.h. Anordnung der Topologie anhand ähnlicher Eigenschaften.

8. Edutella

Die Layer des JXTA – Frameworks:



JXTA software is layered with core functionality developed through an open, collaborative effort, and higher-level services developed by P2P community developers.

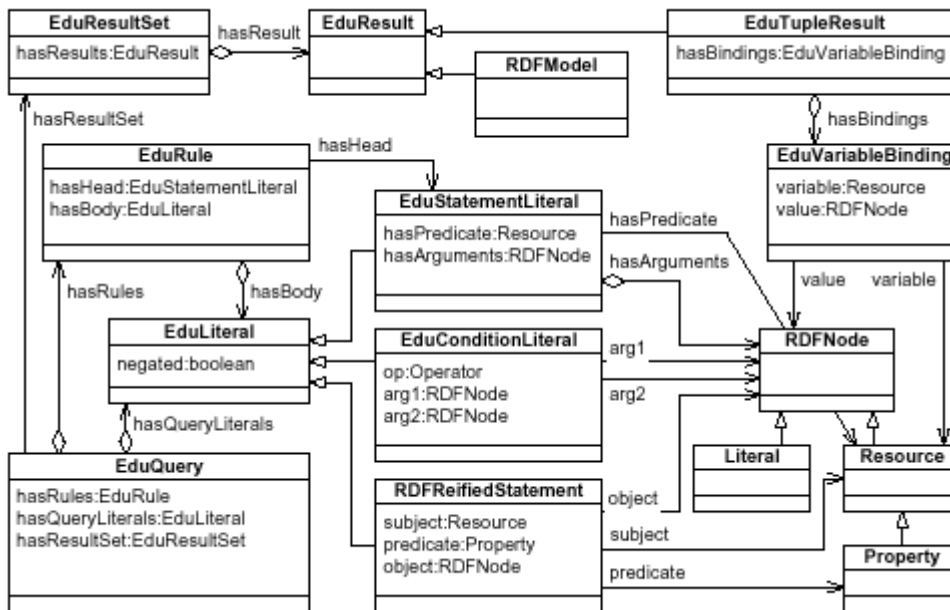
Das Edutella – Projekt basiert auf dem JXTA –Framework.

Die einzelnen Layer basieren auf XML – Protokollen.

Um seine Aufgabe als Funktionierendes P2P – System zu erfüllen muß es folgende Services Anbieten:

- Query – Service (Für standardisierte Abfragen und retrieval der RDF-Metadaten)
- Replication – Service (Für eine gute Verfügbarkeit und load-balancing der Daten)
- Mapping – Service (Vermittelt zwischen verschiedenen Schemen und löst Konflikte)
- Annotation – Service (Markiert die Gespeicherten Daten im Netzwerk, für spätere Suche)

Das Edutella Common Data and Query Exchange Model (ECDM)



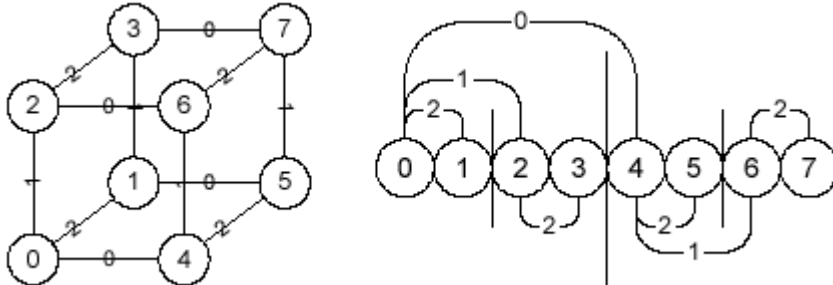
Das ECDM benutzt ein Datalog – Modell für die interne Repräsentation von Abfragen und ihren jeweiligen Antworten. Wrapper können zwischen den einzelnen RDF – basierten Abfragesprachen übersetzen.

9. Effiziente P2P – Topologien

Welche Voraussetzungen sind allgemein für Skalierbare P2P – Topologien notwendig?
In erster Linie die Symmetrie, d.h. jeder Peer sollte ein Wurzelknoten in einem Spannbaum des gesamten Netzwerkes sein. Desweiteren sind hierbei ein niedriger Netzwerkdurchmesser und eine begrenzte Anzahl von Nachbarn zu nennen. (Beides sollte bei $O(\log n)$ liegen. Die Topologie sollte last- balanciert sein und effiziente Broadcast – Algorithmen ermöglichen. Natürlich muß die dynamische Hinzunahme und Entfernung von Peers berücksichtigt werden. Alle diese Voraussetzungen wird durch eine Gruppe von Graphen den sogenannten Cayley – Graphen erfüllt, diese sind regulär, vertex-symmetrisch und rekursiv dekomponierbar. Die bekannteste Instanz der Cayley – Graphen ist der sogenannte Hypercube – Graph. Dieser findet Anwendung in der HyperCuP – Topologie, welche auch für Edutella Verwendung gefunden hat.

10. HyperCuP Peer – To – Peer – Topologie

Ein Hypergraph mit seiner serialisierten Form



Es geht in erster Linie um die Schaffung einer deterministischen Netzwerkstruktur, welche in der Lage ist eine effektive Suche und entsprechendes Broadcasting auszuführen. Mit den Kenntnissen der Eigenschaften bietet sich dafür idealerweise eine HyperCup – Topologie an. Dieses P2P – Netzwerk muss symmetrisch sein, jeder Knoten muss dieselben „Möglichkeiten“ und „Pflichten“ innerhalb des Netzwerks ausüben können. Dieses schließt die Existenz eines zentralen Servers aus. Der Netzwerkdurchmesser ist definiert als der kürzeste Weg zwischen den entferntesten Knoten des Netzwerks, weil aber jeder Knoten als Wurzel eines Spannbaums über das gesamte Netz angesehen werden kann, beträgt dieser, im schlimmsten Fall $O(\log n)$. Ein kompletter Hypergraph besteht aus $N = b^{(L_{max}+1)}$ Knoten, wobei jeder Knoten $(b-1) \cdot (L_{max}+1)$ Nachbarn, $(b-1)$ in jeder Dimension hat und $L_{max}+1$ die Anzahl der Dimensionen ist. Der Netzwerkdurchmesser ist dann genau $\log_b N$.

10. Zusammenfassung / Fazit

In dieser Arbeit sind einige interessante Ansätze erläutert worden, um die bisherigen Ansätze, in puncto Möglichkeit zur Komplexen Metadaten – Abfrage, verteilte Ressourcen, inhomogene Teilnehmer, effektives Routing zu verbessern.

Wenn es gelingt dieses im vollen Umfang in die Praxis umzusetzen, sind neue Möglichkeiten für die Anwendung von P2P – Systemen, besonders im Bereich der Wissenssysteme gegeben.

Im Detail sind die hier vorgestellten Ansätze jedoch nicht unproblematisch.

Der Anspruch von völlig autonomen, inhomogenen Teilnehmern, ist momentan nicht zu realisieren, weil nicht geklärt ist, wie eine vermittelnde Instanz

In einem solchen Fall aussehen kann. Ein weiterer Grund der entgegen der Philosophie der P2P – Systeme geht, ist das die Super – Peers als Kontrollinstanz fungieren müssen.

Die verwendete Super – Peer – Topologie besitzt neben vielen Vorteilen, natürlich den Nachteil, das es unter Umständen nicht zu vermeiden ist, Nachrichten nur mit Hilfe von Broadcasting weiterzuleiten. Auch wenn es aufgrund der viel geringeren Anzahl von Super – Peers im Verhältnis zu Peers und besonders bei einer HyperCuP – Topologie sich nicht so gravierend auswirkt wie bei anderen Systemen, ist das bestimmt nicht wünschenswert. Der Ansatz einer dynamischen Topologie aufgrund von „frequency – counting“ ist, mal abgesehen davon das dies natürlich auch zusätzlichen Aufwand zu Lasten der Prozessleistung und der Bandbreite geht, ebenfalls zu diskutieren.

11. Inhaltsverzeichnis

Referenzen:

- [1] Wolfgang Nejdl, Martin Wolpers, Wolf Siberski, Christoph Schmitz, Mario Schlosser, Ingo Brunkhorst, Alexander Löser: „Super-Peer-Based Routing and Clustering Strategies for RDF-Based Peer-To-Peer Networks“
- [2] Wolfgang Nejdl, Boris Wolf, Changtao Qu, Stefan Decker, Michael Sintek, Ambjörn Naeve, Mikael Nilsson, Matthias Palmer, Tore Risch: „EDUTELLA: A P2P Networking Infrastructure Based on RDF“
- [3] Pierre-Antoine Champin: „RDF Tutorial“
- [4] L.A. Adamic, R.M. Lukose, A.R. Puniyami, and B. A. Huberman: „Search in Power-law Networks. In Physical Review E, 64 46135, 2001“
- [5] P.A. Bernstein, F. Giunchiglia, A. Kementsietsidis, J. Mylopoulos, L. Serafini, and I. Zaihrayeu: „Data management for peer-to-peer computing: A vision. In Proceedings of the Fifth International Workshop on the Web and Databases, Madison, Wisconsin, June 2002.“
- [6] S. Busse: „Model Correspondences in Continuous Engineering of MBIS –doctorial thesis. Logos Verlag, September 2002 „
- [7] S. Chawathe, H. Garcia-Molina, J. Hammer, K. Ireland, Y. Papakonstantinou, J. Ullman, and J. Widom: „The TSIMMIS projekt: Integration of heterogeneous information sources. In Proceedings of IPJS Conference, Tokyo, Japan, October 1994“
- [8] A. Crespo and H. Garcia-Molina. Routing indices for peer-to-peer systems. In Proceedings International Conference on Distributed Computing Systems, July 2002“
- [9] M. Harren, J. M. Hellerstein, R. Huebsch, B. T. Loo, S. Shenker, and I. Stoica: „Complex queries in DHT-based peer-to-peer networks. In F. Kaashoek and A. Rowstron, editors, *Proceedings for the 1st International Workshop on Peer-to-Peer Systems (IPTPS '02)*, March 2002.“
- [10] R. Korfhage: „*Information Storage and Retrieval*. John Wiley, New York, 1997.“

- [11] U. Leser.:“ *Query Planning in Mediator Based Information Systems - doctoral thesis*. TU Berlin, June 2000.“
- [12] G. S. Manku and R. Motwani.: „Approximate frequency counts over data streams. In *Proceedings of the 28th International Conference on Very Large Data Bases*, Hong Kong, China, August 2002.“
- [13] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker.:“ A scalable content addressable network. In *Proceedings of the 2001 Conference on applications, technologies, architectures, and protocols for computer communications*. ACM Press New York, NY, USA, 2001.“
- [14] S. Saroiu, P. K. Gummadi, and S. D. Gribble.:“ A measurement study of peer-to-peer file sharingsystems. In *Proceedings of Multimedia Computing and Networking (MMCN)*, January 2002.“
- [15] M. Schlosser, M. Sintek, S. Decker, and W. Nejdl. :“HyperCuP—Hypercubes, Ontologies and Efficient Search on P2P Networks. In *International Workshop on Agents and Peer-to-Peer Computing*, Bologna, Italy, July 2002.“
- [16] „Semantic overlay networks, November 2002. Submitted for publication.“
- [17] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan. Chord:“ A scalable peer-to-peer lookup service for internet applications. In *Proceedings of the 2001 Conference on applications, technologies, architectures, and protocols for computer communications*. ACM Press New York, NY, USA, 2001.“
- [18] G. Wiederhold.:“ Mediators in the architecture of future information systems. *IEEE Computer*, 25(3):38 – 49, 1992.“
- [19] B. Yang and H. Garcia-Molina.:“ Designing a super-peer network. <http://dbpubs.stanford.edu:8090/pub/2002-13>, 2002.“