

Ein Peer-to-Peer Knowledge Management System basierend auf Topic Maps zur Unterstützung von Wissensflüssen

von Diplom-Informatiker
Thomas Schwotzer
aus Frankfurt (Oder)

von der Fakultät IV - Elektrotechnik und Informatik
der Technischen Universität Berlin
zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften
- Dr.-Ing. -

genehmigte Dissertation

Prüfungsausschuss:
Vorsitzender: Prof. Dr. Hans-Ulrich Heiß
Gutachter: Prof. Dr. Bernd Mahr
Gutachter: Prof. Dr.-Ing. Robert Tolksdorf

Tag der wissenschaftlichen Aussprache: 22. Februar 2006

Berlin 2006
D 83

Das Schreiben einer Dissertation dauert lange. Rückblickend ist es unklar, wann die Arbeit begann. War es der Moment, an dem man den Entschluss fasste, zu promovieren? War es der Moment, an dem die erste Zeile, die nun schon lange gelöscht und vergessen, in ein File getippt wurde? Schaut man aber zurück, erscheint es wie eine wunderbare Fügung von Zufällen und guten Gelegenheiten, die zusammen zu genau diesem Resultat führten. Diese scheinbare Magie haftet aber allen Ereignissen an, deren Ursachen man rückwärts untersucht.

Es ist aber nicht nur der thematische Input, der zu einer Dissertation führt, sondern auch das passende Umfeld. Als These vertrete ich, dass das passende Umfeld sogar wichtiger ist. Wie viele Ideen wurden nicht zu einem Ende gedacht? Treffen sich Idee, Möglichkeit und das eigene brennende Interesse, kann eine so langfristige Arbeit gelingen.

Das brennende Interesse am Nachdenken scheint mir in den Genen zu stecken und so möchte ich mich bei meinen Eltern für diese Anlagen bedanken, aber viel mehr für all die Förderung und Unterstützung, die sie mir zuteil werden ließen.

Ein wichtiges auslösendes Moment für das Entstehen der Arbeit war aber Thomas Preuß, nunmehr Professor an der FH Brandenburg. Ich traf ihn direkt nach meinem Studium 1994 an der BTU Cottbus, wo wir uns als Doktoranden mit verteilten Systemen und Servicegütergarantien beschäftigten. Vielen Dank auch an Professor Harmut König. Er lehnte meine Bewerbung auf einen Posten als Systemadministrator eher belustigt ab und bot mir stattdessen ein Stipendium an. Nach zwei Jahren war eine Verlängerung der Finanzierung nicht möglich und so ging ich nach München zur iXOS AG. iXOS ist Marktführer im Bereich Archive für SAP R/3. So arbeitete ich in SAP und Webumfeld und lernte vieles über Archive, CMS und auch verteilte Archivlösungen. Stellvertretend danke ich vor allem Robert Schwegler, der sowohl fachlich als auch menschlich ein fantastischer Chef war.

1998 wechselte ich nach Berlin zur Condat AG. Thematisch lag die Arbeit zunächst in einem ähnlichen Umfeld. Integrationsprojekte, web- und javabasiert. Ich wurde schnell Projektleiter für ein Projekt für die Deutsche Telekom, dessen Ziel die Erstellung eines Intranet Knowledge Management Systems (IKMS) war. Hier bekam ich einen schnellen und praxisnahen Einstieg in KM. Später wurde ich Produktmanager und verantwortlich für eine mobile Businessplattform namens *Skyware*. Themen wie WAP, GSM, GPRS und UMTS wurden plötzlich relevant. Immer mehr kristallisierte sich auch ein Konzept heraus, dass man damals als *offline client* bezeichnete und das Pate stand für das allgemeinere Konzept der Autonomen Peers in dieser Arbeit. Stellvertretend danke ich herzlich Jürgen Spieler, der mich sowohl zum Projektleiter beförderte und später als Produktmanager *enthronisierte*, wie er es in seiner unnachahmlichen Art bezeichnete. Auch Jimmy Karadimas möchte ich herzlich danken. Vor dieser Zeit waren für mich Informatik-Projekte vorrangig technische Aufgabenstellungen. In kürzester Zeit lernte ich, Controlling und Return-on-investment-Überlegungen nicht als BWL-Unfug abzutun, sondern als wichtige und auch interessante Basis wirtschaftlichen Erfolgs zu sehen.

Während dieser Tätigkeit sah ich die Notwendigkeit, von klassischen Client-Server-Lösungen abzuweichen, vor allem, wenn man sich mit mobilen Lösungen beschäftigt. In der Hektik des Tagesgeschäftes waren solche Ideen aber schwerlich zu Ende zu denken. Und hier kam es zu einer Ansammlung von Zufällen, wie mir rückwirkend scheint.

Skyware wurde zu Beginn 2001 in der ersten Version fertig. Gleichzeitig wurde ein Stiftungslehrstuhl der Telekom an der TU Berlin durch Professor Kurt Geihs besetzt, der wiederum Zweitgutachter der Dissertation meines guten alten Freundes aus den Cottbusser Tagen, Thomas Preuß, war. Er war es, der mich ermutigte, mich mit Kurt zu treffen. Das tat ich dann auch und begann im Mai 2001 an der TU Berlin als erster Mitarbeiter von IVS. Einen herzlichen Dank an Kurt für den Freiraum in der Arbeit, der mir großzügig eingeräumt wurde. Vielen Dank der Deutschen Telekom für die großzügige Ausstattung und die Finanzierung der Reisen zu einigen Konferenzen.

Womit ich mich beschäftigte kann man in den folgenden Seiten nachlesen, und ist nicht verblüffend. Es ging natürlich um Knowledge Management, um Peer-to-Peer Systeme als Alternative und Komplement zu Client-Server-Architekturen und um mobile spontane Netze.

Beschäftigt man sich als Informatiker mit Wissen kommt schnell die Frage nach der Wissensrepräsentation auf. Ich entschied mich für den ISO Standard 13250 Topic Maps. Das hatte mehrere Gründe. Er ist klein und kompakt und einfach zu implementieren. Das war wichtig, wenn man sich mit Programmierung auf Handys beschäftigen wollte. Gleichzeitig zeigte er keine fundamentalen Unterschiede zu RDF, die befürchten lassen könnten, dass es bei einer Überführung zu grundsätzlichen theoretischen Problemen kommen könnte.

2002 nahm ich Kontakt zu Ontopia in Oslo auf. Mitarbeiter der Firma haben den Standard zu wesentlichen Teilen geprägt und sind weltweiter Marktführer im Bereich Topic Map Engines. Ursprünglich hatte ich die Hoffnung, finanzielle Unterstützung zu erhalten. Aber Steve Pepper war auch trotz mitgebrachtem Single Malts nicht zu erweichen. Dafür war die Finanzdecke des jungen und kleinen Unternehmens zu dünn. Aber nicht-monetäre Unterstützung bekam ich uneingeschränkt, so z. B. die kostenlose Nutzung ihres Systems. 2003 hatte ich auch das Vergnügen sechs Wochen in Oslo bei Ontopia, zu verbringen und stellte dort die erste Version einer Topic Map Engine für mobile Geräte fertig. Vorarbeiten hatten einige Studenten geleistet. Hier ist vor allem Björn Tischer zu nennen, dessen Code zwar völlig undokumentiert war, aber außerordentlich stabil lief. Vielen Dank auch den Ontopians, vor allem Pam Genussa und Sylvia Schwab, für die interessanten geschäftlichen Überlegungen. Danke an Lars M. Garshol und Graham Moore für die technischen Debatten. Während einer Diskussion mit Graham wurde mir klar, dass es etwas geben müsste, was ich in dieser Arbeit als *Kontext* bezeichne. Damals hatte ich leider keine Bezeichnung dafür. Danke auch Lars, dass du mich weiter zum Squash mitnahmst, obwohl ich um Klassen schlechter spiele. Danke auch für die wunderbaren Abende und insbesondere die Kostproben von Geir Oves Braukünsten, die mindestens an seine Brillanz als Informatiker heranreichen.

Ab 2004 begann ich meine Ideen zu strukturieren und mit dem Schreiben anzufangen. In der Zeit besuchte ich Messen, um kommerzielle Anwendungen verteilter mobiler Systeme, basierend auf Bluetooth, zu erurieren. Für die Idee, Shark als Verkaufsplattform für digitale Musik zu benutzen, vergab das BMWA 2004 einen Hauptpreis in einem Wettbewerb bei 104 eingereichten Geschäftsideen.

2005 nahm Kurt Geihs einen Ruf aus der Uni Kassel an, was zur Auflösung des Lehrstuhles führte. Für mich bedeutete das auch, dass ich mir einen neuen Doktorvater suchen

musste. Professor Bernd Mahr war schnell dazu bereit, und obwohl er sich bisher nicht oder nur am Rande mit Topic Maps, Knowledge Management und mobilen spontanen Netzen beschäftigt hatte, unterbrach er mich bei meinen ersten Präsentationen bereits recht schnell und stellte sehr knifflige Fragen. Wer Bernd kennt, weiß seine freundliche, offene und neugierige Art zu schätzen. Mich brachte diese inhaltliche Debatte in der Arbeit sehr voran und ich sehe das Bild der untergehenden Sonne noch vor mir, wenn ich in seinem Büro aus dem Fenster schaute und mal wieder einige Fragezeichen im Kopf hatte. Es war eine außerordentlich spannende Zeit, und recht schnell bedauerte ich, dass ich nicht länger an seinem Lehrstuhl FLP gearbeitet habe. Aber die Zeit drängte, da mein Vertrag auslief. Bernd gab mir auch dabei die größte Unterstützung und so reichte ich meine Arbeit im Oktober 2005 ein, verteidigte im Februar 2006 und kann die Arbeit nun im Mai 2006 zur Veröffentlichung freigeben. Vielen vielen Dank Bernd für alles! Dank auch an Professor Robert Tolksdorf, der sich schnell als Zweitgutachter fand, und an Professor Heiß, der als Vorsitzender der Prüfungskommission meinem Drängeln nach einem schnellen Verteidigungstermin mit der angebrachten Ruhe, aber schnellem Handeln entgegenkam.

Nach meinem Ausscheiden aus der TU Berlin werde ich am 17.5.2006 zur neofonie GmbH, Berlin als Leiter Forschung/Entwicklung wechseln. Die neofonie beschäftigt sich mit Suchmaschinen und will ihre Leistungen aber in den Bereich Semantic Web / Topic Maps ausweiten. Dieses Angebot kam auch wieder wie ein Wunder, weil es sowohl inhaltlich als auch zeitlich perfekt passt.

Mein größter Dank gilt aber meiner Freundin, Susanne Walter, die sich jahrelang irgendwelches Informatikzeug anhören musste und sich ansehen durfte, wie ich auch am Wochenende oder abends im Büro war bzw. daheim im Laptop verschwand. Vielen Dank für all die Geduld und die Unterstützung. Vielen Dank auch an meine Tochter Johanna. Ihre Geburt war für den 5.5.2004 vorausgesagt. Das war exakt der Tag, an dem ich meine Arbeiten an der kleinen Mathematik in Kapitel 5 beendete. Am 17.5.2004 kam sie auf die Welt und hat sie ein Stück weit heller gemacht, wie ich meine. Danke, meine Kleine, dass Du mir vor allem in den ersten Monaten gestattest, am Laptop zu sitzen, während du Drehversuche machtest oder herumkrabbeltest. Und Danke auch dafür, dass du das nicht mehr zulässt, seit du laufen kannst und vor allem das Wort *Nein* aussprechen kannst.

Wie lange hat nun das Erstellen dieser Arbeit gedauert? Zwölf Jahre, fünf oder nur zwei? Eine Antwort ist schwer zu finden. Wichtig ist, dass sie fertig ist und nun die Zeit gekommen ist, die Theorie praktisch anzuwenden. Ob das dauerhaft gelingen wird, kann am heutigen Tag nicht entschieden werden. Vielleicht findet sich aber auch noch später etwas in einer Suchmaschine unter den Worten *Shark* oder vielmehr *iSphere*, wie wir das System auch bezeichnen. Unabhängig aber davon, was die Zukunft bringt, kann ich für die Vergangenheit sagen, dass ich sehr dankbar dafür bin, die Gelegenheit bekommen zu haben, einer Idee einmal mehrere Jahre widmen zu können. Ich konnte das umso mehr genießen, nachdem ich einige Jahre das Tagesgeschäft in Softwarefirmen miterleben und -gestalten durfte. Empfehlenswert.

Berlin, den 10. Mai 2006

Inhaltsverzeichnis

1	Einleitung	13
2	Verwandte Arbeiten zu Wissensflüssen	21
2.1	Soziale Systeme	22
2.1.1	Gruppen	22
2.1.2	Gruppe als System	23
2.1.3	Umweltkomplexität und Kontingenz der Handlungsalternativen . .	23
2.1.4	Limitierte Ressource	25
2.1.5	Teilsysteme und Teilumwelten	27
2.1.6	Individuum als System	27
2.1.7	Zusammenfassung	28
2.2	Wissen und Kontext	29
2.2.1	Begriffssysteme	30
2.2.2	Ontologien	33
2.2.3	Weltwissen versus Mikrotheorien	34
2.2.4	Wissensrepräsentation	38
2.2.5	Kontext in der Prädikatenlogik	41
2.2.6	Kontext in Content-Management-Systemen	46
2.2.7	Zusammenfassung	50
2.3	Software-Agenten	51
2.3.1	Kontextbasierter Austausch von Wissen	52
2.3.2	Sprechakte	54
2.3.3	Agentensprachen	55
2.3.4	Zusammenfassung	56
2.4	Knowledge Management (KM)	57
2.4.1	Nonaka und Takeuchi	59
2.4.2	Davenport und Prusak	64
2.4.3	IT-Unterstützung für das SECI-Modell	65
2.4.4	Knowledge Nodes - Bausteine verteilten Wissens	68
2.4.5	Zusammenfassung	71
2.5	Ein P2P Modell für das Knowledge Management	72
2.6	Zusammenfassung	77

3	Autonome kontextsensitive Peers – ACP-Modell	79
3.1	Das Modell	80
3.1.1	Peer	80
3.1.2	Kontext	80
3.1.3	Daten	81
3.1.4	Umgebung	81
3.1.5	Bezeichnung	81
3.1.6	Eigentümer	82
3.1.7	Dateneinheiten	82
3.1.8	Ports	82
3.1.9	Extraktion und Assimilation von Daten	84
3.1.10	Austausch von Dateneinheiten	84
3.1.11	Fluss von Daten	86
3.1.12	Zusammenfassung	87
3.2	Peer-Beziehungen	88
3.2.1	Proxy-Peers	88
3.2.2	Peer-Gruppen	91
3.2.3	Synchronisation und Client-Server	95
3.3	Zusammenfassung	95
4	Shared Knowledge (Shark)	97
4.1	Basiskonzepte	98
4.1.1	Shark Peers	98
4.1.2	Shark Knowledge Nodes	98
4.1.3	Wissen und Interessen	101
4.1.4	Wissen und Kontext	101
4.2	Kontext	102
4.2.1	Erwartungskontexte in Shark	102
4.2.2	Kontextualisierung	103
4.2.3	Spezialisierung der ACP-Funktionen f,g und h	104
4.2.4	Extraktion von Wissen	106
4.2.5	Assimilation von Wissen	106
4.3	Shark-Engine	107
4.3.1	Knowledge Ports	108
4.3.2	Funktionen einer Shark-Engine	109
4.4	Eigentümer, Sender und Empfänger von Wissen	111
4.4.1	Shark Policies	112
4.5	Wissensflüsse	116
4.5.1	Wissensflüsse zwischen Gruppen und Gruppenmitglied	116
4.5.2	Wissensflüsse über Gruppengrenzen	119
4.5.3	Wissensflüsse in Gruppenhierarchien	120
4.5.4	Korona einer Knowledge Node	121
4.6	Zusammenfassung	123

5	Kontextualisierung mit Topic Maps	127
5.1	Topic Maps	127
5.1.1	Grundkonzepte	128
5.1.2	Topic Maps - persönliche Ontologien	131
5.1.3	Gleichheit und Merge von Topics	134
5.1.4	Reference Model	137
5.1.5	Topic Map Query Language	138
5.1.6	Wissen und Kontext in Topic Maps	138
5.1.7	Verteilte Topic Map Engines	139
5.2	Topic Maps als Mengen und Graphen	141
5.3	Connectivity Graphs	143
5.4	Operationen	148
5.4.1	Merge	148
5.4.2	Selektion	148
5.4.3	View	150
5.4.4	Korona	152
5.4.5	Fragmentierung	154
5.5	Kontextualisierung	155
5.5.1	Eigenschaften der Kontextualisierung	158
5.5.2	Vergleich Kontextualisierung mit TMQL	160
5.6	Konkretisierung des Modell Shark	160
5.6.1	Interessen und Themen als Topic Maps	160
5.6.2	Personalisierung / Lokalisierung von Topic Maps	160
5.6.3	Gleichheit von Kontexten und Wissen in Topic Maps	162
5.6.4	Extraktion	162
5.6.5	Assimilation	163
5.6.6	Wissensaustausch mit Topic Maps	164
5.7	Zusammenfassung	166
6	Architektur einer Shark-Engine	167
6.1	Knowledge Exchange Protocol (KEP)	167
6.1.1	Genereller Aufbau einer KEP-Nachricht	168
6.1.2	interest	168
6.1.3	offer	169
6.1.4	accept	169
6.1.5	insert	170
6.1.6	stop	171
6.1.7	Eigenschaften des Protokolls	171
6.1.8	Einige KEP-Szenarien	171
6.1.9	KEP Redirect	173
6.1.10	Das einfachste KEP Szenario	174
6.1.11	Vergleich mit Software-Agenten	175
6.2	Plattformen für die Shark-Engine	175
6.2.1	Shark Central Station	177

Inhaltsverzeichnis

6.2.2	Shark Mobile Station	180
6.2.3	Shark Local Station	181
6.2.4	Mögliche Topologien	182
6.3	Anwendungsszenarien	183
6.3.1	Mobiles verteiltes Knowledge Management System	183
6.3.2	Mobiles und Virales Marketing	185
6.3.3	Sensornetze	187
6.4	Zusammenfassung	189
7	Zusammenfassung und Ausblick	191

Abbildungsverzeichnis

2.1	System und Umwelt	26
2.2	Teilsysteme und Teilumwelten	27
2.3	Das semiotische Dreieck	31
2.4	Eine Ontologie: Leser liest Dissertation	34
2.5	Partitioniertes Weltwissen	36
2.6	Aggregierte Mikrowelten	37
2.7	Beispiel in einem 3-D Context Space	45
2.8	Informationsbasis nach Mylopoulos	49
2.9	Kontextbasierter Wissensaustausch zwischen Agenten	53
2.10	Das Seci-Modell nach Nonaka und Takeuchi	61
2.11	Wissensspirale nach Nonaka und Takeuchi	62
2.12	Zentralistisches Knowledge Management System	66
2.13	Das System Organisation als Kegel	73
2.14	Kreismodell der ontologischen Dimension	74
2.15	Kreismodell mit Relevanzfilter und Strategie	75
2.16	Wissensaustausch zwischen zwei ontologischen Punkten	76
3.1	P2P Datenaustausch	85
3.2	Datenfluss in einen Netz von Peers	87
3.3	Master und Proxy	90
3.4	Ein Meta-Peer	92
3.5	Peer-Gruppe mit vier Mitgliedern und zwei Mastern	93
4.1	Hierarchische Gruppen von Shark-Peers	100
4.2	P2P Wissensaustausch in Shark	106
4.3	Architektur der Shark Engine	108
4.4	Knowledge Node nutzt Shark-Engine	110
4.5	Aktive Shark-Engine	111
4.6	Wissensflüsse innerhalb einer Gruppe	117
4.7	Wissensflüsse über Gruppengrenzen	120
4.8	Korona von Wissen einer Gruppe	122
5.1	Beispiel: Eine Hamlet Topic Map	130
5.2	Beziehung Topic Map und Autor	132
5.3	Nutzung PSI Set für Länder	136

Abbildungsverzeichnis

5.4	Beispiel Topic Map	145
5.5	Connectivity Graph	146
5.6	Selektion	149
5.7	Korona	153
5.8	Fragmentierung	155
5.9	2 Varianten der Extraktion mit Topic Maps	163
5.10	P2P Wissensaustausch in Shark mit Topic Fragmenten	165
6.1	Das Standard KEP Szenario	172
6.2	KEP mit Redirect	173
6.3	Das einfachste KEP Szenario	174
6.4	Shark Station - Implementierung der Shark Engine	177
6.5	Shark Central Station	178
6.6	Shark Mobile Station	180
6.7	Shark Local Station	181
6.8	Beispiel-Topologie von Shark Engines	182
6.9	Sensornetze mit Shark	184
6.10	Mobiles und virales Marketing mit Shark	187
6.11	Sensornetze mit Shark	188

1 Einleitung

Now more than ever we need people who can lead humanity towards technologies that improve society, rather than technologies that simply improve our technology itself.

John Maede (gefunden im Design Museum, London im Sommer 2003)

Diese Arbeit beschäftigt sich mit Wissen und dessen Fluss in Gruppen von Menschen. Wissen ist keine Erfindung der Neuzeit oder gar der Informationstechnologie (IT). Ein qualitativer Unterschied zwischen Mensch und Tier besteht in der menschlichen Fähigkeit, (auch komplexe) Arbeitsschritte vor auszudenken. Menschen können planen. Dazu bedarf es wenigstens der folgenden Dinge: Bewusstsein, Wissen und der Fähigkeit mit Wissen umzugehen: Intelligenz. In grauer Vorzeit entdeckten unsere Vorfahren den Umgang mit Feuer. Es war neues Wissen entstanden. Sie gaben ihr Wissen weiter, indem sie anderen, z. B. ihrem Nachwuchs zeigten wie man ein Feuer entzünden kann. Dieses Verhältnis zwischen Meister und Schüler war über Jahrhunderte hinweg *der* Weg, wie Wissen weitergegeben wurde. Die Weitergabe von Wissen erfolgt dabei durch direkten Austausch. Im Knowledge Management (KM) würde man von einer *Sozialisierung* sprechen.

Die Erfindung der Schrift und später des Buchdruckes änderten den Umgang mit Wissen fundamental. Plötzlich war es möglich, Wissen unabhängig von Menschen zu machen und weiterzugeben. Im KM nennt man das Niederschreiben von Wissen *Externalisierung*. In diesem Prozess hält ein Mensch Wissen auf einem externen Medium (sei es die Wand eines Tempels, ein Pergament oder ein ASCII-File) fest. Andere können dieses so entstandene *explizite Wissen* lesen und damit ihr eigenes Wissen erweitern. Dieser Vorgang wird als *Internalisierung* bezeichnet.

Es gibt Einwände, von explizitem *Wissen* zu sprechen. Ein Dokument könne kein Wissen enthalten, wenn Wissen als Fähigkeit definiert wird. Wissen könne nur bei intelligenten Wissen vorliegen, denn nur sie könnten qualifiziert handeln, siehe dazu auch Abschnitt 2.4. Diese Diskussion verweist auf einen weiteren wesentlichen Begriff: das Hintergrundwissen bzw. allgemeiner der Kontext. Externalisierung von Wissen erfolgt immer in einem Kontext. Das schließt die historische Situation ein, die Kenntnisse der Schreibenden, deren persönliche Situation usw. Als Hintergrundwissen werden gemeinhin die Kenntnisse der Schreibenden bezeichnet.

Auch die Internalisierung erfolgt in einem Kontext. Je mehr sich Kontext der Externalisierung und der Internalisierung unterscheiden, umso größer ist die Gefahr von Fehldeutungen bzw. die Unmöglichkeit, Wissen zu erlangen. So müssen Schreibende und Lesende wenigstens eine Sprache und Terminologie benutzen, die beide verstehen.¹

¹Der Autor dieser Arbeit ist z. B. des Altägyptischen nicht mächtig und kann sich höchstens als inter-

1 Einleitung

Die Möglichkeit der Externalisierung erlaubte es nun aber auch, explizites Wissen aus verschiedenen Quellen zu neuem explizitem Wissen zusammenzufügen. Man spricht im KM von *Kombination*. Kombination ist z. B. auch die Zusammenstellung von unterstützenden Materialien für eine Vorlesung. Kombination unterstützt das Fließen von Wissen über die Jahrhunderte hinweg. Vorhandenes Wissen wird aufgegriffen, teilweise selektiert in einen neuen Kontext gestellt, mit anderem (neuen) Wissen kombiniert und weitergegeben.

Auch nach der Erfindung der Schrift und des Buches blieb der direkt Austausch von Mensch zu Mensch der wesentliche Weg der Wissensverbreitung. Das gilt auch heute im Jahre 2006! Das mag verblüffen. Tatsächlich zeigen Studien, dass Menschen Wissen vor allem von anderen Menschen annehmen, die sie kennen. Die moderne Informationstechnik hat keine fundamentale Änderung im Austausch von Wissen herbeigeführt! Lediglich die Art des Austausches hat sich verändert. So zeigten in der Steinzeit Erfahrene den Jüngeren die Herstellung von Werkzeugen, im Mittelalter übergaben Meister ihren Zöglingen ein Pergament zur Lektüre, während man heute eine E-Mail mit dem Vorlesungsskript versendet. Im Kern bleibt es das Gleiche: Wissen wird zwischen Menschen weitergegeben, die sich kennen und vertrauen. Das galt in den wandernden prähistorischen Horden, genauso wie es in modernen High-Tech-Unternehmen heute gilt.

Diese Tatsache wurde lange Zeit übersehen. Mit der aufkommenden Informationstechnik und insbesondere der künstlichen Intelligenz entwickelte sich in den 1970er bis in die 1990er Jahre hinein der Glaube, dass prinzipiell alles Wissen von Menschen maschinell erfassbar wäre und dass das Entwickeln neuen Wissens ebenso automatisierbar wäre, wie z. B. die Produktion von Waschbecken. Das war ein Trugschluss. Warum?

Unternehmen weisen eine Struktur auf. Organigramme beschreiben sie. Aus keinen Unternehmen sind heute mehr IT-Systeme wegzudenken. Sie dienen der Speicherung von Daten für alle denkbaren Prozesse.

Problematisch wurde die Situation, als die Flüsse von Informationen in Unternehmen durch IT-Systeme unterstützt werden sollten. Diese Entwicklung begann großflächig in den 1980er Jahren. Alle Wege von Informationen galt es zu identifizieren und später durch IT-Systeme zu unterstützen. Das war die Geburtsstunde von Workflow-Management-Systemen. Es gab eine Reihe von Zielen, die damit erreicht werden sollten. Bei Ausfall von Mitarbeitern könnten die Informationsströme automatisch auf andere umgelenkt werden (Stellvertreterregelungen). Die nunmehr explizit vorliegenden Pläne der Informationsflüsse könnten zu Optimierungen genutzt werden. Schlussendlich wollte man die Informationsverarbeitenden Prozesse, die oftmals durch Menschen erledigt wurden, schrittweise automatisieren. Einige dieser Ziele wurden erreicht. Das Ziel der vollständigen Automatisierung nicht.

Dafür entstanden einige neue Probleme, die vor allem von Davenport und Prusak, aber auch von Senge beschrieben wurden, siehe Abschnitt 2.4. Der Einsatz von IT führte zum Abbau von Beschäftigten. Das wirkt sich bis zu bestimmten Grenzen als Effektivitäts-

essierter Laie im Bereich moderner Physik bezeichnen. Insofern kann bei ihm die Betrachtung einer mit Hieroglyphen überdeckten Tempelwand genauso wenig Wissen erzeugen, wie die Lektüre eines Fachartikels einer Konferenz über Quantenphysik. Bei beidem ist nicht genügend Hintergrundwissen vorhanden.

steigerung aus. Mitte der 1990er Jahre wurden aber eher unscharfe Probleme sichtbar. Fehler bei der Arbeit wiederholten sich. Kenntnisse, die bei einem Projekt genutzt wurden, waren plötzlich nicht mehr auffindbar. Aus damals nicht erkennbaren Gründen wurde die Automatisierung von Produktionsprozessen immer mühsamer, das Sammeln der dazu notwendigen Informationen immer langwieriger. Es schien auch, dass sich einige Prozesse der Automatisierung entzogen. Das war die Geburtsstunde des Knowledge Managements. Die erste Leistung bestand darin, *wissensintensive* Prozesse zu identifizieren. Das sind solche, deren Ablauf kaum oder gar nicht bekannt ist und für dessen Durchführung kaum oder keine Vorlagen existieren. Forschung jeglicher Art sind solche Prozesse, aber auch die Entwicklung und Verbesserung von Produkten. Werden solche Prozesse wiederholt, werden sie schrittweise formalisierbar und nach einer Weile nicht mehr wissensintensiv sein.

Ein IT-System kann *nicht*-wissensintensive Prozesse hervorragend unterstützen bzw. automatisieren. Formalisierte Verwaltungsabläufe lassen sich auf Workflowsysteme abbilden. Daher kamen die Steigerungen der Produktivität durch den Einsatz von IT. Versucht man allerdings wissensintensive Prozesse durch ein IT-System zu verwalten, so scheitert das. Warum?

Es sind zwei Gründe. Agierende in wissensintensiven Prozessen betreten per Definition Neuland in ihrer Arbeit. Sie arbeiten kreativ an bisher ungelösten Problemen und oft in Abläufen, die neu sind. Es ist neues Wissen vonnöten und das Wissen muss zwischen den Agierenden fließen. Dieser Prozess kann nicht primär durch Software gesteuert werden. IT-Systeme können ihn lediglich unterstützen.

Das zweite Problem besteht darin, dass ein IT-System Dokumente benötigt. Sollen wissensintensive Prozesse durch IT unterstützt werden, so müssen die Teilnehmer ihr Wissen externalisieren, auf das es andere internalisieren können. In einer laufenden Entwicklung/Forschung ist das nur zum Teil möglich. Es wird noch zu sehen sein, dass Wissen immer in den Köpfen Einzelner entsteht, aber dass der nächste Schritt immer die Diskussion mit anderen ist. Hier liegt nur sehr unscharfes Wissen vor, das nicht oder nur teilweise externalisiert werden kann.

Die erste wegweisende Leistung des noch jungen KM und konkret von Nonaka und Takeuchi [NT95] bestand Mitte der 1990er Jahre darin, die Wichtigkeit von *implizitem* Wissen herauszustellen. Das ist Wissen, das nicht explizit ist, das also nur in den Köpfen der Menschen existiert. Sie zeigten, dass wissensintensive Prozesse wesentlich auf implizitem Wissen basieren, dass es in solchen Prozessen nicht möglich ist, das implizite Wissen zu externalisieren, und dass deshalb Wissen immer in und zwischen Gruppen von Menschen fließt, meistens auch im direkten persönlichen Austausch. IT-Systeme können nur mit explizitem Wissen umgehen. In wissensintensiven Prozessen können sie gut als Rechercheinstrument, Zeiterfassungssystem usw. genutzt werden. Als zentrale Plattform sind sie ungeeignet.

Es ist also *nicht* möglich, Unternehmen mit wissensintensiven Prozessen allein als informationsverarbeitende Maschinen aufzufassen und deren Prozesse mit IT-Systemen zu automatisieren. Sie sind vielmehr als eine Gruppe von Menschen anzusehen, in der Wissen entsteht und zwischen den Mitgliedern der Gruppe fließt. Wie bereits bemerkt, hat sich daran seit Menschengedenken nichts geändert. Die Tatsache wurde nur augenfällig,

1 Einleitung

als die Automatisierung von Produktionsabläufen an Grenzen stieß. Diese Grenze wird durch die Notwendigkeit von Wissen in einem Prozess gezogen.

Anders gesagt, gelang es in der Industriegesellschaft, die menschliche Arbeit zu großen Teilen unnötig zu machen. Das gelingt bei Prozessen, die automatisierbar, also nicht-wissensintensiv sind. Die derzeitige post-industrielle Gesellschaft, oder auch *Wissensgesellschaft* zeichnet sich dadurch aus, dass menschliche Arbeit vor allem in wissensintensiven Prozessen unabkömmlich ist. Gleichzeitig wächst das notwendige Expertenwissen für solche Prozesse beständig, während die Wertschöpfung der einzelnen Teilnehmer ebenso wächst. Das hat zwei Effekte. Einmal kann in der Praxis kein Experte allein einen wissensintensiven Prozess (z. B. Forschung) allein durchführen. Es werden also immer Teams benötigt. Gleichzeitig steigt der Mehrwert eines *Wissens*-Arbeiters beständig. Der effektive Umgang mit Wissen ist heute wesentlicher ökonomischer Faktor von Unternehmen geworden. Es ist nicht übertrieben zu behaupten, dass die Effektivität des Managements des Wissens wesentlich über den Erfolg eines Unternehmens bestimmt.

Seit Mitte der 1990er Jahre weiß man, dass sich Wissen nicht von Menschen trennen lässt. Management von Wissen ist immer das Management von Menschen und die Organisation eines effektiven Wissensaustausches zwischen ihnen. Diese Erkenntnisse führten Ende der 1990er Jahre zu einem neuen Verständnis von Management. Manager waren nunmehr zuständig für die Schaffung einer Arbeitsumgebung, in der engagierte Mitarbeiter arbeiten und sich austauschen können. Manager waren weniger zuständig für die detailgenaue Definition von Arbeitsabläufen für diese Prozesse. High-Tech-Unternehmen strukturierten die Arbeitsumgebung und -abläufe so um, dass informelle Kontakte erleichtert wurden. Man erkannte, dass Wissen zu großen Teilen in informellen Netzen der Mitarbeiter fließen und nicht vorrangig entlang der definierten Hierarchien eines Unternehmens. Die Unterstützung solcher informellen *Communities of Practice* verbesserte die Flüsse von Wissen in Unternehmen.

An der Entwicklung von IT-Systemen ging diese Entwicklung zu einigen Teilen vorbei. Trotz dieser nunmehr zehn Jahre alten Erkenntnisse werden noch heute Knowledge Management Systeme (KMS) gebaut, die das Organigramm eines Unternehmens abbilden und von Wissensflüssen entlang der Firmenhierarchie ausgehen. Nachweisbar spielen dabei aber informelle Netze eine deutlich größere Rolle in Unternehmen, siehe dazu vor allem Abschnitt 2.5. Diesem Problem widmet sich diese Arbeit.

Die meisten KMS basieren auf dem Client-Server-Modell und verlangen grob diesen Ablauf: Ein *Wissens*-Server dient der Ablage von explizitem Wissen. Anwender des Systems sind angehalten, ihr Wissen zu externalisieren und in das System einzuspielen. Es gibt eine Reihe von Überlegungen für Anreizsysteme, auf die nicht weiter eingegangen werden soll. Bei aufkommenden Fragen sind die Angestellten angehalten, zunächst den Wissens-Server nach passendem Wissen zu durchsuchen. Wird man dort nicht fündig, soll man neues Wissen entwickeln. Durch dieses Vorgehen soll Wissen im gesamten Unternehmen verfügbar werden, das Unternehmen soll insgesamt *lernen* und das mehrfache Erarbeiten gleicher Lösungen soll größtenteils ausgeschlossen werden.

Dieser Ansatz erfüllte nicht alle Hoffnungen. Untersuchungen zeigen, dass der Großteil des expliziten Wissens der Mitarbeiter in der Mailboxen bzw. deren Filesystem auf ihrem eigenen PC liegt. Das gilt auch für Unternehmen mit einem eingeführten KMS. Eine

generelle Scheu vor IT kann nicht die Ursache sein, bedeutet doch das Vorliegen von relevanten Mengen expliziten Wissens in den Mailboxen, dass die Mitarbeiter keinerlei Bedenken gegenüber der Nutzung von E-Mail haben.

In dieser Arbeit wird die These vertreten, dass der P2P-Charakter des Wissensaustausches durch aktuelle KMS nicht beachtet wird. Das Versenden einer E-Mail an Bekannte entspricht einem P2P-Austausch. Es wird als direkter Austausch wahrgenommen und bildet die Art des Austausches von Wissen nach, dem Menschen nachgehen, seit sie denken können.

Das Hochladen eines Dokumentes auf einen zentralen Server ist vom Charakter her etwas anders. Es ist ein offizieller Akt der Freigabe von explizitem Wissen für eine größere Nutzergruppe. Der Wissens-Server gehört zumeist auch einer Abteilung oder gar dem Unternehmen. Er wird damit als eine unpersönliche Instanz wahrgenommen. Ganz offensichtlich sind die Hemmschwellen zum Einstellen von Dokumenten darauf deutlich höher als beim Versenden einer E-Mail.

Bei der Recherche gilt Ähnliches. Es hat sich gezeigt, dass Mitarbeiter trotz eingeführter KMS zuerst Kollegen um Hilfe bitten, bevor sie einen zentralen Server nutzen. Das machen sie trotz Kenntnis darüber, dass die Kollegen mit Sicherheit weniger Fachkenntnisse haben als Autoren von Dokumenten in zentralen Servern. Diese Handlungsweise hat die gleiche Ursache. Der direkte Austausch zwischen Menschen ist bevorzugter Weg des Wissensaustausches. So werden zuerst Bekannte befragt. Erst im zweiten Schritt wird ein zentraler Server benutzt.

Diese Erkenntnisse sind – wie gesagt – nicht neu. Mitte der 1990er Jahre erfolgten wesentliche Umstrukturierungen in Unternehmen und es entwickelte sich ein neues Verständnis von Management. Dieser Schritt wurde bei vielen KMS bisher nicht vollzogen. Warum?

Die Entwicklung eines IT-Systems erfolgt meistens in den folgenden Schritten. Die Anwendungsdomäne wird untersucht und ein passendes Modell wird entwickelt. Basierend auf dem Modell wird eine Softwarearchitektur entworfen. Diese wird implementiert und betrieben.

Im Bereich Knowledge Management tut sich ein Problem auf. Die Erkenntnisse bzgl. direktem Austausch von Wissen haben sich noch nicht niedergeschlagen in ein Modell. Ein anerkanntes Modell ist das SECI-Modell von Nonaka und Takeuchi. Das Modell hat eine wesentliche Schwäche. Es beschreibt, wie Wissen zwischen Individuen, Gruppen, Organisationen und intraorganisationell fließt. Gleichzeitig wird verbal darauf hingewiesen, dass der Wissensaustausch primär von Mensch zu Mensch erfolgt. Diese Tatsache ist aber nicht explizit in das SECI-Modell eingeflossen. Hier herrscht noch großer Spielraum für Interpretationen.

Für die Arbeitsorganisation stellte das kein Problem dar. Diagramme und explizite Modelle sind nicht so wesentlich, um eine neue Vorstellung von Management zu entwickeln. Für die Entwicklung eines IT-Systems ist das Problem aber fatal.

Hier setzt diese Arbeit an. Der grundlegende Ansatz ist dieser: Die treibenden Kräfte wissensintensiver Prozesse sind Menschen. Diese tauschen Wissen primär direkt miteinander aus. Ein Fluss von Wissen lässt sich also immer auf eine Summe von direkten Austauschen zurückführen. Ein IT-System, das solche Prozesse unterstützt, muss als

1 Einleitung

eine Unterstützung eines Menschen verstanden werden und nicht als dessen Ersetzung. Es muss die Ablage von explizitem Wissen erlauben und den einfachen direkten (P2P) Austausch. Es muss außerdem die Konstituierung von Gruppen erlauben und den Fluss von Wissen in Gruppen von Menschen unterstützen. Ein solches Peer-to-Peer Knowledge Management System (P2P-KMS) wird in dieser Arbeit entwickelt. Es trägt den Namen *Shark* (*Shared Knowledge*).

Dazu wird in folgenden Schritten vorgegangen. Zunächst werden die relevanten Arbeiten vorgestellt. Das sind insbesondere eine Systemtheorie sozialer Gruppen (Abschnitt 2.1), Arbeiten aus dem Bereich der Wissensrepräsentation (Abschnitt 2.2) und der Agentenforschung (Abschnitt 2.3). Außerdem werden die wesentliche Ansätze aus dem Knowledge Management vorgestellt (Abschnitt 2.4).

Es wird zu sehen sein, dass die Erkenntnis, dass sich der Fluss von Wissen vor allem auf den direkten P2P Austausch zwischen Menschen zurückführen lässt, noch nicht vollständig und explizit Eingang in Modelle des Knowledge Managements gefunden hat und dass sich dadurch Akzeptanzprobleme erklären lassen (Abschnitt 2.4.3). So motiviert wird in Abschnitt 2.5 eine Erweiterung der SECI-Modells von Nonaka und Takeuchi vorgestellt werden, das Flüsse von Wissen vollständig auf P2P Austausche zwischen Individuen zurückführt und auch dem Austausch von Wissen in und zwischen Gruppen und Organisationen erklärt. Diese Erweiterung soll das Peer-to-Peer Knowledge Management (P2P-KM) Modell genannt werden. Es vereint Erkenntnisse der Soziologie und des Knowledge Managements.

In Kapitel 3 erfolgt ein thematischer Sprung. Es wird ein allgemeines P2P-Modell vorgestellt werden. Basiskonzepte des Modells sind Peers, die Daten verwalten können und in der Lage sind, einen *Kontext* wahrzunehmen. Der Begriff des Kontextes wird sehr allgemein gefasst und kann Umgebungswerte genauso wie innere Zustände der Peers umfassen. Es wird weiterhin ein Konzept der *Ports* eingeführt, mit denen Peers Bedingungen an Kontexte beschreiben können unter denen sie willens sind Daten mit anderen Peers auszutauschen. Aus diesen Kernkonzepten ergibt sich auch der Name: Autonomous Context-aware Peers (ACP) Modell. Trotz der Allgemeinheit der Konzepte wird zu sehen sein, dass sich mit dem ACP-Modell Gruppenbildungen von Peers und Kommunikation zwischen und innerhalb von Gruppen modellieren lassen.

In Kapitel 4 werden ACP-Modell und P2P-KM Modell zu einer Referenzarchitektur zusammengefügt. Diese Architektur trägt den Titel *Shark*. Die Kernkonzepte des ACP-Modells bleiben erhalten, werden aber für das Knowledge Management konkretisiert. So verwalten Peers Wissen und beschreiben mittels *Knowledge Ports* Bedingungen, zu denen sie bereit sind, Wissen mit anderen Peers auszutauschen. Diese Bedingungen werden allgemein als Anforderungen an Kontexte definiert und sind konkret Anforderungen an Interessen und Hintergrundwissen der Akteure im System. Das sehr allgemeine Konzept des Kontextes aus dem ACP-Modell wird damit für das Knowledge Management konkretisiert und fließt in die konzeptuelle Komponente (Knowledge Port) ein. In *Shark* wird zusätzlich die abstrakte Funktion der *Kontextualisierung* definiert. Sie dient der Ermittlung von Wissen aus einer Wissensbasis, das zu einem gegebenen Kontext passt. Es wird zu sehen sein, wie sich die Funktionen der Knowledge Ports auf die Kontextualisierung zurückführen lassen.

Eine Referenzarchitektur beschreibt ein Softwaresystem, blendet allerdings konkrete technische Umsetzungen aus. Bei Shark wird insbesondere ignoriert, wie das (explizite) Wissen der Peer repräsentiert wird und welche Protokolle für den Austausch von Wissen eingesetzt werden. Ohne die Wahl eines Repräsentationsformats lässt sich auch die Kontextualisierung nicht konkretisieren.

In Kapitel 5 erfolgen einige Konkretisierungen. Es wird gezeigt, wie der ISO Standard Topic Maps zur Repräsentation von Wissen der Peers benutzt werden kann. Es wird eine Realisierung der Kontextualisierung über Topic Maps definiert werden. Es wird außerdem gezeigt werden, wie ein P2P-Austausch von Wissen vollständig auf die im Topic Map Standard definierten Funktionen *merge* und die Kontextualisierung zurückgeführt werden kann. Dieses Erkenntnis bedeutet, dass existierende Topic Map Engines lediglich durch die Funktion der Kontextualisierung erweitert werden müssen, um Basis eines P2P-Knowledge Management Systems zu sein. Weiterhin wird in Abschnitt 6.1 ein Protokoll für den Wissensaustausch beschrieben, das *Knowledge Exchange Protocol (KEP)*.

Kapitel 6 beschreibt einige Implementierungsmöglichkeiten von Shark, insbesondere auch in mobilen spontanen Netzen. Die Beschreibungen bleiben an der Oberfläche, es wird auf andere (Diplom-) Arbeiten verwiesen, in denen diese Details ausgiebig behandelt werden. Weiterhin werden Anwendungsfelder von Shark auch jenseits des Knowledge Managements genannt (Sensornetze, mobile kontextsensitive Informationssysteme). Am Ende der Arbeit steht eine Zusammenfassung mit einem Ausblick.

Hintergrund der gesamten Arbeit ist das Knowledge Management und der Einsatz von Software in wissensintensiven Prozessen. Im Knowledge Management sind einige Konzepte nicht so trennscharf definiert, wie man sich das wünschen würde. Es existiert nur eine sehr unscharfe Definition von Wissen. So wird im KM die Notwendigkeit des impliziten Wissens betont und damit impliziert, dass IT-Systeme lediglich Hilfsmittel in wissensintensiven Prozessen sein können und niemals die Menschen ersetzen werden. Diese teilweise unscharfen Definitionen und die Unmöglichkeit, den Menschen aus den Systemen und Architekturen wegzudenken, führt teilweise zu einer leichten Unschärfe der Darstellung. Diese wurde bewusst in Kauf genommen, in der Hoffnung, mit der Arbeit einen interdisziplinären Ansatz vorlegen zu können, der helfen könnte einige erkennbare Akzeptanzprobleme zu beheben.

Diese Arbeit ist aber eine ingenieurtechnische. Gleichzeitig stand am Anfang die Erkenntnis, dass IT-Systeme wissensintensive Prozesse niemals automatisieren können. Das ist kein Widerspruch, sondern eine prinzipielle und notwendige Einsicht in die prinzipielle Leistungsfähigkeit von Technik. Jeder Versuch, ein System zur vollständigen Steuerung wissensintensiver Prozesse zu entwerfen, wurde daher vermieden. Implementierungen von Shark sind vielmehr Hilfsmittel und Ergänzungen für die Akteure in wissensintensiven Prozessen – nicht mehr, aber auch nicht weniger. Implementierungen von Shark sollen sich in die üblichen und bereits lange existierenden menschlichen Abläufe bei kreativen Arbeiten einfügen ohne sie ersetzen oder ändern zu wollen. Wenn diese Implementierungen Menschen noch etwas mehr von Routinetätigkeiten entlasten und damit noch ein Stück mehr Freiraum für schöpferisches Arbeiten geben, dann ist das Ziel dieser Arbeit erreicht.

1 Einleitung

2 Verwandte Arbeiten zu Wissensflüssen

In diesem Kapitel werden Arbeiten vorgestellt und diskutiert, die sich mit der Beschreibung von Wissensflüssen beschäftigen. Das Knowledge Management hat unterschiedliche Quellen. Diese Tatsache findet sich auch in diesem Kapitel wieder. So werden Arbeiten aus der Soziologie (Abschnitt 2.1) vorgestellt, der Informatik, konkret der Wissensrepräsentation (Abschnitt 2.2) und der Softwareagenten (Abschnitt 2.3). In Abschnitt 2.4 werden die wesentlichen Arbeiten des Knowledge Managements vorgestellt.

Die soziologische Sicht liefert Erkenntnisse über die Anatomie des Wissensaustausches. Es wird zu sehen sein, dass jedes System durch beschränkte Ressourcen gezwungen ist, die Menge an Wissen, die es empfängt bzw. abgibt zu beschränken. Die Filterung des Empfanges erfolgt anhand von Relevanzbewertungen. Die Filterung bezüglich des gesendeten Wissens erfolgt anhand von Strategien. Weder Relevanzbewertung noch Strategie sind statische Größen. Sie sind abhängig vom Kontext. Es wird ebenfalls zu sehen sein, dass soziale Gruppen hierarchisch strukturiert sein können. Es gibt damit Flüsse von Wissen innerhalb von Gruppen, nämlich zwischen Untergruppen und über Gruppen Grenzen hinaus. Ein weiteres Charakteristikum von Gruppen ist die Arbeitsteilung zwischen den Mitgliedern der Gruppe. Eine Arbeitsteilung existiert auch beim Versenden und beim Empfangen von Wissen.

Arbeiten in der Wissensrepräsentation werden zeigen, dass es zwei gegensätzliche Herangehensweisen an die Modellierung von Wissen gibt: das Konzept des *Weltwissens* und das der *Mikrotheorien*. Durch eine Verteilung des Weltwissen und eine Aggregation von Mikrotheorien lassen sich beide Ansätze einander näher bringen. Sie bergen allerdings Implikationen, die bei der Untersuchung von Knowledge Management Systemen relevant werden. So wird u. a. zu sehen sein, dass ein System, das auf dem P2P-Paradigma beruht, zwingend dem Konzept der Mikrotheorien folgen muss.

Die Übersicht über Softwareagenten dient vor allem der Abgrenzung. Der Ansatz von Agenten ist deutlich weitreichender als der von Knowledge Management Systemen. Ziel von Agenten ist die Modellierung menschlicher Nutzer und damit deren Ersetzung. Daraus ergibt sich die Notwendigkeit der Modellierung von Zielen, sozialen Verhalten, von mentalen Modellen usw. Diese Notwendigkeit existiert in KMS nicht. Es wird zu sehen sein, dass sich deutlich einfachere Konzepte für verteilte KMS finden lassen.

Der vierte Abschnitt gibt einen Überblick über wesentliche Arbeiten im Bereich Knowledge Management. Am Anfang steht ein Einstieg über die Arbeiten von Nonaka und Takeuchi, die Mitte der 1990er Jahre dieses Forschungsgebiet maßgeblich beeinflusst haben. Es wird vor allem das SECI-Modell vorgestellt werden. Es schließt sich eine Übersicht über die Arbeiten von Davenport und Prusak an. Sie folgen der Theorie von Nonaka und Takeuchi und zeigen anhand von praktischen Beispielen, wie sich das SECI-Modell in der Praxis zeigt.

Im Abschnitt 2.4.3 wird diskutiert, wie IT-Systeme prinzipiell das SECI-Modell unterstützen können. Dabei werden zwei wesentliche Erkenntnisse gewonnen werden. Einmal wird zu sehen sein, dass IT prinzipiell an einigen Stellen kaum unterstützend wirken kann. Zum Zweiten wird eine Kritik an den üblichen derzeit verfügbaren Knowledge Management Systemen (KMS) geführt werden. Ergebnis der Kritik wird vor allem auch ein Ruf nach einem verteilten KMS sein. Exemplarisch wird in Abschnitt 2.4.4 ein Ansatz von Bonifacio, Bouquet und Cuel vorgestellt werden, der als der fortgeschrittenste im Bereich der verteilten KMS (DKMS) gilt. Auch hier sind einige Schwachstellen zu erkennen, die diskutiert werden. Insbesondere wird zu sehen sein, dass das DKMS nach Bonifacio und anderen zwar zunächst wie ein Peer-Modell konzipiert wurde, dieses Konzept aber nicht konsequent zu Ende führt, was zu Inkonsistenzen führt.

Im Abschnitt 2.5 wird ein eigenes Modell vorgestellt. Es ist eine Konkretisierung des SECI-Modells und lässt Erkenntnisse aus der Systemtheorie sozialer Gruppen einfließen. Durch diese Kombination gelingt die Beschreibung eines Modells, in dem der Fluss von Wissen konsequent auf den Austausch zwischen Individuen zurückgeführt wird. Der Austausch zwischen Individuen, Gruppen, Organisationen und zwischen Organisationen wird auf ein Rollenmodell zurückgeführt. Dieses P2P-KM Modell wird in den folgenden Kapitel zu einer Softwarearchitektur weiterentwickelt.

2.1 Soziale Systeme

Ursprung und Nutzer jeglichen Wissens ist der Mensch.¹ Das Kommunikationsverhalten von Menschen wird u. a. in der Soziologie untersucht. In diesem Abschnitt soll ein soziologischer systemtheoretischer Ansatz vorgestellt werden, mit dem soziale Gruppen und deren Interaktionen mit der Umwelt und untereinander erklärt werden.

2.1.1 Gruppen

Von einer Gruppe bzw. einem sozialen System kann nach [Luh75] die Rede sein, wenn folgende Bedingungen erfüllt sind:

- Es existiert eine Beziehung zwischen wenigstens drei Personen.
- Es besteht Interesse an langfristiger Zusammenarbeit.
- Handlungsfähigkeit des Systems als System wird angestrebt.

Eine Gruppe besteht demnach aus wenigstens drei Personen, die darin übereinstimmen, dauerhaft miteinander zusammenzuarbeiten. Der letzte Punkt wird von Dunphy [Dun72] als *Gruppenidentität* bezeichnet. Das ist der Wunsch und der Fakt, dass Mitglieder der Gruppe für die Gruppe agieren und nicht mehr (nur) als Individuen.

Die Gruppenidentität beschreibt eine neue Qualität der Zusammenarbeit von Menschen: Es ist ein wesentlicher Unterschied, ob sich einige Menschen zusammenfinden,

¹Genauer: Ein intelligentes Wesen. Bei der Ankunft Außerirdischer muss die Liste der Wissensträger entsprechend erweitert werden.

um z. B. gemeinsam ein Projekt zu bearbeiten oder um gemeinsam und *dauerhaft* eine Struktur zu bilden, die einem Ziel folgt und die vor allem ein *Innen* und *Außen* kennt. Nach [Par59] existieren vier Grundfunktionen, die eine Gruppe erfüllt.

- Adaptation (Anpassung an die Umwelt)
- Goal-Attainment (Zielverwirklichung)
- Integration
- Latent Pattern Maintenance (dauerhafter Strukturerhalt)

Nach den Anfangsbuchstaben werden diese Funktionen auch als das *AGIL-Schema* bezeichnet. Die ersten beiden Funktionen arbeiten im Außenbezug der Gruppe. Die Adaption bezieht sich darauf, dass sich die Gruppe an verändernde Umweltbedingungen anpassen kann. Zielverwirklichung bedeutet, dass eine Konzentration auf ein Ziel vorliegt und verfolgt wird.

Die letzten beiden Funktionen beziehen sich auf Interna einer Gruppe. Der dauerhafte Strukturerhalt meint, dass eine Gruppe eine Struktur herausbilden muss und diese dauerhaft halten soll. Davon war bereits bei den Gruppeneigenschaften die Rede. Integration ist die Funktion, die bei auftretenden Widersprüchen und Spannungen innerhalb der internen Struktur für Ausgleich und Halt sorgt.

2.1.2 Gruppe als System

Ein Ansatz für ein Systemverständnis von Gruppen stammt von Niklas Luhmann [Luh84]. Er hebt die Notwendigkeit hervor, ein System immer in der Interaktion mit seiner Umwelt zu sehen. Sein Ansatz wird auch als *funktional-struktureller Ansatz* bezeichnet. Aus seiner Sicht „stabilisieren (Systeme) mithin eine Differenz zwischen sich und der Umwelt, zwischen Innen und Außen; sie bilden ein sinnhaftes, symbolisch vermitteltes Regulativ zwischen anfallender und jeweils verarbeitbarer Komplexität“. [Wil00]

Die Definition der Umwelt ist in diesem Falle ganz einfach. Umwelt ist alles, das nicht zur Gruppe gehört. Es sind also auch andere Gruppen Teil der Umwelt. Im Folgenden wird die Kommunikation zwischen Gruppe und Umwelt näher untersucht werden.

Zuvor aber eine weitere Begriffsklärung: Systeme werden als *selbstbewusste, strategische soziale Systeme* bezeichnet, wenn sie in der Lage sind, sich selbst als potenzielle Umwelt anderer System zu begreifen und anhand dessen geeignete Interaktionen abzuleiten [Wil00].

2.1.3 Umweltkomplexität und Kontingenz der Handlungsalternativen

Ein System nimmt seine Umwelt wahr und beeinflusst diese. Aus systemtheoretischer Sicht stellt sich die Umwelt als eine komplexe Struktur von Elementen dar, die das System zu erfassen versucht. Dabei wird unter Komplexität Folgendes verstanden:

„Komplexität bezeichnet den Grad der Vielschichtigkeit, Vernetzung und Folgelastigkeit eines Entscheidungsfeldes. Dabei bedeutet Vielschichtigkeit den Grad der funktionalen Differenzierung eines Sozialsystems und die Zahl der bedeutsamen Referenzebenen... Vernetzung heißt Art und Grad wechselseitiger Abhängigkeiten zwischen Teilen sowie zwischen Teil und Ganzem... Folgelastigkeit meint Zahl und Gewicht der durch eine bestimmte Entscheidung in Gang gesetzten Kausalketten oder Folgeprozesse innerhalb des in Frage stehenden Sozialsystems; und der Begriff Entscheidungsfeld weist darauf hin, dass es keine Komplexität an sich gibt, sondern nur in Hinsicht auf ein bestimmtes Problem, welches für ein bestimmtes Problem in einer bestimmten Situation Selektion erfordert.“ [Wil00]

Komplexität ist nun sowohl bei den Systemen als auch bei deren Umwelt zu beobachten. Auch hier besteht ein enger Zusammenhang zwischen beiden. Wie bereits Ross Ashby[Ash56] bemerkte, erhöht ein System seine Eigenkomplexität, wenn es sich an eine wachsende Umweltkomplexität anpasst.

Dazu soll ergänzend bemerkt werden, dass die Umweltkomplexität nicht als objektiv gesehen werden kann. Die Erkenntnis einer komplexen Umwelt bedarf eines Beobachters mit hinreichender Fähigkeit, die Umweltkomplexität auch zu erkennen. Anders gesagt: Ein einfaches, wenig komplexes System wird seine Umwelt auch nur wenig komplexer wahrnehmen als ein komplexeres System.

Was geschieht, wenn ein wenig komplexes System eine komplexere Umwelt betrachtet?

1. Es kann sie als keinen Regeln folgende wahrnehmen,
2. es kann eine höhere Komplexität vermuten und sich mit der Erkenntnis zufrieden geben, diese einfach nicht zu erkennen oder
3. aber es interpretiert die Umwelt weniger komplex, als sie ist, und so komplex, wie sie dazu in der Lage ist.

Alle drei Verfahrensweisen sind zu beobachten. Bei der ersten Variante wird die Welt als chaotisch und nicht-deterministisch wahrgenommen. Das führte in der Frühgeschichte zur Erfindung von Gottheiten, die für die sonst unerklärlichen Naturereignisse zuständig seien. Die zweite Variante bedeutet Einsicht in die eigene unzulängliche Erkenntnisfähigkeit. Hier gibt es zwei Lösungswege. Man kann sich damit zufrieden geben bzw. die Unwissenheit bzw. scheinbare Unerkennbarkeit zum Dogma erheben, wie das u. a. durch die mittelalterliche christliche Kirche geschah oder indem man versucht, Erkenntnis zu gewinnen. Sokrates' Ausspruch „Ich weiß, dass ich nichts weiß“ ist kein Ausruf der Resignation, sondern eine Aufforderung zum Nachdenken.

Der letzten Variante wird immer gefolgt, wenn ein Modell gebildet wird. Ein Modell hat gerade die Eigenschaft, dass es einmal aus einer Reihe von Beobachtungen abgeleitet und zum anderen für bestimmte Zwecke angewendet wird. Die Zweckbestimmtheit erlaubt, dass nicht alle Eigenschaften der Umwelt modelliert werden, sondern nur die, die dem Zweck dienen.

Systeme nehmen Umwelt nicht nur wahr, sie beeinflussen sie auch. Hier kommt der Begriff der Kontingenz der Handlungsalternativen ins Spiel, womit das Spektrum all

dessen gemeint ist, wie ein System auf seine Umwelt einwirken kann. Was tut z. B. eine Gruppe, wenn sie an einem unbekannten Ort einer anderen unbekannten Gruppe begegnet. Die Handlungsalternativen der frühen Steinzeit lagen sicherlich im Bereich zwischen dem Versuch des Kennenlernens über Ignorieren bis hin zum Angriff. In der modernen Neuzeit bieten sich differenzierter Möglichkeiten an, z. B. Austausch von Visitenkarten, Machen von Erinnerungsfotos, Aufbau diplomatischer Beziehungen, Konsultation des Reiseführers usw.

Hohe Komplexität korreliert nicht mit großer Kontingenz. Auch hoch komplexe Systeme, z. B. Behörden, können über nur wenig Handlungsspielräume verfügen, während auch wenig komplexe Systeme, wie die Beziehung zweier Menschen, ein großes Spektrum möglicher Handlungsalternativen aufweisen: Von Hass bis Liebe [Wil00]. Eingehend hat sich u. a. Dörner [Dör89] mit fatalen Effekten von zu einfachen Modellen bei der Beeinflussung komplexer Umweltsysteme beschäftigt ².

2.1.4 Limitierte Ressource

Jedes System verfügt nur über limitierte Ressourcen. Nur ein theoretisches System mit unbegrenzten Ressourcen wäre in der Lage, alle eingehenden Informationen zu verarbeiten und alle Handlungsalternativen zu prüfen.

Limitierung wird in diesem Zusammenhang nicht als rein quantitatives Limit verstanden, sondern auch als qualitatives. So können unzureichende Strukturen eine Verarbeitung verhindern: „Die tieferen Gründe einer 'Überflutung' – hoher Wissensinput gekoppelt mit geringer Fähigkeit zur Synthese – sind nicht in der Strategie der Subeinheiten, sondern eher in der Organisationsstruktur selbst zu finden.“ [Etz91]

Bei einer beliebig komplexen Umwelt entsteht bei der Beobachtung der Umwelt durch das System ein Problem: Welche Informationen der Umwelt sollen verarbeitet werden, welche nicht. Nach Willke filtert ein soziales System, indem es entscheidet, welche Informationen als *relevant* erachtet werden. Dabei ist die Relevanz keine Konstante, sondern abhängig vom inneren Zustand des Systems und von den angebotenen Informationen der Umwelt.

Beispielweise wird der Systemzustand einer Familie im Urlaub dergestalt sein, dass Informationen von der Arbeitsstelle eines Familienangehörigen nur in Ausnahmefällen als relevant angesehen werden. Umgekehrt ist ein Martinshorn eine Information aus der Umwelt, auf die nahezu immer reagiert wird.

Eine ähnliche Problematik entsteht bei den Handlungsalternativen. Nicht jede ist ausführbar, allein wegen der Endlichkeit der Ressourcen. Meist ist es auch nicht möglich, alle Konsequenzen jeder möglichen Handlung zu kalkulieren. Ein System muss sich anders behelfen, um handlungsfähig zu bleiben. Nach Willke bedient es sich einer Strategie, die es erlaubt, eine Handlungsalternative auszuwählen.

²Der Titel des Buches lautet sehr passend „Die Logik des Misslingens“ und führt an vielen (auch realen) Beispielen vor, wie bestens ausgebildete, mit den besten Vorsätzen versehene intelligente Menschen bei scheinbar überschaubaren Systemen in ihren Voraussagen und Einflussnahmen scheitern oder Schaden anrichten.

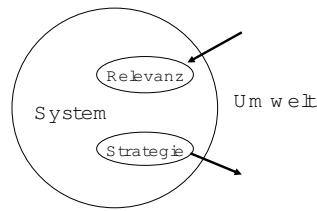


Abbildung 2.1: System und Umwelt

Diese Überlegungen fasst Willke zusammen: „Während also auf der Input-Seite eines sozialen Systems die Reduktion der Umweltkomplexität durch die begrenzten Informationsverarbeitungskapazitäten ... erzwungen wird, erfordert auf der Output-Seite des Systems die Knappheit der Ressourcen eine Auswahl aus den Handlungsalternativen. Dem entspricht ... Konflikte über Relevanz einerseits und Konflikte über Strategien andererseits.“ [Wil00]

Beide Konfliktlösungsprozesse laufen niemals gänzlich unabhängig voneinander ab. In der Individualpsychologie ist der Begriff der „kognitiven Strukturiertheit“ [Dör89] bekannt, in Gruppen gibt es eine Reihe von empirischen Belegen, dass die Auswahl einer Handlungsalternative Rückwirkung auf die Wahrnehmung der Umgebung hat. Intuitiv scheint dieser Zusammenhang ebenfalls klar: Sobald begrenzte Ressourcen in eine Handlungsalternative investiert werden, erfolgt gleichzeitig eine Justierung des Verständnisses darüber, was relevant ist und was nicht. Andersherum wird man eher dazu tendieren, Handlungen in eine Richtung zu unternehmen, die sowieso als relevant erachtet wird.

Abbildung 2.1 veranschaulicht den Zusammenhang zwischen einem System und seiner Umwelt. Das System ist als Kreis dargestellt. Der eingehende Pfeil symbolisiert die eingehenden Informationen. Innerhalb des Systems findet eine Filterung anhand der Relevanz statt. Gemäß einer Strategie kann das System auf die Umwelt einwirken, was durch einen hinausweisenden Pfeil symbolisiert wird.

Ein System kann nun aber nicht so gesehen werden, dass es eher passiv Umwelteinflüssen gegenübersteht, anhand von Relevanzfiltern Informationen verarbeitet und im Gegenzuge Handlungen vollführt. Ein System speichert vielmehr auch eine Reihe von Informationen, die es später wieder abrufen kann. Es wird in diesem Zusammenhang von der „partiellen Autonomie“ gesprochen, womit die Tatsache gemeint ist, dass ein System nicht nur auf Umwelteinflüsse reagiert, sondern durch bereits gespeicherte Informationen (Willke ([Wil00]) spricht von „Wissen“) eigene Präferenzen und Handlungsprinzipien entwickelt hat und diese immer wieder entsprechend anwendet. Diese Tatsache macht die Identität eines Systems aus und unterscheidet es maßgeblich von einem Prozess, der lediglich durch externe Faktoren bestimmt wird.

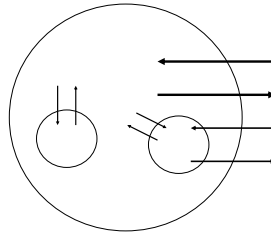


Abbildung 2.2: Teilsysteme und Teilmwelten

2.1.5 Teilsysteme und Teilmwelten

Systeme können Teil eines größeren Ganzen sein und damit zu einem Teilsystem werden. Beispiele dafür sind Arbeitsgruppen, die Teil einer Abteilung sind, die wiederum Teil einer Firma ist. Es empfiehlt sich, eine solche Unterscheidung auch bei der Umwelt vorzunehmen. Es wird in diesem Zusammenhang von *Teilmwelten* gesprochen [Wil00]. Tatsächlich stellt sich die Umwelt eines Gesamtunternehmens anders dar als die Umwelt einer Arbeitsgruppe. Beim Gesamtsystem Unternehmen wird eher an Marktkommunikation gedacht, an Cooperative Identity, an Kooperationen mit anderen Unternehmen usw. Das Teilsystem Arbeitsgruppe interagiert mit einer ganz anderen Teilmwelt, z. B. Arbeitsergebnissen anderer, die für die eigene Arbeit von Interesse sein könnten, neue Tools, die am Markt angeboten werden usw. Hinzu kommt bei den Teilsystemen, dass auch eine Kommunikation innerhalb des größeren Systems erfolgt, also mit anderen Arbeitsgruppen im Unternehmen, anderen Abteilungen usw.

Dunphy unterstreicht diesen Zusammenhang [Dun72]: „The group is not made of individual alone... it has structures and substructures and distinctive cultural pattern. Group interaction properly refers to the interrelation of these structures and patterns“.

Abbildung 2.2 stellt diesen Zusammenhang grafisch dar. Die Kreise symbolisieren Systeme, die Pfeile deuten jeweils die Erfassung der Umwelt (Pfeil in das System hinein) und die Beeinflussung der Umgebung (Pfeil heraus) dar. In der Abbildung sind zwei Teilsysteme symbolisiert, die in einem größeren System eingebunden sind. Das eine Teilsystem interagiert sowohl systemintern als auch mit der Umgebung, was durch zwei Pfeilpaare symbolisiert ist. Das andere Teilsystem interagiert nur mit der systeminternen Umgebung. Das Gesamtsystem interagiert sowohl mit seinen Teilsystemen als auch mit der Systemumgebung. Man kann sich ein Unternehmen mit zwei Abteilungen vorstellen, wobei nur eine auch eine Außenkommunikation betreibt.

2.1.6 Individuum als System

Gruppen bestehen aus Individuen. Wie zu sehen war, zeichnet sich eine Gruppe durch Funktionen aus, die im AGIL-Schema zusammengefasst worden sind. Wie verhält es sich, wenn dieses Schema einmal auf ein Individuum angewendet wird?

AGIL steht für Adaption, Goal-Attainment, Integration und Latent Pattern Maintenance. Menschliche Individuen sind ohne Zweifel in der Lage, sich an ändernde Umgebungsverhältnisse anzupassen. Ebenfalls kann ohne Weiteres jedem menschlichen Individuum die Fähigkeit des zielgerichteten Handelns zugestanden werden. Integration und permanenter Strukturerhalt bezogen sich darauf, dass eine Gruppe eine interne Struktur schafft und diese dauerhaft erhält. Diese Eigenschaft ist per Individuen weniger auf einer sozialen als mehr auf einer biologischen und psychologischen Ebene zu untersuchen. Zweifelsfrei erhalten aber Individuen ihre Struktur über ihre Lebensdauer und nutzen dazu (biologische und psychische) Methoden der Integration.

Für die Bewertung einer Gruppe in seiner Außenwirkung waren aber vor allem auch die beiden ersten Funktionen von Interesse (Anpassung und Zielerreichung). Es kann festgestellt werden, dass bei dieser eingeschränkten Sichtweise deutliche Gemeinsamkeiten zwischen Individuen und Gruppen vorliegen.

Das Gleiche gilt auch für die Beobachtung der Umwelt. Auch Individuen verfügen nur über limitierte Ressourcen zur Beobachtung der Umwelt (ca. 100 Millionen Sinneszellen und ca. 10.000 Milliarden Synapsen [Fr85]). Auch hier findet eine Reduzierung der eingehenden Informationen anhand einer Bewertung der Relevanz statt.

Analoges gilt für die Kontingenz der Handlungsalternativen. Individuen wie Gruppen verfügen über limitierte Ressourcen zur Interaktion mit der Umwelt. Auch hier muss jeweils anhand der aktuellen Gegebenheiten entschieden werden, welche Handlungsalternative anzuwenden ist. Diese Methode kann ebenfalls als Strategie bezeichnet werden.

Im Rest dieser Arbeit werden sowohl Individuen also auch Gruppen als soziale Systeme aufgefasst.

2.1.7 Zusammenfassung

In diesem Abschnitt wurde ein systemtheoretischer Ansatz zur Beschreibung sozialer Gruppen präsentiert und es war zu sehen, dass sowohl Gruppen als auch Individuen als soziale Systeme interpretiert werden können.

Soziale Systeme empfangen aus ihrer Umgebung Informationen, die sie anhand einer Relevanzbewertung filtern. Sie können die Umgebung verändern und wählen anhand einer Strategie aus, welche Handlungen sie vollziehen.

Alle Individuen haben die Eigenschaft, sich als Umwelt anderer Individuen und Gruppen wahrzunehmen und entsprechend zu handeln. *Selbstbewusste strategische soziale Systeme* haben die gleiche Eigenschaft und sind damit für diese Arbeit von Interesse, die sich mit bewussten Teilnehmern in wissensintensiven Prozessen beschäftigt.

Soziologische Modelle erklären natürlich nicht, *wie* Menschen und Gruppen Daten, Informationen bzw. Wissen speichern. Es kann lediglich davon ausgegangen werden, *dass* sie das tun. Soziale Systeme haben weiterhin die Eigenschaft, dass sie ineinander enthalten sein können, das heißt, dass es Subsysteme gibt, die in einem übergeordneten System als Teil vorliegen und dieses Supersystem als Teilumwelt wahrnehmen.

Einige Erkenntnisse der vorgestellten Arbeiten werden für das P2P-KM-Modell in Abschnitt 2.5 einfließen. Das sind die Wege der Informationsfilterung über Relevanz und Strategie und die Erkenntnis, dass sich soziale Systeme zu größeren Systemen zu-

sammenschließen können und dabei innere und äußere Systemumgebungen entstehen, wobei die Kommunikationsprinzipien über diese Grenzen hinaus die gleichen bleiben. Kurz gesagt, wird der systemtheoretische Ansatz, der hier vorgestellt wurde, mit dem SECI-Modell von Nonaka und Takeuchi vereint werden.

2.2 Wissen und Kontext

Dieser Abschnitt geht der Frage nach, was Wissen und Kontext sind. Die Frage wird keinesfalls vollständig beantwortet werden können, beschäftigen sich allein schon Philosophen seit nunmehr wenigstens zweieinhalb Jahrtausenden mit der Erkenntnis, also dem Wissen. Tatsächlich reißt dieser Abschnitt einige Ansätze an, ohne sie im Detail vorzustellen. Das ist an dieser Stelle auch nicht nötig, an der es um einen Überblick und allgemeine Gemeinsamkeiten geht.

Im Wörterbuch der deutschen Sprache von Wahrig ([Wah91]) findet man unter Wissen: „Kenntnisse, Gelehrsamkeit; Kenntnis, Bewusstsein (von etwas)“. Dieses Verständnis von Wissen wird in diesem Abschnitt gelten. Wissen hat mehrere Facetten. Wissen ist Kenntnis über etwas. Dieses Etwas gilt es zunächst zu benennen. Begriffssysteme sind ein Mittel, um diese Aufgabe zu erfüllen. Es bestehen Zusammenhänge, die unter anderem in Ontologien beschrieben werden können.

Wissen kann auf unterschiedlichste Arten repräsentiert werden. Die noch immer gebräuchlichste Form der Wissensrepräsentation ist die menschliche Sprache, gesprochen oder geschrieben. Daneben gibt es Bilder, logische Ausdrücke und semantische Netze. Jedes Repräsentationsformat hat seine Vorzüge gegenüber anderen. Es wird zu sehen sein, dass es ein *bestes* Format nicht gibt.

Wissen steht immer in einem Kontext. Dieser Aussage kann sicherlich intuitiv zugestimmt werden, wenn man sich beispielweise den Ausspruch „Gute Erde!“ anschaut. Ausgesprochen von einem frühmittelalterlichen Bauern ist sicherlich etwas anderes gemeint, als wenn ihn ein Kosmonaut tätigt, der in einer erdnahen Umlaufbahn auf unseren Planeten schaut. Der Kontext ermöglicht zu verstehen, dass sich der eine vermutlich eher auf die Qualität des Bodens bezieht, während der andere wahrscheinlich eher seiner insgesamt positiven Einstellung gegenüber unserem Heimatplaneten Ausdruck verleihen will.

In vielen KI-Systemen ist die Ermittlung des Wahrheitswertes von Aussagen, also von Wissen von Interesse. Bei Systemen für das Knowledge Management ist eher interessant, Wissen aufzufinden, das bei einer Problemlösung hilft. Kontext ist für diese und andere Anwendungsfälle ein erprobtes Mittel der Strukturierung. Es werden eine Anzahl von Beispielen aus der Literatur vorgestellt, wie eine Strukturierung mit Prädikatenlogik aber auch in Informationssystemen erfolgen kann. Die Grundideen sind die gleichen: Es werden Kontexte definiert, die eventuell auch Hierarchien bilden können, und denen wird Wissen zugeordnet.

Prinzipiell gibt es zwei mögliche Vorgehensweisen bei der Strukturierung: den Top-Down- und den Bottom-Up-Ansatz. Vor allem in der Erkenntnisphilosophie ist der erste Ansatz der wesentliche. Ziel vieler Arbeiten war die Schaffung eines Systems, in dem

das Wissen der gesamten Welt gesammelt werden kann. Mit Resultaten dieser Arbeiten agieren wir nahezu täglich, es sind z. B. Enzyklopädien. Der Bottom-Up-Ansatz wird für kleinere praktische Probleme nahezu ausschließlich angewendet: Nur für einen eingeschränkten Diskursbereich wird Wissen gesammelt.

Beide Ansätze werden in Beispielen vorgestellt und miteinander verglichen. Es wird untersucht werden, welcher Ansatz für P2P-Systeme, um die es in dieser Arbeit zentral geht, geeignet ist.

Dieser Abschnitt ist wie folgt strukturiert. Es werden zunächst Begriffssysteme genannt und kurz beschrieben (Abschnitt 2.2.1). Solche Systeme erlauben die Bezeichnung von Gegenständen, über die weiteres Wissen gesammelt werden soll. Ontologien ist ein eigener kurzer Abschnitt gewidmet (2.2.2), der von der bereits genannten Diskussion Top-Down versus Bottom-Up gefolgt wird (Abschnitt 2.2.3).

Danach folgt ein Überblick über die Formen der Wissensrepräsentation und die Rolle des Kontextes darin (Abschnitt 2.2.4). Beispielhaft werden aus den Bereichen Prädikatenlogik (Abschnitt 2.2.5) und Content-Management-Systeme (Abschnitt 2.2.6) Ansätze vorgestellt, wie Wissen mittels Kontext strukturiert werden kann. Auch dieser Abschnitt wird mit einer Zusammenfassung abgeschlossen.

2.2.1 Begriffssysteme

Das semiotische Dreieck

Wissen ist Erkenntnis über *etwas*, das die Aufmerksamkeit eines Betrachters auf sich gezogen hat. Das *Etwas* wird üblicherweise *Gegenstand* genannt. Das ist nicht zwingend ein physischer Gegenstand, auch eine Meinung, ein Gedanke kann der *Gegenstand* weiterer Betrachtung sein.

Ein Gegenstand an sich kann nicht gedacht werden. Vielmehr ist es so, dass sich ein Mensch einen *Begriff* von einem Gegenstand macht. Er erzeugt also eine Vorstellung des Gegenstandes.

Um nun diesen Begriff und damit indirekt den Gegenstand zu benennen, benutzen Menschen *Bezeichnungen*. Damit sind alle drei Bestandteile beisammen, die gemeinsam das *semiotische Dreieck* bilden, siehe Abbildung 2.3.

Links unten ist der Gegenstand. Er ist etwas, auf das die Aufmerksamkeit eines Beobachters gerichtet ist. Der Beobachter entwickelt anhand der Beobachtung einen Begriff vom Gegenstand. Dieses Resultat kann und wird sich wahrscheinlich sogar zwischen verschiedenen Beobachtern unterscheiden. Um über einen Begriff zu diskutieren muss wiederum eine Bezeichnung genutzt werden. Das Verhältnis zwischen der Bezeichnung und dem Gegenstand ist folgende: Eigentlich soll sie einen Gegenstand bezeichnen. Tatsächlich bezeichnet sie aber den Begriff, den sich derjenige gemacht hat, der den Gegenstand gerade bezeichnet.

Ein Beispiel: In Berlin bezeichnet man ein rundes etwa faustgroßes hellbraunes Gebäck, das mit Zucker bestreut ist und mit Marmelade bzw. Pflaumenmus gefüllt ist, als „Pfannkuchen“. In anderen Teilen der Republik ist die Bezeichnung „Berliner“ üblich. „Pfannkuchen“ wiederum ist in anderen Gebieten die Bezeichnung für einen flachen Fladen, der

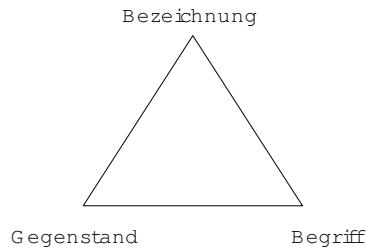


Abbildung 2.3: Das semiotische Dreieck

aus Mehl, Eiern und Milch hergestellt und gebraten wird, den man in Berlin aber eher als „Eierkuchen“ bezeichnen würde. Schaut man nicht genau hin, so kann man einen Eierkuchen auch mit einem Crêpe verwechseln.

Das semiotische Dreieck erlaubt die Erläuterung mehrerer potenzieller Missverständnisse: Es gibt gleiche Bezeichnungen für die unterschiedliche Begriffe. Das sind *Homonyme*. Im Beispiel war das der „Pfannkuchen“. Es steht in Berlin für das Gebäck, in anderen Gebieten für den Fladen. Andere Bezeichnungen sind zwar unterschiedlich, verweisen aber auf den gleichen Begriff. Das sind *Synonyme*. Im Beispiel waren das wiederum der „Pfannkuchen“ in der Berliner Diktion und der „Berliner“ in anderen Gebieten der Republik. Weitere Missverständnisse können über Homophone entstehen, das sind lautgleiche Bezeichnungen für unterschiedliche Begriffe, die aber unterschiedlich geschrieben werden.

Ein weiteres Problem birgt sich in Folgendem: Eine falsche Beobachtung kann zu einer falschen Begriffsbildung und damit zu einer falschen Bezeichnung führen.

Was ist also gemeint, wenn jemand von „Pfannkuchen“ spricht? Es ist leider völlig unklar ohne den Kontext der Aussage zu kennen. Weiß man aber beispielsweise, dass der Sprecher *nicht* die übliche Diktion aus Berlin benutzt, ist man schon einen Schritt weiter. Ist also ein Eierkuchen gemeint?

Das kann man auch noch nicht schließen, denn der Sprechende kann sich einen durch ungenaues Hinsehen erzeugten falschen Begriff gebildet haben und der Gegenstand ist tatsächlich ein Crêpe.

Alle menschlichen Sprachen sind missverständlich in diesen Formen. Diese Erkenntnis zeitigte in der Geschichte zwei Resultate: Einmal entstanden Mythen um eine legendäre vollkommene Sprache, die von einem göttlichen Schöpfer gegeben und unmissverständlich gewesen, aber wegen der babylonischen Sprachverwirrung verschwunden wäre, um die Menschen zu strafen. Der zweite Effekt waren Versuche, solche vollkommenen Sprachen zu konstruieren. Diese Versuche waren nicht erfolgreich, führten aber auch zur Entwicklung von Sprache der Wissensrepräsentation, siehe dazu [Eco94] und insbesondere den Abschnitt über die Kabbalistische Pansemiotik und die Sprachen der künstlichen

Intelligenz.

Das semiotische Dreieck erlaubt, Probleme in der Begriffsbildung zu beschreiben. Im Folgenden werden einige Ansätze erklärt, die dazu dienen sollen, diese Probleme zu vermeiden.

Kategorien

Entitäten können anhand ihrer Eigenschaften beschrieben werden. So könnte man sagen, dass ein Baum *holzartig* in seiner Substanz ist und *länglich und nach oben strebend* in seiner Qualität. Man kann auch sagen, dass ein Hammer *einwirkend* in seiner Aktivität ist, genauso wie ein Amboss eine Einwirkung *erleidet*.

Dieser Ansatz wurde bereits von Aristoteles in seiner Kategorienlehre verfolgt. Er nennt zehn Kategorien³. Durch die Beschreibung jeder Kategorie einer Entität ist letztere nach Aristoteles beschreibbar.

Dieser Ansatz ist Basis von Klassifikationssystemen. In einem solchen System werden Gruppen von Entitäten zu Klassen zusammengefasst. Klassennamen bilden ein kontrolliertes Vokabular. Klassen können hierarchisch strukturiert sein, d. h. Ober- und Unterklassen besitzen. Dabei kann zwischen Monohierarchien und Polyhierarchien unterschieden werden. In Ersterer existiert je Klasse maximal eine Oberklasse, in der Zweiten gilt diese Restriktion nicht. Auf Details solcher Systeme soll nicht weiter eingegangen werden. Informatikern sind Klassifikationen u. a. aus dem Objektorientierten Design (OOD) wohl vertraut, wo Klassen von Objekten definiert werden.

Weiter hinten wird bei der Vorstellung verschiedener Ansätze zur Beschreibung von Kontexten zu sehen sein, dass sie u. a. auch den Kategorien- bzw. Klassifikationssystemen ähneln. Für Beispiele von Klassifikationssystemen, siehe u. a. [Sow00].

Thesauri

Die Bezeichnung Thesaurus hat ihren Ursprung in dem griechischen *thesauros*, was so viel wie Schatz bzw. im Lateinischen wie Schatzkammer bedeutet. Im Mittelalter wurden wissenschaftliche Sammelwerke, vor allem Wörterbücher, Enzyklopädien, als Thesaurus bezeichnet [SM94]. Daraus lässt sich erklären, dass unter einem Thesaurus heute ein spezielles Begriffssystem verstanden wird.

Weiter oben wurde bereits gezeigt, dass die menschliche Sprache missverständlich ist. Eine eindeutige Terminologie ist aber für das Arbeiten, insbesondere das wissenschaftliche Arbeiten, unerlässlich. Daher verwundert es nicht, dass Begriffssysteme wie der Thesaurus nicht nur genutzt, sondern auch standardisiert wurden. Es existiert u. a. der deutsche Standard [DIN87] und ein ISO Standard [ISO86a]. In Erstem ist ein Thesaurus definiert als „eine geordnete Zusammenstellung von Begriffen und ihren (vorwiegend natürlichsprachlichen) Beziehungen, die in einem Dokumentationsgebiet zum Indexieren, Speichern und Wiederauffinden dient.“

³Substanz, Quantität, Qualität, Relation, Ort/Raum, Zeit, Lage, Haben/Besitz/Gewohnheit, Wirken/Aktivität/Tun, Leiden/Passivität/Erleiden

Ein Thesaurus definiert eine eindeutige Abbildung einer Bezeichnung auf einen Begriff. Weiterhin bietet er die Möglichkeit, Synonyme zu erfassen und Homonyme und Polyseme kenntlich zu machen.

Es war bereits zu sehen, dass die menschliche Sprache reich an Synonymen ist. In einem Thesaurus wird für jeden Begriff eine Vorzugsbezeichnung definiert, ein *Deskriptor*. *Nicht-Deskriptoren* können bei einem Deskriptor platziert werden und erlauben den Verweis auf ihren Deskriptor. Damit lassen sich u. a. Synonyme beschreiben (z. B. Heirat (Deskriptor) → Synonym (Nicht-Deskriptor) → Eheschließung (Deskriptor)). Es lassen sich aber auch Beziehungen zwischen Homonymen und Über- bzw. Unterbegriffen definieren.

In den genannten Standards sind Beziehungen definiert. So kennt die DIN: BF - Benutzt Für, BS - Benutze Synonym, OB - Oberbegriff, UB - Unterbegriff, VB - Verwandter Begriff und D - Definition. Letzteres erlaubt eine Definition des Begriffes, der durch den Deskriptor bezeichnet wird. Die ISO kennt analog dazu: UF - Used For, USE/SYN Use Synonym, BT - Broader Term, NT - Narrower Term, RT - Related Term, SN - Scope Note.

Ein Thesaurus ist eine Strukturierung von Begriffen. Durch das Konzept der Deskriptoren ist für jeden Begriff eine eindeutige Vorzugsbezeichnung definiert. Beziehungen zu anderen Begriffen lassen sich durch Nicht-Deskriptoren definieren.

Eine Taxonomie kann als ein vereinfachter Thesaurus interpretiert werden. In einer Taxonomie werden Bezeichnungen lediglich in einen hierarchischen Zusammenhang gestellt. Das Konzept der Nicht-Deskriptoren fehlt.

2.2.2 Ontologien

Die Ontologie ist die „Lehre des Seiens“. Das Wort ist aus dem Griechischen abgeleitet. Die grundsätzliche Frage der Ontologie lässt sich so formulieren: Was ist? Eine nachgeordnete Frage ist: In welchem Verhältnis steht es zu anderem?

Es hat sich das Wort *Entität* zur allgemeinen Benennung von etwas eingebürgert, das *ist*. Das Wort ist abgeleitet von dem lateinischen *ens*, was so viel wie *Seien* bedeutet. Einige Entitäten können durch Menschen direkt beobachtet werden. Locke nennt diesen Vorgang *Sensation*. Andere können durch Nachdenken entstehen (*Reflexion*).

Das Wort Ontologie wird in der Informatik und vor allem der künstlichen Intelligenz eingeschränkter benutzt. Gruber definiert eine Ontologie als die „explizite Spezifikation eines Diskursbereiches“ [Gru93]. Eine *explizite Spezifikation* ist etwas, das außerhalb des menschlichen Geistes (explizit) existiert und Formalien genügt (Spezifikation). Ein Diskursbereich ist ein Gebiet, das für die Autoren einer Ontologie aus einem beliebigen Grund von Interesse ist.

Zur Repräsentation von Ontologien können unterschiedlichste Sprachen eingesetzt werden. Logische Ausdrücke sind ein erprobtes Mittel. Hierarchien und Tabellen wurden seit Aristoteles gern zur Notation benutzt. Allgemeiner werden auch Graphen eingesetzt, wobei die Knoten Entitäten und die Kanten die Relationen zwischen ihnen darstellen. Auch Hierarchien sind durch Graphen gut darstellbar. Auf einige Beispiele wird in Abschnitt 2.2.4 eingegangen werden.



Abbildung 2.4: Eine Ontologie: Leser liest Dissertation

Eine Ontologiesprache per se gibt es nicht. Manchmal genügen auch einfache Skizzen, wie in Abbildung 2.4 dargestellt. Die Entität *Leser* ist verbunden mit der Entität *Dissertation* durch eine (gerichtete) Relation, die den Name *liest* trägt. Der Diskursbereich dieser Ontologie ist sehr eingeschränkt: Er ist das, was Sie gerade mit dieser Arbeit tun.

Einen ausgezeichnete und sehr ausführliche Übersicht zum Thema findet sich in [Sow00]. Unter anderem in [NM01, CC04] wird beispielhaft die Erstellung einer Ontologie diskutiert.

2.2.3 Weltwissen versus Mikrotheorien

Die Wahl des Diskursbereiches lässt eine grobe Unterteilung der Bildung von Ontologien zu. Philosophen tendieren dazu, als Diskursbereich alle denkbaren Entitäten zu sehen, und konzentrieren sich auf allgemeinste Konzepte, von denen man abgeleitet auf konkretere Fragestellungen schließen kann. Das kann als ein Top-Down-Ansatz bezeichnet werden [Sow00].

Eher praxisorientierte Anwendungen wählen als einen Diskursbereich eine Fachdomäne, die es zu bearbeiten gilt. So kann es sinnvoll sein, eine Ontologie für Wein zu erstellen, wie es in [NM01] erfolgte. Natürlich kann eine solche Ontologie verallgemeinert werden, im Sinne, dass der Diskursbereich erweitert wird, z. B. im nächsten Schritt auf alle möglichen Getränke. Dieses Vorgehen kann man als Bottom-Up-Ansatz bezeichnen.

Beim Top-Down-Ansatz spricht man auch davon, dass die Ontologie *Weltwissen* beschreibt. Der Ansatz geht nämlich davon aus, dass es möglich ist, zunächst die abstraktesten denkbaren Konzepte zu ermitteln und von ihnen ausgehend spezialisierte Entitäten einführen zu können. Der Top-Down-Ansatz geht von der Möglichkeit aus, dass es *eine allgemeine Ontologie* geben kann, in der alle denkbaren Entitäten und ihre Beziehungen untereinander eingeordnet werden können.

Beim Bottom-Up-Ansatz spricht man auch von *Mikrotheorien* oder *Mikrowelten*. Der Ansatz geht davon aus, dass zunächst Entitäten gesammelt werden, die nicht allgemein sind, sondern eher konkret. Basierend auf diesen wäre dann eine Verallgemeinerung

möglich. Dieses Vorgehen entspricht der im Software-Engineering sehr verbreiteten Objektorientierten Analyse. Auch hier werden zunächst allgemeine Objekte des Diskursbereiches identifiziert. Davon ausgehend werden Klassen geschaffen, die von den konkreten Objekten abstrahieren.

Beide Ansätze bergen aber jeweils ein prinzipielles Problem. Das Weltwissen ist potenziell enorm groß, vermutlich unendlich groß. Für eine praktische Nutzung ist eine Zerlegung notwendig. Für die Mikrotheorien existiert das umgekehrte Problem: Da tendenziell Mikrotheorien in verschiedensten Diskursbereichen und unterschiedlichsten Abstraktionsniveaus entstehen, ist es sinnvoll, Methoden zu finden, die zwei Mikrotheorien vereinen bzw. vergleichbar machen. Ohne solche Methoden würden Ontologien nicht in anderen Ontologien wiederbenutzt werden können. In den folgenden beiden Abschnitten wird sich diesen beiden Problemen zugewandt. In beiden Fällen kann jeweils ein bestimmtes Verständnis von Kontext helfen, die Probleme zu bearbeiten.

Partitioniertes Weltwissen durch Kontext

Im Ansatz *Weltwissen* wird davon ausgegangen, dass beginnend bei allgemeinsten Entitäten eine Konkretisierung vorgenommen werden kann, die schlussendlich dazu führt, dass alles (gedachte und noch gedachte) Wissen der Welt in diese eine Ontologie eingefügt werden kann. Dabei ist irrelevant, ob dieses Vorgehen als praktische vollständig realisierbar eingeschätzt wird. Es geht darum, ob diese prinzipiell Herangehensweise gewählt werden soll. Beispiele dieses Ansatzes werden weiter hinten vorgestellt.

Eine solche Ontologie wird natürlich unendlich groß werden. Für praktische Anwendungen gilt es, diese in handhabbare Stücke zu zerlegen. Dabei kann eine Form von Kontext eine wichtige Rolle spielen: „Context provides a principled way to cluster, partition and organize knowledge and its dimensions.“ [Bre99b].

Die Partitionierung kann beliebige Ziele haben, z. B. die Partitionierung nach dem Inhalt einer Wissensbasis und nach der Art der Beschreibung der Fakten, z. B. deren Detailtiefe [BGGB01]. Beliebige andere Ziele der Partitionierung sind definierbar. Anhand der Abbildung 2.5 sollen einige Eigenschaften dieses Ansatzes erläutert werden.

In der Abbildung sind die Kontexte A, B, C und D zu sehen, wobei B und C Teile von A sind und D Teil von C⁴. Der dunkle Kreis W sei Wissen, das dem Kontext C zugeordnet ist. Es sind außerdem drei Funktionen durch Pfeile dargestellt, die über W ausgeführt werden bzw. wurden. Die Akquisition soll den Vorgang bezeichnen, wie das Wissen erzeugt wurde und dabei einem Kontext zugeordnet wurde. Aufstieg bedeutet, dass das Wissen von einem konkreteren Kontext in einen allgemeineren überführt wird. Abstieg bedeutet den umgekehrten Weg, also die Überführung von Wissen von einem allgemeineren Kontext in einen konkreteren. Wechsel soll die Verschiebung von Wissen in einen anderen bezeichnen, die nicht in einem hierarchischen Verhältnis zueinander stehen.

Das Aufsteigen von Wissen in einen allgemeineren Kontext bedeutet auch, dass das Wissen allgemein gültiger wird, da der Kontext, in dem es gilt, beständig allumfassender

⁴Zur Veranschaulichung kann man sich unter A „das 20. Jahrhundert“ vorstellen. B könnten „die Goldenen Zwanziger“ sein, C könnte „die Zeit nach 1945“ und D „die Zeit nach 1989“ sein.

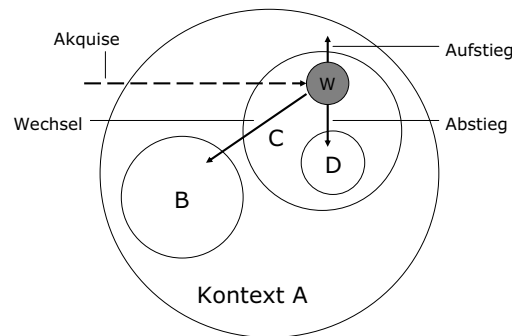


Abbildung 2.5: Partitioniertes Weltwissen

wird. Wie weit kann ein solches Aufsteigen erfolgen? Anders: Gibt es einen äußersten, allgemeinsten Kontext? McCarthy begründet in [McC93], warum es diesen nicht geben kann. Er argumentiert wie folgt: Angenommen, er formulierte die Aussage, dass ein Buch *auf* einem Tisch läge. Ein Kritiker könnte meinen, dass nicht klar sei, was *auf* bedeutet, und dass *auf* ohne Schwerkraft keinen Sinn mache. Es gälte also, den Kontext klarer zu definieren, indem *auf* eine klarere Bedeutung erhält. Ist das geklärt, könnten sich Diskussionen über die notwendige Schwerkraft anschließen oder Debatten darüber, wann etwas als ein Buch gilt oder nur als Broschüre. Wahlweise könnte auch diskutiert werden, was einen Tisch ausmacht. Die Diskussionen könnten über Formen, Materialien und alles weiter Denkbare geführt werden. McCarthy kommt zu dem Schluss, dass es nicht gelingen wird, eine Aussage zu tätigen, die allgemein gilt. Jede Aussage gilt nur in einem konkreten Kontext, der einschränkende Erklärungen macht über die Entitäten, über die Aussagen gemacht werden. Anders gesagt: „There isn’t simply a most general context.“

Semantisch wäre ein solcher allgemeiner, äußerster Kontext auch höchst kompliziert. Was sagt diese äußerste, allgemeinste Hülle aus? Nichts anderes als „gilt immer“. Dieser „Kontext“ bezeichnet faktisch die Kontextfreiheit[BGGB01]. Das heißt aber auch, dass es Wissen gäbe, das außerhalb jedes Kontextes existiert.

In dieser Arbeit wird prinzipiell die Ansicht vertreten, dass es kein kontextfreies Wissen geben kann. Weiter unten wird zu sehen sein, dass diese Meinung im Bereich des Knowledge Management geteilt wird. Im Bereich der künstlichen Intelligenz gibt es aber unterschiedliche Ansichten darüber. Es gibt Systeme, die von einem allgemeinen Kontext ausgehen, andere schließen diesen aus (Abschnitt 2.2.5).

Aggregierte Mikrowelten

Eine Mikrowelt (oder Mikrotheorie) ist eine Ontologie, die für einen eingeschränkten Bereich Gültigkeit hat. Abbildung 2.6 stellt schematisch drei solcher Mikrowelten dar

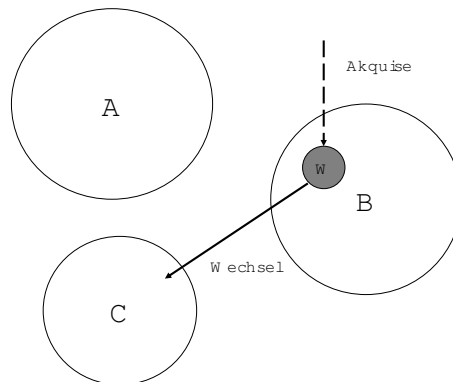


Abbildung 2.6: Aggregierte Mikrowelten

(A, B und C). In C ist exemplarisch Wissen (W) eingezeichnet. Wie jede Ontologie beschreiben auch die Mikrowelten Entitäten für einen eingeschränkten Diskursbereich. Der wesentliche Unterschied zum Ansatz Weltwissen besteht nun darin, dass eine Mikrowelt *nicht* als ein Teil einer übergeordneten und allgemeineren Ontologie verstanden wird. Sie ist autonom und in keiner Hierarchie implizit oder explizit enthalten. Die Frage nach einem äußeren Kontext stellt sich dadurch nicht. Akquiriertes Wissen wird einem Kontext zugewiesen. Wissen kann den Kontext wechseln, allerdings nicht in Form einer Verallgemeinerung oder Konkretisierung, sondern nur in der Form, dass es einen Kontext verlässt und in einen anderen übernommen wird.

Diese Übernahme ist nicht ohne weiteres möglich. Da zwei Kontexte unabhängig voneinander sind (in der Abbildung B und C), können sie ohne weitere Einschränkungen nicht direkt miteinander verglichen werden. Kann allerdings kein Vergleich erfolgen, so kann auch Wissen nicht einfach den Kontext wechseln und dort noch immer Gültigkeit haben. Dieses Problem ist im Bereich der *shared ontologies* (siehe u. a. [Hef01]) wohl bekannt.

Prinzipiell gibt es folgende Lösungsmöglichkeiten:

- Es wird auf ein hierarchisches Modell gewechselt und ein dritter Kontext (er möge X heißen) eingefügt, in dem die beiden Kontexte (B und C) enthalten sind. Ein Wechsel von W aus C nach B bedeutet einen Aufstieg in den Kontext X und einen nachfolgenden Abstieg in den Kontext B.⁵

Das Problem wird auf die Erstellung eines Kontextes X verschoben, der die beiden Kontexte beinhaltet.

⁵Grafisch ähnelt die Situation der Abbildung 2.5, wobei A in X umbenannt werden muss und D nicht vorhanden ist.

- Es wird eine Mediator-Funktion eingeführt, die die Eigenschaft hat, dass sie Wissen W aus einem Kontext C so bearbeitet, dass es in Kontext B gültig ist.

Es ist abhängig von der Anwendung, dem Wissen und den Kontextarten, welche Variante zum Einsatz kommt und ob überhaupt eine einsetzbar ist. Beide Ansätze führen aber zu einer Aggregation der Kontexte, denn es ist eine Möglichkeit geschaffen worden, Wissen zwischen zwei (bisher getrennten) Kontexten auszutauschen. Beispiele dieses Ansatzes werden weiter hinten vorgestellt werden.

Ontologie für autonome Peers

Diese Arbeit beschäftigt sich mit autonomen Peers und Wissensflüssen in spontanen Netzen solcher Peers. Autonomie wurde im ACP-Modell so aufgefasst, dass Peers ihre Daten allein verwalten können und eigenständig Entscheidungen treffen können, ob und was sie mit anderen Peers austauschen wollen. Die Forderung nach Autonomie schließt bereits den Ansatz Weltwissen aus.

Dieser Ansatz verlangt, dass alles Wissen zentralen Konzepten untergeordnet wird. Das Einfügen neuen Wissen muss immer begleitet sein mit Maßnahmen zur Sicherstellung der Konsistenz des Weltwissens. Das erzwingt bei jeder Änderung der Wissensbasis eines Peers eine Kommunikation mit anderen. Ein Einwand könnte sein, dass es im Bereich verteilter Datenbanken optimistische Lockingstrategien gibt, in denen Kopien zentraler Datenbestände durchaus auch dezentral geändert werden können. Diese Strategien erlauben zwar ein sofortiges Ändern, verschieben den Zeitpunkt einer Abstimmung aber nur in die Zukunft. Die Notwendigkeit einer Abstimmung mit anderen Entitäten bleibt bestehen. Die Notwendigkeit der Abstimmung widerspricht der Autonomie eines Peers.

Der Ansatz der aggregierten Mikrowelten ist dagegen für autonome Peers geeignet. Jedes Peer kann autonom Daten und Wissen im eigenen Kontext verwalten und entscheiden, wann es Daten oder Wissen austauschen will. Beim Austausch entsteht das Problem, dass die Kontexte der Kommunikationspartner in irgendeiner Form vergleichbar gemacht werden müssen. Die zwei prinzipiellen Wege wurden weiter oben vorgestellt. Ob und welcher Weg beschritten wird, ist autonome Entscheidung des Peers.

Tatsächlich werden Arbeiten aus dem Bereich Knowledge Management zeigen (Abschnitt 2.4), dass Personen in Unternehmen wie Peers agieren, die autonom Wissen sammeln und mit anderen austauschen, und dass eine zentrale Funktion des Unternehmens darin besteht, Funktionen für die Aggregation der Mikrowelten der Mitarbeitern bereitzustellen.

2.2.4 Wissensrepräsentation

Im vorherigen Abschnitt wurde sich mit allgemeinen Problemen der Erfassung und Strukturierung von Wissen beschäftigt. Dieser Abschnitt befasst sich mit der konkreteren Fragestellung, wie Wissen außerhalb des menschlichen Geistes repräsentiert werden kann. Grundlegend gibt es dazu nach Bibel ([Bib93]) folgende prinzipiellen Möglichkeiten:

- natürliche Sprache

- Bilder
- Prädikatenlogik
- Semantische und Assoziative Netze

Bilder und die natürliche Sprache sind sicherlich die ältesten und noch immer die verbreitetsten Mittel zur Repräsentation von Wissen, siehe auch [Haa91]. Die Prädikatenlogik ist wahrscheinlich die verbreitetste Form für Experten- und Prognosesysteme. Netze sind intuitiv gut erfassbar und haben eine große Verbreitung in Anwendungen des Wissensmanagements und auch im Semantic Web.

Eine *beste* KR-Sprache kann nicht identifiziert werden. Wie auch Bibel betont, hängt die Wahl der Sprache eng mit der geplanten Nutzung des Systems zusammen. Anders gesagt: Je nach den geplanten Operatoren über Wissen kann die eine oder andere Repräsentationsform vorzuziehen sein.

Bibel unterscheidet folgende Operationen:

- Auffinden
- Hinzufügen
- Löschen
- Automatisches Schließen (*Knowledge Reasoning*)

Mit *Auffinden* ist das Finden von Wissen in einer Wissensbasis gemeint und nicht das Finden neuen Wissens außerhalb der Wissensbasis. Letzteres wird als *Wissensakquise* bezeichnet und meint das Füllen der Wissensbasis von außen und nicht von innen durch automatisches Schließen.

Bibel fasst Auffinden, Hinzufügen und Löschen von Wissen als die grundlegenden Funktionen für die Verwaltung von Wissen (*Knowledge Management (KM)*) zusammen. Ein System, das diese Funktionen anbietet, kann als *Knowledge Management System (KMS)* bezeichnet werden. Verfügt das System außerdem über die Möglichkeit des automatischen Schließens, so ist es ein System der künstlichen Intelligenz (*KI-System*). In dieser Bedeutung sollen die Begriffe in dieser Arbeit genutzt werden.

Diese Erkenntnisse sind seit spätestens den 1980er Jahren allgemein anerkannt. Etwa in den 1990er Jahren erweckte verstärkt der Kontext von Wissen bzw. dessen oftmalige Nichtbeachtung das Interesse der KI-Forscher. Etwas harscher ausgedrückt erklärten Ekbja und Maguitman ([EM01]):

The „neglect of context is the greatest single disaster that AI practice has incurred thus far“. Ähnlich klare Meinungen finden sich auch in [BP96a, BP96b]. Was sind die Gründe für dieses eindeutige Urteil?

1. Es war zu sehen, dass wachsende Wissensbasen in irgendeiner Form strukturiert werden müssen, allein schon, um Operationen in einer für Nutzer erträglichen Zeit ausführen zu können, siehe dazu auch ([Len98]). Es geht dort vor allem um das Überprüfen von Thesen und das Suchen von Wissen. Bei beiden Operationen wird

- oftmals in einem eingeschränkten *Kontext* agiert, z. B. wenn in einem Diagnosesystem die Wahrscheinlichkeit von Krankheiten anhand von Symptomen eines Patienten ermittelt wird. An dieser Stelle ist keinerlei allgemeines Wissen notwendig, sondern nur sehr konkretes Fachwissen. Zusammengefasst heißt es in [Bre99a]: „Context is what constrains a problem solving without intervening in it explicitly.“
2. Es gibt keine perfekte Form der Beschreibung von Wissen. Optimalitätskriterien ergeben sich nur aus der Anwendung und den Anwendern. Idealerweise sollte je nach *Kontext* entschieden werden, welche Art der Beschreibung von Wissen die beste ist.
 3. Bei der Wissensakquise spielt der Kontext eine wesentliche Rolle. Akquisiteure haben einen Fokus, unter dem sie Wissen suchen, um es später der Wissensbasis hinzuzufügen. Dieser wurde und wird meist nicht in der Wissensbasis explizit eingearbeitet. Es wird der Anschein der Objektivität erzeugt, wo es doch subjektive Gründe und Befindlichkeiten waren, die zu einer Einfügung des Wissens führten.
 4. Wie schon weiter oben bemerkt, spielt der Kontext bei der Weitergabe von Wissen eine essenzielle Rolle, und zwar in mehreren Spielarten. So sind Sender und Empfänger in einem Zustand (mental Zustand, wenn es sich um Menschen handelt), die Umgebung, in dem der Wissensaustausch stattfinden kann, in Betracht zu ziehen und natürlich wurde das Wissen, das ausgetauscht wird, in einem Kontext erzeugt und der Empfänger wird es in einem anderen Kontext konsumieren. In [Kon95] heißt es dazu: „Context is a set of all entities that influence human (or systems) cognitive behaviour on a particular occasion.“, vergleiche dazu auch [Giu93, SW86].

In diesem Zusammenhang wird oft der Begriff des *Hintergrundwissens* benutzt. Dieser beschreibt eine Menge Wissen, die notwendig ist, um Wissen zu verstehen bzw. korrekt zu interpretieren. Es gibt Hintergrundwissen sowohl bei den Erstellern einer Wissensbasis als auch bei den Nutzern davon. Beide können sich wesentlich unterscheiden.

Man kann auch zwischen einem allgemeinen Hintergrundwissen und einem persönlichen unterscheiden. Das allgemeine umfasst allgemeine anerkannte Erkenntnisse (z. B. die Erde ist rund, Zeit ist relativ), das persönliche umfasst den privaten Kenntnisstand.

Hintergrundwissen gilt es immer zu beachten, wenn Wissen durch Menschen akquiriert wird und wenn es durch Menschen konsumiert wird, und zwar sind beide sowohl bei der Akquise wie bei der Nutzung zu beachten: Ein Wissensakquisiteur sollte potenzielle Nutzungsszenarien vor Augen haben, genauso wie ein Wissensnutzer das Hintergrundwissen derjenigen kennen sollte, die die Wissensbasis gefüllt haben. Im Extremfall ist ein technisch einwandfrei funktionierendes Diagnosesystem, das von Laien mit Wissen gefüllt wurde, nutzlos oder gar gefährlich.⁶

⁶Generell wünschte man sich einen kritischeren Umgang mit Computern allgemein und Informationen aus „dem Internet“ im Besonderen. Das WWW ist ein perfektes Beispiel dafür, wie unsägliche Mengen von Informationen von Menschen erzeugt werden, die manchmal nur geringe Spuren von Hintergrundwissen aufweisen. Konsumenten sollten diese Tatsache beachten.

In den folgenden Unterabschnitten wird sich mit der Repräsentation von Wissen mittels Prädikatenlogik und mittels natürlicher geschriebener Sprache, die in Form von Dokumenten in Informationssystemen gespeichert ist, beschäftigt. Es besteht nicht das Ziel, einen vollständigen Überblick über die Literatur zu geben, sondern vielmehr die wesentlichen Ansätze aus dem Bereich zu präsentieren. Die Beispiele zeigen exemplarisch konkrete Ausprägungen entweder des Ansatzes Weltwissen oder Mikrowelten. Es wird zu sehen sein, wie die Partitionierung bzw. die Aggregation durch Kontexte erfolgt. Die jeweiligen Konzepte von Kontext unterscheiden sich dabei. Die Beispiele dienen im Wesentlichen dazu, generelle Herangehensweisen zu identifizieren, um sie später für die Definition des Referenzmodells eines P2P-KMS einsetzen zu können.

Eine allgemeine Definition von Kontext soll hier nicht gegeben werden. Der wesentliche Grund für diese Verweigerung besteht darin, dass an vielen Stellen in der Literatur gezeigt wurde, dass der Versuch einer allgemeinen Definition von Kontext nicht gelingt. So heißt es u. a. in [EMPW97]: „Eine direkte Antwort, durch die wir hoffen könnten, das Phänomen Kontext aufzuklären, kann es jedoch nicht geben“ bzw. in [Bre99b] „... it is difficult to speak of context independently of its use: We cannot speak about context out of context“.

Diese Erkenntnis der Literatur aufnehmend, soll sich auch in dieser Arbeit dem Kontext zunächst beschreibend genähert werden und Definitionen sollen nur für speziellere und dann auch positiv definierbare Kontextarten abgegeben werden.

2.2.5 Kontext in der Prädikatenlogik

In diesem Abschnitt werden exemplarisch Ansätze vorgestellt, die zeigen, wie mittels Kontext eine Strukturierung von prädikatenlogischen Ausdrücken erfolgen kann. Der Abschnitt verfolgt weder das Ziel einen Einstieg in die Thematik der Logik zu bieten noch einen vollständigen Literaturüberblick. Er erhebt den Anspruch, die wesentlichen Ansätze anhand einiger Beispiele zu präsentieren.

Es wird davon ausgegangen, dass der Leser mit der Logik wenigstens grob vertraut ist. Es soll nur ein sehr knapper Einstieg zur Erinnerung gegeben werden.

Bei der Suche nach dem Begründer der Logik wird zumeist Aristoteles genannt. Tatsächlich geht auf ihn die Aussagenlogik zurück. Aussagen sind beliebige Äußerungen, die wahr oder falsch sein können⁷. Die Aussagenlogik bietet Möglichkeiten, den Wahrheitswert von zusammengesetzten Aussagen zu ermitteln. Die herausragende Leistung besteht darin, dass er abstrahierend von den Inhalten Feststellungen über den Wahrheitsgehalt einer Aussage machen konnte. Bekannt sind seine Syllogismen, wie z. B. folgender:

*Alle Berliner sind Lügner. X ist Berliner. Daher lügt X.*⁸

⁷oder von denen man nicht weiß, ob sie wahr oder falsch sind, aber das interessierte Aristoteles noch nicht.

⁸Das klassische Beispiel beleidigt die Kreter. Da dem Autor die Einheit Europas sehr am Herzen liegt, will er lieber als Nestbeschmutzer gelten und die Berliner - und damit sich selbst - diskreditieren. Im übrigen sind Berliner zwar tendenziell unfreundlich, aber glaubwürdig. Damit ist also das Beispiel zwar konsistent, aber gelogen.

Ohne dass der Mensch X bekannt ist oder es ihn gar gibt, kann man feststellen, dass X ein Lügner ist. Ein prädikatenlogischer Ausdruck entsteht, vereinfacht gesprochen, so dass das Prädikat eines Satzes gewählt wird und ihm das Subjekt zugewiesen wird. X ist Berliner, kann umgeschrieben werden in $istBerliner(x)$. Sobald der Parameter x gesetzt ist, kann der Wahrheitswert der Aussage ermittelt werden. Zum Beispiel gilt:

$$istBerliner(ThomasSchwotzer) = wahr$$

Der gesamte Syllogismus lässt sich übertragen in

$$(\forall x)(istBerliner(x) \supset luegt(x)),$$

was gelesen werden kann als: Für alle x gilt: Wenn x Berliner ist, so lügt er/sie. Es ist bekannt, dass „Thomas Schwotzer“ ein Berliner ist. Außerdem ist bekannt, dass der Syllogismus, also die Implikation wahr ist. Letzteres ist bei einer wahren Voraussetzung nur dann der Fall, wenn auch das Ergebnis wahr ist. Es lässt sich also schließen:

$$luegt(„ThomasSchwotzer“) = wahr$$

Diese Aussage kann gelesen werden als „Es ist wahr, dass Thomas Schwotzer lügt“. Das Erzeugen dieser neuen Aussage aus den vorherigen wird als *Inferenz* bezeichnet.

Es können auch mehrstellige Prädikate definiert werden, wie z. B. $vorname(x, „Thomas“)$, was den Vornamen von x bezeichnet. Einige Sprachen, wie z. B. Prolog basieren auf der Prädikatenlogik.

Im Folgenden sollen exemplarisch einige Ansätze präsentiert werden, die zeigen, wie Kontext in der Prädikatenlogik genutzt werden kann.

Logic of Context (LoC)

Die Logic of Context (LoC) basiert auf den Überlegungen von McCarthy [McC95, McC93] und wurde von Buvac und Mason formalisiert [BM96, BM93]. LoC basiert auf der Prädikatenlogik, siehe auch [BGGB01].

Wie ein Zitat von McCarthy in Abschnitt 2.2.2 zeigte, geht er davon aus, dass Wissen (also hier Aussagen) immer in einem Kontext existieren. Um diese Tatsache auszudrücken führte McCarthy das Prädikat $ist(c, p)$, was gelesen werden soll, als: „Aussage p ist wahr (is true) in Kontext c “. Eine andere Notationsform ist $c : p$.

Weitere wesentliche Eigenschaften der Logic of Context sind folgende:

- Über Kontexte können Aussagen gemacht werden.
- Aussagen können von einem allgemeiner Kontext in einen konkreteren überführt werden. Dazu muss Folgendes gelten: Wenn $ist(c, p)$ in c' bewiesen werden kann, dann kann in den Kontext c eingetreten werden und es kann $c'c : p$ angenommen werden. Dieser Übergang wird als (*enter*) bezeichnet.
- Aussagen können von einem Kontext in einen allgemeineren übergeben. Dazu wird der innere Kontext explizit der Aussage beigefügt. Gilt $c : p$, und c' ist übergeordneter Kontext von c , so gilt: $c' : ist(c, p)$. Dieser Übergang wird als (*leave* oder *transcend*) bezeichnet.

- Beziehungen von Aussagen in unterschiedlichen Kontexten können durch das *Lifting-Axiom* beschrieben werden. Die generelle Form des Axioms ist die folgende:

$$ist(c, p) \Leftrightarrow ist(c', p')$$

Diese Axiome stellen Querverbindungen zwischen Aussagen aus beliebigen Kontexten her, die in keiner Beziehungen stehen müssen. In [BGGB01] wird das Beispiel der Sherlock-Homes-Geschichten gebracht. Die Aussagen, dass Sherlock Holmes nahe Victoria Station lebt, ist richtig, aber nur genau dann, wenn Victoria Station im Kontext von London Nähe Baker Street liegt, wo Mr. Holmes laut Mr. Doyle bekanntermaßen lebte. Das Lifting-Axiom würde etwas formaler wie folgt lauten:

$$ist(\text{Kontext: Homes-Stories, „Homes lebt Nähe Victoria Station“})$$

\Leftrightarrow

$$ist(\text{Kontext: reales London, „Baker Street ist nahe Victoria Station“})$$

- Jeder Kontext ist von einem weiteren allgemeineren Kontext umgeben. In der Abbildung ist das durch die Kästen abgebildet, die weitere Kästen in sich einschließen. Das Verhältnis von Kontexten ist hierarchisch, d. h., für jeden Kontext existiert genau ein einziger einschließender Kontext. Diese Regel gilt infinit, d. h., in LoC gibt es keinen „äußersten“ Kontext.

Diese Eigenschaften machen die Logic of Context zu einem Vertreter des partitionierten Weltwissens. Das Aufsteigen zu allgemeineren Kontexten wird als *leave* oder *transcend* bezeichnet, das Absteigen in einen konkreten Kontext als *enter*. Beziehungen zwischen Wissen beliebiger Kontexte lassen sich durch das *lifting axiom* herstellen.

Partitioned Representation (PR)

Partitioned Representation von Dinsmore [Din91] geht in eine ähnliche Richtung wie die Logic of Context. Die Theorie überträgt die Idee der *mental spaces* [Fau85] in die Logik. Die Konzepte der Partitioned Representation sind folgende:

- Das Kernkonzept ist der *Space*: „A space represents some logically coherent situation or potential reality, where various propositions are treated as true, objects are assumed to exist, and relations between objects are supposed to hold.“ [BGGB01].
- Aussagen werden immer einem Space zugeordnet. Über deren Wahrheitswert ist direkt keine Aussage möglich. Vielmehr verfügt ein jeder Space über einen *primary context*. Das ist eine Funktion, die die Wahrheit einer Aussage in einem Space auf die Erfüllung einer Aussage in einem übergeordneten Kontext zurückführt. Man spricht von einem *context climbing*.
- Es existiert ein äußerster, allgemeinsten Space, der als *BASE* bezeichnet wird. In ihm kann direkt der Wahrheitswert einer Aussage festgestellt werden. Für BASE existiert kein primärer Kontext und die Wahrheitswerte von Aussagen können direkt bestimmt werden: „... formally BASE represents a de-contextualized representation of the world.“

Der Wahrheitswert einer Aussage kann damit erst ermittelt werden, wenn durch context climbing BASE erreicht wurde.

- Es existiert außerdem ein *secondary context*, der Querverbindungen zwischen Spaces erlaubt. Er erlaubt die Nutzung von Aussagen in einem Space, der in keiner speziellen Beziehung zum Space steht, in dem die Aussagen vorliegen.

Dieser Ansatz geht von kontextfreiem Wissen aus, was bereits in Abschnitt 2.2.2 diskutiert und als kritisch eingestuft wurde. Der kontextfreie „Kontext“ wird hier BASE genannt. Mehr noch, es ist erst in diesem Kontext möglich, den Wahrheitswert von Aussagen zu ermitteln. Wie bereits diskutiert ist dieses Prinzip höchst fragwürdig.

Auch dieser Ansatz ist ein Vertreter des partitionierten Weltwissens. Ein Kontext trägt hier den Namen *Space*. Der Aufstieg wird als *context climbing* bezeichnet. Das Absteigen wird nicht gesondert behandelt. Über den *secondary context* können Zusammenhänge zwischen Wissen in Kontexten hergestellt werden, die in keiner direkten hierarchischen Beziehung stehen.

Context Spaces in CYC

Cyc ist eine der weltweit größten Wissensbasen [CC]. Das Projekt wurde Mitte der 1990er Jahre gestartet. Die CYC Wissensbasis umfasste 1998 100.000 Regeln. Die Arbeiten an CYC laufen noch heute, aktuelle Zahlen über Regeln liegen nicht vor.

Zunächst wurde die CYC Wissensbasis flach strukturiert ohne Partitionierungsmittel. Das stellte sich schnell als inadäquat für die wachsende Datenmenge heraus. Das Problem lag vor allem in der schnell wachsenden Rechenzeit für einzelne Anfragen.

Die erste Strukturierung erfolgte basierend auf einem Ansatz von McCarthy [McC93]. Aus verschiedenen Gründen stellte sich dieser Ansatz als nicht praktikabel für große Wissensbasen heraus. Einer davon war, dass allein das *context lifting* sehr viel Rechenzeit in Anspruch nahm, sodass man es nur für triviale Fälle realistisch einsetzen konnte [Len98].

Stattdessen wurde Kontext als mehrdimensionaler (es sind 12 Dimensionen) Raum verstanden. Aussagen werden nun einer Hyperebene im Kontextraum zugewiesen. Die Interpretation ist ähnlich dem *ist* Prädikat von McCarthy: Die Aussage ist wahr im Kontext, der durch die Hyperebene beschrieben wird. Für einen solchen Kontextraum sind Dimensionen notwendig, die möglichst orthogonal zueinander stehen. In CYC sind es folgende Dimensionen:

Absolute Time ist ein bestimmtes absolutes Zeitintervall (z. B. zwischen den Jahren 1900 und 2000, siehe Abbildung 2.7).

Type Of Time ist ein nicht-absoluter Typ eines Zeitintervalls (z.B. nach dem Abendessen).

Absolute Place ist ein bestimmter Ort (z. B. Berlin (Abbildung 2.7)).

Type Of Place ist ein nicht-absoluter Typ eines Platzes (z. B. im Bett).

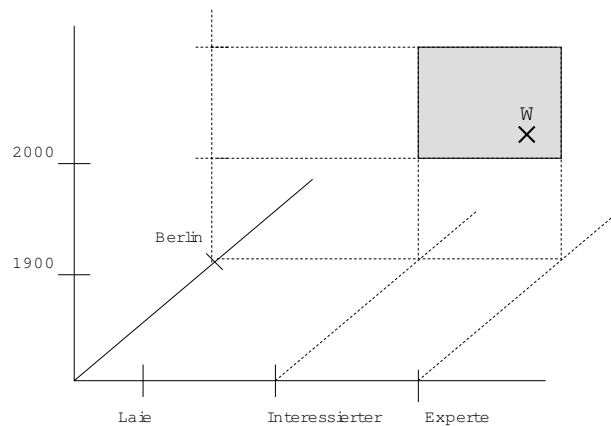


Abbildung 2.7: Beispiel in einem 3-D Context Space

Culture umfasst die religiösen, linguistischen, ethischen, alterstypischen Eigenschaften, die für die Aussagen von Relevanz sind.

Sophistication / Security erlaubt zu beschreiben, wer die Aussagen bereits kennt, in der Lage wäre, sie zu verstehen usw.

Topic / Usage erlaubt die Benennung von Themen, denen die Aussagen zugeordnet werden kann (z. B. Knowledge Management)

Granularity erlaubt die Beschreibung der Detailtiefe, die die zugeordneten Aussagen beachten (z. B. Laie, Interessierter, Experte (Abbildung 2.7)).

Modality / Disposition / Epistemology beschreibt, wer an die Wahrheit der Aussagen glaubt bzw. interessiert daran ist, dass an sie geglaubt wird.

Justification beschreibt den jeweiligen Grad der Geprüftheit einer Aussage. Die Dimension mag sich z. B. bewegen zwischen ungeprüfte intuitive Vermutung bis hin zu allgemein anerkannter bewiesener Behauptung

Let's sind konkrete anwendungsspezifische Festlegungen.

Abbildung 2.7 illustriert ein Beispiel eines Context Spaces. Es wurden nur drei Dimensionen benutzt, um das Beispiel auf den zur Verfügung stehenden zwei Dimensionen in einer lesbaren Form darstellbar zu halten. Im Beispiel wurde der Kontext illustriert, der folgende Arten von Wissen beschreibt: im Zeitraum zwischen den Jahren 1900 und 2000, in Berlin und das Wissen ist passend für Interessierte und für Experten. Ebenfalls wurde Wissen (W) mit einem Kreuz markiert, das auf der Ebene liegen soll, die durch

den Kontext aufgespannt wird. Was könnte W sein? Vielleicht eine Beschreibung der Architektur des genannten Zeitraumes in Berlin, die nicht für Laien geeignet ist.

Context Spaces gehen ebenfalls von der Vorstellung eines Weltwissens aus. Die implizite Annahme des Modells ist, dass es durch die 12 Dimensionen und die geschickte Wahl der Parameter gelingt, potenziell alles Wissen in ein Modell zu bekommen.

Das Vorgehen der Context Spaces lehnt sich an Kategoriensysteme an, siehe Abschnitt 2.2.1. Auch hier werden eine Reihe von (unabhängigen) Eigenschaften genutzt, um Entitäten einzuordnen. Der wesentliche Unterschied besteht aber darin, dass Context Spaces eben Räume beschreiben, in denen Aussagen wahr sind, während durch Kategorien eher versucht wird, einen Punkt zu beschreiben. Außerdem geht es dort um die Zuordnung von Eigenschaften zu Entitäten, während es bei Context Spaces um die Beschreibung geht, in welchem Kontext eine Aussage wahr ist.

Local Model Semantics (LMS)

Ghidini und Giunchiglia haben in [GS94, GG01] ein System unabhängiger lokaler Theorien vorgestellt. Zwischen den Mikrotheorien können Beziehungen definiert werden. Damit können sie zu einer größeren Theorie vereint werden.

Die wesentlichen Eigenschaften des Systems sind

Lokalität Alle Theorien sind lokal definiert. Das Schließen erfolgt ausschließlich in den lokalen Theorien.

Autonomie Jede Theorie ist autonom. Es gibt keine vordefinierten Beziehungen zwischen lokalen Theorien, insbesondere keine hierarchischen Beziehungen.

Kompatibilität Mittels *constraints* können Bedingungen formuliert werden, wie Aussagen zwischen verschiedenen Theorien zusammenhängen. Diese *constraints* können dazu benutzt werden, dass Prozesse des Schließens auch Aussagen anderer Theorien einbeziehen können. Diese Einbeziehungen wirken wie ein temporäres Importieren von Teilen der externen Theorie und widersprechen nicht dem Prinzip der Autonomie und Lokalität.

Auf die konkrete Form der Wissensrepräsentation soll nicht eingegangen werden, dazu sei verwiesen auf [BGGB01, GG01, GS94]. Der LMS-Ansatz folgt im Gegensatz zur LoC und PR eindeutig dem Ansatz aggregierter Mikrotheorien. Die Aggregation erfolgt in diesem Modell durch eine Menge von *constraints*, die Beziehungen zwischen den ansonsten unabhängigen Kontexten herstellen. Dieses Vorgehen entspricht dem Ansatz der Definition einer Funktion, die Wissen eines Kontextes so wandelt, dass es in einem anderen Kontext benutzt werden kann, wie er in Abschnitt 2.2.3 beschrieben wurde.

2.2.6 Kontext in Content-Management-Systemen

In Abschnitt 2.2.4 wurden die prinzipiellen Möglichkeiten zur Präsentation von Wissen aufgezählt. Das waren natürliche Sprache, Bilder, Prädikatenlogik und semantische Netze. Die Prädikatenlogik wurde in Abschnitt 2.2.5 behandelt. Semantische Netze werden

an zwei Beispielen ausführlich in Kapitel 5 vorgestellt. Dieser Abschnitt beschäftigt sich mit der natürlichen Sprache und Bildern.

Sprache kann in verschiedener Form repräsentiert werden. An dieser Stelle sollen nur die digital verarbeitbaren Formate interessieren. Das gesprochene/gesungene Wort kann z. B. in einem MP3-File gespeichert werden. Das geschriebene Wort kann ebenfalls z. B. in PDF-Files gespeichert werden. Alternativ kann ein handschriftlicher Text gescannt und in einem Bildformat abgelegt werden. Es gibt verschiedene Formate für Bilder und der Übergang von einer reinen Schreibschrift zu einer Bildschrift ist teilweise fließend. Es gibt also eine Fülle von Möglichkeiten, natürliche Sprache und Bilder in einem durch Computer verarbeitbaren Format zu repräsentieren.

Der wesentliche Unterschied zwischen Sprache und Bilder auf der einen und logischen Ausdrücken und semantischen Netzen auf der anderen Seite ist der, dass Letztere deutlich besser durch Methoden der KI bearbeitbar sind. Die maschinelle Erkennung der Bedeutung eines geschriebenen Textes, so wie es einem Menschen gelingt, ist bis heute nicht möglich, genauso wenig wie das Erkennen der Bedeutung eines Bildes die technischen Möglichkeiten bei weitem übersteigt. Es ist unklar, wie lang dieser Zustand anhält, vermutlich noch einige Jahre bis Jahrzehnte.

Trotzdem sind Dokumente, die im Wesentlichen aus natürlichsprachlichen Sätzen bestehen und über einige Bilder verfügen, die sich aber keinen Formalien unterwerfen, die es Rechnern erleichtern, sie zu verarbeiten, die dominierende Repräsentationsform von Wissen in unserer Gesellschaft. Ein Beispiel ist dieser Text.

Content ist die übliche Bezeichnung für die Verallgemeinerung digitaler Dokumente beliebiger Formate. Ein *Content Management System (CMS)* ist in der Lage, Content zu verwalten. Das schließt folgende Funktionen ein: Möglichkeiten des Auffindens des Contents (z. B. Volltextsuche oder Suche in Verzeichnisstrukturen), des Hinzufügens von Content und von Löschen von Content. Das Löschen ist oftmals auch mit der Möglichkeit der Archivierung und Versionierung des Contents verbunden, was *Löschen* zu einem mehrstufigen Prozess macht. Da natürliche Sprache und Bilder als Möglichkeit der Wissensrepräsentation gelten, erfüllt ein CMS alle Bedingungen, als ein Knowledge Management System (KMS) bezeichnet zu werden. Etwas formelhafter gilt demnach: *Ein CMS ist ein KMS*.

In der Forschung spricht man weniger von CMS als vielmehr von Informationssystemen oder -basen (*information bases*). Im Folgenden sollen exemplarisch einige Ansätze vorgestellt werden, wie Kontext in solche Systeme eingeführt werden kann.

Mylopoulos

Einen Ansatz zur Partitionierung von Informationsbasen hat Mylopoulos in [MMP95] vorgestellt. Nach Mylopoulos' Modell kann eine Informationsbasis beliebige Informationen enthalten und hat folgende zusätzliche Eigenschaften:

- Eine Informationsbasis ist unterteilt in *units*.
- Eine *context* ist eine spezielle Unit. Ein Kontext repräsentiert eine Dekomposition einer Informationsbasis. Ein Kontext wird durch drei Komponenten beschrieben:

- $contents(c)$ definiert die Menge aller Objekte der Informationsbasis, die dem Kontext c zugeordnet sind bzw. die in c *sichtbar* sind („*visible in c*“ [BGGB01]).
 - $lexicon(c)$ definiert eine Menge von Identifikatoren, die im Kontext gültig sind. Mylopoulos schlägt vor, dass Identifikatoren eine Reihe von alphanumerischen Zeichen sind, die durch Anführungszeichen begrenzt werden (z. B. 'John').
 - Eine Menge von Formeln, die definieren, ob und welche Nutzer berechtigt sind, Transaktion im Kontext auszuführen.
- Zwischen zwei beliebigen Kontexten kann ein *propagation channel* eingeführt werden. Ein solcher Kanal erlaubt es, dass Änderungen in einem Kontext eine Änderung in einem anderen Kontext veranlassen. Ein solcher Channel wird aufgebaut durch folgende Definitionen:

$propagationFrom(c_1, fromContext, authorP)$

$propagationTo(c_2, toContext, authorP)$

Die erste Formel beschreibt, dass Änderungen in *fromContext* in c_1 bekannt gemacht und Änderungen unter den Rechten von *authorP* ausgeführt werden. Die zweite Formel beschreibt, dass Änderungen in c_2 in *toContext* bekannt gegeben werden sollen. Es muss also von exportierender als auch importierender Seite explizit definiert werden, dass Änderungen propagiert werden.

Nach der Ausführung beider Operationen ist *propagation channel* zwischen Kontexten etabliert. Diese Kanäle stellen einen Weg zur Synchronisation zwischen unterschiedlichen Kontexten und damit (Teilen von) Informationsbasen dar.

Abbildung 2.8 veranschaulicht das Modell. Die Kreise im unteren Teil symbolisieren beliebige Objekte als den Content. Die Rechtecke stehen für zwei Kontexte (a und b). Die Pfeile symbolisieren die Zuordnung von Objekten zu Kontexten, die durch die Mengen $content(a)$ bzw. $content(b)$ definiert wurden. Die Objekte erhalten im Kontext jeweils eine Bezeichnung, die im Lexikon ($lexicon(a)$ und $lexicon(b)$) definiert ist. Zwischen a und b wurde eine *propagation channel* angedeutet.

Aus der Abbildung ist zu erkennen, dass Objekte auch mehreren Kontexten zugeordnet werden können und dort jeweils eine andere Bezeichnung tragen können. Die Kreise in den Kontexten wurden bewusst nur durch gestrichelte Linien symbolisiert, um zu unterstreichen, dass nicht die Objekte im Kontext enthalten sind, sondern nur eine Zuordnung zu diesen erfolgte. Anders herum: Ein Kontext enthält Referenzen auf die Objekte und kann sie entsprechend benennen.⁹

In dem Ansatz können selbst Kontexte Objekte sein, so dass auch sie anderen Kontexten zugeordnet werden können, was eine Hierarchiebildung erlaubt.

Theodorakis

Manos Theodorakis hat den Ansatz von Mylopoulos weiterentwickelt ([The99, The02]).

⁹Assoziationen zu symbolischen Links unter Unix sind passend.

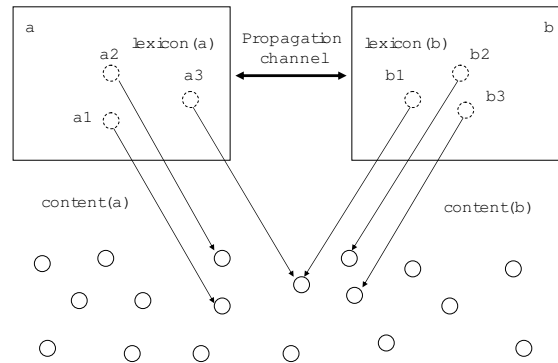


Abbildung 2.8: Informationsbasis nach Mylopoulos

Er versteht unter Kontext Folgendes: „... a context is a higher-level conceptual entity that describes a group of conceptual entities from a particular standpoint.“ Weitere Eigenschaften seines Ansatzes sind folgende und konkretisieren Mylopoulos in einigen Punkten:

- Ein Kontext enthält Objekte oder Kontexte.
- Objekte können Dinge der realen Welt repräsentieren („*represent real world attributes*“) oder aber eine gerichtete binäre Beziehung zwischen zwei Objekten definieren. Letztere Objekte werden als *Linkobjekte* (*link objects*) bezeichnet.
- Objekte können in mehreren Kontexten vorkommen.
- Jeder Kontext hat ein Vokabular, mit dem Objekte benannt werden. Objekte können keinen, einen oder mehrere Namen haben. Die Benennung der Objekte kann in jedem Kontext neu erfolgen. Es gibt keine Abhängigkeiten zwischen den Kontexten bezüglich der Benennung.
- Ein Objekt kann über einen Pfadnamen adressiert werden, d.h. über eine Kette von Namen von Kontexten bis hin zum Namen des Objektes im letzten Kontext.
- Es werden eine Reihe von Funktionen über Objekte definiert, das sind im Wesentlichen: *insert* (Einfügen von Objekten in einen Kontext), *delete* (Löschen von Objekten), *copy* (Kopieren von Kontexten) *lookup* (Suche nach Objekten), *union* (Vereinigung zweier Kontexte bilden), *intersection* (Durchschnitt zweier Kontexte bilden). Für *lookup* und *copy* existieren Varianten, die eine „tiefe“ Suche bzw.

ein „tiefes“ Kopieren erlauben. Damit ist gemeint, dass rekursiv (bis zu einer definierbaren Tiefe) auch enthaltene Kontexte einbezogen werden.

Ein Kontext stellt eine Sicht auf eine Menge von Objekten dar. Ein ganzes Informationssystem kann damit ein Kontext sein, denn es stellt eine Sammlung von Objekten dar, das Nutzern Informationen für einen bestimmten Zweck zur Verfügung stellt. Ein Kontext kann eine Sicht (*view*) auf eine Datenbank sein. Wenn es Objekte gibt, die in mehreren Versionen vorliegen, kann ein Kontext genutzt werden, um auf alle Versionen des Objektes zu verweisen (*multiversion object*). Für aggregierte Objekte kann ein Kontext auf die jeweiligen Objekte in der passenden Version verweisen. Ein Kontext kann die Objekte enthalten, die für eine aktuelle Arbeitsaufgabe benötigt werden.

Theodorakis' Ansatz ist deutlich detailreicher gegenüber Mylopoulos, sowohl was die Arten der Objekte als auch die Ausführung der Operationen über Kontexte angeht.

In beiden Modellen ist es möglich, eine Hierarchie von Kontexten zu definieren, was in aggregierten Mikrowelten nicht möglich war. Es wird in keinem Ansatz ein Auf- oder Absteigen von Referenzen auf Objekte diskutiert, dafür beschreibt Theodorakis wie Pfadnamen die Navigation zu einem Objekt durch eine Hierarchie von Kontexten hindurch ermöglicht. In keinem der beiden Ansätze werden Fragen eines äußersten Kontextes oder die Thematik von Weltwissen diskutiert. Beide Arbeiten sind aus dem Bereich der Strukturierung von Informationssystemen und beschäftigen sich nicht mit der Thematik der Ontologien. Obwohl beide Modelle eher die Idee eines Weltwissens zu atmen scheinen, sollen sie aber keiner der beiden Alternativen endgültig zugeordnet werden. Das sei – bei Bedarf – dem Leser überlassen.

2.2.7 Zusammenfassung

Wissen kann unterschiedlich repräsentiert werden, durch Sprache, Bilder, logische Ausdrücke und semantische Netze. Abstrahiert man von den Darstellungsformen, so bleibt Wissen Kenntnis über Gegenstände. Jeder Gegenstand bedarf einer Bezeichnung, um über ihn kommunizieren zu können. Die Kenntnis kann sich durch Beziehungen des Gegenstandes zu anderen ausdrücken. Die Bezeichnung von Gegenständen und deren Zusammenhänge werden von Begriffssystemen unterstützt. Eine geeignete Repräsentationsform sind z. B. logische Ausdrücke. Wissen kann auch Kenntnis über die Arbeit mit einem Gegenstand sein. Solches Wissen ist oft in Dokumenten niedergelegt, die in Informationssystemen gespeichert sein können.

Es gilt, dass die Wahrheit bzw. die Anwendbarkeit des Wissens nicht allgemein gültig ist, sondern nur in bestimmten Bereichen. Diese Gültigkeitsbereiche werden erfolgreich durch – in Details verschiedene Arten von – Kontext modelliert.

Das Prinzip ist in allen Gebieten das gleiche: Wissen wird einem Kontext zugefügt, sei es in der Logic of Context, indem ein Prädikat eingeführt wird, das die Wahrheit einer Aussage in einem bestimmten Kontext beschreibt, sei es, dass im Ansatz von Theodorakis Entitäten einem Kontext zugeordnet werden, was einer Definition der Anwendbarkeit des Wissens in diesen Kontext bedeutet. Wird von der Struktur abstrahiert, so drückt die Zuordnung von Wissen zu einem Kontext einen Gültigkeitsbereich oder eine Wahrheit

aus. Mehr nicht.

Eine fundamentalere Frage ist eher, wie der Kontext und der dahinterstehende Gültigkeitsbereich aufgefasst werden. Die Gretchenfrage auf diesem Gebiet ist die folgende: Gibt es einen allgemeinen Kontext? Wird diese Frage bejaht, so wird ein allgemein gültiges Wissen postuliert, denn Wissen in einem allgemeinsten Kontext ist immer gültig. Wird die Frage verneint, so kann gefragt werden, ob ein Aufstieg in allgemeinere Kontexte möglich ist. Wird diese Frage bejaht, so geht man von einer unendlichen Hierarchie von Kontexten aus, die sich asymptotisch an einen allgemeinen Kontext annähern.

Der Logic of Context wurde unterstellt, dass auch sie dem Ansatz Weltwissen folgt, obwohl sich McCarthy entschieden gegen die Vorstellung eines allgemeinen Kontextes wendete. Für praktische Anwendungen ist seine philosophische Verneinung eines äußeren Kontextes belanglos, denn aus praktischer Sicht ist es irrelevant, ob sich einem allgemeinen Kontext beliebig nahe angenähert oder ob dieser tatsächlich erreicht wird.

Werden beide Fragen verneint, geht man von lokalen Theorien, den Mikrowelten, aus. Wissen hat dort nur einen eingeschränkten Gültigkeits- bzw. Anwendungsbereich. Kontexte können auch dort hierarchisch strukturiert sein, doch ist der Schritt aus dem äußersten Kontext nicht möglich und dieser äußerste Kontext bedeutet keine Allgemeingültigkeit, sondern die Grenze des Diskursbereiches.

Autonomie von Peers ist nicht vereinbar mit Konzepten, die irgendeiner Form von Weltwissen folgen. Reine P2P-Systeme können nur auf einem Konzept der aggregierten Mikrowelten aufsetzen. Alle andere Versuche führen notwendig zu Inkonsistenzen der Modelle, was – und das soll noch einmal unterstrichen werden – nicht an der Wahl der Wissensrepräsentationssprache liegt, sondern am Verständnis von Kontext.

2.3 Software-Agenten

Dieser Abschnitt beschäftigt sich damit, *wie* Wissen ausgetauscht wird. Mit dem gleichen Thema beschäftigen sich eine Fülle von Arbeiten aus dem Bereich der Software-Agenten, die ein Forschungsgegenstand der künstlichen Intelligenz sind. Ziel der Agentenforschung ist es, solche Softwaresysteme zu gestalten, die das menschliche Verhalten in einem (Teil eines) Prozesse(s) genau abbilden. Ein idealer Agent würde mithin darin bestehen, dass er anstelle eines Menschen agiert und dabei (in einem eingeschränkten Handlungsbereich) genau die Aktionen ausführt, die auch der Mensch ausführen würde.

Weitere Eigenschaften von Agenten sind nach [fip02] Autonomie (Fähigkeit Aktionen selbstständig und ohne Interaktion mit Dritten auszuführen), Zielverfolgung, Besitz mentaler Modelle und die Fähigkeit, mit anderen Agenten zu kommunizieren.

Ziel dieser Arbeit ist es *nicht*, ein Agentensystem zu entwickeln oder es für das Knowledge Management einzusetzen. Ziel dieser Arbeit ist ein System, das Menschen bei wissensintensiven Prozessen unterstützt. Es wird davon ausgegangen, dass sich solche Prozesse gerade dadurch auszeichnen, dass sie (noch) nicht so formalisiert sind, dass sie automatisierbar wären. Es ist damit – per Definition – ein wesentliches Charakteristikum solcher Prozesse, dass sie von Menschen ausgeführt werden. In der Terminologie des vorangegangenen Abschnittes wird in dieser Arbeit ein KMS entwickelt und *kein*

KI-System. Aus diesem Grunde wird nicht detailliert auf Ergebnisse im Bereich der Agentenforschung eingegangen werden. Dieses Kapitel dient zu großen Teilen der Abgrenzung zu diesem Gebiet. Das wird auch in späteren Teilen dieser Arbeit, die mögliche Implementierungen, konkret des Protokolls zum Wissensaustausch, diskutieren, deutlich werden.

Ausgangspunkt der Agentenforschung ist die Untersuchung menschlicher Kommunikation. Von besonderem Interesse sind natürlich Überlegungen, die dem Kontext Beachtung schenken. Diese werden kurz vorgestellt. Einige Ansätze sind auch für das Ziel dieser Arbeit nutzbar.

Dieser Abschnitt ist der kürzeste in diesem Kapitel. Er ist wie folgt strukturiert. Es werden zunächst einige Ergebnisse präsentiert, wie aus Sicht der Agentenforschung ein kontextbasierter Wissensaustausch abläuft (Abschnitt 2.3.1). Danach folgt eine kurzer Überblick über die Sprechakttheorie von Searl, die Basis aller relevanter Agentensprachen ist und sich gut zum Vergleich der später entwickelten Sprache zum Wissensaustausch und den Agentensprachen eignet (Abschnitt 2.3.2). Danach werden nur kurz die bekanntesten Agentensprachen, die FIPA-ACL und KQML, vorgestellt (Abschnitt 2.3.3). Den Abschluss bildet auch hier eine Zusammenfassung.

2.3.1 Kontextbasierter Austausch von Wissen

Die Kommunikation zwischen Agenten soll die menschliche Kommunikation möglichst vollständig nachbilden. Aus diesem Grunde sind die Überlegungen aus dem Bereich der Agentensprachen auch für diese Arbeit relevant, selbst wenn sich die jeweiligen Ziele unterscheiden.

Ziel dieser Arbeit ist die Erstellung eines P2P-KMS, in dem autonome Peers kontextbasiert Wissen austauschen. Darin ist das Ziel noch gleich den Agenten. Die Peers sollen das Wissen aber nicht selber benutzen können, sondern lediglich Menschen bereitstellen. Das ist der zentrale Unterschied zu Agenten, die in irgendeiner Form *intelligent* sind und selber Wissen verarbeiten können.

Jede Kommunikation findet in einem Kontext statt. Zu diesem Kontext tragen die Kommunikationspartner bei und alles andere, was kurz als Umgebung bezeichnet werden kann. Dabei ist zu beachten, dass nur die Werte der Umgebung tatsächlich in einen Kommunikationskontext einfließen, die von wenigstens einem Kommunikationspartner wahrgenommen werden. Wenn sich beispielsweise zwei Personen unterhalten und meinen ohne weitere Zuhörer zu sein, wobei allerdings ein Dritter lauscht, der aber von keinem bemerkt wird, so wird die Kommunikation ablaufen, ohne dass das Vorhandensein des lauschenden Dritten Einfluss auf den Kommunikationskontext hat.

In [CK93] wird aufgezählt, was zu einem Kommunikationskontext beiträgt: *dialogue memory*, *task memory*, *environmental situation*, *psycho-social situation* und ein generelles Wissen über die Welt.

Einige dieser Einflussfaktoren können sich während der Kommunikation ändern, was auch zu einer Änderung des Kommunikationskontextes führen kann. In [Bre99b] ist diese Dynamik wie folgt beschrieben. „The context of communication in human-human dialogue can be defined as a set of transitory mental representations that are dynamically

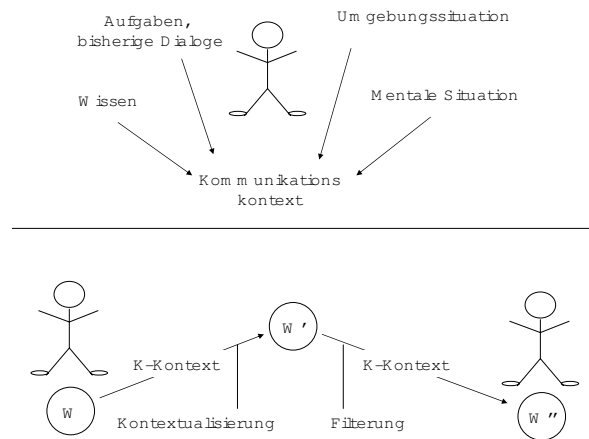


Abbildung 2.9: Kontextbasierter Wissensaustausch zwischen Agenten

built by people during the interaction from the external situation that they perceive and from their knowledge, beliefs and characteristics. These transitory representations are of the past dialogue, the task that they are performing, the spatiotemporal situation, the psychological situation (representation of the other, of the social situation and so on).“

Es sind also eine Reihe von Faktoren, die den Kontext einer Kommunikation beeinflussen, siehe auch den oberen Teil der Abbildung 2.9. Die Umgebung, genauer das, was davon beobachtet werden kann, fließt darin ein. Die mentale Situation der Kommunikationspartner hat ebenfalls Einfluss auf den Austausch. Beide können als eher schnell veränderliche Größen angenommen werden. Weiterhin fließen die bisherigen Erfahrungen der Kommunikationspartner miteinander, deren Ziele und deren Wissen ein. Diese Größen können als eher stabil angesehen werden.¹⁰

Aus dieser Unterscheidung heraus wird auch in [Ezh92] erklärt, dass ein Agent wenigstens zwei Speicherarten haben muss: einen langlebigen für die Wissensbasis, die Ziele des Agenten und dauerhaftes Wissen enthält, und einen kurzlebigen für den Kontext der Kommunikation.

Anhand eines Kommunikationskontextes entscheiden die Kommunikationspartner, ob und welches Wissen sie austauschen. Das macht ein potenzieller Sender, indem er den Kommunikationskontext auf das Wissen anwendet. In [EM93] wird dieser Vorgang *Kontextualisierung* genannt. Andere Kommunikationspartner wiederum benutzen den Kommunikationskontext als einen Filter, um nur das entgegenzunehmen, was im gegebenen Kontext relevant ist, siehe dazu [Bre99b].

¹⁰Das gilt allerdings nicht mehr in Situationen, die dramatische Erkenntnisse offenbaren und damit das Verhältnis zwischen den Kommunikationspartnern plötzlich stark verändern, wie sie in jedem Krimi vorkommen.

Abbildung 2.9 fasst die Überlegungen zusammen. Im oberen Teil der Abbildung ist die Ermittlung des Kommunikationskontextes dargestellt. Die Psychosoziale Situation wird hier als mentale Situation abgekürzt. Auf der linken Seite sind die eher dauerhaften Informationen dargestellt, auf der rechten Seite die eher flüchtigen. Im unteren Teil der Abbildung ist der Austausch von Wissen W dargestellt. Der Kommunikationskontext (K-Kontext) wird einmal benutzt, um das passende Wissen für diese Kommunikation zu ermitteln (Kontextualisierung), und auf der Empfängerseite als Filter eingesetzt. Das Wissen wird also zweimal umgewandelt.

Diese Überlegungen zeigen deutlich Parallelen mit dem Modell sozialer Gruppen aus dem Abschnitt 2.1. Alles andere wäre auch verwunderlich, beschäftigen sich doch all diese Überlegungen mit der Kommunikation zwischen Menschen. Im Modell der sozialen Gruppen wurde anhand einer Strategie ermittelt, welche Handlungen ausgeführt werden und anhand einer Relevanzbewertung wurden die eingehenden Informationen gefiltert. Das Modell galt allgemein für jede Einwirkung einer sozialen Gruppe auf die Umwelt bzw. für die Wahrnehmung der Umwelt. Bezieht man das Modell konkret auf die Kommunikation, so kann man sagen, dass der hier beschriebene Kommunikationskontext Basis für die Strategie und die Relevanzbewertung ist. Auf beide wirken als Ziele Wissen, die aktuelle mentale Situation und die Umgebung ein.

In der Agentenforschung wird nach diesen Untersuchungen gefolgert, dass alle genannten Eingangsgrößen inklusive Funktionen zur Ermittlung des Kommunikationskontextes in einem Agenten verfügbar sein müssen.

In dieser Arbeit kann ein leichter Weg beschritten werden. Ziel ist es, das ACP-Modell für den Bereich Knowledge Management zu spezialisieren. Die Peers werden dann über Wissen verfügen, aber weiterhin die Umgebung beobachten können und autonom entscheiden können, ob und welches Wissen sie austauschen wollen. Es besteht kein Bedarf, mentale Modelle zu entwickeln, um in irgendeiner Form die mentalen Befindlichkeiten von Menschen nachempfinden zu können. Es besteht keine Notwendigkeit, ein Konzept der Zielorientiertheit einzuführen, wenn davon ausgegangen wird, dass das Peer als Hilfsmittel dient, dessen Nutzer seine Ziele kennt und verfolgt. Es müssen lediglich Regeln definiert werden, die sicherstellen, dass sich die Peers an die Regeln zum Austausch von Wissen halten, die die Anwender definiert haben. Ein Peer muss aber, wie ein Agent, seine Umgebung wahrnehmen können, aber nur deutlich weniger *intelligent* damit umgehen.

2.3.2 Sprechakte

Searle hat mit seinem Buch *Speech Acts* aus dem Jahre 1969 geradezu seziert, was passiert, wenn Personen miteinander sprechen. Diese Arbeit ist eine Basis der bekanntesten Ansätze im Bereich der Agentensprachen. Als Referenz wird hier eine deutsche Übersetzung aus dem Jahre 1971 verwendet [Sea71].

Searle untersucht den Akt des Sprechens. Oder wie er es in der Einleitung beschreibt: „Wie kommt es in Situation, in der ein Sprecher einem Zuhörer gegenübersteht und ein akustisches Signal aussendet, zu so bemerkenswerten Dingen wie den folgenden: der Sprecher meint etwas; die Laute, die er von sich gibt, bedeuten etwas; der Zuhörer

versteht, was gemeint ist; der Sprecher behauptet etwas, stellt eine Frage oder gibt einen Befehl?“ Er nennt das, was dabei stattfindet, einen *Sprechakt* und unterscheidet primär drei Arten und hebt einen vierten heraus, der eine Ableitung eines anderen ist:

Äußerungsakt. Dieser Akt umfasst die Tatsache, dass Wörter geäußert werden.

Propositionaler Akt. Dieser Akt umfasst, dass auf etwas Bezug genommen wird, dass etwas referenziert wird bzw. dass eine *Prädikation*, also eine irgendwie geartete Bewertung stattfindet.

Illokutionärer Akt. Dieser Akt umfasst alles, was direkt auf eine Reaktion beim Zuhörer abzielt, also Dinge wie Behaupten, Fragen, Befehlen, Versprechen etc.

Perlokutionärer Akt. Dieser Akt ist eng verbunden mit dem illokutionären Akt. Er bezeichnet aber die *Wirkung*, die ein solcher Akt „auf die Handlungen, Gedanken, Anschauungen usw. der Zuhörer habe. Zum Beispiel kann man jemanden durch Argumentieren *überreden* oder *überzeugen*, durch Warnen *erschrecken* oder *alarmieren*, durch Auffordern *dazu bringen, etwas zu tun*, durch Informieren *überzeugen (aufklären, belehren, anregen, dazu bringen, etwas zu begreifen)*.“ [Sea71]¹¹

Searle stellt fest, dass die Sprechakte nie vollständig getrennt voneinander auftreten. So kann ein illokutionärer Akt durchaus einen propositionalen Akt enthalten, wenn auf irgendetwas referenziert wird. Tiefer soll die Sprechakttheorie nicht diskutiert werden, für Details sei direkt auf [Sea71] verwiesen.

2.3.3 Agentensprachen

Die zwei wesentlichen Agentensprachen sind die *Agent Communication Language* der *Foundation for Intelligent Physical Agents (FIPA)* ([fip02]) und die *Knowledge Query and Manipulation Language* [FLM97].

In beiden Sprachen wird von sehr ähnlichen Agentenmodellen ausgegangen, nämlich, dass Agenten Ziele verfolgen, über Wissen verfügen und prinzipiell die Umgebung wahrnehmen können. Agenten versuchen weiterhin Menschen dahingehend nachzuempfinden, dass sie über mentale Modelle verfügen. Agenten sollen, wie Menschen, nicht nur zwischen wahren und falschem Wissen unterscheiden können, sondern auch vermuten und meinen können. Diese Forderung übersteigt die Möglichkeit der in Abschnitt 2.2 genannten Ansätze. Es bieten sich stattdessen modale Logiken an. Diese fügen das Konzept der *Möglichkeit* in die Logik ein. Es ist dann möglich, zu formulieren, dass eine Aussage möglich sein kann oder nicht oder gar zwingend möglich ist und damit wahr ist. Für eine Übersicht über modale Logiken sei auf [Sow00] verwiesen. Vielleicht der bekannteste Vertreter der modalen Logik war Kripke, dessen Ansatz auch in [fip02] als Basis für die modalen Modelle der Agenten gewählt wurde. Da modale Logiken in dieser Arbeit nicht benutzt werden, wird nicht weiter auf sie eingegangen werden.

¹¹Hervorhebungen durch Searle

ACL und KQML basieren auf der Sprechakttheorie von Searl. Kern der Sprachen ist die Möglichkeit, illokutionäre Sprechakte zu vollziehen. Diese Akte haben das Ziel, bei einem anderen Agenten eine Reaktion hervorzurufen, z. B. das Senden einer Information. Andere Sprachelemente versuchen, das Modell eines Agenten zu ändern, z. B. zu erreichen, dass er an die Richtigkeit einer Aussage glaubt. So werden perlokutionäre Akte nachgestellt. Sprechakte treten oft nicht isoliert auf. Auch die einzelnen Konstrukte der Sprachen haben Anteile, die propositionale und Äußerungsakte sind.

Agentensprachen werden im Laufe der Arbeit keine Rolle spielen. Sie wurden an dieser Stelle im Wesentlichen der Vollständigkeit wegen genannt und dienen in einem folgenden Kapitel der Abgrenzung.

2.3.4 Zusammenfassung

Endgültiges Ziel der Agentenforschung ist der Bau von Systemen, die Menschen (wenigstens in einigen Arbeitsbereichen) ersetzen. Dazu müssen Agenten in irgendeiner Form Ziele verfolgen können und dabei strategisch vorgehen. Zur Erlangung dieses Zieles werden modale Logiken zur Modellierung mentaler Modelle eingesetzt.

Agenten beeinflussen andere Agenten, indem sie mit ihnen Nachrichten austauschen. Es wurden die zwei wesentlichen Agentensprachen vorgestellt, die FIPA-ACL und KQML. Beide leiten sich von der Sprechakttheorie von Searle ab. Es war auch zu sehen, dass Kontexte eine wesentliche Rolle bei der Agentenkommunikation spielen.

In der Einleitung dieser Arbeit wurde bereits betont, dass die vollständige Automatisierung wissensintensiver Prozesse per Definition nicht möglich ist. Solche Prozesse sind – per Definition – nicht oder nur zu geringen Teilen formalisiert. Insofern kann kein IT-System diese Prozesse ausführen.

Der Ansatz der Agentenforschung ist der, dass versucht wird, möglichst viele Eigenschaften von Menschen in IT-Systeme zu übertragen, dass sind insbesondere mentale Modelle, soziales Verhalten, Ziele usw. Der Ansatz, der in dieser Arbeit verfolgt wird, ist ein anderer. Es gilt ein System zu entwickeln, das Hilfsmittel für die Akteure in wissensintensiven Prozessen ist, diese aber in keiner Form ersetzen soll. Insofern ergibt sich nicht die Notwendigkeit der Modellierung mentaler Modelle etc. Das vereinfacht die Aufgabenstellung enorm und trägt – so die These – zu Akzeptanz der Lösung bei.

Trotzdem werden aus der Agentenforschung einige Konzepte übernommen. So wird die Betonung des Kontextes geteilt und Arbeiten zu dessen Formalisierung werden teilweise in das eigene Konzept einfließen. Allerdings deutlich einfacher.

Es soll abschließend noch einmal verkürzt der Unterschied zwischen Agenten und den in dieser Arbeit entwickelten P2P-Ansatz genannt werden. Agenten können als autonome Peers angesehen werden, die mit anderen Agenten / Peers Informationen oder Wissen austauschen. Der fundamentale Unterschied zu dem Ansatz dieser Arbeit besteht darin, dass Agenten tendenziell Menschen *ersetzen* sollen. Das Shark-Peer, das weiter hinten entwickelt wird, soll niemals einen Menschen ersetzen, es soll lediglich *unterstützen*.

2.4 Knowledge Management (KM)

Das Knowledge Management entstand in den 1980er Jahren als eine Managementtheorie. Zentrale Fragestellung des KM ist: Was ist Wissen im Produktionsprozess und wie kann man die Flüsse von Wissen verbessern?

Das Knowledge Management räumte im ersten Schritt mit der Vorstellung auf, dass wissensintensive Unternehmen als informationsverarbeitende Systeme auffassbar wären, dass alles notwendige Wissen prinzipiell formalisierbar wäre, wenn man nur ausreichend Zeit investierte. Tatsächlich gab es lange Zeit Versuche, alle Informationsprozesse eines Unternehmens so detailliert zu verstehen und zu modellieren, dass kein Wissen mehr verloren gehen und dass Wissen, wie ein Werkstoff im Unternehmen bewegt werden könnte. KI-Systeme spielten bei solchen Ansätzen eine essenzielle Rolle, sollten sie doch die Wissensflüsse steuern und die Wissensverarbeitung möglichst automatisieren. Das ideale Ziel bestand in der Automatisierung wissensintensiver Prozesse, wie es zuvor bereits gelungen war, einen großen Teil der Produktionsprozesse unabhängig von menschlicher Arbeit zu machen. Diese Ansätze sind gescheitert. Eine gute geschichtliche Übersicht findet sich in [NT95].

Erfolgreicher waren Ansätze, die Unternehmen als selbst organisierende Systeme auffassten, wie z. B. Senge in „The fifth discipline“ [Sen94]. Er prägte den Begriff der *lernenden Unternehmen*. Tatsächlich verstand man langsam, dass Wissen (was immer es sei) oftmals untrennbar mit Menschen verbunden ist, dass es aber Strukturen gibt, die den Wissensaustausch fördern, und andere, die ihn hemmen. Das Lernen eines Unternehmens besteht darin, die Strukturen so anzupassen, dass sie die Wissensflüsse im Unternehmen in der konkreten wirtschaftlichen Situation möglichst optimal unterstützen. Da sich das Umfeld ändert, müssen auch die Strukturen variabel sein.

Erfolgreiches Management von Wissen bedeutet aus dieser Denkweise heraus, ein Umfeld zu schaffen, in dem sich die Strukturen für Wissensflüsse selber optimal ausbilden können. Dieses Herangehen steht im direkten Gegensatz zu einer zentralen Definition von Prozessen. Manager sollte nicht mehr Prozesse bestimmen, sondern ein Umfeld schaffen, in dem Wissen entstehen und fließen kann. Das war ein Kulturwechsel.

Was unterscheidet aber wissensintensive Prozesse von anderen? Was ist Wissen eigentlich, dass es sich scheinbar einem zentralen und direkten Management entzieht? Oder steckt dahinter nicht doch der Unwillen von Mitarbeitern, einem Management „zu gehorchen“?

Wissensintensive Prozesse sind alle Prozesse, für die die Abfolge von Teilprozessen oft nicht vorhergesagt werden kann bzw. deren Arbeitsschritte (bisher) nicht erklärt wurden bzw. werden können. Ein Beispiel soll das erläutern: Im Mittelalter war der Bau eines Fasses eine Kunst. Bretter müssen geschnitten und geformt werden. Beschläge müssen aus Eisen hergestellt werden. Es gab keine Beschreibung des Vorganges, vielmehr wurde diese Fähigkeit vom Meister zum Gesellen weitergegeben. Die Fassproduktion war ein wissensintensiver Prozess. Heute ist jeder Schritt zur Fassherstellung klar erfasst, der gesamte Prozess ist automatisierbar. Heute sind z. B. die Diagnose einer Krankheit und die Entwicklung eines neuen Mikrochips wissensintensive Prozesse. Intuition, Erfahrung und viel Hintergrundwissen sind notwendig, um sie erfolgreich durchzuführen. Eine

Klassifikation solcher Prozesse findet sich u. a. in [CG04].

Wissen ist in diesem Zusammenhang der Faktor, der bewirkt, dass die Mitarbeiter befähigt sind, wissensintensive Prozesse zu einem (erfolgreichen) Ende zu bringen. Sobald ein Prozess vollständig beschrieben werden kann, hört er – per Definition auf – ein wissensintensiver Prozess zu sein. In [GU04] wird gezeigt, dass in deutschen Unternehmen bereits über ein Drittel der Arbeiten als Wissensarbeiten bezeichnet werden können.

Es ist das Verdienst von Nonaka und Takeuchi, einer breiten Öffentlichkeit die Wichtigkeit von implizitem Wissen bewusst gemacht zu haben. Implizites Wissen sind Fähigkeiten, über die Menschen verfügen, die sie aber gar nicht oder nur schwer verbalisieren können. Hängt ein Prozess zu wesentlichen Teilen von implizitem Wissen ab, gilt er als wissensintensiv.

Dadurch erklärt sich auch der scheinbare Widerstand bei dem Versuch, die Weitergabe von Wissen in Unternehmen von zentraler Stelle aus zu organisieren. Es ist möglich, festzulegen, welches Dokument wann von wem an wen geschickt werden soll. Es ist schwer oder gar nicht mehr festlegbar, wann jemand eine Fähigkeit (von jemand anderem) erlernen soll. Implizites Wissen wird vorrangig im direkten persönlichen Gespräch weitergegeben. Der Mensch ist zu wenig Maschine, als dass solche Kontakte reglementiert werden könnten. Implizites Wissen ist vom eigenen Besitzer nur schwer zu fassen und damit gar nicht mehr durch Externe direkt zu managen.

Viele Definitionen und Beschreibungen im KM sind daher auch unscharf vor allem im Vergleich zu Definitionen aus Naturwissenschaften. Die Ursache liegt in der Beschäftigung mit dem, was gemeinhin als *soft facts* bezeichnet wird, also den Fähigkeiten von Menschen zur Zusammenarbeit bzw. den Rahmenbedingungen, die menschliche Arbeit verbessern.

Ein erfolgreiches Knowledge Management macht sich dadurch bemerkbar, dass passende und flexible Strukturen in Unternehmen vorliegen, die die Entstehung und Weitergabe von Wissen unterstützen. Erst zweitrangig kann Software Unterstützung leisten.

Ein Knowledge Management System ist nur in der Lage, explizites Wissen zu speichern, also Wissen, was in Dokumenten vorliegt. Es kann kein implizites Wissen verarbeiten. Ein KMS muss als Hilfsmittel der wissensintensiven Prozess verstanden werden, die in Unternehmen ablaufen. Insofern befreien KMS die Menschen nicht von wissensintensiver Arbeit, sie unterstützen sie dabei. Es herrscht ein Mangel an Modellen für KMS, die diese Tatsachen beachten. In der Praxis ist das übliche KM-Modell für ein KMS das Organigramm des Unternehmens. Das ist paradox, weil die KM-Theorie gerade zeigt, dass Wissen zu großen Teilen gerade *nicht* den offiziellen Wegen folgt, sondern in informellen Netzen fließt. Tatsächlich steht man vor der paradoxen Situation, dass die empirischen Untersuchungen zeigen, dass Wissensflüsse im Wesentlichen auf P2P-Austauschen beruhen, das Modell und KMS aber auf einem zentralistischen und hierarchischen Ansatz basieren.

Es gibt aber seit dem beginnenden 21. Jahrhundert Bestrebungen, Knowledge Management als ein verteiltes System (*Distributed Knowledge Management Systems (DKMS)*) aufzufassen. Es gibt einige wenige Überlegungen, Knowledge Management als Peer-to-Peer-System zu verstehen. Es ist aber leider auch zu sehen, dass es kein reines P2P-Modell für das Knowledge Management gibt und dass die vorliegenden Ansätze noch

immer, wenigstens in Teilen, über zentralistische Aspekte bzw. Inkonsistenzen verfügen.

Das Kapitel ist wie folgt aufgebaut. Am Anfang wird das SECI-Modell von Nonaka und Takeuchi vorgestellt, das ein Meilenstein im Bereich KM war und auf dem aufsetzend später in dieser Arbeit ein P2P-Modell entwickelt wird. Danach werden einige Vorstellungen von Davenport und Prusak präsentiert, die in einer Fülle von empirischen Studien und Fallbeispielen zeigen konnten, wie Wissen in Unternehmen fließt, und dabei eine Reihe von Begriffen prägten. Im Anschluss werden Ansätze vorgestellt, die zum Ziel haben, KM als verteilte Prozesse zu beschreiben bzw. als Modell für ein verteiltes KMS zu dienen. Am Ende steht eine Zusammenfassung.

2.4.1 Nonaka und Takeuchi

Es ist nicht übertrieben, zu sagen, dass Nonaka und Takeuchi mit ihrem Buch „The Knowledge-creating company“ [NT95] einen Meilenstein gesetzt haben. Ihr wesentliches Verdienst besteht in zwei Punkten: Einmal haben sie die Bedeutung von Dokumenten als Wissensträger relativiert und die Bedeutung von stillem oder implizitem Wissen hervorgehoben. Außerdem haben sie in einem Modell den Umwandlungszyklus von Wissensarten untereinander beschrieben.

Sie unterscheiden generell zwei Wissensarten, *explizites Wissen (explicit knowledge)* und *implizites Wissen (tacit knowledge)*:

„Tacit knowledge is personal, context-specific, and therefore hard to formalize and communicate. Explicit or codified knowledge, on the other hand, refers to knowledge that is transmittable in formal, systematic language.“

In deutschen Übersetzungen wird oft von explizitem und implizitem Wissen gesprochen, wie das auch in dieser Arbeit getan wird. Bei dieser Übersetzung von *tacit*¹² zu implizit verschiebt sich aber ein wenig die Bedeutung. Die Bezeichnung *tacit knowledge* kommt ursprünglich von Polanyi [Pol58a, Pol58b]. Es bezeichnet alle Fähigkeiten und Kenntnisse von Menschen, die sie zur Ausübung von Tätigkeiten einsetzen. Fähigkeiten wie Radfahren sind *tacit knowledge*. Man kann sie erlernen, aber es fällt schwer, sie ausschließlich mit Worten zu beschreiben. Manches Wissen muss erlebt werden und kann nicht verbal vermittelt werden. Polanyi dazu: „We know more than we can tell“. Das Wort implizit hat nicht die Konnotation von *nicht ausdrückbar* wie *tacit*. Trotzdem wird natürlich bei der allgemein üblichen deutschen Terminologie geblieben.

Eine weitere wesentliche Eigenschaft von implizitem Wissen gilt es festzuhalten: „In a strict sense, knowledge is created only by individuals. An organization cannot create knowledge without individuals.“ Diese Tatsache ergibt sich automatisch aus der Definition von implizitem Wissen. Es ist Wissen, das im Kopf eines Menschen vorliegt. Genau dort aber, im Kopf eines Menschen, entsteht ein neuer Gedanke, neues Wissen. Neues Wissen erblickt immer als implizites Wissen die Welt.

Explizites Wissen ist nach Nonaka und Takeuchi Wissen, das außerhalb des menschlichen Geistes kodifiziert ist. Einige meinen, dass explizites Wissen per Definition nicht existieren kann [Old02]. Die Argumentation ist einfach: Egal, ob Wissen als Erkenntnis

¹²Strikt übersetzt: still(schweigend)

oder als eine Fähigkeit zur zielgerichteten Handlung verstanden wird, beides setzt Intelligenz voraus. Sobald nun aber Wissen das intelligente Wesen verlässt und in welcher Form auch immer auf ein definitiv nicht intelligentes Medium kodifiziert wird, hört es auf, Fähigkeit oder Kenntnis zu sein und ist per Definition kein Wissen mehr. Um es plastisch zu sagen: Ein beschriebenes Stück Papier – selbst wenn es mit nobelpreisbezahlten Ideen übersät ist – *kennt nicht* und *handelt nicht*. Es *weiß nichts* und *kann nichts wissen*. Dieses Papier als (explizites) *Wissen* zu bezeichnen sei deshalb unsinnig. Dieser Argumentation kann sich angeschlossen werden.

Man kann das Verständnis von explizitem Wissen aber auch anders fassen. Bezeichnet man etwas, das außerhalb von vernunftbegabten Wesen existiert und das durch Nutzung von diesem neues Wissen bei ihnen erzeugt oder vorhandenes Wissen in Erinnerung bringt, so möge dieses Etwas als explizites Wissen bezeichnet werden.

Das Stück Papier wäre damit explizites Wissen, aber nur für diejenigen, die daraus Wissen erlangen können. Für andere nicht! Hier wird die Tatsache augenfällig, dass auch explizites Wissen nur im Prozess gedacht werden kann: Jemand muss genügend Hintergrundwissen haben und in einer passenden Situation sein, um das Papier zu lesen und zu verstehen, auf dass dieses dann als explizites Wissen bezeichnet werden darf. Streng genommen dürfte man das Papier nur innerhalb dieses Prozesses als explizites Wissen bezeichnen, weil nur dann der Beweis erbracht ist, dass jemand daraus Erkenntnisse oder Fähigkeiten erlangen kann.

Es soll nicht so streng vorgegangen werden. Für den Rest dieser Arbeit möge es genügen, dass mit einer hinreichenden Wahrscheinlichkeit eine Situation herbeigeführt werden kann, in der das Papier als Träger kodifizierten Wissens in beschriebener Art und Weise genutzt werden kann, um es als explizites Wissen zu bezeichnen. Diese Deutung von explizitem Wissen scheint sich mit den Vorstellungen von Nonaka und Takeuchi zu decken, ohne dass sie allerdings solche (eventuell etwas haarspalterischen) Überlegungen angestellt hätten.

Nonaka und Takeuchi bezeichnen ihre Unterscheidung nach den zwei Wissensarten als *epistemologische Dimension*. Es gibt vier Formen der Umwandlung von Wissen auf dieser Dimension, siehe Abbildung 2.10, die im Folgenden genauer erläutert werden. Das Modell wird als SECI-Modell bezeichnet, abgeleitet von den Anfangsbuchstaben der Prozesse *socialization*, *externalization*, *combination*, *internalization*.

Sozialisation (implizit zu implizit) „Socialization is a process of sharing experiences and thereby creating tacit knowledge such as mental models and technical skills.“ Sozialisation findet immer statt, wenn Menschen wechselseitig Fähigkeiten erklären, vorführen und diskutieren. Eine Sozialisation findet z. B. statt, wenn eine Handwerksmeister einem Lehrling den Bau eines Fasses zeigt und wenn in einem Brainstorming Ideen für ein neues Softwaresystem diskutiert werden.

Externalisierung (implizit zu explizit) : „Externalization is a process of articulating tacit knowledge into explicit concepts.“ Externalisierung findet immer statt, wenn implizites Wissen in irgendeiner Form in ein physisches Medium kodiert wird, sei es in gesprochenen, geschriebenen oder gesungenen Worten, Bildern, Grafiken usw.

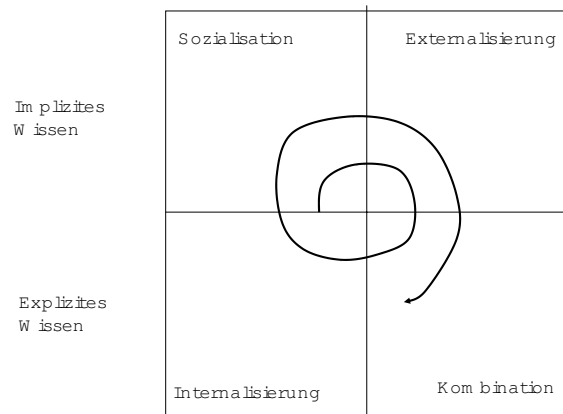


Abbildung 2.10: Das Seci-Modell nach Nonaka und Takeuchi

Kombination (explizit zu explizit) : „Combination is a process of systemizing concepts into a knowledge system.“ Die Kombination umfasst die Erzeugung von explizitem Wissen aus existierendem explizitem Wissen. Das Schreiben einer Dokumentation, basierend auf einer Reihe von Einzelbeschreibungen und einer Sammlung von Gesprächsmitschriften, ist z. B. Kombination.

Internalisierung (explizit zu implizit) : „Internalization is a process of embodying explicit knowledge into tacit knowledge.“ Jedem ist der Vorgang bekannt, wenn man ein neues technisches Gerät in Betrieb nehmen will und dazu das Handbuch liest. Das ist Internalisierung. Eine explizite Beschreibung wird im Zuge dieses Vorganges zu einer persönlichen Fähigkeit.

Das SECI-Modell sagt auch, dass die Prozesse immer in der Reihenfolge Wissensarten ineinander umwandeln. Diese Aussage ist aber nicht sehr hart, da keine *Wissenspartikel* oder Ähnliches identifiziert werden können, die man quasi markieren und auf ihrem Lebensweg beobachten könnte. Insofern ist die Aussage eher eine statistische und empirisch gewonnene Aussage über übliche Wege des Wissens.

Nonaka und Takeuchi führen eine zweite Dimension, die ontologische Dimension, ein. In dieser Dimension sind in folgender Reihenfolge aufgeführt: Individuum, Gruppe, Organisation, Interorganisation. Sie argumentieren, dass Wissen in dieser Dimension wandern kann. Es kann von Individuen auf Gruppen, auf Organisationen usw. und umgekehrt übertragen werden.

Woher die Bezeichnung ontologische Dimension kommt, wird nicht erklärt. Als vollständig passend kann die Bezeichnung allerdings nicht angesehen werden. Soziologischen Dimension hätte den Sachverhalt eventuell treffender dargestellt, da es schließlich um den

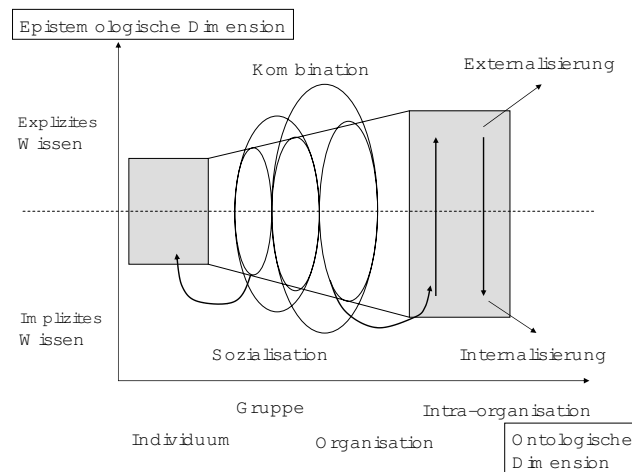


Abbildung 2.11: Wissensspirale nach Nonaka und Takeuchi

Austausch zwischen bzw. innerhalb von sozialen Systemen geht und nicht um eine Spezifikation eines Diskursbereiches, was eine Ontologie nun einmal ist. Aber natürlich soll auch an dieser Stelle die Terminologie der Literatur beibehalten werden.

Zu beachten ist, dass die ontologische Dimension nicht tatsächliche Individuen und Gruppen usw. darstellt, sondern lediglich die Klassen der Individuen, Gruppen usw.

Beide Dimensionen werden als unabhängig angesehen, sie spannen also eine Ebene auf. Das Spiralmodell, Abbildung 2.11, zeigt die möglichen Umwandlungen von Wissen in beiden Dimensionen. Es soll genauer betrachtet werden.

Die möglichen Umwandlungen des Wissens sind mit Spiralen und Pfeilen dargestellt. In der Mitte der Abbildung ist die Grenze zwischen explizitem und implizitem Wissen markiert. Obwohl sie eine Darstellung gewählt haben, die einem Funktionsverlauf ähnelt, soll die Abbildung nicht so interpretiert werden. Sie versucht vielmehr einen Überblick zu geben, wie die vier Umwandlungsprozesse das Wissen in der aufgespannten Ebene „bewegen“, ohne dass Details der Darstellung zu genau genommen werden sollten.

Die Sozialisierung wirkt im unteren Bereich der Ebene, in der Sphäre des impliziten Wissens. Sie bewirkt zusätzlich eine Bewegung auf der ontologischen Dimension. Das ist natürlich, denn der verbale und non-verbale Austausch von Wissen kann auch dafür sorgen, dass Gruppen von einzelnen Individuen lernen und umgekehrt. Solches Wissen kann außerdem von Gruppen auf Organisationen übergehen und umgekehrt. Ein direkter Austausch zwischen Individuen und Organisationen ist ebenfalls denkbar. Wann passiert so etwas? In offiziellen Gruppenbesprechungen, Workshops, während informeller Treffen beim Kaffee oder während des Wartens an einem zentralen Drucker usw. usf.

Die Kombination wirkt im oberen Bereich, in der Sphäre des expliziten Wissens. Hier wird explizites Wissen zusammengefasst und neu strukturiert. Auch hier erfolgt eine

Bewegung auf der ontologischen Dimension. Die Bewegung von links nach rechts findet statt, wenn Wissen Einzelner (Individuen, Gruppen) zusammengefasst und ggf. verallgemeinert und zu Wissen einer übergeordneten Einheit wird. Wissen bewegt sich in umgekehrter Richtung, wenn so kombiniertes Wissen einer übergeordneten Einheit (Organisation, Gruppe) nun auch von Interesse für einen Teil von ihr (Individuen, Gruppen) wird, die das Wissen vorher nicht hatten.

Die Externalisierung und Internalisierung bewirken, dass Wissen zwischen der oberen und unteren Ebene wechselt. Die Abbildung scheint zu suggerieren, dass diese Prozesse keine Bewegung auf der ontologischen Dimension bewirken. Das ist nicht so. Vielmehr bewirkt die Externalisierung eine „Befreiung“ von einer konkreten Position in der ontologischen Dimension. Prinzipiell ist explizites Wissen für jeden im Unternehmen zugänglich. Aus dieser Überlegung heraus ist die Externalisierung eher so zu lesen, dass Wissen aus dem unteren Bereich der Ebene in den oberen Bereich übertragen wird und dort von jeder Position auf der ontologischen Dimension aus benutzt werden kann.

Die Internalisierung kann so verstanden werden, dass von einer beliebigen Position der ontologischen Dimension aus explizites Wissen in implizites Wissen umgewandelt werden kann.

In realen Unternehmen ist ein völlig freies Verteilen und Zugreifen weder erwünscht noch sinnvoll, da Mitarbeiter durch die Informationsflut geradezu gelähmt würden. Jedes Unternehmen hat explizite oder wenigstens implizite Richtlinien über die Weitergabe von Wissen (*Knowledge Management Policy*). Das WWW ist zu großen Teilen ein Beispiel für ein System ohne eine explizite Policy. Technisch kann jeder versuchen jedem alles mitzuteilen. Es existieren aber so genannte *Netikette*, die wenigstens informell den Umgang miteinander regeln. Diese können als einfache Form von KM-Policies angesehen werden.

Ein kleines Beispiel soll die Wanderung von Wissen illustrieren. In der Firma X existiert ein Zeiterfassungssystem. Dieses ist einfach. Jeder Mitarbeiter trägt sich morgens in eine Liste ein und wieder aus, wenn er geht. Solche Listen existieren für jede Abteilung und werden regelmäßig in ein zentrales System eingegeben, auf das aber nicht alle Mitarbeiter Zugriff haben sollen. Herr Y ist zuständig für das regelmäßige Abtippen der Listen, was ihn viel Zeit kostet. Er sinnt auf Verbesserung. Eines Tages fällt ihm ein, eine Webanwendung zu bauen, um damit die Zettel zu ersetzen, und die ihm gleichzeitig ein Dokument liefert, das er direkt in das zentrale Erfassungssystem überspielen kann. Neues Wissen ist geboren. Ist es gut? Er baut die Anwendung und diskutiert das Vorgehen mit seinen Kollegen in der Abteilung. Diese Diskussion ist eine Sozialisierung. Die Abteilung (Gruppe) erfährt von der neuen Idee. Umgekehrt erfährt Herr Y, dass Kollege Z noch nie einen Webbrowser benutzt hat und einen solchen zunächst brauchen würde. In Zuge der Sozialisierung ist Wissen in beide Richtungen der ontologischen Dimension gewandert.

Das System wird in Betrieb genommen und Herr Y schreibt eine kleine Dokumentation (Externalisierung), in die er zusätzlich Beschreibungen für den Gebrauch von Webbrowsern generell einfügt (Kombination), die z. B. Personen, wie dem Kollegen Z, helfen sollen. Er stellt diese Beschreibung ins Intranet (und gibt Herrn Z eine ausgedruckte Version), woraufhin Frau A darauf aufmerksam wird, die Leiterin einer anderen Abteilung ist. Sie liest die Dokumentation und sieht auch für ihre Abteilung ein Verbesserungspotenzial. Es hat eine Internalisierung stattgefunden. Aus einer Diskussion (wiederholte

Sozialisierung) ergibt sich, dass das System mit einigen wenigen Modifikationen auch in ihrer Abteilung einsetzbar ist usw. usf.

2.4.2 Davenport und Prusak

Davenport und Prusak sind Unternehmensberater im Bereich des Knowledge Managements. Inhaltlich können sie der Schule von Nonaka/Takeuchi zugeordnet werden.

Sie beschreiben das Knowledge Management in einem Unternehmen als Markt und prägen dazu die Begriffe *Knowledge Buyer* und *Knowledge Seller*. Wie auf jedem Markt gibt es auch *Knowledge Broker*, die Käufer und Verkäufer zusammenbringen bzw. Wissen weitergeben. Sie zeigen, dass die treibende Kraft dieses Marktes nicht Geld ist. Es sind eher sehr menschliche Motivationen, wie Freude am Helfen, der Wunsch, als wissend wahrgenommen zu werden, einen Platz in einer Gruppe einnehmen zu wollen bzw. diesen zu halten oder schlichtweg jemandem einen Gefallen tun zu wollen. Sie zeigen auch, dass Motivationssysteme, die nur auf Geld basieren, im praktischen Einsatz scheitern. Die Motivation, Wissen weiterzugeben, hat offenbar tiefere menschliche Quellen als das monetäre Gewinnstreben. Belobigungen, das Öffentlich-Machen von Wissenden und das Ausloben von Preisen z. B. für gute Ideen sind erfolgreichere Ansätze. Sie unterstreichen auch die Wichtigkeit der unternehmens-öffentlichen Treffpunkte, wie Wasserspender, Kaffeeecken, Bibliotheken usw., die unerlässlich sind für einen unternehmensinternen Wissensaustausch (Sozialisierung).

Besonders relevant für diese Arbeit ist ihre Feststellung, dass der Austausch von Wissen im direkten Gespräch (Person zu Person) die wesentliche Form des Austausches von Wissen im Unternehmen ist, siehe dazu vor allem [DP00].

Diese Erkenntnis kann gar nicht überbetont werden. Es liegt offenbar in der Natur des „Stoffes“ Wissen, dass es sich nur ungern durch reglementierte Wege und schriftliche Darlegungen ausbreitet. Vielmehr ist nachweisbar seine bevorzugte Form der Verbreitung der direkte Austausch von Mensch zu Mensch.

Sie zeigen an vielen Beispielen, dass die Ausbreitung von Wissen in Unternehmen häufiger über informelle Kanäle erfolgt als über offizielle Strukturen, siehe dazu auch [NCSW04, BC03, MPF04], die Gleiches bemerken und davon ausgehend verschiedenen Tools und Analysemethoden vorschlagen. Sie unterstreichen, dass das keinesfalls als eine Insubordination der Mitarbeiter zu deuten ist, sondern im Gegenteil dahinter der Antrieb steckt, die Arbeit effizient und erfolgreich durchzuführen. Es gilt, diese informellen Kanäle zu erkennen und zu stärken und Möglichkeiten zu geben, dass solche Kanäle entstehen. Sie benutzen den Begriff der *Community of Practice (CoP)*, um Gruppen von Menschen zu beschreiben, die an ähnlichen Themen arbeiten. CoPs seien die wesentlichen Quellen der Wissensentstehung und Verbreitung, siehe auch [FFK03].

Sie untersuchen auch die Einführung von Informationstechnik (IT) für die Unterstützung des Knowledge Managements in Unternehmen. Sie können zeigen, dass ein schlechtes oder nicht vorhandenes Knowledge Management durch IT nicht verbessert werden kann. Umgekehrt gilt aber, dass IT helfen kann, ein gutes und vor allem von den Menschen akzeptiertes KM zu optimieren.

Quelle und Ziel von Wissen ist auch bei ihnen immer der Mensch. Somit ist die Über-

tragung von Wissen niemals losgelöst vom Menschen zu sehen. Formelhaft formulieren sie: „Knowledge Transfer = Transmission + Absorption (and Use)“[DP00]. Wissen, das nach einer Übertragung nicht vom Menschen genutzt wird, gilt nicht als übertragen.

2.4.3 IT-Unterstützung für das SECI-Modell

Diese Arbeit hat als Ziel, ein IT-System für das Knowledge Management zu konzipieren, deshalb soll sich vorhandenen Ansätzen zugewandt werden.

IT-Systeme können das Knowledge Management unterstützen, allerdings nicht alle Prozesse. Zunächst gilt eine wesentliche Einschränkung: IT kann nur mit explizitem Wissen umgehen, nicht mit implizitem. Die Bedeutung des Letzteren wurde aber in den vorherigen Abschnitten gezeigt. Das zeigt wiederholt, dass wissensintensive Prozesse durch IT allein nicht vollzogen werden können. Es sollen trotzdem kurz alle vier Umwandlungsprozesse aus dem SECI-Modell angesprochen werden.

Die Sozialisation sorgt für die Verteilung impliziten Wissens in Unternehmen. Hier kann IT wenig leisten, außer dass sie Menschen hilft, die passenden Ansprechpartner für ein Themengebiet zu finden. Jüngere Beispiele für solche Anwendungen sind z. B. Systeme zur Erfassung von Erfahrungen¹³ der Mitarbeiter ([GU04, RF04]) eine Toolunterstützung zur Bildung von Ad-hoc-Arbeitsgruppen ([HS04]), Methodiken zur Analyse der Wissensflüsse in sozialen Gruppen und der Unterstützung der Bildung von Gruppen ([SST04, FPW04]). Zusammengefasst versuchen diese Systeme entweder existierende Gruppen und deren Wissensflüsse zu identifizieren, auf dass deren Arbeit unterstützt werden kann, oder sie erfassen Wissensgebiete der Mitarbeiter, auf dass diese gefunden werden können, wenn ihre Kenntnisse gefragt sind.

Die Externalisierung wandelt implizites in explizites Wissen um. Dieser Prozess kann durch unterschiedlichste Werkzeuge unterstützt werden, die in irgendeiner Form helfen, Gedanken und Ideen aus dem Kopf auf externe Medien zu bringen. Das können z. B. Mindmapping-Tools sein, um die Gedanken zu strukturieren, und Designwerkzeuge aller Art (z. B. UML-Editoren, CAD-Systeme, Ontologieeditoren, Textsysteme), um Spezifikationen zu verfassen. Solche Systeme sind wohl bekannt.

Die Kombination ist der Prozess, in dem vorhandenes explizites Wissen zu Neuem zusammengefasst wird. Hier können die gleiche Tools wie bei der Externalisierung eingesetzt werden. IT-Systeme sind für die Kombination besonders hilfreich, wenn sie helfen, bereits vorhandenes explizites Wissen aufzufinden, um es zu neuem zu kombinieren. Die Kombination schließt auch den Wechsel der Formate ein, in denen explizites Wissen kodifiziert ist. Aktuell erfolgt die Reformatierung im Wesentlichen durch die einzelnen Mitarbeiter, was hohe Kosten erzeugt. An allgemeinen Plattformen zur Lösung des Problems wird gearbeitet, sie existieren aber noch nicht ([AN04]).

Die Internalisierung ist der Vorgang, der explizites Wissen in implizites wandelt. Auch hier können Suchfunktionen sehr hilfreich sein, um erst einmal das passende explizite Wissen zu finden, das internalisiert werden soll. Jüngere Forschungen beschäftigen sich in diesem Bereich vor allem damit, wenigstens Teile von wissensintensiven Prozessen zu

¹³gern auch als *skills* oder *intellectual capital* bezeichnet

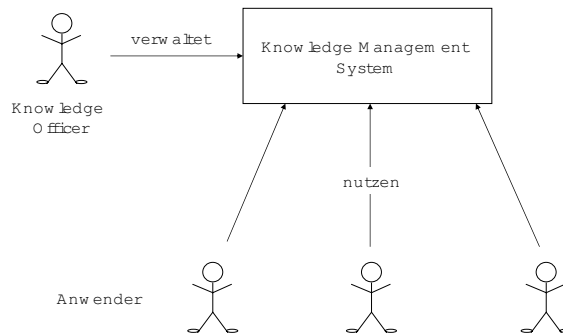


Abbildung 2.12: Zentralistisches Knowledge Management System

verstehen, vor allem die, die in denen explizites Wissen relevant ist und diese durch Workflowsysteme zu unterstützen ([DC04, Dus04, GW04]). Bewährte Ansätze ist die zentrale Bereitstellung von Informationen zu bestimmten Produkten und Prozessen (siehe u. a. [ABG04]).

Die genannten Ansätze haben eines gemeinsam: Sie konzentrieren sich jeweils auf einen Teilaspekt des Knowledge Managements, bezeichnen sich aber trotzdem oftmals als Knowledge Management Systems. Damit soll aber nicht angedeutet werden, dass sie sich als allumfassende Plattform zur Unterstützung aller wissensintensiven Prozesse verstehen. Mit diesem Verständnis soll auch im Folgenden wieder von KMS die Rede sein.

Technik bildet Theorie nicht ab

Jenseits der Diskussion um gespeicherte Formate und die speziellen Funktionen eines KMS ist auch dessen Architektur von Interesse. In den überwiegenden Fällen und in allen im vorherigen Teilabschnitt zitierten Literaturstellen wird in der technischen Plattform auf eine Client-Server-Architektur gesetzt, wie sie in Abbildung 2.12 dargestellt ist. Das ist auch heute noch das überwiegende Vorgehen bei der Konzipierung von KMS ([BBC02]). An zentraler Stelle wird ein KMS betrieben, das inhaltlich von einem *Knowledge Officer* verwaltet wird. Die Anwender greifen auf das System zu. Üblich sind WWW-Interfaces, d. h. der Zugriff erfolgt mittels eines Webbrowsers und HTTP.

Es gibt einige Kritik an diesem Ansatz, nur sind die Lösungsvorschläge ganz unterschiedlich. Die meisten Kritiker argumentieren technisch und erklären, dass pure Client-Server-Architekturen nicht mehr dem Stand der Technik entsprechen, dass stattdessen serviceorientierte Architekturen besser wären ([Woi04]), dass KMS auf Basis verteilter

objektorientierten Plattformen implementiert werden sollten ([GPD04, OS04]) oder dass KMS semantische Netze benutzen sollten ([MVNDGL⁺04]). Die Liste ließe sich fortsetzen.

Die Wahl der Technologie kann aber nur als nachrangige Diskussion angesehen werden. Der fundamentale Gegensatz zwischen Theorie und Technik ist derzeit ein herausragendes Problem im Bereich KM: Die Theorie und empirische Studien zeigen den P2P-Charakter des Austausches von Wissen. Es gibt sogar Studien, die darauf hinweisen, dass trotz aller IT-Unterstützung das meiste explizite Wissen noch immer in Mailboxen und unstrukturiert in Foldern direkt auf den PCs oder Laptops der einzelnen Mitarbeiter gespeichert wird ([SEMK04, TPSS04]).

Die Theorie zeigt eindeutig: Wissen – auch explizites – wird als ein sehr privates Gut verstanden und auch so behandelt. Man speichert es lieber privat und gibt es direkt weiter, z. B. als Ausdruck und als E-Mail. Die Akzeptanzprobleme beginnen dann, wenn Mitarbeiter „ihr“ Wissen auf einen Server bringen und umgekehrt Wissen von einem entfernten System beziehen sollen. Die Dokumente, die dort vorgefunden werden, werden als anonym wahrgenommen. Anders ist das ganz offensichtlich, wenn ein Kollege direkt einen Link oder ein Dokument per E-Mail schickt. Das mag unlogisch erscheinen, aber das ist die Realität. Wissensintensive Prozesse haben viel mit Intuition zu tun und können als sehr kreative Prozesse gelten. Es wäre nur zu verwunderlich, wenn alles logisch ablaufen würde.

Die technische Kritik greift dabei zu kurz. Der entscheidende Nachteil zentraler Systeme ist, dass sie als *entfernte Systeme* wahrgenommen werden. Es ist ein sehr unpersönlicher Vorgang, Dokumente dort einzustellen und von dort zu beziehen. Dabei ist es irrelevant, ob der entfernte Zugriff mittels HTTP oder IIOP erfolgt. Die E-Mail-Kommunikation wird auch nicht durch die Nutzung von SMTP freundlicher, sondern durch den Eindruck der direkten (asynchronen) Kommunikation mit einer anderen Person.

Bonifacio und andere argumentieren in eine ähnliche Richtung. So wird in [BC03] objektives Wissen subjektivem gegenüber gestellt. Als objektives Wissen wird etwas bezeichnet, das zentral bereit steht und wenigstens den Anschein der Allgemeingültigkeit hat. Subjektives Wissen dagegen steht in einem Kontext und ist auch nur dort gültig. Nach der Terminologie aus Abschnitt 2.2 steht das objektive Wissen in einem allgemeinen Kontext, ist praktisch kontextfreies Wissen. Dabei behaupteten bereits Nonaka und Takeuchi in [NT95], dass Wissen *immer* in einem Kontext steht.

Nahezu alle derzeitigen KMS gehen davon aus, dass es objektives Wissen gibt, das seine Objektivität einfach dadurch gewinnt, dass irgendjemand im Unternehmen behauptet hat, dass das nun gälte. Dieser Jemand kann in einigen Fällen sogar benannt werden: Es ist ein *Knowledge Officer (KO)* bzw. der *Chief Knowledge Officer (CKO)*. Ihre Aufgabe besteht darin, das Wissen zu verwalten und verfügbar zu machen. In dem Moment aber, wo sie es freigeben und verfügbar machen, erhält es den Status der Objektivität, also der Kontextfreiheit. Man könnte einwenden, dass natürlich ein Kontext existiert, nämlich das Unternehmen, in dem das Wissen gilt. Für die Wahrnehmung der Menschen ist dieser Hinweis genauso irrelevant, wie McCarthys Behauptung, es gäbe zwar keinen allgemeinen Kontext, aber man könnte beliebig in der Hierarchien der Kontexte aufsteigen.

Nun ist der Kontext *Unternehmen* in realen KMS nicht an der Spitze in einer scheinbar endlosen Kette von Kontexten zu finden. Eher wird die Hierarchie der Kontexte klein sein, wenn es sie überhaupt gibt. Behauptet wird nur, dass die Wahrnehmung der Menschen, die einen großen Teil ihres Lebens und ihrer (intellektuellen) Kraft in ein Unternehmen investieren, dergestalt ist, dass Unternehmenswissen als objektiv gilt und dass vor allem allein schon durch die Existenz eines zentralen Knowledge Officers suggeriert wird.

Ohne Zweifel wird aber innerhalb der meisten Unternehmen dem Prinzip *Weltwissen* gefolgt: Es wird davon ausgegangen, dass es Unternehmenswissen gibt, das von einem CKO verwaltet wird. Dieser delegiert Teilaufgaben an untergeordnete KO der jeweiligen Abteilung und Arbeitsgruppen, so dass eine Hierarchie von Knowledge Officers entsteht. Das entspricht exakt dem Ansatz des partitionierten Weltwissen, für den bereits weiter oben gezeigt wurde, dass er für P2P-Systeme untauglich ist.

Wenn nun die KM-Theorie zeigt, dass der Wissensaustausch inhärent einen P2P-Charakter hat und meist in informellen Netzen abläuft und gleichzeitig IT-Systeme auf dem Konzept des Weltwissen arbeiten, so kann das nicht zu befriedigenden Ergebnissen führen. Das Problem zeigt sich bereits in fehlender Akzeptanz der Nutzer und dem Speichern von Wissen außerhalb des zentralen Unternehmens-KMS in Mailboxen und eigenen Verzeichnissen.

In ähnlicher Richtung argumentieren auch Bonifacio und andere und fordern die Entwicklung von verteilten KMS. Die Verteilung soll dabei nicht nur technischer, sondern viel wichtiger konzeptueller Natur. Wissen soll näher an den Menschen gebracht werden.

Ihrem Ansatz soll sich im Folgenden zugewandt werden. Er kann als der derzeit am weitesten fortgeschrittene Ansatz im Bereich verteilter KMS bezeichnet werden.

2.4.4 Knowledge Nodes - Bausteine verteilten Wissens

Bonifacio, Bouquet und Cuel stellten 2002 in [BBC02] ein Modell von verteilten Wissensmanagementsystemen (*Distributed Knowledge Management Systems (DKMS)*) vor. Kernkonzept ihres Modells ist die *Knowledge Node*. Sie ist eine Abstraktion von Individuen und formellen und informellen Gruppen. Eine Knowledge Node ist eine beliebige Position auf der ontologischen Dimension.

Knowledge Nodes haben folgende Eigenschaften:

- Sie verfügen über eine eigene Wissensbasis mit einer lokalen Ontologie.
- Sie können selbstständig entscheiden, mit welchen Knowledge Nodes sie kommunizieren wollen.
- Ein Wissensaustausch erfolgt nach einem Prozess namens *Meaning Negotiation*. Dabei werden die lokalen Ontologien zweier Knowledge Nodes verglichen. Wissen, das wechselseitig zu beiden lokalen Ontologien passt, wird ausgetauscht.

Das zentrale Konzept sind die Knowledge Nodes. Sie stehen für jede beliebige formelle oder informelle Einheit in einem Unternehmen, bis hin zu den einzelnen Personen. Sie

können als Modellierung von sozialen Gruppen angesehen werden. Als Beispiel werden genannt: Arbeitsgruppen, Communities of Practice (CoP) und Personen.

Der Fluss von Wissen innerhalb eines Unternehmens wird auf eine Menge von Austausch zwischen Knowledge Nodes zurückgeführt. Diese sind autonom in dem Sinne, dass sie selbstständig entscheiden können, wann sie mit welcher anderen Knowledge Node Wissen austauschen wollen. Das Wissen wird in einer lokalen Ontologie gespeichert.

Die Grundkonzepte ähneln in vielen Teilen dem ACP-Modell. Knowledge Nodes sind wie ACP-Peers autonom, sowohl in der Haltung der Daten/des Wissens als auch in der Kommunikation mit anderen.

Das Unternehmenswissen setzt sich in irgendeiner Form aus der Menge aller lokalen Ontologien aller beteiligter Substrukturen des Unternehmens zusammen, die allesamt durch Knowledge Nodes modelliert werden können.

Prinzipiell räumt das Konzept mit der Notwendigkeit einer zentralen Wissensbasis auf und setzt auf das Prinzip der Mikrowelten, als die jede einzelne lokale Ontologie angesehen werden kann. Der Prozess des *Meaning Negotiation* erlaubt den Austausch von Wissen, was eine (teilweise) Aggregation der Mikrowelten erlaubt. Das *Meaning Negotiation* kann als Mediatorfunktion interpretiert werden.

Das sind die Grundprinzipien des Modells. Konkretisiert wird das Modell durch eine Sprache zur Repräsentation der lokalen Ontologien, der Context Markup Language (CTXML), die im Folgenden kurz vorgestellt werden soll.

Context Markup Language (CTXML) Mittels CTXML ([BDS02]) wird die lokale Ontologie einer Knowledge Node beschrieben. CTXML zeichnet sich durch folgende Eigenschaften aus:

- Kontext ist ein Tripel aus einem Identifier für den Kontext, einer Menge von Metainformationen in Form von Attribut/Value-Paaren und „the real content of a context, namely a conceptualization, and is represented as a labelled tree“.
- Bestandteil der Sprache sind drei hierarchische Beziehungen zwischen Kontexten: *is – a*, *is – part* und *inst – of*. Andere, auch nicht hierarchische Beziehungen sind erlaubt, werden aber nicht diskutiert.
- CTXML hat ein eigenes proprietäres XML-Schema, das nicht kompatibel zu anderen Formaten ist, insbesondere nicht zu Standards oder Quasi-Standards der Wissenrepräsentation.
- Die Metadaten sind Besitzer, Gruppe, Zugriffsrechte und eine Änderungshistorie des Kontextes.

Es besteht die Möglichkeit, ein Mapping zwischen zwei Kontexten zu definieren. In einem Mapping kann ein Konzept von einem Kontext (*source*) auf ein Konzept in einem anderen Kontext (*target*) abgebildet werden. Es kann ein spezieller Mediator definiert werden, der die Umwandlung zwischen zwei konkreten Kontexten realisiert ([BS02]). Ein allgemeiner Mediator existiert nicht.

Kritik am Ansatz Knowledge Nodes

Das Konzept der Knowledge Nodes kommt ohne einen zentralen Server aus und löst damit das Kernproblem des zentralistischen Ansatzes. An die Stelle des zentralen Servers treten autonome Knowledge Nodes, die mit beliebigen anderen Knowledge Nodes kommunizieren können. Jede Knowledge Node (KN) verfügt über eine eigene lokale Ontologie. Es existiert eine Mediatorfunktion, die das Übermitteln von Wissen von einer KN zur anderen erlaubt. Das ist eine genaue Umsetzung des Konzeptes der aggregierten Mikrowelten.

Trotz aller Vorteile ist das Konzept an einigen Stellen sowohl konzeptuell als auch technisch nicht konsistent:

- Knowledge Node ist eine Abstraktion von einer *beliebigen* Position auf der ontologischen Dimension. Im Modell steht damit ein Knowledge Node, die eine Person repräsentiert, gleichberechtigt neben einer Knowledge Node, die für eine Gruppe steht. Beide verfügen über eine lokale Ontologie und können mit anderen Knowledge Nodes Wissen austauschen.

Diese Abstraktion erzeugt einige Schwierigkeiten. Es ist zum einen nicht geklärt, wie Wissen in eine lokale Ontologie gelangt. Bei einer Person ist das schnell erklärt: Sie kann neues Wissen entwickeln. Wie aber bereits weiter oben betont wurde, ist die Entstehung neuen Wissens nur bei Personen möglich, nicht aber bei Gruppen. Wie gelangt also neues Wissen in die Gruppe? Sicherlich durch die Mitglieder der Gruppe. Es muss also eine besondere Beziehung zwischen der Knowledge Node „Gruppe“ und den Knowledge Nodes der Mitglieder geben. Im anderen Falle würde sich das Wissen in der Gruppe nicht ändern können oder die lokale Ontologie der Gruppe würde anderweitig verändert werden, was durch das Modell nicht erklärt ist. Letztere Möglichkeit wäre eine große Schwäche des Modells, die erste Variante klingt unwahrscheinlich.

Wahrscheinlicher ist, dass diese Tatsache bisher noch nicht beachtet wurde.

- Ein Austausch von Wissen fände dann zwischen KNs statt, wenn deren lokale Ontologien (in Teilen) zusammenpassten, wird in [BBC02] erklärt. Dieser Ansatz greift für einen realen Einsatz deutlich zu kurz:

Wissen wird in KNs gespeichert und mittels einer lokalen Ontologie strukturiert. Das Strukturierungsmittel wird aber gleichzeitig als ein Wille zum Wissensaustausch interpretiert. Das ist für reale Anwendungen unzulässig, denn es wird das Speichern von privaten bzw. geheimen Informationen unmöglich: Angenommen, es werden zwei Kategorien angelegt *public* und *private*. Beiden wird Wissen zugeordnet. Eine zweite KN möge die gleichen Kategorien anlegen. Das *Meaning Negotiation* wird erkennen, dass beide Kategorien in beiden KN vorkommen und Wissen aus beiden Kategorien austauschen. Es besteht keine Möglichkeit, in irgendeiner Form zu limitieren, dass z. B. privates Wissen nicht ausgetauscht werden soll.

Dass das Vorhandensein von Wissen und der Wunsch zum Austausch streng getrennt werden muss, wurde mehrfach diskutiert ([SG02, SP02, SG03]). Beides wird

in diesem Ansatz unzulässig vermischt.

- Die vorgestellte Sprache CTXML erlaubt nur die Beschreibung von Vererbungs- und Aggregationsbeziehungen. Das ist unzureichend für eine Ontologie. Es heißt zwar, man könne andere Beziehungstypen definieren. Dann stellt sich aber die Frage, wieso anstelle einer proprietären Sprache nicht eine standardisierte Sprache, wie RDF oder Topic Maps, gewählt wurde, die die Eigenschaften von CTXML hat und darüber hinausgeht. Es scheint, dass das Problem einmal erkannt wurde, denn in [BDS02] heißt es, man könne auch andere Repräsentationssprachen benutzen. Das sollte man tun.
- Das *Meaning Negotiation* erlaubt den Vergleich zwischen lokalen, völlig autonom entwickelten Ontologien. Wie in Abschnitt 2.2.2 bereits gezeigt wurde, ist das ohne weitere Einschränkungen prinzipiell nicht möglich. Es werden aber weder Einschränkungen definiert noch wird eine Andeutung über eine mögliche Realisierung dieser Mediatorfunktion gemacht.

2.4.5 Zusammenfassung

Knowledge Management ist als Managementtheorie entstanden und hat zum Ziel, die Wissensflüsse in Unternehmen zu optimieren. Wissen wird im KM immer im Prozess gesehen, also als etwas, das Mitarbeiter zu qualifizierten Handlungen befähigt. Es war zu sehen, dass eine wesentliche Unterscheidung zwischen explizitem und implizitem Wissen gemacht werden kann und muss. Es war zu sehen, dass ein Charakteristikum wissensintensiver Prozesse darin besteht, dass viel implizites Wissen der Mitarbeiter dazu vonnöten ist. Anders gesagt: Sobald ein Prozess vollständig durch explizites Wissen beschrieben werden kann, hört er auf, ein wissensintensiver Prozess zu sein.

Die Theorie und empirische Studien zeigen, dass Wissen in Unternehmen vorrangig in informellen Netzen und direkt zwischen Menschen weitergegeben wird, was sich auch als P2P-Austausch bezeichnen lässt.

IT kann wissensintensive Prozesse unterstützen, sie aber nicht automatisieren. Es wurden Beispiele genannt, wo IT Prozesse im SECI-Modell unterstützen kann. Es war zu sehen, dass nahezu alle KMS in der Architektur einem zentralen Ansatz folgen, der explizit oder implizit dem Ansatz *partitioniertes Weltwissen* folgt. Bereits in einem vorherigen Abschnitt wurde gezeigt, dass P2P-Systeme mit einem solchen Ansatz nicht arbeiten können.

Dieser Widerspruch zwischen Theorie und unterstützender IT hat sich als aktuelles Hauptproblem herausgestellt. Diese Kritik wurde bereits auch von anderen vorgetragen und führte zu Forderungen nach verteilten KMS. Der am weitesten fortgeschrittene Ansatz der Knowledge Nodes wurde vorgestellt. Es war zu sehen, dass dieser Ansatz nicht konsistent ist.

Es gibt offenbar keine konsistente Erklärung, die zum einen zeigt, wie Wissensflüsse in Unternehmen auf P2P-Austausche zwischen Personen zurückgeführt werden können und die zweitens detailreich genug ist, um als Basis für die Entwicklung einer Referenzarchi-

tektur für ein P2P-KMS herzuhalten. Eine solche Erklärung wird im nächsten Abschnitt präsentiert werden.

2.5 Ein P2P Modell für das Knowledge Management

Empirisch wurde festgestellt, dass Wissen in Unternehmen im Wesentlichen durch P2P-Austausche zwischen Personen weitergegeben wird. Bisher wurde nicht hinreichend geklärt, wie, lediglich basierend auf solchen P2P-Kommunikationen und vor dem Hintergrund der Theorie von Nonaka und Takeuchi, Wissen in Unternehmen und in den verschiedenen Hierarchieebenen eines Unternehmens fließt. Eine solche Erklärung wird hier geliefert.

Als Einstieg soll noch einmal die ontologische Dimension des Modells von Nonaka und Takeuchi hinterfragt werden, siehe Abschnitt 2.4.1. In ihren Modellen sind auf dieser Dimension nur die Klassen der Individuen, Gruppen und Organisationen dargestellt. Sie hinterfragen nicht, welche Kommunikationsbeziehungen konkret zwischen tatsächlichen Instanzen dieser Klassen erfolgen. Das soll im Folgenden geschehen. Zunächst wird eine alternative Darstellung der Entitäten der ontologischen Dimension gewählt, siehe Abbildung 2.13.

Das Dreieck möge als Querschnitt eines auf der Spitze stehenden Kegels gedeutet werden. Es beinhaltet drei der vier Positionen der ontologischen Dimension: Individuum, Gruppe und Organisation. In dieser Darstellung werden sie als soziale Systeme aufgefasst, die in einer Subsystem-System-Beziehung stehen, wobei ein Individuum Teil einer Gruppe und diese Teil einer Organisation ist. Individuen bzw. Gruppen können auch Teil mehrerer Gruppen bzw. Organisationen sein, was hier nicht dargestellt ist. Zunächst soll sich der Übersichtlichkeit wegen auf den einfachen Fall konzentriert werden. Komplexere erschließen sich später.

Die Intraorganisation ist nicht explizit dargestellt. Man möge sich weitere Organisationen in der Umgebung vorstellen. Die Pfeile in der Abbildung deuten die möglichen Wege des Wissensaustausches an, die bereits diskutiert wurden. Es ist zu sehen, dass jedes System mit jedem Wissen austauschen kann. Der Austausch von Wissen zwischen Individuum, Gruppe und Organisation wird als Wissensaustausch zwischen Subsystemen und einem übergeordneten System interpretiert, während Kommunikation mit anderen Organisationen als eine Kommunikation mit der Umwelt interpretiert wird. Es erscheint legitim, eine Organisation als *ein* System zu interpretieren, das aus Gruppen und Individuen besteht, und andere Organisationen als andere Systeme. Das wird auch schon durch die Bezeichnung der *Intra*-organisation von Nonaka und Takeuchi impliziert.

Der Wissensaustausch soll nun etwas genauer hinterfragt werden. Unstrittig ist, *dass* ein Austausch zwischen den Ebenen stattfindet. Bis hierher wurde noch nicht hinreichend erklärt, *wie* er stattfindet. Was bedeutet es beispielsweise, wenn eine Gruppe mit der Organisation Wissen austauscht? Es gibt kein Wesen „Gruppe“ oder „Organisation“, das den Mund öffnet und Wissen verkündet oder eine Mitteilung schreibt oder eines davon empfängt. Vielmehr ist es so, dass ein Individuum einem anderen Wissen übermittelt. Ein Austausch zwischen Gruppe und Organisation wird es dadurch, dass

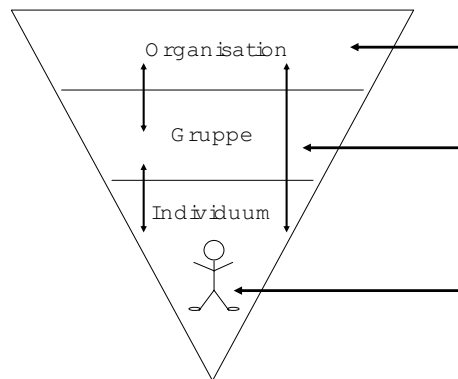


Abbildung 2.13: Das System Organisation als Kegel

ein Individuum in der *Rolle eines Gruppenmitglieds agiert* und das andere in der *Rolle eines Organisationsmitglieds*.

Diese Tatsache soll durch eine weitere Darstellung veranschaulicht werden, siehe Abbildung 2.14. Man möge sich vorstellen, dass diese Abbildung entstanden ist, indem der Kegel aus Abbildung 2.13 von oben betrachtet wird. Die konzentrischen Kreise um das Individuum sind jeweils die Systemgrenzen von Gruppe und Organisation, die in der Seitenansicht Linien sind¹⁴.

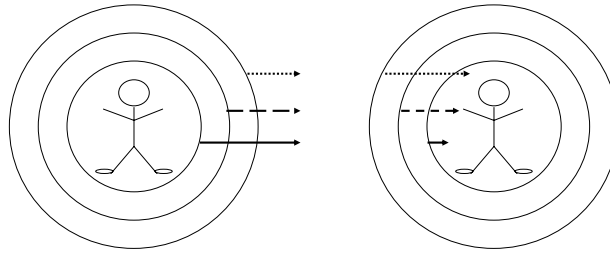
In der Abbildung bietet das linke Individuum Wissen an, was durch die nach außen gerichteten Pfeile symbolisiert wird. In der Abbildung agiert es sogar in drei verschiedenen Rollen: Einmal bietet es individuell Wissen an (durchgehender Pfeil, der vom Kreis des Individuums ausgeht), es bietet Wissen der Gruppe an (Pfeil darüber, ausgehend vom Kreis der Gruppe) und es bietet Wissen der Organisation an (Pfeil darüber, ausgehend vom Kreis der Organisation).

Das Individuum auf der rechten Seite nimmt Wissen entgegen, und zwar auch in drei verschiedenen Rollen, einmal als Mitglied der Organisation, einmal als Mitglied der Gruppe und einmal aus eigenem Interesse. Die Motivation für die Wissensaufnahme wird durch den Kreis symbolisiert, von dem der nach innen gerichtete Pfeil ausgeht.

Diese Darstellung zeigt sowohl, *wie* ein Austausch erfolgt, nämlich immer zwischen Individuen, und zeigt gleichzeitig, in welchen Rollen die Individuen dabei agieren und zwischen welchen Positionen auf der ontologischen Dimension ein Austausch erfolgt.

In Abschnitt 2.1 wurde erklärt, wie es sozialen Systemen gelingt, trotz beschränkter

¹⁴Das Strichmännchen muss sich dabei spontan auf den Rücken geworfen haben, als es bemerkte, dass jemand von oben auf den Kegel schaut. Ansonsten hätte man nur den kahlen und runden Strichmännchenkopf gesehen, der nicht von den anderen Ringen unterschieden werden könnte. Danke.



6

Abbildung 2.14: Kreismodell der ontologischen Dimension

Ressourcen Wissen aus der Umgebung aufzunehmen und Handlungsalternativen auszuwählen. Sie filtern eingehende Informationen anhand einer Relevanzbewertung und wählen anhand einer Strategie eine Handlungsalternative aus.

Hier geht es eingeschränkt um den Austausch von Wissen. Die eingehenden Informationen sind demnach Wissen und mögliche Handlungen bestehen darin, zu entscheiden, ob und welches Wissen gesendet werden soll. Eine weitere Abbildung soll die Relevanzfilter und Strategien zusätzlich im Kreismodell veranschaulichen, siehe Abbildung 2.15.

Was geschieht, wenn Wissen aus der Umgebung angeboten wird? Das Individuum entscheidet je nach Rolle, die es gerade spielt, ob es das Wissen annehmen will oder nicht. In der Rolle des Mitglieds der Organisation wird es sich deren Relevanzfilters bedienen und analog, wenn es als Gruppenmitglied oder Privatperson agiert.

Die Entscheidung, ob Wissen gesendet werden soll, wird ähnlich getroffen. Als Mitglied der Organisation entscheidet das Individuum anhand der Strategie der Organisation, ob und welches Wissen gesendet wird. Analog entscheidet es in den Rollen Gruppenmitglied und Privatperson. Ein Individuum kann gleichzeitig mehrere Rollen spielen und damit auch gleichzeitig in verschiedenen Rollen Wissen senden und entgegennehmen.

Ein kleines Beispiel soll diese Vorgänge illustrieren. Angenommen, ein Mitarbeiter eines Softwareunternehmens ist Implementierer. Er arbeitet in einer Abteilung, in der die vorgeschriebene Entwicklungsplattform „FastCode“ ist. In seiner Freizeit programmiert er zusätzlich für Open Source Entwicklungen. Hier benutzt er die Plattform „OpenQuickCode“, die er gegenüber „FastCode“ bevorzugt. Als ein neuer Mitarbeiter in die Abteilung kommt, informiert er sich u. a. bei unserem Mitarbeiter nach der passenden Entwicklungsplattform. Dieser antwortet ihm Folgendes: „In der Abteilung benutzen wir FastCode. Das ist ganz praktisch, vor allem weil jeder damit arbeitet und es damit keine Reibungsverluste gibt, wenn man das Projekt wechselt. Wenn du aber meine persönliche Meinung hören willst: Ich finde OpenQuickCode deutlich besser. Das benutze ich auch privat.“ Der neue Mitarbeiter installiert nach dieser Mitteilung FastCode auf seinem

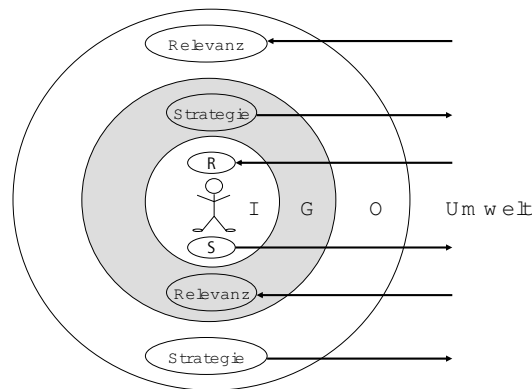


Abbildung 2.15: Kreismodell mit Relevanzfilter und Strategie

Rechner und schaut sich aber gleich auch einmal OpenQuickCode an.

In diesem kleinen Dialog fanden zwei Arten von Wissensaustausch statt. Der dienstältere Mitarbeiter wurde nach einer Entwicklungsplattform gefragt, antwortete zunächst in seiner Rolle als Gruppenmitglied und empfahl FastCode. Das ist Standard in der Gruppe. Danach wechselte er aber in die individuelle Rolle und empfahl OpenQuickCode. Der neue Mitarbeiter hat diese doppelte Information richtig verstanden und entsprechend gehandelt. Als neues Gruppenmitglied hat er das Gruppenwissen angenommen und FastCode installiert. Als Individuum beschäftigt er sich aber auch gleich mit OpenQuickCode. Aus der Handlung des neuen Mitarbeiters kann man auch schließen, dass er ein privates Interesse an Entwicklung hat, sonst hätte er sich nicht mit OpenQuickCode beschäftigt. Denn als Gruppenmitglied muss er OpenQuickCode nicht kennen, es genügt FastCode. Demnach hat ihn ein privates Interesse dazu gebracht, auch den zweiten Hinweis anzunehmen. Abbildung 2.16 illustriert den Vorgang.

In der Realität ist es sicherlich nicht immer möglich, exakt zu bestimmen, in welcher Rolle eine Individuum gerade agiert. Im Geschäftsbereich ist die Situation oftmals eindeutig. Bei Messen ist es z. B. klar, dass man zwar mit Menschen redet, aber diese in der Rolle des Unternehmensmitarbeiters agieren. Ebenfalls agieren Mitarbeiter bei Gesprächen mit (potenziellen) Kunden als Mitarbeiter ihres Unternehmens. Auch hier gibt es natürlich Ausnahmen, wenn ein Mitarbeiter beispielsweise die Absicht hat, das Unternehmen zu wechseln. Dann agiert er oft in zwei Rollen, einmal in der (noch) offiziellen Rolle des loyalen Mitarbeiters, aber gleichzeitig auch in der Rolle eines Menschen auf der Suche nach einem neuen Unternehmen. Diese Wechsel der Rollen sind in der Kommunikation oftmals auch eindeutig wahrnehmbar, wenn nicht gar irritierend.

Dieses Erklärung führt den Austausch von Wissen zwischen beliebigen Punkten auf

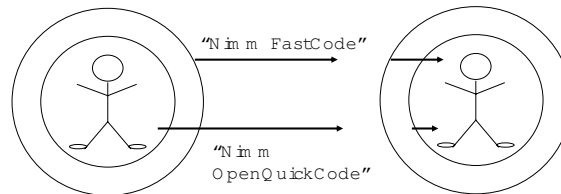


Abbildung 2.16: Wissensaustausch zwischen zwei ontologischen Punkten

der ontologischen Dimension ausschließlich auf die Kommunikation zwischen Individuen zurück, die in unterschiedlichen Rollen agieren. Es ist damit ein reines P2P-Modell, wobei die Peers nur Individuen sind. Die Individuen werden als soziales System verstanden, die Teil weiterer sozialer Systeme sein können usw. Dank dieser Interpretation kann erklärt werden, dass die Filterung von Wissen anhand einer Relevanzbewertung und die Entscheidung, ob und welches Wissen gesendet wird, anhand einer Strategie erfolgen.

Es ist auch einfach zu erklären, wie ein Individuum Mitglied mehrerer Gruppen sein kann. Es übernimmt einfach von allen Gruppen die Relevanzfilter und Strategien und handelt entsprechend.

Dieser Ansatz ist konsistent mit der Theorie von Nonaka und Takeuchi, verfeinert aber deren Vorstellung über die Bewegung von Wissen auf der ontologischen Dimension. Er rückt das Individuum in den Mittelpunkt der Betrachtung, was vor allem auch Davenport und Prusak, basierend auf ihren empirischen Untersuchungen, stets verlangen. Gruppen und Organisationen werden zu Rollen, die ein Individuum spielt. Ein Individuum kann als autonom angenommen werden, im Sinne, dass es selbst sein Wissen verwalten kann und frei ist in der Entscheidung, mit wem es kommuniziert.

Dieser Ansatz erklärt aber noch nicht, was Gruppen- bzw. Unternehmenswissen ist. Die Weiterentwicklung dieses Ansatzes liegt auf der Hand. Ein Individuum ist in seinen Eigenschaften kompatibel mit dem ACP-Modell. Mit diesem ist es aber bereits gelungen, Gruppenbeziehungen darzustellen. Es gilt, diesen Ansatz zu benutzen, um das ACP-Modell für das Knowledge Management zu spezialisieren. Ist das geschehen, existiert eine Referenzarchitektur für ein konsistentes P2P-KMS.

2.6 Zusammenfassung

In diesem Kapitel wurden wesentliche Quellen und relevante Arbeiten des Knowledge Managements vorgestellt. Diese Übersicht wurde auf Gebiete eingeschränkt, die sich mit dem Austausch von Wissen in Gruppen beschäftigen.

Die Systemtheorie erklärte, wie Relevanzbewertungen und Strategien filternd auf den Wissensaustausch einwirken.

Es wurden Ansätze aus den Bereichen Wissensrepräsentation und Softwareagenten vorgestellt. Es war zu sehen, dass ein P2P-KMS zwingend dem Konzept der Mikrotheorien folgen muss. Auf diesen Gedanken wird im Kapitel 5 zurückgekommen werden, wenn beschrieben wird, wie Wissen modelliert und ausgetauscht werden kann. Der Abschnitt über Agenten diente der Abgrenzung. Agentensysteme haben deutlich höhere Ansprüche als (verteilte) KMS. Im folgenden Kapitel wird ein Modell vorgestellt werden, das mit nur wenigen und recht einfachen Grundkonzepten Basis eines P2P-Systems sein kann. Dieses ACP-Modell ist deutlich weniger komplex als übliche Agentensysteme.

Im Knowledge Management gilt es als anerkannt, dass der primäre Austausch von Wissen zwischen Individuen erfolgt. Es war zu sehen, dass diese Erkenntnis noch nicht explizit im SECI-Modell niedergelegt wurde. Aus diesem Grund wurden die Erkenntnisse aus der Systemtheorie sozialer Gruppen auf das SECI-Modell angewendet. Es entstand ein Modell für das Knowledge Management, das Wissensflüsse auf den Austausch von Wissen zwischen Individuen zurückführen kann. Ein solches P2P-KM Modell existierte explizit noch nicht, wurde aber benötigt.

Die Notwendigkeit ergab sich aus einer Analyse des Standes der Entwicklung üblicher Knowledge Management Systeme (KMS). Es war zu sehen, dass diese Systeme einem zentralistischen Ansatz folgen, der Probleme aufwirft. Lösung einiger Probleme stellt eine verteiltes KMS (DKMS) dar. Exemplarisch wurde ein weit entwickelte Konzept eines DKMS von Bonifacio, Bouquet und Cuel vorgestellt. Es war zu sehen, dass das nicht konsequente Umsetzen des P2P-Paradigmas Probleme aufwarf. Daraus ergab sich die Notwendigkeit eines konsistenten und vollständig auf dem P2P-Pradigma beruhenden Modells für das Knowledge Management. Das Modell P2P-KM wird im Kapitel 4 Basis einer Referenzarchitektur für ein P2P-KMS sein.

Im Folgenden wird ein abstraktes P2P-Modell vorgestellt. Es enthält die zentralen Konzepte und beschreibt auf einer allgemeinen Ebene den Austausch von Daten zwischen Peers.

3 Autonome kontextsensitive Peers – ACP-Modell

Im vorangegangenen Kapitel wurden verwandte Arbeiten für die Beschreibung von Wissensflüssen vorgestellt. Am Ende wurde ein Modell präsentiert, das Wissensflüsse auf den P2P-Austausch von Wissen zwischen Individuen zurückführt. Im vorangegangenen Kapitel wurden auch Agentensysteme vorgestellt. Es war zu sehen, dass an Agentensysteme höhere Anforderungen gestellt werden als an KMS. Agentensysteme sollen z. B. menschliche Ziele und soziales Verhalten nachstellen. Diese Forderung steht bei KMS nicht. Ein KMS ist vielmehr ein Hilfsmittel für Menschen, die mit wissensintensiven Arbeiten betraut sind. Ein KMS soll einen Menschen niemals ersetzen.

In dieser Arbeit wird eine Softwarearchitektur für ein P2P-KMS entwickelt. Nachdem im letzten Kapitel ein P2P-Modell für das Knowledge Management entwickelt wurde, soll in diesem Kapitel ein allgemeines P2P-Modell entwickelt werden.

Die im Folgenden vorgestellten Konzepte sind bewusst allgemein gehalten und lassen Raum für Konkretisierungen. Trotz ihrer Allgemeinheit lassen sich aber bereits auf dieser abstrakten Ebene Beziehungen zwischen Peers modellieren, die in P2P-Anwendungen üblich sind. Exemplarisch wird gezeigt, wie das Modell erlaubt Stellvertreterregelungen zu beschreiben, den Zusammenschluss von mehreren Peers zu einer Gruppe, die Erzeugung einer Client-Server-Beziehung zwischen Peers und einer Synchronisationsbeziehung. Diese Beziehungen werden ausschließlich durch die Verteilung von Regeln über den kontextbasierten Austausch von Daten realisiert, bedürfen also keiner weiteren speziellen Konzepte, wie Synchronisationsprotokolle o.Ä.

Die Peers werden im vorgestellten Modell als autonome Entitäten angesehen, die in der Lage sind, die Umgebung wahrzunehmen. Autonomie bedeutet hier, dass sie ohne Rückfrage mit einer weiteren Entität in der Lage sind, zu entscheiden, ob und welche Daten sie austauschen wollen. Aus diesem Grund wird es als Modell *autonomous context-aware peers (ACP)* bzw. *autonomer kontextsensitiver Peers* bezeichnet, kurz als *ACP-Modell*.

Das Modell betrachtet nicht, über welche Kanäle Daten ausgetauscht werden, welche innere Struktur Daten haben, wozu diese dienen. Ebenfalls sind Methoden der Datensicherheit nicht Bestandteil des Modells.

Konzeptuell werden all diese Eigenschaften als Teil eines später näher definierten *Umgebungskontextes* aufgefasst. So sind Qualitätseigenschaften des Kommunikationskanals (Datenrate, Verschlüsselung usw.) Eigenschaften der Umgebung, in der eine Kommunikation erfolgt.

Das ACP-Modell kann in verschiedene Richtungen konkretisiert werden. In dieser Arbeit wird nur ein Weg beschritten: Es dient lediglich als Grundlage für eine Referenzar-

chitektur für ein P2P-KMS, das bereits im folgenden Kapitel vorgestellt wird.

Ohne Informationsverlust hätte man auch dieses Kapitel mit dem folgenden vereinen können und direkt ein P2P-Modell für das Knowledge Management entwerfen können. Das ist aus Gründen der Didaktik nicht erfolgt. Es wird hier dem Vorgehen gefolgt, wie es aus dem Software-Engineering bekannt ist, wenn abstrakte Klassen definiert werden. Abstrakte Klassen beschreiben allgemeine Verfahren und Zusammenhänge, ohne dass sie direkt instanziiert werden können. Sie verhält es sich auch mit dem ACP-Modell. Dieses Modell beschreibt keine konkrete Anwendung. Es beschreibt lediglich, wie autonome Peers miteinander interagieren können. Diese Prinzipien werden später konkret für den Austausch von Wissen genutzt werden. Sie könnten aber auch in anderen Gebiete eingesetzt werden, die in dieser Arbeit nicht ausgeführt werden.

3.1 Das Modell

Die einzelnen Bestandteile des Modells werden im Folgenden vorgestellt werden.

3.1.1 Peer

Das Kernkonzept des Modells ist das *Peer*. Ein Peer hat folgende Eigenschaften:

- In einer eigenen *Datenbasis* (*DB*) kann es Daten strukturiert speichern.
- Es kann Eigenschaften der Umgebung wahrnehmen.
- Es kann eine Bezeichnung tragen.
- Es kann mit anderen Peers Daten austauschen.
- Es verfügt über Ports, die den Austausch mit anderen Peers steuern.
- Es kann Ports mit anderen Peers austauschen.
- Es ist autonom in dem Sinne, dass es vollständig allein über seine Daten und sein Austauschverhalten entscheiden kann.

3.1.2 Kontext

Das zweite Kernkonzept ist der Kontext. Für ihn gilt:

- Ein Kontext (X) definiert eine Beziehung zwischen zwei Entitäten.
- Ein Kontext fügt einer Entität inhaltlich nichts hinzu.
- Im ACP-Modell sind die möglichen Entitäten Peer, Datum und Eigenschaften der Umgebung. Die Einführung weiterer Entitäten zur Konkretisierung dieses Modells ist nicht verboten.
- Ein Kontext ist ein spezielles Datum, siehe nächster Abschnitt.

Kontexte können sowohl existierende Beziehungen beschreiben als auch nicht existierende. Der erste Fall findet Anwendung, wenn z. B. Daten strukturiert werden, siehe folgender Abschnitt. Der zweite Fall findet Anwendung, wenn Erwartungen bzw. Bedingungen formuliert werden. Dann wird von einem *Erwartungskontext* (X_E) gesprochen, siehe auch Abschnitt 3.1.8.

3.1.3 Daten

Für die Daten gilt:

- Daten werden in einer Datenbasis eines Peers gespeichert.
- Die Strukturierung der Daten erfolgt durch Datenkontexte (X_D), das sind spezielle Kontexte, die Daten anderen Daten oder Entitäten zuordnen. Beispiele von Zuordnungen sind inhaltliche Nähe von Daten, Hierarchiebeziehungen, aber auch die Definition von Gültigkeitsbereichen.
- Jedes Datum hat einen Eigentümer. Auch diese Eigenschaft kann als Datenkontext interpretiert werden, in dem ein Datum einem Eigentümer (Peer) zugeordnet wird, siehe Abschnitt 3.1.6.

3.1.4 Umgebung

Für die Umgebung und deren Wahrnehmung gilt:

- *Umgebung* eines Peers ist alles, das nicht zum Peer gehört. Zur Umgebung gehören damit auch andere Peers.
- Eine Umgebung kann unterschiedliche Eigenschaften haben. Diese können prinzipiell durch ein Peer wahrgenommen werden.¹
- Ein *Umgebungskontext* (X_U) stellt die Beziehung zwischen dem Peer und dem von ihm beobachteten Umgebungswerten zu einem bestimmten Zeitpunkt dar. Da sich Umgebungswerte ändern können, sind auch die Umgebungskontexte Änderungen unterworfen.

3.1.5 Bezeichnung

Für Bezeichnungen der Peers gilt:

- Jedes Peer kann eine Bezeichnung führen, muss aber nicht.
- Ein Peer kann seine Bezeichnung anderen Peers bekannt geben.

¹Diese Aussage soll unterstreichen, dass die Umgebung in diesem Modell prinzipiell vollständig erkennbar ist durch ein Peer. Voraussetzung dafür ist, dass das Peer über ausreichende Mittel (z. B. Sensoren) verfügt, um alle Eigenschaften auch praktisch zu erkennen.

- Ein Peer, dessen Bezeichnung einem Peer nicht bekannt ist, wird *anonymes* Peer genannt.
- Ein Peer kann unterschiedliche Bezeichnungen gleichzeitig benutzen.

3.1.6 Eigentümer

Für Eigentümer von Daten gilt:

- Ein Eigentümer ist ein Peer.
- Ein Peer, das ein Datum anlegt, ist dessen Eigentümer.
- Die Eigentümerschaft kann geändert werden.
- Ein anonymes Peer kann ein Eigentümer sein. Damit wird ausgedrückt, dass der Eigentümer ungekannt ist.

3.1.7 Dateneinheiten

Es gilt:

- Daten werden immer in einem Datenkontext übertragen.
- Daten werden immer mit der Bezeichnung des Eigentümers übertragen.
- Der Austausch erfolgt immer direkt zwischen zwei Peers.
- Die Elemente Datum, Datenkontext und Eigentümer werden zusammen als *Dateneinheit (DE)* bezeichnet.

3.1.8 Ports

Ports beschreiben Bedingungen, die gelten müssen, um einen Austausch von Daten zu erlauben, und die zur Ermittlung der zu sendenden bzw. zu empfangenden Daten herangezogen werden müssen. Die Bedingungen werden anhand von Erwartungskontexten beschrieben. Weiterhin verfügen Ports über Funktionen, die Erwartungskontexte miteinander verknüpfen.

Es gibt zwei Typen von Ports: Incoming Ports (IP) und Outgoing Ports (OP).

Für Incoming Ports (IPs) gilt:

- Ein IP verfügt über Erwartungskontext bezüglich der Umgebungswerte (X_{EU}). Er beschreibt Werte, die gelten müssen, sodass ein Austausch von Daten erlaubt ist.
- Ein IP verfügt über eine Funktion f , die ermittelt, ob der aktuelle Umgebungs-kontext X_U zum Erwartungskontext X_{EU} passt. Ist das der Fall, so ist das Peer prinzipiell bereit, Daten über diesen Port zu empfangen.

- Ein IP verfügt über einen *Erwartungskontext bezüglich der Daten* (X_{ED}). Dieser beschreibt die Erwartungen, die erfüllt sein müssen, um Daten zu empfangen. Die Erwartungen können sich sowohl auf den Kontext beziehen, in dem zu empfangene Daten stehen sollen, als auch auf den Umgebungskontext.
- Ein IP verfügt über eine Funktion (g). Diese verknüpft X_U und X_{ED} in der Form, dass ein *effektiver Erwartungskontext bezüglich der Daten* (X_{EDU}) entsteht, der folgende Semantik hat: Der IP ist bereit Daten, die zu diesem Kontext passen, bei den aktuell beobachteten Umgebungswerten entgegenzunehmen.
- Ein IP verfügt über eine Funktion h , die zwei Datenkontexte vergleicht und deren Überschneidungen ermittelt. Das Ergebnis wird als *übereinstimmender Datenkontext (matching data context)* (X_{MD}) bezeichnet.
- Ein IP hat eine Empfängerbezeichnung. Unter dieser Bezeichnung bietet sich das Peer anderen Peers als Empfänger an.
- Ein IP kann alle Erwartungskontexte (X_{ED} , X_{EU} , aber auch die temporären Kontexte X_{EDU} und X_{MD}) und die Empfängerbezeichnung in der Umgebung verfügbar machen. Andere Peers können diese Werte als Umgebungswerte wahrnehmen.
- Ein Incoming Port wird beschrieben durch:

$$IP = \left\{ \begin{array}{c} X_{EU} \\ X_{ED} \\ f : (X_U, X_{EU}) \rightarrow true/false \\ g : (X_U, X_{ED}) \rightarrow X_{EDU} \\ h : (X_D, X_{EDU}) \rightarrow X_{MD} \\ Empfaenger \end{array} \right\}$$

Für Outgoing Ports gilt:

- Ein OP verfügt über einen Erwartungskontext bezüglich dem Umgebungskontext (X_{EU}).
- Ein OP verfügt über eine Funktion f , die anhand der beobachteten Umgebungswerte und X_{EU} ermittelt, ob prinzipiell ein Senden von Daten über diesen Port erlaubt ist.
- Ein OP hat einen Erwartungskontext bezüglich Daten (X_{SD}).² Dazu passende Daten dürfen gesendet werden.
- Ein IP verfügt über eine Funktion (g), die aus dem Umgebungskontext und dem Datenkontext X_{SD} einen Datenkontext ermittelt, der ausdrückt, zu welchem Kontext Daten beim konkreten Umgebungskontext gesendet werden sollen (X_{SDU}).

²SD kann als *zu sendende Daten* gelesen werden.

- Ein OP hat eine Senderbezeichnung. Unter dieser Bezeichnung bietet es anderen Peers Dateneinheiten an.
- Ein OP kann seine Erwartungskontexte (X_{SD} , X_{EU} , X_{SDU}) und seine Senderbezeichnung in der Umgebung verfügbar machen. Andere Peers können sie dann als Teil ihrer Umgebung wahrnehmen.
- Ein OP wird damit beschrieben durch:

$$OP = \left\{ \begin{array}{c} X_{EU} \\ X_{SD} \\ f : X_S \rightarrow true/false \\ g : (X_U, X_{SD}) \rightarrow X_{SDU} \\ Sender \end{array} \right\}$$

3.1.9 Extraktion und Assimilation von Daten

Jedes Peer verfügt außerdem über zwei Funktionen:

Extraktion ist eine Funktion, die, ausgehend von der Datenbasis (DB) des Peers, einem Datenkontext X_D einer Sender- und einer Empfängerbezeichnung eine Dateneinheit erzeugt:

$$extract : (DB, X_D, Sender, Empfaenger) \rightarrow (Daten, X_{D'}, Eigentuerer)$$

Assimilation ist eine Funktion, die, ausgehend von der Datenbasis (DB), dem Sender, dem Empfänger, dem Datenkontext (X_D), den Daten selbst und ihrem Eigentümer eine veränderte Datenbasis (DB') erzeugt, die alle Daten der Dateneinheit, Teile davon enthält oder aber unverändert bleibt.

$$assimilate : (DB, Daten, X_D, Eigentuerer, Sender, Empfaenger) \rightarrow DB'$$

Ports bestehen aus Erwartungskontexten und Funktionen. Kontexte sind spezielle Daten. Es wird davon ausgegangen, dass auch Funktionen auf Daten abgebildet werden können. Damit können Ports, wie andere Daten auch, zwischen Peers ausgetauscht werden.

3.1.10 Austausch von Dateneinheiten

Nach der Vorstellung der Ports, ihrer statischen Erwartungskontexte und den Funktionen zur Bearbeitung der Kontexte soll im Folgenden die Dynamik eines Datenaustausches im ACP-Modell beschrieben werden, siehe Abbildung 3.1.

Dort sind zwei Peers (Kreise) dargestellt. Ein OP ist als Blockpfeil symbolisiert, der aus dem Peer herauszeigt, der IP als hineinzeigender Blockpfeil. Die Dateneinheit ist als Kreis zwischen den Ports dargestellt. In der Abbildung sind der Übersicht wegen nur zwei

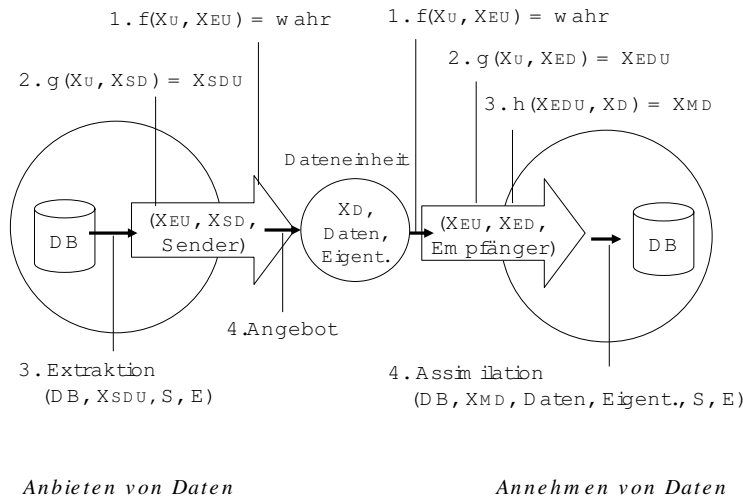


Abbildung 3.1: P2P Datenaustausch

Ports dargestellt. Prinzipiell kann ein Peer beliebig viele Ports haben. Das bedeutet auch, dass die im Folgenden beschriebenen Prozesse parallel beliebig oft stattfinden können.

Sobald ein Peer mit einem OP eine Veränderung in seiner Umgebung wahrnimmt, werden in jedem OP des Peers folgende Schritte ausgeführt:

1. Mit der Funktion f wird überprüft, ob prinzipiell ein Austausch stattfinden soll. Wenn nein, wird der Prozess abgebrochen.
2. Mittels g wird X_{SDU} ermittelt, der beschreibt, zu welchem Datenkontext passende Daten angeboten werden sollen. Ist der Kontext leer, wird der Prozess abgebrochen.
3. Der Extraktion wird X_{SDU} übergeben und zusätzlich die Empfängerbezeichnung, die beim potenziellen Empfänger beobachtet werden konnte. Wurde dort keine Bezeichnung offen gelegt, so wird als Empfänger das anonyme Peer angenommen.
Als Sender wird die Bezeichnung genutzt, die im OP definiert wurde. Ist das Resultat der Extraktion leer, wird der Prozess abgebrochen.
4. Die Dateneinheit wird dem potenziellen Empfänger des anderen Peers angeboten.

Dieser Prozess findet konzeptuell in jedem OP und für jeden potenziellen Empfänger separat und parallel statt.

Die Annahme von Dateneinheiten erfolgt in folgenden Schritten:

1. Mit der Funktion f wird überprüft, ob der Umgebungskontext den definierten Bedingungen genügt. Wenn nicht, hat das Peer bei der gegebenen Umgebung Daten anzunehmen.
2. Mittels g wird ermittelt, zu welchem Kontext Daten beim konkreten Umgebungskontext gehören müssen, um angenommen zu werden (X_{EDU}).
3. Mit h wird der Datenkontext (X_{MD}) ermittelt, der sowohl zum effektiven Erwartungskontext bezüglich der Daten (X_{EDU}) als auch zum Datenkontext der angebotenen Dateneinheit passt. Ist (X_{MD}) nicht leer, wird die Assimilation gestartet.
4. Die Assimilation empfängt als Parameter die just empfangene Dateneinheit, den gerade ermittelten Datenkontext X_{MD} , die Bezeichnung des Senders (soweit bekannt, ansonsten anonymes Peer) und die Empfängerbezeichnung, die im eigenen IP stand. Im Ergebnis der Assimilation werden Teile oder alle Daten integriert oder verworfen.

Die beschriebenen Prozesse können zu großen Teilen gleichzeitig stattfinden bzw. in einer durch das Modell nicht definierten Reihenfolge. So kann es in einem Anwendungsbereich sinnvoll sein, dass zunächst der Anbieter von Daten eine Dateneinheit erzeugt und sie potenziellen Empfängern anbietet, die dann entscheiden, ob sie sie assimilieren wollen oder nicht.

In anderen Szenarien kann es sinnvoll erscheinen, dass zunächst die potenziellen Empfänger ihre effektiven Datenkontexte X_{EDU} ermitteln und sie in der Umgebung verfügbar machen. Potenzielle Sender könnten anhand dieser Erwartungskontexte ihrerseits aktiv werden und passende Dateneinheiten anbieten.

Es kann aber auch ein Aushandlungsprozess stattfinden, in dem Ports zunächst Erwartungskontexte offen legen und diese ggf. sogar anpassen, je nachdem, welche Erwartungen sie bei einem potenziellen Sender bzw. Empfänger beobachten können. Das kann so weit führen, dass der Kontext X_{MD} veröffentlicht wird. Dieser beschreibt, zu welchem Kontext Daten tatsächlich in die Assimilation überführt wurden. Das kann potenziellen Sendern eine sehr genaue Information darüber liefern, unter welchen (äußeren) Umständen das Peer welche Daten annimmt.

Das Modell lässt offen, welche Herangehensweisen gewählt werden oder ob diese sich gar ändern können.

3.1.11 Fluss von Daten

Ein *Fluss von Daten* ist eine Aneinanderreihung von Datenaustauschen mit der zusätzlichen Bedingung, dass wenigstens ein Datum in jedem Austausch von einem Peer zum anderen übertragen wird und bei dem letzten Peer des Flusses assimiliert wird, d.h. in dessen Datenbasis abgelegt wird. Abbildung 3.2 illustriert diesen Vorgang.

Es ist zu sehen, dass ein Peer (A) eine Dateneinheit (DE_1) an B sendet, das seinerseits DE_2 an C sendet. Die Daten in den Dateneinheiten sollen jeweils D_1 und D_2 heißen. Für diese muss Folgendes gelten, wenn ein Fluss von Daten stattfindet:

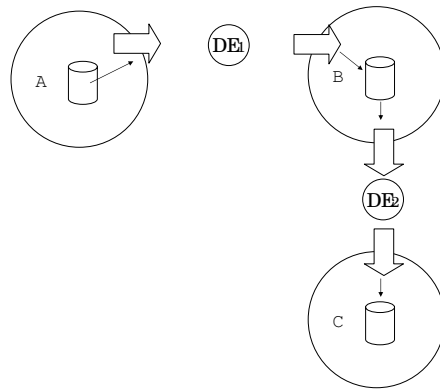


Abbildung 3.2: Datenfluss in einem Netz von Peers

- $D_1 \cap D_2 = D_3 \neq \emptyset$ und
- nach der Assimilation durch C existiert eine Teilmenge D_C der Datenbasis von C ($D_C \subset DB_C$) mit $D_C \subset D_3$ und $D_C \neq \emptyset$.

Man sagt dann, dass die Daten D_C von A nach C geflossen sind.

3.1.12 Zusammenfassung

Das Modell benennt nur sehr allgemeine Konzepte und beschreibt grob den kontextbasierten Austausch von Daten zwischen autonomen kontextsensitiven Peers. Das Modell führt das Konzept Kontext ein. Es werden zwei Anwendungsgebiete von Kontext benutzt, das sind die Strukturierung von Daten und die Formulierung von Erwartungen.

Es wird gezeigt, wie ein Austausch von Daten unter Berücksichtigung des Kontextes erfolgen kann. Die Regeln für den Austausch sind im Modell in Ports modelliert, die sowohl die Erwartungskontexte enthalten als auch Funktionen, die erlauben, Kontexte ineinander umzuwandeln.

Es wird kein Protokoll für den Austausch von Daten definiert. Es werden lediglich die jeweiligen Schritte definiert, die bei einem Austausch ausgeführt werden müssen, aber nicht deren Reihenfolge.

Die Funktionen, die Kontexte ineinander umwandeln, tragen bewusst keine sprechenden Namen. Eine Vergabe von Namen, die eine Semantik implizieren, ist auf dieser abstrakten Ebene gefährlich. Es ist nicht absehbar, welche konkreten Ausprägungen die Funktionen im abgeleiteten Modell haben werden. Es soll vermieden werden, dass Funktionsnamen dieses Modells zu Irritationen in abgeleiteten führen. Das wird durch eine abstrakte Benennung verhindert.

Trotz seiner Allgemeinheit lassen sich bereits mit diesen allgemeinen Konzepten Beziehungen zwischen Peers definieren. Dazu werden im Folgenden einige Beispiele präsentiert.

3.2 Peer-Beziehungen

Ports definieren die Regeln, auf deren Basis ein kontextbasierter Datenaustausch stattfindet. Peers können untereinander Daten und Ports austauschen. Das ermöglicht die Etablierung von speziellen Beziehungen zwischen Peers, wenn ein Peer einen Port von einem anderen Peer übernimmt und an dessen Stelle agiert. Einige dieser Beziehungen sollen im Folgenden diskutiert werden.

3.2.1 Proxy-Peers

Jeder kennt die Situation, wenn man für jemanden anderes etwas tut. So kann man einen Kollegen vertreten und an dessen Stelle etwas verkünden oder etwas entgegennehmen. Im WWW sind Proxies bekannte Entitäten zur Lastverteilung und zur Erhöhung der Ausfallsicherheit. Eine solche Stellvertreter-Beziehung lässt sich auch allgemein mit dem ACP-Modell beschreiben.

Im Folgenden möge ein Peer, das ein anderes Peer vertritt, *Proxy* heißen und das vertretene Peer *Master*. Übertragen auf das ACP-Modell verfügt eine solche Stellvertreterbeziehung über folgende Eigenschaften:

- Der Proxy verfügt über wenigstens einen Port, den es vom Master erhalten hat. Damit führt es seine Stellvertreterfunktion gegenüber dem Master aus.
- Trägt ein Proxy einen OP eines Masters, so muss es zusätzlich Daten vom Master erhalten, die es an Empfänger übertragen soll. Das bedeutet nicht, dass bei jedem Austausch alle Daten des Masters an interessierte Peers übermittelt werden. Auch in seiner Funktion als Proxy erfolgt ein kontextbasierter Austausch von Daten. Es gilt aber die Regel, dass ein Proxy nur Daten anbietet, die auch ein Master anbieten würde.
- Trägt ein Proxy einen IP, so nimmt es anstelle des Masters Daten entgegen. Der Proxy muss die empfangenen Dateneinheiten speichern und an den Master weiterleiten.

Diese Eigenschaften lassen sich durch einige Vorgaben erzeugen. Der Master definiert Folgendes:

1. Der Master definiert einen OP, über den er Proxy-Peers Ports anbietet. Dieser OP stellt einen Meta-Port dar, denn er übermittelt anderen Peers Regeln, die definieren, unter welchen Umständen sie ihn vertreten sollen.

Über diesen Port müssen ebenfalls Daten übermittelt werden, wenn der Master auch OPs an Proxies übermitteln will.

Der Extraktionsprozess dieses OPs stellt Dateneinheiten bereit, die Ports und Daten enthalten, die ein Proxy für seine Stellvertreterfunktion benötigt.

2. Bietet der Master auch IPs an Proxies an, muss er selber einen weiteren IP definieren. Dieser muss beschreiben, dass er von allen denkbaren Proxies Dateneinheiten entgegennehmen will, die die Proxies stellvertretend für ihn empfangen haben.

Der Assimilationsprozess dieses IPs kann beliebig komplex sein. Im einfachsten Fall kann er wie folgt ablaufen.

Proxies haben Dateneinheiten über einen IP des Masters empfangen und leiten diese Dateneinheiten an den Master weiter. Der Master selber verfügt aber auch über den IP, da er ihn schließlich definiert hat. Im einfachsten Fall kann er die weitergeleiteten Datenheiten entgegennehmen und genauso behandeln, als hätte er sie selber empfangen, und direkt dem Assimilationsprozess des entsprechenden IPs übergeben. Dieser Prozess wird im Folgenden anhand einer Abbildung noch einmal erläutert.

Ist ein Peer potenziell bereit, Proxy eines Masters zu werden, muss er Folgendes definieren:

1. Er definiert einen IP, der beschreibt, dass er von einem Master bereit ist, Ports und Daten entgegenzunehmen. Der Assimilationsprozess dieses IP muss dafür sorgen, dass die empfangenen Ports beim Proxy aktiviert werden und dass die Daten des Masters gespeichert werden.

Weiterhin muss dafür gesorgt werden, dass für jeden empfangenen OP ein Extraktionsprozess in der Form vorliegt, dass nur Daten des Masters über diesen OP angeboten werden.

Für jeden IP muss wiederum ein Assimilationsprozess definiert werden, der die Dateneinheiten lediglich entgegennimmt und speichert.

2. Hat der Proxy wenigstens einen IP vom Master entgegengenommen, so muss er einen weiteren OP definieren. Dieser OP beschreibt, dass er dem Master jederzeit die Dateneinheiten übergibt, die er stellvertretend für ihn über IPs entgegengenommen hat.

Der Extraktionsprozess des OPs ist recht einfach. Er muss lediglich die Dateneinheiten ermitteln, die der Proxy gespeichert hat, und sie unverändert weiterleiten.

Abbildung 3.3 illustriert den Vorgang. Es sind zwei Peers dargestellt, ein Master und ein Proxy. In einem Schritt 1 hat der Master jeweils einen IP und einen OP definiert. In einem Schritt 2 hat er sich entschieden, andere Peers als Proxies für diese beiden Ports werben zu wollen. Aus diesem Grund hat er die oben beschriebene Kombination von IP/OP anlegt, die in der Abbildung grafisch zusammengefasst durch einen Doppelpfeil dargestellt und mit Proxies beschriftet ist. Dieser Doppelpfeil steht sowohl für einen OP, der Ports und Daten anbietet, als auch für einen IP, der Dateneinheiten entgegennimmt, die Proxies stellvertretend empfangen haben.

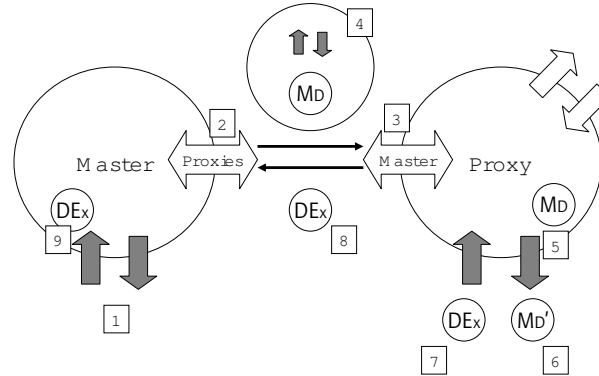


Abbildung 3.3: Master und Proxy

Das rechte Peer hat sich in einem Schritt 3 entschieden, als Proxy des linken Peers zu agieren. Es hat dazu ebenfalls die oben beschriebene IP/OP-Kombination definiert, die auch hier durch einen Doppelpfeil symbolisiert wird.

Im Schritt 4 überträgt der Master dem Proxy eine Dateneinheit. Diese enthält die zwei Ports und Daten des Masters (M_D), die über seinen IP angeboten werden sollen. Die Dateneinheit wird im Schritt 5 assimiliert, was dazu führt, dass nun auch der Proxy diese beiden Ports besitzt und die Daten des Master gespeichert hat.

In Schritt 6 wird beispielhaft gezeigt, dass der Proxy einem nicht dargestellten Peer Daten ($M_{D'}$) stellvertretend anbietet. Wie oben beschrieben gilt $M_{D'} \subset M_D$.

In einem Schritt 7 wird exemplarisch gezeigt, dass der Proxy eine Dateneinheit von einem nicht dargestellten Peer entgegennimmt. Diese Dateneinheit wird gespeichert (nicht dargestellt) und in einem Schritt 8 an den Master übermittelt.

Der Master hat nun eine Dateneinheit von einem Proxy empfangen, die dieser stellvertretend für ihn empfangen hat. Da er selber über den IP verfügt, der zu diesem Empfang geführt hat, könnte er die Dateneinheit direkt der Assimilation zuführen, so als ob er sie selber empfangen hat. Er kann zuvor aber auch andere Operationen ausführen. Es werden an dieser Stelle keine Einschränkungen definiert.

In der Abbildung verfügt der Proxy über zwei weitere Ports rechts oben. Diese möge er selber definiert haben und sie sollen an der Stelle unterstreichen, dass ein Peer neben seiner Funktion als Proxy auch weiterhin autonom agieren kann.

Löschen von Daten und Ports

Das Löschen von Daten und Ports wurde bisher nicht beachtet. Die einfachste Variante basiert auf folgender Grundidee: Der Master besitzt – per Definition – die Originale der

Daten und der Ports, der Proxy lediglich Kopien. Der Master könnte bei jedem Transfer zum Proxy *alle* Daten und Ports senden, über die der Proxy verfügen sollte.

Der Proxy könnte seinerseits im Zuge der Assimilation dieser Dateneinheiten zunächst *alle Daten und Ports* des Masters löschen, um danach die neuen einzuspielen. Somit ist sichergestellt, dass immer die aktuellen Werte beim Proxy vorliegen.

Dieses Vorgehen ist für konkrete Implementierungen sicherlich ungeeignet, die mit dem größtmöglichen Datenaustausch verbunden sind. Es steht realen Implementierungen frei, performante Lösungen zu finden und zu nutzen, die den gleichen Zweck erfüllen.

3.2.2 Peer-Gruppen

Nahezu alle Arbeiten finden heutzutage in Arbeitsgruppen statt. In einer Gruppe arbeiten die Mitglieder arbeitsteilig an der Erreichung von Zielen der Gruppe. Üblicherweise hat ein Mitglied auch noch individuelle Interessen, aber in seiner Funktion als Gruppenmitglied gibt es (freiwillig) Teile seiner Autonomie ab und agiert als Teil eines Ganzen.

Im ACP-Modell lassen sich einige Aspekte der Gruppenarbeit modellieren. So wird es in jeder Gruppe Experten geben, die zu ausgewählten Themen Daten erstellen und diese an die Gruppenmitglieder verteilen. Sie werden auch Daten zu den Themen für die Gruppe bewerten, wenn sie irgendwie in die Gruppe gelangen. Sie werden außerdem teilweise für die Gruppe Regeln festlegen, zu welchen Themen was zu sagen ist und welche Themen von Interesse für die Gruppe sind. Diese Regeln sollten idealerweise alle Gruppenmitglieder übernehmen.

In jeder realen Gruppe werden mehrere solcher Experten existieren. In der Realität wird es auch einen Abstimmungsprozess darüber geben, welche Themen relevant sind und welche nicht und es werden (hoffentlich) eher selten autokratische Entscheidungen Einzelner erfolgen. Die Abstimmungsprozeduren können und sollen im Modell aber nicht dargestellt werden. Es soll nur der Zusammenhang der Experten mit der Gruppe und von Gruppenmitgliedern mit der Gruppe modelliert werden.

Eine solche Gruppenbildung lässt sich mit dem vorliegenden Modell durch die Nutzung der Master-Proxy-Beziehung modellieren. Dazu wird ein *Meta-Peer* eingeführt, das für die Gruppe steht, die eine Menge von Peers bilden wollen. Das Meta-Peer agiert einmal in der Rolle des Masters gegenüber allen diesen Peers und agiert gleichzeitig als Proxy für einige ausgezeichnete Peers in der Gruppe, die spezielle Arbeitsaufgaben übernehmen. Abbildung 3.4 veranschaulicht diesen Zusammenhang und wird im Folgenden ausführlich erklärt.

Die Grundidee des Modells erschließt sich aus folgenden Überlegungen. Eine Arbeitsteilung³ innerhalb einer Gruppe kann dadurch erfolgen, dass Daten, die in einer Gruppe vorliegen bzw. die von der Gruppe entgegengenommen werden, nur von einigen ausgewählten Peers verarbeitet werden und im Gegenzuge der Gruppe wieder zur Verfügung gestellt werden. Weiterhin wird davon ausgegangen, dass prinzipiell jedes Peer der Gruppe in der Lage sein sollte, zu Arbeitsgebieten der Gruppe Aussagen zu treffen und zu Themen, die relevant für die Gruppe sind, Daten entgegenzunehmen.

³bezüglich Datenverarbeitung und -weitergabe, denn auf mehr konzentriert sich das Modell nicht

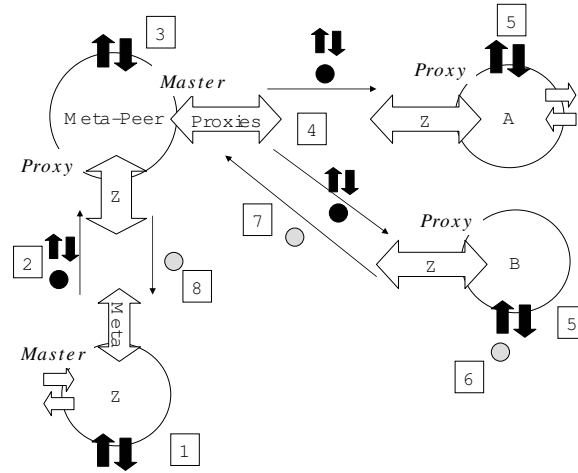


Abbildung 3.4: Ein Meta-Peer

Eine mögliche Modellierung ist in Abbildung 3.4 dargestellt. Die Gruppe wird durch ein zusätzliches Peer, ein Meta-Peer, repräsentiert. Jedes Gruppenmitglied ist Proxy des Meta-Peers und empfängt damit alle gruppenrelevanten Daten und Ports. Einem Proxy steht es dabei frei, nur Teile der empfangenen Daten und Ports anzunehmen. Das ist in der rechten Seite der Abbildung dargestellt (Peers A und B). Das Meta-Peer ist gleichzeitig Proxy - in der Abbildung nur eines - ausgezeichneten Peers (Z), das Daten und Ports für die Gruppe definiert und die eingehenden Daten verarbeitet, siehe linke Seite der Abbildung. Es soll detaillierter auf den Ablauf der Erstellung und den Fluss von Ports und Daten in der Gruppe eingegangen werden.

Zunächst soll sich nur auf die linke Seite der Abbildung konzentriert werden. Z hat zwei Ports (symbolisiert durch dunkle Pfeile) definiert, eine IP und einen OP, die für die gesamte Gruppe gelten sollen. Z hat auch Daten für die Gruppe erzeugt, die in der Abbildung nicht zu sehen sind.

In Schritt 2 überträgt Z beide Ports und die Daten an das Meta-Peer, das in dieser Beziehung als Proxy agiert. Schritt 3 zeigt, dass das Meta-Peer die Ports übernommen hat. Das Gleiche gilt für die Daten, was nicht explizit dargestellt wurde. Gegenüber den Peers A und B agiert das Meta-Peer als Master. Im Schritt 4 überträgt es Daten und Ports an die beiden Peers, die sie ihrerseits aufnehmen, Schritt 5.

In Schritt 6 ist beispielhaft zu sehen, dass das Peer B stellvertretend für die Gruppe ein Datum aufgenommen hat. Wie jedes Proxy speichert es die Daten inklusive des Umgebungskontextes. Beides überträgt es weiter an das Meta-Peer, Schritt 7. Das Meta-Peer agiert nun seinerseits als Proxy und speichert die empfangenen Daten, bis es sie in

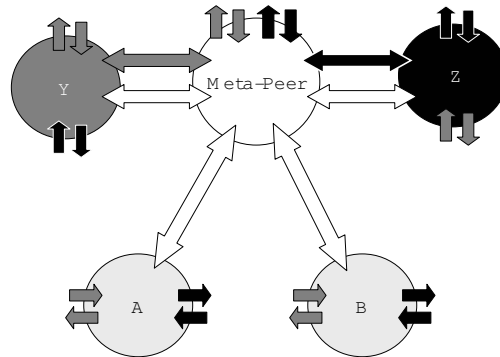


Abbildung 3.5: Peer-Gruppe mit vier Mitgliedern und zwei Mastern

Schritt 8 an Z überträgt, das sie assimilieren kann.

Die Abbildung zeigt nur ein sehr einfaches Beispiel. Insbesondere gibt es nur ein Peer, dass die Ports der Gruppe definiert und Daten, die über diese Ports eingehen, assimiliert. Die anderen Mitglieder der Gruppe (A und B) haben keine Aufgabe bezüglich des Datenaustausches der Gruppe mit der Umgebung. Sie agieren nur als Proxies der Gruppe.

Angewandt auf eine Arbeitsgruppe würde das heißen, dass lediglich ein Mitglied bestimmt, welche Art von Daten generell von Interesse für die Gruppe sind. Mitglieder der Gruppe würden passende Daten entgegennehmen, sie aber nicht verarbeiten, sondern an das eine Mitglied weitergeben. In realen Gruppen wird eine deutlich differenziertere Arbeitsteilung üblich sein.

Diese Konstruktion ist aber auch der Grund, warum Z als Gruppenmitglied nicht wie jedes andere Mitglied der Gruppe auch Proxy des Meta-Peers ist. Das muss es nicht sein, denn als alleiniger Master des Meta-Peers definiert es alle neue Daten und Ports und muss diese nicht beziehen.

In Abbildung 3.5 ist eine komplexere Gruppenstruktur dargestellt und es wird eine weitere grafische Vereinfachung eingeführt. Die dargestellte Gruppe besteht aus vier Mitgliedern (A, B, X und Y), wobei X und Y für die Gruppe Daten bzw. Ports bereitstellen und eingehende Daten bearbeiten. In der Sprechweise des ACP-Modells existieren in der Gruppe zwei Master-Peers.

In der Abbildung wird farblich angedeutet, dass Y für ein Themengebiet der Grup-

pe zuständig ist und Z für ein anderes.⁴ In der Abbildung wird eine weitere grafische Vereinfachung benutzt. Die beiden Doppelpfeile, die bisher Master und Proxy separat definierten, wurden zu einem zusammengefasst, der direkt Master- und Proxy-Peer verbindet. Er trägt die Farbe des Masters. In der Abbildung ist zu sehen, dass Y Master gegenüber der Gruppe ist, ebenso wie Z. Alle vier Gruppenmitglieder sind aber auch Proxy der Gruppe, was durch die weißen Doppelpfeile dargestellt ist.

Aus der Darstellung lassen sich komprimiert eine Reihe von Informationen herauslesen. Das Meta-Peer repräsentiert die Gruppe. Die Ports beschreiben die Regeln, die den Austausch von Daten der Gruppenmitglieder mit anderen Peers regeln. Anhand der Farben der Ports lässt sich ermitteln, welches Gruppenmitglied arbeitsteilig für die Bearbeitung der zu sendenden bzw. der empfangenden Daten zuständig ist. Die Anzahl der Gruppenmitglieder lässt sich einfach dadurch ermitteln, indem gezählt wird, wie oft das Meta-Peer die Funktion des Masters innehat. Das ist in der Abbildung viermal der Fall (weiße Doppelpfeile)

Sender und Empfänger in Gruppen-Ports

Jeder Port enthält eine Bezeichnung für den Empfänger bzw. Sender von Dateneinheiten. Das Modell legt nicht fest, welche Bezeichnung zu wählen ist.

Bei der Modellierung einer Gruppe kann es aber sinnvoll sein, dass die Ports einer Gruppe auch die Bezeichnung der Gruppe als Sender bzw. Empfänger tragen und nicht die Bezeichnung eines einzelnen Gruppenmitgliedes.

Wird dieses Vorgehen gewählt und ein Peer „begegnet“ z. B. dem Peer B aus der Abbildung, so kann es potenziell folgende Daten beobachten, wenn diese offen gelegt sind. Es erkennt die Bezeichnung von B und jeweils zwei IPs und zwei OPs, die jeweils verschiedene Erwartungen für einen Datenaustausch definieren und als Sender- bzw. Empfängerbezeichnung die Bezeichnung der Gruppe tragen und nicht von B. Anhand dieser Beobachtungen kann das Peer nunmehr entscheiden, ob es mit B in seiner Funktion als Gruppenmitglied Daten austauschen will oder nicht.

Hervorzuheben ist der Punkt, dass potenziell erkennbar ist, dass nicht *direkt mit der Gruppe* Daten ausgetauscht werden, sondern mit einem *Mitglied der Gruppe*. Es ist aber nicht erkennbar, ob B die Daten selber bearbeitet oder weiterleitet.

Gruppen von Peer-Gruppen

Ein Meta-Peer kann wie jedes andere Peer Mitglied einer Gruppe sein. Andere Gruppenmitglieder müssen dabei nicht zwingend auch Meta-Peers sein. Damit lassen sich mehrstufige hierarchische Beziehungen von autonomen Peers und Peer-Gruppen modellieren.

⁴Zur Veranschaulichung kann man sich vorstellen, dass X ausgezeichnete Expertin im Bereich Knowledge Management ist und für die Gruppe definiert, welche Unterlagen herausgegeben werden sollen und die eingehenden Informationen bearbeitet. Y kann ein Experte für Mobilfunk und in der Gruppe verantwortlich für die Bearbeitung entsprechender Unterlagen sein. A und B sind Mitarbeiter, die keine spezielle Funktion innehaben, die die Außenkommunikation der Gruppe betrifft.

Es ist nicht erlaubt, dass sich Meta-Peers (transitiv) selbst als Mitglied enthalten. Das widerspricht der grundlegenden Intention der Modellierung von Gruppen mit dem ACP-Modell.

3.2.3 Synchronisation und Client-Server

Es ließen sich noch weitere Beziehungen durch einen entsprechenden Austausch von Ports und Daten definieren, wie z. B. eine Synchronisationsbeziehung zwischen Peers. In dem Fall müssten Dateneinheiten, die ein Peer annimmt, sowohl assimiliert als auch allen anderen Peers übermittelt werden, die synchron gehalten werden wollen.

Eine Client-Server-Beziehung kann als Sonderfall eines P2P-Austausches angesehen werden. Verfügt ein Peer nur über OPs, so kann es als Server interpretiert werden, verfügt es nur über IPs als Client.

3.3 Zusammenfassung

Das ACP-Modell führt auf einer sehr abstrakten Ebene die Konzepte autonome Peers, Kontext und kontextbasierter Datenaustausch ein. Abgeleitet von diesen Konzepten konnte bereits gezeigt werden, was unter kontextbasierten Datenflüssen verstanden wird. Es wurde außerdem gezeigt, wie spezielle Peer-Beziehungen bereits auf dieser abstrakten Ebene modelliert werden können. Gruppen von Peers wurden ausführlich besprochen, da sie im Folgenden eine zentrale Rolle spielen werden. Synchronisations- und Client-Server-Beziehungen wurden nur angedeutet. Letztere wurden als Sonderfall einer P2P-Beziehung identifiziert. Sie spielen im Folgenden keine besondere Rolle und wurden deshalb nicht weiter ausgeführt.

Im folgenden Kapitel wird dieses Modell als Basis einer Referenzarchitektur für ein P2P KMS benutzt werden. Theoretisch hätte man als Basis auch ein Agentensystem einsetzen können. Es war in diesem Kapitel aber zu sehen, dass das ACP-Modell deutlich weniger komplex ist, als ein Agentensystem. Das ist nur logisch, wenn sich die höheren Anforderung an Softwareagenten vor Augen gehalten werden.

Daraus folgt auch, dass die Anforderungen an Hard- und Software für die Implementierung eines auf ACP basierenden Systems deutlich geringer sind, als die Implementierung die auf einem Agentensystem beruht. Das ist positiv vor dem Hintergrund, dass das in dieser Arbeit zu entwickelnde System auch auf mobilen Geräten mit limitierten Ressourcen, wie Handys und PDAs, laufen soll.

4 Shared Knowledge (Shark)

In Abschnitt 2.5 wurde ein Ansatz vorgestellt, wie Wissensflüsse in Unternehmen auf P2P-Austausche von Wissen zwischen Individuen zurückgeführt werden können. Dieser Ansatz soll Basis für die Spezialisierung des ACP-Modells zu einer Referenzarchitektur eines P2P-Knowledge Management Systems sein, die den Namen *Shark* trägt. Shark ist ein Akronym und steht für *Shared Knowledge*.

Eine Referenzarchitektur beschreibt, laut *Open Distributed Processing (ODP)*, die grundlegenden Komponenten einer Architektur, deren wichtigsten Kommunikationsbeziehungen und Informationsflüsse sowie wesentliche Softwarekomponenten. Sie lässt konkrete technische Ausprägungen der einzelnen Komponenten und deren Verteilung offen [Put01]. Insbesondere spezifiziert Shark die Wahl der Wissensrepräsentationssprache und der konkreten Protokolle zum Wissensaustausch nicht. Diese Verfeinerungen fallen in die Beschreibung einer Architektur.

Warum soll überhaupt eine Referenzarchitektur vorgestellt werden? Tatsächlich begannen die Arbeiten an Shark mit der Erstellung einer konkreten Architektur und einer Implementierung, siehe [SG02, SG03]. Es stellte sich aber heraus, dass die Prinzipien von Shark von konkret einsetzbaren Technologien abstrahiert werden können und eine allgemeine Architektur für die Erstellung von P2P-KMS möglich wird. Dieses Vorgehen hat zwei Vorteile: Einmal besteht nicht die Gefahr, dass in der Beschreibung des Systems durch technische Details von den eigentlichen Prinzipien des Systems abgelenkt wird. Zweitens kann eine Referenzarchitektur prinzipiell mit unterschiedlichsten Technologien realisiert werden. Das gilt auch für Shark. Nach diesem Kapitel wird eine mögliche Architektur für Shark vorgestellt, es wird aber nicht behauptet, dass das die *beste* sei. Es wird später gezeigt werden, dass sie sich für einige Anwendungsgebiete aber sehr gut eignet.

Nach ODP wird eine Referenzarchitektur beschrieben durch drei Viewpoints: *Enterprise*, *Informational* und *Computational Viewpoint*. Der erste spezifiziert die Objekte aus der Sicht der Geschäftsprozesse und deren Interaktionen. Der Informational Viewpoint beschreibt das System mit Hinblick auf die Informationen, die in den einzelnen Komponenten vorliegen und zwischen ihnen ausgetauscht werden. Der Computational Viewpoint benennt Software-Objekte und deren Interaktionen, ohne allerdings auf Details der genutzten Technologien und der Verteilungsaspekte einzugehen.

Zu einer Architektur gehören die Beschreibung der eingesetzten Technologien (*Technical Viewpoint*) und Verteilungsaspekt der Softwarekomponenten, Kommunikation zwischen diesen usw. (*Engineering Viewpoint*).

In diesem Kapitel werden die ersten drei Viewpoints beschrieben. Das Kapitel ist wie folgt aufgebaut. Zunächst werden die Basiskonzepte von Shark in Abschnitt 4.1 vorgestellt.

4 Shared Knowledge (Shark)

Im Anschluss wird in Abschnitt 4.2 das Verständnis von Kontext in Shark erklärt und definiert. Insbesondere wird die Funktion *Kontextualisierung* eingeführt, die eine zweistellige Operation zwischen Wissen und Kontext darstellt. Mittels dieser Funktion lassen sich die Funktionen f, g und h aus dem ACP-Modell spezialisieren, was eine deutlich klarere und einfachere Definition der *Knowledge Ports*, wie die Spezialisierung der ACP-Ports in Shark heißen, ermöglicht. Durch dieses Vorgehen lassen sich große Teile eines Wissensaustausches durch Formeln erfassen, was im ACP-Modell nicht gelang.

Abschnitt 4.3 beschreibt die Architektur der Shark-Engine und das Konzept der Knowledge Ports, in denen die Strukturen, die in den vorherigen Abschnitten beschrieben wurden, gespeichert werden und die dazu genutzt werden, einen P2P-Wissensaustausch zu steuern.

Der anschließende Abschnitt 4.4 legt Regeln fest, wie in Shark Bezeichnungen für Eigentümer, Sender und Empfänger von Wissen gesetzt und verändert werden dürfen.

Der abschließende Abschnitt 4.5 zeigt, wie sich die Policies auf die Dynamik des Wissensaustausches und der Wissensflüsse in Netzen von Knowledge Nodes auswirken.

Die einzelnen Kapitel halten sich nicht an Formalien des ODP-Referenzmodells. Sie orientieren sich aber am Modell. So werden im Abschnitt 4.1 die Objekte aus dem Enterprise Viewpoint diskutiert und teilweise bereits der Informationale Viewpoint. Der Abschnitt 4.2 untersucht nur Strukturen dieses Viewpoint. Die Abschnitte 4.3 und 4.4 stellen das System im Wesentlichen vom Computational Viewpoint her dar.

4.1 Basiskonzepte

In diesem Abschnitt werden die Basiskonzepte von Shark aus dem Enterprise Viewpoint her vorgestellt. Sie bilden eine Brücke zwischen den Untersuchungen des Kapitels ?? und Shark und stellen gleichzeitig eine Spezialisierung des ACP-Modells dar.

4.1.1 Shark Peers

Die Spezialisierung der Shark-Peers wurde hinreichend motiviert. Sie sind eine Spezialisierung der ACP-Peers, mit dem wesentlichen Unterschied, dass sie Wissen anstelle allgemeiner Daten verwalten.

Definition 1 (Shark Peers) *modellieren natürliche Personen. Ein Shark-Peer hat folgende Eigenschaften:*

Es verfügt über strukturiertes Wissen, kann Eigenschaften der Umgebung wahrnehmen, eine Bezeichnung und Wissen mit anderen Peers austauschen. Der Austausch von Wissen erfolgt anhand von Outputstrategien und Relevanzfiltern, die sich als Ports im Sinne des ACP-Modells interpretieren lassen.

4.1.2 Shark Knowledge Nodes

Individuen schließen sich zu sozialen Systemen zusammen, um komplexe Aufgaben zu erfüllen. Das gilt insbesondere für wissensintensive Prozesse. Es war zu sehen, dass Wis-

sen in Unternehmen auf der ontologischen Dimension wandert. Shark-Peers modellieren allerdings nur eine Position dieser Dimension, das Individuum.

Das Konzept der ACP-Meta-Peers (Abschnitt 3.2.2) erlaubte es, Peer Gruppen und deren Arbeitsteilung zu modellieren. Dieser Ansatz kann auch für Shark-Peers genutzt werden. Das Konzept der Meta-Peers soll anhand eines Beispiels und für Shark-Peers kurz wiederholt werden, siehe Abbildung 4.1. Die Shark-Peers sind als Strichfiguren in einem Kreis dargestellt, die Doppelpfeile symbolisieren die Kombination von Outputstrategie und Relevanzfilter.

Es werden drei Shark-Peers dargestellt (A, B und C). Wie auch im ACP-Modell werden Doppelpfeile benutzt, um eine Kombination von IP/OP zu symbolisieren. Die weißen Doppelpfeile mögen die privaten Ports der Peers symbolisieren.

A und B sind Mitglieder der Gruppe D. Das wird – analog zum ACP-Modell – durch die dunkleren Doppelpfeile symbolisiert, die Master und Proxy verbinden und die Farbe des Masters in dieser Beziehung tragen. B ist zusätzlich Mitglied der Gruppe E. Die Gruppen D und E und das Individuum G sind Mitglieder der Gruppe F.

Die Gruppen sind jeweils mit gestrichelten Linien gezeichnet, um anzudeuten, dass sie nur Meta-Peers sind, also Repräsentanten einer Gruppe von Peers. In F ist auch angedeutet, dass es Gruppen als Mitglied hat. Man kann sich ein Unternehmen vorstellen, in dem die Kollegen A, B und C arbeiten.

In der Abbildung ist damit eine zweistufige Hierarchie an Shark-Peer-Gruppen dargestellt. Wie bereits schon in einer ACP-Peer-Gruppe übernehmen die Gruppenmitglieder die Ports der Meta-Peers, die die Gruppe repräsentieren. Daher übernehmen D, E und G die Ports der Gruppe F (beispielhaft durch die schwarze IP/OP-Kombination angedeutet). D und E sind ihrerseits Gruppen und verfügen über zusätzliche Ports (dunkle IP/OP-Kombination).

Am Ende der Hierarchie stehen die Individuen. A verfügt wegen seiner Mitgliedschaft in D über deren Ports und zusätzlich über die Ports der übergeordneten Gruppe F. Hinzu kommen die eigenen Ports. B hat eine weitere Quelle von Ports, da es in einer weiteren Gruppe E Mitglied ist. C ist direkt Mitglied in F und hat damit nur eine zusätzliche Quelle von Ports, neben sich selbst.

In der Darstellung ist der Übersicht wegen auf zwei Dinge verzichtet worden: Einmal ist kein Wissen eingezeichnet, das mit OPs von der Gruppe zu den Gruppenmitgliedern übertragen werden muss. Außerdem spielen alle Individuen lediglich die Rolle von Proxies in den Gruppen. Eine Peer-Gruppe benötigt aber wenigstens ein Peer, das Ports und Wissen der Gruppe definiert und als Master gegenüber der Gruppe agiert.

Diese Eigenschaft soll noch einmal unterstrichen werden. Bereits im ACP-Modell wurde die Arbeitsteilung in Gruppen so modelliert, dass Meta-Peers Gruppen *repräsentieren*, während die tatsächliche Arbeit ausschließlich Peers ausführen, die dann als Master-Peers gegenüber der Gruppe agieren. Gruppen sind sowohl in ACP als auch in Shark Mediatoren. Im Knowledge Node Modell von Bonifacio sind aber Gruppen eigenständig, genauso wie Individuen. Das erzeugte einige der genannten Inkonsistenzen.

Das Shark-Peer A möge noch einmal genauer betrachtet werden. Es verfügt über

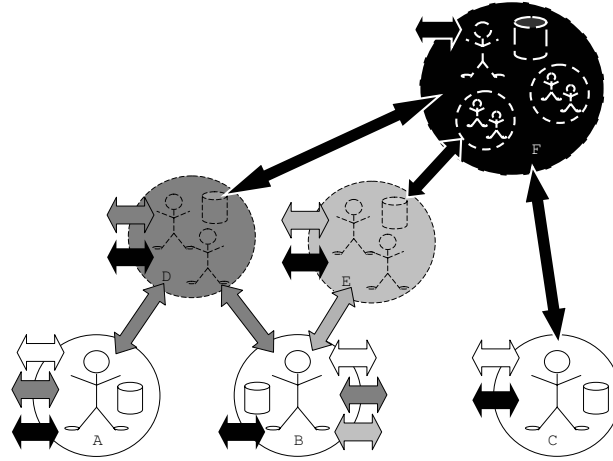


Abbildung 4.1: Hierarchische Gruppen von Shark-Peers

drei IP/OP-Kombinationen¹. A verfügt demnach über den eigenen Relevanzfilter und Outputstrategie, die seiner Gruppe D und deren übergeordneten Gruppe F. Damit ist es ihm möglich, in allen drei Rollen zu agieren. Das ist exakt die Situation, wie sie in Abschnitt 2.5 und konkret in Abbildung 2.15 beschrieben wurde.

Das heißt, dass Peer-Gruppen korrekt die Ergebnisse der Analyse der P2P-Wissens-austausche modellieren und gleichzeitig auf einem Peer-Modell basieren. Peer-Gruppen sind demnach ein geeignetes Mittel, um soziale Gruppen und Gruppen von Gruppen und damit die Positionen auf der ontologischen Dimension außer der des Individuums zu modellieren.

Das soll in einer Definition festgehalten werden.

Definition 2 (Shark Meta-Peers) *sind eine Spezialisierung der ACP-Meta-Peers. An die Stelle der ACP-Peer treten Shark-Peers. Shark Meta-Peers modellieren gleichzeitig soziale Gruppen und Gruppen solcher Gruppen.*

Es ist ein unschöner Zustand, dass die ontologische Dimension durch zwei Konzepte modelliert wird. Daher soll eine Abstraktion stattfinden. Dazu soll sich der Bezeichnung aus Bonifacios Modell bedient werden. Der Grund ist, dass die Bezeichnung als passend empfunden wird, auch wenn das Modell an einigen Stellen nicht konsistent ist. Shark schließt diese Lücken. Es könnte als eine Verbesserung des Ansatzes von Bonifacio und den anderen angesehen werden. Es soll also definiert werden:

Definition 3 (Knowledge Nodes) *sind eine Verallgemeinerung von Shark Peers und*

¹was natürlich symbolisch gemeint ist. Jede Gruppe kann seinen Mitgliedern beliebig viele Ports, also auch gar keine, anbieten.

Shark Meta-Peers. Sie modellieren eine beliebige Position auf der ontologischen Dimension des Spiralmodells von Nonaka und Takeuchi.

Damit sind die Konzepte von Shark aus Sicht des Enterprise Viewpoints beschrieben.

4.1.3 Wissen und Interessen

Das Verständnis von Wissen wurde im letzten Kapitel hinreichend erläutert. Es soll für Shark die Deutung aus dem Knowledge Management übernommen werden.

Definition 4 (Wissen) *ist etwas, das einen Menschen zu einer qualifizierten Handlung befähigt.*

Zu Diskussionen um Wissen sei vor allem auf den Abschnitt 2.4 verwiesen, in dem auch die Unterscheidung zwischen implizitem und explizitem Wissen im Detail präsentiert wurde.

Über Wissen verfügen bedeutet nicht gleichzeitig, dass dieses Wissen auch ausgetauscht werden soll. Diese Problematik wurde in Abschnitt 2.4.4 und auch in [SG02, SP02, SG03] dargelegt. In Shark wird daher ein weiteres Konzept eingeführt:

Definition 5 (Interesse) *ist potenzielle Bereitschaft einer Knowledge Node zum Austausch von bestimmtem Wissen.*

Wissen kann gesendet und empfangen werden. Daher können ein *Empfangsinteresse* und ein *Sendeinteresse* unterschieden werden. Beide sind Konkretisierungen der allgemeineren Erwartungskontexte bezüglich der Daten im ACP-Modell. Der Zusammenhang zwischen Wissen, Interesse und Kontext wird in Kürze dargestellt werden.

4.1.4 Wissen und Kontext

Sowohl im ACP-Modell als auch bei verwandten Arbeiten zum Thema Wissen, Kontext usw. war zu sehen, dass Kontext eine essenzielle Rolle spielt. Es war auch offensichtlich, dass es schwer bis unmöglich ist, eine allgemeine Definition von Kontext zu finden. Auch in Shark werden lediglich konkretere Kontextarten sehr konkret definiert werden können. Das allgemeine Konzept kann auch hier nur sehr allgemein definiert werden.

Definition 6 (Kontext) *ist alles, das nicht direkt zu Wissen gehört, die Ausführung von Funktionen darüber aber vereinfacht bzw. erst ermöglicht.*

Eine Eigenschaft von Kontext kann anhand der Definition festgestellt werden. Kontext ermöglicht bzw. unterstützt den Umgang mit Wissen. Diese Eigenschaft deckt sich mit der Definition von Wissen, siehe Definition 4. In Shark wird Kontext explizit als Wissen verstanden, nämlich Wissen über den Umgang mit Wissen, was man auch als Meta-Wissen bezeichnen kann. Diese Feststellung wird weiter hinten zu positiven, weil vereinfachenden Resultaten führen.

4 Shared Knowledge (Shark)

Es war zu sehen (Abschnitte 2.2 und 2.3), dass Wissen in einem Kontext erstellt wird. Wenn es externalisiert wird, kann von einem Kontext des Schreibens gesprochen werden. Wesentlich ist aber die Tatsache, dass jedwedes Wissen in irgendeinem Kontext steht. Andersherum beschreibt jede Knowledge Node, für welche Wissen sie sich interessiert, ebenfalls anhand eines Kontextes. Dieser Kontextart soll ein Name gegeben werden:

Definition 7 (Thema) *ist ein Kontext, der beschreibt, welches Wissen für eine Knowledge Node relevant ist.*

Der Unterschied zwischen Thema und Interesse ist folgender: Zu Themen speichert und strukturiert eine Knowledge Node Wissen. Erst wenn sie ein Interesse formuliert, ist sie auch bereit, Wissen dazu auszutauschen.

4.2 Kontext

In diesem Abschnitt sollen die Kontextarten für Shark spezialisiert werden und es soll gezeigt werden, welche Kontextarten generell modelliert werden sollen.

In Abschnitt 2.3 und auch in Abbildung 2.9 wurden Eingangsgrößen ermittelt, die zur Bildung eines Kommunikationskontextes führten. Es war bereits dort zu sehen, dass nicht alle für ein KMS von Relevanz sind. In Shark werden folgende Größen modelliert: Wissen, Umgebungskontext und Erwartungen an beide. Mentale Modelle, Ziele, die über eine direkte Kommunikation hinausgehen, werden nicht modelliert. Es wird davon ausgegangen, dass dieses von den Menschen, die ein System, basierend auf Shark, einsetzen, beachtet wird und nicht in Software implementiert werden muss.

Das ACP-Modell unterschied grob zwischen Umgebungs-, Daten- und Erwartungskontexten bezüglich der Ersten. Als Referenzarchitektur werden keine Details bezüglich eines Repräsentationsformates für Kontext definiert. Das erfolgt später. Der Datenkontext wurde bereits als Thema definiert. Der Umgebungskontext wird in Shark nicht weiter verfeinert. Im Folgenden werden aber die Erwartungskontexte diskutiert werden und es wird eine Funktion postuliert werden, die die Umwandlung von Kontexten ineinander erlaubt bzw. eine Erzeugung von kontextualisiertem Wissen ermöglicht.

4.2.1 Erwartungskontexte in Shark

Das ACP-Modell definiert drei Erwartungskontexte: Erwartung bezüglich der Umgebung (X_{EU}), Erwartung bezüglich der zu sendenden Daten (X_{SD}) und Erwartung bezüglich der zu empfangenden Daten (X_{ED}).

Die Erwartung gegenüber der Umgebung wird in Shark nicht spezialisiert. Die Erwartung gegenüber Daten wird in Shark als ein Interesse bezeichnet, siehe Definition 5. Tatsächlich drücken die Erwartungskontexte gegenüber den Daten im ACP-Modell aus, zu welchem Kontext die Daten passen sollen, zu denen ein Peer bereit ist, Daten zu senden bzw. zu empfangen. In Shark wird daher spezialisierend von einem *Sendeinteresse* (I_S) gesprochen bzw. von einem *Empfangsinteresse* (I_E). Im Folgenden werden die Interessen, deren Beziehung untereinander und Wissen detailliert beschrieben.

4.2.2 Kontextualisierung

Am ACP-Modell wurde allgemein davon gesprochen, dass Erwartungskontexte formuliert werden, die durch Funktionen aufeinander angewandt werden können. Dieses Vorgehen wird in Shark deutlich konkretisiert. Das gelingt durch die Einführung der *Kontextualisierung*:

Definition 8 (Kontextualisierung) *ist eine zweistellige Funktion, die einen Kontext (einschränkender Kontext oder einfach Kontext) auf Wissen (Basiswissen) anwendet. Als Resultat entsteht eine Menge kontextualisierten Wissens (Ergebniswissen), das nur das Wissen enthält, das zum einschränkenden Kontext passt.*

Da Kontext spezielles Wissen ist, können auch Kontexte aufeinander angewandt werden. Die Kontextualisierung hat folgende Eigenschaften:

1. *Ein leerer einschränkender Kontext ändert nichts am Wissen, da er nichts einschränkt ($K * \emptyset = K$). In der Terminologie der Algebra würde man von einem neutralen Element sprechen.*
2. *Aus leerem Wissen entsteht nur leeres Wissen ($\emptyset * X = \emptyset$).*
3. *Die Kontextualisierung fügt kein neues Wissen hinzu. Das Ergebniswissen ist immer eine (unechte) Teilmenge des Basiswissens ($K_B * X = K_E; K_E \subset K_B$).*

Diese Definition soll zunächst als Postulat angesehen werden, dass eine solche Funktion existiert, ohne dass Bezug auf eine bestimmte Form der Wissensrepräsentation gemacht wird und ohne dass gezeigt wird, dass eine solche Funktion existieren kann. Ein eigenes Kapitel (5) beschäftigt sich mit einer möglichen Realisierung der Kontextualisierung mit Topic Maps, einer Sprache der Wissensrepräsentation. Das Vorgehen ist auch bei der Definition komplexer Software nicht unüblich, wenn *abstrakte Funktionen* definiert werden, die gerade in einer Referenzarchitektur als ein probates Mittel für die Definition von Funktionen verstanden werden, deren Eigenschaften bekannt, aber deren Implementierung unbekannt ist [GHJV95].

Ein Definition ist nicht beweisbar, sie soll aber sinnvoll und zweckmäßig sein. Die Zweckmäßigkeit wird sich zeigen, wenn sie in der folgenden Darstellung angewendet wird. Die Sinnfälligkeit der einzelnen Eigenschaften der Kontextualisierung soll kurz motiviert werden.

1. Die Kontextualisierung wendet Kontext als eine *Einschränkung* auf Wissen an. Wenn keine Einschränkung getroffen wird, so soll auch das Wissen nicht eingeschränkt werden. Ein leerer Kontext kann als Kontextfreiheit interpretiert werden.
2. Leerem Wissen kann eine Einschränkung nichts hinzufügen. Diese Eigenschaft erscheint logisch und in sich selbst sinnvoll.
3. Da ein Kontext als *Einschränkung* interpretiert wird und nichts hinzufügt, kann das kontextualisierte Wissen höchstens unverändert bleiben, nämlich dann, wenn es vollständig zum beschriebenen Kontext *passt*. Im anderen Fall wird es weniger.

4 Shared Knowledge (Shark)

Im Folgenden wird der Stern als Operationszeichen für die Kontextualisierung benutzt:

$$\text{Basiswissen} * \text{Kontext} = \text{Ergebniswissen}$$

Man sagt, dass der Kontext auf das Basiswissen *kontextualisierend* angewandt wird. Zwei konkrete Formen der Kontextualisierung sollen definiert werden.

Definition 9 (Lokalisierung) *ist eine spezielle Form der Kontextualisierung, in der der einschränkende Kontext eine Spezifikation eines Ortes ist.*

Definition 10 (Personalisierung) *ist eine spezielle Form der Kontextualisierung, bei der Themen einer Knowledge Node auf Wissen angewendet werden. Das Ergebnisswissen ist dann personalisiertes Wissen.*

Eine weitere Definition wird im Folgenden hilfreich sein, wenn komplexere Operationen mit Wissen und Kontexten erfolgen sollen:

Definition 11 (Unabhängige Kontextarten) *Zwei Kontextarten heißen unabhängige Kontextarten, wenn die Reihenfolge ihrer Anwendung auf einen Basiskontext keinen Einfluss auf das Resultat hat:*

Formaler ausgedrückt sind X_1 und X_2 unabhängige Kontextarten, genau dann, wenn bei einer Anwendung auf das Basiswissen K_B gilt:

$$(K_B * X_1) * X_2 = (K_B * X_2) * X_1$$

4.2.3 Spezialisierung der ACP-Funktionen f,g und h

Mittels der Kontextualisierung werden die Funktionen f , g und h des ACP-Modells spezialisiert.

f Die Funktion ermittelt, ob der Umgebungskontext (X_U) und die Erwartung dazu (X_{EU}) zueinander passen. f wird in Shark durch die Kontextualisierung spezialisiert. Dabei ist es irrelevant, ob X_U als Einschränkung über X_{EU} interpretiert wird, oder umgekehrt. Wichtig ist, dass irgendeine Übereinstimmung existiert. In Shark wird f wie folgt definiert:

$$f(X_{EU}, X_U) = \left\{ \begin{array}{l} \text{wahr} \Leftrightarrow X_{EU} * X_U \neq \emptyset \\ \text{falsch} \Leftrightarrow X_{EU} * X_U = \emptyset \end{array} \right\}$$

g Die Funktion g ermittelt im ACP-Modell einen effektiven Datenkontext, ausgehend von einem Erwartungskontext gegenüber Daten. In Shark wurden diese zu Interessen (I_S oder I_E) spezialisiert und werden wie folgt interpretiert. Ein Interesse drückt einen Wunsch nach Austausch von Wissen zu Themen aus. Teil eines Interesses sind damit Themen, aber auch Erwartungen an die Umgebung. Ein Interesse

enthält damit eine Reihe von möglichen Umgebungskontexten. Ein *effektives* Interesse wird in Shark als ein Interesse interpretiert, das in einer konkreten Umgebung gilt. Ein effektives Interesse wird damit aus einem Interesse erzeugt, indem der Umgebungskontext darauf einschränkend, d. h. *kontextualisierend*, einwirkt. Die Funktion g wird in Shark daher wie folgt definiert:

$$g(I_{[S|E]}, X_U) = I_{[S|E]} * X_U = I_{E[S|E]}$$

Das Resultat heißt *effektives Sende- bzw. Empfangsinteresse* (I_{ES} bzw. I_{EE}).

- h** Die Funktion h ermittelt im ACP-Modell anhand effektiver Datenkontexte, welche Daten tatsächlich der Assimilation übergeben werden sollen. In Shark verknüpft h damit ein effektives Empfangsinteresse mit einem effektiven Sendeinteresse. Das Resultat soll ein übereinstimmendes Interesse zwischen Sender und Empfänger beschreiben, anhand dessen der Empfänger Wissen assimilieren kann. In Shark wird das effektive Sendeinteresse als ein Angebot an den potenziellen Empfänger verstanden. Der Sender hat sich festgelegt, zu den genannten Themen im konkret beobachteten Umgebungskontext Wissen zu senden. Der potenzielle Empfänger hat sich mit dem effektiven Empfangsinteresse ebenfalls festgelegt. Nun müssen beide verknüpft werden. In Shark wird definiert, dass das Empfangsinteresse als Einschränkung über dem Sendeinteresse interpretiert wird:

$$h(I_{ES}, I_{EE}) = I_{ES} * I_{EE} = I_X$$

Das Resultat heißt *Austauschinteresse* (I_X).

Durch die Spezialisierung kann auch der Inhalt der Dateneinheit spezieller interpretiert werden. Es wird dort Wissen passend zum effektiven Sendeinteresse übertragen. Es soll als *gesendetes Wissen* (K_S) bezeichnet werden. Auf der Empfängerseite lässt sich ebenfalls definieren, welches Wissen der Assimilation übergeben wird. Es ist genau das Wissen, das entsteht, wenn das Austauschinteresse kontextualisierend auf das gesendete Wissen angewendet wird. Das Resultat möge *Austauschwissen* (K_X) heißen:

$$K_X = K_S * I_X$$

Abbildung 4.2 stellt alle Formeln im Zusammenhang mit dem Wissensaustausch zwischen zwei Shark-Peers dar.

Bis auf die Assimilation und Extraktion, die im Folgenden diskutiert werden, sind damit bereits alle Funktionen und Werte des Datenaustausches im ACP-Modell für den Wissensaustausch in Shark spezialisiert worden, siehe Abbildung 4.2.

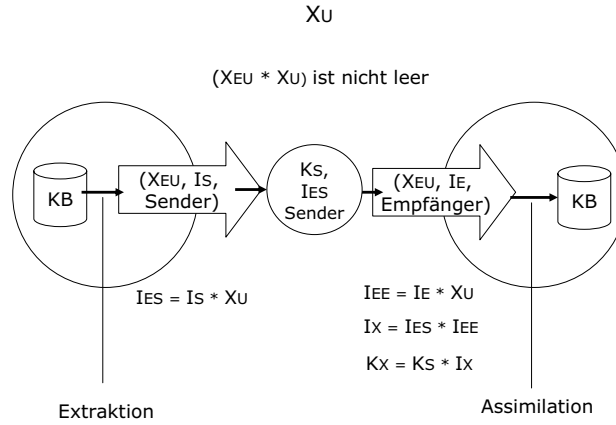


Abbildung 4.2: P2P Wissensaustausch in Shark

4.2.4 Extraktion von Wissen

Die Extraktion ist der Prozess, der Daten aus der Datenbasis des Peers entnimmt und zur Übertragung bereitstellt. Die Extraktion wird in Shark spezialisiert. Es wird davon ausgegangen, dass anhand der Empfängerbezeichnung (wenn unbekannt, dann wird das anonyme Peer als Empfänger benutzt) und des effektiven Sendeinteresses Wissen aus der Wissensbasis entnommen wird. Im Ergebnis wird eine Dateneinheit erstellt, die das Sendewissen enthält, das effektive Sendeinteresse und die Senderbezeichnung aus den OKP.

$$extract : (KB, I_{ES}, Empfaenger) \rightarrow (K_S, I_{ES}, Sender)$$

Die Extraktion wird in diesem Referenzmodell nicht näher spezifiziert. Bei realen Implementierungen wird das ein Zugang zu einer Wissen- oder Datenbasis oder einem CMS sein, der u. a. Zugriffsrechte überprüft. Im Kapitel 5 wird eine minimale Realisierung dieser Funktion über einer Topic Map Engine beschrieben.

4.2.5 Assimilation von Wissen

Die Assimilation ist der Prozess, der empfangenes Wissen in die Wissensbasis eines Peers einfügt. In Shark ist die Funktion wie folgt beschrieben.

$$assimilate : (KB, K_X, I_X, Sender) \rightarrow KB'$$

Für die Assimilation gilt das Gleiche wie für die Extraktion. Es erfolgt keine weitere Spezialisierung der Funktion, sie kann ebenfalls als abstrakte Funktion verstanden werden, die in einer Implementierung durch einen Zugriff auf eine Wissens- oder Datenbasis oder ein CMS ersetzt wird.

Auch für die Assimilation wird in Kapitel 5 eine minimale Implementierung vorgestellt werden.

Damit sind die Konzepte von Shark aus dem Informationale Viewpoint beschrieben. Es wurde insbesondere das Verständnis von Wissen und Kontext definiert. Es wurden einige Kontextarten beschrieben. Mit der Kontextualisierung wurde eine Funktion postuliert, mit deren Hilfe Wissen und Kontexte anhand eines Kontextes eingeschränkt werden können. Dadurch gelang insbesondere eine Spezialisierung der Funktionen f , g und h des ACP-Modells. Der Rest des Kapitels beschäftigt sich mit Konzepten von Shark aus dem Computational Viewpoint heraus.

4.3 Shark-Engine

Bisher wurde Shark vor allem aus dem Enterprise Viewpoint und dem Informationale Viewpoint heraus betrachtet. Nun soll ein Blick aus dem verbleibenden, dem Computational Viewpoint auf Shark geworfen werden. Die zentrale Komponente aus dieser Sicht soll gleich zu Anfang definiert werden.

Definition 12 (Shark Engine) *ist eine Software-Komponente, die einer Knowledge Node zugeordnet ist. Sie hat folgende Eigenschaften:*

Sie speichert Wissen strukturiert in einer eigenen Wissensbasis. Sie kann Eigenschaften der Umgebung wahrnehmen, eine Bezeichnung und Wissen mit anderen Peers austauschen, wobei der Austausch durch Knowledge Ports (KPs) gesteuert wird. Auch Knowledge Ports können ausgetauscht werden. Shark-Engines sind autonom in dem Sinne, dass sie anhand der Knowledge Port entscheiden können, ob ein Wissensaustausch erfolgen soll oder nicht. Outgoing Knowledge Ports (IKPs) erlauben einer Knowledge Node dort ihre Outputstrategie zu modellieren und in Incoming Knowledge Ports (OKPs) deren Relevanzfilter.

Abbildung 4.3 stellt die Teilkomponenten der Shark-Engine und ihre Beziehungen zu einer Knowledge Node dar.

Wie mehrfach beschrieben, muss jede Software, die das KM unterstützt, im Prozess des KM gesehen werden und niemals isoliert davon bzw. als Ersetzung aller KM-Prozesse. Das zeigt auch die Abbildung. Es ist dargestellt, dass eine Knowledge Node KM-Policies definiert. Diese betreffen die allgemeinen KM-Prozesse im Unternehmen.

Teile dieser KM-Policies lassen sich in Form von Interessen und Erwartungen an einen Austausch mit anderen Knowledge Nodes beschreiben, wie es im vorangegangenen Abschnitt erfolgt ist. Diese Beschreibungen kann eine Knowledge Node in eine Shark-Engine mittels der Spezifikation eines Knowledge Ports ablegen. Die Komponente *KP-Management* verwaltet die Knowledge Ports.

Eine Knowledge Node kann explizites Wissen in einer Shark-Engine speichern. Daher existiert eine Wissensbasis, deren Struktur in Shark nicht näher definiert ist. Es muss lediglich möglich sein, Wissen anhand von Themen zu strukturieren und anhand von Themen Interessen zu spezifizieren.

4 Shared Knowledge (Shark)

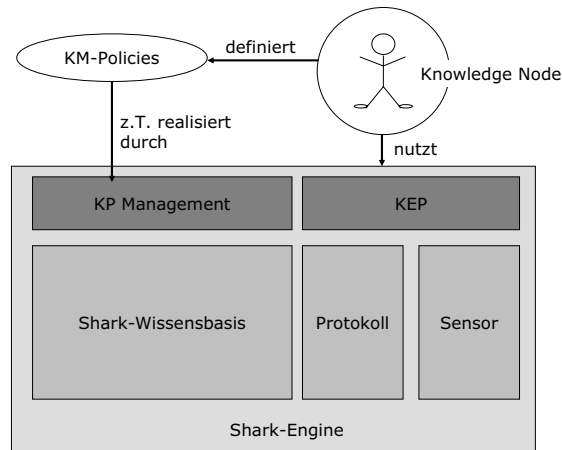


Abbildung 4.3: Architektur der Shark Engine

Knowledge Ports enthalten Regeln, die beschreiben, ob und wie ein Wissensaustausch erfolgen kann. Diese Regeln geben der Shark-Engine Autonomie. Für den Austausch benötigt die Shark-Engine aber ein Austauschprotokoll, das hier als *Knowledge Exchange Protocol (KEP)* bezeichnet wird. KEP kann als Protokoll der Anwendungsebene nach ISO-OSI verstanden werden. Es setzt auf Protokollen der unterliegenden Schichten auf, die hier nicht weiter spezifiziert werden, aber in der Abbildung angedeutet sind.

Eine Shark-Engine kann seine Umgebung beobachten. Es soll und kann nicht ausgeführt werden, welcher Art diese Beobachtungen sein sollen. Als Teilkomponenten wird für diesen Zweck der *Sensor* definiert. Er kann und soll nicht weiter ausgeführt werden. Es sollen nur einige Beispiele genannt werden, was ein Sensor sein kann.

- Webserver, der auf eingehende Nachrichten reagiert (die bemerkbare Änderung der Umwelt besteht lediglich in der Feststellung, dass eine Nachricht eintraf),
- eine Komponente, die nach Bluetooth-Geräten sucht (die Änderung der Umwelt besteht in der Entdeckung eines potenziellen Kommunikationspartners und dem möglichen Aufbau eines mobilen spontanen Netzes)
- Sensor, der die Temperatur misst und anhand von Schwellwerten Mitteilungen abgibt.

Bevor auf die Funktionen einer Shark-Engine und die Interaktionen der einzelnen Teilkomponenten eingegangen wird, sollen zuvor die Knowledge Ports definiert werden.

4.3.1 Knowledge Ports

Eine Shark-Engine spezialisiert die Ports des ACP-Modells zu Knowledge Ports. Diese regeln den Austausch von Wissen zwischen Shark-Engines.

Im ACP-Modell sind die Funktionen f, g und h Teil der Beschreibung eines Ports. In Shark werden die Funktionen auf die Kontextualisierung zurückgeführt. Es genügt daher, die Parameter zu kennen, die Funktionen unterscheiden sich zwischen Knowledge Ports nicht. Ebenfalls wurde der Erwartungskontext bezüglich Daten in Interesse spezialisiert.

Daher müssen die Funktionen nicht mehr Teil der Definition der Knowledge Ports sein, die sich dadurch wesentlich verkürzen:

Ein *Incomming Knowledge Port (IKP)* wird als Spezialisierung eines ACP Incomming Ports definiert als:

$$IKP = (X_{EU}, I_E, \text{Empfänger})$$

Ein *Outgoing Knowledge Port (OKP)* wird als Spezialisierung eines ACP Outgoing Ports definiert als:

$$OKP = (X_{EU}, I_S, \text{Sender})$$

Bereits im ACP-Modell wurde erklärt, dass Ports durch Daten beschrieben werden. Dieses Tatsache gilt auch in Shark. Hier sind die Daten sogar klarer spezifiziert. Die drei Parameter eines Knowledge Ports sollen als dessen Regeln für den Austausch von Wissen verstanden werden. Sie können mittels einer Dateneinheit zwischen Shark-Engines ausgetauscht werden.

Eine Bemerkung zur Sprachregelung im Weiteren: Ein Knowledge Port ist eine Entität, die die Regeln für den Austausch von Wissen enthält *und* gleichzeitig den Austausch von Wissen steuert.

Ein *Austausch von Knowledge Ports* wird in Shark nicht angestrebt, sondern nur ein *Austausch der Regeln eines Knowledge Ports*, die beim Empfänger zur Erzeugung eines Knowledge Ports mit den gleichen Regeln benutzt werden können, was auf das gleiche Resultat hinausläuft.

Als sprachliche Vereinfachung soll trotzdem von einem *Austausch von Knowledge Ports* gesprochen werden, wenn eigentlich von einem *Austausch der Regeln eines Knowledge Ports* die Rede sein müsste.

Im Weiteren soll von einem *Aktivieren eines Knowledge Ports* die Rede sein, wenn Regeln benutzt werden, um einen Knowledge Port anzulegen, der nunmehr bereit ist, einen Wissensaustausch nach diesen Regeln durchzuführen. Das *Deaktivieren eines Knowledge Ports* bezeichnet das Löschen eines Knowledge Ports, ohne dass die Regel des KPs bei der Knowledge Node gelöscht wird. Ein deaktiviert Knowledge Port kann jederzeit wieder aktiviert werden. Das *Löschen eines Knowledge Ports* bezeichnet die Deaktivierung und das gleichzeitige Löschen der Regeln des Knowledge Ports. Ein gelöschter Knowledge Port kann nicht wieder aktiviert werden.

Nun sollen die Funktionen einer Shark-Engine gegenüber seiner Knowledge Node und das Zusammenspiel der Teilkomponenten näher betrachtet werden.

4.3.2 Funktionen einer Shark-Engine

Die Shark-Engine bietet ihrer Knowledge Node folgende Funktionen an:

4 Shared Knowledge (Shark)

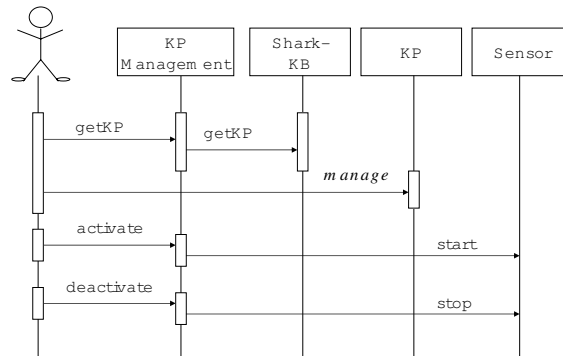


Abbildung 4.4: Knowledge Node nutzt Shark-Engine

Knowledge Ports verwalten Diese Funktion umfasst die Teilfunktionen Anlegen, Verändern und Löschen eines Knowledge Ports.

Shark-Engine aktivieren Aktivieren meint, dass die Shark-Engine nun alle Knowledge Ports aktiviert und, gültige Bedingungen vorausgesetzt, Wissen mit anderen Shark-Engines austauscht.

Shark-Engine deaktivieren Diese Funktion deaktiviert alle Knowledge Ports. Es wird kein Wissen mehr ausgetauscht. Prinzipiell kann auch nach der Deaktivierung ein Zugriff auf die Wissensbasis der Shark-Engine möglich sein.

Die drei Funktionen sind in Abbildung 4.4 durch ein Sequenzdiagramm skizziert, das die Interaktionen der Komponenten zeigt. Das KP-Management verwaltet die Knowledge Ports. Es erlaubt, neue Ports anzulegen und existierende zu verwalten und zu löschen.

Die Aktivierung einer Shark-Engine führt dazu, dass der Sensor aktiviert bzw. deaktiviert wird. Der Sensor beobachtet die Umgebung und übergibt bei einer Änderung dem Knowledge Port Management den aktuellen Umgebungskontext. Das KP-Management kann nun allen Knowledge Ports diesen Umgebungskontext mitteilen, die daraufhin aktiv werden und ermitteln, ob ein Wissensaustausch erfolgen soll.

Knowledge Ports steuern den Wissensaustausch und benutzen dazu ein Protokoll KEP, das hier nicht definiert wird, das aber die einzelnen Schritte des Wissensaustausches, wie sie in Abbildung 4.2 beschrieben wurden, ermöglichen muss. Abbildung 4.5 die Interaktionen der Komponenten einer Shark-Engine während eines Wissensaustausches: Der Sensor gibt dem KP-Management einen veränderten Umgebungskontext bekannt. Dieser wird jedem einzelnen Knowledge Port übergeben, der dann seinen Regeln entsprechend

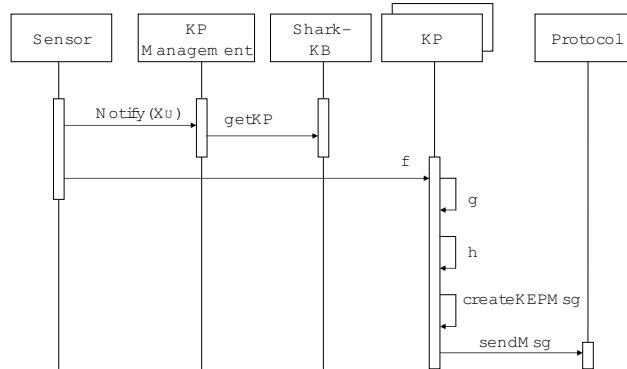


Abbildung 4.5: Aktive Shark-Engine

Wissen austauschen kann oder nicht. Dieser Ablauf ist durch das Aufrufen der Funktionen f, g und h und das Versenden einer KEP-Nachricht skizziert.

In den folgenden Abschnitten sollen weitere Aspekte der Shark-Engine, insbesondere der Vergabe und Veränderung von Bezeichnern von Besitzern, Sendern und Empfängern von Wissen beschrieben werden.

4.4 Eigentümer, Sender und Empfänger von Wissen

Bis hierher wurden die Konzepte Peer, Kontext, Daten, Datenbasis und Ports des ACP-Modell für Shark spezialisiert. Das Konzept der Umgebung und die Tatsache, dass Umgebungen von Peers wahrgenommen werden können, gilt in Shark unverändert. Die Einführung der Kontextualisierung erlaubte, die Funktionen f, g und h zu vereinfachen und die Beschreibung der Knowledge Ports auf einen Kontext, ein Interesse und eine Bezeichnung zu reduzieren. Es wurde ebenfalls gezeigt, wie Wissen von einer Knowledge Node angeboten und von einer anderen entgegengenommen werden kann. Analog können Knowledge Ports übertragen werden und es wurde erklärt, was im Detail damit gemeint ist.

In diesem Abschnitt soll sich eingehender mit den Bezeichnungen von Eigentümer von Wissen, Sender und Empfänger von Knowledge Ports und deren Wechsel beschäftigt werden. Im Gegensatz zum ACP-Modell gelten in Shark einige Einschränkungen, die im Folgenden durch eine Reihe von Policies definiert werden.

Bereits im ACP-Modell wurde definiert, dass jedes Datum einen Eigentümer hat. Dieses Konzept wurde nicht weiter benutzt, d. h., Eigentümer konnten beliebig wechseln. In Shark erfolgen einige Einschränkungen dazu. Am Anfang sollen drei Definitionen

4 Shared Knowledge (Shark)

stehen, die die Benennung von Wissen vor allem in Gruppen vereinfachen werden.

Definition 13 (Eigenes Wissen) *ist Wissen, das in der Wissensbasis einer Knowledge Node steht und ihr gehört.*

Definition 14 (Übernommenes Wissen) *ist Wissen, das in der Wissensbasis einer Knowledge Node steht und kein eigenes Wissen ist.*

Analog kann von *eigenen* bzw. *übernommenen* Knowledge Ports, Kontexten oder Interessen gesprochen werden.

Es soll eine dritte Art von Wissen definiert werden, die nur für Meta-Peers gilt. Meta-Peers repräsentieren Gruppen von Peers. Sie können von Peers ihrer Gruppen Wissen empfangen, das diese stellvertretend für die Gruppe entgegengenommen haben. Dieses Wissen wird zunächst vom jeweiligen Mitglied der Gruppe und später von Meta-Peer gespeichert, ohne es inhaltlich zu bearbeiten oder zu bewerten. Das Meta-Peer überträgt dieses Wissen schlussendlich an das Peer, das Wissen *für die Gruppe* assimiliert.

Per Definition ist diese Form des Wissens *übernommenes Wissen*. Es soll aber eine eigene Bezeichnung erhalten, die in [Sch02] als *unconfirmed knowledge* bezeichnet wurde:

Definition 15 (Unbestätigtes Wissen) *ist übernommenes Wissen, das von einer Knowledge Node stellvertretend für eine andere Knowledge Node empfangen wurde und noch nicht zu dieser übertragen wurde.*

Einfacher gesagt ist unbestätigtes Wissen das Wissen in einer Gruppe, welches auf dem Weg zum Bearbeiter des Wissens ist, aber noch nicht dort angekommen ist.

Jede Zuordnung von Wissen zu einer Kategorie kann mittels eines Kontextes erfolgen. So ist die Zuordnung des eigenen Wissen einfach möglich, indem ein Kontext eine Beziehung zwischen dem Wissen und dem Peer herstellt. Bei den anderen kann ein Beziehung zur Bezeichnung des Besitzers bzw. zu einem Konzept *unbestätigt* hergestellt werden.

4.4.1 Shark Policies

Im ACP-Modell wurde bereits definiert, dass jeder Port eine Sender- bzw. Empfängerbezeichnung trägt. Jedes Peer war aber frei, diese beliebig zu ändern. In Shark werden einige Restriktionen definiert, die sich aus der Modellierung der Fachdomäne Knowledge Management ergeben.

Anlegen von Knowledge Ports

Die folgende Regel schränkt die Wahl der Bezeichnungen in Knowledge Ports ein.

Regel 1 (Sender- und Empfängerbezeichnungen von Ports) *Legt eine Knowledge Node einen Port an, so kann als Sender- bzw. Empfängerbezeichnung nur die eigenen Bezeichnung oder die Bezeichnung „anonymes Peer“ benutzt werden.*

Einer Knowledge Node steht es frei, die Sender- bzw. Empfängerbezeichnung übernommener Knowledge Ports in die eigene oder die Bezeichnung „anonymes Peer“ zu ändern.

Andere Änderungen an diesen Bezeichnungen sind nicht erlaubt.

Die folgende Regel macht Restriktionen über die Änderung der Eigentümerbezeichnung von Wissen.

Regel 2 (Eigentümer von Wissen) *Legt eine Knowledge Node Wissen an, so kann es als Besitzer seine eigene Bezeichnung oder die Bezeichnung „anonymes Peer“ wählen.*

Eine Knowledge Node darf übernommenes Wissen zu eigenem Wissen machen, indem es die Eigentümerbezeichnung in die eigene ändert. Es darf weiterhin eigenes Wissen anonymisieren, indem es ihm den Eigentümer „anonymes Peer“ gibt.

Weitere Änderungen der Eigentümerbezeichnungen sind nicht erlaubt.

Shark legt weder fest, wie Knowledge Nodes generell und speziell Gruppen (und deren Repräsentanten, die Meta-Peers) entstehen noch wie Knowledge Nodes die Bezeichnungen von Gruppen ermitteln können. Es ist ebenfalls jenseits des Modells, wie die Shark-Engines an eine Knowledge Node gebunden werden und eine Bezeichnung erhalten. Ebenfalls ist in Shark nicht definiert, wie eine Shark-Engine Bezeichnungen anderer erhält. Das ist Gegenstand einer Architektur. Es regelt nur die Definition und den Austausch von Wissen und Knowledge Ports.

Es war zu sehen, dass das ACP-Konzept der Meta-Peers geeignet war, um soziale Gruppen zu modellieren. An dieser Stelle soll gezeigt werden, wie auch Shark-Engines solche Strukturen abbilden. Dazu sollen zunächst die Bildung von Gruppen und die Beziehung zwischen Gruppe und Mitgliedern motiviert werden.

Jede Knowledge Node verfügt über Themen. Schließt sich eine Knowledge Node einer Gruppe an, so kann das zwei Gründe haben: Sie möchte an Teilen des Wissens der Gruppe partizipieren oder sie möchte zum Wissen der Gruppe beitragen. Im ersten Fall würde sie als Proxy der Gruppe agieren, im zweiten Fall als Master. Sie kann beide Rollen parallel spielen.

Ist sie ein Proxy der Gruppe, so ist sie nicht gezwungen, alles Wissen der Gruppe zu übernehmen. Das wird weder durch das ACP-Modell verlangt, noch ergibt sich das aus Notwendigkeiten des KM. Im Gegenteil, es kann davon ausgegangen werden, dass für Mitarbeiter nur einige Themen relevant sind.

Ist sie ein Master der Gruppe, so wird sie in der Regel nur zu bestimmten Themen Wissen und Interessen beisteuern.

Wissen kann nur aus einer Gruppe hinaus- oder in eine Gruppe hineinfließen, wenn Shark-Peers Interessen der Gruppe übernehmen und basierend darüber Wissen austauschen. Die Gruppe hat daher ein Eigeninteresse an der Übergabe von Interessen an ihre Mitglieder. Wie gezeigt wurde, soll aber unter dem Namen der Gruppe nur Wissen der Gruppe versendet werden. Das Wissen, das Mitglieder der Gruppe stellvertretend für die Gruppe entgegennehmen, ist zunächst unbestätigtes Wissen. Es wird zu bestätigtem Wissen, sobald das zuständige Shark-Peer entschieden hat, dass das Wissen Teil des Gruppenwissens werden soll.

4 Shared Knowledge (Shark)

Dieses Verhalten kann auf Shark-Engines der beteiligten Knowledge Nodes durch passende Definition der Knowledge Ports abgebildet werden. Das soll mittels der folgenden Regeln erfolgen. Sie stellen eine spezialisierte verkürzte Darstellung des Konzeptes der ACP-Peer-Gruppen dar, siehe Abschnitt 3.2.1.

Regel 3 (Gruppenmitglieder werben) *Eine Shark-Engine, deren Knowledge Node eine Gruppe repräsentiert, darf andere Shark-Engines als Mitglieder werben. In Shark gilt, dass eine Gruppe alles Wissen, das nicht unbestätigt, und alle Themen und Interessen, über die sie verfügt, ihren Mitgliedern offen legen muss. Das muss durch einen OKP erfolgen, der wie folgt definiert ist:*

$$OKP_{Master} = \{Mitglieder \cup X_{EU}, T_G \cup I_G, Gruppe\}$$

Der Parameter Mitglieder kann eine Liste aller potentiellen Mitglieder der Gruppe sein. Er ist als Erwartung gegenüber der Umwelt definiert, denn das Auftauchen eines solchen Peers wird als Veränderung in der Umgebung wahrgenommen. T_G seien die Themen der Gruppe und I_G die Interessen der Gruppe. X_{EU} deutet an, dass noch weitere Bedingungen an die Umwelt gestellt werden können, wie z. B. verschlüsselte Kommunikation o. Ä. Der letzte Parameter beschreibt, dass die Engine die Bezeichnung der eigenen Knowledge Node wählen muss, also die, die die Gruppe repräsentiert.

Gibt es Empfangsinteresse in der Gruppe, so muss die Shark-Engine einen IKP anlegen mit folgende Definitionen:

$$IKP_{Master} = \{Mitglieder \cup X_{EU}, unbestaetigt \cup T_G, Gruppe\}$$

Der IKP beschreibt, dass von Mitgliedern der Gruppe unbestätigtes Wissen zu Themen der Gruppe angenommen wird.

Die Shark-Engine der Gruppe spielt in dieser Konstellation die Rolle eines Masters in einer Master-Proxy-Beziehung, weshalb die Knowledge Ports auch entsprechend benannt sind.

Regel 4 (Gruppenmitglied werden) *Jede Shark-Engine kann sich als Mitglied einer Gruppe anbieten. Dazu muss ein IKP der folgenden Form anlegen:*

$$IKP_{Proxy} = \{Gruppe \cup X_{EU}, T_G' \cup I_G', Mitglied\}$$

Als Anforderungen an die Umgebung werden definiert, dass die Shark-Engine der Gruppe Kommunikationspartner ist. Andere mögliche Werte können gesetzt werden. Es können Teile der Themen und Interessen der Gruppe ausgewählt werden. Der IKP trägt die Bezeichnung der eigenen Knowledge Node, die hier als Mitglied bezeichnet wird.

Ist die Engine auch bereit Empfangsinteressen und damit IKPs der Gruppe anzunehmen, so muss sie einen weiteren OKP definieren der Form:

$$OKP_{Proxy} = \{Gruppe \cup X_{EU}, unbestaetigt \cup I_G', Mitglied\}$$

Der OKP beschreibt, dass sie unbestätigtes Wissen zu den Themen der Gruppe, die ausgewählt wurden, an die Gruppe übermitteln will.

Die Shark-Engine des Mitgliedes spielt in dieser Konstellation die Rolle eines Proxies in einer Master-Proxy-Beziehung, weshalb die Knowledge Ports auch entsprechend benannt sind.

Die folgende Regel beschreibt, wie ein Mitglied einer Gruppe mit Knowledge Ports umgehen muss, die es von der Gruppe empfängt.

Regel 5 (Umgang mit übernommenen Knowledge Ports) *Übernimmt eine Shark-Engine Knowledge Ports von einer Gruppe, so gilt:*

Über übernommene OKPs darf nur Wissen angeboten werden, das dem im OKP bezeichneten Sender gehört. Die Extraktionsprozesse müssen entsprechend definiert sein.

Über übernommene IKPs empfangenes Wissen muss als unbestätigtes Wissen markiert werden. Dieses Wissen darf nicht in einer anderen Form assimiliert werden. Die Assimilationsprozesse müssen entsprechend definiert sein.

Jede Gruppe muss wenigstens ein ausgezeichnetes Peer haben, das Themen und Interessen der Gruppe verwaltet. Diese sollen als Experten bezeichnet werden. Die folgende Regel definiert das Zusammenspiel zwischen ihnen und der Gruppe.

Regel 6 (Beziehung Gruppenexperten-Gruppe) *Eine Shark-Engine einer Knowledge Node, die Experte in einer Gruppe ist, muss folgende Knowledge Ports anlegen:*

$$OKP_{Master} = \{Gruppe \cup X_{EU}, T_E \cup I_E, Experte\}$$

$$IKP_{Master} = \{Gruppe \cup X_{EU}, unbestaetigt \cup T_E, Experte\}$$

Es sind wieder zwei Ports, die den Experten als Master kennzeichnen. Der OKP legt fest, dass der Master bereit ist, Wissen und Interessen zu einigen Themen (T_E) zu versenden. Der IKP legt fest, dass er bereit ist, unbestätigtes Wissen zu seinen Themen anzunehmen.

Die Gruppe muss passende Proxy-Ports anlegen:

$$IKP_{Proxy} = \{Experte \cup X_{EU}, T_E \cup I_E, Gruppe\}$$

$$OKP_{Proxy} = \{Experte \cup X_{EU}, unbestaetigt \cup T_E, Gruppe\}$$

Die Gruppe ist bereit, Wissen und Interessen des Experten zu einigen Themen anzunehmen und im Gegenzug unbestätigtes Wissen zu ausgewählten Themen an den Experten zu senden.

Es folgt eine abschließende Regel über den Umgang mit nicht eigenem Wissen und Knowledge Ports

Regel 7 (Änderung nur eigener KPs und eigenen Wissens) *Eine Knowledge Node darf nur eigenes Wissen und eigene Knowledge Ports in seiner Shark-Engine verändern.*

Die gerade definierten Regeln verfeinern das Konzept der Peer-Gruppen des ACP-Modells dahingehend, dass sie definieren, wie und unter welchen Bedingungen die Bezeichnungen für Eigentümer, Sender und Empfänger von Wissen geändert werden dürfen. Diese Regeln sind notwendig, um die Wissenflüsse, wie sie im Knowledge Management in realen Situationen beobachtet und in Abschnitt 2.5 beschrieben wurden, korrekt zu modellieren. Diese Regeln spezialisieren die Flüsse von Wissen und werden im Folgenden erläutert.

Die Regeln definieren die Beziehung zwischen Shark-Engines, basierend auf dem Ansatz der Master-Proxy-Beziehung, die bereits im ACP-Modell beschrieben wurde. Es soll an dieser Stelle auf die Problematik des Löschens von Wissen und Knowledge Ports aufmerksam gemacht werden. Es gibt keine Methoden, mit der ein Master einem Proxy zum Löschen von Wissen und Knowledge Ports veranlassen kann. Im ACP-Modell wurde eine Variante vorgeschlagen, wie ohne eine Löschfunktion die Proxies trotzdem immer genau über die Daten und Ports verfügen, wie der Master, siehe Abschnitt 3.2.1. Diese Lösung kann auch hier angewendet werden. Es gilt wie im ACP-Modell, dass für reale Implementierungen performantere Lösungen eingesetzt werden können.

In den folgenden Darstellungen werden, wie bereits beim ACP-Modell, die vier Knowledge Ports, die die Master-Proxy-Beziehung realisieren, durch einen Doppelpfeil symbolisiert, der Master und Proxy verbindet und die Farbe des Masters hat. Die vier Knowledge Ports wurden in den Regeln 3, 4 und 6 definiert.

4.5 Wissensflüsse

Die Shark-Policies spezialisieren die Flüsse von Wissen gegenüber den Datenflüssen des ACP-Modells. Insbesondere ist ein Eigentümerkonzept entwickelt worden, das die im KM beobachteten und beschriebenen Wissensflüsse abbildet. Ein solches Konzept fehlt in ACP. Dort wurden lediglich Empfehlungen ausgesprochen, ansonsten konnte jedes Peer beliebig alle Bezeichnungen verändern.

Die Implikationen der Regeln sollen zunächst anhand der Wissensflüsse innerhalb von Gruppen gezeigt werden, danach anhand von Flüssen, die über Gruppengrenzen hinausgehen.

4.5.1 Wissensflüsse zwischen Gruppen und Gruppenmitglied

Die Erläuterungen sollen wieder anhand eines einfachen Beispiels erfolgen. Angenommen, drei Knowledge Nodes (A, X, und Y) haben sich zu einer Gruppe (G) zusammengeschlossen. Sie haben vereinbart, dass X als Experte für die Themen X in der Gruppe agiert und Y für die Themen Y. A ist in dieser Konstellation kein Experte, sondern lediglich einfaches Gruppenmitglied. Die Beziehungen und Wissenflüsse ihrer Shark-Engines sollen im Folgenden untersucht werden, siehe Abbildung 4.6.

Es soll beachtet werden: In dieser Abbildung sind Shark-Engines dargestellt. Im Text wird zumeist von Gruppen, Mitgliedern usw. gesprochen. Das ist nur eine sprachliche Verkürzung. Korrekt müsste es z. B. anstelle von *Gruppe* heißen: *Shark-Engine*, die

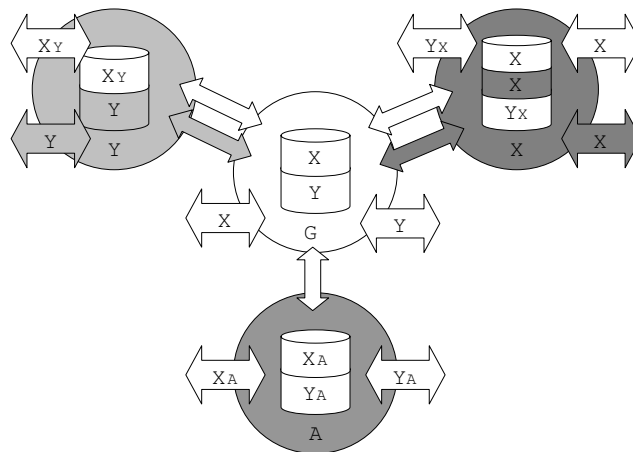


Abbildung 4.6: Wissensflüsse innerhalb einer Gruppe

der *Knowledge Node* zugeordnet ist, die die Gruppe repräsentiert. Aus Gründen der Lesbarkeit wurde die sprachliche Verkürzung gewählt.

Im Zentrum steht die Gruppe. Sie agiert als Master gegenüber jedem Gruppenmitglied (weiße Doppelpfeile). Die Erstellung dieser Beziehung wird durch die Regeln 4 und 3 reglementiert.

X und Y sind Experten in den Gruppen und agieren daher als Master gegenüber G (dunkle Doppelpfeile). Diese Beziehung wurde durch die Regel 6 verfeinert.

Es soll zunächst X betrachtet werden. X ist Experte für die Themen von X, erzeugt also dazu neues Wissen bzw. sortiert empfangenes ein. Er ist daher der Besitzer dieses Wissens. Das ist durch den dunklen Zylinder in X symbolisiert, der mit X beschriftet ist. Er definiert weiterhin Interessen zu diesen Themen, die sich in Knowledge Port wiederfinden. Diese Knowledge Ports hat X angelegt und sich selbst oder das „anonyme Peer“ als Sender bzw. Empfänger eingetragen. Symbolisiert werden diese Knowledge Ports durch den dunklen Doppelpfeil bei X.

X ist Master gegenüber der Gruppe, was die Themen X betrifft. Er kann daher sowohl Wissen als auch Knowledge Ports an die Gruppe übermitteln. Diese nimmt sie entgegen. Gemäß Regel 6 übernimmt die Gruppe das Wissen, das über diesen Weg von X kommt, und ändert nach dem Empfang die Sender- und Empfängerbezeichnungen in den Knowledge Ports in die der Gruppe. Das ist durch die veränderte Farbe dargestellt. Sowohl das Wissen als auch die Knowledge Ports zu den Themen X sind in G weiß dargestellt.

Die gleiche Beziehung herrscht zwischen Y und der Gruppe G. G sammelt also in seiner Funktion als Proxy der Experten der Gruppe Wissen und Knowledge Ports zu den Themen, die die gesamte Gruppe interessieren. Dabei ist G Eigentümer allen Wissens der Gruppe und Sender und Empfänger in den entsprechenden Knowledge Ports.

4 Shared Knowledge (Shark)

G ist Master gegenüber allen Mitgliedern der Gruppe, also A, X und Y. Es soll zunächst Knowledge Node A betrachtet werden. Als Gruppenmitglied sollte sie Interesse wenigstens an einigen Themen der Gruppe haben, sonst müsste sie nicht deren Mitglied sein. Sie muss sich aber nicht für alle Themen interessieren. In der Abbildung interessiert sich A für die einige Themen aus X und Y, die mit X_A und Y_A bezeichnet sind, wobei gelten soll $X_A \subset X$ und $Y_A \subset Y$. Dadurch erhält A nur Teil des Wissens von G. Auch die Knowledge Ports von G zeigen nur ein eingeschränktes Interesse verglichen mit G.

Es soll nun X betrachtet werden. Hier gilt das Gleiche wie für A. Gegenüber den Themen Y kann es ein beliebiges Interesse bekunden, was in dem Fall durch Y_X symbolisiert wird. Das Verhältnis der Knowledge Node X gegenüber den Gruppenthemen X ist interessanter:

X ist der Experte für die Themen X. Es ist aber nicht in der Lage, Knowledge Ports anzulegen, mit denen es unter der Bezeichnung der Gruppe Wissen zum Thema X austauschen kann. Wenn es das will, muss es, wie jedes andere Gruppenmitglied auch, ein Interesse an dem Thema X gegenüber der Gruppe G bekunden und entsprechende Knowledge Ports und Wissen empfangen. In der Abbildung hat sich X beispielhaft für alles Wissen zum Thema X entschieden und auch für alle Knowledge Ports zum Thema X. Es verfügt daher zweimal über das gleiche Wissen und die gleichen Knowledge Ports. Einmal gehört das Wissen ihm selbst, einmal der Gruppe. Die einen Knowledge Ports tragen die Bezeichnung X als Sender bzw. Empfänger, die anderen G^2 .

Im Gegensatz dazu hat sich der zweite Experte Y nur für Teile aus den Themen X entschieden (X_Y). Er ist nicht daran interessiert, unter der Bezeichnung der Gruppe Wissen mit anderen auszutauschen. Es hat also nur Gruppenwissen und Knowledge Ports zu X_Y und eigenes Wissen und KPs zu Y.

Diskussion

Die Beziehung zwischen Gruppenmitgliedern, die keine Experten sind, und der Gruppe birgt keine Überraschungen und entspricht den Beobachtungen in der zu modellierenden Domäne des Knowledge Managements. Als Gruppenmitglieder sollen sie in die Lage versetzt werden, in der Rolle der Gruppe zu agieren. Das gelingt durch dieses Vorgehen. Sie können sich frei entscheiden, welches der Themen und welche Knowledge Ports, d. h. Interessen der Gruppe sie übernehmen wollen. Die Einschränkung ihrer Autonomie liegt darin, dass sie Wissen, das über IKPs der Gruppe empfangen wurde, auch an die Gruppe weiterleiten müssen und es ihnen verboten ist, dieses Wissen zu assimilieren. Sie dürfen auch über OKPs der Gruppe nur Wissen der Gruppe verteilen.

Diese Reglementierung ist weniger hart, als es zunächst erscheint. Jeder Knowledge Node steht es frei, Knowledge Ports zu löschen, anzulegen oder zu übernehmen.

So kann ein Gruppenmitglied jederzeit seine Mitgliedschaft beenden, es kann Kopien der Knowledge Ports der Gruppe anlegen und sogar das Wissen der Gruppe übernehmen. Es darf nur nicht eigenes Wissen über Knowledge Ports der Gruppe senden oder

²Die Situation kann auch so interpretiert werden, dass das Wissen und die Knowledge Ports zu den Themen bzw. Interessen X sowohl *im Kontext der Gruppe* als auch *im Kontext von X* gelten.

umgekehrt. Es muss Wissen an die Gruppe weitergeben, das es stellvertretend für sie empfangen hat³.

Interessant ist auch die Situation bei X. Es scheint paradox, dass X Knowledge Ports unter eigener Bezeichnung anlegen kann und diese auch an die Gruppe übertragen kann, gleichzeitig aber nicht in der Lage ist, direkt einen Knowledge Port unter der Gruppenbezeichnung anzulegen. Dieses Vorgehen modelliert aber die Realität und beachtet, dass in Gruppen als sozialen Systemen über die Outputstrategien und Relevanzfilter entschieden wird. Diese Entscheidungen werden nur in Ausnahmefällen durch einzelne Gruppenmitglieder gefällt werden. Vielmehr wird davon ausgegangen, dass innerhalb der Gruppe weitere Abstimmungsprozesse ablaufen, die allerdings in Shark nicht detailliert modelliert sind und wahrscheinlich nicht einmal durch Software implementiert werden können. Vielleicht findet ein Treffen der Gruppen statt oder ein Gruppenleiter bespricht sich in einem Gremium. Diese Abstimmungsprozesse sind Teil einer KM-Policy eines Unternehmens und werden in Shark nicht im Detail modelliert. Sie verbergen sich aber in der Assimilation der Knowledge Ports und des Wissens, das ein Experte der Gruppe in seiner Rolle als Master übergibt.

Generell kann jeder Austausch zwischen Peers als Arbeitsschritt in einem KM-Prozess interpretiert werden. Das Besondere an diesem Schritt ist, dass in KM und auch in der Modellierung durch Shark davon ausgegangen wird, dass die Übergabe von Wissen eher spontan und direkt zwischen Menschen erfolgt.

Ähnliche Fragen stellen sich auch bei den Freiheiten von A. Es erscheint unrealistisch, dass reale KM-Policies die Möglichkeiten abdecken, dass ein Gruppenmitglied ohne jede Rückfrage Wissen der Gruppe einfach übernehmen darf, ohne weiteres die Mitgliedschaft in der Gruppe beenden darf usw. In der Realität werden weitere Einschränkungen existieren, die über die Shark-Policies hinausgehen.

4.5.2 Wissensflüsse über Gruppengrenzen

Es wurden die Wissensflüsse innerhalb von Gruppen vorgestellt und diskutiert. Es sollen nun die Flüsse untersucht werden, die die Grenze der Gruppe überschreiten.

Abbildung 4.7 stellt die Knowledge Node A aus der vorherigen Abbildung dar. Der Übersicht wegen wurden eine Reihe von Knowledge Ports, die Wissensbasen und nur die zur Erklärung notwendigen Master-Proxy-Beziehungen eingezeichnet. Sie verfügt nach wie vor über Wissen zu den Themen X_A und Y_A und hat entsprechende Knowledge Ports. A wird nur in seiner Funktion als Gruppenmitglied gesehen, daher wird kein persönliches Wissen von A oder deren Knowledge Ports dargestellt oder diskutiert.

In der Abbildung tauscht A Wissen mit einer Knowledge Node Z aus. Wenn A alle Eigenschaften offen gelegt hat, kann Z die Bezeichnung von A erkennen und sehen, dass sie für die Gruppe G Wissen zu einigen Themen anbietet bzw. sucht (X_A und Y_A). In einem Schritt 1 hat A Wissen an Z gesendet. Gemäß der Regel 5 hat A Wissen der Gruppe G gesendet. Es agiert damit – wie gewünscht – klar erkennbar als Mitglied von G und übermittelt nur Wissen von G.

³Es ist den Implementierungen überlassen, die Einhaltung dieser Regeln sicherzustellen, was bei autonomen Peers sicherlich an vielen Stellen nicht trivial ist. Das ist Gegenstand späterer Diskussionen.

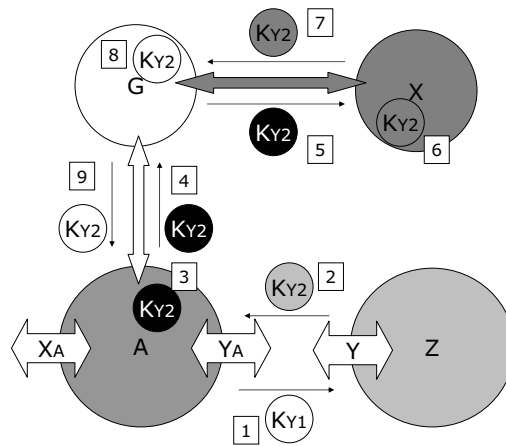


Abbildung 4.7: Wissensflüsse über Gruppengrenzen

In einem Schritt 2 hat es Wissen von Z aufgenommen. Gemäß Regel 5 nimmt es das Wissen entgegen und markiert es in Schritt 3 als *unbestätigtes Wissen* (Wechsel der Farbe nach schwarz). Es leitet es in einem Schritt 4 an die Gruppe weiter, die es seinerseits an einen Experten (hier X) im Schritt 5 weiterleitet. X kann nun das Wissen assimilieren. Wird es dabei nicht verworfen, so übernimmt X im Zuge der Assimilation das Wissen (Schritt 6), siehe Regel 6. Der wiederholte Farbwechsel deutet die Übernahme des Wissens an. X überträgt das Wissen in einem Schritt 7 an die Gruppe, die es nun durch die Änderung der Eigentümerbezeichnung (Schritt 8) auch sichtbar zu Gruppenwissen macht. Dieses neue Wissen kann an jedes Gruppenmitglied verteilt werden (Schritt 9).

Diskussion

Das Szenario zeigt die Wirkung der Regeln, dass über Knowledge Ports nur das Wissen der Sender bzw. Empfänger ausgetauscht werden darf. Der Wissensfluss, der dazu führte, dass ein Gruppenmitglied über das Gruppenwissen verfügt, wurde bereits im vorherigen Abschnitt beschrieben. In diesem Abschnitt wurde zusätzlich der Wissensfluss beschrieben, der entsteht, wenn ein Gruppenmitglied Wissen *stellvertretend für die Gruppe* entgegennimmt. Es wird zunächst als unbestätigt markiert und über zwei Stationen an den Experten in der Gruppe weitergeleitet. Auch dieser Prozess beschreibt reale Prozesse. Eine Person, die in ihrer Rolle als Gruppenmitglied etwas empfangen hat, reicht dieses Wissen innerhalb der Gruppe an den Experten weiter, der damit entsprechend umgeht. Von dort aus fließt es gegebenenfalls über beschriebene gruppeninterne Prozesse wieder zurück zu den einzelnen Mitgliedern der Gruppe.

4.5.3 Wissensflüsse in Gruppenhierarchien

In den bisherigen Darstellungen wurden nur einzelne Gruppen betrachtet. Wie zu sehen war, bilden aber auch Gruppen übergeordnete Gruppen. Die Wissensflüsse in diesen

Strukturen sind nicht deutlich unterschiedlich zu den beschriebenen Flüssen.

Angenommen, die Gruppe G aus obigen Darstellungen wäre selber Mitglied einer weiteren Gruppe GS. Wie jedes Mitglied könnte G Themen und Interessen von GS erhalten. Als Gruppe wiederum ist sie verpflichtet, alle Themen und Interessen, über die sie verfügt, ihren Mitgliedern offen zu legen (Regel 4).

Dadurch wird Wissen von GS, das G übernommen hat, auch allen Mitgliedern der Gruppe G angeboten. Es werden auch alle Knowledge Ports angeboten. Was geschieht, wenn eine Mitglied von G, z. B. A, einen Knowledge Port übernimmt, den G von GS übernommen hat?

Es soll zunächst ein OKP betrachtet werden. Hat G den OKP als Mitglied von GS übernommen, darf sie ihn nicht ändern. Aus dem gleichen Grund bleibt der IKP unverändert, wenn A ihn von G bezieht. Damit gelangt er unverändert von GS zu A. Das Gleiche gilt für das Wissen, das zu den Sendeinteressen des OKP gehört. Es wäre – entsprechende Interessen bei G und A vorausgesetzt – von GS über G zu A geflossen. Nach Regel 5 darf A über diesen OKP nur Wissen senden, das GS gehört. Damit agiert A stellvertretend für GS.

Was geschieht mit einem IKP? Er könnte ebenfalls von GS über G zu A fließen. Wenn G den IKP von GS übernimmt, muss es – gemäß Regel 4 – eine OKP anlegen, der unbestätigtes Wissen zu den Interessen dieses IKP an GS überträgt. Gemäß der gleichen Regel darf der Empfänger dieser Nachricht nur das Peer sein, von dem der Knowledge Port empfangen wurde. Das Gleiche gilt für A. Es empfängt den IKP und richtet einen OKP ein, der unbestätigtes Wissen, das zu den Empfangsinteressen des IKPs passt, an G sendet. Es ist zu beachten, dass der IKP selber an keiner Stelle verändert werden durfte. Wenn A ihn also aktiviert, so präsentiert er als Empfangsbezeichnung entweder GS oder das anonyme Peer. Wird darüber Wissen empfangen, wo wird es von A als unbestätigt markiert (Regel 5), was dazu führt, dass er es an G übermittelt – an G und nicht direkt an GS, denn in G ist A direkt Mitglied und hat eine entsprechende Master-Proxy-Beziehung aufgebaut, nicht mit GS! G wird seinerseits das unbestätigte Wissen über seinen OKP an GS senden. GS wird das Wissen an seinen Experten übertragen.

Dieser Experte kann auch ein Gruppe sein. In dem Fall würde die Experten-Gruppe das Wissen empfangen und an denjenigen weiterleiten, der die Rolle des Experten in der Gruppe hat. Dieser Prozess wird fortgesetzt, bis das Wissen eine Shark-Engine erreicht hat, die einem Shark-Peer gehört, als einem Menschen, der das Wissen bewerten kann.

Damit sei die Diskussion um Wissensflüsse in Shark beendet. Knowledge Ports sind keine komplexe Struktur. Sie enthält lediglich drei Parameter. Sie erlauben aber, recht komplexe Strukturen zu modellieren, die aber lediglich auf einem P2P-Austausch basieren. Entscheidender ist aber zusätzlich: Der Austausch von Wissen zwischen Gruppen erfolgt lediglich zwischen Individuen. Die Verteilung des Wissens innerhalb von Gruppen bedient sich des Konstruktes der Meta-Peers.

4.5.4 Korona einer Knowledge Node

Dieser Abschnitt soll durch eine Überlegung abgeschlossen werden, die einen Zusammenhang zwischen Wissen von Individuen und Gruppen herausarbeitet.

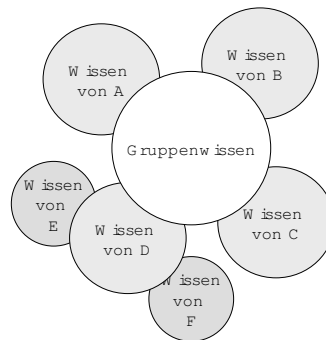


Abbildung 4.8: Korona von Wissen einer Gruppe

Wie aus Forschung im Knowledge Management bekannt ist und wie es auch durch Shark modelliert wird, können nur Individuen Wissen erzeugen. Mehr noch: Nur Individuen kommunizieren mit anderen. Das Konzept der Gruppen und Organisationen ist davon abgeleitet. Das heißt, dass alle Knowledge Nodes, die keine Individuen sind, neues Wissen nur erlangen können, indem Mitglieder von ihnen neues Wissen in die Gruppe einbringen, sei es, dass sie dieses Wissen selbst erzeugt haben oder dass sie das Wissen aus der Umgebung empfangen haben. Dieser Prozess kann über mehrere Hierarchiestufen ablaufen, beginnt aber immer bei einem Individuum.

Abbildung 4.8 illustriert diesen Zusammenhang. Im Zentrum steht das Wissen einer Gruppe. Mitglieder der Gruppe sind A, B, C und D, wobei D ebenfalls eine Gruppe ist, die die Mitglieder E und F hat. Die Abbildung stellt das Gruppenwissen als weißen Kreis dar. Es ist zu sehen, dass die Mitglieder der Gruppen über eigenes Wissen verfügen, dass es aber auch Überlappungen gibt. Woher kann neues Wissen in die Gruppe gelangen? A, B, C und D können Wissen aus der Umgebung aufnehmen bzw. wenn sie Personen sind, neues Wissen erzeugen. D ist eine Gruppe. Sie kann Wissen auch durch ihre Mitglieder E und F beziehen.

Die Gruppe wird über eine Policy verfügen, wie mit Wissen umgegangen wird, das durch Mitglieder in die Gruppe eingebracht wird. Details der Policy sind für diese Diskussion nicht von Interesse. Interessant ist die Tatsache, dass dank der Mitglieder einer Gruppe latent neues Wissen zur Verfügung steht. Dieses Wissen soll als *Korona* bezeichnet werden.

Wissen aus der Korona einer Gruppe kann Gruppenwissen werden. Mehr noch, alles Wissen, das *nicht* durch Austausch aus der Umgebung empfangen wird, *muss* aus der Korona der Gruppe kommen.

Diese Betrachtungsweise deckt sich auch mit praktischen Erfahrungen. Was geschieht, bevor eine neue Technologie, ein neues Werkzeug eingeführt wird? Ein Mitarbeiter beschäftigt sich mit den Alternativen und gewinnt dadurch Wissen. In einem gruppen-internen Prozess (zur Erinnerung, das ist nach Nonaka und Takeuchi die Sozialisierung) wird entschieden, ob und welches Tool in Zukunft von der Abteilung eingesetzt wird. Der Mitarbeiter schreibt dann üblicherweise einige Tipps und Literaturhinweise in ein Dokument (Externalisierung) und stellt es den anderen zur Verfügung. Im letzten Schritt geht das Wissen von der Person in das Gruppenwissen über, geht also von Korona in das Wissen der Gruppe ein. Andere Mitglieder der Gruppe können dieses Wissen nun beziehen und anwenden (Internalisierung).

4.6 Zusammenfassung

In diesem Kapitel wurde die Referenzarchitektur Shark präsentiert. Sie stellt eine Spezialisierung des ACP-Modells dar. Bei der Darstellung wurde sich an den ODP-Sichten orientiert, ohne sich an Formalien zu halten und die Sichten jeweils ganz genau einzuhalten.

In Kapitel ?? wurde Wissen und der Austausch von Wissen untersucht. Diese Untersuchung mündete in Abschnitt 2.5, in dem eine Erklärung vorgelegt wurde, wie Wissensflüsse in Unternehmen auf P2P-Wissensaustausche zwischen Individuen zurückgeführt werden können. Shark hat diese Erklärung in eine Referenzarchitektur übernommen. Shark lässt die konkrete Umsetzung eines System offen. Insbesondere wurde nicht diskutiert, welche Sprache der Wissensrepräsentation geeignet wäre für eine Umsetzung der Prinzipien von Shark, insbesondere Kontextualisierung. Fest steht, dass die Sprache das Prinzip der aggregierten Mikrowelten erlauben muss. Im folgenden Kapitel 5 wird gezeigt, wie Topic Maps für diesen Zweck eingesetzt werden kann.

Es wurde weiterhin eine Definition des Knowledge Exchange Protocols offen gelassen und Überlegungen über Protokolle und Plattformen, auf denen das Protokoll bzw. die gesamte Shark-Engine implementiert werden könnte, siehe dazu Abschnitt 6.3.1.

In Abschnitt 2.12 wurden Eigenschaften zentraler KMS kritisiert. Die Kritik konzentrierte sich vor allem auf folgende Punkte: Ein Chief Knowledge Officer ist nicht für alle KM-Prozesse passend, so dass er zur Entfremdung des Wissens beiträgt. Wissen wird als persönliches Gut angesehen und deshalb mit Vorliebe auf den eigenen Systemen gehalten, trotz Einführung von KMS in Unternehmen. Wissensaustausch erfolgt gern P2P, also auch per E-Mail, aber ungern durch zentrale Server, auf denen Dokumente abgelegt werden.

Shark löst diese Probleme. Die Definition von Wissensflüssen erfolgt dezentral durch Knowledge Nodes mit ihren Shark-Engines. Eine Shark-Engine ist eine Software-Komponente, die einer Knowledge Node zugeordnet ist. Es wird zu sehen sein, dass eine Shark-Engine auch auf einem Handy laufen kann, auf einem PDA und anderen kleinen Geräten und natürlich auf dem Laptop bzw. PC von Mitarbeitern. Solche Geräte können direkt mittels Bluetooth miteinander kommunizieren. Wenn im Referenzmodell von P2P-Kommunikation gesprochen wurde, kann das weniger abstrakt verstanden werden, als

4 Shared Knowledge (Shark)

es klingen mag. Stellt man sich Shark-Engines auf PDAs vor, so können diese direkt mittels Bluetooth Wissen austauschen. Das Wissen kann direkt auf den PDAs eingesehen werden und weiter in einer Gruppe oder anderen Netzen wandern. Mobile Systeme mit der technischen Möglichkeit der Spontanvernetzung bieten die Chance, wirklich *mobiles verteiltes* Wissen zu etablieren, was das ursprüngliche Ziel von Shark war und nach wie vor ist.

Eine Shark-Engine kann aber auch als ein sehr persönliches Gerät verstanden werden. Es gehört einem Menschen. Es ist auch nicht mehr nötig, Wissen aktiv auf einen Server hoch- oder herunterzuladen, geschweige denn auf der Ebene von Dokumenten. Mit Shark müssen Mitarbeiter beschreiben, zu welchen Gruppen sie gehören, welche Themen und Interessen der Gruppe sie teilen und ob sie selber als Experten in der Gruppe mitarbeiten wollen. Der Abstimmungsprozess über die Gruppenbildung ist ein zutiefst sozialer und soll und kann nicht von Shark übernommen werden. Ist aber eine Gruppe etabliert, so erhält jedes Mitglied der Gruppe automatisch das Wissen, das von Interesse ist, und agiert gleichzeitig stellvertretend für die Gruppe, wenn es um den Austausch von Wissen geht. Alle Arbeiten erfolgen mit der eigenen Shark-Engine, die das Wissen dann weitergibt.

KM zeigt, dass es sowohl offizielle, aber vor allem auch inoffizielle Flüsse von Wissen gibt. Knowledge Officer definieren die ersten mittels verbindlicher KM-Policies im Unternehmen. Diese sind notwendig für ein effektives Arbeiten. Daneben aber müssen informelle Netze unterstützt werden. Shark erlaubt beides. Es bedarf in Shark keiner Zentrale, in der die Knowledge Ports definiert werden. Sie werden direkt in einer Shark-Engine definiert. Die offiziellen Gruppenhierarchien können und sollen natürlich zentral definiert werden. Die Bildung von Ad-hoc-Gruppen ist aber ebenso einfach möglich.

In Shark werden Gruppen als Mediatoren aufgefasst, also als Entitäten, die etwas entgegennehmen und weiterleiten. Exakt das ist die Funktion einer sozialen Gruppe. Sie bildet sich und erzeugt eine relativ stabile innere Struktur, die der Arbeitsteilung und der Erzeugung einer Gruppenidentität dient. Das Konzept der Shark-Meta-Peers bildet das direkt ab. Auch einem Meta-Peer wird eine Shark-Engine zugeordnet. Wie die konkrete technische Realisierung aussieht, wurde offen gelassen. Ist es ein dedizierter Gruppenserver oder sogar ein Teil eines Workflowsystems? Das ist der jeweiligen Architektur überlassen.

Shark-Engines können spontan Wissen austauschen. Die Definition der Knowledge Ports erfolgt dezentral. Die Flüsse von Wissen können aus dieser Perspektive als emergente Funktion aufgefasst werden. Sie werden an keiner Stelle zentral definiert, machen aber den Charakter eines Unternehmens aus, siehe auch das Konzept der *lernenden Unternehmen* von Senge [Sen94]. Das gilt auch und gerade, wenn Teile der Wissensflüsse von einem zentralen Chief Knowledge Officer definiert werden. Diese Wissensflüsse werden durch eine große Anzahl variierender informeller Flüsse überlagert. Alle zusammen formen die Wissensflüsse eines Unternehmens und damit sein Identität.

Mit Shark wird der Austausch von Wissen auf Basis von Themen und Interessen definiert und nicht – wie bei den meisten aktuellen Lösungen – auf der Basis von speziellen Prozessen oder gar auf der Dokument-Ebene. Shark hebt die Definition der Wissensflüsse auf eine semantische Ebene und kann ein Beitrag sein, Menschen von der Last befreien,

bei der täglichen Arbeit darüber nachdenken zu müssen, ob zum Thema der Arbeit denn wirklich alle Dokumente beisammen sind. Menschen reden dann von Themen und Interessen. Die Technik, namentlich die Shark-Engine, stellt sicher, dass alle in der Gruppe genau das Wissen haben, was sie wollen, bzw. dass zwischen zwei Menschen auch alles Wissen übertragen wurde, was übertragen werden sollte.

5 Kontextualisierung mit Topic Maps

Im letzten Kapitel wurde eine Referenzarchitektur vorgestellt. In ihr wurden technische Details nicht diskutiert. Zur Erstellung einer Architektur ist das aber vonnöten. In diesem Kapitel sollen Topic Maps für die Wissensbasis eingesetzt werden.

Shark beschrieb Anforderungen an die Wissensbasis. Es muss möglich sein, Wissen anhand von Themen zu strukturieren, und es muss möglich sein, aus den Themen Interessen zu formulieren. Weiterhin wurde die Kontextualisierung definiert, die erlaubt, Kontext auf Wissen bzw. andere Kontexte anzuwenden. In Shark wird Kontext als spezielles Wissen verstanden. Das erzwingt, dass eine Sprache zur Wissensrepräsentation, die in Shark eingesetzt wird, sowohl das Wissen als auch den Kontext ausdrücken kann, wenn Kontext wahlweise auch die Rolle von Wissen spielen können soll.

Außerdem muss die Sprache das Konzept der aggregierten Mikrotheorien unterstützen.

In diesem Kapitel werden Topic Maps vorgestellt. Sie sind ein ISO-Standard und eine Wissensrepräsentationssprache und können der Kategorie der semantischen Netze zugeordnet werden. Sie sind damit ein Beispiel für die letzte, bisher nicht diskutierte prinzipielle Möglichkeit, Wissen zu repräsentieren (siehe Abschnitt 2.2.4). Es wird zu sehen sein, dass das Konzept des Kontextes nicht ausreicht für Shark, dass insbesondere keine direkte Unterstützung für die Kontextualisierung existiert. Aus diesem Grund wird basierend auf einem kleinen mathematischen Modell eine Funktion der Kontextualisierung definiert werden. Das erfolgt über mehrere Zwischenstufen, in denen vier weitere Funktionen über Topic Maps definiert werden.

Das Kapitel ist einfach strukturiert. Zunächst wird der Standard vorgestellt. Es werden für Shark relevante Ansätze gezeigt, die auf dem Standard basieren. Danach widmet sich das Kapitel der Definition der neuen Funktionen. Am Ende kann, spezialisiert durch die Topic Map Kontextualisierung, der Wissensaustausch zwischen Shark-Engines konkreter beschrieben werden. Diesmal ist es auch möglich, die Extraktion und Assimilation in einer einfachsten Version auf Funktionen über Topic Maps zurückzuführen.

Dieses Kapitel ist der letzte Schwerpunkt der Arbeit neben der Analyse des Knowledge Managements und der Referenzarchitektur Shark. Danach wird ein Knowledge Exchange Protocol definiert und weitere Aspekte einer Architektur. Daran schließen sich Anwendungsszenarien für Implementierungen an.

5.1 Topic Maps

Anfang der 1990er Jahre wurde erkannt, dass die Interoperabilität zwischen Thesauri, Indizes, Stichwortverzeichnissen usw. sowohl einige theoretische, aber vor allem praktische Probleme zeigte. Es gab weder standardisierte Sprachen noch Protokolle. Im Jahre

1991 begann daher unter der Leitung von Steve Newcomb die Arbeit an der Initiative *Standard Open Formal Architecture for Browsable Electronic Documents (SOFABED)* im Rahmen der Davenport-Group an der Erarbeitung von Verbesserungsvorschlägen für diese Situation. Ein geschichtlicher Abriss der Arbeiten an Topic Maps soll hier nicht gegeben werden, es sei dazu auf [Ham03] verwiesen.

Die erste Version des Standards wurde im Jahre 2000 als ISO Standard 13250 Topic Maps präsentiert. Topic Maps basierten dort auf der *Hypermedia/Time-based Structuring Language (HyTime)* [N1997], die ihrerseits auf SGML [ISO86b] basiert. SGML setzte sich für praktische Anwendungen nicht durch, dafür wuchs die Popularität von XML. Daher bildete sich etwa zeitgleich mit der Publikation des ISO-Standard ein Konsortium namens *TopicMaps.org*, die 2001 eine Topic Map Repräsentation mittels XML vorlegten: XML Topic Maps ([PM01]). Diese Arbeiten flossen später in den ISO-Standard ein.

Seit der Veröffentlichung des ersten Standards bis heute gibt es zwei Entwicklungsstränge an dem Standard. Der eine arbeitet an einer theoretischen Basis für Topic Maps. Der andere entstand in *Topicmaps.org*, war eher praktisch orientiert und arbeitete an Datenmodellen und Anwendungskonzepten. Um beide Arbeitsstränge wieder zusammenzuführen und um gleichzeitig eine bessere Struktur zu erzeugen, wurde beschlossen, den ISO Standard in mehrere Teile aufzugliedern. Zentrale Bestandteile sind das *Topic Map Referenze Modell (TMRM)* und das *Topic Map Data Model (TMDM)*. Derzeit liegt ein Draft für TMRM in der Version 5 vor ([DN05]) und eine finale Version von TMDM ([GM05]).

Weitere Bestandteile des Standards sind eine Topic Map Query Language (TMQL) und eine Topic Map Constraint Language (TMCL). Beide sind noch in der Diskussion. TMCL wird in dieser Arbeit nicht näher betrachtet. Sie dient der Beschreibung von Einschränkungen, die für spezielle Topic Maps gelten. Damit kann z. B. spezifiziert werden, ob und welche Beziehungen zwischen Klassen von Topics erlaubt sind.

Das Referenz-Modell und TMQL werden im Folgenden kurz vorgestellt. Das Datenmodell bildet allerdings die Basis für die Vorstellung der Konzepte von Topic Maps und wird auch Basis aller weiteren Überlegungen dieses Kapitels sein.

Neben den Standard wurde ein API entwickelt, die derzeit als Quasi-Standard für die Programmierung von Topic Maps gilt: die Topic Map API (TMAPI, siehe www.tmapi.org).

5.1.1 Grundkonzepte

Im Folgenden sollen die zentralen Konzepte von Topic Maps vorgestellt werden. Alle Definitionen sind dem Datenmodell [GM05] entnommen.

Das Modell der Topic Maps bewegt sich zwischen zwei gegensätzlichen Polen: dem Subjekt (*Subject*) und der Informations-Ressource (*Information Resource (IR)*).

Ein Subjekt ist definiert als „anything whatsoever, regardless of whether it exists or has any other specific characteristics, about which anything whatsoever may be asserted by any means whatsoever. In particular, it is anything on which the creator of a topic map chooses to discourse.“ Eine Informationsressource ist „a resource that can be represented as a sequence of bytes, and thus could potentially be retrieved over a network“. Obwohl letzte Definition allgemeiner gefasst ist, kann zur Vereinfachung ein beliebiges Dokument

auf einem Rechner mit einer URI vorgestellt werden.

Ein Subjekt ist per Definition alles, worüber sich irgendjemand auf der Welt aus beliebigen Gründen entschieden hatte nachzusinnen. Diese Definition deckt sich mit der Beschreibung des *Begriffes* im semiotischen Dreieck (Abschnitt 2.2.1). Die zwei anderen Ecken dieses Dreiecks waren der Gegenstand, aus dem Beobachter einen Begriff bilden, und die Bezeichnung, die dem Gegenstand bzw. dem Begriff gegeben wird.

Der Standard unterscheidet zwei Arten von Subjekten. Wenn ein Subjekt eine Informations-Ressource ist, wird es als adressierbares Subjekt (*Addressable Subject*) bezeichnet, als nicht-adressierbares Subjekt (*Non-addressable Subject*) im anderen Fall. Die Benennung erscheint logisch. Auf ein adressierbares Subjekt kann verwiesen werden. Es kann eine Adresse in einem Netzwerk haben.

Ein nicht-adressierbares Subjekt ist schwerer zu fassen. Beispiele sind *Liebe*, *Sowjetunion* und *Hamlet*. Das erste ist ein Konzept, für das es die unterschiedlichsten Vorstellungen gibt, die sich ggf. sogar verändern. Die Sowjetunion gibt es zwar nicht mehr, doch ist sie nach wie vor ein Gegenstand der Diskussion und hat Auswirkungen bis in die Gegenwart. Hamlet ist eine fiktive Figur in einem Drama. Es kann aber auch das Drama von William Shakespeare selbst bezeichnen. Nicht-adressierbare Subjekte zeigen die gleichen Probleme, die bei der Diskussion um das semiotische Dreieck identifiziert wurden. Da auf diese Subjekte nicht direkt verwiesen werden kann, können sich unterschiedliche Beobachter unterschiedliche Begriffe machen und gleiche Bezeichnungen für unterschiedliche Gegenstände benutzen oder unterschiedliche Bezeichnungen für das Gleiche. Das Problem lässt sich durch eine Referenz beheben, auf die verwiesen werden kann. Im realen Leben gibt es diese im Allgemeinen nicht. In Topic Maps nennt sie sich Subjekt Indicator (*Subject Indicator*, SI). Ein SI ist „an information resource that is referred from a topic map in an attempt to unambiguously identify the subject of a topic to a human being. Any information resource can become a subject indicator by being referred to as such from within some topic map, whether or not it was intended by its publisher to be a subject indicator.“ Zur Beschreibung eines Subjektes können beliebig viele SIs benutzt werden. So können Beschreibungen unterschiedlicher Sprachen, in unterschiedlichem Detailierungsgrad usw. benutzt werden. Sie sollten sich natürlich nicht widersprechen, aber das kann maschinell nicht sichergestellt werden. Wird ein SI anderen Personen zugänglich gemacht, so wird von einem *Published Subject Indicator* (PSI) gesprochen.

Dieses Verhältnis ist in der Abbildung 5.1 am Beispiel Hamlets illustriert. Im oberen Teil sind drei Subjekte dargestellt, Hamlet (die Figur), Hamlet (das Drama) und William Shakespeare. Alle Drei sind nicht adressierbar. Sie werden aber mittels PSIs beschrieben. Der Rest der Abbildung erschließt sich im Folgenden.

Wenn ein Subjekt ein *Begriff* in Topic Maps ist, so ist das Topic die *Bezeichnung*. Ein Topic ist „a symbol used within a topic map to represent some subject, about which the creator of the topic map wishes to make statements“. Erklärtes Ziel des Standards ist die Nutzung dieser Symbole, also der Topics in Computern zur Organisation und Repräsentation von Wissen. Ohne eine große Beschränkung kann sich bereits an dieser Stelle unter einem Topic eine Datenstruktur vorgestellt werden, die digital gespeichert und durch Rechner bearbeitet werden kann.

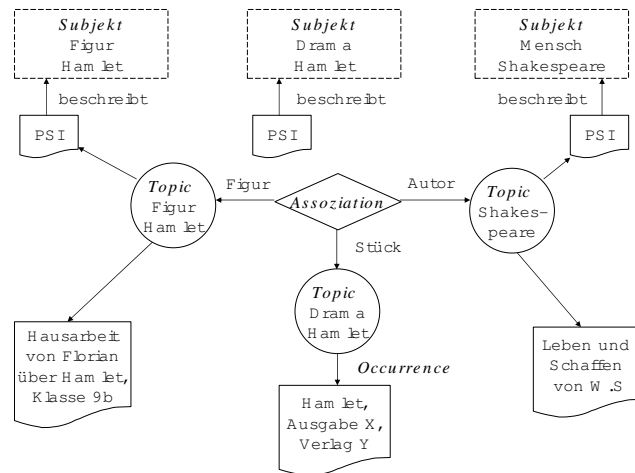


Abbildung 5.1: Beispiel: Eine Hamlet Topic Map

In der Abbildung sind Topics als Kreise dargestellt. Die Beziehung zu den Subjekten wird mittelbar durch die SI (oder sogar PSIs in der Abbildung) hergestellt. Die Pfeile vom Topic zum SI stehen für einen Subjekt Identifier (*Subject Identifier*): „a locator that refers to a subject indicator“. Der Übersicht wegen ist der Subjekt Identifier des Topic Hamlet, das Drama, nicht dargestellt.

In der Abbildung sind nur nicht-adressierbare Subjekte dargestellt. Adressierbare unterscheiden sich in der Darstellung lediglich dadurch, dass sie mittels einer Adresse (*Subject Address*) auf genau eine Informations-Ressource verweisen würden, die sie repräsentieren. Nach dem Standard sollte ein Topic explizit auf ein Subjekt verweisen, muss es aber nicht. Implizit muss es das aber tun. Laut Standard gilt, dass ein Topic immer nur für *ein* Subjekt stehen darf und nicht für eine Vielzahl.

Natürlich sollen auch Beziehungen zwischen Subjekten beschrieben werden können. Dazu dienen Assoziationen (*Associations*) („a representation of a relationship between one or more subjects“). Assoziationen und Topics ähneln sich in einigen Aspekten. Beide sind Repräsentanten (des Subjektes bzw. der Beziehung zwischen Subjekten) und beide werden schlussendlich durch eine Datenstruktur in einem Rechner repräsentiert.

In der Abbildung verbindet eine Assoziation die drei Topics. Dabei sind einige Details zu beachten: Die Assoziation repräsentiert die Beziehung der Subjekte. Die Subjekte und damit die Topic selber haben *keine Kenntnis* über ihre Beziehung untereinander. Daher sind die Pfeile bewusst von der Assoziation zu den Topics gezeichnet. Es wird davon gesprochen, dass Topics *Rollen* in einer Assoziation *spielen*. Die Assoziation in der Abbildung repräsentiert die Beziehung zwischen Hamlet, dem Drama, der Figur Hamlet und Shakespeare. Das Erste ist ein *Stück*, dessen *Autor* Shakespeare ist und in dem Hamlet eine *Figur* ist. Es soll noch einmal hervorgehoben werden, dass die Assoziation

die Beziehung zwischen Subjekten repräsentiert. Das erfolgt in Topic Maps mittelbar über die Topics. Eine Assoziation steht aber *nicht* für eine Beziehung zwischen Topics.

Mit diesen Konzepten ist Topic Maps bereits eine Sprache, die es erlaubt, Ontologien zu beschreiben. Die Topics sind Repräsentanten von Subjekten. Das Erscheinen eines Topics postuliert das Vorhandensein eines Subjektes. Beliebige Beziehungen zwischen Subjekten können durch Assoziationen beschrieben werden.

Topic Maps erlaubt aber auch, weitere Beziehungen zwischen Subjekten und Informations-Ressourcen zu beschreiben, dazu dient die *Occurrence*. Sie ist „a representation of relationship between a subject and an information resource“. Ohne dem Modell zu viel von seiner Allgemeinheit zu nehmen, kann man sich unter Informations-Ressourcen Dokumente im WWW vorstellen. Eine Occurrence beschreibt damit, dass ein solches Dokument (die IR) irgendetwas mit einem Subjekt zu tun hat. In der Abbildung wurde beispielhaft eine Informations-Ressource referenziert, die das Drama beschreibt. Vielleicht ist es eine literaturhistorische Schrift, die willkürlich ein paar fiktive Beispiele erstellt. So kann es eine Hausarbeit von einem Florian (ehemals Klasse 9b)¹ zum Thema Hamlet. Es kann den Text zu Hamlet als Buch geben, eventuell sogar als PDF. Es kann eine Webseite zum Leben und Schaffen von William Shakespeare geben. Occurrences erlauben Topic Map Autoren solche Informations-Ressourcen an Subjekte (mittelbar über Topics) zu binden, siehe Abbildung 5.1.

5.1.2 Topic Maps - persönliche Ontologien

Beachtenswert am Standard ist die Tatsache, dass immer wieder die Rolle der Autoren einer Topic Map betont wird. So werden Subjekte dadurch definiert, dass sich jemand dafür interessiert, was immer es sein mag. Assoziationen stellen Beziehungen zwischen Subjekten aus der Sicht der Autoren dar usw.² Diese Tatsache soll in Abbildung 5.2 noch einmal herausgestellt werden.

Die Abbildung stellt ein Individuum dar, das sich von zwei Gegenständen einen Begriff gemacht hat, nämlich von Hamlet, dem Drama und Shakespeare. In der Abbildung ist das durch die Wolke dargestellt. Bewusst wird von *Begriffen* gesprochen, denn Menschen erzeugen anhand ihrer Beobachtung eines Gegenstandes einen Begriff.

Topic Maps sind ein Mittel, um *Begriffen* eine *Bezeichnung* zu geben und damit über sie kommunizieren zu können. Zu diesem Mittel hat der Autor gegriffen, um seine Begriffe und deren Beziehungen zu modellieren. In der Abbildung wurden zwei Topics namens Hamlet und Shakespeare und zusätzlich eine Assoziation angelegt, die die Beziehung der Subjekte beschreibt.

Unserem Autor stehen aber zusätzlich eine Fülle von Informations-Ressourcen, vor allem über das WWW zur Verfügung. So wird es auch Beschreibungen zu Hamlet und

¹Diese Beispiel soll eher als Negativbeispiel gelesen werden. Eine heute übliche Freitextsuche „im Internet“ würde eine große Anzahl Referenzen auf Dokumente zum Thema Hamlet liefern. Eine inhaltliche Bewertung können diese Maschinen nicht geben. So können sich schulische Ausarbeitung durchaus mit literaturhistorischen Schriften und anderem mischen. Autoren von Topic Maps entscheiden, welche Occurrences sie definieren und entscheiden damit auch über die fachliche Tiefe und Ausrichtung.

²Es ist tatsächlich verwunderlich, dass es kein explizites Konzept der Benennung von Autoren in Topic Maps gibt.

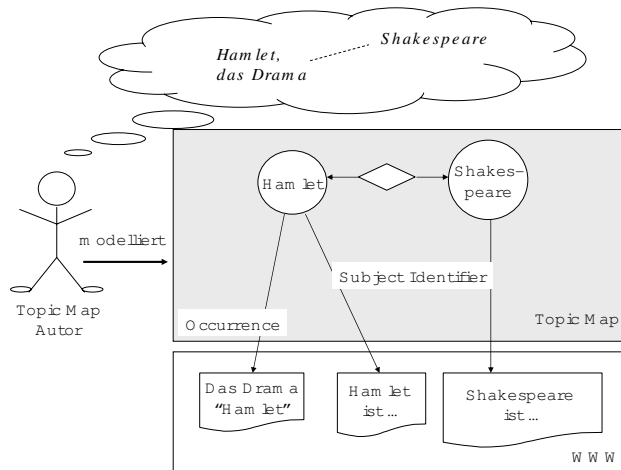


Abbildung 5.2: Beziehung Topic Map und Autor

Shakespeare geben. Diese können zur Beschreibung der Topics als SIs benutzt werden. In der Abbildung sind sie durch Subjekt Identifier referenziert. Außerdem ist exemplarisch eine weitere Informations-Ressource Hamlet gebunden worden.

Die Topic Map in der Mitte der Abbildung stellt drei Dinge dar.

1. Sie enthält zwei Symbole (wie Topics im Standard auch genannt werden) für zwei Begriffe. Die Begriffe sind durch referenzierte Subjekt Indikatoren näher beschrieben. Die Wahl dieser SIs erfolgte individuell durch den Autor. Es ist nicht sicher, dass andere diese Beschreibungen verstehen oder sie mit ihren eigenen Vorstellungen in Übereinstimmung bringen können.
2. Die Assoziation beschreibt die Beziehung zwischen den Begriffen, die aus Sicht des Autors besteht. Auch hier wird in keiner Form eine objektive Richtigkeit angenommen.³
3. Die Occurrence beschreibt wiederum eine persönliche Sicht darauf, was zu einem Subjekt gehört, die keinen Anspruch auf Allgemeingültigkeit haben kann⁴.

Die Bezeichnungen der Begriffe und die Modellierung ihrer Beziehungen formt eine Ontologie. Die Occurrence erlaubt zusätzlich, den Begriffen (Subjekten) mittelbar durch die Topics Informations-Ressourcen zuzuordnen. Die Topic Map erlaubt damit auch die

³Man kann in einer Topic Map problemlos formulieren, dass Elvis Presley von Außerirdischen entführt wurde und glücklich auf einem anderen Planeten wandelt.

⁴Es ist durchaus problemlos möglich, kabbalistische Texte zur Erlangung der endgültigen Wahrheit einem Subjekt zuzuordnen, das Lesenswertes für Neuimmatrikulierte vereinen soll.

Strukturierung von Informations-Ressourcen, wie es für Content Management Systeme üblich ist.

Die Topic Map in der Mitte der Abbildung ist ein Mittel der Kommunikation. Andere sollen anhand einer Topic Map ein Verständnis über Begriffe und damit Gegenstände und deren Beziehungen erlangen und gleichzeitig erfahren, welche Informations-Ressourcen den Begriffen zugeordnet werden können. Dabei ist die Topic Map aber wie eine Meinungsäußerung zu verstehen. Sie wurde von einer Person oder einer Gruppe von Personen erstellt und stellt deren Sicht auf die Welt dar. Diese muss der Empfänger nicht verstehen oder teilen.

Topic Maps folgen durch die konsequente Betonung der persönlichen Sichten dem Ansatz der Mikrotheorien (Abschnitt 2.2.3). Die Funktion *Merge*, siehe weiter unten, erlaubt die Zusammenführung (Aggregation) von Mikrowelten.

Weitere Konzepte

Namen Topics sind die einzigen Konzepte in Topic Maps, die benannt werden können. Konkret können sie beliebig viele Namen tragen, es wird dabei von *Base Names* gesprochen. Das kann z.B. sinnvoll sein, um Topics in mehreren Sprachen zu benennen.

Ein einzelner Basename kann je *Scope*, siehe weiter unten, Varianten haben. So kann es beispielsweise sinnvoll sein, einen Namen in einer Langversion (z. B. „James T. Kirk“), in einer kurzen Variante („Kirk“) z. B. für kleine Bildschirme und in einer Variante für alphabetische Sortierung bereitzustellen („Kirk, James T.“).

Typen Assoziationen, Rollen in Assoziationen und Occurrences können einen Typen haben. Als Typen werden Topics benutzt. Es gibt nur ein Kriterium, das ein Topic zu einem Typen macht: seine Nutzung als solcher.

Scopes Scopes beschreiben Gültigkeitsbereiche von folgenden Konzepten in Topic Maps: Namen und Occurrences von Topics und Assoziationen. Gültigkeitsbereiche können z. B. zeitliche Gültigkeiten sein.

Reification Es ist möglich, ein Konstrukt in einer Topic Map als Subjekt zu benutzen. Das kann u. a. sinnvoll sein, um zu beschreiben, wer welches Konstrukt angelegt hat. Angenommen, in einer Topic Map T wurde ein Dokument D an ein Subjekt S mittels einer Occurrence in Verbindung gesetzt. Mittels der Reification kann dieser Fakt näher beschrieben werden, z. B. dass Autor A diese Zuordnung vornahm.

Inline Occurrences Eine Occurrence ist ein Verweis ausgehend von einem Topic auf eine Informations-Ressource. Sie sagt aus, dass diese IR in irgendeiner Form zum Subjekt gehört.

In realen Anwendungen tritt oftmals die Situation auf, dass kleinere Werte, wie Identifikationsnummern, die sich aus einer Anwendung ergeben, Messwerte oder Ähnliches für ein Subjekt vermerkt werden sollen. Wird der „reinen Lehre“ des Standards gefolgt, so

müsste in einem System eine IR erzeugt werden, dort müsste der Wert eingetragen werden und aus der Topic Map heraus kann dieser Wert mittels einer Occurrence referenziert werden. In praktischen Anwendungen hat sich dieses Vorgehen als aufwendig erwiesen. Stattdessen können solche Werte auch direkt in die Topic Map eingetragen werden. Es erfolgt also kein Verweis hinaus auf eine IR. Die IR ist stattdessen bereits Bestandteil der Topic Map. Solche Informations-Ressourcen werden als *Inline Occurrences* bezeichnet.

Dieses Vorgehen bedeutet keine Abweichung von den genannten Konzepten des Standards. Auch Inline Occurrences sind ohne Abstriche Occurrences im Sinne von Topic Maps. Lediglich im Repräsentationsformat ist ein Unterschied zu „normalen“ Occurrences erkennbar.

Topic Characteristic Der Standard definiert den Begriff der *Topic Characteristic*, das ist die Menge aller Namen des Topics, seiner Occurrences und der Rollen, die es in Assoziationen spielt.

5.1.3 Gleichheit und Merge von Topics

Der Standard definiert für jedes Topic Map Konstrukt, wann zwei seiner Instanzen als gleich gelten. An dieser Stelle soll sich nur mit der Gleichheit von Topics beschäftigt werden und es soll auch nicht der Wortlaut des Standards wiederholt werden, aber dessen Grundidee.

Diese ist einfach und nicht verwunderlich: Zwei Topics gelten als gleich, wenn sie das gleiche Subjekt repräsentieren. Repräsentieren sie ein adressierbares Subjekt, ist diese Tatsache leicht zu ermitteln: Sind die Adressen der Subjekte gleich, gelten sie als gleich. Haben sie andere Adressen, können die Subjekte auch direkt miteinander verglichen werden. Ergibt der Bitvergleich beider Informations-Ressourcen, was adressierbare Subjekte sind, eine Gleichheit, so gelten die Topics auch als gleich.

Komplizierter wird es mit nicht-adressierbaren Subjekten. Ein direkter (binärer) Vergleich der Subjekte ist nicht möglich. Hier gilt: Zwei Topics gelten als gleich, wenn beide mittels Subjekt Identifiers auf wenigstens einen gemeinsamen Subjekt Indicator verweisen.

Diese Regel kann als die zentrale Regel der Ermittlung der Gleichheit angesehen werden, was auch die Bedeutung der Published Subject Indicator unterstreicht. Mengen von PSIs können als eine Art Enzyklopädie angesehen werden, auf die Topic Map Autoren verweisen können, wenn sie neue Topics anlegen und deren Subjekte definieren wollen. Darauf wird in einem Unterabschnitt noch einmal zurückgekommen werden.

Zwei Topic gelten auch dann als gleich, wenn all ihre Eigenschaften (*Topic Characteristics*) gleich sind. Diese Regel kann mehr als eine Heuristik bezeichnet werden, denn als ein Test auf Gleichheit der Subjekte. Es wird aber angenommen, dass zwei Topics das gleiche Subjekt repräsentieren, wenn sie in allen Eigenschaften übereinstimmen.

Die genannten Regeln werden im Standard als minimale Vorgaben bezeichnet, in dem Sinne, dass anwendungsspezifisch weitere Regeln definiert werden können.

Als gleich identifizierte Topics können gemischt werden. Diese Operation nennt sich *Merge* und ist ebenfalls im Standard beschrieben. Es soll kurz erklärt werden: Während

eines Merges werden alle Eigenschaften zweier Topics zusammengelegt und die ursprünglichen Topics werden gelöscht. Das resultierende Topic hat demnach alle Namen der ursprünglichen Topics, alle Occurrences und alle Subject Identifier. Es spielt außerdem alle Rollen in Assoziationen, die eines der ursprünglichen Topics gespielt hat. Repräsentiert ein Topic ein adressierbares Subjekt, verfügt das resultierende Topic über *eine* Adresse dieses Subjektes. Es ist nicht definiert, welches genommen werden soll.

Der Standard definiert auch das Mischen zweier Topic Maps. Das erfolgt einfach dadurch, dass eine Topic Map angelegt wird, die Kopien aller Konstrukte aller Quell-Topic Maps hat. Danach erfolgt ein Mischen der Topics, die gleich sind. Mittels der *Duplicate Suppression* können Duplikate erkannt und entfernt werden. Duplikate sind z. B. zwei Assoziationen, deren exakt gleiche Rollen durch identische Topics gespielt werden. Für Details sei auf den Standard verwiesen.

PSI Sets

In Abschnitt 2.2.3 wurde der Ansatz der aggregierten Mikrowelten dem Ansatz des verteilten Weltwissens gegenübergestellt. Es wurde gezeigt, dass es im Wesentlichen zwei Ansätze gibt, wie Mikrotheorien aggregiert werden können, einmal kann eine übergeordnete Theorie definiert werden, auf die sich die jeweiligen Mikrotheorien beziehen, oder es können Funktionen (Mediatoren) definiert werden, wie eine Mikrotheorie in eine andere umgewandelt werden kann.

Topic Maps sind ein Mittel, um Mikrowelten zu beschreiben. Als Möglichkeit der Aggregation wird dem ersten Weg gefolgt. Das Konzept dazu nennt sich Published Subject Indicator (PSI). Ein PSI ist eine Informations-Ressource, die *möglichst unmissverständlich* versucht, ein Subjekt zu beschreiben. Diese Unmissverständlichkeit kann allein wegen der Unzahl möglichen Hintergrundwissens niemals realisiert werden. Ein perfekter PSI müsste in allen denkbaren Sprachen notiert werden und in einer Art und Weise, die exakt ist, aber auch von Laien und Experten auf dem Gebiet als verständlich und gleichzeitig richtig und exakt anerkannt wird. Der Versuch wird aller Voraussicht nach scheitern.⁵

Ein praktikabler Weg ist stattdessen, dass gleiche Subjekte in unterschiedlichen Sprachen und unterschiedlicher fachlicher Tiefe definiert bzw. erklärt werden und *alle* Erklärungen als Subject Indikatoren vom Topic benutzt werden. Je nach Hintergrundwissen kann der Anwender einen PSI nutzen.

Ein Beispiel soll das prinzipielle Vorgehen erläutern: OASIS hat eine PSI Set für Länder der Erde definiert [OAS02]. Dort sind die Länder in verschiedenen Sprachen genannt. Legt nun ein Autor, der nur Deutsch spricht, ein Topic an, das z. B. *Norwegen* repräsentiert, so findet er den Eintrag für Norwegen und kann ihn referenzieren. Macht ein nur englischsprachiger Autor das Gleiche, findet er den Eintrag für *Norway* und kann ihn ebenfalls referenzieren. Idealerweise sind beide Einträge gleich, siehe Abbildung 5.3.

⁵Zur Übung möge man versuchen – passend zum Einsteinjahr 2005 – das Subjekt *Allgemeine Relativitätstheorie* zu definieren. Randbedingungen: Es soll für einen nur japanisch sprechenden Quantentheoretiker genauso verständlich sein, wie für einen deutschen Landwirt, der seit seiner Schulzeit keinen Blick mehr in ein Physikbuch warf. Danach kann das PSI für *Vorbereitung von Lehmböden für den Dinkelanbau* für die gleiche Zielgruppe entwickelt werden.

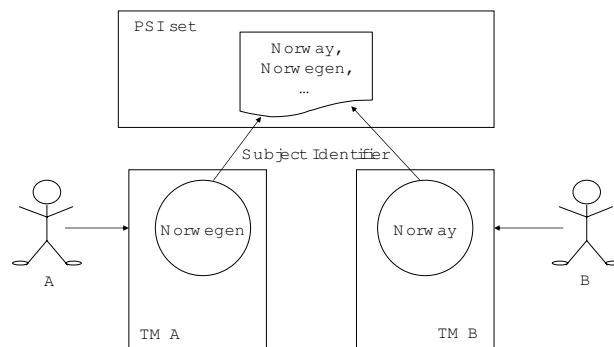


Abbildung 5.3: Nutzung PSI Set für Länder

In der Abbildung ist die beschriebene Konstellation dargestellt. A hat als deutschsprachiger Autor ein Topic namens „Norwegen“ angelegt, während der englischsprachige Autor ein Topic namens „Norway“ erzeugt hat. Namentlich existiert keine Gleichheit zwischen beiden. Glücklicherweise beziehen sich beide auf einen gemeinsamen PSI, der Länder in mehreren Sprachen benennt. Dank dieser Referenz kann ermittelt werden, dass beide Topics gleich sind.

Dieses Herangehen ist alles andere als neu. Die Idee, ein Vokabular oder gar eine *geschriebene* Sprache zu schaffen, die unabhängig von der *gesprochenen* Sprache von allen verstanden werden kann, wird als *Pasigraphie* bezeichnet. Es gibt eine Reihe von Projekten, die bis ins Mittelalter zurückreichen, siehe dazu [Eco94]. Nutzten Topic Map Autoren konsequent und ausschließlich mehrsprachige PSI Sets, um die Subjekte ihrer Topics zu definieren, so sind Topic Maps eine solche Pasigraphie.

Wie bereits in Abschnitt 2.2.1 gezeigt, werden Begriffssysteme und vor allem Thesauri benutzt, um Begriffen zu bezeichnen und zu strukturieren. Ein PSI Set kann auch mit einem Thesaurus verglichen werden. Beide haben das Ziel, eine Menge einheitlicher Bezeichnungen für eine Menge von Begriffen eines Diskursbereiches und deren Beziehungen untereinander zu definieren. Der wesentliche Unterschied liegt darin, dass die Bezeichnung in einem Thesaurus ein Wort (*Deskriptor*) ist, während die Bezeichnung in einer Topic Map das Topic ist, also ein Symbol, das seinerseits mehrere Namen haben kann. Trotzdem kann ein Thesaurus als ein PSI Set interpretiert werden, eben eines mit einer sehr speziellen Struktur. In [Ceb05, SC05] wurde ein System vorgestellt, dass es erlaubt, einen Thesaurus als ein PSI Set zu interpretieren und in ein LDAP-Verzeichnis abzulegen. LDAP ist ein bewährtes verteiltes Verzeichnissystem, das auch Replikation erlaubt. Das System erlaubt die räumliche Verteilung des Thesaurus. Damit kann an

räumlich unterschiedlichen Punkten ein Teil des Thesaurus gepflegt werden. Die Replikationsmechanismen von LDAP gewährleisten, dass die Änderungen propagiert werden. Es wird auch gezeigt, dass und wie dieser verteilte Thesaurus als PSI Set für die verteilte Erstellung von Topic Map benutzt werden kann.

In den letzten Jahren entstanden eine Reihe von Projekten, die Wörterbücher, Lexika und andere Nachschlagewerke für den privaten Anwender kostenfrei im WWW bereitstellten. Exemplarisch seien Wikipedia (www.wikipedia.de) und DMOZ genannt (www.dmoz.org). Solche Systeme erfüllen ebenfalls die Kriterien, die an ein PSI Set gestellt werden: Sie erklären nicht-adressierbare Subjekte, sind öffentlich über das WWW zugänglich und haben eine dauerhafte URLs. Diese Eigenschaften erlauben ihren Einsatz als Subject Indicator.

Das Vorgehen findet seine Grenzen, wenn zwei Topic Map Autoren kein gemeinsames PSI Set verwenden oder kein passendes existiert. Werden zusätzlich auch unterschiedliche Namen für die Topic benutzt und geben den Topics damit unterschiedliche Eigenschaften, gibt es keinen – durch den Standard definierten – Weg festzustellen, ob zwei Topics das gleiche Subjekt repräsentieren.

In [MW04, Mai04] wurde ein Ähnlichkeitsmaß über den *Topic Characteristics* definiert. Dabei ist die Grundidee des Konzeptes folgende. Wenn zwei Topic Map Autoren unabhängig voneinander das gleiche Subjekt durch ein Topic repräsentieren und Informationen dazu sammeln, ist es sehr unwahrscheinlich, dass sie genau den gleichen Namen und exakt die gleichen Occurrences und Assoziationen anlegen, in denen ein Topic eine Rolle spielt. Daher wird ermittelt, wie ähnlich die Topic Characteristics sind. Ist eine gewisse Schwelle überschritten, wird angenommen, dass die Topics das gleiche Subjekt repräsentieren. In [MS05] wurde gezeigt, dass und wie dieser Ansatz mit dem Konzept der Knowledge Ports in Shark zusammenpasst.

5.1.4 Reference Model

Das Referenzmodell hat während der Niederschrift dieser Arbeit die Version 5 ([DN05]) erreicht und bis dahin eine beachtliche Entwicklung durchgemacht. Es wurde immer kleiner und allgemeiner. In dieser Version scheint es sicher, dass es als ISO Standard angenommen wird.

Diese Version 5 soll kurz und der Vollständigkeit halber besprochen werden. Großer Raum wird nicht vonnöten sein, wurde das Modell doch im Vergleich zur Version 4 deutlich verallgemeinert. In der Vorgängerversion waren noch Konzepte, wie *Assertions*, das allgemeinere Konzept von *Assoziationen*, aus dem Datenmodell vorhanden. Außerdem verfügte es über detailliertere und formale Beschreibungen, die dadurch auch schlüssiger waren, was die Identität von Topic Map Konstrukten ausmachte.⁶

⁶An dieser Stelle soll vermerkt werden, dass durch die Reduktion des Referenzmodells von der Version 4 zur Version 5 das Modell Topic Maps an einigen Stellen an Klarheit und Exaktheit verloren hat. Soweit bekannt war der wesentliche Grund eine Beschleunigung des Standardisierungsprozesses und die Ausräumung von Unstimmigkeiten zwischen Referenz- und Datenmodell, was im Endeffekt zu einer massiven Reduzierung des Referenzmodells führte. Die Ziele sind vertretbar und es kann und soll auch nicht darüber geurteilt werden, was zu einem solchen Zusammenschrumpfen des Referenzmodells schlussendlich führte. Das Ergebnis hätte aber besser sein können.

Die Version 5 beschreibt lediglich noch sechs Konstrukte, die im Folgenden kurz erklärt werden sollen. Auch hier ist das *Subject* irgendetwas, von dem ein Topic Map Autor beschlossen hat, es zu notieren. Zur Notation wird ein *Subject Proxy* eingesetzt, was als eine Menge von *Eigenschaften* definiert wird. Die tatsächliche Struktur und eventuelle Einschränkungen der Eigenschaften werden durch eine *Topic Map Application (TMA)* definiert. Eine TMA ist das Topic Map Datenmodell, das den Subject Proxy als Topic definiert und die speziellen Eigenschaften Occurrence, Namen usw. beschreibt.

Ebenfalls wird das Konzept der *Subject Proxy Class* genannt, das ein abstraktes Modell einer Menge von Eigenschaften von Subject Proxys darstellt. Die konkrete Ausprägung dieser Klasse wird wiederum der TMA überlassen. Das letzte Konzept ist die Topic Map selbst, die definiert ist als „A set of subject proxies as defined and governed by one or more Topic Map Applications“.

Es existiert ein mathematisches Modell was detailreicher und klarer als die verbale Beschreibung im zukünftigen Standard darstellt und mit aller Wahrscheinlichkeit Teil des Referenzmodells im ISO-Standard wird, siehe [BS05].

Im Rest der Arbeit wird sich, wenn nicht anders betont, auf das Datenmodell bezogen, wenn vom *Topic Map Standard* die Rede ist.

5.1.5 Topic Map Query Language

Zum Zeitpunkt des Schreibens gibt es keinen verabschiedeten Standard für eine Anfragesprache über Topic Maps. An der Topic Map Query Language (TMQL) wird gearbeitet und es existiert ein *Working Draft* ISO [GB05]. Doch die Prinzipien der Sprache sind bereits klar zu erkennen und es ist nicht absehbar, dass an diesen grundlegende Änderungen vollzogen werden. Daher soll auf TQML kurz eingegangen werden.

Die Sprache ähnelt üblichen Anfragesprachen von Datenbanken, wie SQL. Ein wesentliches Element der Sprache ist das **Select** Statement, mit dem eine Anfrage formuliert wird. Mittels anderer Parameter kann beschrieben werden, welche Ontologie, d. h. welche Topic Map zu verwenden ist und welche Eigenschaften die gesuchten Topic Map Konstrukte haben sollen. Wie auch in Datenbanken üblich, kann so z. B. nach allen Topics mit einem bestimmten Namen gesucht werden.

Als Rückgabe einer TQML-Anfrage kann wahlweise ein einfacher Inhalt (*Simple Content*), wie Namen eines Topics usw., aber auch ein *Topic Map Content* zurückgegeben werden, also Topic Map Konstrukte in ihrer Gesamtheit – Letzter als Referenz oder als Kopie.

Quellen der Topic Map Query Language sind proprietäre Topic Map Anfragesprachen, wie AsTMa ([Bar03]) und TOLOG ([Gar03]).

5.1.6 Wissen und Kontext in Topic Maps

Wissen ist etwas, das zu einer qualifizierten Handlung befähigt, siehe Definition 4 im Modell Shark. Topic Maps sind erklärtermaßen eine Sprache der Wissensrepräsentation [GM05]. Wo ist das Wissen in Topic Maps verborgen?

Wie in vielen Sprachen der Wissensrepräsentation ist auch in Topic Maps der Begriff des Wissens nicht geklärt. Implizit kann vermutet werden, dass unter Wissen die Kenntnis über die Subjekte und deren Beziehungen untereinander und zu Informations-Ressourcen verstanden wird. Das ist die übliche Ansicht in der KI, in der bereits Ontologien als Wissen bezeichnet werden, ohne dass an irgendeiner Stelle gezeigt wurde, dass damit irgendetwas gemacht werden kann, geschweige denn etwas, das nach irgendwelchen Kriterien als etwas Sinnvolles bezeichnet werden könnte.

Kernkonzepte von Shark sind das Wissen und der Kontext. Erstes dient der Befähigung zu einer Handlung (eines Menschen), Zweites dient allem, was hilft, mit Wissen umzugehen (Definition 6 in Shark).

Topic Maps sind schlussendlich Datenstrukturen. Sie erlauben damit nur die Verwaltung expliziten Wissens, das in beliebigen Dokumente gespeichert werden kann. Das einzige Konzept in Topic Maps, das zur Aufnahme beliebiger Contents geeignet ist, sind die Informations-Ressourcen. Aus der Sicht von Shark werden Topic Maps daher wie folgt interpretiert:

Explizites Wissen ist in Informations-Ressourcen gespeichert. Aus Sicht von Shark bleibt die Speicherung des expliziten Wissens opaque, im Sinne, dass keine Eigenschaft der Informations-Ressource das Modell beeinflusst oder von ihm beachtet werden muss. Topics und die durch Assoziationen gebildeten Netze von ihnen werden als Kontext des expliziten Wissens interpretiert. Occurrences stellen die Verbindung zwischen Kontext und explizitem Wissen her. Subjekt Indikatoren sind Hilfsmittel, um ein gemeinsames Verständnis über Kontext zwischen unterschiedlichen Menschen zu erlangen, die mit explizitem Wissen arbeiten.

5.1.7 Verteilte Topic Map Engines

TMShare

Kal Ahmed stellte 2003 eine „P2P-Topic Map System“ namens TMShare vor ([Ahm03]). Der Ansatz orientiert sich stark an den damals in ihrer Blüte stehenden P2P-Filesharing-Tools.

Die Grundidee ist folgende: Wenn in einer Topic Map Engine Wissen nachgefragt wird (im Ansatz werden TOLOG-Anfragen als Beispiel benutzt), so können zwei Situationen entstehen: Entweder das Wissen liegt vor oder nicht.

TMShare erweitert eine Topic Map Engine dahingehend, dass sie Kenntnis über die Existenz anderer TMEs hat. Sie ist also in der Lage, eine Anfrage an diese Engines weiterzuleiten und die dabei ermittelten Ergebnisse lokal zu speichern. Daraus ergibt sich ein Netz von TMEs, die wechselseitig Kopien ihres Wissen speichern und damit auf Anfragen übermitteln können.

Peers agieren lediglich als Zwischenspeicher von Kopien von Wissen und Weiterleiter von (unbefriedigten) Anfragen. Es besteht keine Möglichkeit, Hierarchien von Peers aufzubauen noch die Wissensflüsse in irgendeiner Form thematisch zu strukturieren.

Shark verbirgt die Verteilung seiner Peers nicht, im Gegenteil. Wenn alle Bezeichner offen gelegt sind, ist klar erkennbar, welche Bezeichnung das Peer trägt und in welchen

Rollen es agiert. TMSHare versucht im Gegensatz dazu Verteilungstransparenz zwischen den Topic Map Engines herzustellen. Es kann daher eher von einer verteilten Topic Map Engine, denn von einem P2P-System, im Sinne von ACP oder Shark, die Rede sein.

Remote Access Protocol

Eine neuere Entwicklung, die in ähnliche Richtung geht, ist das *Remote Access Protocol* ([PG04]). Der Ansatz ist im Prinzip der gleiche, wenn er auch bisher in der Theorie noch nicht so weit ausgebaut ist wie TMSHare. Das Protokoll erlaubt den entfernten Zugriff einer Topic Map Engine auf eine andere. Damit soll vor allem für Webanwendungen eine Verteilungstransparenz zwischen den TMEs hergestellt werden. Es gelten die gleichen Kommentare wie für TMSHare.

Federated Topic Maps

Robert Barta entwickelte mit *AsTMA* [Bar03] eine Anfragesprache für Topic Maps und benutzte diese Sprache als Teil einer Föderation von Topic Maps ([Bar04]). Dieser Ansatz kommt Shark deutlich näher als die gerade gezeigten verteilten Topic Map Engines.

Der Topic Map Standard stellt einen direkten Zusammenhang zwischen dem Feststellen der Gleichheit von Konzepten und dem Mischen her. In Vorläufern des ISO-Standards wurden gar gefordert, dass nach Feststellung der Gleichheit ein Mischen der gleichen Konzepten stattfinden muss ([PM01]). Das führte zu so paradoxen Situationen, dass wenn in einer Anwendung zwei unterschiedliche Topic Maps geöffnet wurde, diese automatisch gemischt wurden, wenn die Gleichheit von Konzepten festgestellt wurde. Ein separates Ändern der einzelnen Topic Maps ist dann nur mit großer Mühe oder gar nicht möglich.

Barta argumentiert gegen dieses Vorgehen und betont, dass Topic Maps als private Ontologien anzusehen sind und dass der Standard weniger häufig ein Mischen erzwingen sollte. Sie sollten separat verwaltet werden, aber föderativ zusammengefügt werden können. Aus technischer Sicht stellt eine Föderation eine Menge von Topic Map Engines dar, die wechselseitig aufeinander zugreifen können. Der Zugriff erfolgt mittels der Anfragesprache AsTMA.

Das Konzept folgt der Idee von aggregierten Mikrowelten, wobei mittels AsTMA die Mediatoren definiert werden, die eine Überführung von Wissen von einer TME in die andere erlauben. Allerdings schafft die vorgestellte Form der Föderation im Endeffekt auch wieder nur Verteilungstransparenz.

Obwohl die Wichtigkeit der Personen in dem Prozess betont wird, fließt sie nicht in die technische Realisierung ein. Es besteht keine Möglichkeit, Hierarchien von TMEs zu definieren, die die Beziehungen der Personen untereinander darstellen können. Es existiert kein Nutzer- oder Gruppenkonzept. Der Ansatz erzeugt wiederum lediglich Verteilungstransparenz. Im Vergleich zu TMSHare basieren die föderierten Topic Maps dank AsTMA auf einem deutlich stabileren theoretischen Fundament. Beide stellen eine verteilte Topic Map Engine dar.

5.2 Topic Maps als Mengen und Graphen

Im Folgenden soll ein mathematisches Modell für das Topic Map Datenmodell (TMDM) entworfen werden. Das Modell soll dazu genutzt werden, eine Möglichkeit der Realisierung der Funktion *Kontextualisierung*, die im Modell Shark postuliert wurde, zu definieren.

Die Grundidee der Realisierung besteht darin: Topic Maps können als Graphen aufgefasst werden, wobei die Knoten Topics sind und die Kanten durch Assoziationen bestimmt werden. Da Assoziationen beliebige viele Topics miteinander verknüpfen können, handelt es sich um Hypergraphen. Es werden eine Reihe von Funktionen definiert werden (Selektion, View und Korona), die über Topic Maps arbeiten. Diese werden zusammengefasst werden zu einer Funktion namens *Fragmentierung*. Es wird zu sehen sein, dass die Parameter der Fragmentierung selber eine Topic Map darstellen und dass daraus die Kontextualisierung abgeleitet werden kann. Die Kontextualisierung erlaubt die Anwendung einer Topic Map auf eine andere mit den im Kapitel 4 definierten Eigenschaften. Dabei können willkürlich Topic Maps genommen werden. Das entspricht exakt dem Credo, das diese Arbeit durchzieht, dass Wissen und Kontext nicht trennbar, sondern im Gegenteil austauschbar sind. Diese Art der Kontextualisierung realisiert diesen Anspruch und unterscheidet sich dadurch, z. B. von der Topic Map Query Language, aber auch von Ansätzen für Kontextbeschreibungssprachen, wie z. B. CXML ([BDS02]).

Das Modell basiert vor allem auf einer Reihe von Mengendefinitionen und einige Operationen über den Mengen. Nebenbei wird noch ein kleines Graphenmodell entwickelt, das zwar nicht direkt für die Definition der Kontextualisierung notwendig ist, aber helfen soll, die Essenz von Topic Maps verständlich zu machen.

Inspiziert wurde es von einem Modell von Kipp [Kip03], das für das Topic Maps Referenzmodell (TMRM) entworfen, aber mit der Version 5.0 obsolet wurde, da die Semantik des Modells deutlich abgeschwächt wurde.

Kipp definiert eine Topic Maps wie folgt:

$$M_{Kipp} = (T, R, A, C, Z, I)$$

Dabei gilt:

- T ist die Menge der Typen,
- R ist die Menge der Rollen,
- A ist die Menge der Assertions,
- C ist die Menge der Castings,
- Z ist die Menge der in Beziehung stehenden Elemente,
- I ist die Menge der unabhängigen Elemente.

Typen und Rollen ergeben sich aus der Beschreibung im Kapitel 5. Assertions bezeichnet die Verbindung zwischen zwei Topics im Referenzmodell bis zur Version 4. Ab der

Version 5 gibt es keine Überlegungen mehr dazu im Referenzmodell. Auf Assertions wird nicht mehr eingegangen werden, einmal weil sie nicht mehr Bestandteil eines Standards sind und zum anderen werden sie für dieses Kapitel nicht benötigt. Man kann sich unter Assertions etwas allgemeinere Assoziationen des Datenmodells vorstellen.

Castings existieren nicht im Datenmodell. Sie beschreiben den Zusammenhang zwischen Rollen und Typen.

Die Menge Z umfasst alle Topics der Topic Map, die mit anderen Topics mittels einer Assertion in einer Beziehung stehen. Die Menge I ist das Komplement davon bezüglich aller Topics in der Topic Map.

Kipp definiert zusätzlich eine Menge X, mit der im Folgenden Typen definiert werden können. Es wird zu sehen sein, dass auch in diesem Modell X eine wesentliche Funktion haben wird:

$$X_{Kipp} = T \cup R \cup A \cup C \cup Z$$

Für das Datenmodell kann diese Definition angepasst werden. Da das Konzept der Castings nicht existiert, kann C entfallen. Im Datenmodell existiert stattdessen das Konzept der Scopes, die den Gültigkeitsbereich von Assoziationen und Occurrences einschränken können. Scopes sind keine Topics, sie werden aber durch eine Menge von Topics definiert. Die Kipp'schen Definitionen können für das Datenmodell wie folgt umgestellt werden:

$$M = (T, R, A, S, Z, I)$$

Aus der Menge X entfallen aus dem gleichen Grund die Castings. Die Assoziationen werden im Datenmodell ebenfalls nicht als Topics interpretiert, im Gegensatz zu den ehemaligen Assertions im Referenzmodell.

$$X = T \cup R \cup Z$$

In diesem Kapitel soll daher eine Topic Map definiert werden durch

$$M = (X, A, R, T, S, I)$$

Die Reihenfolge der Mengen wurde in der letzten Definition geändert. Das hat nur darstellerische Gründe; im Folgenden werden einige Definitionen übersichtlicher.

Typen und Scopes

Im Datenmodell existiert kein impliziter Zusammenhang zwischen Typen und anderen Konzepten. Topics, Assoziationen, Occurrences können einen beliebigen Typen haben.

Es lässt sich damit eine Funktion definieren, die einem Topic einen Typen zuordnet.

$$t : T \rightarrow X$$

Jedes typbare Element darf nur einen Typen haben. Daher gibt es eine eindeutige Funktion, die ein Topic auf einen Typen abbildet:

$$t^{-1} : X \rightarrow T$$

Ein Scope kann den Gültigkeitsbereich einer Assoziation, einer Occurrence oder von Topicnamen einschränken. Die Scopes werden von den Topic Map Autoren vergeben und stehen in keinem funktionalen Zusammenhang zu anderen Elementen der Topic Map. Es lässt sich daher eine Funktion s definieren, die eine Zuordnung von Scopes zu beliebig vielen Elementen der gerade genannten Menge vornimmt. Das Modell in diesem Kapitel ignoriert die Namen und Occurrences von Topics. Deshalb kann einschränkend eine Funktion s definiert werden:

$$s : S \rightarrow A$$

Auch hier gilt, dass jedes Element, das einen Scope haben kann, nur maximal einen Scope besitzt. Daher kann eine eindeutige Funktion definiert werden, die eine Assoziation auf einen Scope abbildet:

$$s^{-1} : A \rightarrow S$$

Im Datenmodell definieren Assoziationen die Beziehungen zwischen Topics. Man spricht von Rollen spielenden Topics. Eine Rolle kann von beliebig vielen Topics gespielt werden, aber wenigstens von einem. Der Gültigkeitsbereich einer gesamten Assoziation kann durch einen Scope eingeschränkt werden.⁷

Daraus ergibt sich eine Beschreibung einer Assoziation:

$$A \subseteq 2^{(R \times X)}$$

Es soll eine boolesche Funktion definiert werden, anhand der bestimmt werden kann, ob eine Assoziation in einem Scope gültig ist oder nicht:

$$valid(a, s) = \left\{ \begin{array}{l} true \Leftrightarrow a - valid - in - s \\ false \end{array} \right\}$$

Ausgehend von den bisherigen Überlegungen sollen einige Eigenschaften von Topic Maps mit Graphen beschrieben werden.

5.3 Connectivity Graphs

Zu Beginn sollen einige Definitionen erfolgen.

Definition 16 (Direkt erreichbar) *Zwei Topics heißen direkt erreichbar, wenn sie Rollen in derselben Assoziationen spielen.*

Die Erreichbarkeit kann weiter verfeinert werden, indem beachtet wird, ob zwei Topics unterschiedliche oder gleiche Rollen spielen. Dazu dienen folgende Definitionen:

⁷In der Vorgängerversion konnten Scopes sogar die einzelnen Rollen in einer Assoziation einschränken. Das ist im aktuellen Datenmodell entfallen und macht die Modellierung deutlich leichter.

Definition 17 (Verwandt) *Zwei Topics sind verwandt, wenn sie in derselben Assoziation die gleiche Rolle spielen.*

Definition 18 (Bekannt) *Zwei Topics sind bekannt, wenn sie in derselben Assoziation unterschiedliche Rollen spielen.*

Die verwandten Topics einer Assoziation und bezüglich einer Rolle lassen sich in einer Menge zusammenfassen. Diese Menge soll das Symbol X_{ar} erhalten und gelesen werden als *Verwandtschaftsmenge bezüglich r in a*. Es gilt:

Definition 19 (Menge der direkten Verwandten) $x \in X_{ar}$ genau dann, wenn x die Rolle r in der Assoziation a spielt.

Die Bekanntschaft zwischen Topics einer Assoziation lässt sich, daraus abgeleitet, als Kreuzprodukt über den Mengen der verwandten Topics definieren. Die Menge X_a sei dabei die Menge aller Topics mit einer Bekanntschaftsbeziehung:

Definition 20 (Menge der direkten Bekanntschaften)

$$X_a = X_{ar_1} \times X_{ar_2} \times \dots \times X_{ar_m}$$

wobei m die Anzahl der Rollen in der Assoziation a ist.

Abbildung 5.4 stellt drei Assoziationen dar. Die Assoziation a_1 ist im Scope s_1 gültig, a_2 in s_2 und a_3 in s_3 . In a_1 spielen x_1 bis x_4 Rollen, wobei x_1 und x_2 die gleiche Rolle r_1 spielen. x_1 und x_3 haben den gleichen Typ. In a_3 sind nur zwei Rollen spielende Topics.

In der Abbildung gilt Folgendes bezüglich der gerade vorgenommenen Definitionen:

Verwandschaft x_1 und x_2 sind verwandt, da sie die gleiche Rolle in a_1 spielen. Damit ergeben sich folgende Mengen direkter Verwandtschaft: $X_{a_1r_1} = \{x_1, x_2\}$, $X_{a_1r_3} = \{x_3\}$, $X_{a_1r_4} = \{x_4\}$, $X_{a_2r_5} = \{x_5\}$, $X_{a_2r_6} = \{x_6\}$, $X_{a_2r_7} = \{x_7\}$, $X_{a_3r_8} = \{x_6\}$ und $X_{a_3r_9} = \{x_7\}$

Bekanntschaft x_1 , x_3 und x_4 sind bekannt, genauso wie x_4 , x_5 und x_6 . Sie spielen jeweils in der gleichen Assoziation unterschiedliche Rollen. Damit ergeben sich folgende Mengen direkter Bekanntschaft:

$$X_{a_1} = X_{a_1r_1} \times X_{a_1r_3} \times X_{a_1r_4} = \{(x_1, x_3, x_4), (x_2, x_3, x_4)\}.$$

$$X_{a_2} = X_{a_2r_5} \times X_{a_2r_6} \times X_{a_2r_7} = \{(x_4, x_5, x_6)\}.$$

$$X_{a_3} = X_{a_3r_8} \times X_{a_3r_9} = \{(x_6, x_7)\}.$$

X_a lässt sich als Kante eines Graphen interpretieren. Das soll in folgender Definition geschehen.

Definition 21 (Connectivity Graph einer Assoziation a (C_a)) Eine Topic Map sei $TM_Q = (X_Q, A_Q, R_Q, T_Q, S_Q, I_Q)$.

Dann ist $C_a = (V, E)$ mit

$$V = X_Q$$

$$E = X_a.$$

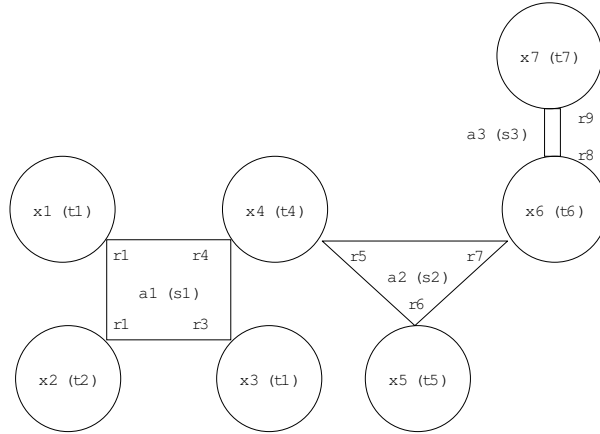


Abbildung 5.4: Beispiel Topic Map

Die Kanten des Connectivity Graph verbinden Topics, die in einer Assoziation miteinander verwandt sind. Die Anzahl der Rollen in a und die Anzahl der Rollen spielenden Topics in a bestimmen, wie viel Topics eine Kante von C_a verbinden. Im Allgemeinen ist C_a ein Hypergraph. Ein Graph entsteht dann, wenn in a nur zwei Rollen existieren oder wenn höchstens zwei Spieler unterschiedliche Rollen spielen.

Diese Überlegungen lassen sich auf eine gesamte Topic Map ausdehnen:

Definition 22 (Connectivity Graph (C)) Eine Topic Map sei

$$TM_Q = (X_Q, A_Q, R_Q, I_Q, T_Q, S_Q).$$

Dann ist $C = (V, E)$ mit

$$V = X_Q \text{ und}$$

$$E = \bigcup_{a=1}^m X_a : m = |A_Q|$$

Der Connectivity Graph einer Topic Map ist ein Hypergraph. Abbildung 5.5 stellt den Connectivity Graphen zur Topic Map aus Abbildung 5.4 dar. Der wesentliche Unterschied zwischen beiden Darstellungen besteht darin, dass die Assoziation $a1$ in zwei Hyperkanten zerfallen ist. Das ergibt sich daraus, dass in $a1$ die Topics $x1$ und $x2$ die gleichen Rollen spielen.

In der Definition des Graphen wurden Kanten, ausgehend von der Bekanntheit von Topics, definiert und nicht allgemein von der direkten Erreichbarkeit. Die Verwandtschaft wurde als sekundär eingestuft. Grund ist eine Interpretation der Semantik von Rollen in einer Assoziation. Die Interpretation ist folgende:

Rollen beschreiben eine Funktion eines Topics in einer Assoziation. Spielen Topics unterschiedliche Rollen, kann eher von einer UND-Beziehung ausgegangen werden. Alle

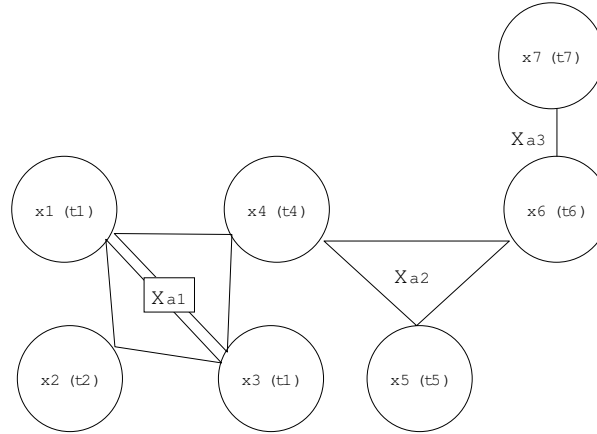


Abbildung 5.5: Connectivity Graph

Topics, der unterschiedlichen Funktionen tragen etwas zu der Verbindung bei. Spielen zwei Topics aber die gleiche Rolle, wird diese eher als eine ODER-Beziehung verstanden. Topics, die die gleiche Rolle innehaben, können als Alternativen angesehen werden. Aus dieser Interpretation heraus ist es konsequent, dass sobald mehr als ein Topic eine Rolle spielt, diese Assoziation als eine Kombination von Alternativen aufgefasst wird. Genau das erfolgte durch die Definition des Connectivity Graphen und wird sich noch einmal bei der Definition der Korona und damit indirekt bei der Definition der Kontextualisierung wiederholen. Der Standard hält sich mit der Interpretation dieses Sachverhaltes sehr zurück. Die hier erfolgte Interpretation verschärft den Standard etwas.⁸

Bei der Definition der Graphen wurden die Scopes außer Acht gelassen. Diese sollen wieder beachtet werden, durch folgende Definition.

Definition 23 (Scoped Connectivity Graph (C_s)) Eine Topic Map sei

$$TM_Q = (X_Q, A_Q, R_Q, I_Q, T_Q, S_Q)$$

und s sei ein Scope.

Dann ist $C_s = (V, E)$ mit

$$V = X_Q \text{ und}$$

$$E = \bigcup_{a=1}^m X_{av} : m \text{ Anzahl der Assoziationen der Topic Map und}$$

⁸Im Referenzmodell bis zur Version 4 war es nicht möglich, dass zwei Topics die gleiche Rolle in einer Assoziation spielen. Vielmehr ergab sich die Rolle aus einem so genannten Casting, das die Assoziation, deren Typen und den Typen der Topics mit betrachtete. Diese Definition war zwar auf den ersten Blick schwerer zu durchschauen, aber inhaltlich klarer, als das jetzige Referenzmodell, das sich damit nicht mehr beschäftigt. Das Datenmodell erlaubt mehrere Topics, die die gleiche Rolle spielen, trägt aber nicht zur Klärung der unscharfen Semantik bei.

$$X_{av} = \left\{ \begin{array}{c} X_a \Leftrightarrow \text{valid}(a, s) \\ \emptyset \end{array} \right\}$$

Wege und Erreichbarkeit

In Graphen existiert der Begriff des Weges. In den Connectivity Graphen, die Hypergraphen sind, ist ein Weg zwischen zwei Topics x_1 und x_n eine Menge folgender Form:

$$P = \{(x_1, X_{a1}), (X_{a1}, x_2), ((x_2, X_{a2}), \dots, ((X_{an}, x_{n+1}))\}$$

Die Elemente der Menge alternieren ganz offensichtlich zwischen Knoten (Topics) und Kanten (Bekanntschafsmengen). So beschreibt das erste Element, dass von x_1 die Kante X_{a1} „betreten“ werden soll. Das zweite Element beschreibt, dass die Kante bei x_2 „verlassen“ werden soll usw. bis x_n . Die Länge des Weges ist n , was der Anzahl der passierten Kanten entspricht.

Analog kann ein Weg zwischen zwei Kanten beschrieben werden:

$$P = \{(X_{a1}, x_1), ((x_1, X_{a2}), (X_{a2}, x_2), \dots, (x_n, (X_{an+1}))\}$$

Hier beginnt der Weg mit einer Kante. Über verschiedene Knoten wird eine Kante X_{an+1} erreicht. Die Länge des Weges entspricht hier der Anzahl der passierten Knoten, also n .

Zwei Knoten bzw. Kanten gelten als *erreichbar*, wenn es einen Weg zwischen ihnen gibt. Der *kürzeste Weg* in einem Graphen ist jeweils der Weg mit dem kleinsten n .

Der Weg zwischen zwei Knoten soll mit $P(x_1, x_2)$ beschrieben werden. Die Länge des Weges soll mit $|P(x_1, x_2)|$ gekennzeichnet sein. Der kürzeste Weg zwischen x_1 und x_2 soll durch $P_{min}(x_1, x_2)$ symbolisiert werden. Analog gilt das für den Weg zwischen Kanten.

Es sollen zwei Mengen im Graphen C definiert werden.

Definition 24 (Nachbarn n-ten Grades) Die Menge von Knoten $X_{x,p=n}$ heie „Menge der benachbarten Knoten n-ten Grades“. Es gelte:

$$y \in X_{x,p=n} \Leftrightarrow \exists P(y, x) \wedge |P_{min}(y, x)| = n$$

Die Menge $X_{x,p=n}$ enthlt die Nachbarn eines Knotens. Es soll die Menge $(X_{X,p=n})$ definiert, werden, die die Nachbarn einer Menge von Knoten umfasst:

$$X_{X,p=n} = \bigcup X_{x,p=n} \forall x \in C$$

Die Menge von Kanten $E_{e,p=n}$ heie „Menge der benachbarten Kanten n-ten Grades“. Es gilt:

$$b \in E_{e,p=n} \Leftrightarrow \exists P(b, a) \wedge |P_{min}(b, a)| = n$$

Die Menge $E_{e,p=n}$ enthlt die Nachbarn einer Kante. Es soll die Menge $(E_{E,p=n})$ definiert, werden, die die Nachbarn einer Menge von Kanten umfasst:

$$X_{E,p=n} = \bigcup E_{e,p=n} \forall e \in C$$

Einige Beispiele sollen die Definitionen erläutern, siehe dazu auch Abbildung 5.5. Dort gilt:

- Alle Knoten und Kanten sind untereinander erreichbar. Man sagt auch, dass der Graph *nicht zerfällt*.
- $P(x2, x1) = 1$, $P(x2, x4) = 2$, $P(x2, x6) = 3$, $P(x2, x7) = 4$
- $X_{x2,p=1} = \{x1, x3\}$, $X_{x2,p=2} = \{x4\}$, $X_{x2,p=3} = \{x6\}$, $X_{x2,p=4} = \{x7\}$, $X_{x2,p=5} = \emptyset$

5.4 Operationen

Auf diesem Modell können einige Funktionen definiert werden.

5.4.1 Merge

In [Kip03] wurde eine Operation definiert, das Merge. Die Definition ist trivial und soll für das Datenmodell adaptiert übernommen werden:

$$merge : (TM \times TM) \rightarrow TM$$

$$merge(J, K) = L \Leftrightarrow (J_X \cup K_X = L_X) \wedge (J_A \cup K_A = L_A) \wedge (J_R \cup K_R = L_R)$$

$$\wedge (J_T \cup K_T = L_T) \wedge (J_S \cup K_S = L_S) \wedge (J_I \cup K_I = L_I)$$

Die Operation erschließt sich schnell. Es werden die jeweiligen Mengen vereinigt und in einer neuen Topic Map zusammengefasst.

Es sollen einige weitere Funktionen definiert werden.

5.4.2 Selektion

Die Selektion soll eine Menge von Topics und Assoziationen aus einer Topic Map auswählen und aus ihnen eine neue (und üblicherweise kleinere) Topic Map erzeugen. Vor der formalen Definition soll die Selektion in Prosa kurz erläutert werden:

Die Selektion dient der Erzeugung einer Topic Map (selektierte Topic Map) aus einer anderen Topic Map (Quelle). Die Selektion erfolgt mittels einer Menge von Topics bzw. Assoziationen (Selektor). Diese beschreiben, welche Topics oder Assoziationen in der selektierten Topic Map vorkommen. Sind diese Topics oder Assoziationen in der Quelle, so liegen sie auch in der selektierten Topic Map vor.

Wird eine Menge von Topics als Selektor angegeben, so liegen auch alle Assoziationen in der selektierten Topic Map, die in der Quelle vorlagen und deren Rollen spielenden Topics *alle* im Selektor liegen. Anders gesagt: Assoziationen der Quelle, die wenigstens

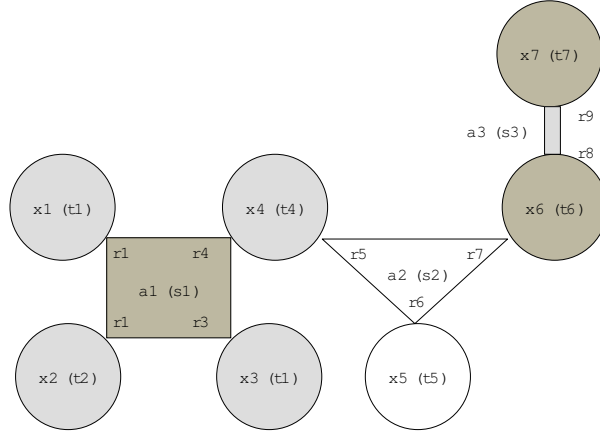


Abbildung 5.6: Selektion

einen Spieler einer Rolle haben, der *nicht* im Selektor liegt, sind *nicht* in der selektierten Topic Map.

Definition 25 (select)

$$\text{select}(TM_Q, X_{SP}, A_{SP}) = (X_S \cup X_{SP}, A_S \cup A_{SP}, R_Q, T_Q, S_Q, I_Q) = TM_S$$

mit $x \in X_S \Leftrightarrow (x \in X_Q \wedge \exists a \in A_{SP}, r : x \in X_{ar})$

und $a \in A_S$, wenn gilt:

- $a \in A_Q$
- für alle Rollen spielenden Topics x in a in TM_Q gilt: $x \in X_{SP}$

Zur Erläuterung: R, T, S und I werden bei der Selektion direkt aus der Quelle übernommen. Es unterscheiden sich lediglich die Mengen X und A.

Die neue Menge X setzt sich zusammen aus den Topics, die als Selektionsparameter angegeben wurden, und neu hinzukommenden. Neue Topics kommen hinzu, wenn sie folgende Bedingungen erfüllen:

- Sie müssen in der Quelle vorliegen ($x \in X_Q$).
- Sie müssen in einer Assoziation aus dem Selektor eine Rolle spielen ($\exists a \in A_{SP}, r : x \in X_{ar}$). Diese Regel sorgt dafür, dass Assoziationen, die im Selektor stehen, mit allen ihren Topics „aufgefüllt“ werden.

Für Assoziationen gilt Ähnliches. Die Selektoren werden in die selektierte Map übernommen. Es können weitere Assoziationen hinzukommen, wenn sie folgende Bedingungen erfüllen:

- Sie müssen in der Quelle vorliegen ($a \in A_Q$).
- Die Assoziation hat in der Quelle einige Rollen spielende Topics. Sie müssen *alle* in X_{SP} vorkommen.

Als Beispiel der Selektion soll die Beispiel-Topic Map von oben wieder aufgegriffen werden. Die Abbildung 5.6 illustriert das Ergebnis einer Selektion. Es wird angenommen, dass die Selektion mit folgenden Parametern aufgerufen wurde:

$$\text{select}(TM_Q, \{x6, x7\}, \{a1\})$$

Die Selektoren wurden in der Abbildung mit einem hellen Grau markiert. Die dunkler bezeichneten Elemente sind hinzugekommen. Die Assoziation a1 ist hinzugekommen, weil alle Rollen spielenden Topics im Selektor vorlagen. Die Topics x6 und x7 sind hinzugekommen, da die Assoziation a3 im Selektor vorlag.

Das Resultat ist

$$X_S = \{x1, x2, x3, x4, x6, x7\}, A_S = \{a1, a3\}$$

Es ist leicht zu sehen, dass Connectivity Graph der resultierenden Map zerfallen ist. Da die Assoziation a2 nicht selektiert wurde, gibt es z. B. keinen Weg mehr zwischen x6 und x7 zu x1, x2, x3 und x4.

Negative Selektion

Als Variante zur Selektion soll die negative Selektion definiert werden. In der ersten Variante wurde mit positiven Beispielen gearbeitet: Die Topics und Assoziationen wurden selektiert, die angegeben waren. Man kann natürlich auch umgekehrt vorgehen und nur die Topics und Assoziationen auswählen, die *nicht* in einer gegebenen Menge liegen. Es soll von der *negativen Selektion* die Rede sein. Die negative Selektion soll einfach als eine Art Komplement der positiven Selektion definiert werden:

Definition 26 (selectNot)

$$\text{selectNot}(TM_Q, X_S, A_S) = TM_S$$

mit $TM_S = (X_Q \setminus X_S, R_Q, A_Q \setminus A_S, I_Q, T_Q, S_Q)$

5.4.3 View

Aus Sicht der Connectivity Graphen ist die Selektion ein Herauslösen eines Untergraphen, wobei die Menge der Knoten (Topics) reduziert wird, aber die Anzahl der Kanten zwischen den Knoten erhalten bleibt.

Die Operation View erzeugt im Gegensatz dazu einen Teilgraphen. Es werden sowohl die Menge der Knoten (Topics) reduziert als auch die Kanten zwischen den Knoten. Die Reduktion der Kanten erfolgt anhand von Rollen: Nur die Topics sind im View, die sowohl in der Quelle vorhanden sind und zusätzlich eine Rolle in einer Assoziation spielen, die in einer gegebenen Menge von Rollen vorhanden sind. Die Assoziationen werden anhand der Scopes reduziert, in denen sie gültig sind.

Der View soll nun formaler definiert werden. View ist eine vierstellige Operation: $view(TM_Q, R_V, T_V, S_V) \rightarrow TM_V$ wobei TM_Q die Quell-Topic Map ist, TM_V die View-Topic Map. R ist eine Menge von Rollen, T eine Menge von Typen und S eine Menge von Scopes.

Der View ist definiert als:

Definition 27 (View)

$$view(TM_Q, R_{VP}, T_{VP}, S_{VP}) = (X_V, R_{VP}, A_V, I_S, T_{VP}, S_{VP}) = TM_V$$

mit

$$a \in A_V \Leftrightarrow (a \in A_Q) \wedge (\exists s \in S_{VP} : valid(a, s)) \wedge (\exists r \in R_{VP} : |X_{ar}| \geq 1)$$

$$x \in X_V \Leftrightarrow (x \in X_Q) \wedge t^{-1}(x) \in T_{VP}$$

Ein View erzeugt eine reduzierte Sicht auf eine Topic Map. Die Reduktion erfolgt anhand einer Menge von Rollen und Scopes. Assoziationen in TM_V müssen eine Reihe von Bedingungen genügen:

- Sie müssen in der Quelle vorliegen ($a \in A_Q$).
- Sie müssen in einem der angegebenen Scopes gültig sein ($\exists s \in S_V : valid(a, s)$).
- In ihnen muss wenigstens ein Topic vorliegen, das eine erlaubte Rolle spielt ($|X_{ar}| \geq 1 : r \in R_V$).

Topics sind in TM_V enthalten, wenn alle folgenden Bedingungen gelten:

- Sie müssen in der Quelle enthalten sein ($x \in X_Q$).
- Sie müssen einen passenden Typen haben ($t^{-1}(x) \in T_V$).
- Sie müssen in einer Assoziation eine erlaubte Rolle spielen ($\exists a : (x \in X_{ar} \wedge r \in R_V)$).
- Die Assoziation muss in TM_V sein ($a \in A_V$).

Aus der Sicht der Connectivity Graphen erzeugt der View einen Untergraphen. Es werden nur die Assoziationen ausgewählt, die einem Scope genügen und es werden nur die Topics ausgewählt, die eine passende Rolle und einen passenden Typen haben. Außerdem gilt, dass eine Assoziation nur dann übernommen wird, wenn sie wenigstens über ein Rollen spielendes Topic verfügt.

Negativer View

Definition 28 (Negativer View)

$$viewNot(TM_Q, R_{VP}, T_{VP}, S_{VP}) = TM_V$$

mit

$$\begin{aligned} TM_V &= (X_V, R_V \setminus R_{VP}, A_V, I_Q, T_V \setminus T_{VP}, S_V \setminus S_{VP}) \text{ und} \\ (x \in X_V &\Leftrightarrow (x \in X_Q) \wedge (t^{-1}(x) \notin T_{VP})) \\ (a \in A_V &\Leftrightarrow (a \in A_Q) \wedge (\forall s \in S_{VP} : not - valid(a, s)) \wedge (not \exists r \in R_{VP} : |X_{ar}| \geq 1)) \end{aligned}$$

Gleichheit von Assoziationen und Topics

Selektion und View basieren auf einem Test auf Gleichheit. Es soll Abschnitt 5.1.3 in Erinnerung gerufen werden, in dem die Gleichheit zwischen Topics diskutiert wurde. Die Gleichheit von Assoziationen wurde dort nicht ausführlich besprochen, hier sei auf den Standard verwiesen ([GM05]).

5.4.4 Korona

Es soll eine weitere Funktion eingeführt werden, die Korona.

Definition 29 (Korona für Topics)

$$coronaX(TM_Q, X_{CP}, R_{CP}, T_{CP}, S_{CP}) = X_C.$$

Es gilt: $x \in X_C$, wenn alle folgenden Bedingungen gelten:

- $x \in X_Q$
- $\exists a \in A_Q, y \in X_{CP}$ mit
 - $y \neq x$
 - x und y sind über a miteinander bekannt
 - x spielt eine Rolle r_x in a mit $r_x \in R_{CP}$
 - y spielt eine Rolle r_y in a mit $r_y \in R_{CP}$
- $\exists s \in S_{CP} : valid(a, s)$
- $t^{-1}(x) \in T_{CP}$
- $t^{-1}(y) \in T_{CP}$

Definition 30 (Korona für Assoziationen)

$$coronaA(TM_Q, A_{SP}) = A_C.$$

Es gilt: $a \in A_C$, wenn alle folgenden Bedingungen gelten:

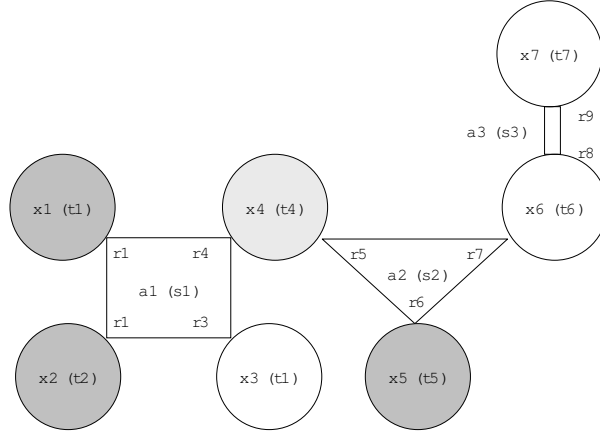


Abbildung 5.7: Korona

- $a \in A_Q$
- $\exists x \in X_Q, b \in A_{SP}$ mit
 - $b \neq a$
 - x spielt eine Rolle r_1 in a mit $r_1 \in R_{CP}$
 - x spielt eine Rolle r_2 in a mit $r_2 \in R_{CP}$
- $t^{-1}(x) \in T_{CP}$
- $\exists s_1 \in S_{CP} : \text{valid}(a, s_1)$
- $\exists s_2 \in S_{CP} : \text{valid}(b, s_2)$

Als Beispiel soll in der nunmehr bekannten Topic Map eine Korona gebildet werden. Der Einfachheit halber soll nur ein einziges Topic selektiert werden ($X_{SP} = \{x4\}$). Als gültige Rollen sollen angesetzt werden $R_{CP} = \{r1, r4, r5, r6\}$. Als gültige Scopes und Typen sollen alle in der Topic Map bekannten gelten: $S_{CP} = S_Q$ und $T_{CP} = T_Q$. Abbildung 5.7 illustriert den Vorgang.

Die Corona von $x4$ ist dunkel dargestellt und besteht aus drei Topics: $x1$, $x2$ und $x5$. Nur diese drei Topics erfüllen die notwendigen Bedingungen. $x3$ ist zwar auch mit $x4$ bekannt, allerdings spielt $x3$ die Rolle $r3$, die nicht in R_{CP} ist. Das Gleiche gilt für $x6$.

5.4.5 Fragmentierung

Die Selektion erlaubt die Bildung von Teilgraphen mittels Selektionsparameter Topics und Assoziationen. Die Korona erlaubt die Ermittlung von Nachbarn selektierter Topics und Assoziationen. Der View wiederum erlaubt die Filterung anhand von Einschränkungen über Rollen, Typen und Scopes.

Die Fragmentierung führt diese Konzepte zusammen und ist damit eine Verallgemeinerung der drei Funktionen. Sie ist wie folgt definiert:

Definition 31 (Positive Fragmentierung)

$$\begin{aligned} & \text{fragment}(TM_Q, X_{SP}, A_{SP}, R_{VP}, S_{VP}, T_{VP}, d) = TM_F \\ TM_F = & \left\{ \begin{array}{l} \text{view}(\text{select}(TM_Q, X_{SP}, A_{SP}), R_{VP}, S_{VP}, T_{VP}) \Leftrightarrow d = 0 \\ \text{fragment}(TM_Q, X_{SP}', A_{SP}', R_{VP}, S_{VP}, T_{VP}, d - 1) \Leftrightarrow d > 0 \end{array} \right\} \\ & \text{mit} \\ & X_{SP}' = X_{SP} \cup \text{coronaX}(TM_Q, X_{SP}, R_{VP}, S_{VP}, T_{VP}) \text{ und} \\ & A_{SP}' = A_{SP} \cup \text{coronaA}(TM_Q, A_{SP}, R_{VP}, S_{VP}, T_{VP}) \end{aligned}$$

Die Idee der Fragmentierung ist recht einfach, obwohl die Rekursion auf den ersten Blick abschrecken mag: Im Kern ist die Fragmentierung eine Selektion mit einem anschließenden filternden View. Wird $d > 1$ vereinbart, so werden die Selektionsparameter erweitert. Das erfolgt durch wiederholte Anwendung der Korona, wodurch die Parameterzahl potenziell mit jedem Schritt wächst. Zu beachten ist, dass die Korona die Gültigkeit der Typen, Rollen und Scopes beobachtet. Mit dem Abbruch der Rekursion wird eine Selektion mit einem anschließenden View durchgeführt. Ein kleines Beispiel über der Beispiel Topic Map mag die Fragmentierung illustrieren, siehe Abbildung 5.8.

Folgende Parameter wurden in dem Beispiel gewählt:

$$X_{SP} = \{x4\}, A_{SP} = \emptyset, R_{VP} = R_Q \setminus \{r6\}, S_{VP} = S_Q, T_{VP} = T_Q, d = 1$$

Mit dem Aufruf der Fragmentierung existiert zunächst nur ein Selektionsparameter: $x4$, in der Abbildung als heller Kreis dargestellt. Es wird die Korona durch den Aufruf von

$$\text{coronaX}(TM_Q, \{x4\}, R_Q \setminus \{r6\}, S_Q, T_Q)$$

gebildet. Das Ergebnis ist $\{x1, x2, x3, x6\}$ (dunkle Kreise). $x5$ ist nicht Bestandteil der Korona, da es die Rolle $r6$ spielt, und diese ist nicht erlaubt in dieser Fragmentierung. Nach der Bildung der Korona von $x4$ wird das Ergebnis mit $x4$ vereinigt. Es entsteht $X_{SP}' = \{x1, x2, x3, x4, x6\}$. CoronaA liefert keine weiteren Assoziationen, da nur eine leere Menge übergeben wurde, die keine Korona hat.

Nach diesem Aufruf ist die Rekursion bereits abgebrochen. Es wird aufgerufen:

$$\text{select}(TM_Q, \{x1, x2, x3, x4, x6\}, \emptyset, R_Q \setminus \{r6\}, S_Q, T_Q)$$

Durch die Selektion kommt die Assoziation $a1$ hinzu. Alle Topics dieser Assoziation sind selektiert. Die Assoziationen $a2$ und $a3$ kommen nicht hinzu, da jeweils ein Topic fehlt.

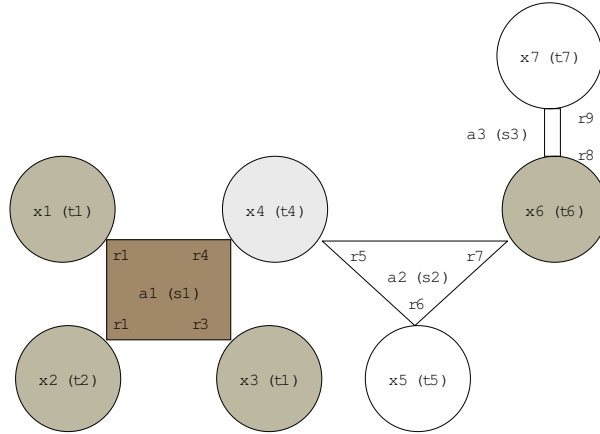


Abbildung 5.8: Fragmentierung

Der im Folgenden aufgerufene View filtert keine weiteren Ergebnisse. Es verbleibt die Topic Map, die in der Abbildung durch dunkle Elemente gekennzeichnet ist.

Es kann praktisch sein, anstelle von positiven Beispielen auch mit negativen zu arbeiten. Das soll durch die negative Fragmentierung ermöglicht werden.

Definition 32 (Negative Fragmentierung)

$$fragmentNot(TM_Q, X_{SP}, A_{SP}, R_{VP}, S_{VP}, T_{VP}, d) = TM_F$$

$$TM_F = \left\{ \begin{array}{l} viewNot(selectNot(TM_Q, X_{SP}, A_{SP}), R_{VP}, S_{VP}, T_{VP}) \Leftrightarrow d = 0 \\ fragmentNot(TM_Q, X_{SP}', A_{SP}', R_{VP}, S_{VP}, T_{VP}, d - 1) \Leftrightarrow d > 0 \end{array} \right\}$$

mit

$$X_{SP}' = X_{SP} \cup coronaX(TM_Q, X_{SP}, R_{VP}, S_{VP}, T_{VP}) \text{ und}$$

$$A_{SP}' = A_{SP} \cup coronaA(TM_Q, A_{SP}, R_{VP}, S_{VP}, T_{VP})$$

Das Vorgehen ist ganz ähnlich. Erst werden die Selektionsparameter rekursiv erweitert, wenn $d > 0$. Danach erfolgt allerdings eine negative Selektion, gefolgt von einem negativen View.

5.5 Kontextualisierung

Betrachtet man die Parameter der Fragmentierung genauer, so kann man feststellen, dass sie zusammen eine Topic Map definieren können. Das ist auch nicht verwunderlich. Um die größtmögliche Auswahl über Elemente in einer Topic Map treffen zu können, müssen über alle Einschränkungen gemacht werden können.

Die Kontextualisierung greift die Tatsache auf und fügt alle Parameter zu einer Topic Map zusammen. Was zunächst wie eine funktionale Vereinfachung aussieht, ist ein, wenn auch kleiner, qualitativer Sprung. Bis zur Fragmentierung bewegten sich die Funktionen auf einem Niveau einer Anfragesprache. Durch die Zusammenführung zu einer Topic Map bietet die Kontextualisierung plötzlich die Möglichkeit, Topic Maps direkt aufeinander anwenden zu können. Es ist nicht mehr nötig, Teile zu entnehmen, um sie als Parameter in eine Suchanfrage zu stecken. Begreift man Topic Maps als Wissensrepräsentations-sprache und setzt man voraus, dass eine Topic Map explizites Wissen enthalten kann, so besteht der qualitative Sprung einfach darin, dass nunmehr mit Wissen nach Wissen gesucht werden kann. Ein kleiner Schritt für die Technik...

Genug mit der Einleitung. Es gibt wie bei der Fragmentierung zwei Versionen. Eine arbeitet mit Positivlisten, die andere mit Negativlisten. Beide Varianten werden nacheinander definiert.

Definition 33 (Positive Kontextualisierung)

$$ctxPlus(TM_Q, TM_{C+}, d) = fragment(TM_K, X_{C+}, A_{C+}, R_{C+}, S_{C+}, T_{C+}, I_{C+}, d)$$

Analog dazu ist die negative Kontextualisierung definiert:

Definition 34 (Negative Kontextualisierung)

$$ctxNot(TM_K, TM_{C-}, d) = fragmentNot(TM_K, X_{C-}, A_{C-}, R_{C-}, S_{C-}, T_{C-}, I_{C-}, d)$$

Die Kontextualisierung wird nun als die Vereinigung beider Varianten definiert. Eine Diskussion dazu folgt weiter unten.

Definition 35 (Kontextualisierung 1)

$$ctx(TM_K, TM_{C+}, TM_{C-}, d) = \\ merge(ctxPlus(TM_K, TM_{C+}, d), ctxNot(TM_K, TM_{C-}, d))$$

Aus dieser Funktion lässt sich eine zweistellige Funktion definieren, indem die letzten drei Parameter in eine Topic Map eingefügt werden.

Das ist ohne Informationsverlust wie folgt möglich. Dazu wird TM_{C-} durch folgende Schritte in TM_{C-}' umgeformt:

- Alle Topics (das sind auch Rollendefinitionen, Typdefinitionen und Topics, die Subjekte repräsentieren) aus TM_{C-} erhalten den zusätzlichen Typen *negativer Kontext*.

- Scopedefinitionen erhalten zusätzlich den Typen *negativer Kontext* in ihrer Beschreibung.
- Jede Assoziation erhält zusätzlich den Scope *negativer Kontext*.

Außerdem wird eine weitere Topic Map (TM_D) erzeugt, die lediglich ein einziges Topic enthält. Eine Occurrence des Topics verweist auf den Wert von d . Das Topic hat den Namen d und repräsentiert das Subjekt

Nun kann ein einziger Parameter (TM_C) erzeugt werden:

$$TM_C = merge(TM_{C+}, TM_{C-}', TM_D)$$

TM_C enthält alle Informationen, um aus ihr die drei ursprünglichen Parameter zu extrahieren.

Diese Funktion soll definiert werden als

$$unwrap(TM_C) = (TM_{C+}, TM_{C-}, d)$$

Die zweistellige Funktion Kontextualisierung kann nun definiert werden als:

Definition 36 (Kontextualisierung)

$$contextualize(TM_K, TM_C) = ctx(TM_K, unwrap(TM_C))$$

Diskussion

Zugegeben, die letzte Argumentation, wie aus einer vierstelligen Funktion eine zweistellige gemacht wird, indem drei Parameter in einen fallen, um als erste Aktion des Aufrufes wieder alle drei ursprünglichen Elemente „auszupacken“, erscheint etwas konstruiert.

Dazu ein paar Punkte: Es ist richtig, dass das eine zunächst recht künstliche Konstruktion ist. Es ergibt sich durch das Zusammenfügen ein Erwartungskontext bezüglich Wissen nach dem Verständniss des Modells Shark. TM_{C+} repräsentiert eine positive Erwartung, TM_{C-} repräsentiert, was nicht gelten soll.

In den eher abstrakten Beschreibungen bei Shark war es allerdings nie notwendig, von einem negativen Kontext zu sprechen. Es ist zu vermuten, dass viele Anwendungen ausschließlich mit der Beschreibung eines positiven Kontextes auskommen. In dem Falle würde tatsächlich eine Topic Map ohne jede Änderung als Kontext interpretiert werden können.

Andere Anwendungen mögen zur Beschreibung eines Kontextes tatsächlich beide Aspekte benötigen. Das mag nur auf den ersten Blick ungewöhnlich erscheinen. In vielen Repräsentationssprachen für Ontologien (und im Standardfall auch bei Topic Maps) ist es nicht üblich, zu definieren, was *nicht* ist, sondern nur, was *ist*. Andere Sprachen bieten aber auch Konstrukte, die Nicht-Zusammenhänge und Nicht-Fakten beschreiben helfen.

So gibt es in DAML, dem Vorläufer von OWL, das Konstrukt, um zu beschreiben, dass eine Klasse *nicht* von einer anderen abgeleitet ist.

Das Vorgehen hier ist aber gerade so, dass eine Ontologie (oder Teile davon), die als positive Beschreibung erstellt wurden, als negativer Kontext interpretiert wird.

In der Praxis sieht das Vorgehen wie folgt aus. Autoren einer Topic Map haben eine Ontologie entworfen. Andere können diese Ontologie als Kontext interpretieren und auswählen, ob sie sie als positive oder negative Erwartung in den Kontext einfließen lassen. In beiden Fällen muss der Autor einer Topic Map nicht vorher beachten, ob seine Topic Map als Wissen oder als Kontext benutzt werden wird. Wissen und Kontext bleiben austauschbar.

Der Parameter *d* ist ein technischer Parameter. In realen Anwendungen ist zu erwarten, dass nicht Anwender direkt oder gar Ontologie-Autoren diesen Parameter setzen. Es ist eher zu erwarten, dass *d* je nach Anwendungsszenario von Administratoren oder *Wissens-Ingenieuren* entsprechend gesetzt wird.

Aus Sicht der Implementierung wird daher eher erwartet, dass *d* als eine Umgebungsvariable auftaucht und weniger Teil des Kontextes ist. Aber das sind Implementierungsspezifika.

Es soll noch einmal zusammengefasst werden, welche Rollen die Ontologien erstellen und nutzen:

- Topic Map-Autoren erstellen eine Topic Map. In anderen Worten, sie erzeugen eine Ontologie. In ihr formalisieren sie einen Diskursbereich.
- Suchende von Wissen nutzen die Ontologien aus. Sie wählen Konzepte, für die sie sich interessieren oder für die sie sich nicht interessieren (positive oder negative Themen oder Interessen). Sie nutzen dazu die immer positiv beschriebenen Ontologien aus. Auf diese Weise lassen sich Erwartungskontexte beliebiger Form darstellen.
- Die Kontextualisierung extrahiert mittels des so formulierten Kontextes passendes Wissen aus einer Wissensbasis.

Zu beachten ist dabei, dass der Kontext ursprünglich als Wissen beschrieben wurde und in keinem Schritt die Repräsentationsform ändern musste. Es wird nur regelmäßig anders interpretiert.

5.5.1 Eigenschaften der Kontextualisierung

In Abschnitt 4.2 wurden Anforderungen an die abstrakte Funktion Kontextualisierung gestellt. Hier soll gezeigt werden, dass die eben definierte Funktion diese Anforderungen erfüllt.

Neutrales Element

Es lässt sich zeigen, dass die leere Topic Map ein neutrales Element der Kontextualisierung ist. Das soll der Vollständigkeit wegen getan werden.

Tatsächlich gilt:

Satz 1 (Neutrales Element der Kontextualisierung ist die leere Topic Map)

Als Symbol für die leere Topic Map soll TM_\emptyset stehen. So kann der Satz formaler geschrieben werden:

$$\text{contextualize}(TM, TM_\emptyset) = TM$$

Der Nachweis lässt sich leicht führen. Es gilt: $\text{contextualize}(TM, TM_\emptyset) = \text{merge}(\text{fragment}(TM, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset, d), \text{fragmentNot}(TM, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset, d)) =$

Die positive Fragmentierung wird eine leere Menge liefern, unabhängig vom gewählten d . Da keine Selektionsparameter angegeben wurden, wird nichts selektiert werden. Umgekehrtes gilt für die negative Fragmentierung. Keine Selektionsparameter bedeutet hier, dass die gesamte Map sofort im ersten Schritt selektiert wird. Auch das ist unabhängig von d . Damit lässt sich die obige Funktion kürzen und das Ergebnis notieren:

$$\text{merge}(TM_\emptyset, TM) = TM.$$

Kritisch angemerkt werden muss allerdings, dass die Nutzung der leeren Topic Maps semantisch wenig sinnvoll ist. Verbal ausgedrückt bedeutet der Erwartungskontext in etwa: „Ich kenne erstmal nicht, was interessant wäre, aber ich würde auch nicht verwerfen, was mir angeboten wird.“ Undifferenzierter geht es kaum.

Weitere Eigenschaften

In Shark wurden neben der Existenz eines neutralen Elements noch weitere Forderungen an die Kontextualisierung gestellt. Diese sollen kurz gezeigt werden.

Satz 2 ($TM_\emptyset * TM = TM_\emptyset$) .

Der Beweis ergibt sich direkt aus der Definition der Selektion, die die initiale Funktion der Kontextualisierung ist. Aus einer leeren Topic Map kann nichts selektiert werden, weder positiv noch negativ. Daher werden auch die folgenden Operationen keine Elemente für das Ergebnis erzeugen. Eine Kontextualisierung einer leeren Topic Map erzeugt eine leere Topic Map.

Satz 3 ($TM_Q * TM_C = TM_E, TM_E$ ist kleiner als TM_Q) .

Im Abschnitt 4.2 wurde von einer Teilmenge gesprochen. Übersetzt kann von einem Teilgraphen gesprochen werden. Tatsächlich wurde gezeigt, dass Selektion und View Unter- bzw. Teilgraphen bezüglich der Connectivity Graphen erzeugen, was wiederum heißt, dass das Ergebnis der Operationen maximal die ursprüngliche Topic Map erzeugt, keinesfalls aber mehr. Damit kann das Ergebnis maximal genauso „groß“ sein, wie die Quelle.

Die beschriebene Kontextualisierung erfüllt damit die Bedingungen von Shark.

5.5.2 Vergleich Kontextualisierung mit TMQL

Sowohl die Kontextualisierung als auch TMQL wählen aus einer Topic Map Teile aus. Der wesentliche Unterschied zwischen beiden Ansätzen ist, dass TMQL eine eigene Sprache ist, während die Kontextualisierung einen Topic Map als einschränkenden Kontext *interpretiert* und entsprechend benutzt.

In der Sprechweise einer Anfragesprache gilt für die Kontextualisierung, dass Anfrage und angefragte Entität das gleiche Format haben. Das erlaubt auch, dass eine Anfrage Resultat einer Kontextualisierung ist. Die Kontextualisierung lässt sich damit nacheinander ausführen, ohne dass an irgendeiner Stelle ein Format gewechselt werden müsste. Es gibt kein technisches Hindernis dafür, dass Anfrage und Angefragtes die Position wechseln können. Das geht in TMQL natürlich nicht, denn sie beschreibt gerade eine völlig eigene Anfragesprache.

Ergebnis der Kontextualisierung ist immer eine Topic Map, konkret ein Fragment. Ergebnis einer TMQL-Anfrage ist entweder eine Menge von Topic Map Konstrukten, also auch ein Fragment oder einzelne Eigenschaften der Konstrukte, wie z. B. Namen. TMQL erlaubt damit den Zugriff auf Teile der Topic Map Konstrukte, was die Kontextualisierung nicht ermöglicht. TMQL könnte nicht als Realisierung der Shark-Kontextualisierung genutzt werden, wie sie in Abschnitt 4.2 beschrieben wurde.

Beide Ansätze sind auch nur bedingt vergleichbar. Während die Kontextualisierung das Ziel hat, Wissen zu selektieren, das zu einem Kontext passt, hat TMQL den Ansatz, aus Wissen Elemente herauszulösen.

5.6 Konkretisierung des Modell Shark

5.6.1 Interessen und Themen als Topic Maps

Topics und Assoziationen wurden als Kontext interpretiert, siehe Abschnitt 5.1.6. Damit wird davon ausgegangen, dass Themen und Interessen als spezielle Kontextarten als Topic Maps beschrieben werden können.

Topic Maps erlauben die Definition von persönlichen Ontologien und die Zuordnung von explizitem Wissen zu den jeweiligen Subjekten. Damit bilden die Netze aus Topics und Assoziationen Themen gemäß der Definition 7.

Interessen sind spezielle Themen, die eine veränderte Semantik haben. Sie bedeuten, dass eine Knowledge Node einen Austausch von Wissen zu diesen Themen wünscht (Definition 5). Eine veränderte Semantik ändert aber nichts an der Struktur. Interessen können ebenfalls mit Topic Maps ausgedrückt werden. So kann ein Empfangsinteresse z. B. als eine Topic Map ohne Occurrences gedacht werden bzw. ein Sendeinteresse als eine Topic Maps, die mittels Occurrences auf Wissen verweist, das gesendet werden darf.

5.6.2 Personalisierung / Lokalisierung von Topic Maps

Personalisierung und Lokalisierung wurden als eine spezielle Form der Kontextualisierung definiert. Diese Vorstellung kann an dieser Stelle konkretisiert werden, wenn man annimmt, dass Wissen in Form einer Topic Map gespeichert ist.

Personalisierung ist eine Kontextualisierung bei der der Kontext eine Person ist. Personalisierung kann generell zwei Ausprägungen haben. Einmal kann davon ausgegangen werden, dass in einer Wissensbasis die Person bekannt ist und alles Wissen ermittelt werden soll, das mit der Person zusammenhängt. Die zweite Interpretation ist die, dass eine Person eine Menge von Interessen hat und dass Wissen zu einem oder mehreren dieser Interessen in einer Wissensbasis vorkommt.

Üblicherweise wird von Personalisierung gesprochen, wenn Wissen *für* eine Person ermittelt werden soll und nicht, wenn Wissen *über* eine Person angeboten wird. Die zweite Interpretation ist damit die übliche.

Die Interessen einer Person lassen sich in einer Topic Map ausdrücken. Das kann im einfachsten Fall eine Menge von Topic sein (z. B. „Fußball, Autos, Motoryachten“), die nicht miteinander verknüpft sind. Das können aber auch beliebige Netze von Topic sein. Dieses Interesse kann nun kontextualisierend auf eine Topic Map angewandt werden, die Wissen enthält. Das Ergebnis einer Kontextualisierung ist ein Topic Map Fragment. Dieses Fragment wird als *personalisierte Topic Map* bezeichnet, denn es enthält nur das Wissen, das (direkt oder indirekt) zum Interesse der Person passt.

Für die Lokalisierung gilt genau das Gleiche. Ein Ort kann als Topic beschrieben werden. Es gibt PSI Sets für die Definition von Orten. Eleganter können auch ganze geografische Bereiche als Topic Map formuliert werden ([Ham03]). Egal, wie sie definiert sind, sie können auf eine Topic Map kontextualisierend angewandt werden, die Wissen enthält und ein Fragment erzeugen. Das Fragment wird als *lokalisierte Topic Map* bezeichnet.

Personalisierung und Lokalisierung können mit Knowledge Ports definiert werden. Mehr noch, Knowledge Ports drücken bereits den Wunsch nach Personalisierung aus. Sie beschreiben die Interessen einer Knowledge Node. Im Zuge des Wissensaustausches mit anderen wird das Empfangsinteresse mit dem Sendeinteresse zu einem Austauschinteresse verknüpft. Es umfasst die Interessen, die der Sender senden und der Empfänger empfangen will. Das Austauschwissen ist ein Topic Map Fragment, dass durch Anwendung des Austauschinteresses das gesendete Wissen des Senders erzeugt wird, das seinerseits durch Anwendung des Sendeinteresses auf die Wissensbasis des Senders erzeugt wurde. Insofern kann der Austausch von Wissen bereits als eine verteilte Personalisierung interpretiert werden, wobei der Interessent indirekt auf eine entfernte Wissensbasis zugreift.

Natürlich können auch Orte in einem Knowledge Port definiert werden. Sie erscheinen dort als ein Teilinteresse der Knowledge Node. entweder zum Senden oder zum Empfangen von Wissen.

In Shark sind die Interessen an eine Knowledge Node gebunden. Es gibt keine irgendwie gearteten *allgemeinen Interessen*. Daher ist jede Knowledge Port immer ein Ausdruck des persönlichen Interesses (oder Gruppeninteresses) und der Austausch von Wissen ist in diesem Sinne eine Personalisierung von Wissen.

5.6.3 Gleichheit von Kontexten und Wissen in Topic Maps

5.6.4 Extraktion

Im ACP-Modell wurden die Prozesse der Extraktion und Assimilation definiert und im Modell Shark für Wissensbasen verfeinert. Die Extraktion ermittelt aus einer Wissensbasis und anhand eines Austauschkontextes eine Untermenge des Wissens in der Wissensbasis, die das Shark Peer anbietet. Die Assimilation nimmt im Gegensatz dazu Wissen entgegen und integriert es in die Wissensbasis.

Geht man davon aus, dass die Wissensbasis mit Topic Maps realisiert ist, können beide Funktionen konkretisiert werden.

In Shark ist die Extraktion wie folgt definiert.

$$extract : (KB, I_{ES}, Empfaenger) \rightarrow (K_S, I_{ES}, Sender)$$

Empfänger- und Senderbezeichnung werden aus der Definition des OKP entnommen bzw. in der Umgebung beobachtet.

In Topic Maps gibt es per se kein Nutzerkonzept. Es wird daher davon ausgegangen, dass in einer realen Implementierung entweder das Nutzerkonzept in Form eines allgemeinen Kontextes implementiert (siehe Personalisierung (5.6.2)) oder eine proprietäre Lösung genutzt wird.

Im ersten Fall kann wiederum eine Topic Map Kontextualisierung benutzt werden, die in diesem Kapitel beschrieben wurde, wobei der einschränkende Kontext der Eigentümer ist. Für diesen Fall kann die Extraktion wie folgt definiert werden, wobei die KB durch eine Topic Map (TM) ersetzt wird:

$$extract(TM, I_{ES}, Empfaenger) = (TM_S, I_{ES}, Sender)$$

mit

$$TM_S = (TM * Empfaenger) * I_{ES}$$

TM_S stellt das *gesendete Wissen* dar.

In den Beschreibungen dieses Kapitels wurden die Occurrences der Topic Maps ignoriert. Das war für die bisherigen Überlegungen statthaft, beschreibt die Kontextualisierung in Topic Maps doch eine Operation, die nur die Topics ermittelt, zu denen Wissen ausgetauscht werden kann. Sind diese ermittelt, können „einfach“ die Occurrences mit übertragen werden. Hier sind zwei Fälle zu unterscheiden, die in Abbildung 5.9 dargestellt sind.

Die Extraktion kann einmal dafür sorgen, dass alle Occurrences aufgelöst und Kopien der referenzierten Informationsressourcen ebenfalls Teil des Topic Map Fragmentes werden. Das gelingt durch Inline-Occurrences. Die zweite Variante besteht darin, dass die Referenzen *nicht* aufgelöst und lediglich die Topic Map Strukturen übertragen werden und *nicht* die referenzierten Informationsressourcen (IR).

Beide Verfahren bieten im Vor- und Nachteile. Bei der Übertragung der IR ist die übertragene Datenmenge größer. Außerdem widersprechen Inline-Occurrences der „reinen Leere“ der Topic Maps, in denen die Topics ein Netz über der Menge der Informationsressourcen bilden. Gleichzeitig profitiert ein Empfänger einer solchen Kopie einer

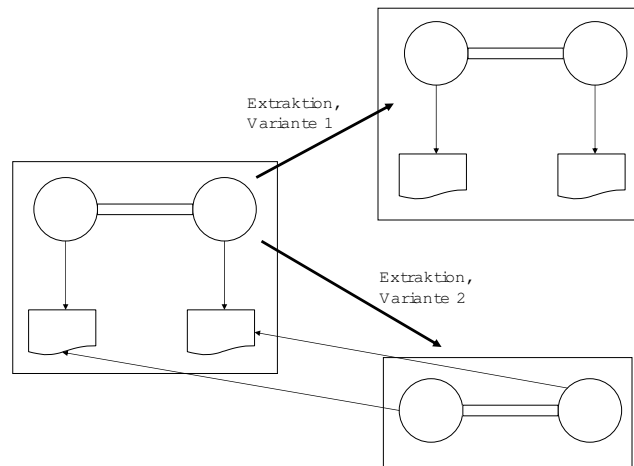


Abbildung 5.9: 2 Varianten der Extraktion mit Topic Maps

IR nicht von deren Aktualisierung. Dieser Punkt stellt sich aber auch als Vorteil heraus. Sollte die IR gelöscht werden, so ist der Empfänger nicht davon betroffen. Weiterhin ist vorteilhaft, dass bei der Übermittlung einer Kopie die IR direkt in der Wissensbasis des Empfängers vorliegt. Sie muss nicht über eine Kommunikationsverbindung geholt werden, die eventuell langsam und teuer ist, wenn sie genutzt werden soll.

Ein weiterer Punkt spricht gegen die Übertragung von Kopien, und das ist die Akzeptanz, weniger bei einzelnen Nutzern, sondern eher bei den Administratoren. Informationsressourcen werden z. B. in CMS verwaltet, die über Zugriffsverwaltung verfügen. Erhält nun ein Nutzer ein Fragment und gleichzeitig eine *Kopie* der IR, so unterliegt diese nicht mehr der Zugriffsverwaltung des CMS. Unterstellt man dem Nutzer bösen Willen oder befürchtet fehlerhafte Software, so kann es geschehen, dass bei einem folgenden Wissensaustausch eine Shark-Engine ein Fragment an einen Dritten übermittelt inklusive der Kopie der IR, die der Dritte eigentlich nicht sehen dürfte.

Diese Gefahr wird reduziert, wenn Occurrences nicht aufgelöst werden und nur Referenzen und Topic Map Fragmente enthalten sind (Variante 2). In dem Fall kann ein unberechtigter Dritter durch fehlerhafte Software lediglich einen Link erhalten, der in ein CMS verweist. Um Zugriff auf die Informations-Ressource zu erlangen, muss der Dritte auch Zugriffsrechte auf dem CMS haben. Gegen bösen Willen von Anwendern nutzt aber auch diese Variante nichts.

5.6.5 Assimilation

In Shark ist die Assimilation definiert als

$$assimilate : (KB, K_X, I_X, Sender) \rightarrow KB'$$

5 Kontextualisierung mit Topic Maps

In Topic Maps kann diese Funktion konkretisiert werden. Geht man davon aus, dass das übertragene Topic Map Fragment direkt in eine Topic Map integriert werden soll, so bietet sich das Merge an. Die einfachste Implementierung einer Assimilation ist daher durch folgende Funktion erklärt, wobei die Wissensbasis KM durch eine Topic Map (TM) ersetzt wurde und das Austauschwissen K_X durch ein Topic Map Fragment TM_X . Das Austauschinteresse ist ebenfalls eine Topic Map, die als Kontext interpretiert wird, aber in dieser Form der Assimilation keine Rolle mehr spielen wird.

Die folgende Funktion stelle eine minimale Implementierung der Assimilation dar:

$$assimilate(TM, TM_X, I_X, Sender) = merge(TM, TM_X)$$

Andere Varianten können den Austauschkontext und den Eigentümer des Wissens in den Prozess der Assimilation einbeziehen. An einigen Stellen war auch für die Zuordnung von Wissen zum Kontext *unbestätigtes Wissen* nötig. Das kann leicht dadurch erfolgen, dass im Zuge des Merge alle Konzepte von TM_X einem Scope *unconfirmed knowledge* zugewiesen werden, was bereits in [SG02] beschrieben wurde.

Löschen in einer Master-Proxy-Beziehung

In Shark ist nicht vorgesehen, dass Peers andere Peers dazu veranlassen, etwas zu löschen. In einer Master-Proxy-Beziehung muss allerdings sichergestellt werden, dass Proxies über gleiches Wissen und gleiche Interessen verfügt wie der Master. In Abschnitt 3.2.1 wurde eine einfache Variante vorgeschlagen, die darauf hinausläuft, dass vor der Assimilation auf Proxy-Seite Wissen und Interessen des Masters gelöscht werden und danach alles neu eingespielt wird.

In den eben definierten Funktionen läuft dieses Vorgehen darauf hinaus, dass zunächst mittels einer negativen Selektion über allem, was dem Master gehört, ein komplettes Löschen erfolgt und danach mittels Merge Wissen und Interessen des Masters vollständig neu eingebracht werden. In realen Implementierungen sind performantere Varianten denkbar und sollten eingesetzt werden.

5.6.6 Wissensaustausch mit Topic Maps

Nun kann der Wissensaustausch, der im letzten Kapitel gegenüber dem ACP-Modell spezialisiert wurde, mittels Topic Maps und den obigen Annahmen weiter spezialisiert werden. Es wird angenommen, dass alle Kontexte als Topic Maps beschrieben werden.

Die Funktion f ermittelte jeweils, ob ein Austausch erfolgen soll. Die Funktion liefert das Ergebnis wahr, wenn gilt

$$X_{EU} * X_U \neq TM_{\emptyset}$$

Die effektiven Interessen werden mittels der Funktion g bestimmt:

$$g(I_{[S|E]}, X_U) = I_{[S|E]} * X_U = I_{E[S|E]}$$

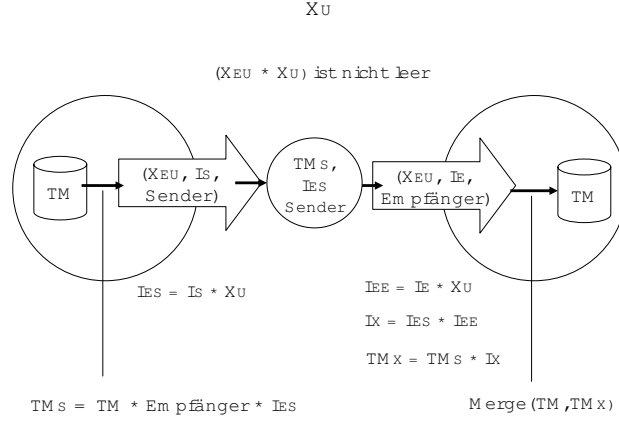


Abbildung 5.10: P2P Wissensaustausch in Shark mit Topic Fragmenten

Das Sendewissen TM_S wird durch die Extraktion ermittelt. Es soll die einfachste Version benutzt werden, die weiter oben beschrieben wurde:

$$TM_S = (TM_{\text{Sender}} * \text{Empfänger}) * I_{ES}$$

Ist es übertragen, wird ein effektives Empfangsinteresse darauf kontextualisierend angewendet:

$$TM_X = TM_S * I_{EE}$$

Das Resultat wird der Assimilation übergeben:

$$\text{merge}(TM_{\text{Empfänger}}, TM_X)$$

Diese Funktionen können in einer Formel zusammengefügt werden:

$$\text{merge}(((TM_{\text{Sender}} * \text{Empfänger}) * (I_S * (X_{U_{\text{Sender}}})) * (I_E * X_{U_{\text{Empfänger}}}), TM_{\text{Empfänger}})$$

unter der Voraussetzung, dass

$$(X_{EU_{\text{Sender}}} * X_{U_{\text{Sender}}} \neq TM_{\emptyset}) \wedge (X_{EU_{\text{Empfänger}}} * X_{U_{\text{Empfänger}}} \neq TM_{\emptyset})$$

Die Bedingung lässt sich wie folgt lesen: Ein Wissensaustausch findet statt, wenn bei Sender und Empfänger jeweils der Erwartungskontext gegenüber der Umgebung kontextualisierend angewandt auf die beobachteten Umgebungswerte nicht leer ist. Der Wissensaustausch lässt sich beschreiben als ein Merge einer Topic Map mit der Topic Map des Empfängers. Abbildung 5.10 fasst die einzelnen Formeln zusammen.

5.7 Zusammenfassung

Topic Maps sind ein Standard zur Wissensrepräsentation. Wegen seiner Ursprünge im Bereich Indexsysteme und Thesauri ist es nicht verwunderlich, dass der Standard gut geeignet ist, Begriffssysteme abzubilden.

Topic Maps erlauben die Beschreibung einer Ontologie und erlauben zusätzlich mittels Occurrences Informations-Ressourcen Subjekten zuzuordnen. Damit können Topic Map Autoren mit einer Sprache zwei Sachverhalte erklären. Sie können Ontologien modellieren und gleichzeitig explizites Wissen strukturieren. Systeme, die den Standard unterstützen, können deshalb auch als CMS bezeichnet werden.

Topic Maps unterstützen explizit das Konzept der Mikrowelten und betonen außerdem, dass Topic Maps aus der Sicht von Autoren entwickelt werden. Der Anspruch, ein globales objektives und damit kontextfreies Weltwissen zu modellieren, existiert nicht. Es war zu sehen, dass das Konzept der PSIs zur Aggregation der Mikrowelten dienen kann.

Der Standard definiert in der aktuellen Version lediglich eine Funktion, das Merge. Das erlaubt, zwei Topic Maps zu mischen. Für P2P-Anwendungen, in denen Teile einer Topic Map (Fragmente) zwischen Peers ausgetauscht werden sollen, genügt dieser Ansatz nicht. Aus diesem Grund wurden die Funktionen Fragmentierung und Kontextualisierung definiert, die wiederum auf den Funktionen View, Selektion und Korona basieren. Zur Definition dieser Funktionen wurde ein kleines mathematisches Modell für Topic Maps entwickelt.

Die in diesem Kapitel beschriebene Funktion Kontextualisierung erfüllt die Vorgaben, die in Shark angesetzt wurden. Damit können Topic Maps sowohl zur Strukturierung von Wissen als auch zur Modellierung von Kontexten eingesetzt werden. Die Nutzung nur einer (standardisierten) Sprache für Wissen und Kontext unterscheidet diesen Ansatz maßgeblich von anderen, siehe Abschnitt 2.4.4. Kontext wird in Shark als (Meta-) Wissen interpretiert und muss deshalb konsequenterweise auch mit einer Sprache zur Wissensrepräsentation darstellbar sein.

Das Kapitel hat auch gezeigt, wie ein P2P-Wissensaustausch auf die Bildung von Fragmenten und einem Merge zurückgeführt werden kann. Damit wurde gezeigt, dass das P2P-Modell für das Knowledge Management aus dem Abschnitt 2.5 realisiert werden kann.

6 Architektur einer Shark-Engine

In Abschnitt 4.3 wurden die Komponenten der Shark-Engine genannt, aber die technischen Ausprägungen nicht diskutiert. Im vorangegangenen Kapitel wurde gezeigt, dass Topic Maps für die Wissensbasis der Shark-Engine eingesetzt werden können.

In diesem Kapitel soll ein mögliches Knowledge Exchange Protocol (KEP) vorgestellt werden. Daran anschließend werden mögliche Implementierungsplattformen für Shark-Engines diskutiert. Dabei wird auch zu sehen sein, auf welche Protokolle KEP aufsetzen kann. Es wird nur grob skizziert, welche Funktionen ein Sensor haben kann.

Die letzten Themen werden nur angerissen und sollen anhand von Implementierungs-ideen zeigen, dass sich das Konzept von Shark nicht im visionären Raum bewegt, sondern durchaus implementierbar ist. Es werden aber keine konkreten Schnittstellenbeschreibungen gegeben oder detaillierte Beschreibungen einer Softwareplattform. Das würde den Rahmen dieser Arbeit sprengen.

6.1 Knowledge Exchange Protocol (KEP)

Im OSI Referenzmodell kann KEP in der Schicht 7 eingeordnet werden. Es ist ein Protokoll der Anwendungsebene und läuft zwischen zwei Shark-Engines (im Weiteren mit SE abgekürzt) ab. Nach Sowa [Sow00] kann das Protokoll dem Knowledge-Level zugeordnet werden.

KEP ist das Protokoll, mit dessen Hilfe ein Knowledge Port einen Wissensaustausch steuert. Dieser Wissensaustausch wird im Wesentlichen dadurch realisiert, dass in der Umgebung ein Peer wahrgenommen wird, eventuell Bezeichnungen seiner Knowledge Ports und die zugehörigen Interessen. Das Protokoll muss die Möglichkeit geben, auf diese Beobachtungen zu reagieren und schlussendlich Wissen an ein Peer zu senden.

Shark beschreibt nicht genau, in welcher Reihenfolge ein Wissensaustausch erfolgen soll. KEP soll hierzu auch keine zusätzlichen Restriktionen setzen.

Die Grundidee von KEP ist folgende: Ein Knowledge Port soll in der Lage sein, die eigenen Interessen der Umgebung bekannt zu geben. Dazu werden die Funktionen **interest** und **offer** dienen, die jeweils Empfangs- und Sendeinteresse publizieren. In Abwandlung und zur Effektivitätssteigerung wird eine Funktion **accept** definiert. Diese Funktion wird benutzt, wenn ein Knowledge Port ein effektives Sendeinteresse beobachtet und daraufhin ein Austauschinteresse erzeugt und der Umgebung bekannt gibt. Im Shark-Modell war diese Möglichkeit nicht direkt vorgesehen. Dort wurde davon ausgegangen, dass Wissen immer zunächst übertragen und erst beim Empfänger zu Austauschwissen gewandelt wird. **Accept** erlaubt, dass ein potenzieller Sender bereits vor der Übertragung das Austauschinteresse erhält und Austauschwissen übertragen kann. Das dient der Optimierung.

Accept erlaubt auch zu beschreiben, wie lange Austauschwissen akzeptiert wird. Damit kann ein Knowledge Port erreichen, dass sich ein OKP das Austauschinteresse merkt und bei veränderter Wissensbasis direkt neues Wissen sendet. Damit kann eine *Knowledge Channel* etabliert werden. Eine Funktion **stop** dient dem (vorzeitigen) Schließen eines solchen Kanals. Die Funktion **insert** dient dem Senden von Wissen.

Im Folgenden werden die Funktionen genauer beschrieben. Dabei wird folgende Sprachregelung benutzt. Die Daten, die mittels KEP übertragen werden, werden als KEP-Nachricht bezeichnet. Alternativ wird auch von einer KEP-Funktion gesprochen. Das erfolgt, wenn aus der Sicht eines Knowledge Ports geschrieben wird, der das Protokoll über eine API benutzen kann.

Nach der Vorstellung der einzelnen Funktionen werden mögliche Szenarien diskutiert.

6.1.1 Genereller Aufbau einer KEP-Nachricht

Eine KEP-Nachricht beinhaltet generell folgende Informationen:

Adresse Der Parameter enthält die Adresse eines unterliegenden Protokolls, an die folgende Nachrichten gesendet werden sollen. Ist der Parameter nicht gesetzt, kann die Absenderadresse für Antworten benutzt werden.¹ Dieser Parameter ist immer optional.

Bezeichnung der sendenden SE Mit dem Parameter kann die Knowledge Node - hier der Wissenssucher - ihre Identität preisgeben. Wenn dieser Parameter nicht angegeben ist, bleibt sie anonym. Dieser Parameter ist immer optional.

Bezeichnung des Senders/Empfängers des Knowledge Ports Ein IKP verfügt über eine Empfängerbezeichnung, unter der Wissen entgegengenommen wird. Ein OKP hat analog eine Senderbezeichnung. Beide müssen nicht mit der Bezeichnung der SE übereinstimmen. Diese kann bei jeder KEP-Nachricht übertragen werden. Dieser Parameter ist immer optional.

Weitere Parameter sind vom jeweiligen Nachrichtstypen abhängig. Im Folgenden werden die Nachrichten im Detail besprochen. Dieser Abschnitt kann übersprungen werden, wenn man sich nur für das generelle Konzept interessiert. Es werden im Folgenden keine Aussagen über die Formate gemacht. Diese können je nach unterliegenden Protokollen variieren. Für das Format des übertragenen Wissens und der Kontexte wird in Kapitel 5 Topic Maps empfohlen und gezeigt, wie damit umgegangen werden kann.

6.1.2 interest

Mit der Nachricht **interest** legt ein IKP Parameter offen, das können sein, die Erwartung an die Umgebung, das Empfangsinteresse bzw. die Empfangsbezeichnung. Letztere

¹Im Generic Connection Framework der Java Microedition könnte eine Adresse z. B. wie folgt lauten: `tcp://schwotzer.sharksystem.de:2222`

ist allerdings bereits Standardparameter jeder Nachricht, genauso wie die Bezeichnung des Peers.

Es steht dem IKP frei, ob es ein effektives Empfangsinteresse oder direkt das Empfangsinteresse offen legen möchte. Neben den KEP-Standardparametern verfügt das **insert** damit über folgende Parameter:

Name	Erklärung	Optional / Pflicht
Erwartung an die Umgebung (X_{EU})	Erwartung an die Umgebung, z. B. Liste der potenziellen Sender von Wissen, Anforderungen an Kommunikationskanäle, Anforderungen an beliebige Umgebungsfaktoren.	optional
Empfangsinteresse (I_E / I_{EE})	Der Parameter definiert das Empfangsinteresse (I_E oder I_{EE}). Wenn der Parameter leer ist, wird davon ausgegangen, dass sich der Wissenssucher für alles interessiert.	optional

6.1.3 offer

Mit der Nachricht **offer** legt ein OKP Parameter offen. Das ist sind die Erwartungen an die Umgebung und das Sendeinteresse. Es gilt auch hier, dass wahlweise ein effektives oder das Sendeinteresse übermittelt werden können.

Neben den KEP-Standardparametern verfügt das **offer** damit über folgende Parameter:

Name	Erklärung	Optional / Pflicht
Erwartung an die Umgebung (X_{EU})	Erwartung an die Umgebung, z. B. Liste der potenziellen Empfänger von Wissen, Anforderungen an Kommunikationskanäle, Anforderungen an beliebige Umgebungsfaktoren.	optional
Sendeinteresse (I_E / I_{SE})	Der Parameter definiert das (effektive) Sendeinteresse. Wenn der Parameter leer ist, wird davon ausgegangen, dass die Shark-Engine prinzipiell zu allem aussagenfähig ist.	optional

6.1.4 accept

Ein **accept** ist aus Sicht der Parameter identisch zu einer **interest**-Nachricht. Der wesentliche Unterschied liegt in der Semantik. Ein **accept** drückt den Wunsch zum Empfang von Wissen aus, während **interest** allgemein bekannt gibt, dass sich ein Knowledge Port prinzipiell für gewisse Themen interessiert.

Ein zusätzlicher Parameter definiert, wie lange die Bereitschaft zum Wissensempfang gilt. Durch diesen Parameter kann ein *Knowledge Channel* etabliert werden, das ist eine Verbindung zwischen zwei Shark-Engines, wobei die eine Änderungen in ihrer Wissens-

basis der anderen mitteilt. Letzterer baut durch das Senden eines **accept** einen solchen Kanal auf.

Accept verfügt damit über folgende Parameter, zusätzliche zu denn allgemein üblichen KEP-Parametern:

Name	Erklärung	Optional / Pflicht
Erwartung an die Umgebung (X_{EU})	Erwartung an die Umgebung, z. B. Liste der potenziellen Sender von Wissen, Anforderungen an Kommunikationskanäle, Anforderungen an beliebige Umgebungsfaktoren.	optional
Empfangsinteresse (I_E / I_{EE}) Dauer	Der Parameter definiert das Empfangsinteresse (I_E oder I_{EE}). Wenn der Parameter leer ist, wird davon ausgegangen, dass sich der Wissenssucher für alles interessiert. Der Parameter definiert, wie lange der IKP Wissen zu diesem Austauschkontext akzeptiert. Es kann angegeben werden, ob der Austausch einmalig erfolgen soll (default), bis zu einem bestimmten Datum oder bis zum expliziten Beenden durch stop .	optional Optional

Für die ersten beiden Parameter gelten die gleichen Überlegungen, die weiter oben für **interest** gemacht wurden. Wenn der Kanal dauerhaft angelegt wird, muss auch überlegt werden, ob das Senden eines effektiven Empfangsinteresses sinnvoll ist oder aber das Senden eines allgemeineren Empfangsinteresses. Der Parameter *Dauer* wurde bereits besprochen.

6.1.5 insert

Mit **insert** übermittelt ein Knowledge Port eine Dateneinheit an eine andere Shark-Engine. Die Parameter der Wissenseinheit wurden bereits im Abschnitt 4.2 definiert. Sie schlagen sich hier als Parameter der Nachricht nieder. Zusätzlich zu den Standard-KEP-Parametern verfügt KEP-**insert** über folgende:

Name	Erklärung	Optional / Pflicht
Austauschkontext (I_X)	Der Parameter definiert den Austauschkontext, der von einem IKP anhand der Funktion h ermittelt wurde, siehe weiter oben.	Pflicht
Eigentümer	Eigentümer des Wissens. Kann anonym sein.	Pflicht
Wissen	Das Sende- bzw. Austauschwissen.	Pflicht

Alle Parameter sind Pflichtparameter. Sie stellen zusammen eine Dateneinheit nach dem ACP-Modell dar.

6.1.6 stop

Ein IKP sendet ein **stop** an einen OKP, um einen *Knowledge Channel* zu schließen. Der Kanal wird durch ein Interesse bezeichnet:

Name	Erklärung	Optional / Pflicht
Empfangsinteresse (I_E / I_{EE})	Der Parameter definiert das Empfangsinteresse, das hier als negatives Interesse interpretiert werden soll, in dem Sinne, dass Wissen, das dazu passt, <i>nicht mehr</i> gesendet werden soll. Wenn der Parameter leer ist, soll zu keinem Thema etwas gesendet werden.	Pflicht

Nun sind die einzelnen Nachrichtenarten definiert. Es sollen im Weiteren Eigenschaften des Protokolls und Szenarien diskutiert werden, wie Wissen ausgetauscht werden kann.

6.1.7 Eigenschaften des Protokolls

Das Protokoll ist zustandsbehaftet, da sich Knowledge Ports prinzipiell beobachtete Umgebungswerte merken können und damit indirekt die gesendeten Nachrichten anderer. Das Protokoll ist asynchron. Eine KEP-Nachricht wird so interpretiert, dass sie Parameter der Umgebung offen legt.

6.1.8 Einige KEP-Szenarien

Bevor auf einzelne Szenarien eingegangen wird, sollen einige Prinzipien diskutiert werden, denen Knowledge Ports folgen können.

Ein IKP kann zwischen zwei prinzipiellen Vorgehensweisen wählen. Er kann als passiver und unerkannter Lauscher seine Umgebung beobachten und Wissen annehmen, wenn es zu seinen Interessen passt. In dem Fall wird er lediglich mit einem **interest** bekannt geben, dass er existiert, aber keine Empfangsinteressen publizieren. Er kann auch seine Empfangsbezeichnung geheim halten und anonym agieren.

Die andere Variante besteht darin, dass er aktiv nach Wissen sucht. Er kann sowohl seine Erwartungen an einen potenziellen Sendern publizieren als auch seine Empfangsinteressen. Ein aktiver IKP sollte auch auf beobachtete Sendeinteressen reagieren, indem er ein **accept** absetzt und damit sein Empfangsinteresse gegenüber diesem Peer bekannt gibt. Dieses könnte dann seinerseits gezielt an ihn Austauschwissen erzeugen und bereitstellen.

Zwischen diesen Extrema gibt es Mittelwege. In den Szenarien werden einige Möglichkeiten gezeigt, aber auch im folgenden Kapitel, wenn Anwendungen diskutiert werden.

Ein OKP hat ebenfalls zwei generelle Möglichkeiten zum Handeln. Auch er kann zunächst passiv die Umgebung beobachten und Wissen dann anbieten, wenn er ein Empfangsinteresse wahrnimmt. Er kann daraufhin Wissen anbieten, muss dabei lediglich das Wissen und das effektive Sendeinteresse übermitteln. Er muss weder Sender-Bezeichnung, Bezeichnung der Shark-Engine noch generelle Sendeinteressen offen legen.

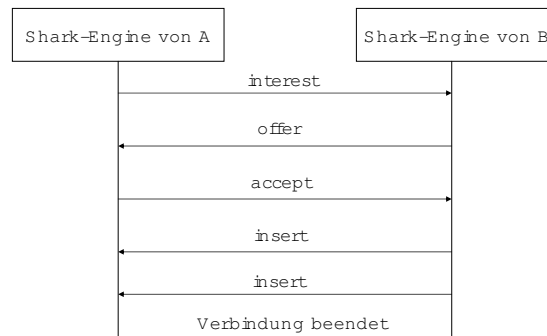


Abbildung 6.1: Das Standard KEP Szenario

Ein alternativer Ansatz besteht darin, aktiv Wissen anzubieten. Er kann alle Parameter, so auch das Sendeinteresse, offen legen.

Die Vorteile in der Offenlegung aller Parameter und der aktiven Suche bestehen darin, dass ein Wissensaustausch effektiver ist. Mit der Kenntnis der Bezeichnung und aller Interessen kann viel genauer ein Angebot über Wissen gemacht werden. Der Austausch direkt von Austauschwissen spart auch Ressourcen.

Der Nachteil besteht in der Offenlegung der Bezeichnung und dem Interessenprofil, was nicht immer erwünscht sein muss. Allgemein kann aber nicht entschieden werden, welche Strategie die beste ist. Es hängt von der Anwendung ab.

Das Standard KEP Szenario

Das erste Szenario soll als das Standard Szenario bezeichnet werden, siehe Abbildung 6.1.

(In dieser und den folgenden Erläuterungen soll eine sprachliche Vereinfachung benutzt werden: Konzeptuell sendet ein Knowledge Port von A eine Nachricht. Vereinfacht soll aber immer davon gesprochen werden, dass A eine Nachricht sendet. Wenn auf parallel sendende Knowledge Ports hingewiesen werden soll, werden genauere Erklärungen genutzt.)

In diesem Szenario wird davon ausgegangen, dass ein potenzieller aktiver Empfänger A auf einen aktiven potenziellen Sender B trifft. A sendet in dem Falle ein **interest** und publiziert damit sein (effektives) Empfangsinteresse in der Umgebung. B teilt daraufhin sein effektives Sendeinteresse durch ein **offer** mit. A sieht nun den potenzieller Sender, kennt dessen Sendeinteressen und kann damit bereits das Austauschinteresse ermitteln und dieses publizieren (**accept**). B benutzt das Austauschinteresse, um Austauschwissen

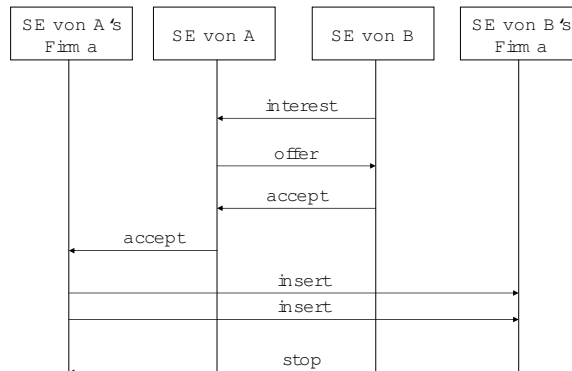


Abbildung 6.2: KEP mit Redirect

zu erzeugen und bietet es in diesem Beispiel durch das Senden von zwei **insert** an. Aus dem Szenario ist nicht ersichtlich, wie viel Zeit zwischen dem Senden vergeht. Es können Millisekunden sein, aber auch Tage.

6.1.9 KEP Redirect

Das folgende Szenario soll ein Redirect von KEP illustrieren, siehe Abbildung 6.2. Auch hier werden wieder die Shark-Engines (SE) von A und B beispielhaft benutzt. Zusätzlich werden noch die Shark-Engines der Firmen von A und B eingeführt.

Auch hier tauschen A und B effektive Interesse wie im Basisszenario. B übermittelt mit dem Senden von **accept** ein Austauschinteresse an A. Dabei benutzt es den Parameter **Antwortadresse** und trägt dort die Adresse der Shark-Engine des eigenen Unternehmens ein.

A wiederum behandelt den Aufruf auch nicht selbst, sondern sendet ihn weiter an die Shark-Engine der eigenen Firma. Diese wiederum bearbeitet ihn.

Der Vorteil dieses zustandsfreien Protokolls besteht darin, dass Nachrichten ohne Veränderung weitergesendet werden können, wie in diesem Beispiel zu sehen. Die Shark-Engine von der Firma von A empfängt nun einen Austauschkontext und eine Adresse. Sie beginnt daraufhin Austauschwissen an die Shark-Engine der Firma von B zu senden. In dem Beispiel beendet die Shark-Engine von B den so etablierten Knowledge Channel nach einer Weile.

Das Szenario könnte z. B. während einer Messe ablaufen. Zwei mobile Mitarbeiter haben übernommene Interessen ihrer Firmen, z. B. auf einem mobilen Gerät. Die Geräte führen die Aushandlungsphase von KEP durch, aber nicht mehr die Austauschphase. Diese delegieren sie an die Shark-Engines ihres Unternehmens, von dem sie auch die

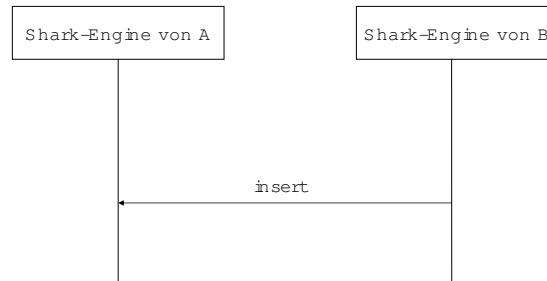


Abbildung 6.3: Das einfachste KEP Szenario

Interessen übernommen haben.

Auf diese Art und Weise kann ein Wissensfluss zwischen Unternehmen durch zwei mobile Mitarbeiter etabliert werden. Das Beispiel illustriert sehr schön, wie die beiden Mitarbeiter in dem Fall ausschließliche in der Rolle als Firmenmitarbeiter aufgetreten sind.

6.1.10 Das einfachste KEP Szenario

Das folgende Szenario ist das einfachste Szenario mit KEP, siehe Abbildung 6.3. Hier sendet die Shark-Engine von B lediglich Austauschwissen zu der Shark-Engine von A.

A kann sich nun entscheiden, ob es das Wissen assimilieren will oder nicht. Die Aushandlungsphase fehlt in diesem Szenario vollständig. Dieses Szenario ist vor allem dann sinnvoll einsetzbar, wenn die mögliche Zeit für einen Wissensaustausch sehr limitiert ist, wie es z. B. bei mobilen Anwendungen der Fall ist. Angenommen, jemand geht bei einer Messe an einem Stand vorbei. Dieser will die eigenen Produkte bewerben. Erfolgt die Kommunikation beispielsweise mit Bluetooth stehen nur wenige Sekunden für eine Übertragung zur Verfügung. Eine komplette Aushandlungsphase ist in dieser Zeit kaum durchführbar. In diesem Fall könnte der Sender auf gut Glück einfach die Werbung versenden. Sind das Austauschwissen und der Austauschkontext passend für den Empfänger, erfolgt eine Assimilation. Im anderen Falle wird das Wissen einfach verworfen.

6.1.11 Vergleich mit Software-Agenten

Im Abschnitt 2.2 wurden Software-Agenten vorgestellt. Diese ähneln in einigen Punkten dem Modell Shark, insgesamt mehr aber noch der Shark-Engines. Letztere hat mit Agenten Folgendes gemein:

- Beide verfügen über eine Wissensbasis.
- Beide sind in der Lage, mit anderen Shark-Engines/Agenten zu kommunizieren.

Hier enden aber auch bereits die Gemeinsamkeiten. Bei einem genaueren Blick zeigen sich die grundlegenden Unterschiede:

- Die Wissensbasen von Agenten enthalten Wissen in Form von Meinungen und Zielen. Die Wissensbasis einer Shark-Engine enthält explizites Wissen und Policies, wie mit diesem im Zuge eines Austausches umgegangen werden soll.

Die Anforderungen der Shark Engine an die Wissensbasis ist damit deutlich geringer. Sie muss in der Lage sein, explizites Wissen mittels eines Kontextes speichern zu können und eine Realisierung der Funktion Kontextualisierung zu implementieren. Über die Realisierung der Policies werden keine weiteren Aussagen gemacht. Diese können im trivialen Fall dazu führen, dass jedes Wissen ohne weiteres in die Wissensbasis integriert wird oder in den Start eines beliebig komplexen Workflows münden.

- Die Semantik der Nachrichten (sie nenne sich *Kommunikationsakte* in ACL bzw. *Performatives* in KQML) von Agentensprachen orientiert sich an der Sprechakttheorie. Die meisten Nachrichten sind die Realisierung eines illokutionären Aktes, viele auch eines perlokutionären. Sie sind also darauf ausgerichtet, eine Reaktion bei einem anderen Agenten zu erzeugen. Da die meisten Agenten auf modaler Logik basieren, ist auch die Semantik der Nachrichten entsprechend. Es gibt keine Anweisungen, sondern eher Mittellungen darüber, dass ein Agent es wünscht, dass ein anderer eine Aktion ausführen würde.

KEP geht von einer Shark-Engine aus, die keine Ziele verfolgt und deren Wissen nicht in einer modalen Logik abgelegt ist. KEP-Nachrichten haben daher eher die Semantik von einer Sprache über Datenbanken: Ein Sender teilt dem Empfänger ein Interesse an Wissen mit (*interest*, *offer*, *accept*) bzw. erklärt, dass er etwas einfügen soll (*insert*).

- Shark-Engines haben keine Ziele, Agenten schon. Knowledge Nodes auch.

6.2 Plattformen für die Shark-Engine

Nunmehr wurde eine mögliche Realisierung von KEP vorgestellt und einige Szenarien diskutiert. Interessant ist nun, auf welchen Plattformen eine Shark-Engine implementiert

werden kann. Im Folgenden werden drei Versionen einer Implementierung der Shark-Engine diskutiert. Vor allem im folgenden Kapitel wird zu sehen sein, dass die parallele Nutzung aller drei Arten vor allem im Bereich des mobilen Knowledge Managements sinnvoll ist, aber auch in anderen Bereichen, wie dem mobilen Marketing bzw. in Sensornetzen. In diesem Kontext sind die folgenden Überlegungen zu sehen.

Auch hier wird nicht sehr ins Detail gegangen werden. Dieser Abschnitt dient der Abrundung der Arbeit und soll zeigen, dass reale Implementierungen von Shark möglich sind, ohne diese detailliert zu beschreiben. Das ist nicht Gegenstand dieser Arbeit. Teilweise wurden Implementierungsdetails auch in [SG03] diskutiert.

Als Einstieg in das Kapitel soll die Architektur der Shark Engine noch einmal aufgegriffen und leicht modifiziert präsentiert werden, siehe Abbildung 6.4. Die erste Veränderung besteht in der Benennung. Um eine klare Trennung zwischen dem Referenzmodell und konkreten Implementierungen zu erreichen, soll im Folgenden von einer *Shark Station* die Rede sein. Formelhaft ausgedrückt: Eine Shark Station ist eine Implementierung einer Shark Engine.

Außerdem steckt hinter dem Namen Shark Station eine Historie: Die ersten Entwicklungen im Rahmen von Shark befassten sich mit der Implementierung einer Topic Map Engine für mobile Geräte. Die 3GPP, das Standardisierungsgremium für mobile Funknetze der dritten Generation (u. a. UMTS) bezeichnet alle mobilen Geräte (PDAs, Handys etc.), *Mobile Stations*. Daher wurde konsequent die mobile Komponente des Systems als *Shark Mobile Station* bezeichnet. Aus dieser Benennung ergaben sich die Bezeichnungen *Central Station* und *Local Station*. Bei der Entwicklung des allgemeinen Shark Modells stellte sich heraus, dass die Benennung *Station* nicht gut ausdrückt, was die Komponente ist, nämlich eine Maschine, die Wissensflüsse steuert. Daher erfolgte im Referenzmodell eine Neubenennung der Komponenten in Shark Engine. Die Bezeichnungen der Implementierungen wurden beibehalten.

Die Architektur der Shark Station (Abbildung 6.4) konkretisiert die Shark Engine (Abbildung 4.3) wie beschrieben. Das *Protokoll* und der *Sensor* sind in eine Komponente, die Verteilungsplattform, zusammengefloßen. Es wird noch zu sehen sein, dass das sinnvoll ist. Es wird prinzipiell davon ausgegangen, dass Topic Maps zur Repräsentation von Wissen und Kontext eingesetzt wird. Es wird allgemein davon ausgegangen, dass Informations-Ressourcen in einem beliebigen Content Management System gespeichert werden können. Die Wissensbasis der Shark-Engine kann daher konkretisiert werden in eine Topic Map Engine (TME), die auf Informations-Ressourcen in einem CMS verweist, entweder dem eigenen lokalen oder einem externen System. Es gibt Systeme, die TME und CMS vereinen, hier sind sie getrennt dargestellt.

Die unterschiedlichen Färbungen sollen andeuten, ob vorhandene Systeme eingesetzt werden können oder ob sie Shark-spezifisch implementiert werden müssen. Die dunklen Komponenten sind Implementierungen von Shark Konzepten (KP-Management, KEP). Die helleren Komponenten (hier die TME) sind Komponenten, die sich an einige Konventionen von Shark halten müssen, bei denen aber keine Shark-spezifische Implementierung notwendig ist. Die weißen Komponenten sind völlig unabhängig von Shark. Das sind hier ein CMS bzw. eine Verteilungsplattform.

Im Folgenden werden die drei genannten Ausprägungen einer solchen Shark-Station

Shark Station	
KP-Management	KEP
TME	Verteilungs- Plattform mit Verzeichniss- diensten
CMS	

Abbildung 6.4: Shark Station - Implementierung der Shark Engine

diskutiert. Dabei ist die Mobile Station eine Implementierung für mobile Geräte, wie Handys und PDAs, die Local Station eine Implementierung für stationäre Systeme, die mittels Kurzstreckenfunk Wissen mit Mobile Stations austauschen können (ortsbasiertes Wissen), und die Central Station, mit deren Hilfe die anderen Stations administriert werden können. Das klingt ein wenig nach einer zentralistischen Lösung, ist es aber nicht. Tatsächlich wird davon ausgegangen, dass die Mobile Station und die Local Station keine Nutzerinterfaces anbieten, sondern dass sie ihr Wissen und Interessen von einer Central Station erhalten. Die Central Station kann aber ein PC eines Mitarbeiters sein, der über weitere Geräte, wie Handy, PDA, Laptop etc., verfügt. Diese Geräte werden von einem PC aus administriert. Die Central Station ist damit die zentrale Administrationseinheit einer technisch motivierte „Gruppe von Shark Stations“, die durch die verschiedenen technischen Geräte gebildet wird, mit denen viele heutzutage täglich arbeiten (müssen). Solche Arten von Gruppen wurden in der Referenzarchitektur nicht beachtet. Man möge sich immer eine Shark Central Station als Implementierung einer Shark-Engine vorstellen, wenn man das Kapitel 4 noch einmal lesen sollte.

6.2.1 Shark Central Station

Abbildung 6.5 stellt eine Shark Central Station (SharkCS) schematisch dar. Sie ist eine spezielle Shark-Station und bietet alle Funktionen einer Shark-Engine an, insbesondere das Management der Knowledge Ports mittels eines KP-Editors und das Aktivieren und Deaktivieren des Systems. Weiterhin ist ein Themen-Editor vorgesehen, der es dem Anwender erlaubt, Themen zu verwalten und den Themen explizites Wissen zuzuordnen. In der SharkCS werden die Themen als Topics verwaltet, die beliebig assoziiert werden können bzw. mit beliebigen Informations-Ressourcen im eigenen CMS bzw. auf

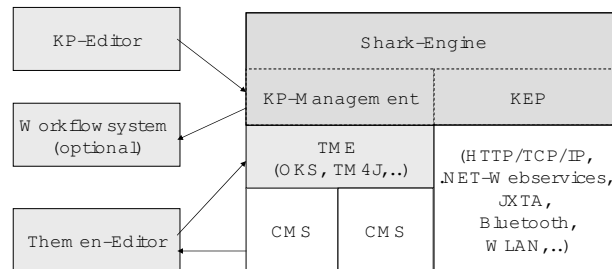


Abbildung 6.5: Shark Central Station

entfernten Systemen verknüpft werden können.

Eine dritte externe Komponente ist optional: ein Workflowsystem. Die Idee wurde bereits in [Sch02] formuliert und in [Yua03] ausführlich diskutiert und realisiert. Wissen gelangt über einen IKP in das System und kann danach assimiliert werden. Dieser Assimilationsprozess kann beliebig komplex sein. In Abschnitt 5.6.4 wurde der denkbar kürzeste präsentiert, indem die Assimilation direkt auf ein Topic Map Merge zurückgeführt wurde. In praktischen Anwendungen kann das nur die Ausnahme sein.

Es wird davon ausgegangen, dass es eine Knowledge Management Policy im Unternehmen gibt, die beschreibt, wie mit Wissen umgegangen wird, das ins Unternehmen hineinfließt. In Shark wird davon ausgegangen, dass das eingehende unbestätigte Wissen zu einem Mitarbeiter fließt, der das Wissen bewertet und damit für die Gruppe freigibt. Der Prozess kann auch beliebig komplexer sein und mehr Mitarbeiter involvieren.

In [SP02, SG02] wurde folgende technische Realisierung dieses Prozesses vorgeschlagen: Eingehendes unbestätigtes Wissen (eingehende Topic Map Fragmente) wird mittels Merge in eine Topic Map integriert und einem Scope *unconfirmed knowledge* zugeordnet. Ein Workflowsystem kann nunmehr den zuständigen Mitarbeiter informieren, dass in einer Topic Map unbestätigtes Wissen vorliegt, das es zu evaluieren gilt. Es könnte eine Referenz auf das Wissen übergeben werden. Der Mitarbeiter kann das Wissen bewerten und verwerfen bzw. übernehmen. Dieses Vorgehen widerspricht den Prinzipien von Shark in der Form, dass das unbestätigte Wissen nicht zu einer Shark-Engine des Mitarbeiters fließt, sondern auf einem zentralen System verbleibt, wo es bewertet werden soll. Dieses Herangehen kann aber entweder in einer Übergangsphase benutzt werden, bevor ein komplettes P2P-KMS installiert wird. Gleichzeitig zeigt es, dass sich Teile der Prinzipien von Shark auch mit zentralistischen Ansätzen umsetzen lassen. Es gelten dann aber

weiterhin die Kritikpunkte an zentralistischen Systemen. In [Yua03] wurde gezeigt, dass und wie eine Integration des Konzeptes Shark in ein Workflowsystem erfolgen kann.

Plattformen für die Central Station

Das Protokoll KEP steuert den Wissensaustausch zwischen Shark Engines. Dieses Protokoll muss auf einer verteilten Plattform aufsetzen, die den Datenaustausch zwischen zwei Prozessen auf unterschiedlichen Rechnern erlaubt. Im Rahmen der Arbeiten an Shark wurden Implementierungen von KEP direkt oberhalb von UDP, TCP, aber auch basierend auf HTTP und Webservices durchgeführt. Alternativ wurde auf RFCOMM aufgesetzt, einem Bluetooth-Protokoll, das mit TCP vergleichbar ist. KEP ist sehr einfach strukturiert und konnte ohne größeren Aufwand auf diese Protokolle abgebildet werden. Die Implementierung erfolgte in Java.

Es spricht nichts dagegen, dass auch komfortablere Plattformen, wie CORBA oder EJB, eingesetzt werden können, wenn das System schon mittels Java und oftmals recht einfacher Protokolle abgebildet werden kann. In der Abbildung werden exemplarisch einige der Protokolle genannt. Es wurden keine praktischen Erfahrungen mit W-LAN gesammelt, allerdings liefen die genannten Implementierungen in TCP/IP-basierten Netzen in einem LAN. W-LAN würde die MAC-Schicht ersetzen. Es spricht kein prinzipielles Problem dagegen, dass ein TCP-basiertes Protokoll nicht auch über W-LAN läuft. Daher wurde es mit aufgeführt.

Es ist ebenfalls JXTA als Plattform aufgeführt. JXTA [Mic] ist eine Entwicklungsplattform für P2P Systeme. Sie liegt in einer vergleichbaren Abstraktionsebene zu CORBA und EJB. JXTA verfügt über ein sehr einfaches Peer-Modell. Das besagt, dass Peers autonom sind und mit anderen Peers Daten austauschen können. JXTA bietet nunmehr Dienste, wie Peers anderen Peers ihre Existenz mitteilen können und die es erlauben, Mitteilungen im Netz der Peers zu verteilen. JXTA entwickelte sich parallel zu den Arbeiten an Shark. Es wurden keine praktischen Erfahrungen gesammelt. Es steht aber zu vermuten, dass das JXTA-Konzept sehr passend als Basis für eine Implementierung von Shark ist.

Als Topic Map Engine wurde erfolgreich OKS der Firma Ontopia AS aus Oslo eingesetzt, siehe [OA].

Die Funktionalität eines Sensors beschränkt sich bei der Central Station und den genannten Plattformen auf wenige mögliche Beobachtungen. In verteilten Plattformen können sich Dienste anmelden. Auch die SharkCS kann sich als Dienst anmelden und wird dadurch für andere sichtbar. Eine „Veränderung der Umgebung“ ergibt sich daraus, dass ein neuer Service in einem Verzeichnisdienst, wie z. B. UDDI, entdeckt wird. Der Sensor muss dazu regelmäßig Verzeichnisdienste abfragen.

Bluetooth bietet einen *Service Discovery* an, das dem Aufbau spontaner Netze dient. Der Sensor kann sich dabei bei diesem Dienst anmelden und eine Notifikation erhalten, wenn ein anderes Bluetooth-fähiges Gerät in der Nähe ist, oder sogar dessen Namen erfahren.

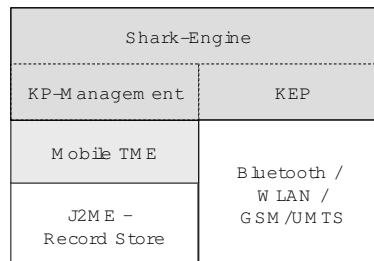


Abbildung 6.6: Shark Mobile Station

6.2.2 Shark Mobile Station

Mobile Geräte, wie PDAs und Mobiltelefone, bieten bereits heute ausreichende Möglichkeiten, um Teile von privatem bzw. Unternehmens-Wissen zu speichern.

Shark macht auch eine klare Trennung zwischen Kontext des Wissens und dem Wissen selbst. In Topic Maps bedeutet das konkret, dass der Kontext durch Topics und Assoziationen beschrieben wird, während mittels Occurrences auf Informationsressourcen verwiesen wird, die das explizite Wissen enthalten. Der Kontext (repräsentiert z. B. mittels XTM) bewegt sich in Größenordnungen von Kilo-Byte. Die Größe von explizitem Wissen, in Form von Dokumenten gespeichert, liegt im Bereich von Mega- und Gigabyte. Den Kontext kann also auch ein Handy verwalten. Das Wissen kann bei Bedarf über ein Funknetz von einem CMS bezogen werden. Aus diesen Überlegungen heraus ist auch die Shark Mobile Station sinnvoll, siehe Abbildung 6.6.

Zielplattform der SharkMS sind mobile Geräte. Eine Implementierung unter Java für kleine Geräte (J2ME) wurde in [SG03] beschrieben. Daten werden dabei im J2ME-eigenen Record Store gespeichert. Eine performantere Implementierung einer TME unter J2ME wird in [Wal05] vorgestellt.

Eine prototypische Implementierung konnte auch zeigen, dass KEP auf einem PDA mittels J2ME implementiert werden kann und auf Bluetooth aufsetzt.²

Es sind keine externen Tools dargestellt, wie bei der Central Station. Tatsächlich sind alle denkbaren Tools optional. Man kann davon ausgehen, dass das Strukturieren von Wissen und Bearbeitung wegen der beschränkten Größe des Bildschirms nicht erfolgen kann. Weder KP- noch Themen-Editor sind damit notwendig, geschweige denn ein

²Vorstellung im März 2003 auf der CeBIT in Hannover

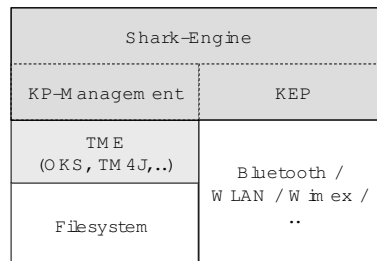


Abbildung 6.7: Shark Local Station

Workflowsystem. In einigen Fällen mag ein Tool zum Navigieren und Ansehen des Wissens sinnvoll sein. Praktikabel ist das vor allem auf PDAs und einigen Smartphones. Auf kleinen Handybildschirmen ist auch das unrealistisch.

Die SharkMS ist aber auch in einer „blinden“ Variante sinnvoll, womit gemeint ist, dass weder das Wissen in irgendeiner Form benutzt werden kann noch dass die Flüsse von Wissen irgendwie verwaltet werden können. Das wird dann sinnvoll, wenn die SharkMS Teil einer Synchronisationsgruppe von Shark Stations ist.

In dem Fall kann sie Teile des Wissens eines Anwenders und vor allem dessen Knowledge Ports aufnehmen. Bauen zwei solche Handys z. B. eine Bluetooth-Verbindung, auf, kann ein Wissensaustausch erfolgen. Das Wissen kann zwar nicht direkt eingesehen werden, aber es wird bei nächster Gelegenheit z. B. auf einen Laptop oder PC oder Ähnliches gelangen und kann dann bearbeitet werden.

Der Sensor der Mobile Station beschränkt sich auf die Anmeldung bei einem Dienst zum Service Discovery, wie er bereits oben bei der Central Station beschrieben wurde.

6.2.3 Shark Local Station

Die Shark Local Station (SharkLS), siehe Abbildung 6.7, entspricht in der Funktionalität der SharkMS ohne Möglichkeit der direkten Interaktion mit einem Anwender. Eine vollständige Implementierung steht noch aus. Die Local Station läuft aber auf einem PC. Als Protokolle sind eher drahtlose Protokolle vorgesehen.

Die SharkLS kann Basis für ortsbasierte Dienste sein. Wird ein solches System an einen bestimmten Ort installiert, so kann es mittels Bluetooth mit vorbeikommenden SharkMS Wissen austauschen. Shark erlaubt die Aushandlung eines Wissensaustausches anhand von Interessen. Wird nun ein Kurzstreckenprotokoll wie z. B. Bluetooth benutzt, so ist

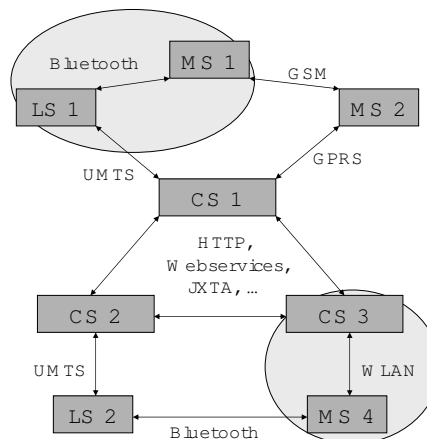


Abbildung 6.8: Beispiel-Topologie von Shark Engines

das Wissen nur in einem eingeschränkten Umfeld beziehbar. Man kann daher auch von ortsbasiertem Wissen sprechen. Ihr Wissen selber kann die SharkLS von einer SharkCS erhalten, die ihr regelmäßig mittels GSM oder UMTS „begegnet“.

Die andere Funktion des Sensors beschränkt sich auch hier auf das Entdecken neuer Bluetooth-Geräte in Senderreichweite.

6.2.4 Mögliche Topologien

Dank der unterschiedlichen Shark Stations sind ganz unterschiedliche Topologien denkbar. In der Abbildung 6.8 sind exemplarisch einige denkbare Kommunikationsbeziehungen dargestellt.

In der Mitte ist eine Dreiergruppe von SharkCS dargestellt. Sie gehören drei Knowledge Nodes. Die Central Stations laufen auf stationären PCs. Sie können jederzeit untereinander Wissen austauschen. Eine Umgebungsänderung, was in Shark immer der Beginn einer Kommunikation ist, kann sich z. B. dadurch ergeben, dass ein Mitarbeiter neues Wissen hinzufügt, das seine SharkCS wegen definierter Knowledge Ports anderen anbietet.

Sie können aber auch mit anderen Typen von Shark Station kommunizieren. So kann CS1 sein Wissen regelmäßig mit LS1 und MS2 abgleichen. Vielleicht ist es die Knowledge Node, die CS1 benutzt eine Filiale und MS2 das Handy des Filialleiters und LS1 ein stationäres System, das Mitarbeiter und Kunden über Neuigkeiten informiert.

Es ist zu sehen, dass mittels Bluetooth die LS1 mit der MS1 etwas austauscht. Vielleicht ist MS1 ein Handy eines Mitarbeiters. MS1 und MS2 tauschen Wissen über GSM aus. Vielleicht hat der Empfang des Wissens von LS1 entweder den Besitzer des Handys mit der MS1 oder die MS1 selber zum, spontanen Wissensaustausch mit MS2 angeregt.

Ähnliches erfolgt im unteren Teil der Abbildung.

Im folgenden Kapitel werden einige Anwendungsbeispiele vorgestellt. Hier sollte nur gezeigt werden, dass sich aus den unterschiedlichen Typen von Shark Stations ganz unterschiedliche Netze ergeben können, beginnend beim stabilen im Internet (z. B. die Central Stations untereinander), spontanen ortsunabhängigen Austausch (z. B. CS1-LS1, MS1-MS2, CS2-LS2) und spontanen ortsabhängigen (LS1-MS1, CS3-MS4).

Shark kann angewandt werden auf PC-basierte P2P Systeme, die auf klassischen Internettechnologien arbeiten, aber auch auf Handys, die mittels Bluetooth spontane Netze ausbilden. Es war zu sehen, dass das technisch machbar ist. Im Folgenden wird gezeigt, dass für alle Szenarien auch sinnvolle Anwendungen existieren.

6.3 Anwendungsszenarien

Zum Abschluss dieses Kapitels soll anhand von drei Anwendungsszenarien gezeigt werden, wo der Einsatz eines Shark-Systems möglich und sinnvoll ist. Der erste Bereich überrascht nicht: Knowledge Management. Die Prinzipien wurden hinreichend erläutert. Es soll anhand einer Darstellung eine mögliche konkrete Installation eines Systems beschrieben werden. Dabei wird das Konzept der Synchronisationsgruppe eingeführt, das das Konzept der Knowledge Ports benutzt, um technische Geräte synchron zu halten.

Das zweite Beispiel kommt aus dem Bereich des mobilen Marketings. Eine ausführlichere Darstellung dieses Beispiels hat einen Hauptpreis in einem Wettbewerb des Bundesministeriums für Wirtschaft und Arbeit für Geschäftsideen im Jahre 2004 gewonnen. Das letzte Beispiel liegt im Bereich Sensornetze.

Die Beispiele sollen zeigen, dass der Ansatz im Bereich Knowledge Management sehr sinnvoll eingesetzt werden kann, aber auch in anderen Bereichen möglich und sinnvoll ist.

6.3.1 Mobiles verteiltes Knowledge Management System

Synchronisationsgruppe

Gerade in wissensintensiven Prozessen verfügen die Mitarbeiter über eine Vielzahl technischer Geräte, wie Handys, PDAs, Laptops und PCs. Ein Problem besteht in der Synchronhaltung der Daten auf allen Geräten.

Es soll hier gezeigt werden, wie durch Definition einiger Knowledge Ports und passende Benennung solcher Geräte eine Synchronisation mehrerer Shark-Stations erreicht werden kann.

Die Grundidee ist folgende: Jede Knowledge Node verfügt über eine Shark-Engine. Im Referenzmodell wurde aber nie die Frage gestellt, was geschieht, wenn sie über mehr als eine verfügt. Die Frage kann dahingehend beantwortet werden, dass *alle* Geräte *einer* Knowledge Node aus der Sicht von Shark als *eines* angesehen werden. Jedes Gerät verfügt über eine Shark-Station. Alle haben aber die gleiche Bezeichnung in Shark.

Dazu werden auf jeder Shark-Station folgende Knowledge Ports definiert, die Bezeichnung der Shark-Station möge KN sein:

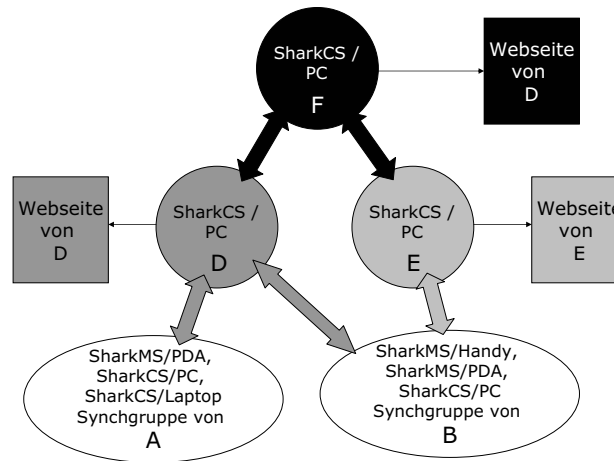


Abbildung 6.9: Sensornetze mit Shark

$$IKP = \{KN, I_{KN} \cup T_{KN}, KN\}$$

$$OKP = \{KN, I_{KN} \cup T_{KN}, KN\}$$

Die Definitionen sind weniger absurd, als sie auf den ersten Blick wirken mögen. Jede Shark-Station beschreibt damit, dass sie mit einer anderen gleichen Namens alles Wissen und Interessen austauschen will.

Geht man von nur einer Person aus, die zwar über mehrere Geräte verfügt, aber Wissen nur an einigen wenigen und dann auch nicht gleichzeitig eingibt, so realisiert diese Definition die Etablierung einer Synchronisationsgruppe. Sobald ein Handy mit einer SharkMS dem PC der Knowledge Node mit einer SharkCS begegnet, kann ein Austausch erfolgen. Das Handy würde in dem Fall unbestätigtes Wissen übertragen, die Central Station hingegen neues Wissen. Kollisionen können nur auftreten, wenn jemand zwei Central Stations betreibt, auf beiden Änderungen durchgeführt hat, die sich widersprechen, und verhindert hat, dass zwischendurch eine Synchronisation erfolgt. Dann wird ein Konflikt entstehen, den allerdings der Nutzer selber lösen kann, schließlich hat er ihn allein erzeugt.

Eine Beispiellarchitektur

Dieser Abschnitt zeigt eine mögliche Architektur zur Unterstützung der Struktur, die bereits im Kapitel 4 zur Motivierung diente, dass Gruppen als Meta-Peers interpretiert werden können, siehe Abbildung 4.1. Aus Gründen der Übersicht wurde die Person G nicht dargestellt. Anstelle der abstrakten Knowledge Ports werden hier die Geräte und die Shark-Station Typen genannt, siehe Abbildung 6.9.

Von jedem Mitarbeiter wird angenommen, dass er über mehrere Geräte verfügt, wie Handys, PDA, Laptop, PC und Ähnliches. Diese werden zu einer Synchronisationsgrup-

pe zusammengeschlossen. Nach außen und vor allem auch gegenüber den Anwendern erscheinen sie, wie gleiche Geräte, im Sinne, dass sie über die gleichen Daten verfügen. Wird Bluetooth eingesetzt, so genügt es, dass die Geräte regelmäßig in Senderreichweite gelangen, um eine Synchronisation sicherzustellen.

Für die Gruppen und das Unternehmen wird jeweils ein dedizierter PC mit einer Central Station vorgeschlagen. Sie sind wenigstens im Intranet erreichbar und können mit den PCs und Laptops der Mitarbeiter Wissen austauschen. In dringenden Fällen können diese Systeme auch die mobilen Geräte der Mitarbeiter erreichen, wenn diese über GSM oder UMTS-Karten verfügen. So könnten Knowledge Ports für besonders wichtige Mitteilungen definiert werden, die dann über alle möglichen Protokolle versuchen, das Wissen so schnell als möglich zu versenden.

Durch diese Architektur ist das Gruppenwissen nicht nur eine emergente Eigenschaft des Wissens der Gruppenmitglieder, sondern liegt in Kopie tatsächlich auf einem Rechner vor. Das hat einen positiven Nebeneffekt. Wenn man ein Intranet weiter pflegen wollte, so läge das Wissen der Gruppen und Super-Gruppen jeweils auf eigenen Systemen vor und könnte über einen Webserver angeboten werden. Das bedeutet aber nicht, dass die Wissensflüsse über die Webfrontends erfolgen müssten. Das würde über die Shark-Infrastruktur erfolgen. Dieses Vorgehen böte sich vor allem in einer Migrationsphase an, wenn man ein P2P-KMS parallel zu einem herkömmlichen Webbasierten Intranet betreiben wollte.

Das möge an Diskussion zu diesem Thema genügen. Die anderen Konzepte und Prinzipien wurden im Rest der Arbeit erschöpfend behandelt.

6.3.2 Mobiles und Virales Marketing

In diesem Abschnitt besteht die Gefahr, sich ein Stück aus dem wissenschaftlichen Bereich zu entfernen. Er kann ohne Verlust an Informationen übersprungen werden. Dieses Anwendungsfeld soll trotzdem präsentiert werden, weil es die Breite der Möglichkeiten für mobile Lösungen, die auf semantischen Strukturen basieren, aufzeigt.

Tatsächlich hat sich gezeigt, dass das Modell Shark in einem derzeit recht neuen Bereich des Marketings, nämlich dem mobilen und dort vor allem dem viralen Marketing, einsetzen lässt.

Mobiles Marketing lässt sich grob in der Formel zusammenfassen: Es ist Marketing, das mit dem Handy zu tun hat. Bekannt sind sicherlich die (illegalen) Werbe-SMS, die in den letzten Jahren zur Qual jedes Handybesitzers wurden.

Ein weiterer Trend im Marketing ist *virales Marketing*. Auch im Bereich Marketing hat man herausgefunden, dass die beste Werbung darin besteht, wenn man einen Tipp unter Freunden bekommt. Fragen unter Freunden und Bekannten wie die folgenden sind nicht unüblich: „Ich brauche X. Was würdest du mir empfehlen?“ X kann jedes denkbare Produkt sein. Unternehmen würden sehr viel dafür geben, wenn in der Antwort ihr Produkt empfohlen würde.

Tatsächlich gibt es schon heute Aktionen, in denen Menschen aus einer Zielgruppe ausgewählt werden, denen dann neue Produkte geschenkt werden, lediglich mit der Bitte verbunden, sie zu benutzen. Die Menschen werden dahingehend ausgesucht, dass sie

Meinungsführer in einem Teil der Zielgruppe sind. Das ist virales Marketing: die bestehenden informellen Netze zwischen Menschen zur Verbreitung von Werbebotschaften nutzen. Das ist nicht neu. Als Stichwort sei Tupperware genannt. Hier wird Plastikgeschirr ausschließlich in privaten „Parties“ vertrieben.

Aus Shark lässt sich eine Plattform ableiten, die sowohl das mobile und virale Marketing unterstützt als auch den Verkauf elektronischer Güter über das Handy.³

Die Grundidee ist einfach: Handys werden mit der Shark Mobile Station ausgerüstet. Wenn sich die Handybesitzer z. B. für Musik interessieren, so können sie mit dem Handy auf eine Webseite surfen und dort *Interessenprofil* herunterladen. Das sind Empfangsinteressen. Interessenprofile können gemischt werden, was technisch auf ein Topic Map Merge hinausläuft. So kann jemand sein Interesse für *HipHop* mit einem Interesse an *Gothic* mischen. Die Nutzer müssen sich nicht identifizieren, wenn sie die Profile laden.

In Musikclubs oder Ähnlichem werden Shark Local Station, die mit Bluetooth ausgerüstet sind an markanten Punkte positioniert und mit Werbung für Musik gefüllt. Aus technischer Sicht werden MP3 Files und Werbetexte mit Verweisen auf Seiten des mobilen Internets hinterlegt und einem Musikthema zugeordnet.

Sobald eine Shark Mobile Station in die Nähe einer SharkLS kommt, erfolgt mittels KEP eine Aushandlung. Wird ein Austauschinteresse ermittelt, wird die Werbung von der SharkLS an die SharkMS mittels Bluetooth übertragen. Dabei entstehen für keine Seite Kosten. Die Werbung kann nun angesehen werden, das Musikfile angehört werden usw.

Wenn der Nutzer das zulässt, wird ein OKP auf seinem Handy erzeugt, der alle empfangene Werbung weitergibt. Trifft nun ein Anwender einen Freund, der ebenfalls über das System verfügt, so ist es wahrscheinlich, dass beide ähnliche Interessen und damit ähnliche IKPs haben. Das kann dazu führen, dass sie sich wechselseitig Werbung übermitteln, die sie von SharkLS empfangen haben. Das ist ein viraler Effekt. Da der Austausch immer mittels Bluetooth erfolgt, also maximal über 10 Meter bei Handys, ist er lokal begrenzt. Shark baut hier die Mund-zu-Mund-Propaganda nach. Das System übermittelt keinen Spam, denn dieser wird mit großer Sicherheit vom ersten Empfänger gelöscht, dem es auch peinlich wäre, Emitter von Spam zu sein. Das gezielte Einspielen von Spam bedarf einer größeren Investition, denn die Nachrichten werden ausschließlich über Bluetooth übermittelt. Man muss entweder Bluetooth-Sender aufbauen und möglichst flächendeckend verteilen oder man muss eine große Anzahl an Menschen mit Handys an passende Plätze bringen, die den Spam verteilen. Beides ist wirtschaftlich völlig unattraktiv. Abbildung 6.10 illustriert das System.

In der Abbildung ist ein Musik-Label zu sehen, das seine neuen CDs bewerben will. Die Knowledge Ports sind aus Gründen der Übersichtlichkeit nicht eingezeichnet. Beispielfhaft ist ein Thema „HipHop“ dargestellt, dessen Occurrence auf ein MP3-File zeigt. Dieses kann z. B. einen Teil der zu bewerbenden CD enthalten. Üblicherweise werden noch weitere Werbungen an das Thema gebunden, so vor allem eine Webseite, die für Handys

³Für die im Folgenden präsentierte Idee wurde im Mai 2004 ein Hauptpreis im Gründerwettbewerb „Mit Multimedia erfolgreich starten“ des Bundesministeriums für Wirtschaft und Arbeit vergeben. Im Folgenden soll nur das technische Konzept, nicht die kommerziellen Überlegungen vorgestellt werden.

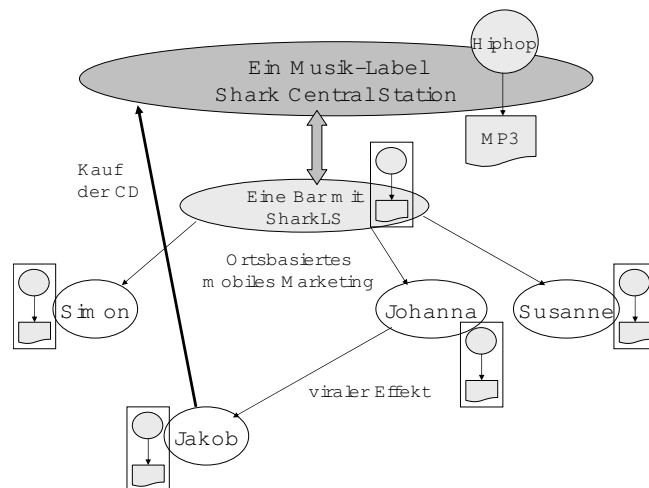


Abbildung 6.10: Mobiles und virales Marketing mit Shark

optimiert ist und auf der weitere Informationen zu erhalten sind und vor allem die CD gekauft werden kann.

In der Abbildung ist ein Club dargestellt, in dem eine Shark Local Station installiert ist. Die SharkLS ist Mitglied der Gruppe des Musik-Labels. Angenommen, in dem Club wird regelmäßig „HipHop“ aufgelegt, so herrscht dort ein Interesse daran. Aus diesem Grund wird regelmäßig Werbung dazu auf die SharkLS übertragen. Diese übermittelt die Werbung an die Shark Mobile Stations der Kunden (z. B. Simon, Johanna, Susanne).

Jakob war nie in dem Club. Allerdings trifft er in der Abbildung irgendwann Johanna. Allein durch die räumliche Nähe und deren beidseitigem Interesse an Hiphop gelangt die Werbung nun auch zu Jakob (viraler Effekt), der sich schlussendlich sogar zu einem Kauf entscheidet. Das hat allerdings weder technisch noch konzeptuell etwas mit Shark zu tun.

6.3.3 Sensornetze

Unter Sensoren sollen kleine Geräte verstanden werden, die bestimmte Werte der Umgebung messen, diese Werte speichern und bei Bedarf weitergeben können.

Für die Zwecke dieser Arbeit sind insbesondere Sensoren von Interesse, die ihre Daten drahtlos an andere abgeben und die nur über eine beschränkte Sendereichweite verfügen.

Solche Netzwerke sind Gegenstand der Forschung. Bekannt sind Begriffe wie *intelligent dust*, die unterstreichen sollen, dass es tatsächlich um miniaturisierte Sender geht. Aber auch so genannte *body area networks* sind von Interesse, die Körperwerte z. B. von Kranken messen, um Alarm schlagen zu können. Auch hier findet zwischen den einzelnen Sensoren konzeptuell ein drahtloser Datenaustausch über kurze Strecken statt.

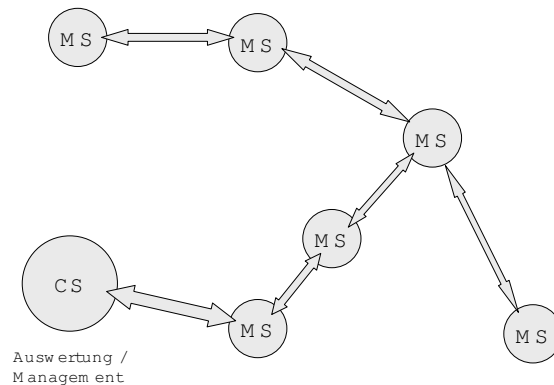


Abbildung 6.11: Sensornetze mit Shark

Besonders aber Systeme, die *intelligent dust* ähneln sind von herausragendem Interesse. Ein exemplarisches Einsatzgebiet soll das Potenzial solcher Systeme illustrieren. Angenommen man will Messungen im Grundwasser vornehmen, und zwar an unterschiedlichen Stellen einer Wasserader. Ohne Nanotechnik bleibt nur der Weg von mehreren Bohrungen an verschiedenen Stelle, um dort jeweils Messungen vorzunehmen. Unglücklicherweise beeinträchtigen solche Bohrungen meist schon die Messungen deutlich.

Eine Alternative besteht in kleinsten Sensoren, die biologisch abbaubar sind. Diese könnten in die Wasserader geleitet werden und bilden dort durch Diffusion eine Wolke aus. Die Sensoren haben keine große Sendereichweite, doch können sie ihre Daten von Sensor zu Sensor weitergeben, so dass schlussendlich im gesamten Netz alle Daten vorliegen. Es genügt, den Kontakt zu einem Sensor herstellen zu können, um alle Daten zu ermitteln.

Werden die Senoren leistungsfähiger und können eine ganze Bandbreite von Daten ermitteln und bestünde der Wunsch, das gesamte Netz dahingehend administrieren zu wollen, welche Daten gesammelt werden sollen, würde die Idee von Shark Hilfe leisten können, siehe Abbildung 6.11.

Jeder Sensor könnte mit einer kleinen Shark Mobile Station versehen werden. Die Wissensbasis dient in dem Falle der Speicherung der gemessenen Daten. Alle Sensoren werden zu einer Synchronisationsgruppe zusammengeschaltet, siehe Abschnitt 6.3.1. Das bedeutet, dass bei jeder Begegnung von zwei Sensoren eine Synchronisation stattfindet. Es ist also möglich, zu administrieren, welche Themen synchronisiert werden sollen. Bei entsprechend gesetzten Rechten würde es genügen, wenn nur ein Mitglied der Synchronisationsgruppe die Ports entsprechend ändern würde. Alle anderen würden die neuen Ports übernehmen und entsprechend handeln.

Es mag auf Anhieb nicht sinnvoll erscheinen, mit *großen semantischen Ansätzen* auf ein scheinbar so kleines Problem loszugehen. Dazu muss daran erinnert werden, dass eine Ontologie sehr klein sein kann. Angenommen, ein Sensor kann die Luftfeuchtigkeit und die Temperatur messen. Er hätte damit potenziell zwei Konzepte in der Ontologie, die definieren, was er messen soll und denen er Messwerte zuordnen könnte. Eine solche Ontologie benötigt nicht nennenswert mehr zusätzlichen Speicher, verglichen mit den Daten, die sowieso gemessen und gespeichert werden müssen.

Die Algorithmen zur Extraktion und Assimilation dürfen denkbar einfach werden und auch die Funktionen zum Vergleich der Interessen laufen auf einen Stringvergleich hinaus. Das wäre natürlich nur die naivste Herangehensweise. Aus Performancegründen lassen sich die wenigsten Strings auch durch Zahlen repräsentieren, was den benötigten Speicherplatz und die Algorithmen noch weiter vereinfacht.

Die Central Station in diesem Netz würde konzeptuell über deutlich mehr Kapazitäten verfügen und der Auswertung der Netzdaten und dessen Administration dienen.

6.4 Zusammenfassung

Dieses Kapitel diente der Abrundung der Arbeit. Es stellte Knowledge Exchange Protocol vor, das den Anforderungen von Shark genügt. Es wurde anhand von drei unterschiedlichen Möglichkeiten diskutiert, wie eine Shark-Engine als Shark-Station implementiert werden kann. Es wurden einige Plattformen genannt, auf denen (Teil-) Implementierungen des Systems erfolgten. Es war zu sehen, dass sich ganz unterschiedliche Netze je nach Implementierungsplattform ausbilden können: Das Prinzip Shark kann sowohl auf Internettechnologien abgebildet werden als auch auf mobile Geräte, die mittels Bluetooth miteinander kommunizieren. Es wurden drei Anwendungsszenarien aus verschiedenen Bereichen betrachtet, in denen Shark-Plattformen Einsatz finden können.

Damit ist das Ende der Arbeit nahezu erreicht. Es folgen eine Zusammenfassung und ein Ausblick.

7 Zusammenfassung und Ausblick

In der Einleitung dieser Arbeit wurde das Problemfeld im Knowledge Management beschrieben: Empirische Studien zeigen, dass die eingesetzten Knowledge Management Systeme nicht die Akzeptanz erfahren würden, wie erhofft. Es gibt eine Reihe von Kritiken an existierenden KMS, die zum allergrößten Teil allerdings technisch motiviert sind. In dieser Arbeit wurde die These vertreten, dass die Akzeptanzprobleme nicht durch Technik zu erklären sind, sondern durch das gewählte Paradigma. Die meisten im Einsatz befindlichen KMS folgen dem Client-Server-Modell und treten Anwendern als „entfernter“ Server entgegen, der mit Wissen versorgt werden muss und von dem Wissen bezogen werden kann. Das Problem stellt das Gefühl des „Entfernten“ dar. Wie Studien zeigten, liegt das meiste explizite Wissen noch heute in den Foldern und Mailboxen der Anwender, auch bei eingeführten Firmen-KMS. Offensichtlich haben Anwender keine Probleme damit, „ihr“ Wissen einem Rechner anzuvertrauen. Es muss nur der „eigene“ sein. Die KM-Theorie und empirische Studien zeigen auch, dass der bevorzugte Austausch von Wissen innerhalb informeller Netze und direkt von Mensch zu Mensch erfolgt, was man auch als P2P-Austausch bezeichnen kann. Dieser direkte Austausch kann auch per E-Mail erfolgen. Die schlechte Akzeptanz mancher KMS kann also nicht allgemein durch mangelnde Akzeptanz von Technik erklärt werden.

In dieser Arbeit wird davon ausgegangen, dass das Kernproblem darin liegt, dass die meisten KMS auf dem Client-Server-Paradigma aufbauen und nicht auf einem P2P-Ansatz. Daher wurde die Aufgabe gestellt, ein P2P-KMS zunächst als Referenzarchitektur zu konzipieren. Dabei war festzustellen, dass die KM-Theorie aktuell noch recht kurz greift. Sie erklärt, *dass* ein Wissensaustausch zwischen Objekten auf der ontologischen Dimension von Nonaka und Takeuchi stattfindet, sie erklärt aber nicht, *wie* das erfolgt, insbesondere klärt sie nicht, wie Wissensaustausche auf den P2P-Austausch zwischen Menschen zurückzuführen sind, die doch allseits beobachtet werden. In Abschnitt 2.5 wurde basierend auf der Theorie von Nonaka und Takeuchi erklärt, wie Wissensflüsse in Unternehmen auf P2P-Austausche von Individuen zurückzuführen sind. Das war die Basis für die Entwicklung der Referenzarchitektur Shark.

Bei Shark wurde allerdings nicht stehen geblieben. Einmal wurde das Grundkonzept der autonomen Peers von Shark bereits im Kapitel 3 allgemeiner präsentiert und später wird es in den Kapiteln 5 und 6 zu einer Architektur konkretisiert. Es ist zu vermuten, dass sich die hier beschriebenen Prinzipien autonomer Peer, die kontextbasiert Wissen bzw. Daten austauschen, auch auf andere Anwendungsgebiete übertragen lassen. Beispiele wurden im Abschnitt 6.3 skizziert.

Die Arbeit hat insgesamt einen großen Bogen geschlagen, von einem allgemeinen Peer-Modell über die Soziologie, künstliche Intelligenz und das Knowledge Management hin zu einer Referenzarchitektur für ein P2P-KMS und der Diskussion einer Umsetzung und

über Einsatzgebiete.

Nur durch diesen Bogen gelang es aber, zunächst über die rein technischen Fragestellungen hinauszugelangen, danach eine Reihe von Technologien vor allem aus dem Bereich KI, aber auch dem Content-Management zu betrachten, um all das wieder für ein P2P-KMS nutzbar zu machen. Ein reines Verengen auf eine technische Fragestellung hätte nicht ausgereicht, um das Ziel, das am Anfang der Arbeit formuliert wurde, zu erreichen. Der große Bogen führte allerdings auch dazu, dass an vielen Stellen Konzepte nur grob diskutiert werden konnten.

Die Arbeit hatte drei Schwerpunkte. Der eine war die Referenzarchitektur Shark. Der zweite war die Umsetzung der Prinzipien der Kontextualisierung mittels Topic Maps. Es war zu sehen, dass sich der Standard ausgezeichnet eignet, um ein P2P-KMS zu unterstützen, und dass es ohne größere Probleme gelingt, den Fragmentaustausch zwischen Topic Map Engines, die Teil einer Shark-Engine sind, zu beschreiben. Die in Kapitel 5 beschriebenen und definierten Funktionen sind auch unabhängig von einem P2P-KMS für andere Zwecke nutzbar. Der letzte Schwerpunkt lag in einem leichtgewichtigen Austauschprotokoll für Wissen in Shark. Es wurde das Knowledge Exchange Protocol vorgestellt. Um den Rahmen der Arbeit nicht zu sprengen, wurden weitere Implementierungsdetails nur angerissen und auch die möglichen bzw. bereits realisierten Anwendungsfälle einer Implementierung von Shark wurden nur skizziert.

Die Arbeit stellte keine alternativen Architekturen oder technische Entscheidungen nebeneinander. So hätte man wahlweise auch untersuchen können, ob sich z. B. RDF oder OWL anstelle von Topic Maps für Shark eignet. Die Antwort ist ja. Beide Ansätze bewegen sich im Bereich semantischer Netze und bei der Umsetzung der Kontextualisierung in Topic Maps wurde zwar tief auf Topic Maps eingegangen, allerdings waren es im Wesentlichen Operationen über Mengen bzw. Graphen, die schlussendlich zu einer passenden Definition führten. Ähnliche Überlegungen sind für RDF/OWL möglich. Jenseits der semantischen Netze wurden keine Ansätze untersucht. Es wird vermutet, dass sich passende finden lassen, wurde doch gezeigt, dass auch in Informationssystemen und in der Prädikatenlogik das Konzept Kontext erfolgreich integriert wurde.

Es gibt heute bereits eine Anzahl von P2P-Plattformen, auf deren Basis P2P-Anwendungen entwickelt werden können. Die meisten Plattformen entstanden parallel zur Arbeit an Shark. Es wurde JXTA genannt, aber es wurde nicht gezeigt, ob und wie eine Umsetzung von Shark erfolgen könnte. Anhand der ähnlichen Grundideen scheint eine Umsetzung keine wesentlichen Probleme aufzuwerfen.

P2P-Systeme scheinen eine wachsende Bedeutung zu erlangen. Basis für diese Vermutung sind zwei Trends, die unter den Schlagworten *Disappearing Computing* / *Ambient Intelligence* einerseits und Nanotechnik andererseits umschrieben werden können, aber lediglich zwei Seiten einer Medaille darstellen: Technik wird kleiner und billiger. Das erlaubt, sie in Alltagsgegenstände zu integrieren. Technik verschwindet dadurch scheinbar, gleichzeitig werden die Gegenständen „intelligenter“. Für reale Anwendungsfälle werden sich diese intelligenten Gegenstände verbinden müssen, sei es z. B. wenn eine „intelligente“ Wand private Urlaubsfotos eines Anwenders bezieht, um sie seinen Freunden im Raum zu präsentieren. In Zukunft wird eine wachsende Zahl kleiner Geräte zusammen bisher übliche oder auch komplexere Anwendungsszenarien ausführen können.

Ein Effekt kann ebenfalls sein, dass persönliche Daten auch physisch zu persönlichem Eigentum werden. Es ist nicht allzu visionär, zu vermuten, dass die Nachfolger der heutigen Memorysticks in der Lage sein werden, die wesentlichen Daten eines Menschen, die er benötigt, seien es berufliche und private, sicher zu speichern. Wenn sie in irgendeiner Form zum festen Equipment der Menschen werden, z. B. als Teil des Handys oder der Kleidung, so könnte jeder Mensch tatsächlich jederzeit sein gesamtes digitales explizites Wissen bei sich haben, ohne die Notwendigkeit, einen schweren Laptop mitzuführen. Diese Daten könnten in einer passenden Umgebung verwendet werden, zum Beispiel wenn man ein Büro betritt und einen Arbeitsplatz benutzt. Datenreplikation wird nur noch zur Datensicherung notwendig sein, aber nicht mehr, um Laptop, PC und andere Geräte synchron zu halten. Eine solche Welt stellt eine Welt von miniaturisierten Peers dar. Mit dem Blick auf eine solche Welt wurden die Konzepte von Shark entwickelt und werden nach Abschluss dieser Arbeit fortgesetzt.

7 Zusammenfassung und Ausblick

Literaturverzeichnis

- [ABG04] AUER, Sören ; BUREK, Patryk ; GRAWE, Tonio: Knowledge Engineering for IT-based Services. In: *Proceedings of the 4th International Conference on Knowledge Management (I-KNOW '04)*, Graz, 2004, S. 213–224
- [Ahm03] AHMED, Kal: TMSHare - Topic Map Fragment Exchange in a Peer-to-Peer-Application. In: *Proceedings of XML Europe 2003, London*, 2003
- [AN04] ATZENBECK, Claus ; NÜRNBERG, Peter J.: Approaching Structure Interoperability. In: *Proceedings of the 4th International Conference on Knowledge Management (I-KNOW '04)*, Graz, 2004, S. 269–278
- [Ash56] ASHBY, Ross: *An Introduction to Cybernetics*. Chapman & Hall, London, 1956
- [Bar03] BARTA, Robert: AsTma? Language Definition, v0.3. Version: 2003. <http://astma.it.bond.edu.au/astma%3F-spec.dbk>. – language description. – Online-Ressource
- [Bar04] BARTA, Robert: Virtual and Federated Topic Maps. In: *Proceedings of XML Europe, Amsterdam*, 2004
- [BBC02] BONIFACIO, Matteo ; BOUQUET, Paolo ; CUEL, Roberta: Knowledge Nodes: The Building Blocks of a Distributed Approach for Knowledge Management. In: *Proceedings of the 2nd International Conference on Knowledge Management (I-KNOW'02)*, Graz, 2002, S. 191–200
- [BC03] BONIFACIO, Matteo ; CAMUSSONE, Pierfranco: Managing the KM-Trade-Off: Knowledge Centralization *versus* Distribution. In: *Proceedings of the 4th International Conference on Knowledge Management (I-KNOW '03)*, Graz, 2003, S. 14–21
- [BDS02] BOUQUET, P. ; DON, A. ; SERAFINI, L.: ConTeXtualizedlocal ontology specification via ctxml. In: *Proceedings of AAAI workshop on Meaning Negotiation, Edmonton/Alberta, Canada, 2002*, 2002
- [BGGB01] BOUQUET, P. ; GHIDINI, C. ; GIUNCHIGLIA, F. ; BLANZIERI, E.: Theories and uses of context in knowledge representation and reasoning / Instituto Trentino di Cultura. 2001 (#0110-28). – technical report

- [Bib93] BIBEL, Wolfgang: *Wissensrepräsentation und Inferenz*. Vieweg Verlag, Braunschweig, 1993
- [BM93] BUVAČ, S. ; MASON, Ian M.: Propositional logic of context. In: *Proceedings of the 11th National Conference on Artificial Intelligence, Menlo Park*, American Association for Artificial Intelligence, 1993, S. 412–419
- [BM96] BUVAČ, S. ; MASON, Ian M.: Quantificational logic of context. In: *Proceedings of the 13th National Conference on Artificial Intelligence, Portland/Oregon*, American Association for Artificial Intelligence, 1996
- [BP96a] *Kapitel* Misuse and nonuse of knowledge-based systems: The past experiences revisited. In: BRÉZILLON, P. ; POMEROL, J.-Ch.: *Implementing Systems for Supporting Management Decisions*. Chapman and Hall, 1996, S. 44–60
- [BP96b] BRÉZILLON, P. ; POMEROL, J.-Ch.: User acceptance of interactive systems: Lessons from knowledge based and decisions support. In: *International Journal of Failure & Lessons Learned in Information Technology Management* 1 (1996), S. 67–75
- [Bre99a] BREZILLON, Patrick: Context in Artificial Intelligence: I. A survey of the literature. In: *Computer & Artificial Intelligence* 18 (4) (1999), S. 321–340
- [Bre99b] BREZILLON, Patrick: Context in Artificial Intelligence: II. Key elements of contexts. In: *Computer & Artificial Intelligence* 18 (5) (1999), S. 425–446
- [BS02] BORGIDA, A. ; SERAFINI, L.: Distributed description logics. In: HORROCKS, I. (Hrsg.) ; TESSARIS, S. (Hrsg.): *Proceedings of the International Workshop on Description Logics (DL2002)*, Toulouse, 2002
- [BS05] BARTA, R. ; SALZER, G.: The Tau Model – Formalizing Topic Maps. In: *Proceedings of the Second Asia-Pacific Conference on Conceptual Modelling (APCCM 2005)*, Newcastle, Australia, Australian Computer Society, 2005
- [CC] CYC CORP, USA: *CYC Project*. <http://www.cyc.com/>
- [CC04] CRISTANI, Matteo ; CUEL, Roberta: A Comprehensive Guidline for Building a Domain Ontology from Scratch. In: *Proceedings of the 4th International Conference on Knowledge Management (I-KNOW '04)*, Graz, 2004, S. 205–212
- [Ceb05] CEBULLA, Agnes: *Ein verteilter Thesaurus basierend auf Topic Maps und LDAP (Diplomarbeit, TU Berlin)*. 2005

- [CG04] CANZANO, Daniela ; GRIMALDI, Michele: Knowledge Management and Collaborations: Knowledge Strategy and Processes in the Knowledge Networks. In: *Proceedings of the 4th International Conference on Knowledge Management (I-KNOW '04)*, Graz, 2004, S. 175–183
- [CK93] CAHOUR, B. ; KARSENTY, L.: Context of dialogue: a cognitive point of view. In: *Proceedings of the IJCAI-93 Workshop on Using Knowledge In Its Context*, Chambéry, 1993, S. 20–29
- [DC04] DIAZ, Alicia ; CANALS, Gerome: Supporting Knowledge Sharing in a Community with Divergence. In: *Proceedings of the 4th International Conference on Knowledge Management (I-KNOW '04)*, Graz, 2004, S. 302–310
- [DIN87] DIN: *Erstellung und Weiterentwicklung von Thesauri*, DIN 1463. 1987
- [Din91] DINSMORE, J.: *Partitioned Representation*. Kluwer Academic Publishers, 1991
- [DN05] DURUSAU, Patrick ; NEWCOMB, Steve: ISO 13250-5: Topic Maps – Reference Model / ISO/IEC JTC 1/SC34. Version: 02 2005. <http://www.isotopicmaps.org/TMRM/TMRM-5.0/TMRM-5.0.html>. – Forschungsbericht. – Online-Ressource
- [Dör89] DÖRNER, Dietrich: *Die Logik des Misslingens*. Rowohlt Taschenbuch Verlag, 1989
- [DP00] DAVENPORT, Thomas H. ; PRUSAK, Laurence: *Working Knowledge*. Havard Business School Press, Boston, Massachusetts, 2000
- [Dun72] DUNPHY, D.: *The Primary Group*. Appleton-Century-Crofts, 1972
- [Dus04] DUSTDAR, Schahram: Reconciling Knowledge Management and Workflow Management Systems: The Activity-based Knowledge Management Approach. In: *Proceedings of the 4th International Conference on Knowledge Management (I-KNOW '04)*, Graz, 2004, S. 457–464
- [Eco94] ECO, Umberto: *Die Suche nach der vollkommenen Sprache*. Deutscher Taschenbuch Verlag (dtv), 1994
- [EM93] EDMONDSON, W. H. ; MEECH, J. F.: A model of context for human-computer interaction. In: *Proceedings of the IJCAI-93 Workshop on Using Knowledge in its Context*, Chambéry, 1993, S. 31–38
- [EM01] EKBIA, Hamid R. ; MAGUITMAN, Ana G.: Context and Relevance: A Pragmatic Approach. In: *Lecture Notes in Computer Science* 2116 (2001), S. 156–169

- [EMPW97] *Kapitel* Bernd Mahr: Gegenstand und Kontext - Eine Theorie der Auffassung. In: EYFERTH, K. ; MAHR, B. ; POSNER, R. ; WYSOTZKI, F.: *KIT-Report 141: Prinzipien der Kontextualisierung*. TU Berlin, Germany, 1997
- [Etz91] ETZIONI, A.: The good polity. Can we design it? In: *American Behavioral Scientist* 3 (1991), S. 549–563
- [Ezh92] *Kapitel* Contextual systems: Is it a way of universal expert system development. In: EZHKOVA, I. V.: *General systems*. New Jersey, USA : G. Klir, 1992
- [Fau85] FAUCONNIER, G.: Mental Spaces: aspects of meaning construction in natural language. In: *MIT Press* (1985)
- [FFK03] FUHR, David ; FUCHS-KITTOWSKI, Frank: Root, Net and Octopus: Case Studies into the CoP Theorie-Practice Gap. In: *Proceedings of the 4th International Conference on Knowledge Management (I-KNOW '03)*, Graz, 2003, S. 22–27
- [fip02] FIPA Communicative Act Library Specification / Foundation for intelligent physical agents. 2002. – FIPA standard
- [FLM97] FININ, T. ; LABROU, Y. ; MAYFIELD, J.: KQML as an agent communication language. In: *Software Agents, MIT Press, Cambridge* (1997)
- [FPW04] FORZI, Tomaso ; PETERS, Meikel ; WINKELMANN, Katrin: A Framework for the Analysis of Knowledge Management within Distributed Value-creating Networks. In: *Proceedings of the 4th International Conference on Knowledge Management (I-KNOW '04)*, Graz, 2004, S. 432–439
- [Fr85] FRSTER, H v.: *Sicht und Einsicht*. Vieweg und Sohn, Braunschweig/Wiesbaden, 1985
- [Gar03] GARSHOL, Lars M.: TOLOG. Version: 2003. <http://www.ontopia.net/topicmaps/materials/tolog.html>. – language description. – Online-Ressource
- [GB05] GARSHOL, Lars M. ; BARTA, Robert: Topic Maps Query Language / ISO/IEC JTC 1/SC34. 2005. – working draft
- [GG01] GHIDINI, C. ; GIUNCHIGLIA, F.: Local Models Semantics or Contextual Reasoning = Locality + Compatibility. In: *Artificial Intelligence* 127 (2) (2001), 04, S. 221–259
- [GHJV95] GAMMA, Erich ; HELM, Richard ; JOHNSON, Ralph ; VLISSIDES, John: *Design Patterns – Elements of Reusable Object-Oriented Software*. Addison Wesley, 1995

- [Giu93] GIUNCHIGLIA, F.: Contextual reasoning. In: *Proceedings of the Workshop on Using Knowledge in its Context, Chambery, 1993*, S. 39–49
- [GM05] GARSHOL, Lars M. ; MOORE, Graham: ISO 13250-2: Topic Maps – Data Model / ISO/IEC JTC 1/SC34. 2005. – final committee draft
- [GPD04] GUIZZARDI, Renata S. S. ; PERINI, Anna ; DIGNUM, Virginia: Providing Knowledge Management Support to Communities of Practice through Agent-oriented Analysis. In: *Proceedings of the 4th International Conference on Knowledge Management (I-KNOW '04)*, Graz, 2004, S. 320–328
- [Gru93] GRUBER, Thomas R.: Towards Principles for the Design of Ontologies Used for Knowledge Sharing. In: *International Journal of Human and Computer Studies* 43(5/6) (1993), S. 907–928
- [GS94] GIUNCHIGLIA, F. ; SERAFINI, L.: Multilanguage hierarchical logics or: how we can do without model logics. In: *Artificial Intelligence* 65 (1) (1994), S. 29–70
- [GU04] GRONAU, Norbert ; USLAR, Mathias: Integration Knowledge Management and Human Resources via Skill Management. In: *Proceedings of the 4th International Conference on Knowledge Management (I-KNOW '04)*, Graz, 2004, S. 135–142
- [GW04] GRONAU, Norbert ; WEBER, Ezard: Defining an Infrastructure for Knowledge Intensive Business Processes. In: *Proceedings of the 4th International Conference on Knowledge Management (I-KNOW '04)*, Graz, 2004, S. 424–431
- [Haa91] HAARMANN, Harald: *Universalgeschichte der Schrift*. 2. überarbeitete Auflage. Campus Verlag, Frankfurt/Main, New York, 1991
- [Ham03] HAMMEL, Holger: *Ortsbasierte Dienste mit Topic Maps (Diplomarbeit, TU Berlin)*. 2003
- [Hef01] HEFLIN, J. D.: *Towards the Semantic Web: Knowledge Representation in a Dynamic, Distributed Environment*, Department of Computer Science. University of Maryland., Diss., 2001. <http://www.cs.umd.edu/projects/plus/SHOE/pubs/heflin-thesis.pdf>. – Online–Ressource
- [HS04] HEFKE, Marke ; STOJANOVIC, Ljiljana: An Ontology-based Approach for Competence Bundling and Composition of ad-hoc Teams in an Organisation. In: *Proceedings of the 4th International Conference on Knowledge Management (I-KNOW '04)*, Graz, 2004, S. 126–134

- [ISO86a] ISO: *Guidelines for the establishment and development of monolingual thesauri, ISO 2788*. 1986
- [ISO86b] ISO: Information processing – Text and office systems – Standard Generalized Markup Language (SGML) / ISO. 1986. – ISO 8879:1986
- [Kip03] KIPP, Neill A.: A Mathematical Formalism for the Topic Maps Reference Model (version 4.x) / ISO/IEC JTC 1/SC34. Version: 10 2003. <http://www.isotopicmaps.org/tmrm/0441.htm>. – for information. – Online-Ressource
- [Kon95] KONKINOV, B.: A Dynamic Approach to Context Modelling. In: *Working Notes of the ICAI-95 Workshop on Modelling Context in Knowledge Representation and Reasoning, Montreal, 1995*
- [Len98] LENAT, Doug: The Dimensions of Context-Space / CYC CORP (www.cyc.com). 1998. – Forschungsbericht. <http://www.cyc.com/context-space.rtf.doc.txt>
- [Luh75] LUHMANN, N.: *Soziologische Aufklärung*. Westdeutscher Verlag, 1975
- [Luh84] LUHMANN, N.: *Soziale Systeme – Grundriß einer allgemeinen Theorie*. Suhrkamp, Frankfurt a.M., 1984
- [Mai04] MAICHER, Lutz: Subject Identification in Topic Maps in Theory and Practice. In: *Berliner XML-Tage 2004*, 2004
- [McC93] MCCARTHY, John: Notes on formalizing context. In: *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence, Chambéry, 1993*, S. 555–560
- [McC95] MCCARTHY, John: A logical AI approach to context / Computer Science Department, Stanford University, Stanford, CA. 1995. – Forschungsbericht
- [Mic] MICROSYSTEMS, SUN: *JXTA*. <http://www.jxta.org/>
- [MMP95] MYLOPOULOS, J. ; MOTSCHNIG-PITRIK, R.: Partitioning Information Bases with Context. In: *Third International Conference on Cooperative Information Systems, Wien, 1995*
- [MPF04] MÜLLER-PROTHMANN, Tobias ; FINKE, Ina: SELaKT - Social Network Analysis as a Method for Expert Localisation and Sustainable Knowledge Transfer. In: *Proceedings of the 4th International Conference on Knowledge Management (I-KNOW '04), Graz, 2004*, S. 115–125
- [MS05] MAICHER, Lutz ; SCHWOTZER, Thomas: Distributed Knowledge Management in the Absence of Shared Vocabularies. In: *Proceedings of*

the 5th International Conference on Knowledge Management (I-KNOW '05). Graz / Austria, July 2005

- [MVNDGL⁺04] MORENO-VERGARA, Nathalie ; NAVAS-DELGADO, Ismael ; GOMEZ-LORA, Antonio C. ; MAR ROLDAN-GARCIA, Maria del ; ALDANA-MONTES, Jose F.: Musa-K: A Practical Step to Integrate Databases and Semantic Web Technologies. In: *Proceedings of the 4th International Conference on Knowledge Management (I-KNOW '04)*, Graz, 2004, S. 388–396
- [MW04] MAICHER, L. ; WITSCHER, H.-F.: Merging of Distributed Topic Maps based on the Subject Identity Measure (SIM). In: *Proceedings of Leipziger Informatiktage, Leipzig (2004)*, 2004
- [N1997] N1920, ISO/IEC J.: Hypermedia/Time-based Structuring Language (HyTime), 2nd Edition / ISO/IEC JTC 1/SC34. 1997. – ISO 10744:1997
- [NCSW04] NOVAK, Jasminko ; CUEL, Roberta ; SARINI, Marcello ; WURST, Michael: A Tool for Supporting Knowledge Creation and Exchange in Knowledge Intensive Organisations. In: *Proceedings of the 4th International Conference on Knowledge Management (I-KNOW '04)*, Graz, 2004, S. 311–319
- [NM01] NOY, Natalya F. ; MCGUINNESS, Deborah L.: Ontology Development 101: A Guide to Creating Your First Ontology / Stanford Knowledge Systems Laboratory. 2001 (#KSL-01-05). – technical report
- [NT95] NONAKA, Ikujiro ; TAKEUCHI, Hirotaka: *The Knowledge-Creating Company*. Oxford University Press, 1995
- [OA] ONTOPIA AS, Oslo: *Ontopia Knowledge Suite*
- [OAS02] OASIS: *Published subjects for countries. PSI-Set basierend auf den ISO 3166-1 Codes fr Länder*. 05 2002. – <http://psi.oasis-open.org/geolang/iso3166/>
- [Old02] OLDENKAMP, Johan H.: Knowledge Cycle Management. In: *Proceedings of the 2nd International Conference on Knowledge Management (I-KNOW '02)*, Graz, 2002, S. 281–293
- [OS04] OLIVEIRA, Jonice ; SOUZA, Jano M.: Improving Knowledge Sharing Objects Representation. In: *Proceedings of the 4th International Conference on Knowledge Management (I-KNOW '04)*, Graz, 2004, S. 261–268
- [Par59] *Kapitel General theory in sociology*. In: PARSON: *Sociology today*. Harper & Row, New York, 1959, S. 3–38

- [PG04] PEPPER, S. ; GARSHOL, L. M.: Seamless Knowledge Spontaneous Knowledge Federation using TMRAP. In: *Presentation at: XML Europe 2004, Amsterdam (2004)*, 2004. – http://www.ontopia.net/topicmaps/learn_more.html
- [PM01] PEPPER, Steve ; MOORE, Graham: XML Topic Maps (XTM) 1.0 / Topicmaps.org. 2001. – Topicmaps.org standard
- [Pol58a] POLANYI, M.: *Personal Knowledge*. University of Chicago Press, 1958
- [Pol58b] POLANYI, M.: *The Tacit Dimension*. Routledge & Kegan Paul, London, 1958
- [Put01] PUTMAN, Janis R.: *Architecturing with RM-ODP*. Prentice Hall, New Jersey, 2001
- [RF04] REINHARDT, Rüdiger ; FLICKER, Anja: Development and Implementation of an Intellectual Capital Management System: Advantages of a Bottom-Up Approach. In: *Proceedings of the 4th International Conference on Knowledge Management (I-KNOW '04)*, Graz, 2004, S. 152–158
- [SC05] SCHWOTZER, Thomas ; CEBULLA, Agnes: Replication of Published Subject Indicator as Thesaurus by means of LDAP. In: *Proceedings of the International Workshop Topic Map Research and Applications (TMRA'05)*, Leipzig, Springer, 2005
- [Sch02] SCHWOTZER, Thomas: Context Driven Spontaneous Knowledge Exchange. In: *Proceedings of the 1st German Workshop on Experience Management 2002 (GWEM02)*, Berlin
- [Sea71] SEARLE, John R.: *Sprechakte*. Suhrkamp Verlag, 1971
- [SEMK04] SWAAK, Janine ; EFIMOVA, Lilia ; MASJA KEMPEN, Mark G.: Finding In-house Knowledge: Patterns and Implications. In: *Proceedings of the 4th International Conference on Knowledge Management (I-KNOW '04)*, Graz, 2004, S. 27–34
- [Sen94] SENGE, Peter M.: *The Fifth Discipline*. Currency Doubleday, New York, 1994
- [SG02] SCHWOTZER, Thomas ; GEIHS, Kurt: Shark – a System for Management, Synchronization and Exchange of Knowledge in Mobile User Groups. In: *J.UCS* 8, Issue 6 (2002)
- [SG03] SCHWOTZER, Thomas ; GEIHS, Kurt: Mobiles verteiltes Wissen. In: *Datenbank Spektrum* 5 (2003)

- [SM94] SIMONE, Rudolph ; MATTHIAS, Hemmje: *Visualisierung von Thesauri zur interaktiven Untersttzung von visuellen Anfragen an Textdatenbanken (GMD-Studie Nr. 247)*. 1994
- [Sow00] SOWA, John F. ; KALLIE SWANSON, Mike S. (Hrsg.): *Knowledge Representation: logical, philosophical and computational foundations*. Brooks and Cole (Thomson Learning), 2000
- [SP02] SCHWOTZER, Thomas ; PREUSS, Thomas: Knowledge Exchange in Spontaneous Networks - Towards Ubiquitous Knowledge. In: *Proceedings of E-World@Syria (ET2EB), Damaskus*, 2002
- [SST04] SCHMITZ, Christoph ; STAAB, Steffen ; TEMPICH, Christoph: Socialisation in Peer-to-Peer Knowledge Managment. In: *Proceedings of the 4th International Conference on Knowledge Management (I-KNOW '04)*, Graz, 2004, S. 35–42
- [SW86] SPERBER, D. ; WILSON, D.: *Relevance: Communication and Cognition*. Basil Blackwell, 1986
- [The99] THEODORAKIS, M.: *Contextualization: An Abstraction Mechanism for Information Modeling*, Department of Computer Science, University of Crete, Greece, Diss., 1999
- [The02] THEODORAKIS, M.: A theory of contexts in information bases. In: *Information Systems* 27 (2002), S. 151–191
- [TPSS04] TEMPICH, Christoph ; PINTO, Sofia ; STAAB, Steffen ; SURE, York: A case study in supporting DIstributed, Loosely-controlled and evolvinG Engineering of oNTologies (DILIGENT). In: *Proceedings of the 4th International Conference on Knowledge Management (I-KNOW '04)*, Graz, 2004, S. 225–232
- [Wah91] WAHRIG, Gerhard: *Deutsches Wörterbuch*. Bertelsmann Lexikon Verlag, 1991
- [Wal05] WALLA, Marcus: *Konzept und Implementierung eines autonomen wissensbasierten Netzwerkes für ein Funktelefon (Diplomarbeit FHTW Berlin)*. 2005
- [Wil00] WILLKE, H.: *Systemtheorie I: Grundlagen*. 6. bearbeitete Auflage. Lucius & Lucius, Stuttgart, 2000
- [Woi04] WOITSCH, Robert: Process Oriented Knowledge Management: A Service Based Approach. In: *Proceedings of the 4th International Conference on Knowledge Management (I-KNOW '04)*, Graz, 2004, S. 408–415

Literaturverzeichnis

- [Yua03] YUAN, Yanping: *A Knowledge Flow System based on Topic Maps and Web Services (Diplomarbeit, TU Berlin)*. 2003