Reputationsbasierte Trust-Metriken im Kontext des Semantic Web

Diplomarbeit
zur Erlangung des akademischen Grades
Diplom-Informatiker

vorgelegt von

Oliver Michael Maresch
(194970)

Gutachter:

Prof. Dr. Bernd Mahr (Technische Universität Berlin)
Dr. Olaf Kath (Technische Universität Berlin)

Betreuter:

Dipl.-Kfm. Christian Bizer (Freie Universität Berlin)
Dipl.-Ing. Michael C. Jaeger (Technische Universität Berlin)

Formale Modelle, Logik und Programmierung
Institut für Telekommunikationssysteme
Fakultät IV (Elektrotechnik und Informatik)
Technische Universität Berlin

© Copyright by

Oliver Michael Maresch

August 2005

Erklärung

Die selbständige und eigenhändige Anfertigung

versichere ich an Eides statt.

Berlin, den 18. August 2005

gez. Oliver Michael Maresch

Danksagung

Ich möchte allen danken, die zum Gelingen dieser Diplomarbeit beigetragen haben. Mein besonderer Dank gebührt meiner Tochter Isabella, die am 5. Mai 2005 das Licht der Welt erblickte, sowie meiner Frau dafür, dass sie die letzten Monaten durchgehalten haben. Chris Bizer danke ich für seine intensive Betreuung und die Einbindung meiner Diplomarbeit in das Projekt der TriQL-Trust-Architektur. Mein Dank geht an Prof. Dr. Bernd Mahr und Michael Jeager, durch die diese Diplomarbeit erst möglich wurde. Zuletzt möchte ich meinen Eltern für ihre immerwährende Unterstützung und das intensive Korrekturlesen meiner Diplomarbeit danken.

Inhaltsverzeichnis

1 Einführung....1
1.1 Konventionen....3
1.2 Semantic Web....4
1.3 Die TriQL-Trust-Architektur....6
1.4 Vertrauen (Trust)....8
1.4.1 Intuitives Beispiel....9
1.4.2 Der Vertrauensbegriff in der Literatur....10
1.4.3 Definition von Vertrauen....12
1.4.4 Definition von Reputation....13
1.5 Problemstellung....14
1.6 Einordnung und Abgrenzung....15
1.7 Aufbau der Diplomarbeit....16

2 Klassifikation von Trust-Metriken in reputationsbasierenden Systemen....17
2.1 Klassifikationen....18
2.1.1 Datenbasis....18
2.1.1.1 Sichtbarkeit der Datenbasis....19
2.1.1.2 Lokalität der Datenbasis....19
2.1.2 Bewertungsschema....20
2.1.2.1 Form von Bewertungen....20
2.1.2.2 Grapheigenschaften....22
2.1.2.3 Identität des Bewerters....23
2.1.3 Bewertungsprozess....24
2.1.3.1 Kontrolle des Bewertungsprozesses....24
2.1.4 Bewertungsalgorithmus....25
2.1.4.1 Perspektive....25
2.1.4.2 Auswahlverfahren....26
2.1.4.3 Evaluierung der Beziehung von Entität....27
2.1.5 Allgemeine Eigenschaften....28
2.1.5.1 Ort der Evaluierung....28
2.1.5.2 Interpretation der Inferierten Ergebnisse....29
2.2 Weitere Eigenschaften von reputationsbasierenden Systemen....30
2.2.1 Bootstrap-Probleme....30
2.2.2 Privatsphäre....31
2.2.3 Skalierbarkeit....32
2.2.4 Qualitätssicherung der Bewertungsinformationen....33
2.2.5 Resistenz gegen Manipulation....33
2.3 Evaluierung der Klassen....34

3 Reputationsbasierende Trust-Metriken....40
3.1 Appelseed....40

3.2 EigenRumor....43
3.3 EigenTrust....47
3.4 Evidence-Based Model....51
3.5 PageRank....56
3.6 PeerTrust....61
3.7 TidalTrust....65
3.8 Weitere reputationsbasierende Trust-Metriken....69

4 Implementierung von Trust-Metriken für die TriQL-Trust-Architektur....71
4.1 Basistechnologie....71
4.1.1 Resource Description Framework (RDF)....71
4.1.2 Named Graphs (NG)....72
4.1.3 SWP – Vokabular für Web-Veröffentlichungen....72
4.1.4 TriQL - Named Graphs Anfragesprache....73
4.2 TriQL-Trust-Architektur....73
4.2.1 Der Query and Trust Evaluation Layer....73
4.2.2 PiggyBank und der TriQL.P Trust Policies Enabled Semantic Web Browser....76
4.3 Trust-Metriken für die TriQL-Trust-Architektur....78
4.3.1 Einbindung in TriQL – Das Metrikinterface....78
4.3.2 Einbindung in den Query and Evaluation Layer....79
4.3.3 Appleseed-Implementierung....82
4.3.4 Ebay-Implementierung....85
4.3.5 Semantic PageRank-Implementierung....85
4.3.6 TidalTrust-Implementierung....87
4.4 Erklärung der Metrikergebnisse....88
4.4.1 Grundlagen....88
4.4.2 Erklärungen für die TriQL-Trust-Architektur....91
4.4.2.1 Informationsquellen von Metrikinferenzen....92
4.4.2.2 Erklärung des Bewertungsprozesses....93
4.5 Anwendungsbeispiel Finanzdatenbeispiel mit TidalTrust....95
4.6 Anwendungsbeispiel FOAF:knows mit Semantic PageRank....100

5 Zusammenfassung....104
5.1 Bewertung....104
5.2 Ausblick....105

Anhang A: Abkürzungsverzeichnis....106

Anhang B: Literaturverzeichnis....107

Abbildungsverzeichnis

Abbildung 1: TriQL-Trust-Architektur....8
Abbildung 2: Merkmale reputationsbasierender Systeme....17
Abbildung 3: Matrix für die Eigenvektorberechnung in EigenRumor [ERRB05]....44
Abbildung 4: Normalisierung der Bewertungen in EigenTrust [EARM03]....48
Abbildung 5: Basis des EigenTrust-Algorithmus[EARM03]....49
Abbildung 6: Formel des mathematischen Modells der PeerTrust-Metrik [RBTM03]....62
Abbildung 7: Struktur des Semantic Web Publishing RDF-Vokabulars [NGPT05]....73
Abbildung 8: Generierung einer TriQL.P Anfrage [TPWB]....75
Abbildung 9: Ausführung der Vertrauensanalyse [TPWB]....76
Abbildung 10: Überblick TriQL.P-Browser....77
Abbildung 11: Beispiel Anfrage mit TriQL.P unter Verwendung der Metrikschnittstelle....79
Abbildung 12: Programmierschnittstelle für rangbasierende Metriken 81
Abbildung 13: Programmierschnittstelle für schwellenwertbasierende Metriken....82
Abbildung 14: Beispielanwendung der TriQL.P-Metrikschnittstelle für die Appleseed....84
Abbildung 15: Beispielanwendung der TriQL.P-Metrikschnittstelle für die Ebay-Metrik....85
Abbildung 16: Beispielanwendung der TriQL.P-Metrikschnittstelle für SemanticPageRank .86
Abbildung 17: Beispielanwendung der TriQL.P-Metrikschnittstelle für die TidalTrust....88
Abbildung 18: Quellenangabe einer Metrikinferenz....92
Abbildung 19: Entscheidungsstruktur eines Metrikprozesses....94
Abbildung 20: Trust-Policy: TidalTrust Value above 0.5....96
Abbildung 21: Ergebnis einer Trust-Policy unter Verwendung von TidalTrust....97
Abbildung 22: Erklärung eines Ergebnisses unter Verwendung von TidalTrust....99
Abbildung 23: Trust-Policy: Most popular FOAF Person....101
Abbildung 24: Ergebnis einer Trust-Policy unter Verwendung von Semantic PageRank....102
Abbildung 25: Erklärung eines Ergebnisses unter Verwendung von Semantic PageRank....103

Tabellenverzeichnis

Tabelle 1: Eigenschaften von reputationsbasierenden Systemen 30
Tabelle 2: Zusammenfassung der Metrikeigenschaften 70

1 Einführung

Die Frage nach dem Vertrauen stellt sich Individuen, seit sie sich durch die Kooperation mit Dritten Vorteile erhoffen. Dabei müssen sie vorher in Kooperationen investieren und das Risiko eingehen, ihre Investition zu verlieren. Dennoch ist Vertrauen eines der menschlichen Grundbedürfnisse, ohne das Zusammenleben nicht möglich wäre. Familien, Freunde und Geschäftspartner werden durch Vertrauen zusammengehalten. Bricht das Vertrauen, so führen uns die Wege meist auseinander.

In der modernen Informationstechnologie suchen Nutzer oder Softwareagenten nach Informationen. Die Daten, die sie dabei aus integrierten Informationssystemen wie dem World Wide Web (WWW) und dem Semantic Web erhalten, zeichnen sich durch extremen Umfang, sehr verschieden strukturierter Datenquellen und großen Unterschieden in der Informationsqualität aus [QDQA02]. Die zunehmend hohe Abhängigkeit von qualitativ hochwertigen Informationen für die Entscheidungsfindung stellt für die Nutzung integrierter Informationssysteme ein immer größeres Problem dar. Geringe Informationsqualität führt fortwährend zu mehr und mehr sozialen wie kommerziellen Konsequenzen und somit zu Unsicherheit.

Die Verwendung von Methoden zur Auswahl qualitativer hochwertiger Informationen kann hier ein Mittel sein, um die Komplexität der Auswahlverfahren zu reduzieren und somit effiziente Entscheidungen über die Qualität von Informationen zu treffen. Diese Methoden sollen Informationskonsumern helfen, trotz der Unsicherheit aus der breiten Informationsbasis, begründete Entscheidungen für die Nutzung von Informationsquellen zu treffen, und somit darauf zu vertrauen, das der Informationsprovider Informationen mit hoher Qualität bereitgestellt hat.

Die Umsetzung des sozialen Konzepts Vertrauen in die Informationstechnologie gestaltet sich als eine herausfordernde Aufgabe für die Wissenschaft. Der neuen virtuellen Welt fehlen einige Vertrauensmechanismen, die sich aus der realen Welt nicht übertragen lassen. Lik Mui schreibt in seiner Ph.D. These

[CMTR04], dass subtile Vertrauensmerkmale, die aus der menschlichen Kommunikation resultieren wie der Tonfall, die Körpersprache, das Handreichen oder das allgemeine Auftreten einer Person, keine Realisierung in der Informationstechnologie kennen. Ebenfalls wie Paul Resnik u.a. [SCCP01] argumentiert Mui, dass der einfache Identitätswechsel sowie das problemlose Ein- und Austreten aus Online-Communities im Gegensatz zur realen Welt steht. Zudem werden Kooperationen oft zwischen Fremden eingegangen, so dass sich keine langjährigen Vertrauensbeziehungen entwickeln können. Aus diesen Tatsachen resultiert ein geringes Vertrauen in potentielle Kooperationspartner und geringe Erwartungen in zukünftige Transaktionen bei der Verwendung von moderner Informationstechnologie.

Bisherige Ansätze lösen nur Teile des Gesamtproblems. Public Key Infrastrukturen (PKI) leisten einen entscheidenden Beitrag für die Identitätsüberprüfung von Individuen und die unverfälschte Übertragung von Informationen. PKIs zielen vor allem auf Sicherheit in der Datenübertragung ab und lassen Vertrauen als soziales Konzept ausser Acht [TCTS04]. Die Überprüfung der Vertrauenswürdigkeit eines Individuums und der Qualität von Informationen können von PKIs nicht erbracht werden.

Reputationsbasierende Systeme können hier eine Brücke schlagen, um mit Hilfe der Aufzeichnung, Verbreitung und Speicherung von Transaktionsdaten eine Historie des Verhaltens von Agenten zu führen [CMTR04]. Die aus den Transaktionsfeedbacks resultierende Reputation sowie persönliche Meinungen über ein Individuum können verwendet werden, um sein zukünftiges Verhalten vorhersagen zu können und somit über seine Vertrauenswürdigkeit zu entscheiden [RS00]. Agenten, die eine gute Reputation hinsichtlich der Verbreitung von Informationen mit einer hohen Qualität haben, wird sicherlich zugetraut hochwertige Informationen zu liefern.

Bei der Verbreitung von Bewertungen und Empfehlungen für den Aufbau von Reputation können die Vorteile moderner Informationstechnologie genutzt werden, die die Verteilung von Informationen schnell und strukturiert vornimmt.

Darüberhinaus wird das soziale Konzept der Reputation nicht nur für menschliche Nutzer übertragen, sondern es wird ebenso für Softwareagenten zugänglich.

Aus dieser Motivation heraus entstanden in den letzten Jahren verschiedenste reputationsbasierende Systeme. Zudem bietet das sich allmählich entwickelnde Semantic Web neue Möglichkeiten, um reputationsbasierende Informationen zu formulieren und zu verbreiten. Die von Christian Bizer und Radoslaw Oldakowski vorgestellte TriQL-Trust-Architektur [CCBT04] ist ein Beispiel für Anwendungen des Semantic Web, die Reputation für die Berechnung von Vertrauen einsetzen möchte. Um bereits vorhandene reputationsbasierende Metriken einsetzen zu können, stellt sich prinzipiell die Frage nach deren Eigenschaften und Eignung für neue Systeme.

Bevor in dieser Diplomarbeit auf dieses Thema im Detail eingegangen wird, sollen in den Abschnitten 1.1 bis 1.4 einige Begriffe erläutert und wichtige Hintergrundinformationen vermittelt werden. Abschnitt 1.5 und 1.6 wird kurz den Aufgabe-Lösungs-Ansatz der Arbeit wiedergeben und die Arbeit in den wissenschaftlichen Kontext einordnen, bevor Abschnitt 1.7 den weiteren Aufbau der Arbeit vorstellt.

1.1 Konventionen

In dieser Diplomarbeit werden einige Begriffe immer wieder verwendet. Um die Lesbarkeit zu erhöhen werden einige Begriffe unter einem Schlagwort zusammengefasst.

In dieser Diplomarbeit wird in letzter Konsequenz die Anwendbarkeit von einzelnen reputationsbasierenden Trust-Metriken für die TriQL-Trust-Architektur entschieden, die klar auf menschliche Nutzer ausgerichtet ist. Die vorhergehende Analyse wird allerdings nicht auf menschliche Nutzung eingeschränkt, sondern bezieht sich gleichfals auf Softwareagenten, die autonom auf Kooperationen eingehen können. Um nicht alle möglichen Anwender wiederholt nennen zu müssen, wird im folgenden nur noch das Wort *Entität* verwendet.

Die Entitäten können mit Trust-Metriken Vertrauen evaluieren. Als *Ziel des Vertrauens* wird dabei diejenige Entität bezeichnet, deren Vertrauenswürdigkeit

bewertet werden soll. Die *Quelle des Vertrauens* ist jene Entität, die eine Evaluierung der Vertrauenswürdigkeit anstößt. Eine Trust-Metrik berechnet also immer das Vertrauensverhältnis zwischen Quellen des Vertrauens und ihrer Ziele.

1.2 Semantic Web

> *„The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation." (Tim Berners-Lee, James Hendler, Ora Lassila, The Semantic Web, Scientific American, May 2001)*

Die Informationen im bisherigen WWW wurden für menschliche Benutzer geschaffen. Webseiten konzentrieren sich auf die Darstellung von Informationen. Der Ansatz des Semantic Web geht über dieses Ziel hinaus. Informationen im Web sollen von Maschinen wie Menschen manipulierbar sein, damit Softwareagenten untereinander autonom agieren und dem Menschen die Suche und Bearbeitung von Informationen erleichtern können. Um die Interoperabilität von Datenbeständen zum Austausch und Verarbeiten von Informationen sicherstellen zu können, steht die Integration von Informationen im Vordergrund.

Um die Voraussetzungen für die Entwicklung eines Semantic Web zu schaffen, initiierte das World Wide Web Consortium (W3C) [W3C] die Semantic Web Activity, um mit der Entwicklung von offenen Standards und Tools den Entwicklungsprozess voranzutreiben. Für die Beschreibung von Daten wurde das Resource Description Framework (RDF) [RDF] als Standard geschaffen. Mit „Semantic Web“ wird allgemein hin das integrierte Informationsystem referenziert, welches auf den Strukturen des WWW aufbaut und maschineninterpretierbare Informationen nach dem RDF-Standard enthält.

Die eigentliche Idee des Semantic Web ist aber nicht auf RDF beschränkt. Um die Ziele des Semantic Web zu verdeutlichen sollen kurz zwei reale Anwendung des heutigen Semantic Web vorgestellt werden. Sie sollen veranschaulichen, wie die Integration von Informationen im Semantic Web funktioniert und wie durch die Verknüpfung von Informationen durch Programme Mehrwerte entsteht.

Als erstes Beispiel soll das „Flink – The Who is Who of the Semantic Web“

Projekt [FLINK] vorgestellt werden. Auf der Projekt-Homepage ist eine Anwendung gehostet, die Informationen von mehreren internationalen Semantic Web Veranstaltungen teilnehmerbezogene Informationen zusammenträgt. Informationen aus verschiedenen Quellen über verschiedenste Protokolle und Formate hinweg werden in einen Datenbestand automatisch über verschiedene Data Mining Methoden integriert. So werden in FLINK sowohl Informationen aus bereits in RDF formatierte FOAF-Dateien, wie aus Emails über IMAP oder POP3 Protokolle, aus Google-Suchanfrage und aus Literaturdatenbanken genutzt. Jede Information wird zudem mit weiteren Metainformationen versehen, die die Fundstellen und die Zeit des Fundes der Informationen festhalten.

Neben der schlichten Integration und Präsentation der Informationen der Teilnehmer im Who-is-Who werden die Teilnehmer in einem Netzwerk in Beziehung zueinander gesetzt. Zu diesem Zweck werden unter anderem Google-Anfragen benutzt, bei denen die Verbindung zweier Teilnehmer aus der Anzahl der Webseiten bestimmt wird, auf denen beide zusammen namentlich erwähnt werden. Die Ergebnisse dieser Netzwerkanalyse lassen sich mit einem Graph-browser durchsuchen. Eine sehr schöne Anwendung, ist die Positionierung der Namen der Teilnehmer auf Satellitenbildern, anhand von Positionsdaten, die in Zusammenhang mit den Teilnehmern gefunden oder über Adressen bestimmt wurden.

Mit der Anwendung von Satellitenbildern kommen wir zum zweiten Beispiel, Google Earth [GEAR]. Auf den ersten Blick erscheint Google Earth als eine Anwendung die Karten und Satellitenbilder miteinander verknüpft. Das besondere an Google Earth sind allerdings nicht nur die verblüffend detailreichen Bilder, sondern auch die Möglichkeiten beliebige ortsbezogene Informationen in die Karten zu integrieren. So kann jeder Benutzer bestimmten Orten einen Namen geben, ihm weitere Informationen zuordnen und sie mit anderen Benutzern durch den Austausch von Metainformationen teilen. Für die USA sind diese Informationen bereits so detailliert, dass man mit dem Stadtnamen, dem Straßennamen und der Hausnummer praktisch in jedermanns Vorgarten zoomen kann.

Tankstellen, Restaurants, Hotels oder Geschäfte können in die Karte eingetragen werden und auf ihre Angebote verweisen.

Das die integrierten Informationen einen Mehrwert schaffen zeigt die Diskussion um sicherheitsrelevante Bereiche, die von Google Earth ebenfalls abgebildet werden. So lässt sich ohne große Probleme das Weisse Haus in Washington betrachten. Am 8. August 2005 meldet zum Beispiel die Zeitung „The Australian“, dass der Chef der australischen Atomaufsicht befürchtet, das Informationen über das einzige australische Atomkraftwerk „Lucas Heights“ Anschläge auf das Atomkraftwerk begünstigen würde. Er forderte die Firma Google auf, die entsprechenden Informationen und Bilder aus der Öffentlichkeit zu entfernen.

Die Anwendung Goolge Earth unterstützt zwar die Standards der Semantic Web Activity nicht, dennoch wird mit Google Earth die Idee der Informationsintegration in ein weltweites Netzwerk vorangetrieben. Google Earth zeigt zudem das kommerzielle Potential, welches hinter dieser Technologie steckt.

1.3 Die TriQL-Trust-Architektur

Das Semantic Web hat per Definition eine offene und verteilte Informationstruktur [SW01]. Ähnlich wie im WWW können Informationen von jedermann hinzugefügt werden. Dabei variiert die Qualität der Informationen in ihrer Korrektheit, Aktualität und der subjektiven Wahrnehmung sehr stark. Vor diesem Hintergrund sollten Aussagen eher als Behauptungen betrachtet werden, die erst durch eine Evaluierung ihrer Glaubwürdigkeit mit positivem Ergebnis zur Tatsache werden [CCBT04]. Es gilt also die Frage zu klären: „Vertraue ich darauf, dass eine Information eine hohe Qualität hinsichtlich meiner Qualitätskriterien hat?“

Die von Christian Bizer und Radoslaw Oldakowski vorgestellte Architektur [CCBT04] ist eine prototypische Implementierung für die Evaluierung der Informationsqualität von Informationen aus dem Semantic Web. Durch die Evaluierung soll einem Benutzer bei der Entscheidung, eine Information zu nutzen, geholfen werden. Sie verwendet einen ganzheitlichen Ansatz von der Daten-

sammlung und Verifikation der Herkunft über die Auswertung bis hin zur Erklärung der Ergebnisse für Endanwender. Das erklärte Ziel ist es, einem Endanwender die Möglichkeit zu geben auf seinem Rechner die Informationen, die für ihn persönlich von hoher Qualität sind, aus dem Semantic Web heraus zu filtern.

Die Bewertung wird auf drei Mechanismen zurückgeführt:

- Kontentbasierende Mechanismen nehmen die Information selbst zur Bewertungsgrundlage („Wurde die Information bereits anderswo veröffentlicht?“),
- Kontextbasierende Mechanismen verwenden den Kontext in dem eine Aussage geäußert wurde als Informationsgrundlage („Glaube nur Informationen, die vom Hersteller eines Produktes geschrieben wurden!“).
- Reputationsbasierende Mechanismen werten die Reputation eines Autors bzw. direkt die Qualität einer Information aus.

Die verschiedenen Mechanismen lassen sich in Trust-Policies kombinieren und auf verschiedene Situationen anpassen. Somit kann jeder Nutzer selber definieren, welche Maßstäben Informationen erfüllen müssen, um als qualitativ hochwertig betrachtet zu werden.

Die Trust-Policies implementieren Filter, die verschiedene Methoden für die Bewertung der Informationsqualität integrieren. Die Verwendung von reputationsbasierenden Methoden für die Beurteilung des Vertrauens von Informationsprovidern ist hier nur eine von vielen. Um nicht nur Filter für Informationsqualität zu benutzen, sondern den Anwender auch davon zu überzeugen, dass eine Information eine hohe Qualität besitzt, muss zweifelnden Anwendern die Möglichkeit gegeben werden, sich von der Richtigkeit der Inferenz selbst zu überzeugen. Das Mittel, das die TriQL-Trust-Architektur hierfür vorsieht, sind Erklärungen der Ergebnisse. Durch Wissen über die angewandte Trust-Policy, der Erläuterung von Inferenzen und der verwendeten Datenquellen kann sich der Anwender überzeugen lassen.

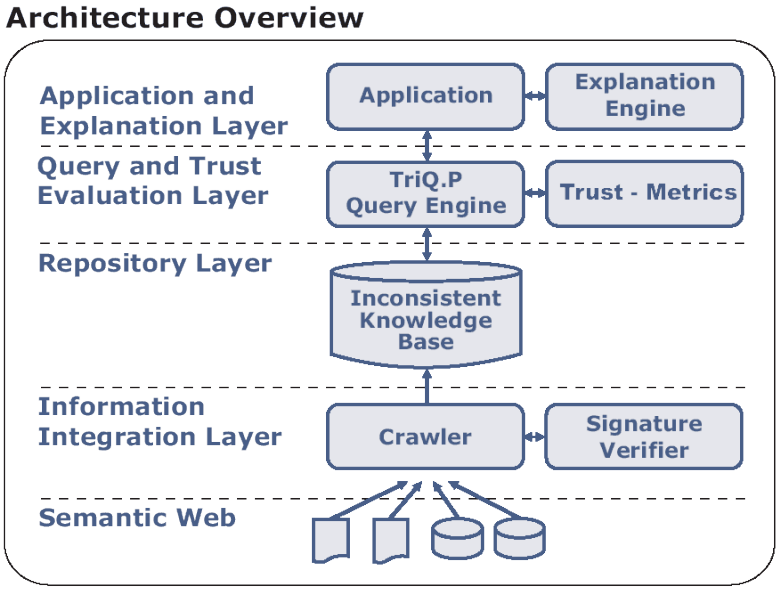

Abbildung 1: TriQL-Trust-Architektur

Die TriQL-Trust-Architektur unterteilt sich in mehrere Layer (Abbildung 1):

- Der *Information Integration Layer* sammelt Daten aus dem Semantic Web und verifiziert Signaturen, um die Autorisation der Daten zu bestimmen.
- Der *Repository-Layer* speichert die gesammelten Daten sowie Daten aus der Auswertung von Querys.
- Im *Query and Trust Evaluation Layer* befindet sich die TriQL.P Query Engine, die Suchanfragen unter der Berücksichtigung der ausgewählten Trust-Policy auswertet. In diesem Layer befinden sich alle Mechanismen, die die Informationsqualität evaluieren, insbesonderer die reputationsbasierenden Trust-Metriken, die im weiteren Verlauf dieser Diplomarbeit in den Mittelpunkt rücken.
- An oberster Stelle befindet sich der *Application and Explanation Layer*. Er beinhaltet Module zur Weiterverarbeitung oder Präsentation sowie der Erklärung der Ergebnisse.

1.4 Vertrauen (Trust)

Für eine genau Analyse von reputationsbasierenden Systemen ist es notwendig, sich über die Begriffe Vertrauen (Trust) und Reputation bewusst zu werden und zu einem einheitlichen Verständnis zu kommen. Vor allem die Inter-

pretation des Begriffs Vertrauen variiert stark in der Literatur.

Eine wichtige Differenzierung muss zwischen Vertrauen (Trust) und Sicherheit (Security) gemacht werden, da im Kontext von Sicherheit oft das Wort Vertrauen benutzt wird. Dort referenziert Vertrauen die Sicherstellung der Identität und der Vertraulichkeit in der Kommunikation von Entität mittels kryptologischer Verfahren. Ferner hat der Begriff Web-of-Trust (WoT) hier eine andere Bedeutung. Er referenziert ein Netzwerk aus gegenseitig signierten Schlüsseln mit asynchronen kryptographischen Verfahren. In dieser Diplomarbeit wird Vertrauen als ein soziales Konzept verstanden. Der Begriff Web-of-Trust kommt in diesem Kontext die Bedeutung eines Netzwerkes von Vertrauensbewertungen zwischen Entitäten zu.

Um das soziale Konzept Vertrauen zu verdeutlichen, welches in die Welt der Informationstechnologie übertragen werden soll, und um sich langsam an eine Definition anzunähern, soll ein kleines intuitives Beispiel veranschaulichen, was Vertrauen im Zwischenmenschlichen bedeutet. Im Anschluss sollen aus der Literatur bekannte Definitionen und Eigenschaften von Vertrauen in der Informationstechnologie aufgezeigt werden, bevor eine eigene Definition von Vertrauen und Reputation im Speziellen aufgestellt wird.

1.4.1 Intuitives Beispiel

Vertrauen sie ihrem Blumenhändler an der Ecke? Sie werden sagen: „Ja, ich wurde immer zuvorkommend bedient, er versteht es die Blumen zu binden und manchmal haben sie sogar zwei Wochen gehalten.“ Doch welche Aussagen stecken tatsächlich hinter diesen Worten? Vertrauen sie ihm wirklich?

An dieser Aussage lässt sich sehr gut demonstrieren, welche Faktoren für Vertrauensaussagen relevant sind. Sicher meinen sie nicht, dass sie ihrem Blumenhändler ihre Geldgeschäfte anvertrauen würden. Der Kontext des Blumenhandels bestimmt hier, bei welcher Art von Transaktionen sie ihrem Blumenhändler vertrauen. Vertrauen ist also abhängig vom Kontext. Zudem können sich verschiedene Kontexte überlagern und gegenseitig das Vertrauen beeinflussen. Sollten sie z.B. mit ihrem Blumenhändler einmal in Streit geraten, weil sie den

Preis für rote Rosen für zu hoch halten, wird ihr Vertrauen in ihn vor dem Hintergrund des Streites sinken. Eine prinzipiell vertrauenswürdige Person kann im Kontext einer Verteidigungssituation unglaubwürdig werden, da persönliche Interessen überwiegen und im Gegensatz zu ihren Interessen stehen.

Haben sie schon mal darüber nachgedacht, ob der Person, der sie die Blumen schenken, diese wirklich gefallen? Natürlich haben sie das, doch zeigt es, dass Vertrauen sehr subjektiv ist. In diesem Fall entfaltet sich die Subjektivität über den persönlichen Geschmack, aber auch politische Gründe oder schlechte Erfahrungen nehmen Einfluss.

Woher haben sie überhaupt von dem Geschäft ihres Blumenhändlers erfahren? Wurde es ihnen vielleicht von einem Nachbarn empfohlen? Was bewegte sie in den Laden ihres Blumenhändlers einzutreten, bevor sie persönliche Erfahrungen mit ihm gemacht hatten? Empfehlungen sind eine Form um positive Reputation auszudrücken. Ihr Nachbar, der ein guter Freund ist, hatte bereits selber positive Erfahrungen gemacht und schätzte den Blumenhändler bereits für seine Leistungen hoch ein. Die Reputation einer Person, die aus persönlichen Erfahrungen und aus Informationen von vertrauten Personen entsteht, nimmt entscheidend Einfluss auf das Vertrauen, welches wir Personen entgegenbringen.

Und woher wissen sie, dass der letzte Strauß, den sie bei ihrem Blumenhändler gekauft haben wirklich hübsch anzusehen war? Natürlich sehen sie dem Strauß an, ob er ihnen gefällt. Sie bewerten den Gegenstand der Transaktion. Für die Informationstechnologie bezeichnet man den Gegenstand der Transaktion als Kontent. Ihr Vertrauen, das der soeben gekaufte Blumenstrauß gefallen wird, ist selbstverständlich vom Blumenstrauß abhängig.

1.4.2 Der Vertrauensbegriff in der Literatur

Das Thema Vertrauen wird in der Wissenschaft interdisziplinär diskutiert. Wissenschaftler aus dem Bereich der Soziologie, Psychologie, Wirtschaftswissenschaften sowie der Informatik bemühen sich den Begriff zu verstehen und ihn zu definieren. Um den Begriff und seine Aspekte hinsichtlich der Analyse von reputationsbasierenden Systemen zu verstehen und den Fokus auf die Informa-

tionstechnologie nicht zu verlieren, werden mit Ausnahmen aus der Soziologie ausschließlich Referenzen aus der Informationstechnologie verwendet.

Stephen P. Marsh analysiert in seiner vielzitierten Ph.D. These [FTCC94] Vertrauen hinsichtlich der Entscheidungsfindung von autonomen Agenten, die über Kooperationen und Kollaborationen mit anderen Agenten entscheiden müssen. Er definierte Vertrauen auf der Grundlage der Definition des Soziologen Mortan Deutsch [CTTN62]. Sie besagt, das eine Person Alice einer Person Bob vertraut, wenn Alice einen Weg wählt, dessen Ausgang im Wesentlichen von Bob abhängt und dessen negativer Ausgang mehr Gewicht hat als der positive. Alice nimmt an, dass Bob positiv gegenüber ihr eingestellt ist, sie vertraut ihm. Wählt Alice den Weg nicht, so hat sie kein Vertrauen zu Bob. Die Anwendung dieser Definition von Vertrauen impliziert, dass sich Vertrauen erst in der Entscheidung für oder gegen einen Weg manifestiert.

Lik Mui definiert Vertrauen in seiner Ph.D. These [CMTR04] als subjektive Erwartung eines Agenten über das zukünftige Verhalten eines anderen Agenten. Die Erwartungshaltung resultiert dabei aus der quantitativen Zusammenfassung vergangener Transaktionen. Eine Entität entwickelt aus der Historie heraus eine persönliche Erwartungshaltung gegenüber einer anderen Entität und ihrem zukünftigen Verhalten. Dieses Model lässt die Möglichkeit offen, Vertrauen sogar ohne eine konkrete Entscheidungssituation zu formulieren.

Weit gehend einig ist sich die Wissenschaft über die Subjektivität und Kontextabhängigkeit von Vertrauen [CMTR04][CCBT04][FTCC94][CTSN05]. Jede Entität bildet sich eine eigene Meinung auf der Basis eigener Erfahrungen und den propagierten Erfahrungen anderer in ihrem Umfeld. In Abhängigkeit vom Thema entstehen verschiedene Meinungen über die Vertrauenswürdigkeit einzelner Entitäten zu bestimmten Themen.

Ein weiteres wichtiges Merkmal von Vertrauen ist das soziale Konzept Reputation. Personen mit einer starken Reputation wird tendenziell eher vertraut. Friedman und Resnik [SCCP01] beschreiben Reputation als eines der zwei wichtigsten Eigenschaften für das Eingehen von Kooperation. Neben der wieder-

holten Inanspruchnahme eines Services einer Entität X wird mittels Reputation versucht, das Risiko von negativen Folgen für zukünftige Kooperationen zu minimieren. Erfahrungen über vergangene Kooperationen mit X werden verbreitet und können von anderen Agenten zur Bildung einer persönlichen Meinung herangezogen werden.

Lik Mui behandelt den Zusammenhang zwischen Reputation und Vertrauen sehr intensiv in [CMTR04]. Er definiert Reputation als die Auffassung eines Agenten über die Intensionen und Normen eines anderen Agenten.

1.4.3 Definition von Vertrauen

Der dieser Arbeit zugrunde liegende Begriff Vertrauen ist wie folgt definiert:

> Eine Entität *A* vertraut einer Entität *B* hinsichtlich eines Services *S*, wenn *A* glaubt, das *B* den Service *S* in der erwünschten Qualität leisten würde.

Der Begriff Vertrauen betrifft ein soziales Konzept. Aus diesem Grund betrifft die Definition nur autonom handelnde Agenten, die als Entitäten bezeichnet werden. Sie können sich entscheiden, ob sie glauben, dass ein Angebot wahrhaftig ist und ihre Erwartungen erfüllen wird.

In dieser Definition wird keine Aussage über das Zustandekommen des Glaubens der Entität *A* gemacht, um keinen Weg der Vertrauensbildung auszugrenzen. Es können also sowohl willkürliche Meinungen, Vorurteile, Empfehlungen oder Datenanalysen die Grundlage des Vertrauens sein. Welche Informationen zur Vertauensbildung herangezogen werden, entscheidet jede Entität für sich.

Der verwendete Konjunktiv II impliziert, das *A* nicht nur unter einer persönlichen Entscheidungssituation Vertrauen zu *B* aufbaut, sondern auch ohne die Absicht den Service selber zu Nutzen die Vertrauenswürdigkeit von *B* beurteilen kann. Unter der Verwendung dieser Definition werden Empfehlungen als Mittel der Weitergabe von Vertrauensinformationen möglich.

Der Service und seine erwünschte Qualität geben den Kontext vor. Es kann sich dabei um eine Information handeln, wie eine Restaurantempfehlung, oder eine

Handlung wie die Reparatur eines Autos. Für den Restaurantbesuch mag unsere Erwartung sein, nicht enttäuscht zu werden, also eine für unseren Geschmack korrekte Information erhalten zu haben. Von der Autoreparatur werden wir sicherlich fordern, das sie fachgerecht und sicher ausgeführt wurde. Während das letztere Beispiel Servicequalität betrifft, geht es im ersten speziell um Informationsqualität. Der Kontext gibt also neben vielen weiteren Umständen auch Parameter für eine Bewertung vor. Da es in der Informationstechnik und insbesondere im Semantic Web um Informationen geht, soll die Informationsqualität hier in den Vordergrund rücken.

Informationsqualität ist sehr vielschichtig. Für die Bewertung von Informationen sind je nach Kontext verschiedene Aspekte relevant. Zu den wichtigsten Eigenschaften zählen die Aktualität, die Glaubwürdigkeit und Korrektheit. Richard Y. Wang u.a. [DQA02] listen in ihren Arbeiten über die Bewertung von Unternehmensinformationen weitere Aspekte auf.

Vertrauen bezieht sich also im Kontext des Semantic Web darauf, das eine Entität *A* glaubt, dass eine Entität *B* ihr Informationen mit hoher Qualität liefert. Die Eigenschaften der Informationsqualität, die dabei eine Rolle spielen, werden durch den Kontext vorgegeben. Ist eine Person dafür bekannt, dass sie immer glaubwürdige Informationen verbreitet, hilft ihr dies nicht, wenn im aktuellen Kontext nach aktuellen Informationen gesucht wird.

1.4.4 Definition von Reputation

Der Begriff Reputation wird in dieser Diplomarbeit wie folgt definiert:

> Reputation ist die Auffassung, die eine Entität *A* über die Einstellungen und Intensionen einer Entität *B* hat.

Diese Definition ist an die von Lik Mui formulierte Definition von Reputation *„preception that an agent has about another's intensions and norms“* [CMTR04] angelehnt. Sie lässt bewusst offen, auf welchem Wege die Reputation entstanden sein könnte. Eine Reputation kann also sowohl aus Vorurteilen hervorgehen, als auch das Ergebnis einer genauen Analyse von historischen Informationen sein. Auf welchem Wege eine Reputation zustande gekommen ist, hängt in erster Line

von den persönlichen Vorgehensweisen einer Entität und ihrem Wissen über andere Entitäten ab.

Da Reputation hier als ein soziales Konzept verstanden wird, bezieht sie sich ausschließlich auf autonom handelnde Agenten, die hier als Entität bezeichnet werden. Bewertungen von Informationen oder Services werden daher nicht als Reputation aufgefasst. Sie werden durch Informations- oder Servicequalität referenziert.

Reputation alleine führt allerdings noch nicht zu Vertrauen. Die Beziehung zwischen Vertrauen und Reputation wird in dieser Diplomarbeit wie folgt betrachtet:

> Eine Entität *A* vertraut einer Entität *B* auf der Basis der Reputation R_{AB}, die *A* von *B* besitzt, wenn A aufgrund von R_{AB} glaubt, dass B gemäß seinen Erwartungen handeln wird.

In diesem Fall nennt man die Reputation R_{AB} eine positive Reputation aus der Sicht von *A*. Lässt die Reputation R_{AB} die Entität *A* eher zu dem Schluss kommen, dass *B* die gewünschte Leistung nicht erbringen wird, so nennt man R_{AB} eine negative Reputation. Ob sich eine Reputation positiv oder negativ auswirkt, hängt vom Kontext ab.

1.5 Problemstellung

Die Literatur hält bereits eine Sammlung von reputationsbasierenden Trust-Metriken bereit, die für die Bewertung von Vertrauen geeignet sind, wie wir in Kapitel 3 sehen werden. Um sie einer Wiederverwendung in verschiedenen Systemen zuführen zu können, benötigt man ein grundlegendes Verständnis der Eigenschaften von der reputationsbasierenden Systemen, in denen reputationsbasierende Trust-Metriken eingesetzt werden, um für neue Systeme geeignete Metriken auswählen zu können. Zu diesem Zweck soll eine Klassifikation aufgestellt werden, die die wesentlichen Merkmale von reputationsbasierenden Systemen aufsplittet.

Diese Klassifikation soll anschließend auf aus der Literatur bekannte reputa-

tionsbasierende Systeme angewandt werden, um letztendlich geeignete reputationsbasierende Trust-Metriken für den Einsatz in Systemen für das Semantic Web wie der TriQL-Trust-Architektur zu identifizieren. Einige der identifizierten Metriken sollen anschließend für die TriQL-Trust-Architektur prototypisch implementiert und in das bestehende System integriert werden.

Da Vertrauen nicht aus der Inferenz einer Trust-Metrik allein entsteht, sondern durch das Verständnis der Inferenz, ist es von entscheidender Bedeutung, dass das Ergebnis einer Trust-Metrik für den Nutzer nachvollziehbar wird. Um dies zu erreichen sollen die ausgewählten Metriken um die Funktionalität eines Erklärungsgenerators erweitert werden, der den Evaluierungsprozess der Metrik erläutert.

1.6 Einordnung und Abgrenzung

In der Literatur sind bereits mehrere Klassifikationen für Trust-Metriken bekannt. Cai-Nicolas Ziegler erstellte im Rahmen seiner Doktorarbeit [TDRS05] eine allgemeine Klassifikation für Trust-Metriken. Er kommt zu dem Schluss, dass sich lokale zentralberechnende Gruppen-Trust-Metriken besonders für die Evaluierung von Vertrauen eignen.

Qing Zhang, Ting Yu und Irwin Keith stellten in [CSTF04] eine weitere Klassifikation für reputationsbasierenden Trust-Metriken vor. Sie gehen davon aus, dass die einzelnen Entitäten über Transaktionen miteinander kommunizieren, wobei eine Transaktion ein unidirektionaler kommunikativer Akt ist, der zwischen zwei Entitäten stattfindet. Die Vertrauenswürdigkeit eines Ziels des Vertrauens wird als die voraussichtliche Qualität seines Services in einer neuen Transaktion verstanden. Desweiteren Klassifizieren sie einige reputationsbasierende Trust-Metriken.

Beide Klassifikationen überschneiden sich teilweise. Sie lassen dabei die Unterscheidung von Metrikalgorithmen, die ohne jegliche Bedeutung der Ein- und Ausgabedaten über bestimmten Datenstrukturen rechnen, und Komponenten des Umfeldes, die Daten zu Informationen werden lassen sowie Informationen speichern und erstellen, außer Betracht.

Lik Mui beschreibt in seiner Ph.D. These [CMTR04] ein Klassifikationsschema für die Einordung von Reputation. Er nutzt seine Klassifikation, um ein mathematisches Model zur Berechnung von Reputation aufzustellen.

1.7 Aufbau der Diplomarbeit

Im folgenden Kapitel 2 wird die Klassifikation aufgestellt, die zur Untersuchung von reputationsbasierenden Systemen herangezogen werden soll. Zudem werden für das Semantic Web und die TriQL-Trust-Architektur relevante Eigenschaften von Trust-Metriken anhand der Klassifikation identifiziert. Anschließend werden in Kapitel 3 aus der Literatur bekannte reputationsbasierende Trust-Metriken klassifiziert, um ihre Eigenschaften besser zu verstehen und sie letztendlich einer Wiederverwendung zuführen zu können. Die Erläuterung der Implementierung geeigneter Trust-Metriken mit Erklärungsgeneratoren für die TriQL-Trust-Architektur sowie Anwendungsbeispiele befinden sich in Kapitel 4, bevor mit Kapitel 5 eine Zusammenfassung und eine Diskussion diese Arbeit abschließen.

2 Klassifikation von Trust-Metriken in reputationsbasierenden Systemen

Reputationsbasierende Trust-Metriken sind keine Erfindung des Semantic Web. Die meisten bekannten Trust-Metriken wurden für andere Szenarien als dem Semantic Web entwickelt. Sehr unterschiedliche Metriken wie PageRank [PRCR98] eingesetzt bei Google oder EigenTrust [EARM03] für Peer-To-Peer Netzwerke kamen in den letzten Jahren auf, um die Vertrauenswürdigkeit von Entitäten zu bewerten. Doch welche Trust-Metriken eignen sich für den Kontext des Semanic Web?

Um nicht immer wieder Metriken neu entwickeln zu müssen, sondern vorhandene Metriken in reputationsbasierenden Systemen wiederverwenden zu können, benötigt man ein Werkzeug, um die Eigenschaften von Metriken und der sie umgebenden Systemen beurteilen zu können. Ein tieferes Verständnis ihrer wesentlichen Eigenschaften wird benötigt, damit rationale Entscheidungen über die Einsatzmöglichkeiten von reputationsbasierenden Metriken in anderen Kontexten möglich werden. An dieser Stelle wird eine Klassifikation für reputationsbasierende Systeme vorgeschlagen.

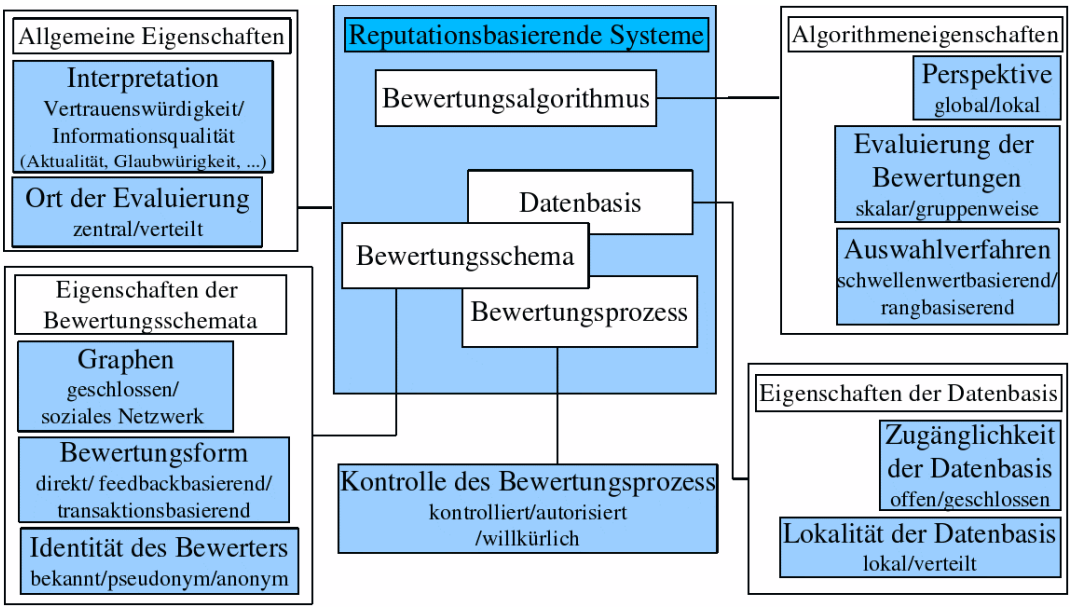

Abbildung 2: Merkmale reputationsbasierender Systeme

Reputationsbasierende Systeme lassen sich in mehrere Komponenten zerlegen,

die für die Bewertung von Reputation von Bedeutung sind. Neben den Eigenschaften des eigentlichen Bewertungsalgorithmus spielen Eigenschaften rund um die Speicherung, die Erstellung und die Organisation der Informationsbasis eine wichtige Rolle. In Abbildung 2 sind die einzelnen Komponenten eines reputationsbasierenden Systems und ihre Eigenschaften dargestellt. In Abschnitt 2.1 wird auf die einzelnen Komponenten eingegangen, um ihre Eigenschaften herauszuarbeiten. Die einzelnen Eigenschaften dienen als Dimensionen der Klassifikation. Weitere Eigenschaften, die sich nicht nominal erfassen lassen sowie Eigenschaften, die sich aus dem Zusammenwirken einzelner Eigenschaften aus der Klassifikation ergeben, werden in 2.2 diskutiert. Anschließend werden in Abschnitt 2.3 geeignete Klassen für reputationsbasierende Systeme für das Semantic Web identifiziert.

2.1 Klassifikationen

Die Beurteilung von reputationsbasierenden Systemen ebnet uns den Weg zum Einsatz ihrer reputationsbasierenden Trust-Metriken in neuen Systemen. Die Klassifikation soll uns mit ihren zentralen Eigenschaften vertraut machen. Dem Umfeld, in dem die eigentlichen Algorithmen der reputationsbasierenden Trust-Metriken arbeiten, kommt dabei ein genauso starkes Gewicht zu wie dem Algorithmus selber. Diese Betrachtungsweise ermöglicht es, auf der einen Seite die Funktionsweise der Algorithmen isoliert zu verstehen und auf der anderen Seite den Einsatz in reputationsbasierenden Systemen zu veranschaulichen.

Die einzelnen Komponenten reputationsbasierender Systeme und ihre Eigenschaften werden im Folgenden vorgestellt und sind in Tabelle 1 auf Seite 30 nochmals zusammengefasst.

2.1.1 Datenbasis

Die Art und Weise, in der Daten in reputationsbasierenden Systemen gespeichert werden, ist unabhängig vom verwendeten Algorithmus. Dennoch sind die Eigenschaften der Datenhaltung entscheidend für die inferierten Vertrauensbeziehungen. Daher können sie bei der Analyse von reputationsbasierenden Systemen nicht fehlen. Die Datenhaltung steht in engem Zusammenhang mit dem

Bewertungsprozess zur Datenerhebung und den verwendeten Bewertungsschemata. Die Datenbasis bezieht sich ausschließlich auf die Speicherung der Daten. Die Struktur der Daten wird durch die verwendeten Bewertungsschemata bestimmt. Die Umstände, unter denen die Erstellung von Reputationsinformationen geschieht, fällt dem Bewertungsprozess zu.

2.1.1.1 Sichtbarkeit der Datenbasis

An die Gestalt der Datenbasis werden verschiedene Anforderungen gestellt. Die Sichtbarkeit der Informationen, die für die Evaluierung von Metriken verwendet werden, nimmt entscheidenden Einfluss auf die Ausdrucksstärke von inferierten Beziehungen. Durch Zugangskontrollen wird die ausführende Entität einer Metrik wiedererkennbar für den Informationsprovider, der die Daten daraufhin gezielt für die ausführende Entität manipulieren kann. Metriken, die durch Zugangskontrollen geschützte Informationen verwenden, müssen also damit umgehen, das die Informationen gezielt verändert sein könnten.

In offenen Informationsstrukturen können Informationen nicht gezielt an Entitäten des Netzwerkes übermittelt werden. Die Öffentlichkeit schafft stärkeres Vertrauen in die Informationsqualität, da jede Entität selber die Eigenschaften der Information prüfen kann. Ob ein Informationsprovider also vertrauenswürdig ist hinsichtlich der Verbreitung von qualitativ hochwertigen Informationen, kann besser von einer Vielzahl an Entitäten geprüft und kommuniziert werden. Dies gilt z.B. für Metriken, die Informationen des WWW oder des Semantic Web zur Evaluierung verwenden.

2.1.1.2 Lokalität der Datenbasis

Die Art und Weise, in der die Datenbasis für eine Vertrauensevaluierung gespeichert ist, unterscheidet sich nach verteilter und zentraler Speicherung. Metriken, die für verteilte Architekturen wie P2P-Systeme entwickelt wurden, greifen in der Regel auf verteilt gespeicherte Daten zurück. Ein Peer wird hier als autonomer Softwareagent verstanden. Dabei speichern die einzelnen Peers nicht unbedingt ihre eigenen Daten. Im Falle von PeerTrust [RBTM03][TDPP02] werden die Daten eines Peers in anderen Peers gespeichert. Eine solche Archi-

tektur setzt Mechanismen voraus, mit deren Hilfe Informationen wieder aufgefunden werden können. Im WWW ist dieser Mechanismus z.B. durch URLs und das HTTP-Protokoll implementiert.

Der verteilten Datenhaltung steht die zentrale Datenhaltung gegenüber. Der gesamte Datenbestand wird in einer Instanz gespeichert und über sie abgerufen. Diese Instanzen sind häufig zentrale Autoritäten, die über die Verwaltung der Daten hinaus auch Mitglieder autorisieren und deren Reputation bewerten. Das von Ebay verwendete System für Bieter, Verkäufer und deren Ratings verwendet den Ansatz der zentralen Datenhaltung.

2.1.2 Bewertungsschema

Die Datenstruktur, die einem reputationsbasierenden System zugrunde liegt, wird durch die Bewertungsschemata organisiert. Sie definieren die Beziehungen zwischen den Entitäten und die Form der Informationen, die das System benötigt. Sie geben die Datenstruktur vor, die mit dem verwendeten Algorithmus kompatibel sein muss. Die Bewertungsschemata heben sich jedoch von den Datenstrukturen der Metrikalgorithmen ab, da sie nicht nur Daten enthalten, sondern mit einer Interpretation verknüpft sind, die wiederum Einfluss auf die Interpretation der Ergebnisse der Algorithmen nimmt.

2.1.2.1 Form von Bewertungen

Die Art in der Bewertungen in der Datenbasis enthalten sind, können verschiedene Formen annehmen. Je nach System werden Bewertungen mehr oder weniger direkt ausgedrückt. Direkte Bewertungen enthalten eine klare Aussage über die Stärke einer Beziehung zu einer anderen Entität. Im Fall der Advogato-Metrik von Raphael Levien [ARTM04] gibt es nur die Aussage, dass eine Entität einer anderen „voll vertraut“. Das Vokabular der Trust-Ontology [MSTO] des Mindswap Projektes [MDSP] sieht hingegen gewichtet Vertrauensbewertungen vor, die in Abhängigkeit eines Kontextes formuliert werden können. Für den Anwender einer Trust-Metrik ist bei der Verwendung von direkten Bewertungen nicht mehr nachvollziehbar, wie die verwendeten Aussagen zustande gekommen sind. Es könnte sich um eine persönliche Meinung handeln oder einen synte-

tischen Wert, der z.B. eine Zusammenfassung von Transaktionsfeedbacks sein könnte. Sogar willkürliche Werte könnten in die Datenbasis eingehen. Eine Trust-Metrik, die direkte Bewertungen als Datengrundlage verwendet, muss sich vor Manipulationen durch gezielte Falschbewertung schützen.

Feedbacks enthalten keine direkten Bewertungen einer Entität, sondern immer die Bewertung des Verhaltens einer Entität in einer Transaktion. Sie sind daher eine besondere Form der transaktionsbasierenden Daten. Sie sind Produkte von Kooperationen, die über das bisherige Verhalten einer Entität Auskunft geben. Über ihre Auswertung gelangt man zu Aussagen über das historische Verhalten einer Entität, welches für eine Prognose des zukünftigen Verhaltens verwendet werden kann. Wer also in der Vergangenheit, qualitativ hochwertige Informationen verbreitete, dem wird man eher vertrauen, in der Zukunft wieder so zu handeln. Friedman und Resnik bezeichnen die wiederholte Inanspruchnahme von guten Services als einer der wichtigsten Merkmale von Vertrauen [SCCP01].

Transaktionsbasierende Informationen sind Daten die aus der Aufzeichnung von Transaktionen resultieren. Dies können gesicherte Protokolle oder Referenzlisten von vorherigen Transaktionspartnern sein. Sie werden hier getrennt von feedbackbasierenden Informationen betrachtet, weil sie keine Bewertungen enthalten. Sie drücken nur das Vorhandensein von der Transaktion und ihre objektiven Umstände aus.

Diese Dimension steht in engem Zusammenhang zu der in [CSTF04] vorgestellten Dimension der transaktionsbasierenden und meinungsbasierenden Metriken. Die direkten Bewertungen gehen aber über meinungsbasierende Aussagen hinaus, da sie alle Bewertungen mit einschließen, die direkt über Dritte gemacht werden, also auch jene, die ein syntetischer Wert über transaktionsbasierenden Daten sind. Entscheiden ist, das die Entstehung der direkten Bewertungen für den Informationskonsumer nicht nachvollziehbar ist. Feedback- und transaktionsbasierende Metriken werden dort sonst nicht unterschieden und unter transaktionsbasierenden Metriken behandelt.

2.1.2.2 Grapheigenschaften

Reputationsbasierende Systeme bauen zwischen Entitäten Beziehungen auf. Egal ob die gespeicherten Informationen aus Transaktionen, deren Feedbacks oder aus direkten Bewertungen zwischen Entitäten stammen, sie beschreiben Beziehungen zwischen Entitäten. Auf einem abstrakteren Niveau können Beziehungen als benannte Kanten in einem gerichteten Graphen dargestellt werden. Die Beziehungen bauen somit einen Graphen auf, dessen Knoten die Entitäten eines Netzwerkes sind und deren Beziehungen benannte Kanten darstellen. Auf dieser Grundlage lassen sich verschiedene nominelle Eigenschaften aufzählen, die für die Auswertungen der Algorithmen und ihrer Interpretationen Auswirkungen haben.

In bestimmten Algorithmen werden geschlossene Graphen verwendet. Jeder Knoten in einem geschlossenen gerichteten Graphen, der eine Kante besitzt, die zum Knoten führt, besitzt auch eine Kante, die vom Knoten wegführt. Es gibt in einem solchen Graphen keine Blätter ,wie es etwa in baumartigen Graphen der Fall ist. Der bei Google [GOGL] eingesetzte PageRank-Algorithmus benötigt einen solchen Graphen für die Ausführung, in dem die Knoten Webseiten und die gerichteten Kanten Hyperlinks sind. Bevor der eigentliche PageRank-Algorithmus angewandt werden kann, werden die Links zu Webseiten, die im gescannten Internet einem Blatt entsprechen, heraus gefiltert [PRCR98].

Soziale Netzwerke sind ein weiteres Merkmal von Graphen in reputationsbasierenden Systemen. Sie zeichnen sich durch die Abbildung sozialer Beziehungen in einem Netzwerkgraphen aus. Die Annahme, dass es sich bei den verwendeten Informationen um Daten aus einem sozialen Netzwerk handelt, wird vor allem bei pfadbasierenden Metrik-Algorithmen verwendet. Pfadbasierende Metriken suchen Pfade zwischen Knoten entlang der Kanten in einem Graphen. In [TNSW03] und [TDRS05] wird argumentiert, dass sich skalierende Metriken, unter Annahme dass Bewertungen im Semantic Web ein soziales Netzwerken aufspannen, gut realisieren lassen. Sie stützt ihre Aussage auf Stanley Milgram's „Small World Problem“ [SMP67], nach dem alle Personen in den Vereinigten

Staaten von Amerika im Durchschnitt über 6 Kanten miteinander verbunden sind. Die darin enthaltene Annahme, dass die Pfadlängen zwischen Entitäten in einem sozialen Netzwerk nur logarithmisch mit der Anzahl der Entitäten steigt, wird als obere Schranke für den Aufwand von pfadbasierenden Metriken verwendet. Die von Jennifer Golbeck propagierte TidalTrust-Metrik [CTSN05] würde demnach eine obere Schranke für die Größe des zu untersuchenden Netzwerkes besitzen, die den Aufwand beschränkt.

Eine weiterer Typ von Graphen, der in Trust-Metriken Verwendung findet, ist der bipartite Graph. Der EigenRumor Algorithmus von Ko Fujimura u.a. [ERRB05] verwendet bipartite Graphen für die Berechnung der Reputation von Blog-Artikeln. Bipartite Graphen $G = (V,E)$ zeichnen sich dadurch aus, dass sie die Knoten V in zwei Mengen A und B unterteilen lassen, so dass für eine Kanten $(v,w) \in E$ im Graphen gilt, entweder $v \in A$ und $w \in B$ oder $w \in A$ und $v \in B$.

Die hier vorgestellten Merkmale von Graphen in reputationsbasierenden Systemen sind keineswegs vollständig. Es lassen sich weitere Eigenschaften unterscheiden.

2.1.2.3 Identität des Bewerters

Die Identität des Bewerters spielt für seine Bewertungen eine wichtige Rolle. Die Bewertungsinformationen selbst sind schließlich erst einmal nicht von besserer Qualität als andere Informationen. In offenen Systemen ohne einen kontrollierten Bewertungsprozess ist es daher sehr wichtig die Identität des Autors prüfen zu können. Über die Vertrauenswürdigkeit des Autor kann wiederum auf die Informationsqualität der Bewertungsinformationen geschlossen werden.

Werden in einem System Pseudonyme verwendet, so können diese ebenfalls für die Prüfung der Bewertungen verwendet werden. Die Voraussetzung ist allerdings, das die Pseudonyme nicht leicht gewechselt werden können, wie es Friedman und Resnik in [SCCP01] darlegen.

Bleiben die Bewerter anonym, so lassen sich die Bewertungen nicht bestimmten Entitäten zuordnen. Die Sicherung der Bewertungsqualität hinsichtlich der Mehrfachbewertung wird sehr schwierig. Die Bewertungen sind

somit genau so unsicher wie alle anderen Informationen.

2.1.3 Bewertungsprozess

Unter dem Bewertungsprozess versteht man den Ablauf und die Umstände unter dem die Bewertungen von Entitäten oder Kontent stattfinden. Dieser Teil eines reputationsbasierenden Systems ist geeignet, um durch das System Einfluss auf die Informationsbasis zu nehmen. Die Verwendung von Bewertungsschemata und die Identität von Bewertern kann hier kontrolliert werden.

2.1.3.1 Kontrolle des Bewertungsprozesses

Die Kontrolle des Bewertungsprozesses bezieht sich auf die Überprüfung der Autorisation und der Kontrolle geäußerten Informationen hinsichtlich der Einhaltung der Bewertungsschemata und der Informationsqualität der Bewertungen. Durch einen vollständig kontrollierten Prozess mit Identitätsprüfung des Autors, der Prüfung der Schemata und der Plausibilität der Informationen, können reputationsbasierende Systeme die Datenerstellung vollständig kontrollieren. Das System benötigt hierzu eine vertrauenswürdige Instanz, die die Prüfung vornimmt. Community-Server eigenen sich aufgrund ihrer zentralen Struktur für eine Implementierung solcher Systeme.

Mit weniger Kontrolle kommen Systeme aus, die die Prüfung auf eine Zugangsbeschränkung reduzieren. Sie prüfen lediglich, ob eine Entität autorisiert ist, eine Bewertung abzugeben. Die Verantwortung für die Einhaltung der Bewertungschemata und der Sicherstellung von Qualität der Bewertungen geht dabei auf die bewertende Entität über.

Willkürliche Bewertungen werden nicht kontrolliert. Jeder kann beliebiges äussern. Es gibt keine Instanz, die in irgendeiner Form eine Kontrolle ausübt. Die Prüfung der Qualität kann nur noch über die Plausibilität der Informationen geprüft werden. Diese Prüfung bleibt aber die Aufgabe des Nutzers der Information. Eine Trust-Metrik muss also selber auf die Sicherstellung der Qualität der verwendeten Informationen eingehen.

2.1.4 Bewertungsalgorithmus

In reputationsbasierenden Systemen ist der Bewertungsalgorithmus das Herzstück. Er stellt unabhängig von der Bedeutung der Eingabedaten und der Interpretation der Ergebnisse nur eine Rechenregel dar. Er legt dem Anwender Restriktionen hinsichtlich der Datenstruktur der Eingabe und der Ausgabe auf. Die Bedeutung der Eingabedaten und die Interpretation der Ergebnisse sind letztendlich abhängig vom Anwender bzw. vom System, in dem der Algorithmus angewandt wird.

2.1.4.1 Perspektive

Die Perspektive beeinflusst die Interpretation und die Gültigkeit einer Bewertung. Globale Trust-Metriken betrachten alle Entitäten eines Netzwerkes global [TDRS05]. Ihre Ergebnisse sind für alle Entitäten allgemein gültig, da ihre Bewertungen objektiv sind [CSTF04]. Sie eignen sich daher besonders für zentralisierte Umgebungen mit vollständiger Information. Dort werden die Bewertungen der Metriken von einer zentralen vertrauenswürdigen Instanz einmal für alle Entitäten der Umgebung berechnet. Das bekannteste Beispiel für ein solches System ist der Google-Index mit seinem PageRank-Algorithmus [PRCR98].

Eine lokale Perspektive berücksichtigt eher persönliche Interessen der einzelnen Entitäten [TDRS05]. Sie verwenden die Identität des Benutzers als zusätzlichen Parameter. Ihre Bewertungen sind daher nach [CSTF04] subjektiv bezüglich der Quelle des Vertrauens. Häufig entsteht die Subjektivität durch die Auswertung von Beziehungen der Quelle des Vertrauens zu ihren Nachbarn bzw. indirekt über Pfade von Beziehungen über ihre Nachbarn. Einen weiteren subjektiven Ansatz stellt die Berechnung von Ähnlichkeiten zwischen Benutzern zur Bewertung dar [ICTI05]. Das sich aufbauende Subnetzwerk ist das lokale Web-of-Trust (WoT) aus der subjektiven Sicht der Quelle des Vertrauens. Die in [TDRS05] vorgestellte Appleseed-Metrik verwendet diese Sichtweise.

Trust-Metriken mit einer lokalen Perspektive könne auf partiellen Trust-Graphen arbeiten [TDRS05]. Durch ihre starke Lokalisierung können bei

Inferenzen in Abhängigkeit der Quellen des Vertrauens sehr unterschiedliche Ergebnisse errechnet werden [CMTR04].

Felix Naumann merkt in seiner Doktorarbeit an [QDQA02], dass subjektive Kriterien immer vom Nutzer erfragt werden müssen, um seine persönlichen Einstellungen zu erfahren. Daher sind subjektive Verfahren immer auf die direkte oder indirekte Unterstützung des Nutzers angewiesen.

2.1.4.2 Auswahlverfahren

Trust-Metriken sollen die Entscheidungsfindung unterstützen. Bei einer großen Auswahl von Möglichkeiten sollen sie diejenigen aufzeigen, die für den Benutzer am gewinnbringendsten sind. Die in den Algorithmen verwendeten Auswahlverfahren unterscheiden sich dabei grundlegend in die Kategorien der rangbasierenden und schwellenwertbasierenden Auswahlverfahren.

Rangbasierende Verfahren bewerten eine Entität relativ zu ihren Konkurrenten. Sie liefern immer Ergebnisse, solange Entitäten die gewünschten Eigenschaften besitzen. Sie ermitteln die besten Entitäten, hinsichtlich des Bewertungskriteriums, also die Entität mit der besten Bewertung, was nicht bedeutet, dass diese auch wirklich gut ist. Der Benutzer bestimmt dabei wie vielen Entitäten er vertraut.

Rangbasierende Trust-Metriken sind immer auch Metriken die gruppenweise Beziehungen evaluieren, da ein Rang nur eine Bedeutung in der Relation zu weiteren Rängen in der selben Bewertung hat. Somit werden immer Gruppen von Entitäten bewertet. Die von Cai-Nicolas Ziegler vorgestellte Appleseed-Metrik [TDRS05] bewertet eine Gruppe von Entitäten im lokalen Umfeld der evaluierenden Entität und ordnet sie in eine Rangfolge der vertrauenswürdigsten Entitäten.

Schwellenwertbasierende Verfahren errechnen einen mindestens ordinalen Wert hinsichtlich des Bewertungskriteriums. Der resultierende Wert einer schwellenwertbasierenden Trust-Metrik kann z.B. als Bewertung des Vertrauens einer Entität oder der Glaubwürdigkeit einer Information interpretiert werden. Jede zu bewertende Entität wird einzeln bewertet. Das Verfahren trifft dabei selbst

Entscheidungen anhand eines vorgegebenen Schwellenwertes. Überschreitet die Bewertung einer Entität den Schwellenwert, so wird sie positiv hinsichtlich des Bewertungskriteriums bewertet. Die von Jennifer Golbeck vorgestellte TidalTrust-Metrik [CTSN05] evaluiert eine Bewertung zwischen Entitäten, die als Ersatz für eine nicht vorhandene direkte Bewertung verwendet werden soll. Eine vom Benutzer vorgegebene Schwelle entscheidet darüber, ob die Entscheidung zugunsten der bewerteten Entität ausfällt oder nicht.

Trust-Metriken, die schwellenwertbasierend vorgehen, können die Entitäten, deren Bewertungen den Schwellenwert überschreiten, wiederum rangbasierend sortieren. Dabei kann z.B. nach den errechneten Bewertungen sortiert werden, um die Entität mit der besten Bewertung zu ermitteln. Schwellenwertbasierende Trust-Metriken eignen sich, um ein Minimum an Vertrauen oder Informationsqualität sicherzustellen [CSTF04]. Ihr Vorgehen kann allerdings dazu führen, dass keine Entität als positiv bewertet wird, da keine Bewertung den Schwellenwert überschritt.

2.1.4.3 Evaluierung der Beziehung von Entität

Die Ergebnisse einer Trust-Metrik können für eine Beziehung zwischen zwei Entitäten oder für eine Gruppe berechnet werden. Gruppen-Trust-Metriken evaluieren Beziehungen zwischen Entitäten in eine Gruppe von Entitäten parallel [ARTM04]. Die Beziehung zu einem Gruppenmitglied ist abhängig von den Beziehungen, die die weiteren Gruppenmitglieder untereinander besitzen. Lokale Gruppen-Trust-Metriken berechnen das Vertrauen für einen Teil der Gesamtmenge der Entitäten eines Systems und verwenden die Beziehungen dieses Teils für die Berechnung.

Skalare Trust-Metriken hingegen bewerten die Beziehung einer Quelle des Vertrauens zu ihrem Ziel absolut. Ihre Ergebnisse sind von der Evaluierung von Beziehungen in andere Ziele vollkommen unabhängig interpretierbar. Cai-Nicolas Ziegler betont in seiner Doktorarbeit [TDRS05], dass die Auswertungen von skalaren Trust-Metriken immer nur lokale Gültigkeit haben und zentral berechnet

werden[1].

2.1.5 Allgemeine Eigenschaften

Die allgemeinen Eigenschaften beziehen sich auf ein gesamtes reputationsbasierendes System. Aus der Kombination der verwendeten Metrik-Algorithmen und Eigenschaften der Datenhaltung ergeben sich Eigenschaften für das Gesamtsystem. Die hier vorgestellten Eigenschaften beziehen sich auf Implementierungs- und Anwendungsdetails.

2.1.5.1 Ort der Evaluierung

Je nach Implementierung eines reputationsbasierenden Systems kann die Berechnung durch den Metrik-Algorithmus von jeder Quelle des Vertrauens selbst oder der Gesamtheit der Entitäten berechnet werden. Zentral rechnende reputationsbasierende Systeme berechnen Bewertungen auf einer einzigen Maschine. Sie setzen öffentliche verteilte Trust-Informationen oder in der berechnenden Maschine zentralisierte Informationen voraus. Verteilte Informationen müssen zur Evaluierung auf einer Maschine zusammengeführt werden, so das zentral ausgeführte Metriken inhärent zentralisierte Informationsstrukturen voraussetzen. Die TriQL-Trust-Architektur [CCBT04] verwendet die öffentlich verfügbaren und verteilten Informationen des Semantic Web und zentralisiert sie durch einen Crawler in einem zentralen Datenspeicher.

Verteilt rechnende reputationsbasierende Systeme lösen verteilte Probleme. In der Folge werden die Teilprobleme auf verschiedenen Maschinen berechnet. Jede vertrauenswürdige Instanz bekommt die Auswertungen ihrer Vorgänger und vereinigt sie mit den eigenen Bewertungen zu einer synthetischen Bewertung. Ein solcher Bewertungsprozess ist notwendigerweise asynchron und in der Konvergenz abhängig von der Bereitschaft der einzelnen Entitäten, eine Auswertung vorzunehmen.

1 In der Regel wird die zentrale Berechnung der Trust-Metrik von der Entität, die die Bewertung auslöst, durchgeführt.

Verteilte Trust-Metriken sind auch global, da die Teillösungen letztlich zu einer globalen Gesamtlösung zusammengesetzt werden. Alle Entitäten erhalten die gleichen Bewertungen zu allen anderen Entitäten im System. [TDRS05]

2.1.5.2 Interpretation der Inferierten Ergebnisse

Die Ergebnisse von Trust-Metriken sind oft keine allgemeinen Vertrauensbewertungen zwischen Entitäten. In Anhängigkeit des verwendeten Metrik-Algorithmus und der Semantik der verwendeten Informationen können die Ergebnisse Eigenschaften wie die verschiedenen Aspekte von Informationsqualität (Aktualität, Glaubwürdigkeit, Relevanz etc.) oder Servicequalität beurteilen. Die Verwendung von Transaktionsfeedbacks als Informationen für die Auswertung, deutet eher auf die Bewertung des Verhaltens einer Entität in vergangenen Transaktionen hin, wie es beim PeerTrust-System [RBTM03][TDPP02] geschieht. Werden direkte Vertrauensbewertungen verwendet, werden die inferierten Werte in der Regel wieder als Vertrauensbewertungen interpretiert, z.B. bei TidalTrust [CTSN05] ist dies der Fall. Vertrauen wird dabei in der Regel hinsichtlich eines bestimmten Kontextes bewertet. Wird Informationsqualität bewertet, so heisst Vertrauen z.B., dass einer Entität hinsichtlich der Verbreitung von aktuellen oder glaubwürdigen Informationen vertraut wird. Auch die Bewertung von Servicequalität könnte auf diese Weise mit Trust-Metriken vorgenommen werden. Die in der TriQL-Trust-Architektur verwendeten Trust-Policies könnten hierzu verwendet werden, um verschiedenen Bewertungen für die Beurteilung von Informationsqualität nutzbar zu machen. Die konkrete Interpretation hängt aber stark vom Einsatzzweck der Metriken ab.

Einheiten	*Dimensionen*	*Ausprägungen*
Metrikalgorithmus	Perspektive	global, lokal
	Auswahlverfahren	rangbasierend, schwellenwertbasierend
	Evaluierung von Beziehungen	gruppenbasierend, skalar
Datenbasis	Sichtbarkeit	offen, geschlossen
	Lokalität	verteilt, zentral
Bewertungsschema	Grapheigenschaften	geschlossen, partiell, soziales Netzwerk, bipartit
	Form von Bewertungen	direkt, feedbackbasierend, transaktionsbasierend
	Identität des Bewerters	bekannt, pseudonym, anonym
Bewertungsprozess	Erstellung von Bewertungen	kontrolliert, autorisiert, willkürlich
Allgemeine Eingenschaften	Ort der Evaluierung	verteilt, verteilt
	Interpretation	Vertrauenswürdikeit, Informationsqualität(Glaubwürdigkeit, Relevanz, Aktualität, ..), Servicequalität

Tabelle 1: Eigenschaften von reputationsbasierenden Systemen

2.2 Weitere Eigenschaften von reputationsbasierenden Systemen

In diesem Abschnitt wird auf Eigenschaften eingegangen, die sich aus dem Zusammenspiel der im vorherigen Abschnitt vorgestellten Dimensionen ergeben.

2.2.1 Bootstrap-Probleme[2]

Die Annahmen über die Datenbasis eines reputationsbasierenden Systems variieren von einfachen Hyperlinks (PageRank) bis hin zu direkt formulierten, gewichteten Vertrauensbewertungen, die eine WoT aufbauen (TidalTrust). Je stärker die Annahmen über die Datenbasis sind und je mehr die subjektive Perspektive berücksichtigt wird, umso stärker neigen die Metriken zu Problemen, wenn wenig Informationen für eine Analyse vorhanden sind. Insbesondere bei Systemen mit lokaler Perspektive, die über sozialen Netzwerken inferieren, muss ein nicht unerheblicher Aufwand betrieben werden, um da Netzwerk zu initialisieren. Jede Entität, die in ein solches soziales Netzwerk neu eintreten möchte, begegnet diesem Problem von neuem. Sie kennt noch niemanden, hat also

2 *Bootstrap-Problem: „Die Münchhausen-Methode bezeichnet allgemein, dass ein System sich selbst in Gang setzt. Die Bezeichnung spielt auf die deutsche Legende von Münchausen-Methode an, der sich an seinen eigenen Haaren aus einem Sumpf gezogen haben soll. In der amerikanischen Fassung benutzte er seine Stiefelschlaufen, was die englische Bezeichnung Bootstrapping für diese Methode begründete.*

Bei Computern bezieht sich der Begriff auf jeden Prozess, bei dem ein einfaches System ein kompliziertes System aktiviert. Es ist das Problem, ein bestimmtes System zu starten, ohne dass dieses bereits funktioniert. Dies scheint so unmöglich wie sich selbst aus dem Sumpf zu ziehen. (...)“ (Wikipedia, „Münchausen-Methode“)

auch noch keine Beziehungen zu anderen. Beziehungen zu anderen Entität zu inferieren wird somit unmöglich, da keine Beziehungen zwischen den Entitäten existieren. Der Aufbau eines neuen WoT gestaltet sich zudem schwierig durch das geringe Grundvertrauen, das Newcomern entgegengebracht wird[3].

Eine Lösung für dieses Problem können mehrstufige Evaluierungsprozess für Beziehungen sein. Guha [ORS03] schlägt vor, zuerst Vertrauen aus der subjektiven Sicht der Entität zu evaluieren, d.h. eine Metrik mit lokaler Perspektive zu verwenden, und erst wenn nicht genügend positive Ergebnisse gefunden wurden, auf eine objektive Bewertungen mit einer Metrik für eine globale Perspektive umzuschalten. Auf diese Weise könnten gute Ergebnisse selbst dann erzielt werden, wenn ein Newcomer sich erst in ein soziales Netzwerk integrieren muss. Er verwendet erste Beziehungen auf objektiver Basis und kann nach den ersten Erfahrungen sich selbst ein Urteil bilden, welches anschließend in eigene Bewertungen und den Aufbau seines persönlichen WoT einfließen kann.

2.2.2 Privatsphäre

Privatsphäre und Vertrauen erscheinen diametral zu einander. Systeme, die Metriken für offene Informationsstrukturen einsetzen und transaktionsbasierende oder feedbackbasierende Informationen zur Auswertung heranziehen, scheinen keinerlei Privatsphäre zu gewährleisten. Schließlich kann jedermann auf alle Informationen bis ins keinste Detail zugreifen. Auf der anderen Seite benötigen Metriken für verteilte, geschlossene Informationsstrukturen mit verteilter Berechnung nur wirklich notwendige Informationen in der Form von syntetischen Bewertungen.

Der ersten Metrik würde man intuitiv eine effektive und vertrauenswürdige Bewertung bescheinigen, da man in der Lage ist, die Auswertung selbst vorzunehmen und alle Nutzer gemeinsam in der Lage sind, die Korrektheit der Informationen zu prüfen. Bei der zweiten Metrik würde man eher skeptisch sein,

3 Friedman und Resnik argumentieren in [SCCP01], das Newcomer in IT-Systemen durch das leichte Tauschen von Pseudonymen im Internet eine prinzipiell schlechtere Reputation besitzen, da sie potentiell Akteure sein könnten, die mit ihrem Newcomer-Dasein ihre schlechte Reputation mit ihrem bisherigen Pseudonym abstreifen wollen. In Kontrast hierzu steht die reale Welt, in der das Wechseln der Identität hohen Aufwand fordert.

schließlich kontrolliert man den Auswertungsprozess nicht mehr selbst und kann die Qualität der Informationen nicht prüfen, da man die Orginaldaten nicht kennt.

In [CSTF04] merken Qing Zhang u.a. an, dass direkte Vertrauensbewertungen in Form von Meinungen die Privatsphäre besser schützen können. In offenen Informationsstrukturen würden so nur noch Meinungen sichtbar, nicht aber die meinungsbildenden Informationen.

Einen weiteren Kompromiss zwischen Vertrauen und Privatsphäre stellen Community-Server dar, die als zentrale vertrauenswürdige Autorität auftreten. Sie zeichnen sich in der Regel durch eine zentrale geschlossene Datenhaltung aus und berechnen Bewertungen zentral mit globalen Metriken. Community-Mitglieder müssen ihre vertrauensrelevanten Informationen nur gegenüber dem Community-Server bekannt geben. Die Privatsphäre wird somit besser geschützt.

2.2.3 Skalierbarkeit

Die Skalierbarkeit hängt neben der Komplexität des eigentlichen Bewertungsalgorithmus wesentlich von der Menge der Informationen, die von der Metrik ausgewertet werden müssen, und der Lokalität der Informationen ab. Starken Einfluss auf die Informationsmenge nimmt die Art der Bewertung.

Angenommen, eine Entität unterhält Beziehungen zu n Nachbarn. Bei der Verwendung einer Metrik auf der Basis von direkten Bewertungen müsste X also n Bewertungen abgeben, um seine Beziehungen im Netzwerk publik zu machen. Bei transaktions- oder feedbackbasierenden Informationen müsste X mindestens zu jedem Nachbarn eine Aussage machen. Häufig sind die Informationsmengen aber deutlich größer, da die durchschnittliche Menge an Transaktionen pro Kooperationspartner m als Multiplikator hinzu kommen. Somit wächst die Menge an Aussagen, die es in einer Evaluierung zu berücksichtigen gilt, auf $n * m$ Aussagen an.

Eine Verringerung des Aufwandes von Trust-Metriken können pfadbasierende Metriken auf der Basis von sozialen Netzwerken sein. Sie verwenden direkte Bewertungen und sind in ihrem Aufwand aufgrund der sozialen Struktur des Netzwerkes beschränkt (siehe 2.1.2.2).

2.2.4 Qualitätssicherung der Bewertungsinformationen

Die Grundlage für qualitativ hochwertige Bewertungsinformationen sind klare handhabbare Bewertungschemata. Nach den von Thomas R. Gruber geprägten Designkriterien für Ontologien [TPDO93] sind vor allem die Klarheit, die Kohärenz und die minimale Annahmen für das Bewertungsschema von Bedeutung. Zudem können Aussagen über den Zeitpunkt der Äußerung einer Bewertung sowie weitere Informationen über den Kontext der Bewertung die Qualität verbessern.

Die Qualitätssicherung der Bewertungen wird durch einen kontrollierten Bewertungsprozess sichergestellt. Unter der Voraussetzung, das ein qualitativ hochwertiges Bewertungsschema verwendet wird, kann eine zentrale Instanz über die Einhaltung des Schemas und der Plausibilität der Bewertungsinformationen wachen. An dieser Stelle müssen die Bewertungsinformationen wiederum bewertet werden. Es entsteht eine Rekursionskette von Bewertungen, die durchbrochen werden muss. Soziale Netzwerke mit Bewertungen der Bewerter können hier eine Lösung sein, da sie durch den Algorithmus selbst Bewerter mit einer schlechten Bewertungsqualität selektieren können.

2.2.5 Resistenz gegen Manipulation

Reputationsbasierende Systeme müssen sich als resistent gegen Manipulationen erweisen. Nur Systeme, die einen Angriff durch verfälschte Informationen abwehren können, können sich im Feld der Trust-Metriken behaupten. Ein großes Problem sind die von Friedman und Resnik [SCCP01] diskutierten „Cheap Pseudonyms“. Wenn Entitäten in einem Netzwerk wie dem Internet Pseudonyme verwenden und diese leicht wechseln können, besteht für Entitäten, die sich unkooperativ verhalten die Möglichkeit, ihre in Folge schlechte Reputation leicht abzustreifen und neu zu beginnen. Lösen ließe sich dieses Problem z.B. mit der Verwendung von Public Key Infrastrukturen (PKI), um die Identität einer Entität zu prüfen. Friedman und Resnik merken allerdings an, dass Identifikation nicht für jede Umgebung geeignet ist. So gibt es Internetforen, die nur durch die Sicherstellung von Anonymität funktionieren. Ein Forum zu

Diskussion für HIV-Patienten kann ohne Anonymität nicht bestehen. Umgehen lässt sich dieses Problem nur durch System, die die Identität von Bewertern ausser Acht lassen. Bei der Verwendung des PageRank-Algorithmus [PRCR98] können die Bewerter anonym bleiben.

Raphael Levien befasste sich in seiner Ph.D. These [ARTM04] mit angriffsresistenten Trust-Metriken. Er formulierte hinsichtlich des Angriffs auf die Datenbasis durch Manipulationen das „bottleneck property“. Es besagt, dass die Bewertung, die eine Entität *s* einer Entität *t* gibt, sich nicht signifikant ändert, wenn *t* die Bewertungen seiner Nachfolger ändert. Bewertet eine gute Entität eine schlechte, so ist deren Bewertung nicht abhängig von den Manipulationen der schlechten Entität. Trust-Metriken, die diese Bedingung erfüllen, isolieren schlechte Entitäten vom Netzwerk, so dass sie kaum Einfluss auf Bewertungen nehmen können. Levien schließt auf der Basis seiner Untersuchungen seiner Advogato Trust-Metrik und dem PageRank-Algorithmus, dass Gruppen-Trust-Metriken im Gegensatz zu skalaren Trust-Metriken das „bottleneck property“ erfüllen.

Metriken für direkte Bewertungen können leichter manipuliert werden, da sich die Entstehung der Aussage nicht nachvollziehen lässt. Qing Zhang u.a.[CSTF04] bestätigen dies für meinungsbasierende Trust-Metriken. Öffentliche transaktions- oder feedbackbasierende Bewertungen können hingegen schwerer manipuliert werden, da die zugrunde liegenden Daten von Transanktionspartnern überprüft werden können.

2.3 Evaluierung der Klassen

Die in Abschnitt 1.3 vorgestellte TriQL-Trust-Architektur soll als Anwendung des Semantic Web um reputationsbasierenden Trust-Metriken erweitert werden und somit zu einem reputationsbasierenden System werden. Doch welche Metriken sind geeignet? Die im Abschnitt 2.1 entwickelten Dimensionen sollen in diesem Abschnitt eine Anwendung finden. Die allgemeine Anforderungen an Anwendungen des Semantic Web sowie der TriQL-Trust-Architektur als Applikation des Semantic Web geben dabei die Parameter für die Analyse

geeigneter Metriken vor.

Die TriQL-Trust-Architektur geht von vornherein davon aus, dass Informationsqualität subjektiv [CCBT04] ist und somit aus einer lokalen Perspektive heraus gesehen wird[4], da ein Nutzer prinzipiell seine Interessen vertreten sehen möchte. Damit haben geeignete Trust-Metriken eine lokale Perspektive. Sie sollen das Bewertungen ausgehend von der Entität berechnen, die die Architektur betreibt und dabei persönliche Interessen wahren. Die meisten lokalen Trust-Metriken realisiert die Lokalität durch die Auswertung von Beziehungen zu direkten Nachbarn in einem sozialen Netzwerk.

Eine globale Perspektive scheint schon aus praktischen Gesichtspunkten für das Semantic Web ungeeignet. Selbst wenn das Semantic Web um einige Dimensionen kleiner bleibt als das WWW, so überschreiten die enorme Anzahl an Entitäten und ihre Bewertungen doch die Kapazitäten eines normalen Rechners. Globale Trust-Metriken für das Semantic Web sind daher für wenig geeignet [TDRS05]. Lediglich Benutzer wie Goolge für das WWW haben die Möglichkeit genügend Leistung für globale Bewertungen bereitzustellen. Dennoch gibt es Einsatzzwecke für die globale Trust-Metriken eine Lösung darstellen, z.B. wenn lokale Trust-Metriken aufgrund eines Bootstrap-Problems ungenügend Ergebnisse liefern. Zudem lassen sie sich auf lokalisierten Informationen ausführen, so dass über die Vorauswahl an Informationen eine performante und lokalisierte Bewertung möglich wird.

Im Semantic Web kann keine vollständige Information vorausgesetzt werden. Die Datenmengen lassen es schlicht nicht zu. Da das Semantic Web ebenso wie das World Wide Web eine rein verteilte und offene Informationsstruktur hat, ist es so gut wie unmöglich sicherzustellen, dass alle vertrauensrelevanten Informationen jeder Entität berücksichtigt wurden. Die Auswertung von Informationen aus dem Semantic Web wird daher immer unter der Berücksichtigung der „Open

4 Obwohl Vertrauen in der TriQL-Trust-Architektur als subjektiv betrachtet wird, lässt die TriQL-Trust-Architektur auch objektive Kriterien zu. Je nach eingesetzter Trust-Policy kann ein Anwender zwischen objektiven und subjektiven Bewertungskriterien wechseln. Zu dem gibt es wie in 2.2.1 beschrieben Anwendungsbeispiele, in denen die Anwendung von objektiven Kriterien (globale Perspektive) auch in Systemen mit lokaler Perspektiv Sinn macht.

World Assumption“[5] durchgeführt. Dies spiegelt sich ebenfalls in den Standards der Semantic Web Activity wieder, so z.B. in der Definition der Semantik des RDF Standards [RDFS]. Trust-Metriken für den Kontext des Semantic Web sollten daher mit partiellen Informationen auskommen. Einige Trust-Metriken mit lokaler Perspektive benötigen lediglich sehr kleine Subnetzwerke, weil sie durch die Voraussetzung eines soziales Netzwerk den Aufwand für das Suchen von Entität reduzieren können.

Die verteilte Berechnung von Vertrauen scheint für den Einsatz im Semantic Web nicht praktikabel. Sie setzt eine standardisierte Plattform voraus, die gewährleistet, das alle Entitäten den selben Bewertungsalgorithmus anwenden. Dies zu gewährleisten ist schwierig, da die einzelnen Entitäten im Semantic Web autonom agieren, und kein Interesse an einer Kooperation für die Vertrauensauswertung zeigen müssen. Auch die Individualität von Nutzern würden so nicht berücksichtigt. Nicht jeder Nutzer muss mit dem Standardalgorithmus zufrieden sein. Vertrauensbewertungen müssen schließlich aus der Sicht des Benutzers mit geeigneten Trust-Metriken berechnet werden, damit er den Ergebnisse glauben kann. Die Anpassung der Evaluierung mit individuellen Trust-Metriken und Trust-Policies scheint nur in einer zentralisierten Berechnung auf einem vertrauenswürdigen Rechner effektiv realisierbar. Die Austauschbarkeit von Trust-Metriken soll in der TriQL-Trust-Architektur durch ein offenes Interface für Trust-Metriken und den Einsatz von individuellen anpassbaren Trust-Policies realisiert werden. Eine zentralisierte Evaluierung scheint also die geeignetere Wahl. [TDRS05][CCBT04]

5 Die „Open World Assuption“ besagt, das Aussagen, die weder bestätigt noch widerlegt werden können, als unbekannt gelten. Dies steht im Gegensatz zur „Closed World Assumption“, die in diesem Fall davon ausgeht, das die Aussage falsch ist. Zudem führen Widersprüche in der Informationsbasis nicht zu Fehlern, sondern zu Inferenzen. Werden z.B. in einer Datenbasis für eine Person zwei Väter namentlich genannt, so ist dies nach der „Closed-World-Assumption“ ein Fehler, nach der „Open-World-Assumtion“ würde stattdessen geschlossen, das die Namen die gleiche Person bezeichnen.

Aus praktischen Gesichtspunkten scheint eine verteilte Berechnung des Vertrauens nicht möglich.

1. Die Anzahl aller Entitäten im Semantic Web ist sehr groß. Jede Entität müsste über alle anderen Entitäten im Semantic Web Beziehungen verwalten.
2. Die Konvergenz einer verteilten Trust-Metrik hängt von der Bereitschaft der einzelnen Agenten ab, ihre Teilaufgabe der Metrik zu lösen. [TDRS05]

Cai-Nicolas Ziegler schlägt in [TDRS05] lokale zentralisierte Gruppen-Trust-Metriken für das Semantic Web vor. Er entscheidet sich für Gruppen-Trust-Metriken, da sie nach Raphael Levien's „bottleneck property“ [ARTM04] resistenter gegen Manipulationen sind als skalare Metriken. Lässt man das „bottleneck property“ ausser Betracht, so scheint die Entscheidung zwischen Gruppen-Trust-Metriken oder skalaren Trust-Metrik relativ offen.

Die Entscheidung über das Auswahlverfahren bleibt für das Design von Systemen für das nach Semantic Web. Während rangbasierende Metriken immer alle Entität gemeinsam bewerten und zueinander in Beziehung setzen, evaluieren schwellenwertbasierende Metriken jede Entität einzeln.

Die Informationsstrukturen im WWW wie im Semantic Web sind offen und verteilt. Da verteilte Berechnungen unpraktikabel sind, müssen die für eine Bewertung notwendigen Informationen bei der zentral berechnenden Instanz zentralisiert werden. In der TriQL-Trust-Architektur wird dies durch einen Crawler für das Sammeln der Informationen und ein Repository zur Speicherung umgesetzt.

Die Form, in der Informationen für die reputationsbasierende Bewertung gespeichert sind, unterliegen keiner prinzipiellen Beschränkung. Die Bewertungsschemata sind also vollständig von der Implementierung des einzelnen reputationsbasierenden Systems abhängig. Günstig sind allerdings Bewertungschemata,

die dem Benutzer keinen zusätzlichen Aufwand zur Verwaltung von Reputationsinformationen auferlegen. Der PageRank-Algorithmus [PRCR98] erscheint in diesem Sinne optimal, da es einfach ohnehin vorhandene Hyperlinks nutzt.

Die Kontrolle des Bewertungsprozesses kann ebenfalls beliebige Gestalt annehmen. Typischer Weise werden Informationen im Semantic Web wie im WWW willkürlich geäussert, dennoch ist die Verwendung von kontrollierten Prozessen oder autorisierten Äusserungen denkbar, solange die erzeugten Informationen offen zugänglich sind. Bei der Verwendung von willkürlich Äusserungen von Bewertungen müssen Trust-Metriken selber Methoden entwickeln, wie sie die Qualität der Bewertungen sicherstellen können.

Die Interpretation hängt im wesentlichen von der Anwendung ab. Mit der TriQL-Trust-Architektur soll die Informationsqualität beurteilt werden, damit Informationen mit geringer Qualität heraus gefiltert werden können. Anwender formulieren zu diesem Zweck Trust-Policies, in denen sie Trust-Metriken einsetzen können. Welche Interpretation sie den Ergebnissen der Metriken beimessen und in welcher Beziehung sie zur Informationsqualität stehen, legen sie dabei selber fest.

Aus den ausgewählten Eigenschaften für reputationsbasierende Systeme für das Semantic Web und der TriQL-Trust-Architektur ergeben sich Konsequenzen für die in Abschnitt 2.2 vorgestellten Querschnittseigenschaften. Durch die offene und verteilte Informationsstruktur wird die Realisierung von Privatsphäre sehr schwierig. Benötigte Informationen sind prinzipiell für alle sichtbar. Zudem gestaltet sich die Qualitätssicherung sehr schwierig, da es keine Autorität gibt, die über der Einhaltung von Bewertungschemata wacht. Dies macht einfache Bewertungsschemata umso wichtiger. Die Fragen zur Resistenz gegen Manipulationen, der Skalierbarkeit und dem Auftreten von Bootstrap-Problemen bleiben weiterhin unbeantwortet. Sie hängen im wesentlichen von den ausgewählten Trust-Metriken ab. Hier bleibt Gestaltungsspielraum für die Entwicklung von Anwendungen des Semantic Web.

Zusammengefasst, werden in dieser Arbeit die folgenden Eigenschaften von

reputationsbasierenden Systemen für den Einsatz im Semantic Web gefordert:

- lokale Perspektive (nur als Fallback-Lösung auch globale Perspektiven)
- zentralisierte Evaluierung
- offene verteilte Datenbasis, die in Teilen zentralisiert werden muss
- vorwiegend willkürliche Erstellung von Reputationsinformationen

3 Reputationsbasierende Trust-Metriken

Die Literatur hält viele Metriken bereit, die prinzipiell geeignet sind, Beziehungen zwischen Entitäten zu analysieren. Die Bandbreite reicht von Algorithmen, die ursprünglich für Bewertung von Webseiten geschaffen wurden (PageRank [PRCR98]), bis hin zu Metriken, die explizit für den Einsatz im Kontext des Semantic Web entwickelt wurden. Nachdem in Kapitel 2 die Eigenschaften von reputationsbasierenden Trust-Metriken und der sie umgebenden Systeme diskutiert wurden, soll die entwickelte Klassifikation nun auf reputationsbasierende Systeme und Metriken angewandt werden.

In den folgenden Abschnitten 3.1 bis 3.7 werden sieben aus der Literatur bekannte Systeme klassifiziert, und anschließend anhand der in Abschnitt 2.3 erarbeiteten Eigenschaften zum Semantic Web in Bezug gesetzt. Die Analyse soll dabei zuerst den Metrikalgorithmus unabhängig von der Interpretation der Ein- und Ausgaben vorstellen und klassifizieren. Anschließend wird das System, in dem der Algorithmus in der Literatur vorgestellt wurde, als Beispiel für eine Implementierung untersucht. Weitere Metriken, die für die Entwicklung der Klassifikationen berücksichtigt wurden, aber hier nicht im Detail analysiert werden, werden im Anschluss in Abschnitt 3.8 kurz eingeordnet.

3.1 Appelseed

Der von Cai-Nicolas Ziegler in seiner Doktorarbeit [TDRS05] vorgestellte Appelseed-Algorithmus wurde speziell für den Einsatz im Semantic Web geschaffen. In einem iterativen Prozess wird über einem Graphen von gerichteten und gewichteten Kanten ein bestimmtes Potential mittels Spreading Activation Strategien [SSCN03] verteilt. Das Ergebnis ist ein Ranking, dass Knoten des Graphen nach dem Potential ordnet, welches über den gerichteten Kanten an sie verteilt wurde.

Eigenschaften des Metrikalgorithmus

Der Appleseed-Algorithmus inferiert über einem Graphen aus gerichteten und gewichteten Kanten, die als direkte Bewertungen betrachtet werden. Zu Beginn des Bewertungsprozesses wird in den Knoten der Quelle des Vertrauens x eine

gewisse Potential $in(x)_0$ injiziert, welches auf die Nachbarn im Netzwerk verteilt werden soll. Jeder Nachbar behält dabei eine gewisse Menge des Potentials für sich anteilig von dem Potential, welches über das Netzwerk in ihn injiziert wird. Mit einem Streuungsfaktor d bestimmt man den Anteil des injizierten Potentials, das ein Nachbar nicht für sich behält. Je größer der Streuungsfaktor ist, je mehr Entitäten werden in das Ranking einbezogen. Der Anteil, den ein Knoten nicht für sich behält, wird an seine Nachbarn verteilt unter Berücksichtigung der Gewichte der Kanten zu den Nachbarn. Dieser Prozess wird wiederholt, bis die Änderungen des in den einzelnen Knoten gespeicherten Potentials zur vorherigen Iteration in keinem Knoten das Minimum T_c überschreitet.

Cai-Nicolas Ziegler ordnete seine Appleseed-Metrik nach einem eigenen Klassifikationsschema als eine lokale Gruppen-Trust-Metrik ein. Die lokale Perspektive entsteht aus der Position der Quelle des Vertrauens im Netzwerkgraphen. Die Bewertung der einzelnen Knoten geschieht durch die verwendeten Spreading Activation Strategien in Abhängigkeit von den Bewertungen der anderen Knoten im Netzwerk. Auf diese Weise sind die Bewertungen also immer von der Gruppe der durch den Algorithmus berücksichtigten Knoten aus dem lokalen WoT abhängig.

Datenbasis

Die Appleseed-Metrik wurde vor dem Hintergrund des Einsatzes in dezentralen Recommender-Systemen geschaffen. Dies lässt auf eine verteilte offene Informationsstruktur schliessen. Dennoch ist der Algorithmus in keiner Weise an diese Vorgabe gebunden. Der Einsatzzweck lässt außerdem auf eine eher willkürliche Äusserung von Bewertungen schliessen. Aber auch hier sind keine festen Annahmen für ein Zielsystem formuliert.

Die Appleseed-Metrik wurde für die Berechnung eines Rankings der vertrauenswürdigsten Entitäten geschaffen. Unter der Voraussetzung, dass der zugrunde liegende Graph ein soziales Netzwerk von gewichteten Vertrauensbewertungen zwischen den Entitäten eines Netzwerkes aufspannt und dass das zu verteilende Potential eine Menge an Vertrauen ist, kann das Ranking als eine

Ordnung der vertrauenswürdigsten Entität in persönlichen WoT der Quelle des Vertrauens interpretiert werden.

Die Bewertung von Entitäten wird von jeder Entität des Netzwerkes auf einer gegebenen Skala vorgenommen. Entitäten müssen dabei mindestens mit einem Pseudonym bekannt sein, um sie direkt bewerten zu können.

Weitere Eigenschaften

Wie andere lokale Metriken, die über sozialen Netzwerken inferieren, leidet die Appleseed-Metrik unter einem Bootstrap-Problem. Eine Entität, die noch keine direkte Bewertungen gegenüber einer anderen Entität geäußert hat, erhält durch die Appleseed-Metrik nur ein leeres Ranking. Zudem werden direkte Bewertungen selbst geäußert und verwaltet, was einen hohen Aufwand und ein gutes Verständnis des Bewertungsschemas beim Benutzer voraussetzt.

Die Appleseed-Metrik bietet einen hohen Schutz vor Manipulationen des Rankings durch einzelne Entitäten. Obwohl nicht nachvollziehbare direkte Bewertungen für die Berechnung herangezogen werden, bietet die Kontrolle des sozialen Netzwerkes und die Berechnung des Vertrauens in einer Gruppen-Trust-Metrik nach Levien [ARTM04] einen guten Schutz. Ein Problem bleiben alleine „Cheap Pseudonyms“ [SCCP01], die es prinzipiell ermöglichen, das eine reale Entität sich hinter mehreren Pseudonymen verbirgt und so ein höheres Gewicht in einem sozialen Netzwerk erlangen kann.

Reputationsbasierende Systeme, die die Appleseed-Metrik einsetzen, können von einer guten Skalierung des Systems mit der Anzahl der Entitäten im Netzwerk profitieren, da die Skalierung im wesentlichen nicht von der Größe des Netzwerkes abhängt. Der Rechenaufwand wird vornehmlich durch die Menge des injizierten Vertrauens $in(x)_0$ und den Streuungsfaktor d bestimmt. Sie entscheiden darüber, wie viele Entitäten aus dem lokalen Umfeld der Quelle des Vertrauens für die Aufstellung des Rankings berücksichtigt werden. Das Netzwerk kann daher beliebig wachsen, ohne dass sich der Aufwand im wesentlichen verändert. In Experimenten konnte Cai-Nicolas Ziegler [TDRS05] zeigen, dass der iterative Bewertungsprozess der Appleseed-Metrik schnell terminiert. Zudem wird durch

die Verwendung von direkten Bewertungen die Menge an auszuwertenden Informationen beschränkt.

Als ein Problem der Appleseed-Metrik könnte sich deren Parametrisierung erweisen. Spreading Activation Strategien sind nicht weitläufig bekannt, so dass die Bedeutung der einzelnen Parameter nicht intuitiv verständlich ist. Die richtige Parametrisierung für die Metrik wird von einem Benutzer Erfahrung erfordern. Für die drei Parameter injiziertes Potential bzw. Vertrauen $in(x)_0$, den Streuungsfaktor d und die Schranke T_c hält Ziegler Empfehlungen bereit.

Um die Metrik weiter an reale Bedingungen anpassen zu können, wurden weite Parameter geschaffen. So lassen sich schlechte Bewertungen stärker bestrafen, die Menge der insgesamt analysierten Knoten beschränken und die maximale Pfadlänge zu einem Knoten im Graphen bestimmen.

Einordnung in den Kontext des Semantic Web

Nach dem vorgestellten Klassifikationsschema scheint sich die Appleseed-Metrik gut für den Einsatz im Semantic Web zu eignen. Sie erfüllte alle Kriterien die zuvor als wesentlich für das Semantic Web herausgearbeitet wurden. Sie verwendet eine lokale Perspektive und sie kann auf offene verteilte Informationsstrukturen angewandt werden. Da ihr kein kontrollierter Bewertungsprozess zugrunde liegt, müssen die verwerteten Daten zuvor auf ihre Herkunft geprüft werden, da die Identität der Bewerter wenigstens als Pseudonym bekannt sein muss. Ihr rangbasierendes Auswahlverfahren gewährleistet immer Ergebnisse, solange die Quelle des Vertrauens Bewertungen über andere Entitäten veröffentlicht hat. Unter der Verwendung eines Vertrauensnetzwerkes listet die Metrik die vertrauenswürdigsten Entitäten der Reihenfolge nach auf.

3.2 EigenRumor

Der von Ko Fujimura u.a. entworfene EigenRumor Algorithmus [ERRB05] wurde für Bewertungen von hoch relevanten und glaubwürdigen Blog-Artikel in Blog-Communities entwickelt. Mit seiner Hilfe sollten Suchergebnisse aus Blog-Communities sortiert werden. Der Algorithmus basiert ähnlich wie PageRank auf der iterativen Annäherung von Eigenvektoren. EigenRumor wurde entwickelt, um

die nur wenig verlinkten Blog-Artikel effektiver bewerten zu können, als es z.B. mit PageRank der Fall ist. Die Reputation eines Artikels wird dabei durch die Reputation seines Autors und seiner Bewerter bestimmt.

In einer exemplarischen Implementierung wurde das Ranking von EigenRumor mit zwei anderen Mechanismen für das Sortieren von Blog-Artikel verglichen. In einer Stichprobe von 40 Personen bewerteten 45%, dass EigenRumor bessere Ergebnisse liefert. 48% konnten hingegen keine wesentlichen Unterscheide zwischen den Ergebnissen ausmachen. Lediglich 7% der Teilnehmer bewerteten die Ergebnisse der anderen beiden Mechanismen als besser.

Eigenschaften des Metrikalgorithmus

Für die Berechnung der Reputation von Artikeln werden zwei bipartite gerichtete Graphen benötigt. Die Knoten der beider Graphen werden nach Agenten und Objekten partitioniert. Die gerichteten Kanten der Graphen verlaufen von den Agenten zu den Objekten. Der erste Graph stellte die Agenten und ihre veröffentlichten Objekte in Beziehung. Der zweite Graph zeigt an, welche Agenten welche Objekte positiv bewertet haben.

Basierend auf den beiden Graphen werden die Matrizen *P* für die Veröffentlichungen und *E* für die Bewertungen aufgestellt. Sie projizieren Agenten auf die entsprechenden Objekte. Die Reputation der Autoren, der Bewerter und zuletzt der Blog-Artikel werden in Vektoren $\vec{a}$, $\vec{h}$ und $\vec{r}$ dargestellt. Die Reputation der Artikel $\vec{r}$ lässt sich nun anteilig aus der Reputation der Autoren $\vec{a}$ und Bewerter $\vec{h}$ mithilfe der Matrizen P^T und E^T berechnen. Da $\vec{a}$ und $\vec{h}$ selber wieder mit *P* und *E* aus $\vec{r}$ berechnet werden, lässt sich eine Matrix *S* (siehe Abbildung 3) konstruieren, deren Eigenvektor $\vec{r}$ ist.

$$S=(\alpha P^T P+(1-\alpha)E^T E)$$

Abbildung 3: Matrix für die Eigenvektorberechnung in EigenRumor [ERRB05]

Durch eine Annäherung von $\vec{r}$ durch die iterative Anwendung von *S* auf $\vec{r}$ werden die Objekte bewertet. Das Ergebnis ist ein Ranking der Objekte nach den Bewertungen. Nebenbei entsteht zudem eine Bewertung der Bewerter und

Autoren von Objekten. Durch die Matrizenoperationen werden alle Objekte parallel bewertet. Der errechnete Rank eines Objektes hat globale Gültigkeit, da der Algorithmus keinerlei subjektive Einflüsse berücksichtigt.

Der Algorithmus zeigt, wie sich zwei verschiedene Bewertungsaspekte für die Bewertung berücksichtigen lassen. Das vorgestellte Modell lässt sich sicher für weitere Aspekte erweitern, die dann gewichtet in die Bewertung einfließen. So könnten durchaus verschiedene Aspekte von Informationsqualität über entsprechende Bewertungsmatrizen berücksichtigt werden. Zusätzlich lassen sich die Bewerter der einzelnen Aspekte bewerten.

Datenbasis

Wie bereits beschrieben werden zwei bipartite gerichtete Graphen für die Evaluierung der Reputation von Objekte benötigt. Während der eine direkte Bewertungen der Objekte enthält, reflektiert der zweite die Veröffentlichungen eines Autors. Es werden also sowohl direkte Bewertungen der Objekte als auch Transaktionsdaten der Veröffentlichung für die Bewertung herangezogen.

In der Interpretation von Blog-Communities sind die Objekte die Blog-Artikel und die Agenten die Community-Mitglieder. Um die beiden Graphen zu ermitteln, wurden die Hyperlinkstrukturen analysiert. Die Veröffentlichungen der einzelnen Blogger wurden durch Hyperlinks zwischen den Hauptseiten eines Blogs und seinen Blog-Artikeln gemessen. Als Bewertungen wurden Hyperlinks zwischen den Blog-Artikeln interpretiert.

Da die Identitäten der einzelnen Blogger bekannt sein müssen, dürfen nur autorisierte Nutzer Blog-Artikel in ihrem Blog veröffentlichen. Der Bewertungsprozess wird aber nur teilweise kontrolliert. Das Blog-System steuert in der Regel die Anordnung der Blog-Artikel zur Hauptseite des Blogs, so dass dem Nutzer kein Spielraum bei der Bekanntmachung seiner Veröffentlichungen bleibt, die die Veröffentlichungsmatrix P^T bestimmen. Die Bewertungen in Form von Hyperlinks zwischen den Blog-Artikeln werden allerdings von den Bloggern selber beim Schreiben ihrer Artikel erstellt und unterliegen somit nur der Autorisierung durch das Blog-System.

Die Informationen sind in Blog-Systemen öffentlich verfügbar, so dass jeder die Hyperlinkstrukturen einer Blog-Community selber evaluieren kann. Obwohl Blog-Systeme oftmals eine zentrale Datenhaltung verwenden, kann der Algorithmus auch auf verteilten Informationen arbeiten, solange die Identitäten der Autoren und der Bewerter prüfbar sind.

Weitere Eigenschaften

Der EigenRumor Algorithmus bietet für Blog-Communities im Vergleich zu PageRank eine deutlich bessere Bewertungsfunktionalität. Da nur sehr wenige Blog-Artikel Hyperlinks besitzen, würde PageRank sehr viele Blog-Artikel nicht bewerten, weil Seiten, die keine ausgehenden Hyperlinks besitzen, aus einem Graphe entfernt werden müssen, damit der PageRank Algorithmus angewandt werden kann.

Die Manipulierbarkeit von einem reputationsbasierenden System auf der Basis von EigenRumor hängt entscheidend von der Prüfung der Identität der Autoren ab. Die von Friedman und Resnik propagierten „Cheap Pseudonyms“ [SCCP01] stellen für ein solches System ein großes Problem dar. Mit ihnen können Autoren ihre eigenen Artikel durch intensive Querreferenzierung durch Artikel der verschiedenen Pseudonyme im Ranking nach oben katapultieren. Die Verwendung von direkten Bewertungen, die für den Algorithmus nicht nachvollziehbar sind, fördert ebenfalls die Manipulierbarkeit. Positiv wirkt sich dagegen die transaktionsbasierende Analyse der Veröffentlichungsinformationen aus.

Die Qualität der Informationsbasis sollte relativ gut gesichert sein. Die Trennung von Veröffentlichungs- und Bewertungsinformationen, sowie die Autorisierung von Benutzern wird durch die Blog-Systeme gewährleistet. Da die zugrunde liegende Datenstruktur wie bei PageRank aus einfachen Hyperlinks besteht, werden keine besonderen Ansprüche hinsichtlich der Einhaltung von Bewertungschemata an die Blogger gestellt.

Wie PageRank eignet sich EigenRumor als Alternative, wenn subjektive Metriken auf der Basis von sozialen Netzwerken keine Ergebnisse liefern können. Da die vorhandenen Strukturen von Blog-Systemen genutzt werden, um die

Bewertung von Artikeln vorzunehmen, entsteht kein Boostrap-Problem und kein weiterer Aufwand für den Blogger, um das System zu pflegen.

Einordnung in den Kontext des Semantic Web

Für den Einsatz im Semantic Web nach den aufgestellten Kriterien scheint EigenRumor wenig geeignet zu sein. Für die Anwendung des Algorithmus in einer verteilten Informationsarchitektur fehlen die Mechanismen, um die Identitäten der Autoren zu prüfen. Mit dem Semantic Web Publishing Vokabular [NGPT05] wurde zwar eine Möglichkeit vorgestellt, um die Herkunft von Veröffentlichungen im Semantic Web überprüfen zu können, doch wird von diesen Möglichkeiten bisher kein Gebrauch gemacht.

Abgesehen von dem Problem mit der Bestimmung von Identitäten ist der EigenRumor-Algorithmus eine globale Metrik, die zumindest als Fallbacklösung einsetzbar ist, wenn lokale Metriken keine Ergebnisse liefern. Offene und verteilte Informationsstrukturen bereiten der Metrik hingegen keine Schwierigkeiten.

3.3 EigenTrust

Der von Sepandar D. Kamvar u.a. vorgestellte EigenTrust-Algorithmus [EARM03] wurde speziell für P2P-Systeme entworfen. Es sollen Peers identifiziert werden, die schlechte Services leisten, um diese vom Netzwerk ausschließen zu können. Auf diese Weise kann die Anzahl an Services mit einer schlechten Qualität reduziert werden, so dass die Gesamtqualität des Serviceangebots in einem P2P-Netzwerk steigt.

Um diese Problemstellung gut und effizient lösen zu können, verwendet der EigenTrust-Algorithmus wie PageRank [PRCR98] die Berechnung eines Eigenvektors. Auf der Basis einer Bewertungsmatrix, die Bewertungen von Peers über Peers enthält, wird der Eigenvektor, verteilt über das gesamte P2P-System, berechnet. Unter der Verwendung einiger Verfeinerungen des Algorithmus konnte gezeigt werden, dass EigenTrust mit sehr hoher Effektivität Peers mit schlechten Services identifiziert.

Eigenschaften des Metrikalgorithmus

Der EigenTrust-Algorithmus berechnet aus einer Bewertungsmatrix C den

Eigenvektoren $\vec{t}$. Er enthält die global gültigen Bewertungen der Entitäten eines Netzwerkes. Die einzelnen Bewertungen c_{ij} in der Matrix werden aus vielen einzelnen Bewertungen berechnet, die eine Entität *i* über die Entität *j* gesammelt hat. Sie können positiv oder negativ sein. Aus der Differenz der positiven und negativen Einzelbewertungen werden die Bewertungen c_{ij} berechnet. Um die Bewertungen c_{ij} der Entitäten vergleichen zu können, werden sie mit der in Abbildung 4 dargestellten Formel normalisiert, so dass die Summe aller Bewertungen c_{ij} einer Entität sich zu 1 addieren. Die Normalisierung hat den Effekt, das Entitäten mit denen noch nie interagiert wurde, von denen, die schlechte Feedbacks erhalten haben, nicht mehr unterscheidbar sind, da negative Bewertungen s_{ij} abgeschnitten und auf 0 gesetzt werden.

$$c_{ij} = \frac{max(s_{ij}, 0)}{\sum_j max(s_{ij}, 0)}$$

Abbildung 4: Normalisierung der Bewertungen in EigenTrust [EARM03]

Durch die iterative Anwendung von *C* auf einen Startvektor $\vec{t}^{(0)}$, kann ein Eigenvektor $\vec{t}$ für *C* ermittelt werden, der die global gültigen Bewertungen der Entitäten im Netzwerk enthält. Durch die Beschaffenheit von *C* und die Wahl eines Startvektors $\vec{t}^{(0)}$, dessen Werte $t_i^{(0)}$ sich zu 1 addieren, bleiben die Summe der Werte von $\vec{t}^{(k)}$ nach *k* Anwendungen von *C* auf $\vec{t}^{(0)}$ eins.

Der verwendete Startvektor $\vec{p}$ wird aus den Bewertungen einiger hinsichtlich des Bewertungskriteriums von vornherein vertrauenswürdiger Entitäten *P* erzeugt. Dabei werden alle als gleich vertrauenswürdig bewertet, in dem alle p_i mit $i \epsilon P$ mit $1/|P|$ bewertet werden, während alle anderen Entitäten die Bewertung 0 erhalten. Durch die Verwendung des Startvektors $\vec{p}$ wird die Termination des iterativen Annäherungsprozesses garantiert. Zusätzlich lassen sich mit $\vec{p}$ zwei weitere Probleme lösen. Wenn eine Entität *i* selber keine Bewertungen über andere Entitäten besitzt, weil sie bisher mit niemanden interagierte, kann $\vec{p}$ als Standardbewertung den eigenen Bewertungsvektor ersetzen. Auf der anderen Seite lässt sich mit $\vec{p}$ der Einfluss von verabredeten böswilligen Entitäten

reduzieren, in dem man in jeder Iteration dem Vektor $\vec{p}$ einen gewissen Anteil *a* am globalen Bewertungsvektor $\vec{t}^{(k)}$ lässt. Dieses Model entspricht dem in PageRank [PRCR98] verwendeten Konzept des Random-Surfers. Übertragen auf diesen Algorithmus bedeutet dies, dass eine Entität nach einer Reihe von Inanspruchnahmen schlechter Services frustriert sein wird und mit der Wahrscheinlichkeit *a* die Möglichkeit hat, sich wieder einer apriori vertrauenswürdigen Entität zuzuwenden. Abbildung 5 stellt die resultierende Formel für die Bestimmung des Eigenvektors $\vec{t}$ im EigenTrust-Algorithmus dar.

$$\vec{t}^{(k+1)} = (1-a)C^T \vec{t}^{(k)} + a\vec{p}$$

Abbildung 5: Basis des EigenTrust-Algorithmus[EARM03]

Die Bedeutung der einzelnen Werte t_i der Entität im Ergebnisvektor haben alleine keine Bedeutung hinsichtlich eines Bewertungskriteriums. Erst in Relation mit den Bewertungen anderer Entitäten können Aussagen gemacht werden. Die Auswahl gut bewerteter Entität wird daher mit einem rangbasierenden Verfahren getroffen.

Datenbasis

Der EigenTrust-Algorithmus wird von Sepandar D. Kamvar u.a. für die Bewertung der Servicequalität von Peers in einem P2P-Netzwerk benutzt. Die Peers entsprechen hier also den Entitäten, die als Softwareagenten aufgefasst werden können.

Die Basisinformationen für die Auswertung des EigenTrust-Algorithmus sind Feedbacks auf Transaktionen, die jeder Peer gesammelt und gespeichert hat. Die Einzelbewertungen werden durch Feedbackinformationen aus den einzelnen Transaktionen zwischen Peers realisiert, die den Wert 1 für zufriedenstellende und den Wert -1 für nicht zufriedenstellende Transaktionen annehmen können. Aus diesem einfachen Bewertungsschema berechnet jeder Peer aus seinen persönlichen Erfahrungen die Bewertung der anderen Peers mittels des normalisierten Überschusses an zufriedenstellenden Transaktionen (siehe Abbildung 4).

Da aus Gründen geringerer Manipulierbarkeit die Peers nicht ihre eigenen

Servicebewertungen berechnen, sondern die Berechnung von mehreren anderen Peers durchgeführt wird, müssen die Peers ihre persönlichen Bewertungen c_{ij} auf der Basis ihrer Feedbacks veröffentlichen, damit die globale Bewertungsmatrix C aufgebaut werden kann.

Die Identität der einzelnen Peers muss dem EigenTrust-Algorithmus nur als Pseudonym bekannt sein. Das Problem von „Cheap Pseudonyms“ nach Resnik [SCCP01] wird vom EigenTrust-Algorithmus nicht direkt adressiert. Durch das leichte Wechseln der Pseudonyme können böswillige Entitäten in P2P-Systemen mehrere Peers betreiben und mit diesen die eigenen Peers positiv und alle anderen negativ bewerten. Sepandar D. Kamvar u.a. gehen davon aus, dass ein reputationsbasierendes P2P-System auf der Basis von EigenTrust dennoch effektiv böswillige Peers identifiziert. Sie belegen dies mit ihren experimentellen Ergebnissen, die auch dann gute Ergebnisse bescheinigen, wenn große Teile eines P2P-Systems aus böswilligen Peers bestehen.

Weitere Eigenschaften

Durch die Verwendung von Peers, die von vornherein vertrauenswürdig sind, wird jedes Bootstrap-Problem umgangen. Ein Peer ohne jede persönliche Erfahrung kann sich sofort an der Berechnung der globalen Bewertungen beteiligen und von den Ergebnissen profitieren. Als geeignete Peers denen apriori vertraut wird, werden die Peers der Initiatoren eines Netzwerkes vorgeschlagen, die ein begründetes Interesse am Erfolg eines Netzwerkes haben und daher als vertrauenswürdig hinsichtlich einer positiven Servicequalität gelten können.

Durch die Synthese der Feedbacks zu lokalen Servicebewertungen durch jeden Peer wird die Datenmenge, die zwischen den Peers zur verteilten Berechnung ausgetauscht werden muss, auf die syntetischen Bewertungen c_{ij} reduziert. Der Faktor für die durchschnittliche Anzahl der zu übertragenden Feedbacks zwischen zwei Peers fällt weg und verbessert damit die Skalierbarkeit hinsichtlich der benötigten Bandbreite des Netzwerkes. Zudem konnten Sepandar D. Kamvar u.a. zeigen, dass die Näherung des Eigenvektors schon nach wenigen Iterationen konvergiert.

Die Qualitätssicherung der Bewertungsinformationen liegt bei den einzelnen Peers. Sie verwalten ihre Feedbacks und generieren die syntetischen Servicebewertungen. In einem P2P-System werden die nötigen Mechanismen in der Regel durch wenige Standardimplementierungen realisiert, die die Bewertungen kontrollieren. Dennoch kann niemand kontrollieren, ob das Bewertungsschema tatsächlich im einzelnen Peer implementiert wurde.

Gegen Manipulation des P2P-Systems durch falsche syntetische Bewertungen scheint der EigenTrust-Algorithmus gut gewappnet zu sein. In Experimenten wurde gezeigt, dass der EigenTrust-Algorithmus selbst bei einem großen Anteil an böswilligen Peers, die gezielt falsche Informationen verbreiteten, gute Ergebnisse liefert. Nach Raphael Levien's „bottleneck property“ sollte der EigenTrust als Gruppen-Trust-Metrik ebenfalls eine hohe Resistenz aufweisen. Das Problem der „Cheap Pseudonyms“ wird als inhärent Vorhanden akzeptiert, und muss durch den Bewertungsmechanismus des Algorithmus selbst gelöst werden.

Einordnung in den Kontext des Semantic Web

Ähnlich wie andere reputationsbasierende Metriken für P2P-Systeme passt der EigenTrust-Algorithmus nur wenig auf die getroffenen Annahmen für den Einsatz im Semantic Web. Zum Einen entsprechen die globalen Bewertungen nur wenig den persönlichen Anforderungen eines Nutzers. Auch wenn in Varianten des Algorithmus lokale Perspektiven stärkere Berücksichtigung finden, so behält der Kern der Metrik doch eine globale Perspektive. Auf der Anderen Seite ist es schwer vorstellbar, das eine standardisierte Implementierung eines solchen Peers für jede Entität im Semantic Web installiert ist, um die verteilte Berechnung des globalen Bewertungsvektors $\vec{t}$ zu berechnen. Verteilte Systeme erscheinen daher vor dem Hintergrund des Semantic Web eher ungeeignet.

3.4 Evidence-Based Model

Bin Yu und Munindar P. Singh definieren in [EMDR02] ein Modell für die Bewertung der Servicequalität. Die Entwicklung des Modells wird durch den fortschreitenden Einsatz von Agenten in verteilten Umgebungen motiviert, die selbstständig über ihre Kooperationspartner entscheiden müssen. Auf der Basis

von Informationen aus vergangenen Transaktionen werden mithilfe der Dempster-Shafer-Theory [MTE76] Entscheidungen über die voraussichtliche Qualität eines Services eines möglichen Kooperationspartners gefällt.

Das Modell sieht eine zweistufige Informationsbeschaffung vor. In der ersten Stufe verwendet ein Agent *A* seine eigenen Transaktionsinformationen aus vergangenen Kooperationen mit einem Agenten *B*, um über die Vertrauenswürdigkeit von *B* hinsichtlich eines guten Services zu entscheiden. Wenn *A* nur ungenügende Informationen aus Transaktionen mit *B* gesammelt hat, kommt eine zweite Stufe zum Tragen. *A* befragt in diesem Fall vertrauenswürdige Agenten über ihre Meinung über die Servicequalität von *B* und fügt deren Bewertungen zusammen.

Eine Besonderheit dieses Models ist der explizite Umgang mit Misstrauen, welches nur wenige Metriken berücksichtigen. Im Evidence-Based Model gehört Misstrauen zum elementaren Bestandteil der Vertrauensevaluierung.

Eigenschaften des Metrikalgorithmus

Die Dempster-Shafer-Theory verwendet Wahrscheinlichkeiten über dem Eintreten von Ereignissen. Sind keine Informationen für die Bestimmung von Wahrscheinlichkeiten bekannt, so werden sie mit 0% geschätzt. Gibt es Informationen, so werden die Wahrscheinlichkeiten neu bewertet. In dem Model von Yu und Singh werden nur zwei Ereignisse betrachtet, ein positives und ein negatives. In der Interpretation von Yu und Singh bedeutete das positive Ereignis $m(\{T\})$, dass einer Entität vertraut wird. Das negative Ereignis $m(\{\neg T\})$ ist entsprechend die Entscheidung, dass eine Entität nicht vertrauenswürdig ist.

Um die Wahrscheinlichkeiten der Ereignisse zu bestimmen, wird eine untere und obere Schranke (ω und Ω) definiert, die sich auf der Bewertungsskala von $[0..1]$ befinden. Der Anteil an den Bewertungen einer Entität, die eine Bewertung bis einschließlich der unteren Schranke ω besitzen, wird als die Wahrscheinlichkeit des Eintreten des negativen Ereignisses gewertet. Die Wahrscheinlichkeit für ein positives Ereignis ist der Anteil der Bewertungen mit einem Wert von mindestens Ω. Der Anteil der Bewertungen zwischen den Schranken

$m(\{T, \neg T\})$ steht für die Ungewissheit. Ob eine Entität positiv bewertet wird, wird mittels einer Schranke entschieden, die fest legt, wie groß die Differenz von $m(\{T\}) - m(\{\neg T\})$ sein muss, damit jener Entität vertraut wird.

Wenn eine Entität *A* selber nicht genügend Transaktionsinformationen über eine Entität *B* besitzt, so fragt *A* Entitäten, denen sie vertraut, nach ihrer Meinung über *B*. Sie können *A* entweder ihre Meinung als Wahrscheinlichkeiten über den Ereignissen geben oder falls sie selber nicht genug Informationen besitzen, ihre eigenen Vertrauten zurückgeben. Vertraute sind Entitäten die von der Quelle des Vertrauens hinsichtlich des Bewertungskriteriums positiv bewertet wurden. Auf diese Weise wird das lokale WoT einer Entität sukzessiv nach Meinungen von Entitäten durchsucht, die bereits mit *B* zusammengearbeitet haben. Die Meinungen der einzelnen Entitäten werden durch *A* zu einer Gesamtmeinung zusammengesetzt[6]. Die Bewertung der Entität *B* wird somit aus einer Gruppe von Entitäten entwickelt, die bereits Transaktionen mit *B* hatten. Die Evaluierung der Servicequalität wird im ersten Schritt skalar über die eigenen Bewertungen und im zweiten alternativen Schritt in der Gruppe entschieden.

Datenbasis

Das von Yu und Singh vorgestellte Model wurde für verteilte Agentensysteme entworfen. Die Entitäten im System sind die Agenten, die die Servicequalität der anderen Agenten bewerten wollen.

Die verwendete Informationsstruktur ist verteilt, aber nur teilweise offen. Die Feedbackinformationen der einzelnen Agenten werden in ihrem lokalen Speicher geschlossen verwahrt und nur die auf deren Basis inferierten Meinungen werden auf Anfrage weitergegeben.

Die Datenbasis verwendet zwei verschiedene Bewertungsschemata. Das erste organisiert die Bewertungen von Transaktionen mit anderen Agenten im lokalen

6 Yu und Singh referenzieren Meinungen als lokale und totale Meinungen. Dies darf nicht mit lokaler und globaler Perspektive verwechselt werden. Die lokalen Meinungen betreffen nur die Informationen die ein Agent selber über andere hat. Die totale Meinung referenziert die inferierte Gesamtmeinung aus dem persönlichen WoT. Ihre Perspektive ist lokal, da nur Agenten und ihre Vertrauten als vertrauenswürdig gelten, die ein Agent aus seiner eigenen Erfahrung als vertrauenswürdig betrachtet. Nur sie sind Teil des persönlichen WoT.

Speicher eines jeden Agenten. Die Servicequalität für die Transaktion mit anderen Agenten wird auf einer Skala von $[0..1]$ in Schritten von $0{,}1$ bewertet. Das zweite Schema übermittelt die Meinungen der Agenten als ein Tupel aus drei Wahrscheinlichkeiten $(m(\{T\}), m(\{T, \neg T\}), m(\{\neg T\}))$ für die Entscheidungsfindung nach der Dempster-Shafer-Theorie. Sie entsprechen direkten Bewertungen, da die Grundlagen der geäußerten Meinung für andere Agenten nicht nachvollziehbar sind. Ein Agent kann nicht kontrollieren, ob es sich bei der geäußerten Meinung um eine Zusammenfassung der Transaktionsinformationen nach dem Model handelt.

Im zweiten Schritt des Algorithmus wird über das persönliche WoT eines Agenten nach Meinungen von anderen Agenten gesucht. Durch die Auswertung der Transaktionsinformationen im lokalen Speicher eines jeden Agenten und den daraus resultierenden Ergebnissen wird ein soziales Netzwerk aufgebaut. Im Gegensatz zu anderen Metriken, die soziale Netzwerke analysieren, wird das Netzwerk nicht direkt zum Bewerten von Agenten verwendet, sondern nur zur Suche von Agenten, die bereits selbst mit dem Ziel des Vertrauens Kontakt gehabt haben.

Um die Bewertungen der Servicequalität eines Agenten für die Bewertung seines zukünftigen Verhaltens verwenden zu können, muss er wiedererkennbar sein. Daher muss die Identität eines Agenten wenigstens mit einem Pseudonym bekannt sein.

Obwohl über den Bewertungsprozess selbst keine Aussage gemacht wird, lässt der angestrebte Einsatz in Softwareagenten eher auf eine willkürliche Bewertungsabgabe schließen. Da Agenten autonom und somit ohne die Kontrolle von einer zentralen Instanz arbeiten, können sie selbständig über den Umgang mit Bewertungen entscheiden. Die Einhaltung des vorgestellten Models, insbesondere der lokalen Berechnung der Servicequalität aus den Feedbacks, ist daher eher eine Selbstverpflichtung.

Weitere Eigenschaften

Obwohl die eigentliche Bewertung von Agenten nicht auf der Basis des

sozialen Netzwerkes vorgenommen wird, leidet das System dennoch unter Boostrap-Problemen. Wenn ein Agent noch keine Transaktionen mit anderen Agenten hatte, so besitzt er weder eigene Transaktionsinformationen über das Ziel des Vertrauens noch Bewertungen über andere Agenten. Die Servicequalität kann so nicht evaluiert werden. Das Modell kann zwar in einem solchen Fall immer noch Ergebnisse liefern, aber nur unter der Bedingung, das der Schwellenwert für vertrauenswürdige Agenten auf 0 reduziert wird (, da $m(\{T\})=m(\{\neg T\})=0$ geschätzt wird). Dies bedeutet allerdings, dass jedem Agenten vertraut wird und macht somit den Einsatz des Modells unnötig.

Die von Qing Zhang u.a. in [CSTF04] bemerkte höhere Privatsphäre von meinungsbasierenden Bewertungen, wird am Evidence-Based Model sehr deutlich. Veröffentlichen müssen die Agenten lediglich die Meinungen, die auf der Basis ihrer gesammelten Feedbackinformationen errechnet wurden. Auf diese Weise bleiben die Feedbackinformationen privat.

Da Meinungen aus den Feedbacks dezentral von jedem Agenten selbst berechnet werden, hängt deren Berechnung nicht von der Anzahl der Agenten ab. Lediglich die Suche nach Agenten, die mit dem Ziel des Vertrauens bereits Kontakt hatten, hängt von der Menge der Agenten ab. Da die Suche über ein soziales Netzwerk erfolgt, kann von der guten Skalierbarkeit von Algorithmen, die auf der Basis von sozialen Netzwerken Pfade suchen, Gebrauch gemacht werden.

Für die Einhaltung der Bewertungschemata sind die Agenten selbst zuständig. Durch den Druck gute Services zu leisten, werden sie allerdings dazu angehalten, das Bewertungsschema für den Meinungsaustausch zwischen den Agenten einzuhalten, da der Austausch von Bewertungen selbst als Service verstanden werden kann. Durch das soziale Netzwerk werden zudem schlechte Bewerter identifiziert und nur Meinungen von vertrauenswürdigen Bewertern eingeholt.

Der entscheidende Schwachpunkt hinsichtlich der Manipulierbarkeit liegt im Austausch von syntetischen Bewertungen, die als Meinungen interpretiert werden. Wenn die zugrunde liegenden Feedbacksinformationen verschlossen bleiben, kann nicht überprüft werden, ob ein Agent sich an die Berechnungsmethode für die

Wahrscheinlichkeiten der Ereignisse gehalten hat. Durch die Veröffentlichung der Feedbacksinformationen können die Berechnungen wiederum von der Quelle des Vertrauens selbst berechnet oder zumindest überprüft werden. Die Verwendung von persönlichen Feedbacksinformationen gewährt die größte Sicherheit gegen Manipulationen. Deren Veröffentlichung würde allerdings die Privatsphäre empfindlich beeinflussen. Nach Raphael Levien [ARTM04] bietet aber auch die alternative Berechnung mit einer Evaluierung von Informationen aus einer Gruppe eine hohe Sicherheit von Manipulationen.

Einordnung in den Kontext des Semantic Web

Das Evidence-Based Model von Bin Yu und Munindar P. Singh eignet sich nach der vorgestellten Klassifikation nur bedingt für den Einsatz im Semantic Web. Als Problem stellt sich vor allem die verteilte Berechnung von Meinungen dar. Ohne diese Eigenschaft, dass jeder Agent seine Meinungen aus seinen vergangenen Transaktionen berechnet, sondern nur eine Meinung nach dem gegebenen Schema für die Wahrscheinlichkeiten der Ereignisse äußert, könnte dieses Modell gut im Semantic Web eingesetzt werden. Es kann auf verteilten offenen Datenbasen arbeiten und und lässt sich zentral berechnen. Ein reputationsbasierendes System könnte somit ein zweistufiges Verfahren implementieren, welches die persönlichen Erfahrungen vorzieht und nur bei Bedarf auf Informationen von Vertrauten zurückgreift.

3.5 PageRank

Der von Lawrence Page u.a. veröffentlichte PageRank-Algorithmus [PRCR98] wurde vor allem durch die ursprünglich für das Ranking von Webseiten bei Google [GOGL] entwickelte Variante bekannt. Die Internetseiten des WWW, die zu einer Suchanfrage im Google-Index gefunden werden, werden mittels des Ergebnisses des Algorithmus in eine Rangfolge gebracht. An oberster Stelle sollten möglichst die Internetseiten angezeigt werden, die mit hoher Wahrscheinlichkeit relevant für den Suchmaschinenbenutzer sind.

Die ursprüngliche Idee für die Entwicklung von PageRank war es, Internetseiten einen hohen Rang und somit eine hohe Relevanz zu geben, auf die viele

andere Internetseiten mit Hyperlinks verweisen. Zusätzlich sollte ein Hyperlink ein Gewicht bekommen, welches sich aus dem Rang der referenzierenden Seite ergibt. Somit bekommen Internetseiten ebenso einen hohen Rang, wenn sie zwar nur von wenigen aber hoch relevanten Internetseiten referenziert werden.

Eigenschaften des Metrikalgorithmus

Die mathematische Grundlage des PageRank-Algorithmus ist die Lineare Algebra und ihre Matrizen-Transformationen. Auf der Basis eines Graphen $G=(V,E)$ mit gerichteten Kanten $(x,y)\epsilon E$ und $x,y\epsilon V$ wird eine Quadratmatrix *A* aufgestellt. $A_{x,y}$ ist *1* wenn der Knoten *x* eine gerichtete Kante zu *y* besitzt oder 0 andernfalls. Unter dieser Voraussetzung lässt sich ein Ranking mit der Bestimmung des Eigenvektors $\vec{r}$ und des Eigenwerts *c* lösen, so dass gilt $\vec{r}=cA\vec{r}$. Die Bestimmung von $\vec{r}$ durch den Algorithmus wird durch die wiederholte Anwendung von *A* auf einen nicht degenerierten Anfangsvektor geleistet [PRCR98].

Als Datengrundlage wird der gesamte Graph G verwendet, der keine Knoten enthält, die eine eingehende Kante besitzen aber keine ausgehende. Der Graph darf also keine Blätter wie in einer Baumstruktur besitzen. Der Algorithmus berücksichtigt keinerlei lokale Perspektive und inferiert daher für jede Entität das gleiche Ergebnis. Auch nach der Klassifikation von Cai-Nicolas Ziegler [TDRS05] ist der PageRank-Algorithmus eine globale Gruppen-Trust-Metrik.

Die gruppenweise Berechnung wird durch die Matrixoperatoren realisiert. Mit der Ausführung einer Transformation, werden die PageRank-Werte für alle Knoten gleichzeitig einen weiteren Schritt angenähert.

Datenbasis

In der ursprünglichen Anwendung des PageRank-Algorithmus werden Seiten im World Wide Web hinsichtlich ihrer Relevanz bewertet, um Suchergebnisse aus dem Goolge-Index nach ihrer Relevanz zu sortieren. Das World Wide Web wird für die Anwendung als ein gerichteter Graph, in dem die Seiten die Knoten und die Hyperlinks die gerichteten Kanten sind, betrachtet. Analog dazu besteht das Semantic Web rund um das Resource Description Framework in seiner

Abstraktion ebenfalls aus einem gerichteten Graphen mit dem Unterschied, dass die Kanten mit Kennzeichen versehen sind, die ihre semantische Bedeutung identifizieren. Da sich ein Graph mit Kennzeichen einfach auf einen Graphen ohne Kennzeichen abstrahieren lässt, kann der ursprüngliche PageRank-Algorithmus auch für Knoten in RDF Graphen verwendet werden.

Der ursprüngliche Einsatzzweck für das Ranking von Internetseiten, gibt eine klare Informationsstruktur vor. Die Daten im WWW sind verteilt und offen abgelegt. Für den Algorithmus spielt es allerdings keine Rolle, ob die Informationen zentral oder dezentral gespeichert werden. In der Implementierung von Google werden die Daten durch einen Crawler erst in einen zentralen Datenbestand integriert, bevor der Algorithmus ausgeführt wird. Die Entscheidung zentral oder verteilte Datenhaltung bleibt also eher eine Designentscheidung für das Zielsystem, in dem der Algorithmus eingesetzt werden soll.

Der offene Zugang zu den Informationen spielt eine größere Rolle. Für Trust-Metriken mit einer globalen Perspektive ist es wichtig, das jede Entität die gleichen Daten erhalten kann. Da jede Entität auf der Basis der gleichen Daten gleiche Ergebnisse inferiert, sind die Ergebnisse vergleichbar und somit überprüfbar. Um also eine eigene Auswertung anstellen zu können, muss die Datenbasis öffentlich zugänglich sein. Benötigt man keine Vergleiche, z.B. weil der Datenbestand durch eine Vorauswahl personalisiert wird, sind auch geschlossene Datenhaltungen denkbar.

PageRank verwendet keine direkten Bewertungen oder Feedbacks. Die Datenbasis enthält ausschließlich Hyperlinks zwischen Internetseiten. In der Interpretation des Google-Ranking ist ein Hyperlink ein Rating hinsichtlich der Relevanz einer Internetseite. Da diese Interpretation sicher nicht der ursprünglichen Intension des Autors eines Hyperlinks entspricht, lassen sich Hyperlinks nur als transaktionsbasierende Daten interpretieren, wobei die Transaktion in der Erstellung eines Hyperlinks zum Zweck der Referenzierung einer relevanten Internetseite liegt. Die Interpretation als Rating geschieht also erst bei der Anwendung des Algorithmus.

Die Anwendung des Algorithmus auf das WWW birgt ein Problem, das durch das verwendete Datenmodell hervorgerufen wird. Knoten, die eingehende Kanten aber keine ausgehenden Kanten besitzen, werden für den Algorithmus zur Falle. Kanten, die auf diese Knoten verweisen werden als „Dangling Links“ bezeichnet. Die entsprechenden Knoten akkumulieren Gewichte, da es nicht über Kanten an andere verteilt werden kann. Dieses Gewicht geht dem Model verloren. Um dies zu unterbinden, muss der Graph um Dangling Links bereinigt werden. Das Resultat ist ein geschlossener Graph, der keine Blätter wie in Baumstrukturen besitzt. Die Knoten, auf die die „Dangling Links“ verweisen, werden ohne Bewertung an das Ende des Rankings angehängt, damit sie in den Ergebnissen von Suchanfragen nicht verloren gehen.

Über die Identität der Autoren werden keine Annahmen getroffen, sie bleiben anonym. Die Relevanz einer Internetseite wird ausschließlich über der Struktur des Netzwerkgraphen des WWW inferiert.

Weitere Eigenschaften

Der PageRank-Algorithmus kennt kein Boostrap-Problem wie Metriken, die von einem WoT abhängen. Solange offene Informationen verwendet werden, kann jede Entität sofort den Algorithmus auf ihnen ausführen, um zu den global gültigen Ergebnissen zu kommen. PageRank kann somit eingesetzt werden, wenn lokale Trust-Metriken nicht genug Ergebnisse liefern können [ORS03] und kann so helfen Bootstrap-Probleme zu überwinden.

Da die verwendeten Informationen per se öffentlich sind, gibt es keine Privatsphäre. Die Qualität der veröffentlichten Informationen im WWW wird ebenfalls von niemandem kontrolliert und es gibt keine speziellen Bewertungsschemata, die von Autoren eingehalten werden müssen. Der Algorithmus selbst muss also dafür sorgen, dass die Qualität der Informationen für die Berechnung brauchbarer Ergebnisse gegeben ist.

In [ALSS98] beschreiben Sergey Brin und Lawrence Page alle wesentlichen Teile des Google-Systems als nahezu linear im Aufwand. Selbst unter dieser Voraussetzung bleibt der Gesamtaufwand für die Berechnung eines Rankings für

das WWW enorm, da die Menge der Knoten in Netzwerkgraphen sehr groß ist und stetig steigt. Für kleine Mengen an Daten, deren Größe nahezu konstant ist, kann der PageRank-Algorithmus dennoch gut und schnell Ergebnisse berechnen.

Durch den hohen Aufwand für große Datenmengen kann nicht für jede Bewertung eine neue Berechnung angestoßen werden. Die Berechnungen würden zu viel Zeit in Anspruch nehmen. Um doch zu Ergebnissen zu kommen, müssten die Ergebnisse gecacht werden, um jederzeit auf Anfragen reagieren zu können. Aus diesem Sachverhalt resultiert eine geringe Aktualität der Rankings. Für die Evaluierung von Vertrauen stellt dies ein Problem dar, da unkooperatives Verhalten einer Entität oft zu starken Einbußen des Vertrauens führt, die sich nicht über einen Cache abbilden lassen. Es könnte somit passieren, das einer Entität auf der Basis eines veralteten Rankings fälschlicherweise vertraut wird.

Ein großer Vorteil des PageRank-Algorithmus ist der geringe Aufwand, den der Benutzer selbst betreiben muss. Da kein besonderes Bewertungsschema vorhanden ist, und gegebene Strukturen wie Hyperlinks ausgenutzt werden, hat der Benutzer keinen Aufwand, um Informationen zu verwalten. Zudem lässt sich der Algorithmus sehr leicht parametrisieren. Der einzige Eingabeparameter ist der Netzwerkgraph.

Da der Algorithmus die Identitäten der Autoren von Kante bzw. Hyperlinks vollkommen ausser Acht lässt, ist nur schwer überprüfbar, ob sich verschiedene Seiten eines Autors einfach nur gegenseitig referenzieren. Auf diese Weise kann eine Entität den Rang ihrer Knoten bzw. Internetseiten manipulieren. Vor diese Form der Manipulation kann sich der Algorithmus selbst nicht schützen. Dies wird nur durch den Umstand kompensiert, dass ein hoher Rang nicht nur von der Anzahl der Referenzen abhängt, sondern auch vom Gewicht der referenzierenden Seite. Nach Raphael Levien [ARTM04] wirkt sich positiv aus, dass PageRank eine Gruppen-Trust-Metrik ist, daher nach dem „bottleneck property“ nur schwer manipulierbar ist.

Einordnung in den Kontext des Semantic Web

Der PageRank-Algorithmus entspricht nicht ganz den Kriterien, die anhand

der Klassifikation für das Semantic Web aufgestellt wurden. Sein Einsatzgebiet im WWW entspricht zwar in vielen Punkten dem des Semantic Web, er kann auf offenen verteilten Informationsstrukturen arbeiten, dennoch berücksichtigt der Algorithmus die persönliche Sichtweise von Nutzer nicht, da er global gültige Rankings inferiert. Die einzige Möglichkeit, den PageRank-Algorithmus in einem personalisierten System einzusetzen, besteht in der Personalisierung des Netzwerkgraphen durch eine Vorauswahl von Informationen.

Im Semantic Web könnte der PageRank-Algorithmus allerdings dennoch von hoher Bedeutung sein. Als Alternative zu lokalen Metriken auf der Basis eines WoT könnte PageRank sich gut eignen. Im Semantic Web gibt es keine zentrale vertrauenswürdige Instanz, die Neulinge in ein WoT einbindet. Selbst wenn ein Benutzern gut in ein WoT eingebunden ist, können Situationen auftreten, die einen Blick über den Rand des eigenen WoT hinaus erfordern. Dies geschieht z.B., wenn der Benutzer in einen Kontext wechselt, in dem er sonst nicht agiert. In diesen Fällen leiden die meisten lokalen Trust-Metriken unter einem Bootstrap-Problem und können eventuell keine Bewertungen berechnen. Mit PageRank würde für solche Fälle eine Fallback-Lösung bereitstehen, die mit einer globalen Perspektive erste Ergebnisse bringt, bis ein persönliches WoT aufgebaut ist. [ORS03].

3.6 PeerTrust

Die von Ling Liu und Xiong Li in [RBTM03][TDPP02] vorgestellte PeerTrust-Metrik wurde für den Einsatz in P2P-Systemen entwickelt. Sie besteht aus einem mathematischen Heuristik für die Bewertung von Vertrauen und wurde beispielhaft für die P-Grid [PGRI01] Architektur implementiert.

Die Metrik inferiert über Transaktions- und Feedbackinformationen. Verschiedene Aspekte von feedbackbasierenden Systemen werden in die Vertrauensbewertung eingebunden. Die wichtigsten Aspekte sind die Anzahl der Transaktionen, die Feedbacks auf Transaktionen und das Vertrauen in den Feedbackgeber. Weiter optionale Aspekte lassen sich durch die Metrik berücksichtigen. So können Einflüsse aus dem Kontext der einzelnen Transaktionen, z.B.

die Abweichung von der durchschnittlichen Größe des Transaktionsgegenstands, oder aus dem Kontext der Community, in die Bewertung eingehen.

Eigenschaften des Metrikalgorithmus

Das vollständige Modell berechnet das Vertrauen eines Peers *u* auf der Basis von Transaktions- und Feedbackinformationen. Die drei wichtigsten Bestandteile des Modells sind die Zufriedenheit des Transaktionspartners in der i-ten Transaktion S(u,i), das Vertrauen in den Partner glaubwürdige Informationen zu verbreiten *Cr(p(u,i))*, wobei *p(u,i)* den Partner in der *i*-ten Transaktion identifiziert, und die Gesamtzahl der geleisteten Transaktionen *I(u)*. Zudem enthält das Modell Faktoren für den Kontext der Transaktion *TF(u,i)* und der Community *CF (u)*. Balanciert werden die transaktions- und communityabhängigen Teile des Modells durch die Variablen α und β. Das mathematische Modell ist in Abbildung 6 vollständig dargestellt.

$$T(u) = \alpha \cdot \frac{\sum_{i=1}^{I(u)} S(u,i) \cdot Cr(p(u,i)) \cdot TF(u,i)}{I(u)} + \beta \cdot CF(u)$$

Abbildung 6: Formel des mathematischen Modells der PeerTrust-Metrik [RBTM03]

Wie die einzelnen Komponenten des Modells implementiert werden, wird dem Design der reputationsbasierenden Systemen überlassen. Da die Bedeutung der Komponenten Bestandteil des Modells sind, kann an dieser Stelle nicht stärker von der Interpretation als Vertrauensbewertung abgesehen werden. Über die Gewichte der Transaktionskontexte kann gesteuert werden, welche Transaktionen aus welchem Kontext stärker für die Vertrauensbewertung berücksichtigt werden.

Die Bewertung wird auf der Grundlagen aller vergangener Transaktionen berechnet, die dem System bekannt sind. Wer das Vertrauen einer Entität berechnet haben möchte, spielt für die Evaluation keine Rolle, da keine lokale Perspektive berücksichtigt wird. Das Ergebnis hängt von den Umständen aller ehemaligen Transaktionen ab. Somit ist PeerTrust eine globale Gruppen-Trust-Metrik.

Das Resultat ergibt sich mittels eines Schwellenwertes für die minimale Bewertung, die eine vertrauenswürdige Entität besitzen muss. Um sicherzustellen, dass ausreichend Informationen für aussagekräftige Inferenzen vorhanden sind, kann mittels eines zweiten Schwellenwertes, die minimale Anzahl an vergangenen Transaktionen bestimmt werden.

Datenbasis

Die Grundlage einer Vertrauensevaluierung sind feedbackbasierende Informationen, aber auch transaktionsbasierende Informationen, da die Anzahl an Transaktionen und die Identität des Transaktionspartners in die Kalkulation der Metrik eingehen. Über die Sichtbarkeit und Lokalität sowie den Bewertungsprozess macht der Metrik-Algorithmus keine Annahmen. Dies hängt sehr stark von der Implementierung eines reputationsbasierenden Systems ab. Die Beispielimplementierung der Metrik für das P-Grid File-Sharing-System soll daher die weitere Analyse bestimmen.

Im P-Grid-System werden die Transaktions- und Feedbackinformationen dezentral gespeichert. Dabei speichert jeder Peer die Informationen über einen anderen Peer und wertet deren Informationen auf Anfrage aus. Da jeder Peer die Information über die Vertrauenswürdigkeit eines anderen Peers erhalten kann, sind die Informationen öffentlich verfügbar. Der Bewertungsprozess wird durch einen Manager kontrolliert, der die gesamten Transaktions- und Feedbackinformationen verwaltet, Feedbacks sendet und empfängt sowie die Bewertungen vornimmt. Die Berechnung der einzelnen Bewertungen wird zwar jeweils auf einem Rechner ausgeführt, da aber die Quelle des Vertrauens die Bewertung nicht selber berechnet, sondern dies von anderen Peers übernommen wird, verwendet die P-Grid-Implementierung eine verteilte Struktur für die Metrikausführung.

Weitere Eigenschaften

Das P-Grid-System mit der Implementierung der PeerTrust-Metrik zeichnet sich durch eine geringe Privatsphäre aus. Eine Entität hat keine Kontrolle darüber, wer im System seine Informationen verwaltet. Die Informationen enthalten alle Details über vergangenes Verhalten. Der Halter der Informationen kann über die

Auswertung der Metrik hinaus, z.B. die favorisierten Transaktionspartner ermitteln.

Durch die Berechnung von Vertrauensbewertungen beim Datenhalter werden die hohen Transportkosten für große Transaktionsdatenmengen zur berechnenden Entität vermieden. Nur die Ergebnisse der Auswertung müssen übermittelt werden. Obwohl die Berechnung der Metrik aus der Sicht der Quelle des Vertrauens dezentral vorgenommen wird, so wird die eigentliche Bewertung eines einzelnen Peers dennoch zentral vom Datenhalter berechnet.

Die Qualität der Feedbacks hinsichtlich der Einhaltung des Bewertungschemas und der Kontrolle der Identitäten der Transaktionsteilnehmer gewährleistet die Implementierung des P-Grid-Systems mit ihrem Trust Manager. Die Qualität der Transaktions- und Feedbackinformationen sollte daher lediglich von den Feedbacks abhängen. Das System behält die volle Kontrolle über die Informationen von der Erzeugung bis hin zu Auswertung.

Die Transaktionsinformationen lassen sich nur durch den Datenhalter manipulieren. Peers, die mit schlechten Feedbacks auf gute Transaktionen reagieren, nehmen anhand ihrer in Folge geringer bewerteten Glaubwürdigkeit nur geringen Einfluss auf die Bewertungen. Zudem müssten durch die gruppenweise Auswertung sehr viele Transaktionsinformationen manipuliert werden. Der einzige, der mit verheerenden Folgen die Ergebnisse manipulieren kann, ist der Datenhalter selbst. Er kann die Daten manipulieren oder einen anderen Bewertungsmechanismus einsetzten.

Einordnung in den Kontext des Semantic Web

Für den Einsatz in reputationsbasierenden Systemen des Semantic Web erscheint eine Implementierung des Modells, wie sie für das P-Grid System vorgenommen wurde, eher ungeeignet. Die Verwendung einer einheitlichen Implementierung, die die Datenerhebung kontrolliert, ist nur sehr schwer durchsetzbar. Zudem erfordern die Implementierungen von globalen transaktionsbasierenden Systemen die Einhaltung eines klaren einheitlichen Bewertungschemas durch alle Entitäten für alle Transaktionen im Netzwerk.

Persönliche Interessen einer Quelle des Vertrauens werden von der PeerTrust-Metrik nicht berücksichtigt. Jeder Peer erhält eine objektiv berechnete Vertrauensbewertung. Auch wenn die Metrik die Eigenschaften der zugrunde liegenden Informationsstrukturen offen lässt, und somit offene verteilte Informationsstrukturen möglich sind, so scheint eine einheitliche Implementierung für jede Entität im gesamten Semantic Web in einem reputationsbasierenden System auf der Basis von PeerTrust relativ unrealistisch.

3.7 TidalTrust

Die von Jennifer Golbeck in ihrer Ph.D. These [CTSN05] vorgestellte TidalTrust-Metrik wurde aus den Erfahrungen aus zwei sozialen Vertrauensnetzwerken entwickelt. Das Trust-Projekt „Trust and Reputation in Web Based Social Networks“ [MSTR] baut ein Vertrauensnetzwerk auf der Basis von verteilten Informationen im Semantic Web auf. Das FilmTrust Projekt [FILM] verbindet soziale Netzwerke mit einem Filmbewertungssystem.

Die Grundlage der TidalTrust-Metrik sind soziale Netzwerke, die aus Bewertungen von Entitäten durch andere Entitäten bestehen. Bei der Verwendung von Vertrauensbewertungen sollten die inferierten Vertrauensbewertungen zwischen zwei Entitäten möglichst nahe an tatsächliche Bewertungen heran kommen. Jennifer Golbeck konnte zeigen, dass ihr Algorithmus mit hoher Präzision Vertrauensbewertungen inferieren kann, in dem sie einzelne Bewertungen in Netzwerken entfernte, und diese durch die Metrik über das Netzwerk berechnen ließ.

Eigenschaften des Metrikalgorithmus

Der Metrikalgorithmus berechnet auf der Basis von Netzwerkgraphen mit gerichteten und gewichteten Kanten eine virtuelle Kante von einer Quelle zu einem Ziel, sofern keine Kante zwischen ihnen existiert. Das Gewicht der virtuellen Kante wird dabei aus den Kanten der kürzesten Pfade zwischen den beiden Entitäten berechnet.

In einem ersten Schritt sucht der Algorithmus in einem Breitensuchverfahren einen kürzesten Pfad zwischen den Entitäten. Kürzere Pfade führen nach Jennifer

Golbeck [CTSN05] zu einer höheren Genauigkeit der inferierten Werte. Nachdem alle Pfade dieser Länge zwischen der Quelle und dem Ziel gefunden wurden, bricht das Suchverfahren ab.

Während der Suche wird der maximale Flow der kürzesten Pfade berechnet, der sich aus der Kante mit dem kleinsten Gewicht entlang des Pfades ergibt. In die im zweiten Schritt folgende Berechnung des Gewichts der virtuellen Kante zwischen der Quelle und dem Ziel werden nur jene Pfade einbezogen, die keine Kante mit einer Bewertung unterhalb des maximalen Flows besitzen. Jennifer Golbeck konnte zeigen, dass höhere Vertrauensbewertungen in einem zugrunde liegenden sozialen Netzwerk zu einer höheren Genauigkeit der inferierten Vertrauensbewertungen führt [CTSN05]. Auf diese Weise wird verhindert, dass Knoten, die eine schlechte Bewertung bekommen haben, mit ihren Bewertungen Einfluss auf das Endergebnis nehmen können.

Im zweiten Schritt wird die Bewertung der virtuellen Kante errechnet. Für jede Entität entlang der ausgewählten Pfade wird die Bewertung des Ziels des Vertrauens als virtuelle Kante berechnet, in dem die Bewertungen der Nachfolger im Pfad durch einen gewichteten Mittelwert zusammengefasst werden. Eine Ausnahme bilden nur die letzen beiden Entitäten im Pfad, da die Letzte das Ziel des Vertrauens selbst ist und die vorletzten Entitäten selber eine Bewertung des Ziels haben. Dieser Prozess wird fortgesetzt bis er die Quelle des Vertrauens erreicht hat.

Das Gewicht der virtuellen Kante zwischen der Quelle und dem Ziel des Vertrauens ist das Ergebnis des TidalTrust-Metrik. Er hängt klar von der Lokalität einer Entität im Netzwerkgraphen ab. Die Perspektive auf das Netzwerk ist also lokal. Von der Position der Quelle des Vertrauens aus wird nur ein kleiner Teil der Bewertungen des Netzwerkes ausgewertet, da die Länge der kürzesten Pfade aufgrund des sozialen Netzwerkes nach Stanley Milgram [SMP67] relativ kurz bleibt.

Ob die berechnete Bewertung auf für einen positiven oder negativen Schluss in das Ziel schließen lässt, wird mit einem Schwellenwert entschieden. Je nach

Anforderungen und verwendetem Bewertungschema lässt sich mit ihm steuern, wie stark eine akzeptable Bewertung sein muss.

Datenbasis

Um mit der TidalTrust-Metrik Vertrauen inferieren zu können, setzt Jennifer Golbeck soziale Netzwerke mit gerichteten und gewichteten Vertrauensbewertungen voraus. Auf dieser Basis kann das Ergebnis des Algorithmus als gewichtete Vertrauensbewertung interpretiert werden. Da Vertrauensbewertungen zwischen Entitäten eines Netzwerkes die Grundlage der Metrik sind, müssen die Identitäten der Entitäten mindestens als Pseudonyme bekannt sein. Das Formulieren von direkten Vertrauensbeziehungen wird vom Benutzer von reputationsbasierenden Systemen auf der Basis der TidalTrust-Metrik einen hohen Verwaltungsaufwand fordern, da er seine Vertrauensbewertungen zu Nachbarn selber ausdrücken muss.

Über die Sichtbarkeit und die Lokalität der Daten sowie deren Entstehung trifft der Algorithmus keine Annahmen. Auch der Bewertungsprozess kann von kontrolliert bis willkürlich variieren. Im „Trust and Reputation“ Projekt [MSTR] wird eine verteilte offene Informationstruktur verwendet, die durch im Semantic Web verteilte FOAF Dateien [FOAF] realisiert ist. Sie werden von jedem Autor selbst verwaltet, so dass aus Sicht eines reputationsbasierenden Systems der Bewertungsprozess willkürlich abläuft.

Das FilmTrust Projekt [FILM] hingegen verwendet eine zentrale Informationsarchitektur. Die Einhaltung des Bewertungsschemas und die Prüfung der Identität wird durch das System gewährleistet. Die Informationen werden nur teilweise veröffentlicht und nur registrierten Mitgliedern zugänglich gemacht. Für beide Netzwerke kann die TidalTrust-Metrik eingesetzt werden.

Weitere Eigenschaften

Die TidalTrust-Metrik wurde explizit für soziale Netzwerke geschaffen. Ihr wohnen daher die typischen Probleme von Metriken für soziale Netzwerke inne. Für den Fall, dass ein Neuling in ein WoT eintritt, hat er noch keine Bewertungen verfasst und kann somit keine Bewertungen evaluieren.

Ein großer Vorteil besteht für den Algorithmus in der guten Skalierung mit der Größe des sozialen Netzwerkes. Durch die Verwendung von direkten Bewertungen müssen im Vergleich zu transaktionsbasierenden Daten nur wenig Daten ausgewertet werden. Der viel größere Vorteil liegt aber in der Tatsache, das die TidalTrust-Metrik nur Bewertungen im lokalen Umfeld der Quelle des Vertrauens auswertet. Unter der Annahme, das die Netzwerkdaten ein soziales Netzwerk aufspannen, kann davon ausgegangen werden, das der Aufwand für das Finden kürzester Pfade logarithmisch mit der Anzahl an Knoten im Netzwerk wächst.

Hinsichtlich der Manipulierbarkeit der Metrik durch verfälschte Bewertungen sollten die Identitäten der Entitäten im Netzwerk bekannt sein, um zu verhindern, das sich Entitäten mehrfach anmelden. Durch Scheinidentitäten können sich einzelne Entitäten sonst bessere Bewertungen organisieren oder die Reputation anderer Entitäten beeinflussen.

Nach Raphael Levien stellt die Manipulierbarkeit ein großes Problem für Metrik dar, die skalare Vertrauensbewertungen evaluieren. Im Falle der TidalTrust-Metrik müssen die kürzesten Pfade manipuliert werden. Da aus allen Pfaden zur Quelle des Vertrauens nur die kürzesten verwendet werden, müssen nur wenige Informationen manipuliert werden, um das Ergebnis gezielt zu verändern. Zudem wird die Entstehung von Bewertungen durch die Verwendung von direkten Bewertungen nicht mehr nachvollziehbar. Jeder könnte also gezielt falsche Bewertungen abgeben. Dieser Umstand kann nur durch die Kontrolle des sozialen Netzwerkes durch die Bestrafung von schlechten Bewertern mit negativen Bewertungen kompensiert werden.

Zur Privatsphäre und zur Qualitätssicherung können keine Aussagen gemacht werden, da diese im Wesentlichen von der Datenbasis und dem Bewertungsprozess abhängen. Da über diese Bereiche keine Annahmen gemacht wurden, hängen diese Eigenschaften von der Implementierung eines reputationsbasierenden Systems mit der TidalTrust-Metrik ab.

Ein großer Vorteil der TidalTrust-Metrik liegt im einfachen Verständnis ihrer

Ergebnisse sowie ihrer leichten Parametrisierung. Ihr Ergebnis kann als eine virtuelle Kante mit einem Gewicht verstanden werden, die einer eigenen Bewertung des Ziels des Vertrauens entspricht. Als Parameter benötigt sie nur den Graphen des sozialen Netzwerkes mit den Bewertungen, den Schwellenwert für das minimale Gewicht, dass die virtuelle Kante annehmen muss, und die Identifikation der Quelle und des Ziels des Vertrauens.

Einordung in den Kontext des Semantic Web

Anhand der Kriterien, die in dieser Diplomarbeit für die Verwendbarkeit von Trust-Metriken im Semantic Web aufgestellt wurden, eignet sich die TidalTrust-Metrik für den Einsatz in reputationsbasierenden Systemen für das Semantic Web. Ihre lokale Perspektive gewährleistet die Berücksichtigung subjektiver Interessen eines Benutzers. Die Offenheit für verschiedene Informationsstrukturen macht auch Implementierungen für offene und verteilte Informationsstrukturen möglich. Zudem spricht ihre unkomplizierte Handhabung und das intuitive Verständnis für einen benutzernahen Einsatz.

3.8 Weitere reputationsbasierende Trust-Metriken

Über die bis hierher vorgestellten reputationsbasierenden Trust-Metriken hinaus hält die Literatur eine weitaus größere Sammlung an Trust-Metriken bereit. Die zuvor vorgestellten Metriken wurden vor allem wegen ihrer Aktualität und Relevanz für das Thema der reputationsbasierenden Trust-Metriken für das Semantic Web ausgewählt. Zudem sollte ein möglichst breites Spektrum an Metriken vorgestellt werden, um das Potenzial der Klassifikation für reputationsbasierende Metriken aufzuzeigen.

Für die Aufstellung der Klassifikation wurden weit mehr Metriken berücksichtigt, als intensiv diskutiert wurden. Die meisten von ihnen stehen in einem engen Zusammenhang mit den diskutierten Metriken, da sie als Grundlage dienten oder Weiterentwicklung sind. Zu den Metriken, die eine Eigenvektorbestimmung verwenden, gehören neben den diskutierten PageRank [PRCR98], EigenTrust [EARM03] und EigenRumor [ERRB05] auch die von Guha in [ORS03] entwickelten PageRank-basierenden Metriken und der von Richardson u.a. entwickelte

Distributed PageRank-Algorithmus für P2P-Systeme [EPPP03]. Als Vertreter der pfadbasierenden Metriken wurde die TidalTrust-Metrik im Detail betrachtet. Weitere pfadbasierende Metriken finden sich in [AMIT04] mit dem Reputation Inference Algorithm und in [CPGN03] mit dem Vertrauensmechanismus im NICE Framework für verteilte Anwendungen. Ein ähnliches Model zum Evidence-Based Model [EMDR02] findet sich in [MTPP01]. Dort werden Beschwerden über Peers in einem P2P-System mittels einer Heuristik ausgewertet und hinsichtlich ihrer Vertrauenswürdigkeit gute Services zu leisten interpretiert.

Eigenschaften	*Appleseed*	*EigenRumor*	*EigenTrust*	*Evidence-Based Model*	*PageRAnk*	*PeerTrust*	*TidalTrust*
Perspektive	l	g	g	l	g	g	l
Auswahlverfahren	r	r	r	t	r	s	s
Vertrauensbeziehungen	g	g	g	s+g	g	g	s
Sichtbarkeit der Datenbasis	o/(g)	o/(g)	o	o(+g)	o/(g)	o/(g)	o/g
Lokalität der Datenbasis	v/(z)	z/(v)	v	v	v/(z)	v/(z)	v/z
Identität des Bewerters	b/p	b/p	p	b/p	a	b	b/p
Grapheigenschaften	sn	b	-	(sn)	g	-	sn
Form von Bewertungen	d	t+d	f	f+d	t	f+t	d
Erstellung von Bewertungen	w	a(+k)	k	w	w	k	w
Ort der Evaluierung	z	z	v	z(v)	z	v	z
Interpretation	v	r	s	s	r	v	v

Legende:

Perspektive: g = global, l = lokal
Auswahlverfahren: s = schwellenwertbasierend, r = rangbasierend
Vertrauensbeziehungen: s = skalar, g = gruppenweise
Sichtbarkeit: o = offen, g = geschlossen
Lokalität: z = zentral, v = verteilt
Identität des Bewerters: i = bekannt, p = pseudonym, a = anonym
Form von Bewertungen: d = direkt, f = feedbackbasierend, t = transaktionsbasierend
Grapheigenschaften: sn = soziales Netzwerk, g = geschlossener Graph, b = bipartiter Graph
Erstellung von Bewertungen: k = kontrolliert, a = autorisiert, w = willkürlich
Interpretation: v = Vertrauenswürdigkeit, r = Relevanz, s = Servicequalität

Tabelle 2: Zusammenfassung der Metrikeigenschaften

4 Implementierung von Trust-Metriken für die TriQL-Trust-Architektur

In diesem Kapitel wird detailiert auf die Realisierung von reputationsbasierenden Trust-Metriken für die TriQL-Trust-Architektur eingegangen. Um zu einem tieferen Verständnis des Umfeldes der Architektur und der Implementierung der Metriken zu gelangen, sollen in 4.1 die Basistechnologien kurz erläutert werden. Im Anschluss werden in 4.2 die relevanten Teile für die Implementierung von Metriken der Architektur aufgegriffen. Abschnitt 4.3 geht auf die allgemeinen Schnittstellen der Metriken in der TriQL.P-Anfragesprache und der Anbindung an den *Query and Evaluation Layer* sowie die vier implementierten Metriken ein. Die Generierung von Erklärungen durch die Metriken wird in 4.4 erläutert. Abschliessend werden in 4.5 und 4.6 zwei Anwendungsbeispiele der TriQL-Trust Architektur unter Verwendung von zwei implementierten Metriken gezeigt.

4.1 Basistechnologie

4.1.1 Resource Description Framework (RDF)

RDF [RDF] wird von der Semantic Web Activity des W3C [W3C] entwickelt. Es handelt sich dabei um eine standardisierte aussagenlogische Sprache zur Beschreibung von Informationen durch Metainformationen. Ihre Aussagen bestehen aus Tupeln mit drei Ressourcen, genannt Triple. Die erste Ressource genannt Subjekt identifiziert den zu beschreibenden Gegenstand. Die dritte Ressource genannt Objekt kann sowohl ein anderer Gegenstand also auch ein Literal, also ein konkretes Datum sein. Die verbliebene zweite Ressource identifiziert eine Beziehung zwischen dem Subjekt und dem Objekt und wird Prädikat genannt. Ein einfaches Beispiel in einer Pseudo-RDF-Syntax wäre: „Oliver“ „kennt“ „Chris.“

Auf der Basis dieser Struktur können komplexe Sachverhalte mit einfachen Aussagen beschrieben werden. Sie bauen auf diese Weise einen Graphen von Aussagen auf, in dem Subjekte und Objekte die Knoten und zusammen mit Prädikaten zu gerichteten Kanten werden. Die Identifikation von Ressourcen wird dabei

mit URIs realisiert.

4.1.2 Named Graphs (NG)

Nach dem mit RDF bereits einige Erfahrungen gemacht wurden, kamen immer wieder Probleme auf, die sich mit Hilfe der einfachen Struktur von RDF nur schwer lösen ließen. Einige Initiativen versuchten daher RDF, um neue Konzepte zu erweitern, um die Aussagekraft zu erhöhen. Named Graphs (NG) [NG] sind eine solche Erweiterung.

In RDF werden komplexe Aussagen mit einem Graphen von Aussagen dargestellt. Der Standard selbst bringt allerdings keine Mechanismen mit, die es ermöglichen, komplexere Aussagen also Teilgraphen zueinander in Beziehung zu setzten. Die Realisierung von Zugriffskontrollen auf Subgraphen, der Bestimmung der Herkunft der Aussagen oder die Bewertung von Aussagen gestaltet sich so schwierig [NGPT05]. Um dieses Problem zu lösen wird der RDF Standard um Namen für Graphen erweitert, wobei die Namen durch URIs umgesetzt werden. Eine Aussage gehört somit immer zu mindestens einem Graphen, so dass eine Aussage nun kein Tripel sondern ein Quad (ein Tupel mit vier Objekten) ist. Der Graphname kann als Ressource in Aussagen verwendet werden, wodurch Aussagen über Graphen möglich werden.

4.1.3 SWP – Vokabular für Web-Veröffentlichungen

Eine der ersten Anwendungen von NGs ist das Semantic Web Publishing (SWP) Vokabular [NGPT05]. Es definiert einige Klassen und Beziehungen in RDF, mit denen sich Autoritäten für den Inhalt von RDF Graphen verantwortlich zeichnen können. In Abbildung 7 ist die Struktur des Vokabulars veranschaulicht. Als zentrales Konzept des SWP Vokabulars gilt die Klasse swp:Warrant. Instanzen dieser Klasse stellen die Verbindungen zwischen einer Autorität und der von ihr veröffentlichten Graphen dar. Digitale Signaturen stellen sicher, dass eine nicht autorisierte Veränderung eines veröffentlichten Graphen erkannt und die Identität von Autoritäten bestätigt werden kann.

Auf diese Weise kann die Herkunft von Informationen im Semantic Web bestimmt und Verantwortlichkeit für Inhalte aufgezeichnet werden.

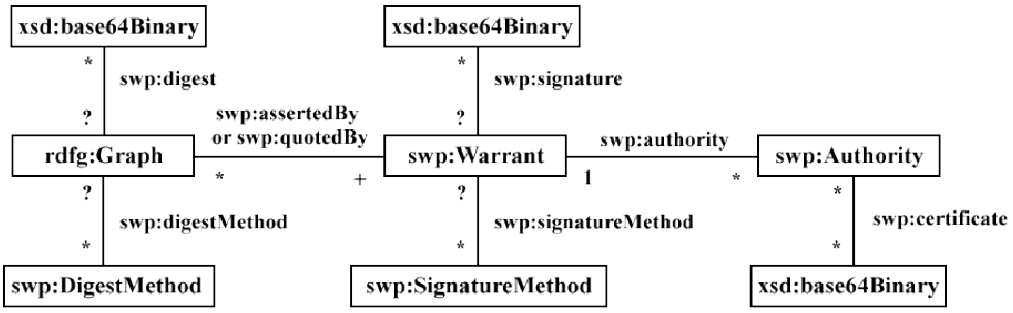

Abbildung 7: Struktur des Semantic Web Publishing RDF-Vokabulars [NGPT05]

4.1.4 TriQL - Named Graphs Anfragesprache

Die Erweiterung von RDF um NG fordert ebenfalls die Erweiterung von Tools für den Zugriff auf den Datenbestand. TriQL [TQL] wurde ursprünglich als Erweiterung der RDF Anfragesprache RDQL [RDQL] für NGs entwickelt. Statt wie bisher mit einem einzigen anonymen Graphmuster über einem globalen anonymen Graphen nach Informationen zu suchen, sollen TriQL-Anfragen mit verschiedenen benannten Graphmustern zugleich über der Menge aller Graphen suchen. Das Ergebnis sind Informationen auf, die alle Graphmuster passen.

4.2 TriQL-Trust-Architektur

In Abschnitt 1.3 wurde die TriQL-Trust-Architektur bereits grob vorgestellt. Im Folgenden soll detailiert auf den *Query and Trust Evaluation Layer* eingegangen werden, der die Implementierung der reputationsbasierenden Trust-Metriken enthält. Anschließend wird der „TriQL.P Trust Policy Enabled Semantic Web Browser“ vorgestellt, der sowohl eine Anwendung im *Application and Explanation Layer* als auch eine Implementierung des *Information Integration Layers* ist. Der *Repository Layer* besteht im wesentlichen aus einer Datenbank, auf die hier nicht weiter eingegangen werden soll.

4.2.1 Der Query and Trust Evaluation Layer

Der mehrheitlich von Richard Cyganiak entwickelte *Query and Trust Evaluation Layer* (siehe Abbildung 1) nimmt Anfragen und Trust-Policies vom *Application and Explanation Layer* entgegen und führt die Anfragen auf der Basis der Informationen im *Repository-Layer* aus. Die Anfragen werden in Form von RDF Triple-Mustern gestellt. Trust-Policies, die zuvor geladen werden müssen,

könne mit URIs, die sie identifizieren, ausgewählt werden.

Eine Trust-Policy entspricht aus der Sicht von RDF und NGs einem Muster von NG-Strukturen. Mit der Verwendung von Trust-Policies wird die Möglichkeit geschaffen problemspezifische Graphmuster zu definieren. So kann auf verschiedene Kontexte flexibel mit der Auswahl einer passenden Trust-Policy reagiert werden.

Die Ergebnisse einer Anfrage werden hinsichtlich ihrer Informationsqualität bewertet, in dem sie und die sie umgebenden Graphstrukturen im Repository mit dem Muster der Policy zur Deckung gebracht werden. Kommt eine Deckung zustande, wird die Informationsqualität als hinreichend hoch betrachtet. Einfache Graphstrukturen, werden mit statischen Graphmustern abgefragt. Für komplexere Strukturen wie z.B. die Reputation einer Entität auf der Basis der Ratinginformationen im Graphen lassen sich hingegen nicht mehr mit einfachen Graphmustern prüfen. Metriken sind hier der Ausweg, den die TriQL-Trust-Architektur einschlägt, da sie komplexe Strukturen durch fest definierte Regeln nach und nach aufbauen und prüfen können. Diese Eigenschaft wird von den reputationsbasierenden Metriken genutzt.

Trust-Policies werden mit dem RDF Vokabular „Trust Policy Language“ (TPL) formuliert. In Abbildung 8 ist eine solche Policy dargestellt. Sie enthält vor allem Graphmuster, die mit Vorlagen für die Generierung der Erklärungen der Evaluierung versehen sind. Desweiteren können COUNT-Direktiven oder Metriken enthalten sein.

Für die Ausführung der original TriQL-Anfrage unter Beachtung der Trust-Policy, wird eine neue Anfrage in TriQL.P aus der Originalanfrage und den Mustern und Metriken in der Trust-Policy generiert. Abbildung 8 stellt eine solche Generation schematisch dar. Die Originalanfrage wird in ein Graphmuster eingepackt, welches von den Graphmustern in den Trust-Policies referenziert werden kann. In die TriQL.P-Anfrage werden anschließend die Graphmuster der Policies angehängt bevor COUNT-Direktiven und Metriken hinzugefügt werden.

Prinzipiell sollen verschiedene Metriken und COUNT in einer Policy

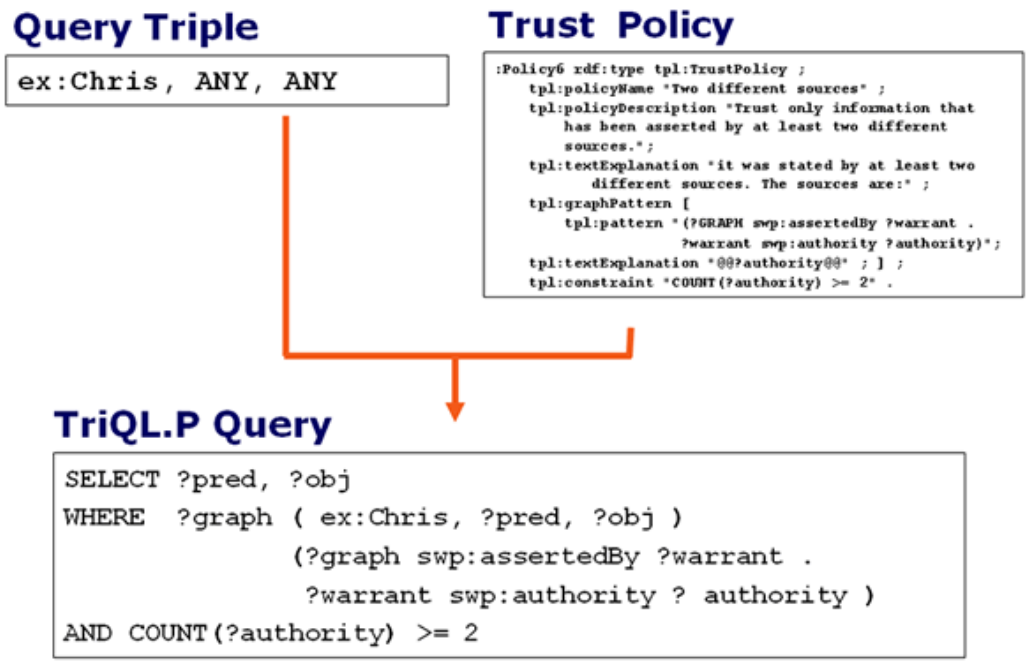

Abbildung 8: Generierung einer TriQL.P Anfrage [TPWB]

verwendet werden können. Aufgrund der Limitierungen der aktuellen Implementierung können nur mehrere schwellenwertbasierende Trust-Metriken in einer Policy ausgeführt werden. Rangbasierende Metriken und COUNT können nur einzeln verwendet werden. Auf die Verwendung von Graphmustern hat dies keinen Einfluss.

In der Ausführung der TriQL.P-Anfrage werden zuerst passende Informationen auf die Originalanfrage gebunden. Anschließend wird auf sie ein Graphmuster nach dem anderen angewandt. Sobald ein Graphmuster nicht auf eine Ergebnisinformation passt, wird sie heraus gefiltert. Die verbleibende Menge an Informationen wird zuletzt mit den Metriken überprüft. Für alle verbleibenden Informationen wird anschließend an die Vertrauensbewertung eine Erklärung generiert, die aus den Erklärungsvorlagen der Graphmuster und den Erklärungen der Metrikinferenzen zusammengesetzt wird. In Abbildung 9 wird dieser Prozess schematisch dargestellt.

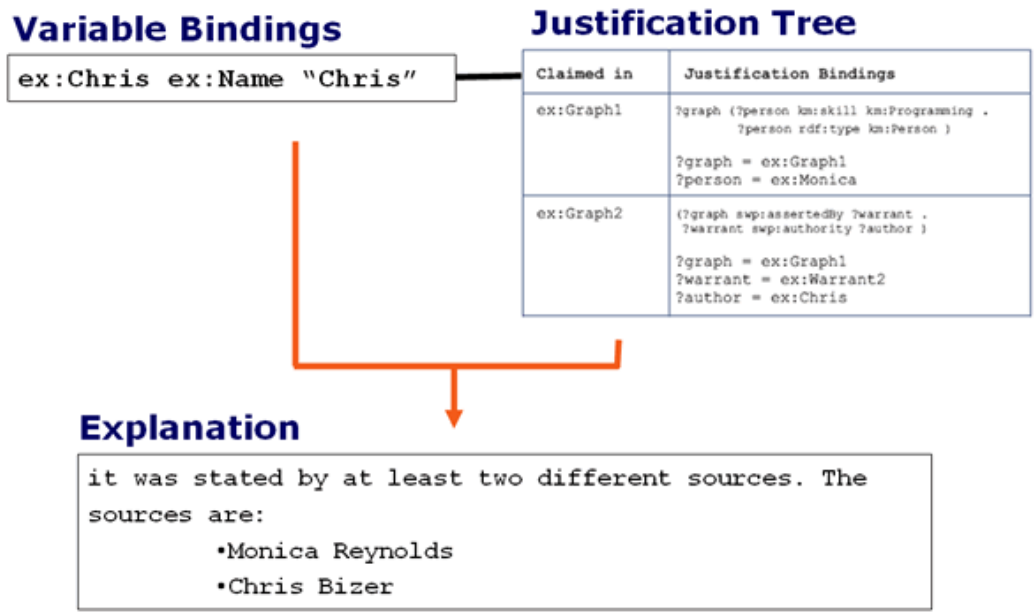

Abbildung 9: Ausführung der Vertrauensanalyse [TPWB]

Die Ergebnisinformationen werden als RDF Triple Menge an den *Application and Explanation Layer* zurückgegeben. Die Erklärungen der Inferenzen zu jedem Triple werden vom *Query and Trust Evaluation Layer* im *Repository-Layer* gecacht und auf Anfrage des *Application and Explanation Layer* herausgegeben.

4.2.2 PiggyBank und der TriQL.P Trust Policies Enabled Semantic Web Browser

PiggyBank [PIGY] ist eine Erweiterung des Firefox Browsers aus dem SIMILE Projekt [SIMI], um Firefox zu einem Semantic Web Browser zu ergänzen. Ein lokales RDF Repository wird verwendet, um Informationen aus dem WWW bei Bedarf zu sammeln und in das Repository zu integrieren. Dabei werden nicht nur Informationen, die in RDF kodiert sind, integriert, sondern insbesondere auch Informationen von gewöhnlichen Webseiten durch Screengraber eingebunden. Durch das lokale Repository kann man gezielt nach Informationen suchen und ergänzende Informationen hinzufügen.

Die TriQL-Trust-Architektur wurde von Tobias Gauß in PiggyBank integriert, um auf der einen Seite eine Anwendung im *Application and Explanation Layer* mit einer guten Oberfläche zu implementieren und auf der anderen Seite die Funktionalitäten von PiggyBank für den *Information Integration Layer* einsetzten

zu können. Das Ergebnis ist der „TriQL.P Trust Policy Enabled Semantic Web Browser“

Die Oberfläche von PiggiBank wurde um zwei Elemente erweitert, um die Funktionalität der TriQL-Trust-Architektur zugänglich zu machen. Mit einem neuen Auswahlfeld können Trust-Policies für die Anwendung auf Suchergebnisse ausgewählt werden. Die Ergebnislisten enthalten entsprechend der Trust-Policies nur vertrauenswürdige Informationen. In Abbildung 10 sehen wir die Policy „TidalTrust Value above 0,5“ ausgewählt. Neben den Ergebnisinformationen wird als zweite Erweiterung ein Button mit einem Fragezeichen dargestellt. Hinter im verbergen sich Erklärungen, warum die Informationen vertrauenswürdig sind. Um neue Trust-Policies und vorhandene Daten in das Repository laden zu können, wurde zusätzlich die Menustruktur von Firefox ergänzt.

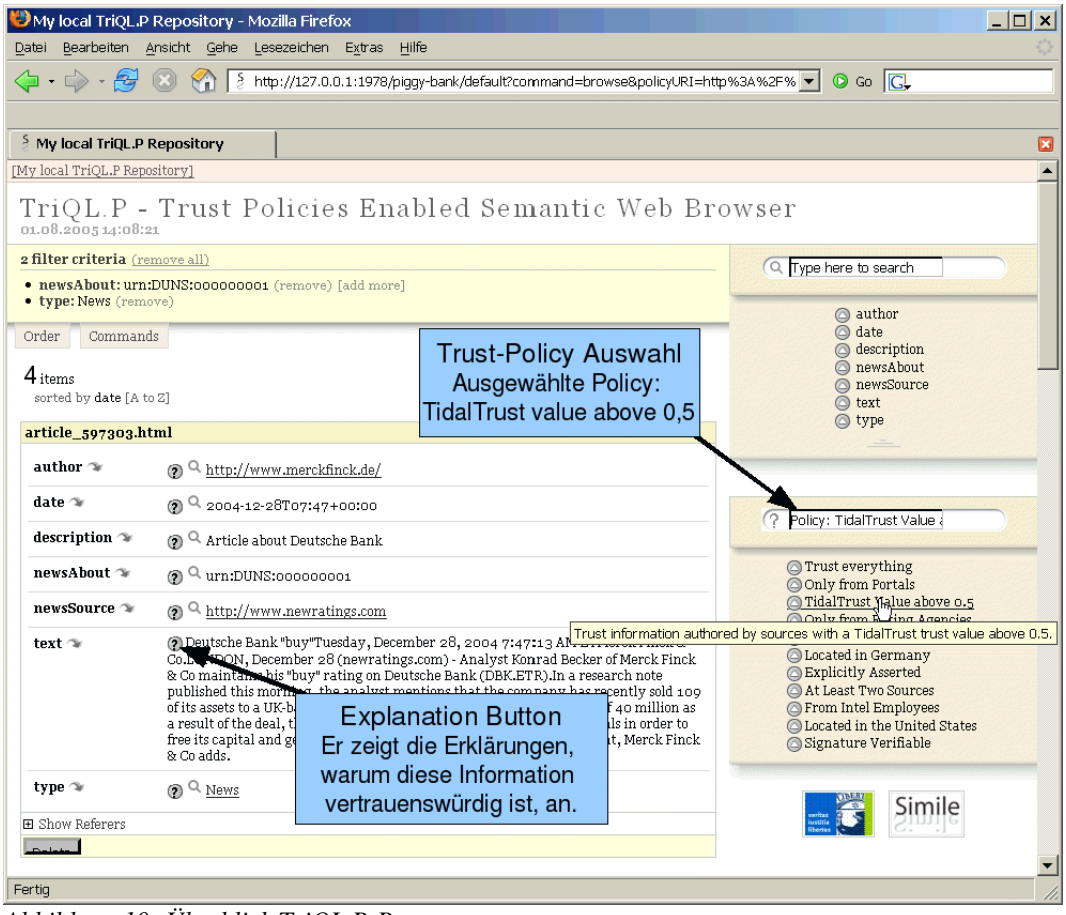

Abbildung 10: Überblick TriQL.P-Browser

Im Hintergrund werden die Suchanfragen nicht mehr nur als Abfragen des

Repositories ausgeführt, sondern durch die TriQL-Trust-Architektur gleichzeitig anhand einer Trust-Policy geprüft und gefiltert.

4.3 Trust-Metriken für die TriQL-Trust-Architektur

Trust-Metriken werden in der TriQL-Trust-Architektur für die Bewertung verschiedener Aspekte von Vertrauen und Informationsqualität verwendet. Die vorgestellten Metriken verwenden reputationsbasierende Mechanismen, um die Vertrauenswürdigkeit von Informationsprovidern und der Qualität von Informationen zu evaluieren. Wie die einzelnen Bewertungen der Metriken zu dem Ziel der TriQL-Trust-Architektur Informationsqualität zu bewerten stehen, wird durch die Einbindung in Trust-Policies geregelt.

Die Trust-Metriken inferieren durch die Analyse von Informationen hinsichtlich der Bewertungskriterien. Die zu analysierenden Strukturen, die Merkmalen wie Reputation zugrunde liegen, können eine hohe Komplexität erreichen und sind mit statischen Graphmuster nicht erfassbar. Sie machen die Analyse von Graphstrukturen über klar definierten Regeln notwendig. Reputation wird daher in der TriQL-Trust-Architektur mit speziellen Trust-Metriken inferiert, die jeweils ihr eigenes Regelwerk für die Evaluierung von Vertrauen oder Informationsqualität mit sich bringen.

Zunächst soll die Einbindung von Trust-Metriken in die Anfragesprache TriQL erläutert werden. Anschließend wird die Anbindung der Metrikimplementierungen in die TriQL-Query-Engine erläutert, bevor auf die einzelnen implementierten Metriken eingegangen wird.

4.3.1 Einbindung in TriQL – Das Metrikinterface

Die Abfragesprache TriQL kennt keine Möglichkeiten, um weitere Funktionalitäten aufzunehmen. Um komplexere Strukturen abfragen zu können, müssen jedoch neue Funktionen z.B. durch Metriken eingebunden werden. Aus diesem Grund wurden die TriQL Syntax erweitert zu TriQL.P [TQLP]. Neben einer COUNT Direktive kam eine offenes Schnittstelle für Metriken hinzu.

Die Schnittstelle wird durch das Schlüsselwort „METRIC“ identifiziert und

enthält in nachfolgenden Klammern eine kommaseperierte offene Liste von Argumenten, die RDF Ressourcen sind. Das erste Argument identifiziert dabei immer die Metrik mit einer URI. Sie muss mit der URI übereinstimmen, die von der getURI-Methode der entsprechenden Metrik zurückgeben wird (siehe Metrikschnittellen in Abbildung 12 und Abbildung 13). Die weiteren Argumente sind abhängig von der Metrik und werden direkt an sie weitergeleitet und von ihr ausgewertet. Da die Argumente vom Typ einer RDF Ressource sind, können sie beliebige Informationen enthalten. Die Metrik selbst legt fest, welche Argumente sie an welcher Stelle mit welchem Typ akzeptiert.

```
SELECT ?info
WHERE ?graph ( ex:SomeItem ex:SomeProperty ?info )
AND
METRIC(metric:MetricName, argument1, ... , agrumentN)

USING ex    FOR   <http://www.example.org/vocabulary#>
      metric FOR  <http://www.examlple,org/metric#>
```

Abbildung 11: Beispiel Anfrage mit TriQL.P unter Verwendung der Metrikschnittstelle

Dieses Interface hat sich in der Handhabung für die vier implementierten Metriken als sehr flexibel erwiesen. Neben einfachen Daten von verschiedensten Typen können auch komplexe Strukturen an die Metriken übergeben werden. Komplexe Strukturen werden durch eine URI oder ein Blank Node[7] des Graphen identifiziert und übergeben. Die Schnittstelle eignet sich somit für beliebig komplexe Metrikschnittstellen.

4.3.2 Einbindung in den Query and Evaluation Layer

Der *Query and Evaluation Layer* filtert die Ergebnisse einer Anfrage zuerst mit den Graphmustern der Trust-Policies und wendet auf deren Ergebnisse die Metriken an. Dabei werden zuerst die schwellenwertbasierenden Metriken angewendet, da sie sich auf jede Information der Ergebnismenge anwenden lassen. Zuletzt werden die rangbasierenden Metriken ausgeführt, wobei diese der

7 Blank Nodes sind ein RDF Konzept. Sie sind Knoten in einem RDF Graphen, die keinen Namen in Form einer URI besitzen. Sie können daher nicht direkt adressiert werden, und kommen nur über die Struktur des Graphen zur Geltung. Mehr Details lassen sich unter [RDF] nachlesen.

Beschränkung unterliegen, dass pro Anfrage nur eine rangbasierende Metrik bzw. COUNT-Direktive ausgeführt werden kann.

Die Unterscheidung nach rangbasierenden und schwellenwertbasierenden Metriken wird notwendig, weil sie grundverschieden in der Ausführung sind. Rangbasierende Metriken werden auf eine komplette Ergebnismenge angewendet, da die einzelnen Elemente durch das Ranking zueinander in Beziehung gesetzt werden. Das Ergebnisranking enthält wiederum alle Informationen, die bewertet werden sollten. Schwellenwertbasierende Metriken werden hingegen auf jedes Element einzeln angewandt. Die Bewertungen erfolgen unabhängig von den Bewertungen anderer Ergebnisse auf eine Anfrage. Das Ergebnis ist eine gefilterte Untermenge der zu bewertenden Informationen. Auf Grund dieser Unterschiede muss die Architektur diese beiden Metriktypen trennen. Eine Metrik implementiert daher eines von zwei Schnittstellen für Metriken.

Da der *Query and Evaluation Layer* keine Sortierung nach Rang gestatten, sondern nur das Filtern von vertrauenswürdigen Informationen, musste für rangbasierende Metriken eine Lösung gefunden werden, wie die Ergebnisse eines Rankings effizient für die Filterung nutzbar sind. Die verwendete Lösung liegt in der Bestimmung der Top X Informationen, denen eine positive Bewertung hinsichtlich des Bewertungskriteriums bescheinigt wird. Diese Entscheidung hat bei einem sortierten Ranking der Benutzer zur Zeit der Anzeigen der Ergebnisse zu treffen. Diese Entscheidung wurde nun durch das Festlegen der Top X durch ein Argument in die Metrikausführung verlagert, so dass die Metrik über eine positive oder negative Bewertung anhand der Position einer Information im Ranking entscheiden kann. Dieses Verfahren funktioniert nur unter der Annahme, dass die Metrik ein für den Benutzer ideales Ranking produziert, an dem ein Benutzer keine weiteren Selektionen vornehmen muss. Diese Annahme ist sehr stark, sie stellt aber in der aktuellen Implementierung der TriQL-Trust-Architektur die einzige Möglichkeit dar, rangbasierende Metriken zu realisieren.

Da die Architektur für rangbasierende Metriken jede Information einer Ergebnismenge überprüfen muss, ob die Information sich in den Top X befindet, wird

jedes mal das Ranking benötigt. Da das Ranking einer Ergebnismenge von den Argumenten der Metrik ausser dem Ziel des Vertrauens und dem Top-X Argument abhängt und die weitern Argumente oft gleich bleiben, macht es Sinn, Rankings zu cachen und nur dann ein neues Ranking zu berechnen, wenn sich die weiteren Argumente ändern.

Rangbasierende Metriken werden ausgeführt, wenn alle anderen Filterprozesse über Graphmuster und schwellenwertbasierende Metriken abgeschlossen sind. Zu diesem Zeitpunkt der Anfrageauswertung, enthält die Ergebnismenge zu jeder Information alle Belegungen der Variablen der Anfrage. Um die rangbasierende Metrik ausführen zu können, müssen für alle Informationen die entsprechenden Parametrisierungen der rangbasierenden Metrik vorliegen, damit die Metrik auf die gesamte Ergebnismenge angewendet werden kann. Die Parametrisierungen werden aus den Variablenbelegungen erzeugt und durch eine Liste von Argumentlisten realisiert. Über die init-Schnittstelle (Abbildung 12) wir die Parametrisierung als eine Tabelle der Metrikargumente für die Evaluierung übergeben. Jede Reihe dieser Tabelle entspricht einem Satz Argumente für die jeweilige zu bewertende Information. Die Ergebnisse, der anschließenden Evaluierung werden im Cache der Metrik gespeichert. Aus dem Cache heraus können Ergebnisse der Metrikevaluierung erfragt und erklärt werden.

Abbildung 12: Programmierschnittstelle für rangbasierende Metriken

Die Implementierung von schwellenwertbasierenden Metriken fallen wesentlich einfacher aus. Für jede Information in der Ergebnismenge einer Anfrage wird aufs Neue eine Bewertung vorgenommen, die zusammen mit dem Schwellenwert über das Vertrauen entscheidet. Mit einer Liste von Argumenten, die über die calculateMetric-Schnittstelle (Abbildung 13) an die Metrik übergeben wird, evaluiert die Metrik und gibt das Ergebnis mit Erklärung im Rückgabetyp

EvaluationResult zurück.

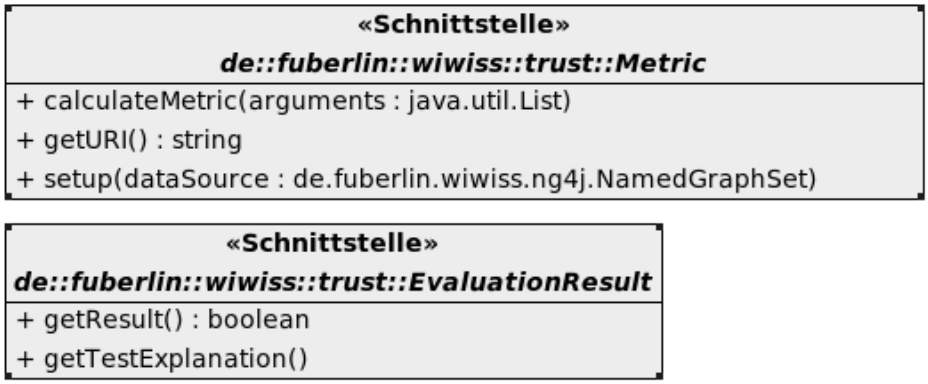

Abbildung 13: Programmierschnittstelle für schwellenwertbasierende Metriken

4.3.3 Appleseed-Implementierung

Die in 3.1 analysierte Appleseed-Metrik konnte sich durch ihre Eigenschaften gut für den Einsatz in der TriQL-Trust-Architektur qualifizieren. Ihre lokale Perspektive und ihre Ausrichtung für verteilte und offene Informationsstrukturen lassen sie als nahezu ideal erscheinen.

Für die Informationsgrundlage der Appleseed-Metrik sowie der TidalTrust-Metrik wurden das Vokabular der Trust-Ontology [MSTO] des „Trust and Reputation“ Projektes [MSTR] von Jennifer Golbeck verwendet. Sie definiert gewichtete Vertrauensaussagen von 0 für kein Vertrauen bis 10 für voll vertrauenswürdig, mit denen Entitäten sich gegenseitig bewerten können. Ein soziales Netzwerk, welches mit der Appleseed-Implementierung der TriQL-Trust-Architektur analysiert wird, muss daher dieses Vokabular verwenden.

Die Metrik prüft mit den Möglichkeiten des SWP Vokabulars, ob die Bewertungen tatsächlich von der Entität geäußert wurden, die als Subjekt in der Bewertung steht. Dies ist notwendig, da niemand die Qualität der Bewertungs-informtionen kontrolliert, aber die Identität des Autors von Bewertungen wichtig für eine geringe Manipulierbarkeit ist. Somit wird gewährleistet, das keine andere Entität für einen Bewertungen vornimmt.

Die Metrik benötigt für die Ausführung mindestens 4 Argumente. Das erste Argument identifiziert die Quelle des Vertrauens. Dies ist notwendig, da die Appleseed-Metrik eine lokale Trust-Metrik ist, die die Ausgangsposition in einem

WoT kennen muss, um das persönliche Umfeld der Quelle des Vertrauens analysieren zu können. Im zweiten Argument wird die Identität des Ziels des Vertrauens übergeben. Das Ziel soll sich im errechneten Ranking mindestens unter den Top *N* befinden, um als vertrauenswürdig zu gelten. *N* wird durch das dritte Argument an die Metrik übergeben. Das vierte benötigte Argument enthält die Injektion $in(x)_0$, ein Potenzial, das an Entitäten im persönlichen WoT der Quelle des Vertrauens mit Spreading Activation Strategien verteilt werden soll. Dieses Potenzial kann hier auch als Vertrauen interpretiert werden.

Die weitern 5 optionalen Argumente werden bei Bedarf in der offenen Parameterliste angehängt. Sind sie nicht vorhanden, werden sie mit Standard-werten belegt, die sich in Experimenten von Cai-Nicolas Ziegler [TDRS05] als geeignet ergeben haben. Das erste optionale Argument ist der Streuungsfaktor *d*, der darüber entscheidet, welcher Anteil des Potenzials, das in eine Entität injiziert wird, von ihr an Nachfolger weitergeben wird. Der Streuungsfaktor bewegt sich im Wertebereich von 0 bis 1 und wird auf 0,85 gesetzt, falls dieses Argument fehlt. Das zweite optionale Argument T_c bestimmt das Minimum an Änderung des Potentials zwischen zwei Iterationen des Berechnungsprozesses. Gibt es keine bewertete Entität, die eine Änderung um das Minimum aufweist, bricht der Annäherungspropzess ab. Wird kein Wert angegeben, so wird 0,05 als Standardwert verwendet. Das fünfte und sechste Argument bestimmen die maximale Anzahl untersuchter Knoten *M* und den maximalen Abstand *l* von Knoten zur Quelle des Vertrauens. Sie dienen vornehmlich zur Begrenzung des untersuchten WoT und werden standardmäßig auf eine Menge von 200 Entitäten und einer maximalen Pfadlänge von 6 gesetzt. Das letzte Argument *e* wird verwendet, um Entitäten die schlecht bewertet werden, stärker zu bestrafen. Die Gewichte der Kanten werden in der Metrik auf den Wertebereich [0..1] abgebildet. Wenn man diese Gewichte mit einem Exponenten größer 1 potenziert, werden Entitäten, die sehr hoch Bewertungen erhalten, also nahe 1 bewertet werden, deutlich besser gestellt als Entitäten, die mit geringeren Bewertungen bewertet werden. Wird dieses Argument nicht gesetzt wird der Exponent auf 1 gesetzt.

Die Parametrisierung der Appleseed-Metrik gestaltet sich mit den vielen Parametern sehr komplex. In der Beispielanwendung in Abbildung 14 wird die vollständige Parametrisierung mit den Standardwerten der optionalen Argumente gezeigt. Das Ziel des Vertrauens wird dabei unter den Top 10 gesucht. Das injizierte Potenzial beträgt 200. Da die Reihenfolge der Argumente wichtig ist, müssen für den Fall, das eines der hinteren optionalen Argumente gesetzt werden soll, zugleich die davorstehenden gesetzt werden.

```
...
METRIC(     metric:Appleseed,
            <http://source.net/>,
            <http://sink.net/>,
            10, 200,0.85, 0.05, 200, 6, 1)
USING metric FOR  <http://www.wiwiss.fu-berlin.de/suhl/bizer/TPL/>
```

Abbildung 14: Beispielanwendung der TriQL.P-Metrikschnittstelle für die Appleseed

Als rangbasierende Metrik implementiert Appleseed die RankBasedMetric-Schnittstelle. Da nicht für die Bewertung jeder einzelnen Information einer Ergebnissmenge ein neues Ranking berechnet werden muss, werden die Rankings gecacht. Solange alle Argumente ausser der Identität des Ziels des Vertrauens und die Anzahl der vertrauenswürdigen Entität *N* am Anfang des Rankings gleich bleiben, können Bewertungen aus den gecachten Rankings für vorhergehende Bewertungen in der Anfrage beantwortet werden. Ändern sich die Argumente für die Bewertungen der verschiedenen Informationen, z.B. weil sie über eine Variable gebunden werden, wird für die geänderten Argumente ein neues Ranking erstellt und gecacht.

4.3.4 Ebay-Implementierung

Die Ebay-Metric [EBAY] wurde in dieser Arbeit nicht als Metrik untersucht. Sie wurde dennoch implementiert, weil sie weithin bekannt ist und sich ihre Ergebnisse aufgrund der einfachen Berechnung leicht erläutern lassen. Sie steht somit als einfaches Demonstationsobjekt für die Erklärung von Metrikergebnissen bereit. Sie wird hier nicht als wissenschaftlich untersuchte Metrik eingeführt und ist Demonstrationszwecken vorbehalten.

Ebay berechnet die Reputation eines Käufers/Verkäufers aus der Differenz der positiven und negativen Bewertungen von den Transaktionspartnern in Auktionen. Um dieses Konzept zu übertragen, wurde für die Informationsgrundlagen der implementierten Metrik ein einfaches Vokabular geschaffen, das drei Aussagen für positive, neutrale und negative Bewertungen zwischen RDF Ressourcen enthält. Die Bewertungen auf dieser Grundlage sind nicht wie bei Ebay an Transaktionen gebunden und können willkürlich gesetzt werden.

Die Ebay-Metrik besitzt lediglich zwei Parameter. Der erste enthält die Identifikation der zu bewertenden Entität als URI oder Blank Node[7], der zweite ist der Schwellenwert für die minimale Bewertung, die eine vertrauenswürdige Entität haben muss. Eine Beispielanwendung wird in Abbildung 15 gezeigt.

```
...
METRIC(     metric:SemanticPageRank,
            <http://someone.net/>,
            10,
            foaf:knows)
USING metric FOR  <http://www.wiwiss.fu-berlin.de/suhl/bizer/TPL/>
```

Abbildung 15: Beispielanwendung der TriQL.P-Metrikschnittstelle für die Ebay-Metrik

4.3.5 Semantic PageRank-Implementierung

Obwohl der PageRank Algorithmus eine globale Perspektive hat (siehe 3.5), wurde er dennoch für die TriQL-Trust-Architektur implementiert. Die von Guha [ORS03] angeführten Argumente, dass globale Metriken eine gute Alternative darstellen können, wenn lokale Metriken keine Ergebnisse liefern, z.B. wenn eine Entität sich noch kein WoT aufgebaut hat, haben überzeugt. Zudem muss der Benutzer keinen Aufwand betreiben, um sein persönliches WoT zu verwalten. Aus diesen Gründen und, weil die weiteren Eigenschaften des Algorithmus nicht gegen einen Einsatz sprechen, wurde der PageRank-Algorithmus für die TriQL-Trust-Architektur implementiert.

Die Frage, die sich für eine Implementierung des PageRank-Algorithmus für Semantic Web Applikationen stellt, ist, welche Knoten und Kanten der Graphen im Semantic Web sollen berücksichtigen werden. Bei der Implementierung des

PageRank-Algorithmus für die Architektur, genannt Semantic PageRank, wurde diese Frage bewusst offen gelassen. Die Implementierung sieht vor, dass in der Argumentenliste der Metrik nach der Identität des Ziels des Vertrauens und der Anzahl der akzeptierten Entität zu Beginn des Rankings eine offene Liste für RDF Property URIs folgt. Sie werden in den RDF Aussagen als Prädikat verwendet und identifizieren die Kanten in RDF Graphen, die für die PageRank-Berechnung herangezogen werden sollen. Die Knoten des Graphen sind die Knoten an den Enden der ausgewählten Kanten. Alle aufgelisteten Properties werden durch den Algorithmus als gleich gewichtet angenommen und korrespondieren mit Hyperlinks des WWW im Google-Index. Eine typische Anwendung wird in Abschnitt 4.6 für das FOAF Property foaf:knows vorgestellt. In Abbildung 16 ist ein Beispiel für eine solche Anwendung in TriQL.P abgebildet.

```
...
METRIC(     metric:SemanticPageRank,
            <http://someone.net/>,
            10,
            foaf:knows)
USING foaf  FOR   <http://xmlns.com/foaf/0.1/>
      metric FOR  <http://www.wiwiss.fu-berlin.de/suhl/bizer/TPL/>
```

Abbildung 16: Beispielanwendung der TriQL.P-Metrikschnittstelle für SemanticPageRank

Da PageRank zu den rangbasierenden Metriken gehört, implementiert Semantic PageRank die RangBasedMetric-Schnittstelle der Architektur. Das Cachekonzept des Metrik speichert ein Rankings für eine TriQL.P-Anfrage. Da das Ranking alleine von den ausgewählten Properties abhängt, können TriQL.P-Anfragen nach der Bewertung von Entität durch die Metrik aus dem Cache beantwortet werden, bis die Liste der Properties verändert wird. Sind die Properties in der Anfrage fest definiert und nicht an Variablen gebunden, so muss pro TriQL.P-Anfrage nur ein Ranking berechnet werden. Im Beispiel in Abbildung 16 ist dies der Fall.

4.3.6 TidalTrust-Implementierung

Wie zuvor schon die Appleseed-Metrik eignet sich die von Jennifer Golbeck

vorgestellte TidalTrust-Metrik ebenfalls gut für eine Implementierung in der TriQL-Trust-Architektur. Wie bereits in 3.7 beschrieben, erfüllt sie alle aufgestellten Kriterien für den Einsatz in Applikationen des Semantic Web. Als schwellenwertbasierende Metrik implementiert sich die einfache Metric-Schnittstelle der Architektur.

Die Implementierung der TidalTrust-Metrik verwendet für die Formulierung sozialer Vertrauensnetzwerke wie die Appleseed-Metrik das Vokabular der Trust-Ontology [MSTO] des „Trust and Reputation“ Projektes [MSTR] von Jennifer Golbeck. Auf der Basis der gewichteten Kanten inferiert die Metrik eine virtuelle gewichtete Kante zwischen der Quelle und dem Ziel des Vertrauens. Die ersten beiden Parameter der Metrik bestimmen daher die Identitäten dieser beiden Entitäten. Durch die Verwendung des SWP Vokabulars für die Feststellung der verantwortlichen Entitäten eines Quellgraphen, wird bei der Metrikausführung berücksichtigt, das nur Bewertungen verwendet werden, in denen der Veröffentlicher auch als Subjekt auf tritt. Dadurch werden nur Bewertungen des Veröffentlicher selbst über andere Entitäten anerkannt.

Die Implementierung der TidalTrust-Metrik kennt drei Argumente. Die ersten beiden Argumente enthalten die Identitäten der Quelle und des Ziels des Vertrauens zwischen denen eine virtuelle gerichtete und gewichtete Kante berechnet werden soll. Die Gewichte der Kanten des sozialen Bewertungsnetzwerkes werden durch die Implementierung auf den Wertebereich [0..1] normalisiert. Das dritte Argument, welches den Schwellenwert angibt, den das Gewicht der virtuellen Kante übersteigen muss, damit das Ziel eine positive Bewertung erhält, liegt daher im selben Wertebereich. Eine Beispielanwendung der Metrik findet sich in Abbildung 17 für einen Schwellenwert von 0,5.

```
...
METRIC(     metric:TidalTrust,
            <http://source.net/>,
            <http://sink.net/>,
            0,5)
USING metric FOR  <http://www.wiwiss.fu-berlin.de/suhl/bizer/TPL/>
```

Abbildung 17: Beispielanwendung der TriQL.P-Metrikschnittstelle für die TidalTrust

4.4 Erklärung der Metrikergebnisse

Damit Benutzer tatsächlich Vertrauen in die inferierten Ergebnisse entwickeln, ist das grundlegende Verständnis der verwendeten Informationen und der verwendeten Mechanismen notwendig [TAOW04]. Nur durch die Nachvollziehbarkeit der Inferenzen wird es einem Benutzer möglich, die Informationen, die mit einer hohen Informationsqualität bewertet wurden, für seine Entscheidungsprozesse zu verwenden. Vor allem im professionellen kommerziellen Umfeld müssen Entscheidungsgrundlagen erklärbar sein, um Verantwortlichen die Möglichkeiten zu einer eigenen Vertrauen schaffenden Bewertung zu geben.

Die TriQL-Trust-Architektur wird für menschliche Nutzer entwickelt [CCBT04]. Natürlichsprachliche Erklärungen der Ergebnisse rücken daher ins Zentrum der Betrachtung. Dieser Abschnitt soll sich im weiteren mit einem Konzept für die Generierung von Erklärungen in Trust-Metriken der TriQL-Trust-Architektur befassen. Zuerst werden die Grundlagen vorgestellt, die die anschließende Implementierung bestimmen sollen.

4.4.1 Grundlagen

Vertrauen in die Ergebnisse von Schlussprozessen ist von einer Vielzahl an Faktoren abhängig. Um Entscheidungen zu treffen, müssen Nutzer gute Kenntnisse über die Applikationen besitzen, die ihnen die Entscheidungsgrundlage beschaffen sollen, und wissen die Ergebnisse anzuwenden. Entscheidungen werden unter der Berücksichtigung der Informationsqualität der Basisdaten, des verwendeten Schlussverfahrens und der Umstände der Entscheidung gefällt [EASW04]. Für die Anwendung von Trust-Metriken bedeutet dies, dem Benutzer zu verdeutlichen, warum die Metrik zu dem Schluss kommt, dass eine Person

vertrauenswürdig, ein Dokument glaubwürdig oder ein Service von hoher Qualität ist.

Neben der Vertrauensbildung beim Benutzer selbst werden Erklärungen für die Verteidigung der Inferenzen gegenüber Dritten benötigt. Ergebnisse müssen oft anderen erklärt werden. Auch Aktualisierungen der Ergebnisse hinsichtlich einer neuen oder geänderten Informationsbasis müssen nachvollziehbar sein. Detailierte Erläuterungen der Zwischenprodukte in Schlussprozessen helfen ähnliche Schlüsse zu ziehen. Mit TELLIS stellten Gil Yolanda und Ratnakar Varun [TICT02] ein System vor, dass diese Aufgaben unterstützen soll. Mit Annotationen über berücksichtigte Quellen im Schlussprozess soll der gesamte Schlussprozess transparenter werden. Die Annotationen werden wiederum dazu verwendet, die Quellen zu bewerten.

Deborah McGuinness und Paulo Pinheiro da Silva sehen in Metadaten über die Herkunft von Information eine der wichtigsten Informationen für die Erklärung von Schlussprozessen. Oft befriedigen sie schon die Erklärungsbedürfnisse von Benutzern [TAOW04]. Insbesondere wenn der verwendeten Applikation und ihren Inferenzmechanismen zugetraut wird, effektive Bewertungen zu inferieren, bleiben Informationen über die Informationsquellen als wichtiges Merkmal einer Inferenz.

In Systemen, in denen Schlüsse oder Zwischenprodukte für weitere Schlussverfahren weiterverwendet werden sollen, ist die Austauschbarkeit der Inferenzprotokolle über Systemgrenzen hinaus von besonderer Bedeutung. Der Inference Web Ansatz [EASW04] des „Knowledge Systems, Artificial Intelligence Laboratory“ der Standford University stellt ein Framework bereit, welches sich unter anderem das Speichern, den Austausch, die Rekombination und die Darstellung von regelbasierenden Schlüssen aus dem Semantic Web zum Ziel gesetzt hat. Neben Informationen über die Informationsquellen werden viele weitere Merkmale aus Schlussverfahren festgehalten. Durch die Identität der schließenden Instanz sowie die verwendeten Regeln, Methoden und deren Implementierungen können sich Benutzer überzeugen lassen von den Ergebnissen und

sie nach Maßgabe inklusiv ihrer Protokolle weiterverwenden. Zudem lassen sich Schlüsse aus verschiedenen Schlussprozessen im Framework vergleichen.

Bei der Akzeptanz geht es im wesentlichen, um die Nachvollziehbarkeit von Ergebnissen. Verschiedene Anwender haben unterschiedliche Vorstellungen von den nötigen Erklärungen. Inferenzsysteme, die Ergebnisse für die Verwendung in weiteren Inferenzen aufbewahren und veröffentlichen, werden möglichst alle Informationen bis ins kleinste Detail speichern, da der Kontext in dem die Ergebnisse weiterverwendet werden, unbekannt ist. Zudem müssen die Informationen in einem allgemein akzeptierten Austauschformat veröffentlicht werden. Sowohl TELLIS als auch das Inference Web entwickeln aus dieser Notwendigkeit ein eigenes RDF Vokabular.

Die Protokolle von Schlussverfahren sind häufig komplex und abstrakt, und müssen daher für einen menschlichen Nutzer vereinfacht werden [EASW04]. Für unerfahrene Benutzer wird es in der Regel sinnvoll sein, eine möglichst einfache Erläuterung zu generieren, da sie sich von zu vielen Details eher verwirren lassen. Ein erfahrener Nutzer wird hingegen mehr Details verlangen, um sich von der Glaubwürdigkeit einer Inferenz überzeugen zu können. Das Niveau, auf dem Benutzer nach Erklärungen verlangen, variiert also. Dennoch muss es möglich sein bei weiteren Fragen nach Details auf diese zu reagieren.

Der im Inferenz Web gewählte Ansatz enthält zwei Komponenten. Zum Einen wird von den vollständigen Protokollen der Schlussprozesse abstrahiert, um verschiedene Niveaus an Details bereitzustellen. Zum Anderen wird dem Nutzer nur ein kleiner Ausschnitt der gesamten Information präsentiert, die durch Nachfragen immer weiter verfeinert werden kann [EASW04]. Wie viele Informationen über eine Inferenz präsentiert werden, hängt also von der Erfahrung des Benutzers ab. Der Kontext, in dem eine Inferenz geprüft wird, nimmt entscheidend Einfluss auf die Granularität der zu überprüfenden Details.

4.4.2 Erklärungen für die TriQL-Trust-Architektur

Die Anforderungen der TriQL-Trust-Architektur sehen die Erklärung von Inferenzen von vornherein vor [CCBT04]. Die Trust-Policiy Language (TPL), in

der Graphmuster und Metriken zu Trust-Policies zusammengestellt werden, wurde explizit dafür ausgelegt. So lässt sich jedes Graphmuster mit einer natürlich sprachlichen Erklärung seiner Bedeutung versehen. Im Unterschied zu Graphmustern sind die zugrunde liegenden Graphstrukturen von Metrikinferenzen nicht statisch festgelegt. Sie variieren mit den Eingabeparametern, so dass sie die Erklärungen von Metrikinferenzen komplexer gestaltet.

Um zu einem akzeptablen Konzept für die Erklärung von Metrikinferenzen zu kommen, wurde zunächst das Einsatzszenario geprüft. Das Usecase der TriQL-Trust-Architektur sieht vor, das ein einzelner Benutzer, auf der Basis der Daten, die er mit dem *Information and Integration Layer* der Architektur auf seinem eigenen Rechner gesammelt hat, glaubwürdige Informationen herausfiltert. Er wendet dazu Trust-Policies an, die er an seine persönlichen Bedürfnisse und Situationen angepasst hat. Es ist sogar möglich, eigene neue Metriken einzubinden, so dass davon ausgegangen werden kann, dass den Implementierungen der Metriken vertraut wird. Eine Weitergabe von Ergebnissen für weitere Inferenzen ist nicht expliziter Bestandteil des Usecases. Die Austauschbarkeit der Informationen mit anderen Systemen steht daher nicht weiter im Mittelpunkt und wird zudem durch die starke Personalisierung durch lokale Metriken und eine personalisierte Auswahl an Informationen im *Repository Layer* erschwert. Die Erklärungen werden nur innerhalb der Architektur, zwischen den Layern der Architektur im RDF Format ausgetauscht.

Im Zentrum des Usecases stehen daher Erklärungen für den Endbenutzer. Die Generierung wird daher auf natürlichsprachliche Erklärungen beschränkt. Da sich Unsicherheiten durch die berechnende Instanz durch das Usecase ausschließen lassen, werden die wesentlichen Informationen bestimmt durch die Informationsquellen und den Bewertungsprozess der Metrik.

4.4.2.1 Informationsquellen von Metrikinferenzen

Als wichtigste Information für den Nutzer bleibt, die Herkunft der gesammelten Informationen. Im heutigen WWW und Semantic Web können die Autoren von Inhalten nur schwer verfolgt werden. In manchen Internetseiten sind

zwar Metainformationen über Autoren enthalten, allerdings lassen sich diese Aussagen kaum formal verifizieren.

Das Semantic Web Publishing (SWP) RDF Vokabular soll hier Abhilfe schaffen. Die Metriken gehen davon aus, dass alle Inhalte des Semantic Web in Named Graphs (NG) abgelegt sind. Durch das Warrant Konzept des SWP wird somit die Herkunft von Informationen prüfbar (siehe 4.1.3). Mit der Überprüfung von Signaturen auf den Graphen wird nicht nur die Lokalität der Informationen, sondern auch die Verantwortlichkeit für Inhalte klar dargestellt. Schlüsse zwischen der Vertrauenswürdigkeit eines Autors und der Informationsqualität seiner Veröffentlichungen werden so möglich.

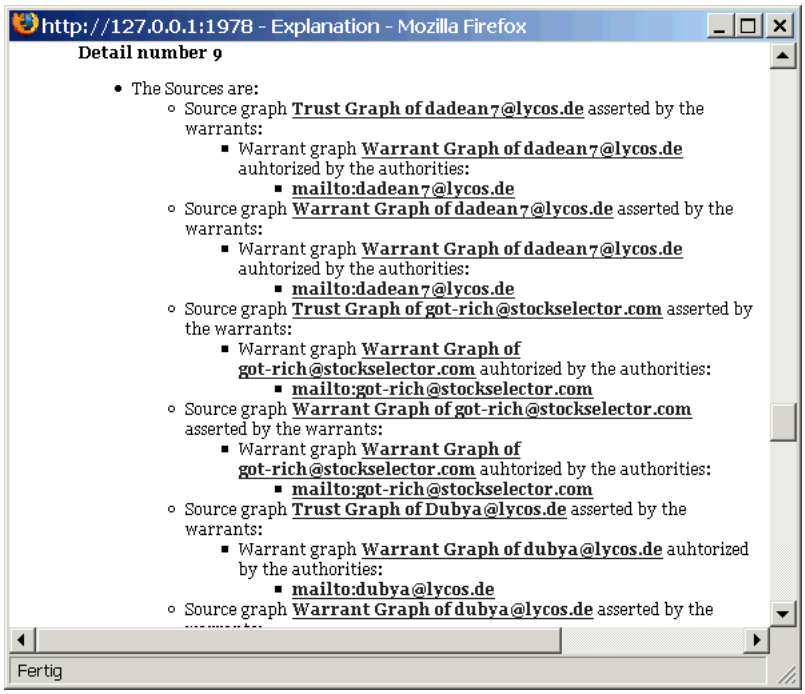

Abbildung 18: Quellenangabe einer Metrikinferenz

Die Metriken prüfen vor der Verwendung selbständig die Identität von Autoritäten und binden sie zu einer Zusammenfassung der Quellen zusammen. Die Quellenangaben enthalten also die Quellen, ihre verantwortlichen Autoritäten und die Warrants die beide zueinander in Beziehung setzen. Ein Beispiel für eine solche Quellenangabe zeigt Abbildung 18 in einer Darstellung des TriQL.P-Browser. Alle implementierten Metriken weisen ein solche Zusammenfassung

auf. Hinter den dargestellten Namen verbergen sich URIs zu den entsprechenden RDF Ressourcen, die die Autoritäten, Warrants und Graphen repräsentieren.

4.4.2.2 Erklärung des Bewertungsprozesses

Während die Quellenangaben noch sehr einheitlich implementiert sind, laufen die Formen der Erklärungen des Bewertungsprozess auseinander. Es gibt kein einheitliches Konzept wie der Berechnungsprozess von Metriken erklärt werden kann.

In der Abstraktion lassen sich Metrikinferenzen als Schlüsse mittels vieler kleiner Ableitungen beschreiben. Die Menge an Ableitungsregeln und die Anwendungsreihenfolge sind dabei durch die Metrik terminiert. Auf diese Weise könnten Metrikinferenzen mit vielen kleinen Ableitungen erklärt werden. Die große Menge an Ableitungen würde allerdings die Erklärungen zu einer nicht mehr handhabbaren Datenmenge anwachsen lassen.

Um die Erklärungen in der TriQL-Trust-Architektur handhabbarer zu machen, wird von den einzelnen Ableitungen abstrahiert, in dem sie zu komplexeren Operationen zusammengefasst werden. Diese Operationen werden als atomar betrachtet. Auf dieser Basis lassen sich die Bedeutungen von Rechenoperationen beschreiben. Die Erklärungen dieser Operationen bestimmen die feinste Granularität der Details einer Erklärung.

Obwohl die Bewertungsprozesse der Metriken sehr unterschiedlich sind, können doch für die verschiedenen Metrikeigenschaften Aussagen über wichtige Bestandteile der Erklärung gemacht werden. Rangbasierende Metriken wie die implementierte Appleseed- oder PageRank-Metrik sollten das Ranking selbst enthalten. Zudem sollte jede bewertete Entität zusammen mit ihrem inferierten Wert für die Ordnung des Rankings angegeben sein. Die Ränge und die Bewertungen werden somit prüfbar.

Algorithmen, die auf der Annäherung eines Eigenvektors basieren, sollten diesen ebenfalls in die Erklärungen aufnehmen. Wird der Vektor direkt für ein Ranking verwendet wie bei PageRank, so würde dies mit der Auflistung des Rankings mit Bewertungen bereits realisiert. Die Angabe von Zwischenprodukten

des Eigenvektors über die Iterationen im Annäherungsverfahren ist wenig sinnvoll, da nur entscheidend ist, ob der inferierte Vektor tatsächlich ein Eigenvektor ist. Dies lässt sich mit dem Endprodukt alleine prüfen.

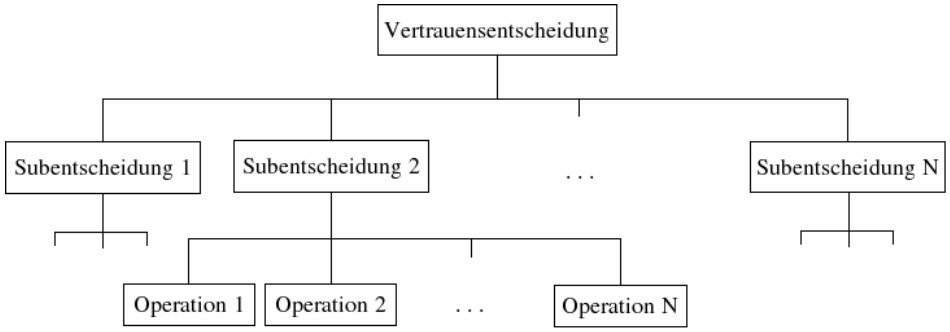

Abbildung 19: Entscheidungsstruktur eines Metrikprozesses

Pfadbasierende Metriken wie die TidalTrust-Metrik sollten prinzipiell die verwendeten Pfade aufzeigen. Benutzer können so selber überprüfen, ob diese existieren. Werden durch die Metrik Gewichte entlang der einzelnen Kanten in den Pfaden berücksichtigt, so sollten die Gewichte ebenfalls in den Erklärungen enthalten sein.

Entscheidungen über die Auswahl von verschiedenen Alternativen sollten immer dokumentiert werden. Hierzu sollte nicht nur das Ergebnis der Auswahl festgehalten werden, sondern auch die Parameter, die der Entscheidung zugrunde liegen. Im Bewertungsprozess der TidalTrust-Metrik werden nur Pfade ausgewählt, die der minimalen Pfadlänge entsprechen. Es sollte also ebenfalls belegt werden, das die ausgewählten Pfade tatsächlich diese Länge besitzen.

Um verschiedene Detaillevels zu unterscheiden, wurde der Baum der einzelnen Schlüsse in der Metrikausführung aufgeschlüsselt (siehe Abbildung 19). Auf oberster Ebene wird daher immer zuerst die positive oder negative Entscheidung hinsichtlich der Bewertungskriterien präsentiert. Jeder Schluss verfügt über weitere Details, die wiederum einzelne Schlüsse im Auswertungsprozess oder die Ergebnisse von atomaren Operationen sind. Die Quellenangaben werden immer auf der zweiten Ebene des Entscheidungsbaums dargestellt, damit sie schnell zugänglich sind.

In Abbildung 22 auf Seite 98 ist eine Erklärung eines positiv bewerteten

Triples im TriQL.P-Browser dargestellt. Im mit „because: ...“ überschriebenen Abschnitt beginnt die eigentliche Erklärung der Metrikinferenz. Während im ersten Punkt noch die Inferenz aus einem Graphmuster der verwendeten Trust-Policy erläutert wird, stellt der zweite Punkt das Ergebnis der TidalTrust-Inferenz klar. Sie besagt, das den Privatebankies Merck und Fink vertraut wird. In der Darstellung des TriQL.P-Browsers kann man nun über ein Link zum ersten Detail der Erklärung springen, welches sich in Abbildung 22 direkt anschließt. Es werden die Grundlagen für die Vertrauensentscheidung erläutert, die selber Entscheidungen sind. Über weitere Links kann man wiederum deren Grundlagen einsehen. Als letztes Detail des „Detail number 1“ steht die Zusammenfassung der Quellenangaben, in deren Details wiederum ein Liste wie in Abbildung 18 zu finden ist.

4.5 Anwendungsbeispiel Finanzdatenbeispiel mit TidalTrust

Für die Demonstration der TriQL-Trust-Architektur wurde ursprünglich ein Szenario aus dem Finanzbereich gewählt, in dem das Bewerten von Unternehmen von hoher Bedeutung ist. Tobias Gauß entwickelte zu diesen Zweck eine Ontologie für Unternehmensinformationen, Unternehmensbewertungen und Nachrichten. Der Datensatz ist unter [TPWB] veröffentlicht. Die Ontologie wurde weiterentwickelt, um mit zusätzlichen Bewertungsinformationen über Unternehmen, Autoren und einfache Personen, für die Demonstration der implementierten Trust-Metriken geeignet zu sein.

Als Usecase für dieses Anwendungsbeispiel wird hier ein Informationskonsument mit der Identifikation durch seine Email (dadean7@lycos.de) angenommen, der mit seinem Firefox-Browser und dem „TriQL Trust Policies Enabled Semantic Web Browser“ Plugin im Semantic Web nach Finanzinformationen sucht. Da er die Qualität der Nachrichten nicht selber prüfen kann, möchte er sicherstellen, dass er nur Nachrichten von Autoren erhält, die eine positive Reputation besitzen. Da unser Beispielbenutzer die Autoren der Artikel selbst nicht kennt, benötigt er eine Methode, um die Autoren bewerten zu lassen. Er entscheidet sich vertraute Personen in die Bewertung einzubeziehen, also ein

soziales Netzwerk zu befragen.

Da unser Benutzer öfters nach Informationen sucht, deren Qualität er nicht selbst beurteilen kann, hat er bereits eine Trust-Policy formuliert, die seinen Bedürfnissen gerecht wird. In Abbildung 20 ist diese Trust-Policy dargestellt. Zu Beginn erhält die Trust-Policy einen Namen und eine Beschreibung, damit sie im Browser wiedererkennbar ist. Anschließend ist eine Vorlage für die Erklärung des einzigen Graphmusters formuliert. Das Graphmuster fragt nach dem Autor einer Information, die mit der Systemvariable ?SUBJ im Ergebnis der Orginalanfrage referenziert wird. Die Variable ?author wird in der Erklärung verwendet, um die Vorlage mit dem entsprechenden Namen zu ergänzen. Das Graphmuster stellt sicher, das der Autor einer Information bekannt ist.

```
policies:Policy8  rdf:type    tpl:TrustPolicy ;
      tpl:policyName "TidalTrust Value above 0.5" ;
      tpl:policyDescription "Trust information authored by sources
            with a TidalTrust trust value above 0.5." ;
      tpl:textExplanation "It was stated by @@?author@@" ;
      tpl:graphPattern [
            tpl:pattern """ ?GRAPH (?SUBJ fin:author ?author )
                  """ ;
      ] ;
      tpl:constraint "METRIC(tpl:TidalTrustMetric, ?USER, ?author,
      0.5)" .
```

Abbildung 20: Trust-Policy: TidalTrust Value above 0.5

Nun muss der Autor noch bewertet werden. Für die Bewertung von Entitäten über ein soziales Netzwerk wurden mit Appleseed und TidalTrust zwei Metriken implementiert, die für eine Bewertung in Frage kommen. Aus Gründen der einfacheren Handhabung der Parametrisierung wird hier die TidalTrust-Metrik verwendet. Sie ist schwellenwertbasiert und besitzt eine lokale Perspektive. Unser Benutzer lässt das Vertrauen, welches er in einen Autor haben soll, mit der TidalTrust-Metrik über seinem persönlichen WoT bewerten. Er hängt daher die TidalTrust-Metrik an die Trust-Policy an, und parametrisiert sie mit sich selbst durch die Systemvariable ?USER als Quelle des Vertrauens, dem Autor als Ziel des Vertrauens und einem Schwellenwert für die minimale Bewertung, die ein

vertrauenswürdiger Autor bekommen muss. Mit der Verwendung der Metrik, werden also nur noch Informationen angezeigt, deren Autoren bekannt sind und eine TidalTrust-Bewertung von 0,5 aus der Sicht von dadean7@lycos.de bekommen.

Die Sammlung von Informationen kann jetzt beginnen. Im Beispiel wurde zu diesem Zweck die Finanzdaten-Ontology mit den Bewertungsinformationen geladen.

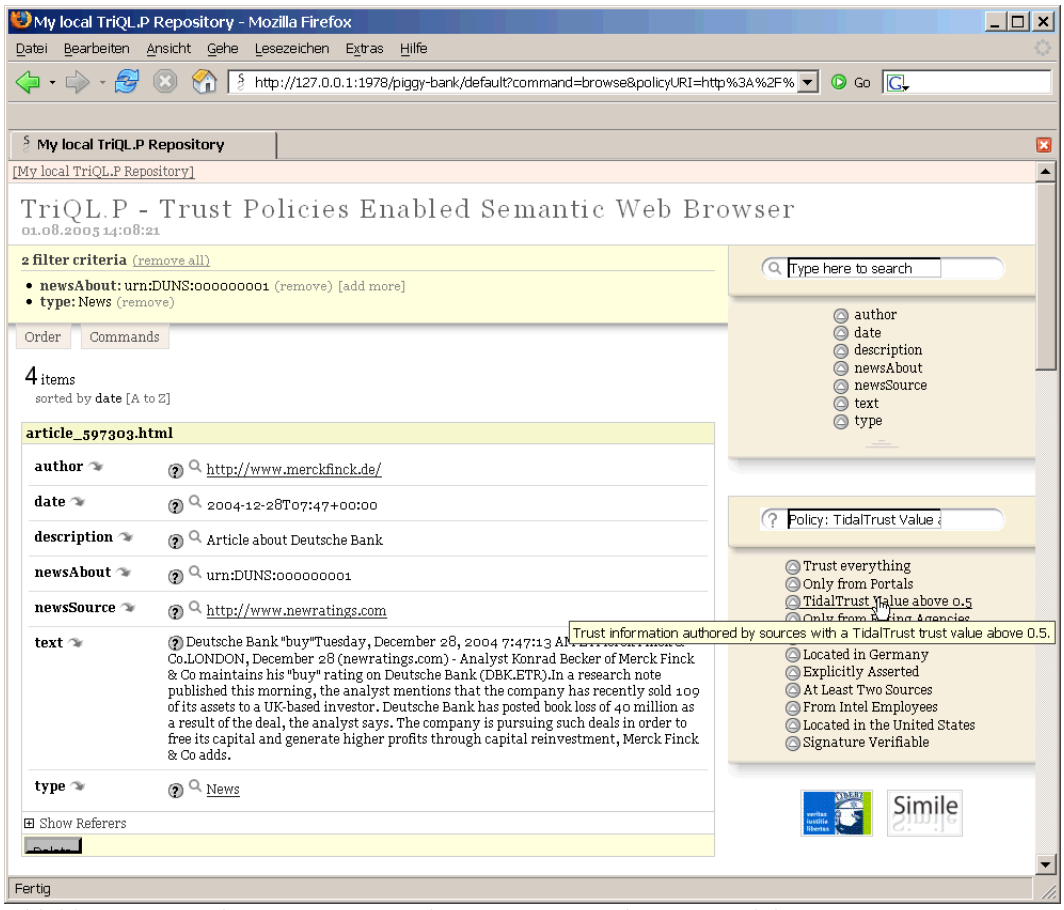

Abbildung 21: Ergebnis einer Trust-Policy unter Verwendung von TidalTrust

Mit dem „TriQL.P Trust Policies Enabled Semantic Web Browser“ wählt unser Benutzer also die gewünschte Trust-Policy aus. Alle Informationen, die vom Benutzer nun ausgewählt werden, werden anhand der Policy überprüft. Ergibt die Prüfung, dass der Autor der Information bekannt und er vertrauenswürdig ist, so wird die Information angezeigt. Andernfalls wird sie heraus gefiltert. Der Benutzer kann über die Benutzerschnittstellen nach Nachrichten suchen, z.B. in dem er nach Informationen vom Typ „News“ sucht. Die Ergebnisse der Suche

sind jeweils durch die Policy gefiltert. In Abbildung 21 ist das Ergebnis einer solchen Suche auf dem Beispieldatensatz für Finanzinformationen dargestellt. Der Benutzer sucht in der Abbildung nach Informationen über die Entität urn:DUNS:000000001 vom Typ „News“ (zuerkennen im Abschnitt „filter criteria“ in Abbildung 21). Die erste von vier Informationen, die durch die Trust-Policy herausgefiltert wurden, wird präsentiert. urn:DUNS:000000001 identifiziert im Beispieldatensatz die Deutsche Bank.

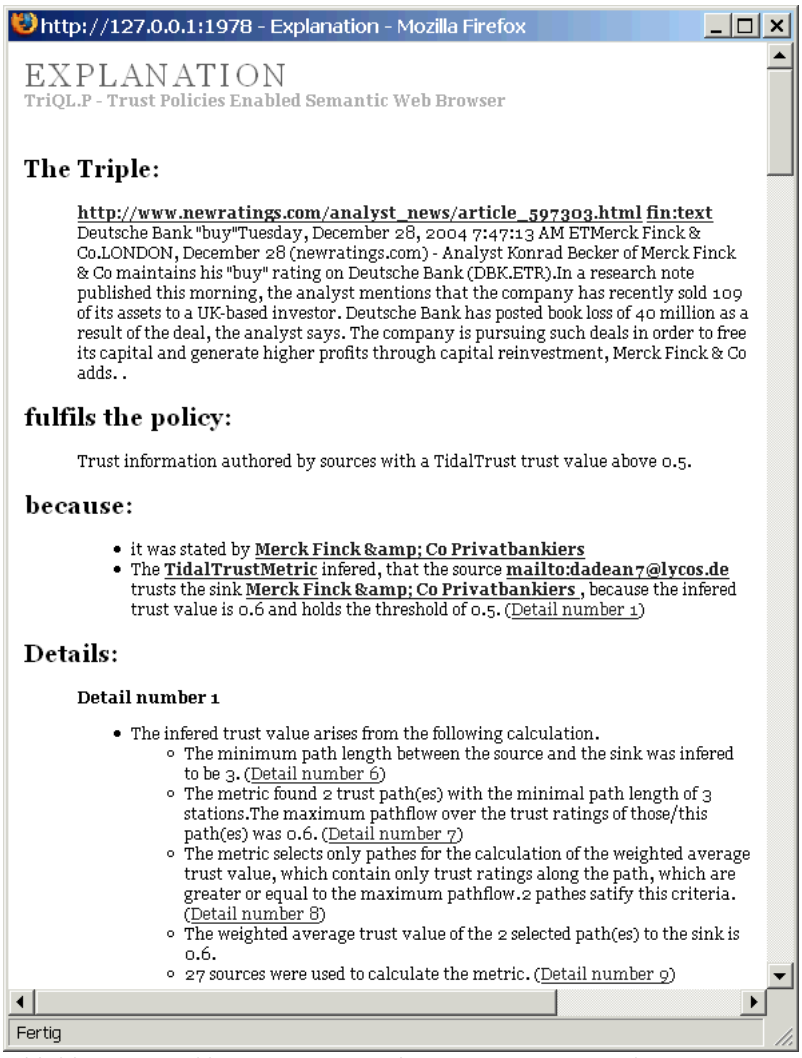

Abbildung 22: Erklärung eines Ergebnisses unter Verwendung von TidalTrust

Über den Erklärungsbutton kann der Benutzer die Erklärung für die Evaluierung der Trust-Policy überprüfen. Wie in Abbildung 22 zu sehen ist, wurde die Nachricht von den Privatebankiers Merck Fink veröffentlicht. Die TidalTrust-Metrik bewertete sie mit einer Vertrauensbewertung von 0,6, welche klar die

definierte Schranke von 0,5 einhält. In der ersten Detailerläuterung des Metrikergebnisses, werden die Einzelentscheidungen der Metrik aufgeschlüsselt. Es wird beschrieben, dass gemäß der Metrik die minimale Pfadlänge 3 beträgt, es zwei Pfade gibt die diese Pfadlänge besitzen und beide das Maximum-Pathflow Kriterium erfüllen. In weitern Details können die einzelnen Pfade inspiziert werden (hier nicht dargestellt). Nachdem das Ergebnis der Metrik noch einmal wiederholt wird, sehen wir die Kurzzusammenfassung der Quellen, in deren Details sich alle verwendeten Quellgraphen wie deren Veröffentlicher in der in Abbildung 18 vorgestellten Form befinden.

4.6 Anwendungsbeispiel FOAF:knows mit Semantic PageRank

Mit der zweiten Beispielanwendung soll die Anwendung der Semantic PageRank-Metrik als rangbasierende und globale Metrik demonstriert werden. Für die Demonstration wurde die Aufgabe gestellt, aus dem bekannten FOAF Netzwerk [FOAF] die bekanntesten Persönlichkeiten zu ermitteln. Zu diesem Zweck sollte das Netzwerk aus foaf:knows Aussagen durch den PageRank-Algorithmus analysiert werden. Das Prädikat foaf:knows verbindet zwei Personen miteinander zu einer Aussage der Form „Oliver“ „kennt“ „Chris“. Die verwendeten Daten wurden aus einer Datenbasis extrahiert, die Andreas Harth im Rahmen seines SECO Projektes [SECO04] aus dem Semantic Web gesammelt hat. Sie integriert viele verschiedene Datenquellen des Semantic Web unter anderem auch Dateien im FOAF-Format, mit denen Personen ihre Beziehungen zu anderen ausdrücken können.

Leider konnte für die Bewertung mit Semantic PageRank nicht die gesamte FOAF Datenbasis verwendet werden. Zum einen lassen Performanzprobleme der TriQL-Trust-Architektur den Import der gesamten Daten nicht zu. Zum anderen enthielt die extrahierte Datenbasis eine große Menge foaf:knows-Aussagen, deren Objekte lediglich Blätter im Netzwerk darstellen, d.h. die selber nicht Subjekt in foaf:knows-Aussagen sind. Zudem wurde die Auswahl der Personen auf jene beschränkt, die eine foaf:name Aussage mit ihrem Namen enthielten, da dies für eine geeignete Anzeige im Browser notwendig ist. Die gesamten FOAF Daten

wurden also auf jene Personen beschränkt, die im Netzwerkgraphen kein Blatt darstellten und eine foaf:name Aussage besitzen. Die verbleibenden 136 Personen stellen also nur einen kleinen Teil des FOAF Netzwerkes dar. Das Ergebnis der Metrik ist also aus Gründen der geringen Qualität der Datenbasis nicht repräsentativ.

Ein weiters Problem, das sich bei der Filterung der Daten herausstellte, sind die von Friedman und Resnik diskutierten „Cheap Pseudonyms“ [SCCP01]. In den FOAF Daten zeigt sich sehr deutlich, das die für die Identifikation der Personen verwendeten URIs als Pseudonyme problematisch sind. Viele Personen werden im Netzwerk durch verschiedene URIs identifiziert, so dass sie anfänglich mehrfach im Ranking der Semantic PageRank-Metrik auftauchten. Dieses Problem wurde für den verwendeten Datensatz behoben.

```
:Policy1 rdf:type tpl:TrustPolicy ;
      tpl:policyName "Most popular FOAF Person" ;
      tpl:policyDescription "Trust the 10 most popular Person in
            the FOAF network by PageRank" ;
      tpl:textExplanation "The Semantic PageRank Algorithm
            inferred, that you should trust @@?SUBJ@@, because
            he/she was ranked in the Top-Ten of the foaf:knows
            network." ;
      tpl:constraint "METRIC(tpl:SemanticPageRank, ?SUBJ, 10,
            foaf:knows) " .
```

Abbildung 23: Trust-Policy: Most popular FOAF Person

In Abbildung 23 befindet sich die Trust-Policy, die verwendet werden soll, um die 10 populärsten FOAF Personen zu ermitteln. Sie enthält nur eine Erklärungsvorlage und eine Metrikausführung. Die Semantic PageRank-Metrik wird dabei mit dem Subjekt der Suchergebnisse als Ziel des Vertrauens parametrisiert. Sie muss sich in den Top 10 des untersuchten Netzwerkgraphen befinden. Der Netzwerkgraph wird durch das foaf:knows Prädikat spezifiziert. Da die Architektur keine Inferenzen hinsichtlich der Verfeinerungen des foaf:knows Prädikats ausführt, könnten die Parameterliste um jene verfeinerten Prädikate erweitert werden, um so ein größeres Netzwerk zu durchsuchen. Im Beispiel wird das

Netzwerk auf foaf:knows-Aussagen beschränkt.

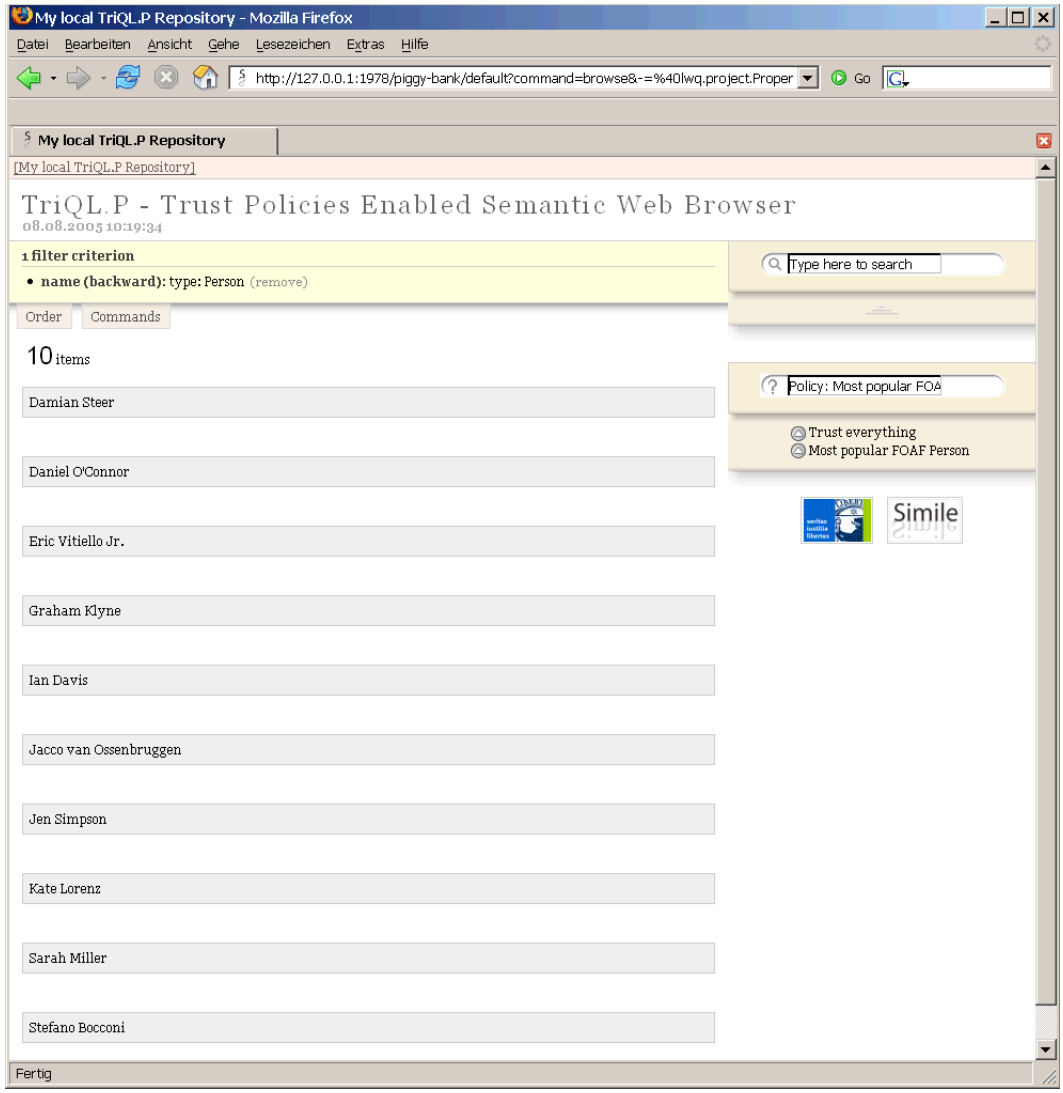

Abbildung 24: Ergebnis einer Trust-Policy unter Verwendung von Semantic PageRank

Durch die Auswahl der Trust-Policy „Most popular FOAF Person“ wird die vorgestellte Trust-Policy aktiviert. Wenn der Benutzer nun nach Personen im Repository sucht, z.B. in dem er Informationen vom Typ Person auswählt, werden nur Personen angezeigt, die sich in den Top 10 des FOAF Netzwerkes nach PageRank befinden. In Abbildung 24 ist das Ergebnis für den Beispieldatensatz angezeigt. Wenn wir uns die Erklärung für eine Person im Suchergebnis anschauen (Abbildung 25), so wird das Ranking präsentiert in dem die Person bewertet ist. Für diese einfache Trust-Policy, in der die Metrik das einzige Auswahlkriterium darstellt, können wir sogar erkennen, das alle Personen in den

Top 10 sich auch im Gesamtergebnis des „TriQL.P Trust Policy Enabled Semantic Web Browsers“ befinden..

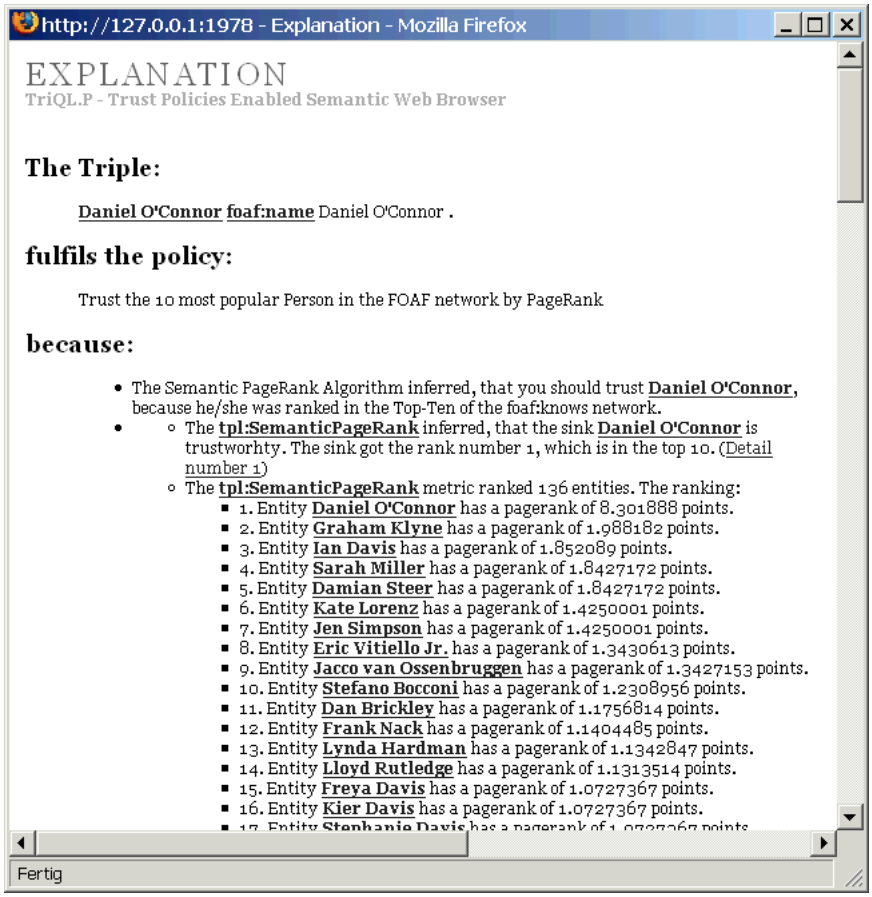

Abbildung 25: Erklärung eines Ergebnisses unter Verwendung von Semantic PageRank

5 Zusammenfassung

In der Literatur sind bereits viele reputationsbasierende Metriken aus verschiedenen Systemen bekannt. Die TriQL-Trust-Architektur als eine Anwendung des Semantic Web ist ein System, welches reputationsbasierende Metriken wiederverwenden soll, um die Bewertung von Informationsqualität zu unterstützen. Doch welche Metriken eignen sich für den Einsatz in Anwendungen des Semantic Web.

In dieser Diplomarbeit wird eine Klassifikation für die Beurteilung von reputationsbasierenden Systemen vorgestellt, um deren Eigenschaften und Eignungen beurteilen zu können. Die analysierten Dimensionen der Klassifikation trennen dabei Eigenschaften der reputationsbasierenden Metriken der Systeme von Eigenschaften des Umfelds. So ergeben sich Dimensionen für die Algorithmen auf der einen Seite und Dimensionen für die Datenhaltung und -erstellung auf der anderen Seite.

Um für den Einsatz im Semantic Web geeignete Metriken identifizieren und die Eignung der Klassifikation für reputationsbasierende Systeme zu zeigen, wurde die Klassifikation auf ein Feld von sieben reputationsbasierenden Systemen angewandt. Durch den Abgleich der Eigenschaften der Metriken mit denen des Semantic Web sowie der TriQL-Trust-Architektur wurden geeignete Metriken identifiziert und für die Architektur implementiert.

5.1 Bewertung

Mit der Klassifikation für reputationsbasierende Systeme steht nun ein Werkzeug zur Verfügung, um die Wiederverwendung von reputationsbasierenden Metriken in neuen Systemen zu erleichtern. Durch die Erkenntnis der Eigenschaften von Metriken und Systemen, können begründete Entscheidungen für oder wieder der Verwendung der einzelnen reputationsbasierenden Metriken getroffen werden.

Die Anwendung der Klassifikation für die Einordung von reputationsbasierenden Systemen auf sieben Metriken, führte zu einer geeigneten Auswahl von Trust-Metriken für den Einsatz im Semantic Web. Für die TriQL-Trust-Architektur konnten Metriken ausgewählt werden, die den Bedürfnissen der Architektur

und der Usecases gerecht werden, um die Bewertung von Informationsqualität im Semantic Web mit reputationsbasierenden Metriken zu unterstützen. Die TriQL-Trust-Architektur wurde somit zu einem reputationsbasierenden System.

5.2 Ausblick

Für die Beurteilung von reputationsbasierenden Trust-Metriken für das Semantic Web konnte die Klassifikation angewandt werden. Es bleibt zu zeigen, das die Klassifikation nicht nur für die Beurteilung von Metriken für das Semantic Web taugt, sondern auch für Systeme mit anderen Anforderungen, wie z.B. P2P-Systemen.

Die Eignung der ausgewählten Metriken konnte bisher nur für kleine Beispieldatensätze demonstriert werden. Es bleibt zu zeigen, ob die Metriken sich im praktischen Einsatz auf Daten des Semantic Web mit vielen verteilten Informationsquellen bewähren. Zudem bleibt das Feld der reputationsbasierenden Trust-Metriken in Bewegung und wird neue Metriken hervorbringen, die den Ansprüchen des Semantic Web gerecht werden.

Anhang A: Abkürzungsverzeichnis

FOAF	Friend of a Friend
NG	Named Graphs
P2P	Peer-to-Peer
PKI	Public Key Infrastruktur
QoS	Quality of Service
RDF	Resource Description Framework
RDQL	RDF Data Query Language
SWP	Semantic Web Publishing
TPL	Trust Policy Language
WoT	Web of Trust
WWW	World Wide Web
W3C	World Wide Web Consortium
URI	Uniform Resource Identifier
URL	Uniform Resource Locator

Anhang B: Literaturverzeichnis

[ALSS98] : Brin, Sergey; Page, Lawrence. The Anatomy of a Large-Scale Hypertextual Web Search Engine. In Computer Networks and ISDN Systems, Vol. 30, Issue 1-7, 1998.

[AMIT04] : Golbeck, Jennifer; Hendler, James. Accuracy of Metrics for Inferring Trust and Reputation in Semantic Web-Based Social Networkd. In Proc. 14th International Conference on Knowledge Engineering and Knowledge Management (EKAW), 2004.

[ARTM04] : Levien, Raphael. Attack Resistant Trust Metrics. (Ph. d. Thesis) 2004.

[CCBT04] : Bizer, Christian; Oldakowski, Radoslaw. Using Context- and Content-Based Trust Policies on the Semantic Web. In Proc. 14th International World Wide Web Conference, 2004.

[CMTR04] : Lik Mui. Computational Models of Trust and Reputation: Agents, Evolutionary Games. (Ph. d. Thesis) 2003.

[CPGN03] : Bhattacharjee, Bobby; Lee, Seungjoon; Sherwood, Rob. Cooperative Peer Groups in NICE. In Proc. 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), 2003..

[CSTF04] : Zhang, Qing; Yu, Ting, Irwin, Keith. A Classification Scheme for Trust Functions in Reputation-based Trust Management. In Proc. International Semantic Web Conference 2004

[CTSN05] : Golbeck, Jennifer. Computing and Applying Trust in Web-Based Social Networks.(Ph. d. Thesis) 2005.

[CTTN62] : Deutsch, Morton. Cooperation and Trust: Some Theoretical Notes. In Jones, M.R. (Hsg.), Nebraska Symposium on Motivation, 1962.

[DQA02] : Pipino, Leo L.; Lee, Yang W.; Wang, Richard Y.. Data Quality Assessment. In Communications of the ACM, Vol. 45, Issue 4, 2002. S. 211-218.

[EARM03] : Kamvar, Sepandar D.; Schlosser, Mario T.; Garcia-Molina, Hector. The EigenTrust Algorithm for Reputation Management in P2P Networks. In Proc. The 12th International conference on World Wide Web, 2003.

[EASW04] : McGuinness, Deborah L., da Silva, Paulo Pinheiro. Explaining Answers from the Semantic Web: The Inference Web Approach. In Journal of Web Semantics, Vol. 1, Nummer 4, 2004. S. 397-413.

[EBAY] : Ebay, http://www.ebay.com.

[EMDR02] : Yu, Bin; Singh, Munindar P.. An Evidential Model of Distributed Reputation Management. In ,Proc. First International Joint

Conference on Autonomous Agents and Multiagent Systems, 2002.

[ERRB05] : Fujimura, Ko; Inoue, Takafumi; Sugisaki, Masayuki. The EigenRumor Algorithm for Ranking Blogs. In Proc. 2nd Annual Workshop on the Weblogging Ecosystem: Aggregation, Analysis and Dynamics, 2005.

[FILM] : FilmTrust. http://trust.mindswap.org/FilmTrust/.

[FLINK] : FLINK. http://prauw.cs.vu.nl:8080/flink/index.jsp.

[FOAF] : The Friend of a Friend project. http://www.foaf-project.org/.

[FTCC94] : Marsh, Stephen P.. Formalising Trust as a Computational Concept. (Ph. d. Thesis) 1994.

[GEAR] : Google Earth. http://earth.google.com/.

[GOGL] : Google. http://www.google.com.

[ICTI05] : Ziegler, Cai-Nicolas; Golbeck, Jennifer. Investigating Correlations of Trust and Interest Similarity. Submitted to the Journal of Artificial Intelligence Research, 2005.

[MDSP] : mindswap - Maryland Information and Network Dynamics Lab Semantic Web Agent. http://www.mindswap.org/.

[MSTO] : Trust-Ontology veröffentlicht auf im Mindswap "Trust and Reputation in Web Based Social Networks“ Projekt. http://trust.mindswap.org/ont/trust.owl.

[MSTR] : Trust and Reputation in Web Based Social Networks. http://trust.mindswap.org/.

[MTE76] : Shafer, Glenn. A Mathematical Theory of Evidence. Princeton University Press, 1976.

[MTPP01] : Aberer, Karl; Zoran, Despotovic. Managing Trust in a Peer-2-Peer Information System. In Proc. 10th International Conference on Information and Knowledge Management, 2001.

[NG] : Named Graphs. http://www.wiwiss.fu-berlin.de/suhl/bizer/ng4j/.

[NGPT05] : Carroll, Jeremy J.; Bizer, Chistian; Hayes, Pat; Stickler, Patrick. Named Graphs, Provenance and Trust. In Proc. 14th International World Wide Web Conference, 2004.

[ORS03] : R. Guha. Open rating systems. Technical report, Stanford University, 2003.

[PGRI01] : Aberer, Karl. P-grid: A self-organizing access structure for p2p information systems.. In Proc. Sixth International Conference on Cooperative Information Systems, 2001.

[PIGY] : PiggyBank. http://simile.mit.edu/piggy-bank/index.html.

[PRCR98] : Page, Lawrence; Brin, Sergey; Motwani, Rajeev; Winograd, Terry. The PageRank Citation Ranking: Bringing Order to the Web.

Technical report, Stanford Digital Library Technologies Project, 1998.

[QDQA02] : Naumann, Felix. Query-Driven Query Answering for Integrated Information Systems. Lecture Notes in Computer Science, Vol. 2261, Springer 2002.

[RBTM03] : Liu, Ling; Xiong, Li. A Reputation-Based Trust Model for Peer-to-Peer eCommerce Communities. In Proc. IEEE Conference on E-Commerce, 2003.

[RDF] : Resource Description Framework. http://www.w3.org/RDF/.

[RDFS] : Hayes, Patrick. RDF Semantics. W3C Recommendation, Oktober 2004.

[RDQL] : RDQL - A Query Language for RDF. http://www.w3.org/Submission/RDQL/.

[RS00] : Resnick, Paul; Zeckenhausen, Richard; Friedman, Eric; Kowabara, Ko. Reputation Systems. In Communications of the ACM, Vol. 43, Issue 12, 2000. S. 45-48.

[SCCP01] : Friedman, Eric J.; Resnik, Paul. The Social Cost of Cheap Pseudonyms. In Journal of Economics and Management Strategy, Vol. 10, Issue 2, 2001. S. 173-199.

[SECO04] : Harth, Andreas. SECO: Mediation Services for Semantic Web Data. In IEEE Intelligent Systems, Vol. 19, Number 3, 2004. S. 66-71.

[SIMI] : Semantic Interoperability of Metadata and Information in unLike Environment SIMILE. http://simile.mit.edu/.

[SMP67] : Milgram, Stanley. The small world problem. Psychology Today, 61, 1967.

[SSCN03] : Ceglowski, M.; Coburn, A. ; Cuadrado, J.. Semantic search of unstructured data useing contextual network graphs. In , Vol. , Nummer , .

[SW01] : Berners-Lee, Tim; Hendler, James; Lassila, Ora. The Semantic Web. In Scientific American, Vol. 284, Number. 5, 2001. S. 34-43.

[TAOW04] : McGuinness, Deborah L.; da Silva, Paulo Pinheiro. Trusting Answers on the Web. In Mark T. Maybury (Hsg.), New Directions in Question Answering. Kapitel 21, AAAI/MIT Press, 2004.

[TCTS04] : Golbeck, Jennifer; Parsia, Bijan. Trusting claims from trusted sources: Trust network based filtering of aggregated claims. Submitted to the International Semantic Web Confernence 2004.

[TDPP02] : Liu, Ling; Xiong, Li. Building Trust in Decentralized Peer-to-Peer Electronic Communities. In Proc. Fifth International Conference on Electronic Commerce Research, 2002.

[TDRS05] : Ziegler, Cai-Nicolas. Towards Decentralized Recommender

Systems. (Doktorarbeit) , 2005.

[TICT02] : Gil, Yolanda; Ratnakar, Varun. Trust Information Sources One Citizen at a Time. In Proc. First International Semantic Web Conference, 2002.

[TNSW03] : Golbeck, Jennifer; Parsia, Bijan; Hendler, James. Trust Networks on the Semantic Web. In Proc. 7th International Workshop on Cooperatice Information Agents, 2003.

[TPDO93] : Gruber, Thomas R.. Toward Principles for the Design of Ontologies Used for Knowledge Sharing. In Guarino, N.; Poli, R., Formal Ontology in Conceptual Analysis and Knowledge Representation, Kluwer Academic Publishers, 1993. S. 907-928.

[TPWB] : TriQL.P - Trust Policies Enabled Semantic Web Browser. http://www.wiwiss.fu-berlin.de/suhl/bizer/TriQLP/browser/index.htm.

[TQL] : TriQL - A Query Language for Named Graphs. http://www.wiwiss.fu-berlin.de/suhl/bizer/TriQL/Spec/.

[TQLP] : TriQL.P - A Query Language for Querying Named Graphs Published by Untrustworthy Sources. http://www.wiwiss.fu-berlin.de/suhl/bizer/TriQLP/Spec/.

[W3C] : WorldWide Web Consortium. http://www.w3.org/.