DISKUSSIONEN ALS SZENARIO ZUR KO-KONSTRUKTION VON WISSEN MIT VISUELLEN SPRACHEN

Von der Fakultät für Ingenieurwissenschaften der
Universität Duisburg-Essen
zur Erlangung des akademischen Grades eines

Doktor *Ing.*

genehmigte Dissertation

von

Katrin Simone Gaßner

aus

Berlin

Referent: Prof. Dr. Heinz Ulrich Hoppe
Korreferent: Prof. Dr. Jürgen Ziegler
Tag der mündlichen Prüfung: 27.10.2003

VORWORT

Die technische Unterstützung von Gruppenarbeit hat sich in der Informatik bereits zu einem umfangreichen Forschungsthema entwickelt. Unterstützungsbedarf entsteht beispielsweise aus projektorientierten Arbeitsformen, in denen zur Kosten- und Zeitersparnis auch von verteilten Orten aus kooperiert werden soll. Durch diesen Bedarf, die örtliche Trennung von Arbeitsgruppen zu überbrücken, steht bei der Umsetzung oft die Realisierung der Kommunikationskanäle im Vordergrund. Meistens wird einfach die Möglichkeit geschaffen, Dialoge schriftlich zu führen.

Während meiner Arbeit entwickelte sich die Überzeugung, dass die eigentliche konstruktive Arbeit aber nach wie vor hauptsächlich in der „face-to-face"-Situation geleistet wird, einer Situation, der oft gar kein Unterstützungspotential durch Software, Tools und Devices zuerkannt wird. Trotzdem scheint gerade dort der flexible Einsatz von Medien einen Erfolgsfaktor zu bilden. Die Passung der Medien auf die Kommunikationssituation und die Arbeitsprozesse wird dort unmittelbar gefordert und ein Mehrwert jenseits des Kommunikationskanals erwartet.

Im Verlauf dieser Arbeit wird deshalb diskutiert, wie Medien gestaltet werden können, um eine Kommunikation, speziell Diskussionen, zu bereichern. Als zentrale Aspekte stellen sich der Bedarf zur flexiblen Wissenskommunikation heraus, die Unterstützung eines Dokumentationsprozesses einschließlich der Informationsvernetzung, die Vermeidung von Medienbrüchen sowie die Einbettung der Tools in größere Arbeitszusammenhänge, was auch auf den nicht-technischen Kontext hinweist. Daraus ergibt sich ein weitgehend interdisziplinäres Thema mit vielen Überschneidungen zur Lernunterstützung einschließlich der Wissenskonstruktion, Wissenselizitation und Reflexion.

Im Zentrum der Arbeit stehen visuelle Sprachen als Hilfsmittel der Wissenselaboration. Diese werden in ihrem Gebrauch untersucht und ihr Nutzen für Implementierungen hervorgehoben. Es werden zwei Systeme realisiert und vorgestellt, eines behandelt das Problem der nachhaltigen Dokumentation, das andere realisiert eine an Darstellungsformen (Perspektiven) orientierte Diskussionsunterstützung.

Danksagungen

Bedanken möchte ich mich vor allen Anderen bei Prof. Hoppe, der diese Arbeit betreut hat und der durch den inspirierenden Forschungskontext in der Arbeitsgruppe Collide die Rahmenbedingungen für diese Arbeit geschaffen hat. Die durch Prof. Hoppe eingeflossenen Anregungen und die Freiräume bei der Gestaltung haben sich zu einer hervorragenden, kreativen Arbeitsgrundlage verbunden.

Besonders danke ich außerdem Frank Tewissen und Martin Mühlenbrock, die mit der Entwicklung der Basissysteme Dalis, MatchMaker (C++ und TNG) sowie dem CardBoard einen Ausgangspunkt sowohl für das DiscBoard als auch für den FreeStyler zur Verfügung gestellt haben. Frank Tewissen realisierte, fortgeführt von Marc Jansen, mit den MatchMaker-Entwicklungen maßgeblich die Grundlagen für die replizierten Architekturen, die in den Systemen aufgegriffen worden sind.

Die Entwicklungsarbeiten im vorliegenden Umfang konnten nur unter Mitwirkung einiger ambitionierter Studenten durchgeführt werden: Stephan Buschmann implementierte im Rahmen eines Projektes zusammen mit Martin Fendrich große Teile des FreeStylers. Andreas Lösch realisierte einen wesentlichen Anteil des Basissystems CardBoard.

Wichtig und konstruktiv war die enge Zusammenarbeit mit meinen Arbeitskollegen, bei denen ich mich ganz herzlich für die Unterstützung bedanke. Für eine unschätzbare Hilfe im Hintergrund sorgten Andreas Lingnau und Marc Jansen mit der Wartung der technischen Infrastruktur sowie Niels Pinkwart durch die kooperative Arbeitsweise bei der Systementwicklung.

In besonderem Maße haben mich Martin Fischer und Frank Tewissen durch ihre ausführliche Korrektur meiner Arbeit unterstützt, sowie Gernot Grube, der den „Blick von außen“ wagte.

Mein ganz persönlicher und großer Dank gilt Martin Fischer, der zu jedem erdenklichen Zeitpunkt bereit war, mit mir inhaltliche und konzeptuelle Fragen zu diskutieren. Letztendlich hat er während meiner Promotionszeit die Weltkugel etwa 3 Mal umrundet, um die Distanz unterschiedlicher Wohnorte zu überwinden.

INHALTSVERZEICHNIS

MOTIVATION 1

EINLEITUNG UND ÜBERBLICK 5

TEIL I VISUALISIERUNGEN UND REPRÄSENTATIONEN ZUR DISKUSSIONSUNTERSTÜTZUNG: PROZESSE DER WISSENSELABORATION 23

1 DISKUSSIONSUNTERSTÜTZUNG 25

1.1 Diskussionen 25
1.2 Diskussionen als Mittel der Wissenskonstruktion 27
1.3 Diskussionen als Bestandteil betrieblichen Wissensmanagements 29
1.4 Sind Diskussionen effektiv? 33
1.5 Technische und Softwarebasierte Diskussionsunterstützung 36
1.5.1 Ansätze und Zielstellungen 37
1.5.2 Technische Unterstützung der Kooperation 44

2 VERWENDUNG VISUELLER SPRACHEN 51

2.1 Visuelles Programmieren und Visualisierung von Programmen 52
2.2 Formalisierung visueller Sprachen und Schließen in visuellen Repräsentationen 53
2.3 Lernunterstützung 59
2.4 Wissenselizitation 61
2.5 Kommunikationsmedien und Wissensprodukte 62
2.5.1 Kommunikationsmedien 63
2.5.2 Wissensprodukte 67

3 SYSTEME AUF BASIS VISUELLER SPRACHEN 73

3.1 Architektur der Basis-Systeme für die Diskussionsunterstützung 73
3.1.1 MatchMaker 74
3.1.2 Dalis 75
3.1.3 CardBoard 77
3.1.4 Die visuellen Sprachen im CardBoard 80
3.2 Systeme im Vergleich 83
3.2.1 Belvedere 84
3.2.2 Chips 86
3.2.3 CoLab: Cognoter/Argnoter 88
3.2.4 Csile 90
3.2.5 Dolphin 92
3.2.6 gIBIS 93

3.2.7 MindManager 95
3.2.8 OneNote 96
3.2.9 Sepia 98
3.3 Erfahrungen bei der Verwendung visueller Sprachen 100
3.4 Synopse 102
3.4.1 Software/Tool 104
3.4.2 Realisierung des Szenarios 105
3.4.3 Organisatorische Einbettung 105

4 DISKUSSIONSUNTERSTÜTZUNG: EINE VORSTUDIE ZUR HYPOTHESENBILDUNG 111

4.1 Zielsetzung 111
4.2 Setting 113
4.3 Verlauf 116
4.4 Auswertung 118
4.4.1 Kooperatives Editieren in den Workspaces 118
4.4.2 Verständlichkeit der Darstellung 119
4.4.3 Einarbeitung 120
4.4.4 Moderationsstrategie 121

5 ANSÄTZE ZUR UNTERSTÜTZUNG KOOPERATIVER WISSENSELABORATION 125

5.1 Konzepte der Systeme DiscBoard und FreeStyler 129
5.1.1 DiscBoard 129
5.1.2 FreeStyler 131
5.2 Systemarchitektur 132

TEIL II IMPLEMENTIERUNG ZWEIER SYSTEMBEISPIELE 135

6 FALLSTUDIE I: WISSENSELABORATION DURCH PERSPEKTIVEN: DAS SYSTEM DISCBOARD 137

6.1 Perspektiven 137
6.1.1 „Epistemic forms“ als Perspektiven 138
6.1.2 Modellierung mit „epistemic games“ 141
6.1.3 Perspektivenwechsel 143
6.2 Erweiterungen des CardBoards 147
6.2.1 „Joint“ Workspaces 147
6.2.2 Paletten 150
6.3 Die visuellen Sprachen für die Diskussionsunterstützung: Die Perspektiven 151
6.3.1 Allgemein zur Gestaltung der visuellen Sprachen 152
6.3.2 Die visuelle Sprache zur Diskussionsstrukturierung (Protokoll) 154
6.3.3 Die visuelle Sprache zum Explorieren 155
6.3.4 Die visuelle Sprache für das Brainstorming 158
6.3.5 Die „Topic Map“ 161
6.3.6 Die visuelle Sprache zum Argumentieren 163
6.3.7 Die visuelle Sprache für die Pro-/Kontratabellen 167
6.4 Die Funktionspalette 168
6.5 Implementierungskonzept der multi-Perspektivik 171

6.5.1 Anforderungen und Ziele 171
6.5.2 Konzept der Implementierung 173
6.6 Abbildung von Karten: Knotencluster 187
6.6.1 Das Generieren der Knotencluster 190
6.6.2 Zustandsmarkierungen 193
6.7 Abbildung von Strukturen 204
6.7.1 Strukturen 205
6.7.2 Das Strukturmodell 210
6.7.3 Strukturüberprüfung 211
6.7.4 Strukturabbildungen 215
6.7.5 Löschen in Strukturen 218
6.8 Das Diskussionsmodell 219
6.8.1 Abbildungsdefinitionen für Inhaltskarten 219
6.8.2 Abbildungsdefinitionen für Strukturen 223
6.9 Sortierungen 237
6.10 Zusammenfassung 239
6.10.1 Die Perspektiven 240
6.10.2 Unterstützung eines Diskussionsprozesses 241
6.11 Diskussion 244

7 FALLSTUDIE II: WISSENSELABORATION DURCH DOKUMENTATIONSPROZESSE: DAS SYSTEM FREESTYLER 247

7.1 Wissensmanagement: Das Projekt Trendmonitoring 249
7.1.1 Ansatz 250
7.1.2 Trendkontexte 251
7.1.3 Analyse der Arbeitsorganisation 251
7.1.4 Systemanforderungen 257
7.1.5 Der FreeStyler Version 1 259
7.1.6 Beispiel für die Entwicklung eines Trendkontextes 267
7.2 FreeStyler Version 2 272
7.2.1 Einsatz des FreeStylers im Deutschunterricht 274
7.2.2 Einordnung und Verwendung 278
7.3 Zusammenfassung 279
7.4 Diskussion 281

8 ZUSAMMENFASSUNG UND AUSBLICK 285

8.1 Zentrale Aspekte 285
8.2 Diskussion 288
8.3 Ausblick 291

9 LITERATURVERZEICHNIS 295

10 LINK-VERZEICHNIS 311

Seine Meinung ist die rechte, wenn er spricht,
müsst Ihr verstummen. Sonst erklärt er euch für
Schlechte oder nennt euch gar die Dummen.

Wilhelm Busch

Idee aus Lehmann (1998)

MOTIVATION

Seit einigen Jahren wird in Betrieben zunehmend versucht, explizit Knowledge Management zu betreiben. Im Hintergrund steht das Problem, dass das Wissen der Mitarbeiter und Mitarbeiterinnen eine immer wichtigere Ressource im Wettbewerb darstellt, aber in betrieblichen Arbeitsabläufen nicht ausreichend genutzt wird. Knowledge Management soll diese Ressource besser erschließen und auch dazu beitragen, von individuellem Wissen unabhängiger zu werden. Allerdings sind eine Vielzahl von Erwartungen und Ansprüchen, die an das Knowledge Management gerichtet werden, häufig widersprüchlich oder unreflektiert und bewirken bei der Einführung der Maßnahmen unüberwindbare Probleme.

Zum einen soll Wissen besser zugreifbar werden. Das bedeutet, dass Mitarbeiter und Mitarbeiterinnen ihre Kompetenzen vermehrt vermitteln oder anbieten sollen. Dafür muss transparent sein, wer welche Kompetenzen hat und wie Zuständigkeiten verteilt sind. Darüber hinausgehend soll Wissen sogar unabhängig von den Kompetenzträgern verfügbar sein. Daraus folgt dessen Abspaltung von den Personen, also die Externalisierung von komplexen Wissensstrukturen. Über externe Repräsentationen soll es so transportiert werden, dass sich Mitarbeiter und Mitarbeiterinnen auf dessen Grundlage einfach informieren oder sogar vertieft in Themen einarbeiten können.

Neben dem Ziel der allgemeinen Verfügbarkeit soll Wissen außerdem aktiv verteilt werden können. Gerade in großen Betrieben soll das auch automatisiert möglich sein. Um das zu erreichen, müssen die anfallenden Informationen den Empfängern zugeordnet werden, die dafür häufig über Kompetenzprofile charakterisiert werden sollen.

Dem idealen Ziel, Wissen öffentlich zu teilen, steht jedoch beispielsweise oft der individuelle Widerstand der Mitarbeiter und Mitarbeiterinnen entgegen, ihr in eigener Kraft angeeignetes Wissen allgemein zur Verfügung zu stellen. Kompetenzprofile können erschwerend sehr persönlich und individuell sein. Deren transparente Darstellung stößt deshalb auf besonderen Widerstand vieler Mitarbeiter und Mitarbeiterinnen, der in diesem Fall auch kaum durch organisatorische Maßnahmen aufgelöst werden kann.

Lösungen zur Externalisierung und Repräsentation von Wissen werden leichter in kleineren Gruppen gefunden und sinnvoll durch nicht-technische Maßnahmen, z.B. „vertrauensbildende“ und kommunikationsfördernde Managementmethoden, begleitet.

Die Einschränkung auf kleinere Gruppen beinhaltet jedoch einen Widerspruch und zwar in Bezug auf den Bedarf an Wissensmanagement. Denn mit Wissensmanagement ist meistens die Hoffnung verbunden, gerade in großen, zumeist internationalen Betrieben den Transfer von Wissen zu verbessern.
Vielleicht allerdings das schwerwiegendste Problem beim „knowledge sharing“ auf Basis externer Repräsentationen liegt im Aufwand, diese zu erstellen und zu verwalten.
Es kann versucht werden, durch arbeitsorganisatorische Maßnahmen, Personen für diese Aufgaben freizustellen. Aber das scheint besonders im mittleren Management nur schwer durchsetzbar zu sein, denn hier lauert ein Kostenproblem, das in der Konkurrenz von Projekten oder generell Organisationseinheiten liegt. Handelt es sich nämlich um Wirtschaftseinheiten, so sind Arbeitspakete, die zwar dem Unternehmen insgesamt, nicht aber direkt der Gruppe selbst zugute kommen, oft schlichtweg nicht abzurechnen und damit nicht erwünscht:
Während diese Arbeit entstand, war ich an einigen Diskussionen mit Vertretern und Vertreterinnen aus Betrieben beteiligt, die sich über Möglichkeiten des Wissens-managements informierten. Es zeigte sich neben den einzelnen Problemen mit der Bilanzierung, der Externalisierung von Daten an sich und den sensibleren persönlichen Daten der Kompetenzprofile, dass sich diese Probleme auch noch gegenseitig beeinflussten. In einem Fall stellte sich heraus, dass die Hilfestellungen, die von anderen Abteilungen nachgefragt wurden, im vorliegenden Abrechnungssystem nicht abgerechnet werden konnten. Die Einführung der Kompetenzprofile, also ein System, firmenweit Services anzubieten, hätte dieses Problem potenziert. Das abteilungsübergreifende Angebot von Hilfestellungen war deshalb einerseits Ziel der Wissensmanagement-Initiative und gleichzeitig in der bestehenden Organisation nicht durchführbar.
Generell scheint das moderne Modell der Unterteilung eines Betriebes in „profit center“, schnell mit einem übergeordneten Ansatz des Knowledge Management zu kollidieren. Verstärkt wird dieser Konflikt durch die Befürchtung, dass besonders kompetente und entsprechend „nachgefragte“ Mitarbeiterinnen und Mitarbeiter nicht mehr genügend Zeit mit ihren Kernaufgaben verbringen.
Im Zusammenhang mit der externen Repräsentation von Inhalten stellt sich immer auch die Frage nach deren Lebensdauer und damit auch dem Aufwand und den Kosten für deren Darstellung. Bei langlebigen, statischen Informationen lässt sich der Aufwand für die Repräsentation für die meisten Betriebe eher rechtfertigen und abschätzen. Handelt es sich um kurzlebige, dynamische Informationen, erhöht sich der Aufwand, da eine ständige Aktualisierung der repräsentierten Inhalte hinzukommt. Wegen der Unwäg-

barkeiten beim Zeitaufwand, bei den Kosten und dem davon ausgehenden Nutzen werden deshalb die dynamischen Informationen oft vernachlässigt. Dynamische Informationen können aus den täglichen Erfahrungen entstehen, sich schnell ändern und hochgradig unvollständig oder gar zusammenhanglos sein. Sie sind meistens in einem wenig elaboriertem Stadium und können ohne erheblichen Aufwand gar nicht selbsterklärend dargestellt werden. Werden sie allerdings generell ausgeschlossen, so ist zu kritisieren, dass einem Wissensmanagement als Verwaltung und Gewinnung von Erfahrungen und Wissen nicht genüge getan wird, denn dieses bezieht sich ja häufig gerade auf dynamisches Wissen.

Natürlich ist die Darstellung solches dynamischen Wissens nur ein Ausschnitt aus dem Bereich des Wissensmanagements und zwar einer mit einem sehr weit reichenden Ziel. Vielen ist schon ausreichend geholfen, wenn die statischen, langlebigen Daten geeignet verwaltet werden. Allerdings handelt es sich dabei eher um ein Datenmanagement als um ein Wissensmanagement. Darin besteht übrigens schon eine relevante Unterscheidung, die auch unterschiedliche technische Unterstützungen nach sich zieht. Während für ein Datenmanagement hauptsächlich die technische Realisierung gelöst werden muss, die Daten aber weitgehend feststehen, muss beim Wissensmanagement zusätzlich geklärt werden, welche Inhalte verwaltet und wie diese erhoben und repräsentiert werden.

Dazu ein Beispiel: Könnten Sie formulieren, worin Ihre Erfahrung z.B. bei der Auswahl eines Restaurants besteht? Vermutlich fallen Ihnen viele Aussagen der Form „das hängt von der Gruppe ab,..“ oder „unter der Voraussetzung, dass ...“ ein. Um Erfahrungen notieren zu können, muss bekannt sein für welchen Zweck sie verwendet werden sollen, ansonsten explodiert die Menge der Informationen, die repräsentiert werden müsste. Nicht jedes Detail kann explizit dargestellt werden. Auch der Detaillierungs- und Formalisierungsgrad muss klar sein. Neben dem Aufwand, die Erfahrungen niederzuschreiben, beinhaltet schon die Reflexion über Inhalte einen enormen Aufwand. Was sind überhaupt Erfahrungen und worin besteht das Wissen, das man sich an einem Tag erworben hat. Conklin und Burgess-Yakemovic (1996) weisen dazu auf die Anti-Proportionalität hin, dass je geringer die Kosten für die Dokumentation bzw. Repräsentation sind, desto größer die Kosten beim Zugriff auf die Information werden. Ein Ansatz, dynamisches Wissen weitgehend formal zu repräsentieren, greift also voraussichtlich zu weit, um eine befriedigende Lösung für den alltäglichen Gebrauch zu finden.

Aber nicht nur das „was“ sondern auch das „wer“ muss bei der Externalisierung und Repräsentation geklärt werden, wenn dies fester Bestandteil der Arbeitsabläufe werden

soll. Würden Informationen zentral durch dafür zuständige Mitarbeiter oder Mitarbeiterinnen dokumentiert, müssten die aktuellen Informationen erst den zuständigen Personen zugetragen werden, was beinahe unmöglich scheint und vermutlich auch nicht sinnvoll wäre. Es bleibt also fast zwingend nur eine dezentrale Wartung übrig. Aber selbst dann ist von der Notwendigkeit einer Nachbearbeitung von Gesprächen, Darstellung von Ideen, etc. auszugehen, die höchst mühsam wäre. Sie bedeutet meistens einen deutlichen zusätzlichen Aufwand, der im Allgemeinen einer geringen Priorität unterliegt.

Das dynamische Erfahrungswissen entsteht dezentral und kann auch nur so festgehalten werden. Ob den Mitarbeitern und Mitarbeiterinnen hierfür Zeit bleibt, ist in erster Linie ein organisatorisches Problem, das unter Umständen durch technische Unterstützung reduziert werden kann. Die Probleme der Bilanzierung sind in die organisatorischen Probleme einzuordnen genau wie die Schaffung der „psychologischen“ und „sozialen“ Voraussetzungen, damit die Mitarbeiter und Mitarbeiterinnen dazu bereit sind.

Bedeutet das nun, dass es nicht sinnvoll ist, das täglich entstehende Wissen zu dokumentieren und zugreifbar zu machen? Oder was kann trotzdem aus dem Alltagsarbeitsleben an Wissen gewonnen werden und wie?

EINLEITUNG UND ÜBERBLICK

Arbeitshypothese. Diese Arbeit verfolgt den Ansatz, Teamsitzungen durch Kommunikationsmedien anzureichern, um einerseits die Diskussionen selbst zu unterstützen und andererseits daraus unmittelbar Dokumentationen zu erhalten. Die Diskussion wird als „natürliche Notizsituation" ausgenutzt, „natürlich" insofern, als dass auch in herkömmlichen Diskussionen oft Skizzen und Notizen als Hilfsmittel verwendet werden, um den inhaltlichen Austausch zu unterstützen (Lehmann, 1998). Diese Darstellungen werden somit als Medien während der Kommunikation eingesetzt und dienen gleichermaßen der Externalisierung besprochener Inhalte.

Der Ansatz baut auf der These auf, dass Diskussionen ein ganz wichtiger Zeitpunkt sind, um dynamisches Wissen zu erfassen. Ihre Bedeutung für alltägliche Arbeitsabläufe wird häufig unterschätzt. Beispielsweise nehmen Meetings – zentrales Forum für Diskussionen – nach Bargiela-Chiappini und Harris (1997) rund 70 % der gesamten Arbeitszeit eines Managers ein. Auch in Slavin et al. (2001) werden „face-to-face"-Meetings potentiell als am weitesten verbreitete Form von wissensbezogenen Aktivitäten hervorgehoben.

Wissenselaboration. Nach Nonaka (1994) wird Wissen dadurch gewonnen, dass implizite Informationen immer wieder externalisiert, elaboriert und internalisiert werden. In einem asynchronen Kommunikationsszenario werden dazu in vielen Fällen externe Repräsentationen integriert, die hier als Medien verstanden werden. Diskussionen bieten einen Rahmen, in dem Zyklen der Externalisierung des Wissens und dessen Internalisierung unmittelbar ineinander greifen. Generell eignen sich Diskussionen im besonderen Maße, um implizites Wissen zu äußern (van Joolingen, 2000; [1]).

Manche Vorgehensweisen, die während einer Diskussion unmittelbar durchgeführt werden, können mit Lernmethoden verglichen werden, die dazu entwickelt wurden, den Reflexionsprozess anzuregen. Die Besonderheit besteht darin, dass sie während der Kommunikation ohne besonderen Aufwand eingesetzt werden. Beispielsweise werden Aussagen oft direkt hinterfragt, was zu weitergehenden Erklärungen führt. Dieser Prozess kann mit dem Elizitieren von „Selbst-Erklärungen" („self-explanation": VanLehn, Jones & Chi 1992; Chi, De Leeuw, Chiu & Lavancher, 1994) beim individuellen Lernen

[1] Stellung wird dazu auch im Magizin *„iX Magazin für professionelle Informationstechnik"* genommen (iX: 7/2000). Siehe dazu Referenz Versteegen (2000).

verglichen werden. Bei „Selbst-Erklärungen“ muss bestehendes Wissen mit neuen Wissenselementen integriert werden, wobei es zu Konflikten kommen kann, wenn ein fehlerhaftes oder unvollständiges Wissensmodell mit neuen Informationen nicht mehr kompatibel ist. Durch das konstruktive Lösen solcher und anderer Konflikte in Gruppen bzw. Lerngruppen kann nicht nur Neugier geweckt, sondern auch die Suche nach zusätzlicher Information angestoßen werden (Johnson & Johnson, 1985). So wird mehr Zeit mit der inhaltlichen Fragestellung verbracht und tendenziell eine positive Haltung zu den besprochenen Inhalten entwickelt.

Andere Methoden, die vergleichbar in Diskussionen umgesetzt werden, sind das kritische Hinterfragen bzw. die Nachforschung („critical inquiry“). Darin sollen (wissenschaftliche) Theorien oder auch allgemein Ansichten und Aussagen frageorientiert untersucht werden. Neben der Aufarbeitung des Gegenstandsbereiches muss dafür eine eigene Perspektive entwickelt werden, eine Fragestrategie aufgebaut werden und eine Fokussierung auf relevante Aspekte stattfinden.

Das Argumentieren, speziell das wissenschaftliche Argumentieren (Suthers, Toth & Weiner, 1997), fordert ebenfalls die Reflexion von Inhaltsbereichen, insbesondere die Verknüpfung von Daten und Fakten mit Schlussfolgerungen.

Diese Parallelen zwischen Prozessen der Diskussion und Lernmethoden stützen die These, dass Diskussionen geeignet zur Wissenselizitation und -konstruktion sind. Diskussionen dienen der Exploration von Themen, dem Zusammentragen von Informationen, die über Personen verteilt vorliegen, der Konstruktion eines gemeinsamen Verständnisses oder auch dem Klären von Zusammenhängen. Alle diese Handlungen bewirken normalerweise Lerneffekte bei den Beteiligten.

Die Qualität der erarbeiteten Diskussionsergebnisse und damit deren Akzeptanz und Einfluss auf weitere Arbeiten hängen stark vom Verlauf der Diskussion ab. Dies wird aber weitgehend subjektiv beurteilt, denn die besprochenen Inhalte können selten objektiv und abschließend beurteilt werden, da viel zu viele vage Informationen mit einfließen. Z.B. sind Designs, Planungen und Entscheidungen erst an ihrem Erfolg zu messen. Akzeptanz und Auswirkung hängt deshalb davon ab wie tiefgehend diskutiert wurde oder auch davon, welche Sachkenntnis den jeweils Beteiligten zugeschrieben wurde. Ebenso wichtig ist das soziale Verhalten. Werden Entscheidungen über Machtpositionen durchgesetzt, so wirkt sich das genauso negativ aus, als würden sich die Anwesenden aus taktischen Gründen nicht an der Diskussion beteiligen.

Die Art des Diskussionsverlaufes und des nicht-fachlichen, rollenbezogenen Einflusses sowie die Überzeugungskraft der vorgebrachten Fakten und Argumente und die geeignete

Zusammensetzung einer Diskussionsgruppe bewirken entscheidend, ob und wie die Ergebnisse akzeptiert und umgesetzt werden. Diese Gruppe von Aspekten möchte ich mit dem Begriff der *Plausibilität* der Diskussionsergebnisse zusammenfassen.

Nachhaltigkeit. Betrachtet man Diskussionen als Mittel der Wissenskonstruktion, führt das direkt zum Bedarf an *Nachhaltigkeit* der Diskussionsergebnisse. Werden die Gespräche nur mündlich geführt, so sind die Inhalte flüchtig. Schafft man es, die Visualisierung vielleicht sogar mit methodischen Ansätzen zu kombinieren, so erreicht man über deren medialen Charakter hinaus einerseits die Unterstützung der Externalisierung und der Konstruktion des Wissens und erhält andererseits direkt eine Dokumentation. Während die Diskussion selbst also zum Externalisieren von Inhalten führt, dient die Vorgabe der Repräsentationsmittel einer „weichen" Formalisierung.

Schafft man es, die Diskussion mit Visualisierungen so zu unterstützen, dass diese gleichzeitig der Dokumentation dienen, so schafft man einen Mehrwert auf zwei Ebenen: der Unterstützung der Externalisierung und der Wiederverwendung z.B. für weiterführende Dokumentationen. Ich sehe die Dokumentation als zweistufigen Prozess an, in dem als erstes dynamisches Wissen festgehalten wird, das aber trotzdem retrievalfähig sein muss. Diese Daten können in zusätzlichen Arbeitsschritten für die langfristig ausgelegte Dokumentation herangezogen werden.

Das Problem der Wiederverwendbarkeit entsteht natürlich nicht durch den Einsatz eines Computer-Tools, sondern existiert auch allgemein bei der Kommunikation. Lehmann (1998) führt zwar an, dass Visualisierungen generell zur Dokumentation geeignet sind, aber dies ist nach meiner Ansicht stark von den Mitteln abhängig, die verwendet werden und auch von der Disziplin bei der Nachbereitung. Mitschriften werden meistens nicht sinnvoll verwahrt, so dass später kein Zugriff mehr möglich ist. Tafelmitschriften werden weggewischt und Skizzen in den Papierkorb geworfen. Ironisch betrachtet tritt so das Problem, die Mitschriften in der Gruppe produktiv neu zu verwenden, gar nicht erst auf.

Die Verwendung von Computer-Tools eröffnet gerade im Bereich der Wiederverwendung viele Optionen. Nur bei der kreativen Gestaltung sind noch deutliche Lücken zu erkennen. Dies ist ein Grund, weshalb sich diese Arbeit wesentlich mit der technisch gestützten kreativen Gestaltung von Visualisierungen auseinander setzt. Bei der Unterstützung von Kreativsitzungen steht in dieser Arbeit nicht die Abkopplung von Wissen von Personen, sondern die Externalisierung von Gesprächsinhalten im Vordergrund. Auf diese Weise sollen implizite Wissensressourcen erschlossen und in die Arbeitsabläufe integriert werden.

In der Arbeit wird die Frage, wann es sich bei den repräsentierten Inhalten um Information und wann um Wissen handelt, nicht weiter thematisiert. Im Allgemeinen werde ich davon ausgehen, dass Informationen dadurch, dass sie aufgenommen und verstanden oder einfach interpretiert werden, in Wissen übergehen. Das deckt sich weitgehend mit der Sicht von Borghoff und Pareschi (1998). Ich werde im weiteren aber keine begriffliche Trennung zwischen dem extern dargestellten Wissen und dem internalisierten Wissen vornehmen, auch wenn das extern dargestellte streng genommen meistens als Information bezeichnet werden müsste.

Visuelle Sprachen. Visuelle Sprachen werden hier als Hilfsmittel verwendet, um externe Repräsentationen zu erstellen. Im Sinne von Lakin (1990) werden sie als Kommunikationsmedien eingesetzt. Der Sprachaspekt beinhaltet bereits eine Annahme über syntaktische Konventionen bei der Formulierung von Repräsentationen. Außerdem stellt der Begriff der visuellen Sprache die Visualisierung als maßgebliche Eigenschaft dieser Sprachen dar.

Im vorliegenden Ansatz werden mit visuellen Sprachen immer semi-formale Repräsentationen erstellt, die den Vorteil haben, individuelle nicht formalisierte Inhaltseingaben mit formalen interpretierbaren Elementen zu verbinden ([dynagloss1];[dynagloss2]). Kremer (1998) unterscheidet zusätzlich auch informelle Sprachen, die keinen Beschränkungen unterliegen: „A visual language, ..., is any form of communication that relies on two- or three-dimensional graphics ...“. Bei der vollkommen freien Gestaltung von Inhalten, können Objekte jedoch schwieriger mit einem interaktiven Verhalten ausgestattet werden, es sei denn eine Programmierung durch Nutzerinnen und Nutzer wird unterstützt.

Für die Verwendung visueller Sprachen werden in dieser Arbeit fünf Zielsetzungen herausgearbeitet, die sich allerdings nicht gegenseitig ausschließen:

- Visuelles Programmieren und Visualisierung von Programmen
- Formalisierung visueller Sprachen und Schließen in visuellen Repräsentationen
- Lernunterstützung
- Wissenselizitation
- Kommunikationsmedien und Wissensprodukte

Nach dieser Kategorisierung kann der Ansatz der Diskussionsunterstützung den drei letzten Zielen zugeordnet werden: Im Sinne von Lakin (1990) sind die externen Repräsentationen in diesem Ansatz der Diskussionsunterstützung als Kommunikationsmedien zu verstehen. Das Lernen wird indirekt unterstützt, indem Reflexionen durch die externen Darstellungen erleichtert und angestoßen werden. Durch bestimmte Dar-

stellungsarten wird die Wissenselizitation unterstützt; es werden vor allem Wissenslücken „sichtbar“ gemacht. Die Repräsentationen sind Arbeitsprodukte, die auf die Erstellung von Wissensmodellen abzielen. Dabei ist die Modellierung oder auch die Wissenskonstruktion der Prozess, in dem neue Überzeugungen gewonnen werden. Werden die jeweiligen Ergebnisse dokumentiert, so kann das als Wissensprodukt bezeichnet werden.
Als Mittel der Externalisierung werden in diesem Ansatz visuelle Sprachen verwendet, um Diskussionsinhalte diagrammatisch aufzuarbeiten. Mit ihnen wird der „Diskussionsraum“ aufgebaut und strukturiert. Die Diagramme werden formal als Graphen verwaltet, sie sind jedoch für Nutzerinnen und Nutzer nicht immer als Graphen erkennbar.
Für die strukturierte Darstellung von Inhalten werden häufig Graph- oder Netzstrukturen verwendet, die im Wesentlichen durch drei Ziele motiviert werden (Jüngst & Strittmatter, 1995; Fischer, Gräsel, Kittel & Mandl, 1996):

- Darstellung und Diagnose von Wissensstrukturen
- Lernen komplexer Sachverhalte
- Hilfsmittel zum selbstgesteuerten Lernen

Maßgeblich durch psychologische Theorien begründet, wurden deshalb diverse „Mapping“-Verfahren entwickelt, wie „Concept Mapping“, „Mind Mapping“ oder Strukturlegeverfahren, die entsprechend methodisch eingebettet aber genauso zur Wissenskonstruktion, zur Wissenselizitation oder sogar als Grundlage für Kreativtechniken herangezogen werden können.
Unabhängig von psychologischen Begründungen sind Graphen aus Informatik-Sicht mächtige Instrumente. Werden mit visuellen Sprachen Graphen erst einmal grafisch erstellt, kann diese Repräsentation intern in einer entsprechenden Datenstruktur verwaltet werden. So stehen die Darstellungen weiterführenden Berechnungen wie der Feststellung von Zyklen, Wegen, etc. offen. Je nach Art der Sprache, können beispielsweise auch semantische Netze aufgebaut werden, die intern z.B. in eine prädikatenlogische oder terminologische Form übertragen werden. Mit ihnen können auch gut ER-Modelle dargestellt werden. Graphdarstellungen sind außerdem Grundlage für verschiedene dynamische Modellierungen wie im „Systems Dynamics“-Ansatz (Forrester, 1968; Bollen, 2001). Werden keine allgemeinen Aspekte von Graphen implementiert, sondern spezielle Repräsentationen verwendet, so spielen typischerweise die Knoten- und Kantenannotationen eine Rolle, die dann Objekte (Instanzen), Konzepte, Variablennamen, Rollen, Prädikate, etc. spezifizieren. Das heißt, dass diese Annotationen in einen Formalismus einfließen können.

Der Ansatz kooperativer visueller Sprachen (Hoppe, Gaßner, Mühlenbrock & Tewissen, 2000) ermöglicht fließende Übergänge in dem Spektrum zwischen syntaktischer, grafisch-interaktiver Informationsstrukturierung und semantischer Verarbeitung. Die visuellen Sprachen stellen die Mittel zur Protokollierung, Strukturierung und Überarbeitung zur Verfügung. Sie bieten Inhaltscontainer und Relationen an, um die Inhalte diagrammatisch aufzubereiten. Eine Menge von Knotenarten und Relationen und die Möglichkeit daraus Graphen zu entwickeln, stellt die Syntax einer visuellen Sprache dar. Die Syntax der visuellen Sprachen beschreibt also tendenziell eine Symbol-Sprache mit der allein keine Inhalte formuliert werden sollen, die jedoch Aussagen über Bezüge zwischen und Klassifikationen von Beiträgen zulässt. Unterschiedliche visuelle Sprachen entstehen, indem Mengen von Symbolen zusammengefasst werden, die sich durch Farbe, Form oder Benennungen unterscheiden. Für die Sprachobjekte ist eine operationale Semantik definierbar.

Diese Beschreibung von visuellen Sprachen folgt der Sicht von Suthers (1999A), der dort repräsentationale Notationen als Menge primitiver Elemente einführt, aus denen mit einem repräsentationalen Tool repräsentationale Artefakte konstruiert werden können. Ausdrücke einer visuellen Sprache - gemäß Suthers (1999A) repräsentationale Artefakte - sind externe, visualisierte Repräsentationen.

Externalisierungen. Es ist davon auszugehen, dass spezialisierte sowie informelle Darstellungen nicht nur das Gedächtnis entlasten, sondern maßgeblich das Denken selbst beeinflussen (Jonassen & Carr, 2000; Perkins, 1993; Pea, 1985). Die meisten modernen Technologien wären ohne spezifische Darstellungsformen und Interaktionen nicht denkbar („assistance tool“: Resnick, 1991). Das gilt für Bauzeichnungen für Berechnungen, CAD, Software-Engineering mit Datenflussdiagrammen, UML, mathematische Berechnungen mit ihren Ausdrucksmitteln, Diagramme wie Statistiken, etc. Die Konkreten Artefakte dienen dabei vor allem als Bezugmedium in einer Kommunikation („artifact-centered discourse“: Suthers & Xu, 2002).

Weiterhin haben Externalisierungen eine wichtige Funktion bei der Entlastung des Gedächtnisses. Das gilt nicht nur für persönliche Notizen, sondern auch für Gruppensituationen und zwar sowohl für die synchrone und die asynchrone als auch für die örtlich verteilte oder nicht verteilte Situation. Bei Notizen und Visualisierungen, die aus der synchronen Gruppenarbeit entstehen, ist diese Entlastung unmittelbar ersichtlich. Externalisierungen im Sinne eines „corporate memory“ bilden aus einer übergeordneten Perspektive, aus Sicht des Unternehmens, eine Entlastung dadurch, dass Informationen geteilt werden können.

Durch den Einsatz bestimmter visueller Sprachen oder Repräsentationsmittel bei der Externalisierung und Darstellung von Inhalten können bereits Deutungen und Schlussfolgerungen erleichtert oder Schwerpunkte bei der Interpretation gesetzt werden (Suthers, 2003; Zhang, 1997).

Anhand zweidimensionaler oder räumlicher Darstellungen können Skizzen, Bilder oder räumliche Anordnungen auch im Detail diskutiert werden. Die externe Darstellung erleichtert, auf Ausschnitte zuzugreifen, also auch optisch zu fokussieren.

Das Notieren selbst, das natürlich auch der Dokumentation dient, zielt außerdem bereits auf die Präzisierung, Reflexion und Filterung. Dafür muss zwar im Gespräch die Strukturierung mit diskutiert werden, sie ist jedoch lange nicht so weitgehend, wie es für formale Wissensrepräsentationen mit dem Ziel formaler Schlussfolgerungen notwendig wäre. Ein Ziel ist sogar, dass der Mehrwert für die Nutzerinnen und Nutzer durch die strukturierenden Hilfsmittel und die Dokumentation so hoch ist, dass er den Aufwand, der durch die Integration eines Tools in die Diskussion entsteht, nivelliert.

Suthers (1999A) beschreibt die Abhängigkeit zwischen der Visualisierung und der Aufgabenstellung („representational bias“). Die Art der Darstellung stellt gleichzeitig auch eine Perspektive dar, unter der man eine Fragstellung betrachtet. Für bestimmte Aufgaben oder Fragestellungen können entsprechend bestimmte Hilfsmittel besser oder schlechter geeignet sein. Einerseits bestimmt die Repräsentation, welche Information wieder aus ihr entnommen werden kann, andererseits kann sie aufzeigen, welche Informationen fehlen.

Daraus, dass unterschiedliche Repräsentationsmittel verschiedene Aspekte in den Vordergrund stellen, entwickelt Jonassen und Carr (2000) einen „active learning strategies”-Lernansatz, dessen Kernidee im bewussten Einsatz unterschiedlicher Repräsentationsmittel besteht.

Kommunikationsmedium. In dieser Arbeit werden die Darstellungen mit visuellen Sprachen als prozessorientierte Kommunikationsmedien begriffen, die während der Wissenskommunikation, der Kommunikation von Menschen zum Wissensaustausch oder zur Wissenskonstruktion (siehe auch Kapitel 2.5), Verwendung finden.

Es wird im Weiteren keine Einordnung in eine umfangreiche Mediendebatte vorgenommen. Stattdessen wird nur ein Aspekt von Medien herausgegriffen: Problematisiert wird eine zu passive Sicht auf Computermedien im Sinne von Informationsmedien, die bis heute weitgehend dem Paradigma von Büchern, Zeitungen oder Fernseh- bzw. Videoübertragungen folgen (Kay, 1998), also der Idee von Massenmedien. Vernachlässigt wird, dass gerade der Entstehungsprozess dieser Resultate oft durch eine

ausgiebige Wissenskommunikation begleitet wird, in der vielfältige Darstellungsmittel benötigt werden. Das Verstehen von Zusammenhängen und Sachverhalten ist weniger ein passives Rezipieren, als ein aktiver Umgang mit Inhalten, die in den Kommunikationsprozess eingehen. Gerade in der aktiven Gestaltung besteht im Prinzip eine Stärke von Computermedien.

Erläutert wird dazu punktuell der Medienbegriff nach Dewey in der Interpretation von Vogel (2001). Kommunikative Aspekte stehen darin im Vordergrund, obwohl Deweys Medienbegriff speziell für die Kunstrezeption erläutert wird. Betont wird, dass vor allem vielfältige Ausdrucksmöglichkeiten notwendig sind, die am Besten auch individuell formbar sind.

Auf Grundlage dieses Medienbegriffs ist die Rolle, die man Medien beimisst, neu zu überdenken. Meistens wird ihnen eine eher repräsentationale Rolle in einem Produzenten-/Konsumentenverhältnis zugedacht bei dem die Medien ihre Interpretationsgrundlagen mitliefern müssen (z.B. statische Darstellungen von Informationen auf Webseiten). In dieser eher ökonomischen Sicht besteht kaum ein Handlungsspielraum für die Konsumentenseite. In dem von Vogel dargelegten Medienbegriff ist das nicht so. Die Kommunikation mit Hilfe von Medien ist danach ein Ausgangspunkt rationalen Verstehens bei dem die Kommunizierenden situativ und individuell eine gemeinsame Ausdrucksweise entwickeln. Das Medium prägt dabei die Ausdrucksweise: Einfluss haben sowohl die technischen Gegebenheiten des Mediums als auch die Möglichkeiten des jeweiligen Notationssystems. Der – nach Dewey ästhetische – Erfahrungshintergrund manifestiert sich sowohl in der Gestaltung der Mittel als auch in deren Rezeption.

Den Einfluss, den Medien auf die Wissensgenese ausüben, charakterisiere ich als aktiv oder passiv. Ein passiver Einfluss eines Mediums entsteht schon dadurch, dass eine visuelle Sprache verwendet wird, die den Sprachumfang und die Notation bzw. die Symbole festlegt. Die Wirkung von Darstellungen beeinflusst maßgeblich den Verlauf der Kommunikation. Ein aktiver Einfluss des Mediums liegt dann vor, wenn es in die Gestaltung eingreifen kann, wenn es direkten Einfluss auf den Prozess nimmt also z.B. wenn es eine Methode definiert. Selbstverständlich ist der Übergang zwischen passiven und aktiven Medien fließend.

Vergleichbare Ansätze. Es existieren wenige Systeme, die explizit für eine Diskussionsunterstützung in einem „face-to-face“-Szenario implementiert wurden, obwohl die meisten Gesprächssituationen noch immer „face-to-face“ stattfinden (z.B. Schule, Sitzungen). Unter einer „face-to-face“-Situation wird verstanden, dass sich die Beteiligten am gleichen Ort befinden und direkten Blickkontakt haben. Gerade weil nicht

die verteilte Situation betrachtet wird, treten hier die Darstellungsmittel, die die Software anbietet, in den Vordergrund und werden nicht von der technischen Ausgestaltung der Kommunikation abgelenkt. Verglichen werden deshalb Ansätze, die ebenfalls die gestalterischen Mittel problematisieren (siehe dazu Kapitel 3.2).

Zu erkennen sind vier zentrale Motivationen:

1. Sitzungsunterstützung
2. Kreativitätsunterstützung
3. Begründung und Entwicklung von Designs
4. Lernen

- *Sitzungsunterstützung.* Ein vergleichbares System stellt das Sepia-System (Streitz, Haake, et al., 1992; Streitz, Haake, et al., 1998) dar, insbesondere in seiner Ausprägung zum Dolphin-System (Streitz, Geißler, Haake & Hol, 1994). Sepia bietet vier „Diskussionsräume" mit spezifischen visuellen Sprachen an. Dolphin erweitert das Sepia-System vor allem um eine intuitive Benutzung z.B. durch Gestenerkennung sowie durch die Möglichkeit handschriftlicher Eingaben. Der Arbeitsprozess wird im Sepia-System hauptsächlich durch das Design der visuellen Sprachen unterstützt, also durch die Typen der zur Verfügung stehenden Objekte. Die automatisierten Abbildungen von Inhaltsobjekten beziehen sich ausschließlich auf einzelne Objekte.

  Ebenfalls gut vergleichbar sind die Systeme Cognoter und Argnoter (Stefik, Foster, Bobrow, Kahn, Lanning & Suchman, 1987), die aus dem bei XEROX PARC angesiedelten CoLab-Projekt hervorgegangen sind. Cognoter ist zur kooperativen Vorbereitung von Präsentationen entwickelt worden, Argnoter zur Diskussionsbegleitung. Beiträge werden als Knoten in Graphen notiert. Jeweils wird eine Vorgehensweise aus Brainstorming, Organisation und Evaluation (Cognoter) bzw. vorschlagen, diskutieren und auswerten (Argnoter) empfohlen aber nicht durch die Systeme herbeigeführt.
- *Kreativitätsunterstützung.* Der Bedarf an Systemen zur Kreativitätsunterstützung ist vor allem in den wissensorientierten Branchen (z.B. Forschung, Design, Consulting, Software, Werbung und Medien) in den vergangenen Jahren stark gestiegen. Die dafür eingesetzten Systeme arbeiten meistens mit visuellen Sprachen, sind aber meistens auf ganz konkrete Methoden ausgerichtet. Viele bieten das „Mind Mapping" an (z.B. [MindManager], [visimap]), bei dem Ideen oder Vorstellungen in „Landkarten" dargestellt werden. Häufig können allerdings nur baumartige

Strukturen entwickelt werden. Typischerweise wird eine automatische, optisch ausgewogene Strukturierung angeboten[2].

- *Begründung und Entwicklung von Designs („Design Rationale").* „Design Rationale" (Moran & Carroll, 1996) steht für einen ganzen Forschungsbereich bei dem es darum geht, Software- sowie Produktdesign so zu unterstützen, dass der Entwicklungsprozess transparent wird, daraus Fehler erkannt und diese in weiteren Entwicklungsschritten vermieden werden können. Viele Systementwicklungen in diesem Bereich gehen davon aus, dass die Strukturierung und explizite Darstellung von Argumenten während des Designs ein durchdachteres Vorgehen bewirkt und vor allem das Re-Design unterstützt. Besonders bekannt geworden ist der gIBIS-Ansatz (Conklin & Begemann, 1987). Aus „Design Rationale"-Ansätzen sind schon früh Betrachtungen über die Gestaltung und den Einsatz von visuellen Sprachen hervorgegangen (Buckingham Shum & Hammond, 1994).
- *Lernen.* Im Lernbereich werden ebenfalls visuelle Sprachen eingesetzt. Beispielsweise sollen mit dem Lehr-/Lern-System Belvedere (Suthers, Weiner, Connelly & Paolucci, 1995) Prinzipien des wissenschaftlichen Argumentierens vermittelt und trainiert werden. Argumentationsprinzipien können anhand der Strukturen überprüft werden und ermöglichen Hinweise durch ein intelligentes Hilfesystem.

In Kapitel 3.2 werden diese und ein paar weitere Systeme noch detaillierter erläutert und verschiedene Defizite aufgezeigt. Vorweggenommen sei hier, dass in den meisten Systemen Informationen, die allein durch die Art der Darstellung unmittelbar vorliegen, kaum ausgenutzt werden. Dazu gehören Art, Ort, Autor, Kontexte und Interaktionen. Denkbar wären attributgesteuerte, automatische oder benutzerdefinierte Anordnungen, Abstraktionen, Ausblendungen, etc. Außer bei den Systemen zum „Design Rationale" wird die Wiederverwendung von Materialien und die Integration von Informationsressourcen noch kaum thematisiert. Spezifische Ablageformate erschweren die Interaktion mit anderen Werkzeugen und Suchmaschinen. Insbesondere die großen, kompakten Systeme wie Sepia oder gIBIS waren zumindest in den zitierten Versionen sehr geschlossene Systeme. Darunter leiden potentiell vor allem zeitlich ausgedehnte Arbeits- und Lernprozesse.

Im Rahmen der wenigsten Systementwicklungen wird Stellung dazu bezogen, wie realistisch der Nutzungsaufwand in Relation zu den zu erwartenden Vorzügen ist. Bei vielen Ansätzen ist unklar, wie die Systemnutzung in den täglichen Arbeitsprozess

[2] In der Computerzeitung c't (c't: 20/1998) findet sich dazu eine interessante Zusammenschau von Systemen. Siehe dazu Referenz Jungbluth (1998).

integriert werden soll. Zu erwarten ist eine große Abneigung, wenn die Nutzung der Systeme nicht an anderen Stellen eine Arbeitserleichterung bewirkt. Möglich wären gerade auf Basis semi-formaler Sprachen teil-automatisierte Dokumentationsprozesse.

Abgrenzung: „Dialogue-Games". Der hier angestrebte Prozess der Wissenselaboration, zielt auf einen Übergang von informellen hin zu stärker strukturierten Darstellungen. In diesem Prozess bildet das Kommunikationsmedium gerade nicht den alleinigen Kommunikationskanal, sondern erweitert die Ausdrucksmöglichkeiten. Dadurch sind deutlich weniger schriftliche Äußerungen notwendig als beispielsweise in einem Chat und die Darstellung hat bereits eine filternde Funktion in Bezug auf relevante Beiträge. Die Relationen zu benennen, in denen diese Beiträge zueinander stehen, stellt ein ganz wesentliches Element einer Strukturierung dar. Zur Strukturierung werden visuelle Sprachen verwendet, die sich an der Idee epistemischer Spiele orientieren, die zur Wissenskonstruktion und zum Erkenntnisgewinn führen sollen (vgl. Kapitel 6.1).

Davon abzugrenzen sind „Dialogue-Games" (Levin & Moore, 1977), die typische Kommunikationsmuster in Bezug auf deren Funktion klassifizieren, wie „helping", „information-seeking", „information-probing" und „instruction" (Levin & Moore, 1977, S. 399-400). Dafür werden linguistisch-analytische Herangehensweisen gewählt, die nicht unmittelbar dazu eingesetzt werden können, Beiträge und Kommunikationsmuster schon während einer Kommunikationssituation zu bestimmen. Das Ziel von Levin und Moore besteht auch nicht darin, sondern sie wollen unter anderem zeigen, wie Sprache zielgerichtet eingesetzt werden kann.

Ebenfalls analytisch setzen Pilkington und Mallen (1996) „Dialogue-Games" ein. Sie untersuchen Gesprächsmuster, die in 2- und 3-Personensettings aus einem oder zwei Lernenden und jeweils einem Tutor bzw. Tutorin entstehen. Im Mittelpunkt steht die Frage, welchen Einfluss die Situation auf die Reflexion und damit auf den Lerneffekt hat. Es zeigte sich, das insbesondere „inquiries", „hints" und „prompts" weiterführende Begründungen hervorriefen, die dort als reflexionsorientiert angesehen werden (Pilkington & Mallen, 1996, S. 216). Diese „moves" traten in der Bedingung mit einer lernenden Person prozentual deutlich häufiger auf.

Neben beispielsweise lerntheoretischen Ansätzen (vgl. VanLehn et al., 1992; Chi et al., 1994; Palinscar & Brown, 1989) stützen auch linguistische Analysen die Annahme, dass Lernen in direktem Zusammenhang damit steht, dass reflektiert und mit Inkonsistenzen sowie Widersprüchen umgegangen wird, um Begriffsänderungen hervorzurufen und Umstände neu zu interpretieren (vgl. Pilkington & Mallen, 1996). Baker (1995) beschreibt einen Lernprozess als Verhandlungsprozess („negotiation") in dem „moves"

Funktionen wie „Angebot“ und „Attacke“ verfolgen, worin sich inhaltliche Widersprüche erkennen lassen.

„Dialogue-Games“ sind zu anspruchsvoll, um sie während einer Kommunikation als Muster einzusetzen, nach dem Beiträge klassifiziert werden. Sie können jedoch bei der Dialoggestaltung eingesetzt werden. Manche Ansätze geben für die Gestaltung von Dialogen Satzeröffnungen vor (z.B. Robertson, Good & Pain, 1998). Diese Satzfragmente legen die Nutzerinnen und Nutzer auf bestimmte Aussageformen fest und sollen durch diese Normierung Effekte auf die inhaltliche Gestaltung ausüben (Baker, de Vries, Lind & Quignard, 2001). Baker und Lund (1996) können in einem Vergleich auf Basis des Systems C-Chene zeigen, dass solche Satzeröffnungen durchaus den Anteil inhaltsbezogener Äußerungen erhöhen kann. Allerdings können Dialogmuster typischerweise nur in Zusammenhang mit einem vollständig verschriftlichten Dialog konstruktiv eingesetzt werden. Sie bieten sich deshalb eher für die Gestaltung einer Mensch-Computer-Interaktion an, sowie für die Beeinflussung von Lerndialogen in verteilten Szenarios.

Szenario. Im Verlauf dieser Arbeit werden zwei Systeme vorgestellt, das DiscBoard (Kapitel 6) und der FreeStyler (Kapitel 7), die beide das Szenario einer technisch angereicherten Diskussionssituation bedienen (Abbildung 1), in der sich alle Beteiligten sehen und auch im herkömmlichen Sinne miteinander kommunizieren können.

Das Szenario dieses Ansatzes ist damit eine technisch angereicherte „face-to-face“-Situation, in der die eigentliche Diskussion wie bisher mündlich geführt wird. Die Technologien werden dazu genutzt, um digitale Darstellungen anzufertigen, die Kooperation technisch zu unterstützen, diverse Medien zur Verfügung zu haben, ohne Medienbrüche zu provozieren und Software zur methodischen Unterstützung einzusetzen. Externe Darstellungen bieten in dieser Situation einen gemeinsamen Fokus. Ob es sich dabei um eine gut geeignete aber aufwendige Rückprojektion, eine Projektion auf ein interaktives Whiteboard oder eine einfache Projektion des Computerbildschirms handelt, spielt eine untergeordnete Rolle.

Zusätzlich zur elektronischen Tafel können die beteiligten Personen durchaus individuell mit (vernetzten) Computern ausgestattet sein. Ob dies von Vorteil ist, hängt weitgehend von der angestrebten Situation ab, ob beispielsweise Einzelarbeitsphasen mit einbezogen werden, wie in Lernszenarien im Schulunterricht oder individuelle Arbeitspakete, die dann später zusammengeführt werden, oder auch, ob die verwendete Software gemeinsame Arbeitsbereiche und individuelle Annotationen überhaupt erlaubt.

Die ausschließliche Verwendung individueller Rechner, hätte einen verteilten Fokus zur Folge. Jede Person würde dabei vermehrt den eigenen Display betrachten und so von der üblichen Gesprächsinteraktion abgelenkt sein. Dabei besteht die Gefahr, dass die gegenseitige Ansprache behindert wird, was demotivierend wirkt. Schon allein die Konzentration auf den eigenen Bildschirm kann verhindern, der Dynamik der Situation zu folgen und diese mit zu gestalten. Die Beteiligten entgleiten dann hinter ihren Computern der Diskussion. Dieses Phänomen kann durch große und schlecht platzierte Bildschirme negativ verstärkt werden, wenn schon dadurch der Sichtkontakt behindert ist.

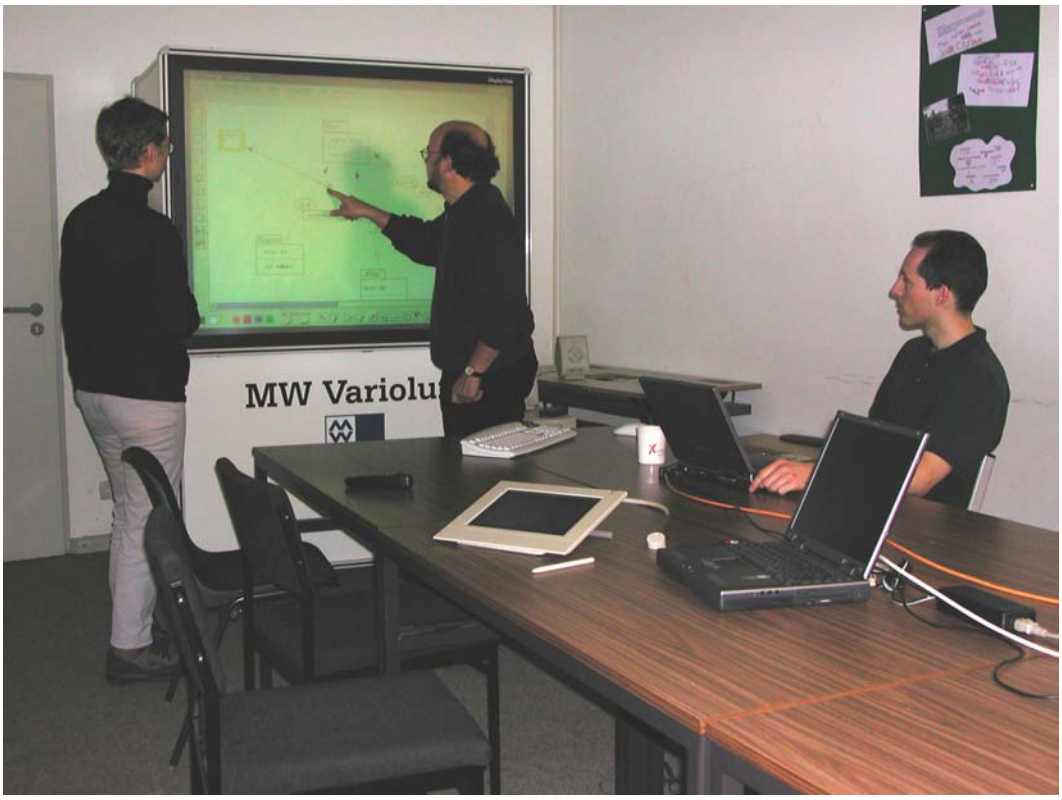


Abbildung 1

„face-to-face"-Kommunikationsszenario.

Sowohl das DiscBoard als auch der FreeStyler setzten auf einer verteilten Architektur auf, die mehreren Personen die synchrone Eingabe in gemeinsame Arbeitsbereiche erlaubt. Aus dieser komfortablen Konstellation ergeben sich verschiedene Optionen, die Kommunikation zu gestalten.

Die Architektur kann genutzt werden, um beim komfortablen Austausch von Inhalten Medienbrüche zu vermeiden. Auch kann mit ihr die Arbeitssituation offener gestaltet werden, wenn mehrere Personen von persönlichen Computern aus Eingaben auf einer zentralen Tafel anfertigen können. Vorbereitung, Ergebnisintegration und Präsentation

können unterstützt werden. Evtl. können dadurch auch Ängste vermindert werden, sich alleine an der Tafel darzustellen, und anonyme Eingaben ermöglicht werden. Außerdem ist ein schnellerer Wechsel der Eingaben zwischen den Personen denkbar.

Systementwicklungen zur Ko-Konstruktion von Wissen mit visuellen Sprachen. Im Rahmen dieser Arbeit wurden zwei Systeme entwickelt, die in Gesprächs- bzw. Diskussionsszenarien verwendet werden sollen, um in der Gruppe Wissen zu ko-konstruieren bzw. zu elaborieren. Als Ausdrucksmittel werden jeweils visuelle Sprachen verwendet, in einem Fall ergänzt um die Möglichkeit, handschriftliche Eingaben – Notizen und Skizzen – anzufertigen. In beiden Fällen wird die „beiläufige" Externalisierung während der Wissenskommunikation als Ausgangspunkt für die Dokumentation besprochener Inhalte gesehen.

Beide Systeme integrieren einen Struktur- und einen Prozessansatz. Strukturorientiert bedeutet, dass die logische Struktur der Inhalte im Vordergrund steht (Conklin & Burgess-Yakemovic, 1996). Dies wird durch die visuellen Sprachen unterstützt. Die Prozessorientierung wird jedoch nicht, wie in Conklin und Burgess-Yakemovic (1996) ausgeführt, durch die Repräsentation der Historie der Inhalte erreicht, sondern durch die implizite Unterstützung eines Arbeitsablaufes.

Die Systeme DiscBoard und FreeStyler, die in den folgenden beiden Abschnitten eingeführt werden, stellen jeweils andere Aspekte in Bezug auf ihre Strukturierungsmöglichkeiten und ihre Prozessunterstützung in den Vordergrund. Während mit dem DiscBoard ein Perspektiven-Ansatz vorgestellt wird, in dem automatisch Inhalte zwischen den Perspektiven gefiltert und restrukturiert werden, fokussiert der FreeStyler auf kreative Darstellungsmöglichkeiten, die Dokumentation und die Wiederverwendung systemeigener aber auch externer Dokumente.

Nach Jonassen (1992) sind kognitive Tools dadurch charakterisierte, dass mit ihnen Lernende zwischen Wissenseinheiten Zusammenhänge herstellen können, um damit Wissen zu konstruieren. Demnach können beide Systementwicklungen als kognitive Tools (Jonassen, 1992; Lajoie, 1993) eingeordnet werden, die einen kreativen Prozess bzw. Denkprozesse anregen und unterstützen.

- *DiscBoard.* Das System DiscBoard (Kapitel 6) unterstützt schwerpunktmäßig das Elizitieren und die darauf folgende Arbeit mit den Inhaltsobjekten. Auf Basis von visuellen Sprachen werden Perspektiven realisiert, die epistemische Formen (vgl. Kapitel 6.1) als Zielstrukturen zur Verfügung stellen (Collins & Ferguson, 1993). Die visuellen Sprachen bilden die Mengen der Kategorien und Relationen, unter denen ein Phänomen betrachtet wird. Über die visuellen Sprachen wird implizit ein

epistemischer Prozess unterstützt, ein Prozess der Wissenselaboration. Es wird davon ausgegangen, dass sich der Diskussionsprozess in Phasen mit spezifischen Vorgehensweisen und dazu passenden Darstellungsformen unterteilt. Die Perspektiven harmonisieren mit den verschiedenen Phasen der Diskussion und dienen der variablen Sicht auf die Inhalte.

Allgemein sind Ansätze, die Perspektiven auf Inhalte ermöglichen, immer dort sinnvoll einsetzbar, wo die darzustellenden Inhalte sehr komplex sind oder nur unvollständig repräsentiert werden können oder sollen, denn durch die expliziten Perspektiven sollen die Kriterien der Darstellungen transparent werden. In Diskussionen können Inhalte generell nur in Ausschnitten aufgearbeitet werden. Trotzdem sollen die relevanten Aspekte nicht übersehen werden. Durch den Einsatz von Perspektiven sollen Inhalte besser verstanden und ihre Vollständigkeit hinterfragt werden.

Der Perspektivenbegriff legt die Idee des Perspektivenwechsels nahe. Mit dem DiscBoard wird eine Möglichkeit vorgestellt, die Perspektivenwechsel, die Übergänge zwischen den Perspektiven, zu unterstützen. Basierend auf der automatischen Erkennung, Filterung und Interpretation von Strukturen werden Inhalte aus einer Perspektive in andere - potentiell strukturell verändert - übernommen und so der Arbeitsprozess unterstützt. Diese Operationen reagieren auf Zustände der aufgebauten Strukturen und führen dazu, dass in anderen Perspektiven Strukturen eingefügt werden.

Die Abbildungen basieren auf einem Diskussionsmodell. Ihr Hauptziel besteht darin, weitere Diskussionsschritte zu motivieren und strukturierte Übersichten über die diskutierten Inhalte anzubieten. Von diesem Initialzustand aus können die Inhalte in den Perspektiven unabhängig voneinander entwickelt werden. Es wird keine Konsistenz zwischen den Perspektiven verlangt.

Das DiscBoard stellt eine Erweiterung und Spezialisierung des CardBoard-Systems dar (Mühlenbrock, Tewissen & Hoppe, 1997; Gaßner, Tewissen, Mühlenbrock, Loesch & Hoppe, 1998; Hoppe, Gaßner et al., 2000), welches wesentliche Grundfunktionalitäten für die Arbeit mit visuellen Sprachen zur Verfügung stellt. Das CardBoard-System wird vorbereitend zur Beschreibung der DiscBoard-Entwicklung (Kapitel 6) in Kapitel 3.1.3 erläutert.

- *FreeStyler*. Das zweite System, der FreeStyler (Hoppe & Gaßner, 2002; siehe Kapitel 7), legt den Schwerpunkt auf das Wissensmanagement, auf die kreative Darstellung von Inhalten und deren Wiederverwendung. Bei dessen Entwicklung

wurde zum gesamten Prozess der Dokumentation, des Retrievals und des Aufbaus von Repositories Stellung genommen. Kombiniert wird ein kommunikationsorientierter Ansatz mit einem Datenverwaltungsansatz. Dies ist insofern ungewöhnlich, als dass in den meisten Systementwicklungen nur auf einen von beiden Aspekten eingegangen wird. Viele Systeme sollen hauptsächlich technische Lösungen für das Wiederfinden, die Verwaltung und die Analyse von Daten bieten (z.B. Datenbanken und Systeme zum „data mining"). Kommunikationsorientierte Ansätze sind dagegen fast immer an organisatorische, also nicht-technische Maßnahmen gebunden.

Der FreeStyler ist in ganz verschiedenen Szenarien nutzbar, die während eines Arbeitsprozesses durchlaufen werden, wie Telefongespräche, Sitzungen, Vorbereitung und Dokumentation. Bei der Entwicklung des FreeStylers wurde davon ausgegangen, dass Wissen allmählich, während des Arbeitsalltags, konstruiert wird. Unterschiedliche Informationsquellen sollten als Wissenselemente integrierbar sein, um daraus Informationsstrukturen aufzubauen. Die Darstellungen zielen auf strukturierte Übersichten von Inhaltsbereichen. Für diese Strukturen stehen auch hier visuelle Ausdrucksmittel zur Verfügung, die zwar allgemeiner aber vielfältiger gehalten sind als beim DiscBoard.

Der FreeStyler unterstützt die Wissensexternalisierung, Strukturierung, den Zugriff auf Daten und die Suche nach Daten. Neben der asynchronen Arbeitssituation stellt der FreeStyler aber vor allem die Mittel zur Verfügung, schon während der Gespräche gemeinsam Probleme zu strukturieren und Notizen, auch handschriftlich, anzufertigen. Existiert ein „Repository" in dem Dokumente, Medien oder Informationen verwaltet werden, so kann der FreeStyler zur Darstellung von Gegenstandsbereichen herangezogen werden und stellt außerdem die Möglichkeit zur Verfügung, eine übergeordnete, strukturierende Sicht auf das „Repository" einzunehmen.

Insgesamt wird somit ein Wissenszyklus unterstützt, in dem beginnend von schlichten Notizen stärker überarbeitete Darstellungen entwickelt werden, gleichzeitig externe Informationen integrierbar und referenzierbar sind und wieder durch Notizen ergänzt werden können. Die Wiederverwendung sowohl von FreeStyler-Dokumenten als auch von anderen Informationsquellen wird einerseits durch die Nutzungsvarianz gefördert, andererseits maßgeblich durch Retrieval-Funktionen unterstützt. Die in einer Arbeitsgruppe wechselseitig entwickelten Informationsstrukturen dienen dann auch dem Informationsaustausch. Sie stellen damit ein externes Gedächtnis dieser Gruppe dar.

Ausblick

Diese Arbeit gliedert sich in zwei Teile. Im ersten Teil wird allgemein auf Diskussionen eingegangen und das Themenspektrum visueller Sprachen aufgearbeitet. Vorgestellt und analysiert werden dann verschiedene Systeme, an denen Ansätze und offene Fragen dargestellt werden.
Der zweite Teil stellt die Systeme DiscBoard und FreeStyler vor, die im Rahmen dieser Dissertation realisiert wurden.

Teil 1:

Kapitel 1 befasst sich mit Diskussionen, in welchem Kontext sie stattfinden und wann ein Bedarf an Diskussionen und Diskussionsunterstützung besteht. Eingegangen wird auch auf kritischen Einschätzungen.
Kapitel 2 beschäftigt sich mit visuellen Sprachen. Anhand von Zielsetzungen wird deren Einsatz aufgeschlüsselt.
Kapitel 3 stellt verschiedene Systeme vor, die visuelle Sprachen für kommunikative Szenarios zur Verfügung stellen. Diese Systeme werden verglichen und der Versuch unternommen, offene Entwicklungsrichtungen daraus zu begründen.
Kapitel 4 referiert eine Vorstudie zur Entwicklung des DiscBoards.
Kapitel 5 stellt kurz den Ansatz des DiscBoards sowie den des FreeStylers vor.

Teil 2:

Kapitel 6 stellt die Entwicklung und in Ausschnitten die Implementierung des DiscBoards vor.
Kapitel 7 stellt die Designentscheidungen und die Architektur des FreeStylers vor.
Kapitel 8 bietet eine Zusammenfassung und kritische Einschätzung der Systeme. In einem Ausblick werden mögliche weitere Schritte erläutert.

TEIL I

VISUALISIERUNGEN UND REPRÄSENTATIONEN ZUR DISKUSSIONSUNTERSTÜTZUNG: PROZESSE DER WISSENSELABORATION

1 DISKUSSIONSUNTERSTÜTZUNG

Dieses Kapitel soll streiflichtartig erklären, an welchen Punkten eine computerbasierte Diskussionsunterstützung ansetzen kann. Es wird deutlich werden, dass die Stärken entsprechender Tools nicht darin liegen, das Wissen und die Kreativität einzubringen, sondern beides „hervorzulocken“ und gleichzeitig Dokumentationsarbeit zu leisten. Erläutert wird in diesem Kontext die Bedeutung von Diskussionen als Basis für Formulierung von Wissen bei der kooperativen Arbeit. Sie können in Betrieben und allgemein beim Lernen gezielt zur Förderung des Informationsflusses und von Wissensbildungsprozessen eingesetzt werden. Allerdings sind Diskussionen kein leicht zu kontrollierendes Vorgehen. Deshalb wird im Folgenden auch auf Probleme und Phänomene eingegangen, die sich negativ auf den Verlauf von Diskussionen auswirken können.

1.1 Diskussionen

Zunächst soll geklärt werden, was weithin unter Diskussionen verstanden wird. Diskussionen werden als Situationen definiert, in denen kleine Gruppen „face-to-face“, miteinander interagieren (z.B. Blicke, Gesten, Sprache, Medien), um Informationen auszutauschen, die dazu führen, für ein gemeinsames Problem eine Entscheidung herbeizuführen (siehe dazu Gully, 1968; Bormann, 1975; Meack, 1990). Dafür wird typischerweise das Problem analysiert, mögliche Lösungen gesucht und im Allgemeinen davon ausgegangen, dass alle Beteiligten zumindest versuchen, eine Entscheidung anzustreben, die für alle in Frage kommt.

Bei dem in einer Diskussion behandelten Thema sollte es sich tatsächlich um ein gemeinsames Problem handeln, das außerdem eine gewisse Komplexität aufweist. Daraus kann zumindest der Bedarf nach einem zielgerichteten Vorgehen und einer allgemein akzeptierten Lösung abgeleitet werden („orderly development“, Gully, 1968).

Natürlich ist nicht jedes Gespräch eine Diskussion. In Maeck (1990) wird die Diskussion als Spezialfall von Gesprächen aufgeführt. Gully (1968) unterteilt Diskussionen in zwei Typen. Eine Art dient im Wesentlichen dem Informationsaustausch, die andere der Entscheidungsfindung. Das entspricht auch der Einteilung nach Lehmann (1998). Dort werden Kommunikationen mit Sachinhalten (keine Beziehungs-/ Klärungsgespräche) in vier Gruppen unterteilt: Informations- bzw. Lehrveranstaltung, Problem- und Innova-

tionsbesprechungen, Verhandlungen und Befragungen. Problem- und Innovationsbesprechungen sowie Verhandlungen können Diskussionen sein. Davon sind erstere eher dem Informationsaustausch, letztere eher der Entscheidungsfindung zuzuordnen. Natürlich können verschiedene Kommunikationstypen aufeinander folgen. So geht beispielsweise der Entscheidungsfindung normalerweise ein Informationsaustausch voraus.

Typische Arten von Diskussionen zum Informationsaustausch sind Studiengruppen (*Ist der neue Markt eine geeignete Geldanlagemöglichkeit?; Wie werden Projekte sinnvoll organisiert und woran scheitern sie?*) oder „runde Tische".

Komitees (*z.B. Wie kann man gegen Gewalt von „Rechts" vorgehen?*) oder Boards (*z.B. Wie kann man es sinnvoll schaffen, dass Schulen mit Internetzgängen ausgestattet werden?*) sollen dagegen oft Entscheidungsempfehlungen ausarbeiten oder sogar umsetzen. Die folgende Liste zählt Merkmale auf, die typisch für Diskussionen sind:

- Sie werden in kleinen Gruppen geführt (Bormann, 1975).
- Es wird „face-to-face" kommuniziert (Bormann, 1975; Gulley, 1968)[3].
- Es wird zu allen, in die Gruppe gesprochen (Bormann, 1975).
- Eine Diskussion kann aus mehreren Treffen bestehen (Bormann, 1975).
- Es gibt eine Interaktion zwischen den Teilnehmenden (Gulley, 1968).
- Alle Teilnehmenden beteiligen sich (Bormann, 1975; Gulley, 1968).
- Es gibt ein gemeinsames Ziel und die Erwartung einer zielgerichteten Diskussion (Bormann, 1975; Gulley, 1968; Maeck, 1990).
- Es wird ein spezifiziertes Problem erörtert, Informationen dazu ausgetauscht und Lösungen gesucht. (Maeck, 1990; Gully, 1968).

Diese Kriterien deuten eine Vielzahl von Einflussgrößen auf den Diskussionsverlauf an sowie mögliche Störungen, wenn sich die Personen nicht entsprechend verhalten. Zu den Einflussgrößen gehören außerdem der Zeitpunkt der Reaktion in einem Verlauf zwischen synchron und asynchron, die räumliche Verteilung der Beteiligten und die Größe der Gruppe. Relevant ist auch die Verteilung von Expertise in der Gruppe und implizite oder explizite Rollenverteilungen. Sehr leicht werden Diskussionen negativ beeinflusst, wenn Personen aus der Interaktion ausgeschlossen oder Machtpositionen ausgenutzt werden. Unterschiedliche Erwartungshaltungen können genauso hinderlich sein wie unterschiedliche Zielsetzungen. Auf diesen Fragenbereich wird in einem der folgenden Unterkapitel (1.4) vertieft eingegangen.

[3] Dass ausschließlich die „face-to-face"-Situation für Diskussionen als relevant angenommen wird, ist meiner Ansicht nach hauptsächlich auf den Erscheinungszeitpunkt der zitierten Literatur zurückzuführen. Gully weist allerdings schon darauf hin, dass eine rein schriftlich geführte Diskussion eine interessante Veränderung der Situation hervorrufen würde.

1.2 Diskussionen als Mittel der Wissenskonstruktion

Die Rolle von Diskussionen kann einerseits in einem gesellschaftlichen Kontext betrachtet werden, andererseits konkreter als methodischer Ansatz, um bestimmte Ziele zu erreichen. Pea (1993) argumentiert auf einer gesellschaftlichen Ebene, dass sich die Rolle von Diskussionen in den letzten Jahrzehnten verschoben hat. Er sieht, dass heute Probleme häufiger kooperativ gelöst werden, und sich deshalb auch der Intelligenzbegriff mehr auf Gruppen als auf Einzelne ausdehnen sollte. Seine eigentliche Argumentation ist allerdings viel radikaler. Er erläutert, dass jede intelligente Leistung verteilt ist („distributed cognition"), dass es resultierend keine intelligente Leistung ohne die Kommunikation gibt. Soziales und kulturelles Wissen wird darin nicht nur über Personen kommuniziert, sondern ist auch in Objekten im Sinne gesellschaftstypischer Artfakte kodiert. Implizit enthalten diese Wissen, angefangen vom Busfahrplan über Kontroll-instrumente bis hin zu Notationssystemen.

Wissensgewinn erscheint nach Pea (1993) deshalb immer als weitgehend individuelle Wissenskonstruktion, die im größeren Kontext gesellschaftlich beeinflusst ist. Durch die Notwendigkeit der Kommunikation für intelligente Handlungen haben vor allem individuelle Ansichten und „herrschende Meinungen" einen wesentlichen Einfluss auf die jeweiligen Überzeugungen. Die Gesamtheit der „Überzeugungen" einer Gesellschaft formt das gesellschaftliche Wissen, dass selbst wieder neues Wissen prägt. Pea (1993) leitet daraus u.a. Lerndesigns ab.

Ebenfalls eine zyklische Beschreibung zur Wissenskonstruktion, die die Kommunikation in den Mittelpunkt rückt, wird auch in Stahl (2000) vorgestellt. Die individuellen Ansichten beruhen darin auf dem subjektiven Verständnis der Umwelt. Durch die Artikulation dieses Vorverständnisses werden die Ansichten als öffentliche Bemerkungen diskutierbar. Mit anderen Personen kann dann argumentiert und können Begriffe geklärt werden, um so ein gemeinsames Verständnis aufzubauen. Dieses Wissen kann sich zu einem gesellschaftlichen Wissen entwickeln und sich letztlich auch wieder in Artefakten niederschlagen. Dieser Prozess beeinflusst dann wieder das subjektive Verstehen der Individuen. Stahl (2000) empfiehlt als ein Resultat daraus die Abbildung der Phasen der Wissenskonstruktion auf Software-Komponenten zur Unterstützung kooperativer Wissensbildung.

Gerade in diesen gesellschaftlichen Ansätzen zur Wissenskonstruktion wird immer wieder betont, dass Weitergabe und Re-Formulierung der Inhalte den eigentlichen Konstruktionsprozess ausmachen. Es gibt immer wieder Zyklen aus:

- Externalisierung unter Zuhilfenahme von Hilfsmitteln (z.B. Repräsentationsformalismen, Skizzen, Sprache, etc.).
- Überarbeitung der dargestellten Inhalte (z.B. restrukturieren, argumentieren, Bezüge herstellen, etc.).
- Internalisierung der Inhalte[4].

Ein solcher Prozess der Wissensbildung wird explizit angestrebt, wenn Diskussionen im Rahmen von organisatorischen, betrieblichen Maßnahmen oder beim Lernen provoziert werden. Dass es sich dabei in Betrieben um eine bewusste Methodik der Betriebsorganisation handelt, ist allerdings eine neuere Tendenz, die Theorien aus dem Bereich des Knowledge Managements aufgreift. Darin hat der Informationsfluss eine zentrale Bedeutung (Borghoff & Pareschi, 1998) und die Diskussion ist eine Möglichkeit, diesen unkompliziert zu verbessern. Auch der Prozess der Wissensbildung in den oben genannten Phasen wird beim Wissensmanagement betont (Nonaka, 1994; Nonaka & Takeuchi, 1995). Am Informationsfluss hängt das Entstehen eines innerbetrieblichen, gemeinsamen Wissensbestandes, der wichtig für die interne Kooperation und generell für die Weiterentwicklung und Selbsteinschätzung der Organisation ist.

Lehmann (1998) nennt verschiedene Ziele, die durch den expliziten Einsatz von Diskussionen erreicht werden können. Dazu gehört die Aktivierung des Denkpotentials, Überwindung von Expertenblindheit, Erlangung einer Vielfalt von Ideen und Hintergrundwissen bei komplexen Problemen, Entwicklung von Kooperation auch um persönlichen Interessenskonflikten entgegenzuwirken, Förderung des Leistungswettbewerbs, Unterbindung von Führungsarroganz und Mobilisierung der Teilnehmer und Teilnehmerinnen durch Mitverantwortung. Außerdem ruft eine breitere Beteiligung (Partizipation) an Entscheidungen eine höhere Akzeptanz hervor.

An vielen Punkten müssen sich die Mitarbeiter und Mitarbeiterinnen in Betrieben abstimmen. Delegieren und koordinieren, motivieren und organisieren, differenzierte Kontrolle ausüben, planen und entscheiden sind typische Kommunikationssituationen, die oft zwingend an den Austausch zweier oder mehrerer Personen gebunden sind (Lehmann, 1998).

Diskussionen können im auch organisationalen Rahmen als Lernmethode eingeordnet werden. Das Lernen geschieht allerdings oft implizit, ist ein „Nebenprodukt“ aus dem gewollten Informationsaustausch. Es besteht zwar ein Bedarf, das Wissen der Organisation zu vermehren, die Maße dafür sind aber schwierig zu definieren und messen typischerweise nicht das Wissen des einzelnen sondern extern repräsentiertes Wissen.

[4] Die Internalisierung beinhaltet vor allem das gegenseitige Verstehen, die Bildung neuer Sichtweisen sowie das Erlernen von Inhalten. Sie ist deshalb nicht unbedingt ein eigenständiger Arbeitsschritt.

Darin besteht ein Unterschied zum Einsatz von Diskussionen als explizite Lernmethode, z.B. in Schulen. Dort werden im Gegensatz meistens individuelle Lernerfolge beurteilt.

1.3 Diskussionen als Bestandteil betrieblichen Wissensmanagements

Insbesondere in inhaltsgenerierenden Bereichen wie Forschung, Design, Consulting, Software, Werbung und Medien nimmt die Notwendigkeit zu, mit schnell anwachsendem oder kurzlebigem Wissen und Informationen umgehen zu müssen. Ständig wird neues Wissen akquiriert (Informationen) oder gewonnen (Erfahrungen), das mit bestehendem Wissen zusammengeführt werden muss.

Dieses Wissen, das „Know How“ und „Know What“, liegt aber oft „nur“ als implizites Wissen (tacit knowledge) über die Kompetenz der Mitarbeiterinnen und Mitarbeiter vor, die es in die Arbeitsabläufe einbringen. In dieser Kompetenz, dem „tacit knowledge“, liegt ein ganz wesentlicher Wert eines Betriebes begründet („intangible asset“: Nonaka, 1994; Borghoff & Pareschi, 1998; [skyrme48]; [skyrme49]; [skyrme51]), der oft jedoch nur schwer einschätzbar ist und außerdem durch die Fluktuation der Mitarbeiter ein großes Risiko für das Unternehmen birgt. Diesen und angrenzenden Problembereichen widmet sich ein großer Bereich des Wissensmanagements, typischerweise unter den folgenden Zielstellungen:

- Entwicklung neuen Wissens
- Sicherung bestehenden Wissens
- Verteilung von Wissen
- Kombination bestehenden Wissens

Aus diesem Kontext des Wissensmanagements wird hier vor allem die Anforderung abgeleitet, dass sich Diskussionen und Diskussionsunterstützung nicht auf die singuläre Diskussionssituation beziehen sollten, sondern auch Konzepte für die Verwaltung, den Zugriff und die Integration der Diskussionsergebnisse mit anderen Inhalten und Medien liefern muss. Eingebettet in entsprechende technologische und softwarebasierte Arbeitsumgebungen können Diskussionen zu allen vier oben genannten Aspekten Lösungen beitragen.

Um den Bezug von Diskussionen zum Wissensmanagement deutlicher zu machen, werden im Folgenden verschiedene Konzepte dazu erläutert:

Kooperative Betriebsorganisation. Die Organisation und Struktur von Unternehmen hat sich seit ca. 20 Jahren maßgeblich verändert. Diese Veränderung ist im weitesten Sinne

durch den Begriff der Dezentralisierung zu beschreiben, die sich auf den Abbau von Hierarchien und der Segmentierung und Flexibilisierung von Betriebsbereichen bezieht (Dehnbostel, 1995). Eine häufig geäußerte These besteht auch darin, dass dieser Trend mit komplexer werdenden Fragestellungen einher geht (z.B. Schmuck, 1985). Charakteristisch sind eine auftragsbezogene Produktion, relativ hohe Autonomie unterer Hierarchieebenen und vor allem Gruppenarbeit (Binkelmann, 1993). Aus wirtschaftlichen Beweggründen wurden z.B. „profit centers", Ertragszentren oder Leistungseinheiten gebildet (Kruse, 1995). Für Organisationsformen kleinerer Betriebseinheiten stehen z.B. die Konzepte der flexiblen Fertigung, der Fertigungsinseln und Fertigungszellen, oder projektbezogene Arbeitsgruppen. Projekte hielten meistens dort Einzug, wo kundenorientiert entwickelt wird. Diese Ausrichtung auf Kunden, damit auch die verbesserte Dienstleistung und die Notwendigkeit zu experimentieren, wurde zu einem wichtigen Wettbewerbsfaktor (Pedler, Boydell & Burgoyne, 1996).

Tabelle 1 stellt verschiedene ökonomisch-organisatorische Trends vor, die parallel zur Dezentralisierung festzustellen sind. Ihnen zuordenbar, dargestellt auf der rechten Seite der Tabelle, sind verschiedene Formen des kooperativen Arbeitens, Lernens und Anforderungen an technische Unterstützung. Diese Gegenüberstellung soll verdeutlichen, dass der Bedarf an Kooperation, und damit verbunden an Diskussionen, durch diese Organisationsformen ausgelöst werden.

Aus dieser Vielfalt von Anforderungen kann man erkennen, dass die benötigten Lösungen in keiner Weise trivial sind, denn ganze Forschungsbereiche müssen dafür zu Rate gezogen werden und eine ganze Palette technischer Lösungen steht zur Verfügung. Aus dieser Fülle ergeben sich letztlich gravierende Schwierigkeiten, wenn konkrete Lösungen für das Wissensmanagement gefunden oder entwickelt werden sollen.

Lernende Organisation. Ein ganz wesentliches Konzept des Wissensmanagements ist das der „lernenden Organisation". Dieses zielt darauf ab, dass sich das explizite Wissen einer Organisation vermehrt. Natürlich haben auch Methoden für das organisationale Lernen die Individuen im Blick. Die Lernziele sind jedoch eher auf die Organisation verschoben. Auf Ebene der Gruppe und der Organisation besteht im Gegensatz zum Individuum eine noch größere Schwierigkeit, den Lernerfolg einzuschätzen. Gemäß vieler Definitionen liegt Lernen dann vor, wenn eine Verbesserung eines Verhaltens in vergleichbaren Situationen beobachtet werden kann (Simon, 1983; Michalski, 1986). Es wird davon ausgegangen, dass eine alleinige Akkumulation von Information nicht als Lernen bezeichnet werden kann, sondern relevant ist, wie mit diesen Informationen umgegangen wird.

Ökonomisch-organisatorischer Trend	Daraus entstehende Anforderungen
Dezentralisierung	Verteiltes, kooperatives Arbeiten
Spezialisierung	Kooperation zwischen Spezialistinnen und Spezialisten und zur Beratung, dafür Kooperationsunterstützung
Schnelle Produktionszyklen und Kundenorientierung	Aktivitäten: - Informationszugriffe - Informationsauswertung - Re-Engineering } Technologiebereiche: - Informationsverwaltung - Informationsaustausch - Informationsnetze
Flache Hierarchien Gruppenarbeit	Neue Organisationsmethoden (siehe Tabelle 2)
Lernende Organisationseinheiten statt systematische Schulung oder Selbstentwicklung (Pedler et al., 1996)	Eigenverantwortliches Lernen
Integration von individueller und organisatorischer Entwicklung (Sattelberger, 1996)	Integration des Lernens in den Arbeitsalltag (z.B. in Gesprächen, Diskussionen) Rückführung individuellen Wissens und Erfahrungen in die Organisation (z.B. in Gesprächen, Diskussionen)

Tabelle 1

Ökonomisch-organgisatorische Trends gegenübergestellt zu technisch/organisatorischen Anforderungen.

Als zentrales Mittel des organisationalen Lernens wird vor allem die Weitergabe von Wissen angesehen, die deshalb oft durch Management-Maßnahmen unterstützt wird. Dabei wird eben nicht mehr nur auf den Lernerfolg des Einzelnen fokussiert, sondern durch oft groß angelegte Dokumentationsmaßnahmen das explizit „sichtbare" Wissen vermehrt.

> *„ ... knowledge management can be explained as the management of the environment that makes knowledge flow through all the different phases of its life-cycle."*
> *(Borghoff & Pareschi, 1998, S. 7)*

> *„ ... ‚managing for knowledge' is about creating a thriving work and learning environment that fosters the continuous creating, aggregation, and use/reuse of both organizational and personal knowledge in the pursuit of new business value."*
> *(Borghoff & Pareschi, 1998, Vorwort)*

Beispiele kooperativer Lernansätze sind recht umfangreich in [KM-methoden] aufgeführt und werden in Ausschnitten in Tabelle 2 aufgegriffen.

Borghoff und Pareschi (1998) stellen Komponenten eines Wissensmanagements vor, die vor allem den Informationsfluss organisieren. Die Maßnahmen sind dort in drei Kategorien unterteilt:

- Stützen der Kooperation und Kommunikation (z.B. „shared" Workspaces, Unterstützung von Arbeitsabläufen, Monitoring der Abläufe, Sammeln von Erfahrungen)

- Dokumentieren (z.B. Repositories, Organisation von Zugriffen, Dokumentenmanagement)
- Vernetzung der Inhalte (z.B. „Wissenslandkarten“, Navigation, „Concept Maps“, Simulationen, Entscheidungsgrundlagen/ Begründungen)

Ziel	**Methoden**
persönliche Meisterung	z.B. Coaching, Einbindung von Kooperationspartnern, Feedback, Konfrontation mit neuen Aufgaben
mentale Modelle	z.B. Beziehungslandkarten, Erfahrungsgemeinschaften, informelle Gruppenbeziehungen, Interaktionsanalyse
gemeinsame Visionen	z.B. Ideal-Organisation, Intergruppen-Team-Entwicklung, Projekt-Prototyping
Lernen in der Gruppe	z.B. Erfahrungsgemeinschaften, Moderierte Teamsitzung, Stärken-/Schwächenkataloge
Denken in Systemen	z.B. Beziehungslandkarten, Ideal-Organisation, Interaktionsanalyse

Tabelle 2

Ziele organisationalen Lernens und Beispiele für Methoden [KM-methoden].

Selbstorganisation. Vor dem Hintergrund der Dezentralisierung mit stärker selbstorganisierten Arbeitsformen, ist ein verantwortlicher und qualifizierter Einfluss der Mitarbeiter und Mitarbeiterinnen auf die jeweilige Organisation nicht nur wünschenswert, sondern notwendiger Bestandteil. Das betrifft sowohl die Abstimmungs- als auch die Lernprozesse. Dehnbostel (1995) sieht bei den neuen Arbeitsstrukturen eine zunehmende Bedeutung von kontinuierlichem Lernen am Arbeitsplatz, das das Lernen und Arbeiten im Team sowie eine kontinuierliche und kooperative Selbstqualifizierung und Selbstorganisation voraussetzt. Diese Selbstqualifizierung ist an vielen Punkten vermutlich die einzige Möglichkeit eines „Knowledge Worker“ bzw. eines Teams, sich weiterzubilden, um mit dem Tempo der Entwicklungen standzuhalten.

Die Weiterbildung von Mitarbeiterinnen und Mitarbeitern gestaltet sich in wissensintensiven Branchen generell schwierig. Das liegt z.B. daran, dass die gewonnenen Erfahrungen und das Wissen, teilweise speziell auf die Organisation zugeschnitten ist, sich auf dem neuesten Stand befindet oder diesen sogar mit bestimmt. Das schließt oft eine geregelte Weiterbildung in Form externer Kurse aus. Einer Weiterbildung kann deshalb oft nur im Kontext von Forschungs- und Entwicklungsprojekten (R&D Projekte) nachgekommen werden. Darin enthalten Gruppengespräche hohe Lernpotentiale, da sich die Beteiligten gegenseitig abstimmen, vertreten, selbst organisieren, Entscheidungsfreiräume ausfüllen und eigenverantwortlich handeln müssen. Nach Ritter und

Zink (1994) dienen Gruppen damit u.a. der Weiterentwicklung der fachlichen und menschlichen Kompetenz und erhöhen die Selbständigkeit.
Die Gruppendiskussion wird zu einem Mittel, neue Informationen zu verbreiten, in den Arbeitsablauf aufzunehmen und eben in den bestehenden Pool von Informationen einzubinden. Sie ist oft die einzige Möglichkeit, neue Ideen zu fixieren und zu konzeptualisieren. Diskussionen sind gerade dann sinnvoll, wenn ein unterschiedliches Verständnis von einem Gesprächsgegenstand vorliegt. Um ein „shared understanding“ (Roschelle & Teasley, 1995) zu erreichen, dient die Kommunikation dazu, abweichende Interpretationen aufzudecken und Fehlkonzeptionen zu entschlüsseln.

1.4 Sind Diskussionen effektiv?

In Bezug auf den gezielten Einsatz von Diskussionen, werden vor allem zwei Kritikpunkte geäußert. Der eine bezweifelt den Sinn von Diskussionen wegen der vielfach gemachten Erfahrung, dass sie sehr unbefriedigend und ergebnislos verlaufen. Der andere Hauptkritikpunkt wird vor dem Hintergrund von Lernsituationen geäußert. Danach bestehen fundamentale Zweifel, dass kooperative Methoden, wie eben Diskussionen, zu besseren Lernerfolgen für die einzelnen Personen führen als individuelles Lernen. Während der erste Kritikpunkt im Rahmen methodischer Mängel, wie schlechter Moderation, Aufbereitung, Mitschrift etc., und der sozialen Konflikte einer Gruppe, wie beispielsweise Rollenverteilung, Erwartungen und Machtverhältnisse, untersucht werden muss, sind Erklärungen zum zweiten Kritikpunkt lerntheoretischer Natur.
Die lerntheoretisch begründete Frage nach dem Lernerfolg folgt vor allem aus unterschiedlichen Ansichten über Maße zum Lernerfolg, also was als Lernerfolg angesehen und akzeptiert wird und wie dieser gemessen werden kann. Auf der einen Seite des Spektrums wird gefordert, dass der individuelle Lernerfolg das Maß für die Güte einer Lernmethode ist. Das heißt, dass jeder einzelne im Vergleich zu vorher etwas gelernt haben sollte. Auf der anderen Seite des Spektrums wird vertreten, dass es allein schon wichtig ist, die Kooperation selbst zu lernen (Pea, 1993; Johnson & Johnson, 1985; Kagan, 1985). Hier wären also Leistungsbewertungen von ganzen Gruppen angebracht, die den Lernerfolg des einzelnen nicht unbedingt zur Voraussetzung haben, sondern beispielsweise effektive Kooperation bewerten.
Kritisch ist dabei das Maß selbst, also was die gelernten Inhalte sind, die zum Schluss festgestellt werden und die Wahl der Testmethode. Z.B. könnte am Ende allein Faktenwissen geprüft werden. Das würde kooperative Methoden vermutlich benachteiligen, da

dort beispielsweise auch Vorgehensweisen gelernt werden, die in dieses Maß nicht eingehen würden. Allerdings ist es sehr schwierig, methodisches Wissen oder Geschicklichkeit bei der Kooperation zu testen.

Zu den möglichen kognitiven Schwierigkeiten, die bei jeder Art des Lernens auftreten können, kommen bei der Kooperation, für die hier die Diskussion als Beispiel steht, alle erdenklichen Probleme hinzu, die aus der Zusammenarbeit von Menschen resultieren. Manche dieser sozialen Faktoren sind offensichtlich, andere eher verdeckt.

Eines der nicht unmittelbar erkennbaren Phänomene besteht im Umgang mit verteiltem und geteiltem Wissen. Unter verteiltem Wissen ist der Teil zu verstehen, der nicht allen Gesprächsbeteiligten bekannt ist, entsprechend ist das geteilte das gemeinsame Wissen. Eine Annahme bzw. Motivation bei der Gruppenarbeit besteht nun darin, dass das verteilte Wissen ausgetauscht wird und so „bessere“ Ergebnisse erzielt werden können als bei der Einzelarbeit, denn das Wissen der Gruppe wäre als Vereinungsmenge des Wissens der einzelnen Personen denkbar.

Weiterhin werden bessere Ergebnisse durch die größeren Ressourcen in Form von Arbeitskapazität, Informationen und Ideen erwartet. Die Kombination dieser Ideen kann durch Assoziationen ad hoc neue Ideen und Lösungsansätze evozieren, was in der individuellen Arbeit nicht gleichermaßen möglich ist. Diese kreativen Prozessen sind aber vor allem als Option einer Kooperation zu sehen, die in keiner Weise automatisch erreicht wird:

Stasser und Titus (1985) belegen eine Tendenz, dass hauptsächlich geteiltes Wissen in die Kommunikation eingebracht wird, also gerade der Teil, der sowieso schon allen bekannt ist, das ungeteilte aber kaum. Werden Personen Expertenrollen zugewiesen oder ist diese Rolle aus einer Arbeitssituation heraus klar, so steigt die Wahrscheinlichkeit, dass auch ungeteiltes Wissen mit in Diskussion eingebracht wird. Das liegt u.a. daran, dass die Personen selbst besser einschätzen können, welche Informationen anderen nicht bekannt sind. So wird es leichter, relevante Anschlusspunkte aus den Beiträgen anderer herauszugreifen, darauf zu fokussieren und sich an die entsprechenden Informationen zu erinnern (Stasser, Stewart & Wittenbaum, 1995).

Trotzdem wird die Qualität von Entscheidungen besser, wenn viel ungeteiltes Wissen eingebracht wird. Signifikant wirkt sich das dann aus, wenn hauptsächlich diese neue Information durch einen Moderator oder eine Moderatorin hervorgehoben wird (Larson, Christensen, Franz & Abbott, 1998). Eine Diskussionsunterstützung sollte deshalb darauf hinwirken, dass gerade dieses vorher nicht geteilte Wissen verstärkt in die Diskussion einfließt.

Die geschilderte Beobachtung könnte ein Resultat des Phänomens der kognitiven Konsonanz (Lehmann, 1998) sein. Dieses Phänomen besteht darin, dass auf bereits bekannte und akzeptierte Information fokussiert wird. Vielleicht geschieht das, um sich darin selbst zu bestätigen oder einfach den Weg des geringen Widerstandes einzuschlagen. Die kognitive Dissonanz beschreibt die Beobachtung, dass Informationen, die nicht ins eigene „Weltbild" passen vermindert aufgenommen oder schlicht „uminterpretiert" werden. Die Dissonanz kann soweit gehen, dass auch korrekte Beiträge als unglaubwürdig klassifiziert werden (Lehmann, 1998).

Dieser Umgang mit verteiltem und geteiltem Wissen kann unmittelbar dazu führen, dass, wie oben schon als Hauptkritikpunkt an Diskussionen genannt, der Sinn derselben angezweifelt wird, denn man „erfährt ja nie etwas Neues". Dass gerade die Verteiltheit von Information ein Problem darstellt entbehrt nicht einer gewissen Ironie, denn ein angestrebtes Ziel einer Diskussion ist gerade der Informationsaustausch. Wird arbeitsteilig gearbeitet, so fällt der Zusammenführung der Informationen eine große Bedeutung zu. Nur wenn die Beteiligten es schaffen, die erworbenen Informationen den anderen geeignet mitzuteilen, ergibt sich ein Mehrwert für die anderen durch einen unmittelbaren Informationszugewinn. Die dargebotenen Informationen müssen also geeignet elizitiert, gefiltert und aufbereitet werden.

In Gesprächen, in denen sich die Teilnehmerinnen und Teilnehmer nicht sehr gut kennen, kommt es außerdem zur gegenseitigen Einschätzung vorhandenen Wissens, besonders von Begriffswissen. Dieser Prozess wird als „Grounding" (Baker, Hansen, Joiner & Traum, 1999) bezeichnet. Sicherlich hinderlich ist, wenn die Diskussionsebene der Beteiligten zu unterschiedlich ist, wenn zu viel Energie in das „Grounding" investiert werden muss. Das „Grounding"-Problem ist beispielsweise für Experten-Laien-Gesprächen untersucht worden (Bromme, Nückles & Rambow, 1999). Zu Beginn eines Gespräches wird ein gewisses Vorwissen antizipiert. Dabei kann es sowohl zur Über- als auch zur Unterschätzung des Wissens des Gegenübers kommen. Wird das von einem oder beiden Beteiligten erkannt, so kann es in der Folge zu einer verschärften Fehleinschätzung in Bezug auf das Restwissen und damit auf das Erklärungsniveau kommen, was wiederum den Gesprächsverlauf empfindlich stören kann.

Ansonsten können auch organisationale oder wirtschaftliche Gründe die Zusammenarbeit in der Gruppe beeinflussen. Wenn z.B. Vertreter und Vertreterinnen unterschiedlicher Projekte zusammenarbeiten, kann sich eine Konkurrenzsituation ergeben, die an sich nicht für den Betrieb besteht, wohl aber im Sinne der Wirtschaftlichkeit einzelner Organisationseinheiten.

Als Fazit muss festgestellt werden, dass eine Gruppe zwar an manchen Stellen bessere Ergebnisse liefert als Individuen, aber trotzdem nicht so viel wie man erwarten würde. Positiv wirkt sich eine Gruppenkonstellation auf Entscheidungsfindungsprozesse aus. Insbesondere in Bereichen, in denen Entscheidungen auf sehr unsicherer Grundlage getroffen werden müssen, spielen Erfahrungen, herausgebildete Wertesysteme, Gefühle und Einstellungen (Lehmann, 1998) eine wichtige Rolle. In einer Gruppe kann diese stark subjektive Grundlage abgeschwächt werden, da mehrere Ansichten aufeinandertreffen und sich gegenseitig in Frage stellen oder sich untermauern können.
Im Sinne eines „shared understanding" sind Diskussionen dazu geeignet ein gemeinsames Verständnis von Konzepten, Herangehensweisen und Begrifflichkeiten zu entwickeln (Roschelle & Teasley, 1995). In der Gruppe besteht die sehr einfache und direkte Möglichkeit Fehlkonzepte aufzudecken. Allerdings sind Diskussionsergebnisse selten objektivierbar. Dieses Problem trifft besonders auf Diskussionen im Berufsalltag zu. Ergebnisse wie Prognosen, Entscheidungen, Einordnungen von Entwicklungen müssen sich erst im Laufe der Zeit bewähren. Sie gewinnen ihre Plausibilität (siehe Kapitel „Einleitung und Überblick") erst einmal nur aus der subjektiven Bewertung des Diskussionsverlaufes.
Der Wert von Diskussionen kann hier nicht abschließend beurteilt werden, doch Gully (1968) bringt verschiedene Aspekte dazu etwas ironisch auf den Punkt: Jede Gruppe ist sicher nur so gut wie ihre Mitglieder. Sind alle Personen motiviert, talentiert in der Sache und bei der Kommunikation und gibt es auch noch eine gute Gesprächsleitung, so wundert es nicht, wenn auch gute Ergebnisse erzielt werden. Trotzdem würden in diesem Fall auch die Individuen gute Entscheidungen hervorbringen.
Ich vertrete die eher pragmatische Ansicht, dass Diskussionen vor allem in Arbeitskontexten unausweichlich sind, um Inhalte zu klären, zu planen, sich abzustimmen, da sie schlicht und einfach nicht vermieden werden können. Deshalb sehe ich einen Vorteil darin, diese so zu gestalten, dass ihre Qualität und Nachhaltigkeit verbessert wird.

1.5 Technische und Softwarebasierte Diskussionsunterstützung

Im Folgenden wird versucht, einen Bedarf an Diskussionsunterstützung durch Software bzw. Computertechnologie aus den Aktivitäten und Bedürfnissen während einer Diskussion zu begründen. Das steht vielen eher technisch motivierten Ansätzen gegenüber, die den Unterstützungsbedarf hauptsächlich aus der räumlich verteilten Situation

ableiten. Daraus resultiert, dass die Gestaltungsmöglichkeiten der ausgetauschten Inhalte bisher wenig thematisiert wird. Offenbar wird der Einfluss unterschätzt, den ein Kommunikationsmedium auf die Kommunikation ausübt: Während der Kommunikation kann es diese gleichermaßen einschränken wie bereichern; im vielfachen Einsatz kann dessen Einfluss sogar soweit reichen, dass sich ganze Kulturphänomene herausbilden (z.B. öffentliches Telefonieren, Symbolsprache bei Chats, SMS).

1.5.1 Ansätze und Zielstellungen

Als Ausgangspunkt einer Diskussionsunterstützung wird nicht die Kommunikation allgemein betrachtet, sondern spezieller die Wissenskommunikation (siehe auch Kapitel 2.5). In vielen Situationen, wie Entscheidungen, Analysen, Produktentwicklungen, Designfindungen oder generell bei der gemeinsamen Klärung von Fragen, muss Wissen ausgetauscht werden, um zu einem Ziel zu gelangen. Durch die Vielfalt möglicher Inhalte werden dafür meist komfortable und individuelle Ausdrucksmöglichkeiten benötigt, denn die Inhalte können gerade dabei sehr kompliziert werden.

Aus dem Zusammenspiel der technischen Möglichkeiten, der kooperativen Arbeitssituation und der Interpretation der Wissenskommunikation als Lernen eröffnet sich ein ganzes Spektrum an Unterstützungsmöglichkeiten.

Visualisierung/Externalisierung. Ein Unterstützungsbedarf bei Diskussionen wird hier im Bereich der Externalisierungen von Wissenselementen und -strukturen gesehen, die ansonsten nur als individuelle Ansichten oder implizite Annahmen („tacit preunderstanding“, Stahl, 2000) vorhanden sind. Im Gespräch werden diese Ansichten und Erfahrungen geäußert und stehen nur so einer Reflexion in der Gruppe offen. Dies wiederum bildet einen wichtigen Schlüssel, um Wissenszyklen (Nonaka, 1994; Stahl, 2000) ins Leben zu rufen.

Ein wesentliches Hilfsmittel stellen dafür Visualisierungen dar, über die Repräsentationen der besprochenen Inhalte geschaffen werden. Mit den externen Darstellungen sind implizit bereits bestimmte Repräsentationsmöglichkeiten verbunden und unter Umständen schon Methoden assoziiert. Die externe Darstellung fokussiert immer auf bestimmte Aspekte, wie Zusammenhänge (z.B. Graphen) oder Vollständigkeit einer Darstellung (z.B. tabellarische Rasterungen und Fallunterscheidungen) (Suthers, 1999A & 1999B). Die Repräsentationen wie die Methoden, z.B. Wissenselizitation oder Kreativmethoden, haben einen Einfluss darauf, welches Wissen in die Gruppe hineingetragen wird. Dies wird verstärkt durch die Rückwirkungen, die die Darstellungen

auf die Arbeit in der Gruppe hat. Diese Wirkungen von Darstellungsformen können damit auch der primäre Grund zur Erstellung von Externalisierungen sein.

Sowohl die Repräsentation als auch die Methoden können durch kooperative Software geeignet unterstützt werden. Die softwaregestützte (Re-)Strukturierung ist sehr viel unproblematischer als mit herkömmlichen Mitteln wie Karteikarten (z.B. Metaplan) oder an einer klassischen Tafel. Es können größere Datenmengen bewältigt und direkt zugreifbar hinterlegt werden. Digitale Handschriftliche Notizen ermöglichen entsprechend zu einer Tafel oder einem Whiteboard völlig freie Eingaben. Die Interoperabilität zwischen Tools, z.B. über standardisierte Datenformate, ermöglicht einen komfortablen „ad hoc"-Einsatz unterschiedlicher Medien, usw.

Durch Software kann außerdem eine große Bandbreite von Methoden unterstützt werden. Bespiele sind Assoziationsmethoden, die durch Bildmaterial, Restrukturierung und verteilte Eingabe unterstützt werden ähnlich wie Kreativmethoden wie „Mind Mapping", Fallunterscheidungen z.B. mit Tabellen, Meta-Plan, etc.

Entlastung des Gedächtnisses („group memory"). Durch Externalisierungen soll das Gedächtnis entlastet werden (Lajoie, 1993). Die Darstellungen bilden ein externes Gruppengedächtnis, das während der Kooperation für alle sichtbar sein und dadurch passiv und aktiv zur Verfügung stehen sollte. Insbesondere können Hypothesen leichter vorgehalten und erst bei Bedarf überprüft werden (Lajoie, 1993).

Diskussionen gewinnen durch gegenseitige Bezüge, das Wiederaufnehmen von Aussagen und das Klären von Fragen an Qualität. Die Verständigung über den Diskussionsverlauf, Rückbezüge, Exkurse, Ansprechen bestimmter beteiligter Personen und Turntaking ist bei vielen Medien schwierig. Restrukturierungen der Inhalte sind ohne Wiederholungen oft nicht möglich. In Video- und Audio-Konferenzen können oft gar keine Materialien einbezogen werden. Selbst Newsgroups bieten eine sehr eingeschränkte Möglichkeit, Bezüge herzustellen.

Die Abhängigkeit der Beiträge sollte in dem verwendeten Medium repräsentierbar sein, denn gerade hier treten bei ausschließlich verbaler Kommunikation Probleme auf, da nach einer gewissen Zeit Beiträge vergessen werden oder sich in der Erinnerung verändern.

Strukturierung/Wissenskonstruktion. Tools, die eine Wissenskonstruktion über die Strukturierung von Inhalten oder allgemeiner Artefakten (z.B. Simulationen oder Designs) erlauben, werden als kognitive Tools (Jonassen, 1992; Lajoie, 1993) oder Mind Tools bezeichnet. Deren Mächtigkeit liegt dabei weniger in der Möglichkeit, vorhandene Fragen in quantitativer Hinsicht besser zu lösen („amplification"), als vielmehr darin, die

Fragestellung und Herangehensweise selbst zu verändern (Pea, 1985). „Cognitive tools also include the forms of reasoning and argumentation that are accepted as normative in given cultures“ (Resnick, 1991, S. 8).

Das Ziel beim Einsatz externer Repräsentationen besteht in der Strukturierung bzw. Modellierung von Inhalten. Die Strukturierung betont ein analytisches Vorgehen bei dem Gedanken hinterfragt oder Zusammenhänge verstanden werden sollen. Die hinzugewonnen Überzeugungen sollen hier als konstruiertes Wissen angesehen werden.

Modellierungen haben per se den Charakter einer Wissenskonstruktion. Nach Stachowiak (1973) bildet ein Modell Eigenschaften der „realen Welt“ ab. Es handelt sich um eine eingeschränkte Abbildung also eine Auswahl von Eigenschaften („Abbildungs- und Verkürzungsmerkmal“). Diese Auswahl ist zwingend, da eben nicht alle Eigenschaften abgebildet werden können, sie kann in Teilen, bewusst oder unbewusst durchgeführt werden. Jede Modellierung ist zweckgebunden und subjektiv („pragmatisches Merkmal“). Der Vorgang der Modellierung beschreibt den Weg, um zu einem Modell zu gelangen. Das Modell muss kein Artefakt sein, sondern es kann sich dabei im weitesten Sinn auch um individuelles Wissen handeln. In diesem Sinn spreche ich auch von Wissensmodellierung, wenn der Prozess gemeint ist, in dem neue Überzeugungen gewonnen werden.

Die Hilfsmittel der Wissensmodellierung können rein syntaktisch, passiv sein. In diesem Fall werden die Inhalte grafisch-interaktiv strukturiert. Die Modellierungsobjekte können allerdings auch aktive Objekte sein, die durch hinterlegte Semantiken eine umfangreiche Palette von Operationen anbieten oder ihre Reaktionen aus dem Modellierungskontext beziehen. So kann das entstandene Modell interaktiv mit Hilfe „intelligenter“ Objekte getestet und die Ergebnisse direkt wieder in die Kommunikation eingebracht werden („computational objects to think with“) (Pinkwart, Hoppe, Bollen & Fuhlrott, 2002). Je nach Spezialisierungsgrad und angestrebtem Produktziel können Objektsemantiken sehr speziell auf einen Kontext adaptiert sein oder eher allgemeine Hilfsmittel zur Verfügung stellen. Bei einer Diskussions- oder Sitzungsunterstützung kann sich die Wissenskonstruktion z.B. auf gemeinsame Begrifflichkeiten beziehen oder dazu dienen einen Inhaltskontext aufzubauen. Die Visualisierungen sind gleichzeitig zur Gewinnung von Meta-Wissen geeignet. So können Interessensfelder abgesteckt, Missverständnisse verdeutlicht oder Beteiligungen thematisiert werden.

Dokumentation/Retrieval. Ein weiterer Bedarf an Diskussionsunterstützung besteht in der Dokumentation. Traditionell beansprucht die Dokumentation von Sitzungsergebnissen einen großen Überarbeitungsaufwand. Die Ergebnisse müssen einigermaßen

zusammenhängend und verständlich sein und gehen deshalb u.U. über die besprochenen Details hinaus, fordern sogar zusätzliche Recherchearbeit.

Die heftigsten Kritikpunkte an einer technischen Gesprächsunterstützung ranken sich letztlich genau darum, dass der Aufwand bei der Erstellung der externen Repräsentationen und deren Nutzen nicht verhältnismäßig sind. Das bezieht sich sowohl auf die nachhaltige Nutzung als auch auf die Nutzung während der Diskussion. Diese Schwierigkeit bei der Abwägung von Kosten und Nutzen wird noch dadurch verstärkt, dass das Einarbeiten einer Gruppe zum versierten Umgang mit den technischen Unterstützungstools zeitaufwendig ist. So befindet man sich im unangenehmen Dilemma, dass man einerseits gerne eine Dokumentation der Diskussion hätte, andererseits die Arbeit dafür für zu aufwendig hält. Ein Ziel muss also sein, die Möglichkeiten der Visualisierung so attraktiv zu gestalten, dass sie einen ausreichenden Mehrwert für die Diskussion bildet, die Dokumentation aber kaum zusätzlichen Aufwands bedarf. Die These dieser Arbeit liegt gerade darin, dass Diskussionen als Szenario dazu geeignet sind, Dokumentationen zu erstellen, da sowieso oft Notizen angefertigt werden. Nur der Dokumentation davon wird bisher kaum Beachtung geschenkt.

Sollen die Dokumentationen hilfreich für spätere Gespräche sein oder sogar als Grundlage für spätere Dokumente dienen (z.B. „Deliverables“, Veröffentlichungen, Projektbeschreibungen, Ergebnisberichte), so müssen sie in Repositories indexiert oder zumindest durchsuchbar zur Verfügung stehen. Mit formalen, semi-fomalen oder schematisierten Darstellungen kann aus diesen Notizen u.U. zusätzlich Information extrahiert werden. Quasi standardisierte Formate wie XML ermöglichen einen Zugriff auf Daten und deren Filterung, was bedarfsspezifisch implementiert werden kann. Über diese Darstellungen ist es beispielsweise auch möglich, die Erstellung standardisierter Dokumente teilweise zu automatisieren.

Semi-formale visuelle Sprachen bieten viele Möglichkeiten, die bei der Modellierung entstandenen Diagramme, Notizen, Modelle und Skizzen sehr strukturiert in den Dokumentationsprozess einzubeziehen. Abstrakte Beschreibungen in den gespeicherten Daten ermöglichen zielgerichtetes Suchen nicht nur über konkrete Inhalte, die als „Klartext“ darin enthalten sind, sondern auch über Objekttypen und Attribute. Auch automatisch erstellte Vernetzungen zwischen Dokumenten oder Objekten sind möglich.

Feedback. Feedback-Funktionen können das „group well-being“ beeinflussen, wie es in der TIP-Theorie von McGrath (1991) erläutert wird. Gemäß der TIP-Theorie („time, interaction, performance“) muss es für eine Arbeitsgruppe möglich sein, sich fortzuentwickeln („member-support“). Dazu gehören beispielsweise der Aufbau loyalen

Verhaltens, die Kommunikation aller Gruppenmitglieder sowie die Entwicklung von „group well-being“, z.B. eine angemessene Interaktion und Rollenverteilung. Sowohl in einer computergestützten[5] als auch in einer computerangereicherten Wissenskommunikation können diese Aspekte durch Feedback unterstützt werden. Mögliche Ziele des Feedbacks sind danach die Verbesserung der Motivation, die Verbesserung der Performance, verbesserte Interaktion und Kommunikation.

Inhaltsbezogenes Feedback kann dialogorientiert oder beispielsweise auch akustisch gestaltet sein (Tewissen, Lingnau, Hoppe, Mannhaupt & Nischk, 2001). Das Kommunikationsmedium kann Feedback liefern, indem statistische Übersichten oder Interpretationen über die Interaktionen in der Gruppe angeboten werden. Statistiken über Beteiligung und Art der Beteiligung können eine Rückwirkung auf die Kooperation haben.

In einer Abwandlung kooperativer Szenarien müssen außerdem nicht alle Mitglieder der Lerngemeinschaft natürliche Personen sein. „Künstliche Mitlernende“ oder Akteure (Paiva, Machado & Prada, 2001) können den Lernprozess fördern, indem sie Erarbeitetes hinterfragen und Erklärungen einfordern usw. Sie können bestimmte Rollen und Aufgaben in einem Kommunikationsprozess übernehmen. Denkbar sind Moderationstätigkeiten, Strukturierungen, Methodenunterstützung und Visualisierungen. Hoffmann (2002) argumentiert, dass auch das Bewusstsein über Aktivitäten, die erst in der Zukunft stattfinden, die aktuellen Handlungen beeinflusst („Zukunftsawareness“).

In Bezug auf Feedback werden zwei Varianten unterschieden: „push“-Ansätze bieten Information direkt an, bei „pull“-Ansätzen muss darauf erst zugegriffen werden. Dourish und Bellotti (1992) sprechen dafür, dass in einem verteilten Szenario bei der Arbeit in „shared“ Workspaces” Information über die Arbeit der Beteiligten als Hintergrundinformation angeboten werden sollte („shared feedback“), also ohne Aufwand für den Zugriff („passively collected“). Die Begründung dafür besteht darin, dass das Bewusstsein („awareness“) darüber, was Beteiligte in der Kooperation gerade tun und warum, einem notwendigen Kontext für die eigenen Aktivitäten bildet.

Diskussionsanalyse. Es ist möglich, prozessorientierte und strukturorientierte Kriterien einer Diskussion zu definieren und diese als Analysegrundlage einzusetzen. Ein Ziel könnte darin bestehen, die Ergebnisse in Form von Feedback wieder in die Diskussion einfließen zu lassen, ein weiteres könnte in einer anschließenden Analyse bestehen.

[5] Unter einer computergestützten Wissenskommunikation sei an dieser Stelle eine Situation zu verstehen, in der die Beteiligten örtlich verteilt sind und allein durch die mediale Unterstützung miteinander kommunizieren können. Eine computerangereicherte Wissenskommunikation steht hier für eine „face-to-face“-Situation, in der Computermedien zusätzlich zur klassischen Kommunikation eingesetzt werden.

Beispielsweise könnten Diskussionsstränge als Graphen visualisiert werden oder die absolute oder relative Beteiligung in Diagrammen. Wünschenswert wäre die Visualisierung von Themen- und Interessensgruppen, offenen Fragen, Widersprüchen, etc. In Rückgriff auf das Kapitel 1.4, in dem das Problem aufgezeigt wurde, dass hauptsächlich geteiltes Wissen in eine Diskussion eingebracht wird, wäre es sinnvoll herauszufinden, inwieweit auf redundante Informationen fokussiert wird.

Ein ebenfalls wichtiger Aspekt ist der der Kooperation. In Mühlenbrock (2001) und Mühlenbrock und Hoppe (2001) wird ein Ansatz vorgestellt, der auf Basis allgemeiner Aktivitäten auf Objekten („delete", „create", „modify") Aktionsketten analysiert. Unter Hinzunahme der Information über den Urheber einer Aktion, können diese Ketten z.B. als kooperativ, konstruktiv oder destruktiv klassifiziert werden. Für diese Klassifikation werden im Vorfeld gültige Muster definiert. In Zumbach, Mühlenbrock, Jansen, Reimann und Hoppe (2002) werden solche Aktionsketten auch für Dialogsituationen analysiert.

Filter/Sortierung. Es sind zwei Extreme einer Diskussion denkbar: Einerseits kann es passieren, dass ungenügend und nicht erschöpfend diskutiert wird, andererseits können eine Unzahl von wichtigen sowie unwichtigen Einzelinformation eingebracht werden.

Durch automatische Sortierungen oder Layouts kann eine bessere Übersicht erreicht werden. Diese Möglichkeit wird von den meisten „Mind Mapping"-Produkten zur Verfügung gestellt (vgl. hierzu die Systemvorstellungen in Kapitel 3.2), die entstehende Baumstrukturen geeignet platzieren. Im vorliegenden Ansatz werden Strukturen mit dazugehörigen Layouts definiert. Diese dienen der Übersichtlichkeit und zeigen dabei auf den ersten Blick, wie ausgewogen beispielsweise Argumente aufgebaut sind. Auch Gruppierungen sind Sortierungen, die auf bestimmte Eigenschaften von Objekten, Beiträgen, etc. zugreifen, wie Farbe oder Autorschaft.

Durch Filter können größere oder unzusammenhängende Menge von Diskussionsbeiträgen in Teilmengen aufgesplittert werden oder Bezüge zwischen Beiträgen ausgeblendet werden. Wie durch Sortierungen kann auch so eine bessere Übersicht erreicht werden. Zusätzlich dienen die Kriterien auf denen die Filter agieren dazu, auf bestimmte Aspekte zu fokussieren. Beispiel dafür sind Perspektiven (dieser Aspekt wird in Kapitel 6.1 aufgegriffen), die z.B. für Meinungen stehen, also auf Basis der Autorschaft filtern (Stahl, 2001). Im vorliegenden Ansatz werden Objekttypen und Strukturdefinitionen als Grundlage für Filter ausgenutzt.

Wissensbasierte Unterstützung. Im klassischen Vorgehen wissensbasierter Systeme kann die Intelligenz eines Tools über die Modellierung einer Wissensdomäne erreicht werden, also durch streng formalisiertes Wissen. Das modellierte Wissen kann bei der

Diskussionsunterstützung allerdings kaum auf die Diskussionsinhalte bezogen werden, da beliebige Inhalte diskutiert werden sollen. Möglich ist allerdings, Meta-Wissen über den Verlauf von Diskussionen zu definieren. Ein Beispiel ist Prozesswissen über die Art der Dokumentation von Diskussionen, so dass ein Workflow bestimmt werden kann.

Moderation. Moderation wird in erster Linie durch Personen ausgeübt, kann aber durch Funktionen des Kommunikationstools unterstützt werden. Moderationsfunktionen können die Vergabe von Turns kontrollieren, um auf die Beteiligung von Personen Einfluss auszuüben. In verteilten Szenarien können auch verschiedene Verfahren der Integration von vorbereiteten Materialien möglich sein. Bei Entscheidungen können „Voting"-Funktionen wichtig werden.

Die Moderation muss generell den inhaltlichen Verlauf der Kommunikation beeinflussen können, z.B. durch Vorgabe von Themen, Zusammenfassungen und das Aufdecken von offenen Diskussionspunkten. Bei Lernszenarien besteht für Lehrerinnen und Lehrer die Notwendigkeit, den Ablauf der Lehrsituation zu beeinflussen. Für sie muss kontrollierbar sein, wer welche Ergebnisse abliefern soll und zu welchem Zeitpunkt dies geschehen soll. Auch die Zusammenstellung von Arbeitsgruppen kann technisch unterstützt werden.

In der durch externe Repräsentationen angereicherten „face-to-face" Situation wird die Kommunikation durch die zusätzliche Verwendung von Tools komplexer. In einer Vorstudie, die für diese Arbeit durchgeführt wurde (Kapitel 4), war deutlich zu erkennen, dass der Diskussionsfluss dadurch leicht gestört werden konnte. Durch eine Moderatorin konnte darauf allerdings ein hilfreicher Einfluss ausgeübt werden. Einer ihrer maßgeblichen Einflussnahmen bestand in der Koordination von Sprech- und Schreibphasen. Es zeigte sich also indirekt, dass auch für eine solche Situation erst ein adäquates Kommunikationsverhalten gelernt werden muss. Die Moderation durch eine Person wird im Verlauf dieser Arbeit allerdings nicht weiter thematisiert.

Kooperationsphasen. Kooperative Arbeitssituationen sind offenbar unmittelbar an die drei Phasen Sammeln, Reflektieren und Zusammenfassen gebunden. Lehmann (1998) gliedert Diskussionen in drei Phasen: Eröffnung, Arbeit am Problem, Abschluss. Die Phasen nach Gully (1968) können darauf abgebildet werden: Problem und Definition fällt in die Eröffnung, Analyse, Vorschlag von Alternativen in die Arbeit am Problem und Gewichtung der Alternativen und die Entscheidung beschließen die Diskussion. Ähnliche Phasen werden auch von Carlson (1995) im Zusammenhang mit dem Broca-System (siehe Kapitel 3.2) konstatiert. Bei der Erstellung von wissenschaftlichen Texten sollen dort erst einmal Ideen gesammelt und dann elaboriert werden. Erst danach werden die

einzelnen Aspekte gegliedert. Als Resultat entsteht dort erst einmal eine sogenannte „belief structure“.

Olson et al. (1996) fundieren solche Phasen in einer umfangreichen Studie, die auf dem Einsatz der IBIS-Methode beruht. Sie stellen dort fest, dass Meeting-übergreifend, bei unterschiedlichen Design-Fragestellungen, sich nur etwa 40% des Gesprächs auf die eigentliche Design-Fragestellung beziehen. 30% des Gesprächs werden durch reflektive Aktivitäten wie „Walkthroughs“ und Zusammenfassungen ausgefüllt. 20% sind reine Koordinationsaktivitäten. In einer Querklassifikation zeigt sich, dass insgesamt ein Drittel der Kommunikation aus klärenden Gesprächen besteht. Diese Klärung kann sowohl inhaltlich sein, wie koordinativ. Ebenfalls deutlich gestützt wird in Olsen et al. die Annahme, dass zwischen diesen rein inhaltlichen und den reflektiven Phasen sehr oft und unwillkürlich hin- und hergewechselt wird.

In der kooperativen Arbeitssituation manifestieren sich die Arbeitsphasen auch in den externen Darstellungen. Sowohl die Darstellung als auch die Arbeitsphasen können als divergent oder konvergent charakterisiert werden. Damit ist die Auseinanderentwicklung von Ideen, Ansichten, Möglichkeiten, etc. zu verstehen oder einfach deren deutliche Zunahme (Divergenz) bzw. die Integration, Zusammenfassung und Wartung von Inhalten (Konvergenz). Auf entsprechende Phasen sind wir auch im Vortest zur Systementwicklung (Kapitel 4) der Diskussionsunterstützung gestoßen (Gaßner & Hoppe, 2000). Wir konnten dort das Sammeln, Reflektieren und Zusammenfassen als natürliches Vorgehen der Gruppe beobachten. Diese Phasen werden immer wieder durchlaufen, ohne dass sie explizit angeleitet wurden.

In die Reflexionsphase fallen prinzipiell Fragen nach dem warum und wie, die in Hatano und Inagaki (1991) als Voraussetzung für ein Verständnis („comprehension“) eines Problems angesehen werden. Um zu diesem Verständnis zu gelangen, müssen Inferenzen gezogen, Plausibilitäten überprüft sowie neue mit alter Information in Übereinstimmung gebracht werden.

Weiterhin sind bei der Kooperation die gemeinsamen Sammel- und Reflexionsphasen notwendig, um in der Gruppe direkt, ohne Vorbereitung, ein Endresultat zu notieren.

1.5.2 Technische Unterstützung der Kooperation

Im Kontext von „face-to-face“-Diskussionen werden die möglichen technischen Unterstützungen nicht geringer als in der verteilten Situation. Das liegt vor allem daran, dass dieses Szenario immer fließend in das örtlich verteilte sowie in das asynchrone Szenario übergehen kann: Auch in der direkten Diskussion soll unter Umständen Kontakt zu einem

externen Experten aufgenommen werden. Dies ist zumindest für Lehrsituationen gut planbar. Die Asynchronizität wird alleine schon dadurch hergestellt, dass auf vorangegangene Ergebnisse zugegriffen werden soll.

Für eine technische Unterstützung einer Diskussion werden hier drei Bereiche charakterisiert:

- *Software/Tools:* Unterstützung der konkreten Diskussion beispielsweise durch Visualisierungen, digitale Handschrift, Methoden, methodische Unterstützung der Gruppendynamik und –kooperation etc.
- *Realisierung des Szenarios*: Unterstützung der technischen Kooperation durch gemeinsame Arbeitsbereiche („shared" Workspaces), Realisierung von Konzepten zum „ubiquitous computing", zu „augmented reality" und zu „Roomware", Funk-LANs, etc. (Wird im Folgenden noch kurz erläutert.)
- *Organisatorische Einbettung:* Technische Unterstützung bei der Dokumentation, beim Retrieval, beim Aufbau eines „corporate memory", etc. im Sinne eines langfristig angelegten Wissensmanagements.

Der erste Aspekt ist zentrales Thema dieser Arbeit und wird deshalb hier nicht noch einmal gesondert aufgegriffen. Dem dritten Aspekt – Wissensmanagement – hat sich bereits das Kapitel 1.3 gewidmet, allerdings ohne auf die möglichen Technologien einzugehen, die in diesem Bereich ausgesprochen vielfältig sind. In den folgenden Abschnitten werden einige für die Diskussion interessante technische Ansätze zum zweiten Punkt – Realisierung des Szenarios – kurz erläutert.

Roomware. Der in Streitz, Tandler, Müller-Tomfelde und Konomi (2001) erläuterte Begriff der „Roomware" fasst diverse Ideen zusammen, wie die „face-to-face"-Kooperation mit technisch-organisatorischen Hilfsmitteln ausgestattet werden kann. Elektronische Tafeln sind zur Dokumentation der Ergebnisse genauso geeignet, wie zur flexiblen Arbeit in der Gruppe auf Basis von Tools. Vorgestellt werden dort interaktive Tische und großflächige Projektionen, so dass ganze „Tafelwände" entstehen, auf denen interaktiv gearbeitet werden kann und die teilweise auch Gesten als Bearbeitungskommandos interpretieren können. Objekte können mit „Schwung" in eine andere Ecke geschubst werden. Die Raumempfindung wird so ein wichtiges Strukturierungskriterium. Ein weiterer wichtiger Schritt könnte die Verwendung mehrer Cursor an einem Board sein, so dass quasi jede Person ihren eigenen Stift besitzt.

Weiterhin werden individuelle „Passenger"-Objekte (z.B. Schlüssel, Brille) vorgestellt, an die Informationen gebunden werden, über die z.B. Dateirechte und Rechnerzugänge organisiert werden können.

Hinter dem Konzept der „Roomware“ steht die Überzeugung, dass die kooperative Arbeit nicht nur durch konkrete Software unterstützt werden sollte, sondern dass außerdem ein kreativer und unkomplizierter Umgang mit der Technologie notwendig ist. Es werden unterschiedliche Geräte mit spezialisierten Funktionen eingesetzt („ubiquitous computing“) und reale Objekte mit digitaler Information überlagert („augmented reality“). In diesem Konzept sind Funkschnittstellen zur Datenübertragung im lokalen Netzwerk zwischen den Geräten unbedingt sinnvoll.

Kopplung, Replikation. Auch im „face-to-face“-Szenario kann es Situationen geben, in denen es sinnvoll ist, synchron in gemeinsamen Arbeitsbereichen („shared“ Workspaces) zu arbeiten. Allerdings verschiebt sich die Bedeutung des „shared“ Workspace von einem notwendigen Mittel der Kommunikation in einer verteilten Kooperation hin zu einem methodisch motivierten Medium: Im Lernbereich können so z.B. Kinder jeweils mit ihrem eigenen Stift am gleichen Bild arbeiten und sich dabei mündlich absprechen oder wie in Tewissen, Lingnau und Hoppe (2000) gemeinsam Worte schreiben und sich vorlesen lassen, was u.a. dazu führt, dass die Kinder kooperativ ihre Fehler beheben. In einer Sitzung können über „shared“ Workspaces beispielsweise vorbereitete Teile integriert werden. Notizsituationen können komfortabler gestaltet werden, indem vom Platz aus Eingaben angefertigt werden, die dann für alle anderen sichtbar z.B. an der Tafel erscheinen. Anonyme Eingaben sind einfach umzusetzen und können so methodisch in eine Diskussion aufgenommen werden.

Die Replikation von Daten zwischen laufenden Applikationen kann als Spezialfall einer Kopplung angesehen werden. In Coulouris, Dollimore und Kindberg (2001) wird Replikation als Technik zur Verbesserung von Services eingeführt. Dazu werden drei Ziele differenziert: 1) Verbesserung der Performanz 2) Erhöhte Verfügbarkeit 3) Fehlertoleranz. Beispiele zur Performanzverbesserung ist das Datencaching von Web-Clients oder die Duplizierung von Daten auf mehrere abhängige Server, die sich die Anfragen aufteilen können. Durch die Verteilung von Daten auf unabhängige Server sind die Services besser Verfügbar, da sie unabhängiger von Netz- und Serverproblemen sind. Fehlertoleranz wird auf die Korrektheit von Daten bezogen. Durch die Replikation von Daten können unterschiedliche Zustände abgeglichen werden.

Eine Replikation ist dann möglich, wenn die Daten zugänglich sind. Beispielsweise können Dateien einfach kopiert werden, solange Zugriff auf das Filesystem besteht. Werden laufende Applikationen repliziert, so muss deren Zustand bzw. der Zustand einzelner Objekte oder Datenstrukturen übermittelt werden. In Java beispielsweise können manche Objekte „serialisiert“ werden. Dadurch besteht für diese die einfache

Möglichkeit, sie in einer anderen Applikation identisch wieder herzustellen. Ist dies jedoch nicht möglich, so müssen andere Mittel zur Kopplung oder Replikation implementiert werden.

- *Master/Slave-Konstellation.* Manche Ansätze haben zum Ziel, beliebige Applikationen zu koppeln, um sie verteilt und synchron verwenden zu können. Das Problem für die Kopplung liegt dabei in der Beliebigkeit, denn dadurch können keine Voraussetzungen über die Art der Applikation, deren Objekte oder Datenstrukturen getroffen werden.

  Das Problem kann dadurch gelöst werden, dass dort wo die konkrete Applikation installiert ist, diese als „Master“ eingesetzt wird. An Kooperierende „Slaves“ wird nur ein „Bild“ der Oberfläche übermittelt. Beispielsweise Maus-Events werden von dort zurückübertragen und über ihre Koordinaten den Funktionen des „Master“ zugeordnet und dort ausgeführt.

  Solche Ansätze gehen jedoch mit dem Verlust der Daten für die „Slave“-Seite nach Beendigung einer Sitzung einher. Darin besteht auch der Grund, warum in Lerner-Tutor-Szenarien die „Master“-Applikation bei den Lernenden liegen sollte, denn denen sollen danach eher die Daten zur Verfügung stehen als dem Tutor bzw. der Tutorin. Das zieht natürlich ein organisatorisches Problem bzw. ein Kostenproblem nach sich, denn es ist einfach und günstiger eine Applikation ein Mal bei der Lehrperson zu installieren, als bei allen Lernenden.
- *Replikation über zentrale Modelle: MatchMaker TNG.* MatchMaker TNG *(*Jansen, 2002) wird unter anderem für die Kopplung der FreeStyler-Applikation verwendet (Kapitel 7.1.5). Er ist in Java implementiert und nutzt RMI zur Realisierung der Kopplung. Die koppelbaren Applikationen bilden zusammen mit dem MatchMaker TNG eine Client/Server-Architektur. Der Server verwaltet für die koppelbaren Objekte der Applikationen ein Modell, deren Bestandteile bei der Applikationsentwicklung frei definierbar sind. Diese Objektmodelle sind in einem Baum strukturiert, so dass auch gezielt auf Äste dieses Baumes zugegriffen werden kann. Ein Ast entspricht dem Modell eines Objektes, z.B. einem Fenster, mit seinen untergeordneten Objekten, z.B. alle Objekte, die in einem Fenster liegen. Synchronisiert sich eine Applikation mit weiteren, so werden initial die Modelle des gewünschten Bereiches übernommen und daraus neue Objekte generiert, die identisch zu den gekoppelten Objekten der anderen Applikationen sind. Später registrieren die gekoppelten Objekte nur noch, wenn sich das Modell verändert und passen sich an. Bei dieser Client/Server-Architektur handelt es sich um eine

replizierte Architektur, in der die Clients eigenständige Applikationen sind. Wird eine Kopplung beendet, so stehen die Daten, die währenddessen erstellt worden sind, auch weiterhin allen Beteiligten der Kopplung zur Verfügung.

- *Services: Sun JavaSpaces.* JavaSpaces ([JavaSpaces]; Jansen, 2002) verwaltet Aufgaben und Daten gekapselt in Objekten in einem „virtuellen Raum". Auf diese Einträge wird von Applikationen über eine Rasterdefinition zugegriffen, die nicht unbedingt nur auf ein konkretes Objekt zutrifft. Auf die Objekte sind sowohl parallele (nicht blockend) als auch sequenzialisierte (blockend) Zugriffe möglich. Neben dem Zugriff auf Initiative der Clients werden auch Informationen über neue Objekte an die Applikationen verteilt. Das Konzept scheint weniger dafür gedacht zu sein, konkrete Objekte zu synchronisieren, als dazu, Services zu verteilen. Ein Servicekonzept der Form, dass Anfragen in den „virtuellen Raum" gestellt werden und dann auf die erste Antwort oder eine Antwort eines bestimmten Typs gewartet wird, liegt diesem Konzept näher.
- *Nachrichtenkanäle: Java Shared Data Toolkit (JSDT).* JSDT ([JSDT]; Jansen, 2002) stellt mit der Session Factory eine zentrale Komponente zur Verfügung, die den technischen Transfer von Nachrichten organisiert. Zur Kommunikation werden „Channels" erstellt, die für andere Clients geöffnet werden können. Gesendet werden Nachrichtenobjekte, die eingebunden in Events versendet werden. Jeweils kann gewählt werden, ob ein Kanal die korrekte Reihenfolge der Nachrichten zusichert oder nicht. Die Kommunikation ist generell nachrichtenorientiert. Es gibt in dem Sinne keine zentrale Instanz, die den Zustand der gekoppelten Applikationen vorhält.
- *Zentrales „event handling": Habanero.* Habanero, ebenfalls eine Java-Implementierung, realisiert die Kopplung über die Verteilung von Events ([habanero]; Jansen, 2002). Damit ist das System vergleichbar mit dem Ansatz des MatchMaker C++ (Kapitel 3.1.1), der für die Kopplung des DiscBoards verwendet wird. Events werden verteilt, wenn sich an den Objekten einer Applikation etwas ändert. Diese Kommunikation beinhaltet einen geringen Aufwand beim Datentransfer. Ohne begleitende Konzepte ist es dabei aber schwierig, einen aktuellen Stand von Applikationen zu erhalten. Habanero löst das, indem eine Applikation ihren aktuellen Stand versendet, wenn eine neue Applikation hinzukommt. Ein entsprechendes Vorgehen wurde auch für das DiscBoard gewählt, allerdings nicht in der allgemeinen Form implementiert wie bei Habanero.

Im Folgenden wird unter Bezug auf Jansen (2002) versucht, einige technische Kriterien der Kopplungskonzepte in Relation zur kooperativen Situation zu erläutern.

- *Partielle Kopplung.* Arbeitsbereiche partiell zu koppeln bedeutet, dass nur bestimmte Bereiche oder Objekte synchronisiert werden. Das ist z.B. der Fall, wenn eine Applikation selbst mehrere Fenster anbietet aber nicht alle synchronisiert sind. Diese Teilweise Kopplung beschränkt sich jedoch nicht auf Fenster, sondern kann sich auch auf Objekte beziehen. Beim Lernen können partielle Kopplungen beispielsweise für „jigsaw"-Settings ausgenutzt werden, in denen den Beteiligten unterschiedliche Ausgangsinformationen zugrunde liegen, diese aber in eine gemeinsame Lösung einfließen sollen.
- Eine partielle Kopplung ist sowohl mit dem MatchMaker TNG, durch Zugriff auf Äste des Synchronisationsbaumes, als auch mit Habanero zu realisieren, wo Clientseitig entschieden wird, welche Events versendet werden. Auch mit JavaSpaces kann eine partielle Kopplung durch die Rasterdefinitionen erreicht werden. Mit JSDT ist dies schwieriger. Dafür müssten die Clients eine Verwaltungsstruktur implementieren, die die Eventinteraktion auch für komplexe, nicht serialisierbare Objekte erlaubt, die so nicht in die Channels geschrieben werden können.
- *Temporäre Kopplung.* In den meisten Fällen ist die Kopplungsphase zeitlich begrenzt. Danach wollen die Beteiligten potentiell individuell weiterarbeiten. Das Hinzukommen und Verlassen einer Sitzung zieht auf der technischen Ebene das Problem der „late comer" nach sich. Beteiligt sich ein neues Mitglied an einer Kopplung, so muss dieses den aktuellen Stand übermittelt bekommen. Typischerweise ist das kritisch, da sich während der Übermittlungszeit bereits etwas an den Daten ändern kann, so dass es zu inkonsistenten Zuständen kommt. MatchMaker TNG kann damit relativ elegant umgehen. Ein neues Mitglied ruft einfach den aktuellen Stand über den Synchronisationsbaum ab. Ändert sich währenddessen etwas an den dort vorgehaltenen Modellen, wird wie bei jeder Änderung ein Event versendet und nach der Initialisierung wird dieser dann zusätzlich bearbeitet.

  In einem reinen Event-Ansatz gestalten sich Initialisierungsprozesse schwieriger, denn ein Zustand ist nur aus der verketteten Ausführung aller vorangegangenen Events herzustellen. Eventbasierte Synchronisationskonzepte müssen deshalb für das „late comer"-Problem einen gesonderten Initialisierungsprozess zur Verfügung stellen.

2 VERWENDUNG VISUELLER SPRACHEN

Der Begriff der visuellen Sprache ist bereits vielfach und in unterschiedlichen Bedeutungen verwendet worden (z.B. Lakin, 1990; Kremer, 1998; Myers, 1990; Wang, Lee & Zeevat, 1995). Auch gibt es Ansätze, die im Prinzip visuelle Sprachen verwenden, diese aber nicht als solche klassifizieren (z.B. Streitz, Haake et al., 1998; Suthers, Weiner et al., 1995; Buckingham Shum & Hammond, 1994; Jacobson, Booch & Rumbaugh, 1999; Jungbluth, 1998).

Im Verlauf dieser Arbeit konnten fünf Ziele voneinander abgegrenzt werden, für die visuelle Sprachen eingesetzt werden:

- Visuelles Programmieren und Visualisierung von Programmen
- Formalisierung visueller Sprachen und Schließen in visuellen Repräsentationen
- Lernunterstützung
- Wissenselizitation
- Kommunikationsmedien und Wissensprodukte

Visuelle Sprachen werden unabhängig vom Einsatzbereich meistens benutzt, um Strukturen und damit Zusammenhänge graphisch zu verdeutlichen, um Sachverhalte neu zu begreifen. Es werden Inhaltsbereiche erarbeitet und zwar jeweils mit einem ungewissen, konstruierten Ergebnis. Eine Ausnahme davon stellt nur das Schließen in visuellen Repräsentationen dar, wobei auch dort gerade strukturelle Eigenschaften von Objekten und deren Verhältnis zueinander ausgenutzt werden. Das im Vorfeld offene Ergebnis einer Wissenskonstruktion macht den Bedarf deutlich, auch den Konstruktionsprozess mit zu unterstützen.

Systeme, die visuelle Sprachen einsetzen, lassen sich allerdings nur selten genau einer dieser Zielrichtungen zuordnen, wenn auch meistens eine davon den jeweiligen Ansatz bestimmt. Im Verlauf dieses Kapitels werden die genannten Verwendungsbereiche kurz erläutert. Besonderer Raum wird dem Einsatz von visuellen Sprachen als Kommunikationsmedien eingeräumt, da dies auch den Ansatz für die Diskussionsunterstützung vorbereitet.

2.1 Visuelles Programmieren und Visualisierung von Programmen

Ein großer Teil der visuellen Sprachen wurde für die Programmierung entwickelt, also zur Erstellung von Programmen. Sie werden deshalb auch als visuelle Programmiersprachen bezeichnet. Murray und McDaid (1993) unterteilen die Repräsentationsformalismen in diagrammatische (z.B. Flow Charts), ikonische und formularbasierte. Eine ihrer Feststellungen besteht darin, dass eine starke Korrelation zwischen dem Repräsentationsformalismus und dem repräsentierten Gegenstand besteht. Diagrammatische Darstellungen eignen sich danach besonders für die prozedurale Programmierung und ikonische Darstellungen (Symbole oder Kurzdarstellungen für grundlegende Elemente einer Programmiersprache) sollen vor allem Programmieranfängern den Einstieg erleichtern. Formulare eignen sich weniger für die Entwicklung von Programmen als dazu, Daten zu erheben.

Die Unterscheidung in Visualisierungen von Programmen und visuelle Programmiersprachen wird von Myers (1990) und Murray und McDaid (1993) geteilt, wenn auch letztere die visuellen Programmiersprachen weiter aufschlüsseln. Im Gegensatz zu visuellen Programmiersprachen beschränken sich Programmvisualisierungen auf die Darstellung des Programm unter bestimmten Aspekten. In Myers (1990) werden die Programmvisualisierungen nach zwei Dimensionen unterschieden: Dimension 1) Wird der Code, die Daten oder der Algorithmus visualisiert? Dimension 2) Sind die Darstellungen statisch oder dynamisch? Ein Beispiel einer dynamischen Entwicklung von Daten sind z.B. Mandelbrot-Darstellungen.

Myers (1990) stellt für beide Ansätze - „program visualization“ und „visual programming“ - Taxonomien auf. Der Begriff der visuellen Sprache („visual language“) wird von ihm übergeordnet verwendet: „‘Visual Languages‘ refer to all systems that uses graphics,...“ (Myers, 1990, S. 99).

Wollte man beide Ansätze voneinander abgrenzen, so müsste dies über die Veränderungsmöglichkeiten erfolgen: Durch Visualisierungen werden Programme und Daten gerade nicht verändert, da ansonsten ein Programmiervorgang vorläge.

Es ist jedoch zu erkennen, dass diese Trennung heute nicht mehr zwingend ist. Im Gegenteil wachsen beide Ansätze sogar zusammen. Ein Beispiel dafür ist die Modellierung mit UML-Tools (z.B. [rational]), zum einen wird darin die Konzeption von Software mit Hilfe verschiedener Diagrammtypen unterstützt, zusätzlich die Übertragung dieser Darstellungen in Code-Bestandteile. Soweit möglich beziehen sich die einzelnen

Diagrammtypen aufeinander über die verwendeten Objekte oder werden sogar auseinander generiert. Für Klassen- und Methoden wird automatisch ein „Programmrumpf“ erstellt. Bei der Modellierung entsteht so nicht nur die Software, sondern das entwickelte Modell dient gleichzeitig der Dokumentation und eben einer bestimmten Art der Visualisierung des Programmcodes.

2.2 Formalisierung visueller Sprachen und Schließen in visuellen Repräsentationen

Neben dem visuellen Programmieren gibt es einen weiteren Forschungsbereich, „visual language research“, der sich vor allem mit formalen Aspekten visueller Sprachen auseinandersetzt. Fragen nach formalen Modellen, der Semantik und dem Schließen mit visuellen Sprachen stehen darin im Zentrum. Motiviert wird dieser Ansatz hauptsächlich darüber, dass davon ausgegangen wird, dass Bilder, graphische Darstellungen oder Diagramme Bedeutungen vermitteln können und so Schlussweisen zulassen, die ohne diese Darstellungen schwer zugänglich wären. Logische Schlüsse, die aus den Visualisierungen „sichtbar“ werden, sollen auch vom Computer ausgeführt werden.
Unter dem Schließen auf visuellen Repräsentationen wird im Allgemeinen das Schließen von Prämissen, die als Diagramme oder Muster repräsentiert sind, auf Konklusionen verstanden, die selbst wieder Diagramme oder Muster sind. Ein in diesem Kontext häufig genanntes Beispiel (z.B. Wang, Lee & Zeevat, 1995; Stenning & Oberlander, 1995) sind Mengendarstellungen in Form von Kreisen, mit denen z.B. Syllogismen dargestellt werden können. Ein Syllogismus ist beispielsweise der Folgende: Keine M sind P; Alle M sind S; (es gibt M) ⇒ Nicht alle S sind P. Grafisch kann der Schluss durch zwei Diagramme dargestellt werden, die die Prämissen bilden, und einem, das die Konklusion darstellt (Abbildung 2).

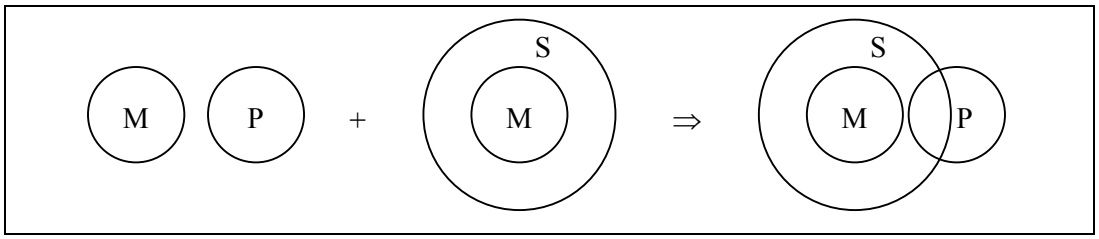


Abbildung 2

Grafische Repräsentation des Syllogismus Keine M sind P; Alle M sind S; (es gibt M) ⇒ Nicht alle S sind P.

Für das Forschungsfeld der visuellen Sprachen ergeben sich verschiedene naheliegende Fragestellungen, die von Wang und Lee (1993, S. 328) sehr intuitiv umrissen werden.

> 1. *How can pictures and their properties be formally described?*
> 2. *What information can one see in a picture?*
> 3. *What is a meaning interpretation?*
> 4. *How does an interpretation determine the meaning of a picture in its use?*
>
> *(Wang & Lee, 1993, S. 328)*

Ein generelles Problem beim Schließen auf graphischen Repräsentationen besteht darin, die relevante Information aus den Diagrammen herauszugreifen. In Wang, Lee und Zeevat (1995) werden dazu verschiedene Probleme angesprochen. Eines besteht darin, dass Diagramme gerade ein intuitives Herangehen ausnutzen. Das heißt, dass bestimmte strukturelle Eigenschaften des Diagramms für menschliche Betrachtende zwar sichtbar sind, durchaus aber nicht trivial und in keinem Fall allgemein gültig für eine Diagrammart definierbar sind. Als Beispiel wird dort die Euler'sche Mengendarstellung in Form von Kreisen aufgegriffen. Die Kreisdarstellung hat für die meisten Personen intuitiv die Bedeutung von Mengen. In dieser Interpretation wird der Schluss aus Abbildung 3, der besagt, dass die „entahlten"-Relation transitiv ist, unmittelbar als korrekt angenommen.

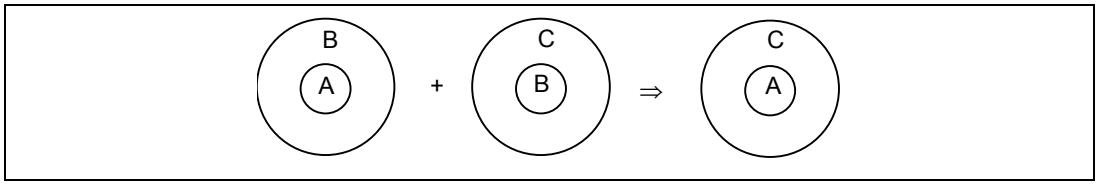


Abbildung 3

Transitivität der „enthalten"-Relation bei Mengen.

In einer logischen Formalisierung des Sachverhaltes wäre dies aber nicht der Fall. Würden die Kreise als beliebige Relation r verstanden werden, so wäre der entsprechende Schluss nicht zulässig.

$$r(A,B) \quad \wedge \quad r(B,C) \quad \not\Rightarrow \quad r(A,C)$$

Die Forderung der Transitivität müsste explizit hinzugefügt werden, könnte aber nicht aus den Prämissen abgeleitet werden. Eine visuelle Darstellung enthält also typischerweise implizite Informationen. Gerade diese Tatsache erklärt, warum visuelle Darstellungen oft hilfreich sind.

Wang, Lee und Zeevat (1995) kommen zu einem Ansatz für das Schließen in Diagrammen, für das fünf Forderungen zu erfüllen sind.

1. Die Syntax der visuellen Sprache muss ebenso definiert werden wie die Art der Informationen, die in einem Diagramm relevant sind.

2. Für die formalen Symbole der Diagramme muss eine Interpretation bzgl. der Interpretationsdomäne angegeben werden.
3. Die korrekte Verwendung der Sprache muss klar sein, auch ohne die Verwendung der Schlüsse (meaning).
4. Auf dieser Grundlage können Schlüsse angegeben werden.
5. Können mehrere Konklusionen aus den Prämissen abgeleitet werden, so muss spezifiziert werden, in welchem Verhältnis diese zueinander stehen.

Im Folgenden werden exemplarisch einige Ansätze vorgestellt, die sich jeweils mit einzelnen oder mehreren dieser Aspekte befassen und angrenzende Fragestellungen benannt.

- *Pixeldarstellung.* Ein Problem beim Umgang mit visuellen Sprachen besteht darin, die verwendeten Objekte in eine systeminterpretierbare Form zu überführen. Meistens ist das eine Aufgabe der Editoren, mit denen die Diagramme erstellt werden, wenn in diesen das Wissen darüber vorliegt, welche Elemente wie repräsentiert werden. Die visuelle Sprache kann durch die Editoren vordefiniert sein.
  Bottoni, Costabile, Levialdi und Mussio (1995) versuchen das Problem, bedeutungstragende Elemente aus einer Darstellung zu ermitteln, allgemeiner zu lösen. Sie formalisieren Bilder auf Grundlage von Pixelbereichen. Ein digitales Bild wird als Funktion definiert, die Integers, die für Reihen r und Spalten c stehen, in eine Pixeldarstellung abbildet.

  $$\text{digital image: „i: } \{1, ..., r\} \times \{1, ..., c\} \rightarrow P\text{“}$$

  Prinzipiell sind sie dadurch unabhängig vom Darstellungsmedium. Trotzdem ist aber eine Interpretation des Dargestellten normalerweise an eine Domäne und einen bestimmten Programmkontext gekoppelt. Unklar ist deshalb, ob diese allgemeine Herangehensweise wirklich für jede Art von Darstellungen einen Vorteil birgt.
  Sinnvoll kann sie dann sein, wenn die Darstellungen sich nicht aus geometrischen Elementen zusammensetzen, wie bei Fotos oder Röntgenaufnahmen. Dieser Ansatz der Interpretation von Pixelbereichen, also die Interpretation von Mustern, kann als Bilderkennung verstanden werden. Bottoni et al. (1995) sehen dafür in ihrem Ansatz die Möglichkeit vor, dass Nutzerinnen und Nutzer Teilstrukturen Attribute zuordnen („Description“), wie z.B. die Anzahl oder auch Ausprägung von Tumoren, die in einer Röntgenaufnahme sichtbar sind. Durch die Zuweisung einer Interpretation zu einem Bild, die der „Description“ entspricht (int(i)=d) wird eine Relation zwischen Bild und Interpretation hergestellt. Über eine Definition einer Materialisierungsfunktion (mat(d)=i), werden Bilder aus ihren Beschreibungen generiert.

- *Relationale Grammatik.* Grammatiken werden verwendet, um die Syntax von Sprachen zu spezifizieren. Jedoch sind Grammatiken nicht ausreichend dazu geeignet, räumliche Arrangements zu beschreiben, wie sie für visuelle Sprachen charakteristisch sind. Crimi et al. (1991) entwickeln deshalb den Ansatz relationaler Grammatiken.

  Eine relationale Grammatik G enthält neben den üblichen Komponenten wie den terminalen und den nicht-terminalen Symbolen (V_T und V_N), dem Startsymbol S und den Produktionsregeln P noch Symbole, die Relationen darstellen (V_R) und Evaluationsregeln R:

  $$(G=V_N, V_T, V_R, S, P, R).$$

  Die Evaluationsbedingungen werden jeweils den Produktionen hinzugefügt. Beispielsweise kann für ein Objekte o definiert werden, dass es neben einem schon existierenden Objekt liegen soll. Sei D ein Startsymbol für ein Diagramm, dann könnte eine Produktionsregel folgendermaßen lauten:

  $$D ::= \{o, D\}\ \{adjacent(o, D)\}$$

  Eine in Crimi et al. (1991) vorgeschlagene Evaluationsregel dazu wäre beispielsweise:

  $$adjacent(o, D) :- left(o, D)$$

  $$left(o_1, D_1) :- (D_1 \Rightarrow \{o_2\}) \vee (D_1 \Rightarrow \{o_2, D_2\}), left(o_1, o_2)$$

  Evaluationsregeln dieser Art sind als Domänenmodell formulierbar. Mit ihnen ist es sowohl möglich Strukturen zu definieren, z.B. Klassen von Graphen (siehe dazu Crimi et al. 1991), als auch räumliche Bedingungen.

  Als Nachteil dieses Ansatzes relationaler Grammatiken werden zwei Aspekte genannt. Zum einen können sich die Produktionsconstraints nur für Objekte beziehen, die in der Produktionsregel definiert sind. Der zweite Aspekt besteht im hohen Aufwand, der für das Parsen der Ausdrücke einer Sprache notwendig ist.

- *Graphgrammatiken.* Häufig sind Diagramme in Graphen überführbar oder sind unmittelbar Graphdarstellungen. Zur Überprüfung der syntaktischen Korrektheit der Graphen könnten Graphgrammatiken eingesetzt werden. Allerdings geben schon Editoren, mit denen Diagramme erstellt werden können, implizit eine Syntax vor, so dass eine Syntaxüberprüfung gar nicht notwendig ist. Ein Einsatz von Graph-grammatiken kann also vor allem dann nutzbringend sein, wenn nur Ausschnitte der erstellten Graphen erkannt werden sollen oder wenn die Graphdarstellung in eine andere Darstellung überführt werden soll, z.B. in eine abstraktere Beschreibung.

Rekers (1994) stellt dazu einen Ansatz vor (Abbildung 4), in dem die im Editor erstellten Graphen in zwei Stufen abstrahiert werden. Die erste Ebene bilden die graphischen Objekte, die über ihre Erscheinung, wie Farbe, Größe, Position, beschrieben werden. In der zweiten Abstraktionsebene wird ein „spatial relations graph“ erstellt. Dieser vernachlässigt das tatsächliche Erscheinungsbild der Objekte und beschreibt dafür, in welcher räumlichen Relation die Objekte stehen. Beispiele sind „contains“, „touches“, „connects“ und „labels“. In der dritten Abstraktionsebene können dann bestimmte Typen von Graphen definiert werden.

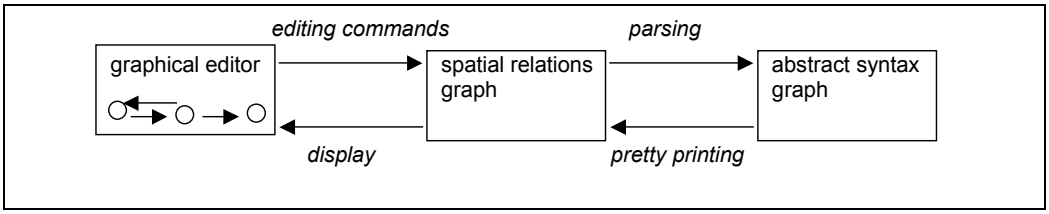


Abbildung 4

Drei Abstraktionsebenen einer Graphdarstellung. Abbildung in Anlehnung an Rekers (1994).

Rekers (1994) verwendet die Graphgrammatik an dem mit „parsing“ bezeichneten Zeitpunkt. Unmittelbar damit verbunden ist die Aufschlüsselung der Darstellung in die Teile, die der abstrakten Definition entsprechen und dem Rest. In einem Beispiel definiert er deterministische endliche Automaten auf der abstraktesten Ebene und erkennt alle jene Teile der Darstellung, die dieser Definition genügen. Akehurst (2000) überträgt diesen Ansatz auf UML-Diagramme.

Der beschriebene Ansatz verwendet Graphgrammatiken insofern ungewöhnlich, als dass er sie zur „Übersetzung“ zwischen Sprachen verwendet. An sich verfolgen Graphgrammatiken einen transformativen Ansatz. Sie definieren für Teilgraphen, wie diese durch andere Teilgraphen ersetzt werden können. Für die Ersetzung werden zusätzliche Einbettungsregeln angegeben. Eine Übersetzung von einer Darstellung in einer visuellen Sprache in eine andere könnte theoretisch einfach dadurch implementiert werden, dass für Regeln der Graphtransformation keine Einbettungsregeln mit angegeben werden. In diesem Fall würde jedoch die Information verloren gehen, welche Bestandteile bereits in welcher Weise übersetzt wurden. Außerdem könnte ein bestehendes Diagramm, das schon aus einer Produktion hervorgegangen ist, nicht erweitert werden. Rekers (1994) führt die Übersetzung auf die Einbettung zurück, indem ein Graph immer um die übersetzte Struktur erweitert wird. Dabei wird markiert, aus welchen Teilen die Erweiterung hervorgegangen ist, um eine Anwendung der gleichen Produktionsregel auf den gleichen Teilgraph zu verhindern.

In Rekers wird vor allem auf den Aufwand zum Parsen hingewiesen. Der entsteht vor allem deshalb, weil kein inkrementelles Parsen definiert ist und weil keine implizite Reihenfolge auf den Produktionen angegeben werden kann.

- *Signaturen.* Wang und Lee (1993) thematisieren, dass sich die Bedeutung graphischer Elemente durch deren Benutzung ergibt. Daraus leiten sie ab, dass die Interpretationen oft keine festen Eigenschaften der Darstellungen sind, sondern Eigenschaften davon, wie sie verwendet werden. Eine wichtige Unterscheidung besteht zwischen grafischer Inferenz („graphical inference“) und visuellem Schließen („visual reasoning“). Ersteres beschreibt den algorithmischen Vorgang, aus einer graphischen Darstellung die räumlichen Eigenschaften abzuleiten. Letzteres beschreibt das Schließen in einer typischerweise nicht graphischen Domäne auf Basis der grafischen Darstellung. Die grafische Inferenz ist domänenunabhängig während das visuelle Schließen domänenabhängig ist und durch Nutzerinnen und Nutzer zugeordnet werden kann.

  Über Signaturen beschreiben sie visuelle Sprachen. Darin werden Sorten, Funktionen und Prädikate definiert sowie eine Ordnung auf den Sortensymbolen. Eine grafische Theorie besteht aus Axiomen, die für diese Signatur gelten sollen.

  Eine Situation wird eingeführt als die maximale Menge aller Aussagen (Prädikaten-ausdrücke), mit denen ein Bild beschrieben werden kann, die aus der grafischen Theorie folgen. Im Gegensatz dazu sind „basic facts“ ein Teil der Situation, der direkt aus dem Bild ablesbar ist.

  Die geometrische Semantik eines konkreten Bildes ist durch einen Tupel aus einer Subsignatur beschrieben, mit der die Objekte des Bildes beschrieben werden können und der Menge der “basic facts“.

  Eine Interpretation bildet die Prädikate der Signatur in eine Domäne ab. Wang und Lee (1993) stellen dafür das System GAR vor, dass eine solche Zuordnung von Interpretationen erlaubt.

Die dargestellten Ansätze verdeutlichen vor allem den umfangreichen theoretischen Hintergrund, der den Themenbereich des Schließens in visuellen Sprachen umgibt.

Aus meiner Sicht interessant ist der Ansatz, Graphgrammatiken einzusetzen, um abstraktere Repräsentationen der Darstellungen zu erhalten. Auch wenn dies in den referierten Ansätzen nicht durchgeführt wurde, ermöglicht dies, später operationale Semantiken für bestimmte Graphmodelle zu hinterlegen.

Der Signaturenansatz betont die Notwendigkeit, dass Darstellungen meistens eine Interpretation im Kontext einer Verwendung zugeordnet werden sollte. Axiomatisiert

werden räumliche oder auch logische Beziehungen zwischen Objekten. Diese Axiome werden dann für Domänen interpretiert, so dass dann ein logischer Schluss für eine Domäne möglich ist. Das heißt, dass der Ansatz einen „alltäglichen" Einsatz andeutet. Eine Situation als maximale Menge aller folgenden Aussagen zu definieren, scheint mir jedoch sehr weit gegriffen, da damit im Prinzip ein vollständige Logik gefordert wird.
In diesem Zusammenhang ist es interessant, sich der Bandbreite visueller Darstellungen bewusst zu werden, wie sie eindrucksvoll in Lohse, Biolsi, Walker und Rueter (1994) aufgeschlüsselt wird. Darin werden 60 verschiedene Repräsentationen u.a. Repräsentationstypen zugeordnet wie „structure diagrams, cartograms, maps, graphic tables, process diagramms, icons, time charts, network charts, pictures, tables, graphs".

2.3 Lernunterstützung

Die Verwendung von visuellen Sprachen beim Lernen geht im Wesentlichen auf Annahmen aus psychologischen Wissenstheorien zurück. Mandl und Fischer (2000) ordnen Strukturierungsansätze drei Perspektiven zu: der psychometrisch-strukturellen Perspektive, der kognitiven Perspektive und der pädagogisch-psychologischen Perspektive.
Psychometrisch-strukturelle Ansätze zielen darauf ab, „psychische Phänomene und deren Veränderung mathematisch zu quantifizieren" (Mandl & Fischer, 2000, S. 4). Ein Beispiel dafür stellt die Struktur-Lege-Technik (Scheele & Gröben, 1988) dar. Darin werden über das Anlegen von Begriffsnetzen „subjektive Theorien" (Mandl & Fischer, 2000) erhoben, an denen bei mehrmaligen Durchläufen Veränderungen erkannt und bewertet werden können.
Unter der kognitiven Perspektive wird davon ausgegangen, dass „das Wissen um Sachverhalte und Zusammenhänge kognitiv in der Form vernetzter Begriffe kodiert ist" (Mandl & Fischer, 2000, S. 5). Deshalb wird angenommen, dass Netzstrukturen zwischen Begriffen dafür eine angemessene externe Darstellung sind. Vor diesem Hintergrund wurden z.B. „Concept Mapping"-Verfahren zur Darstellung von Begriffsschemata und Begriffszusammenhängen entwickelt.
Pädagogisch-psychologische Ansätze zielen auf die „Unterstützung von Verstehensprozessen". Sie sollen vor allem das selbstgesteuerte Lernen unterstützen. Darin wird das generelle Problem adressiert, komplexe Information aufzuarbeiten. Um diesen Vorgang zu unterstützen wird häufig mit Visualisierungen gearbeitet. Sie sollen z.B. dazu anregen, Aussagen zu präzisieren und zu strukturieren und auch in Teilaspekte oder Teilprobleme

zu zerlegen, um so den Schwierigkeitsgrades eines Problems zu reduzieren (Fischer, Gräsel et al., 1996). Eine Schwierigkeit besteht beispielsweise im Umgang mit widersprüchlichen Einzelaspekten, die wirkt teilweise so verwirrend wirken können, dass das dahinter stehende Problem nicht mehr strukturiert gelöst wird (Mandl & Fischer, 2000). Weiterhin verbinden Graphiken eine Vielzahl von Informationen und ermöglichen deshalb eine reichhaltigere Erinnerung. Es wird auch davon ausgegangen, dass die aktive Auseinandersetzung mit Inhalten bei der Erstellung von Maps das Lernen fördert. Aber auch negative Aspekte werden genannt. Beispielsweise gehen nicht alle Personen im gleichen Maße versiert mit solchen visuellen Darstellungen um, weshalb es auch bei dieser Lernmethode zu einer Ungleichbehandlung kommen kann.

Eine (lern-)methodische Einteilung der Verwendung von Wissensstrukturdarstellungen stellen Jüngst und Strittmatter (1995) vor. Sie trennen „Techniken zur Repräsentation“ (für wissensdiagnostische Anwendungen), Techniken „zur Übertragung/Vermittlung (im didaktischen Sinne/ Instruktion)“ und Techniken „zum Erwerb von Wissensstrukturen (im Sinne der Selbstinstruktion)“.

Für viele Ansätze werden „Concept Maps“ eingesetzt, in denen Begriffe durch beschriftete Relationen in ein semantisches Verhältnis gestellt werden. Darauf können sowohl wissensdiagnostische wie auch instruktive Verfahren aufsetzen. Entsprechend ist „Concept Mapping“, die größte Teilgruppe visueller Sprachen (Kremer, 1998), selbst eher eine Aussage über die repräsentationalen Hilfsmittel als über ein konkretes Verfahren.

Jüngst und Strittmatter (1995) erläutern noch viele weitere netzbasierte Ansätze, wie z.B. „Pathfinder“-Netze und „Spinnen-Netze“. „Pathfinder“-Netze werden dem Bereich der Diagnose von Wissen zugerechnet. Die Relationen werden gewichtet bzgl. der semantischen Ähnlichkeit zwischen Begriffen. Eine Ähnlichkeit kann dann auch über Pfade kalkuliert werden. Die „Spinnen-Netze“ sollen dafür verwendet werden, allgemeinere Ideen in Teilideen zu untergliedern. Nach Jüngst und Strittmatter (1995) fallen sie deshalb in die Kategorie der Verfahren zur Selbstinstruktion.

Häufig ist die Intention, „Mapping“-Verfahren zum Lernen einzusetzen, gepaart mit der Idee, überprüfbare Ergebnisse zu erhalten, um einen Wissensstand bewerten zu können. In Bernd, Jüngst, Hippchen und Strittmatter (2000) wird ein Lernverfahren vorgestellt, das auf einem gegebenen Expertennetz basiert. Dieses wird durch Expertinnen oder Experten erstellt und liegt dann einer Bewertung als Maßstab zugrunde. In Bernd et al. (2000) wird ein Netz als eine Art Lückentext eingesetzt. In die lückenhafte Struktur müssen dann aus vorgegebenen Begriffs- und Relationslisten die Inhalte eingefügt

werden. Insbesondere sollen im Verlauf für Relationen auch Gegenrelationen gefunden werden.

Können Begriffe und Bezeichnungen für Relation frei gewählt werden, entsteht ein Problem bei der Bewertung von Netzen, die dann kaum noch automatisch ermittelt werden kann, da keine semantische Abbildung der neuen auf vorhandene Begriffe existiert. Das würde die Existenz einer modellierten Domäne, z.B. in Form von Ontologien, voraussetzen und ist prinzipiell mit großem Wissensrepräsentationsaufwand verbunden.

In Eckert (2000) und Weber und Schumann (2000) werden solche begriffsorientierten Diagnose-Verfahren in Richtung strukturorientierter Diagnose-Verfahren erweitert. Bewertung wie Konzeptzentralität, Distanzen oder Zerklüftetheit stellen strukturelle Eigenschaften dar, die zur Bewertung der Netze herangezogen werden.

Die meisten Lernansätze, die visuelle Sprachen einsetzen, sind für die individuelle Arbeit entwickelt worden. Sobald eine Diskussionssituation angestrebt wird, verschiebt sich der Schwerpunkt von der Instruktion oder der Diagnose hin zur Konstruktion von Inhalten. Kooperativ nutzbare Software wird dann wichtig, die zu Überarbeitungen anregt und flexible Konstruktionselemente anbietet.

2.4 Wissenselizitation

Bei Wissenselizitationsmethoden ist eine deutliche Verzahnung von Methoden und Fundierungen aus verschiedensten Bereichen festzustellen wie Psychologie, Pädagogik, Wissens-Engineering, Kognitionswissenschaften, Philosophie (Cooke, 1994). Der Prozess der Wissenselizitation kann auch als Externalisierung bezeichnet werden und wird ausschließlich auf menschliche Wissensquellen bezogen (Cooke, 1994). Dieser Prozess ist eingebettet in die Wissensakquisition, die von Cooke ausdrücklich als Wissenskonstruktion und nicht als Auffinden von Daten verstanden[6] wird und kann deshalb fast immer auch als Lernprozess eingestuft werden.

Hinter der „differential access hypothesis“ (zitiert in Cooke, 1994, S. 804), wonach mit unterschiedlichen Methoden auch unterschiedliches Wissen erhoben wird, stehen vergleichbare Erfahrungen, wie hinter den „active learning strategies“ nach Jonassen und Carr (2000), nach denen durch unterschiedliche Repräsentationen ein tieferes Verständnis von Inhalten erreicht werden soll. In allen Fällen wird von einem deutlichen Einfluss der

[6] „knowledge acquisition is modeling or construction, not mining“ (Cooke, 1994, S. 802)

Repräsentationsmittel auf die erhobenen oder dargestellten Ergebnisse erwartet, wie das auch in Suthers (1999A, 1999B, 2003) geschildert wird.
Cooke differenziert neben Wissenselizitationsmethoden, die auf der Beobachtung oder Interviews aufsetzen solche, die Prozesse auswerten und solche, die Konzepttechniken einsetzen, in denen Begriffe strukturiert und zueinander ins Verhältnis gesetzt werden. Oft werden dazu Darstellungsweisen wie Tabellen und Graphen gewählt, in denen fehlende Relationen und Konzepte offensichtlich werden.
Ein Beispiel dazu stellt das „repertory grid"-Verfahren dar, das von Kelly (1970) im Kontext seiner „personal construct theory" entwickelt wurde. Darin wird davon ausgegangen, dass Begriffe subjektiv und individuell konstruiert sind. Sie werden deshalb „Konstrukte" genannt. In „repertory grids" werden diese Konstrukte weiter in ihre Ausprägungen oder Instanzen unterteilt. Zu diesen bilden dann individuelle Kriterien und Einschätzung eine weitere Dimension, nach der die Instanzen eingeschätzt werden sollen. Gaines und Shaw (1993) stellen diesen Ansatz als Methode der Wissenselizitation vor und verbinden sie beispielsweise mit einem Cluster-Verfahren, bei dem Konzepte generiert werden.

2.5 Kommunikationsmedien und Wissensprodukte

> Unter **Wissensprodukten** verstehe ich hier das Ergebnis eines Arbeitsprozesses, in das Wissen verschiedener Personen eingeflossen ist.

Der Prozess, der zu einem Wissensprodukt führt, ist vor allem auch ein Kommunikationsprozess. Visuelle Sprachen werden eingesetzt, um diesen Prozess zu unterstützen. Das Produkt können dann Darstellungen der Ergebnisse mit diesen visuellen Sprachen sein oder auch Gegenstände, Texte, Designs, Konzepte, etc., die auf Grundlage dieser Darstellungen entwickelt werden. Der Kommunikationsprozess, der zu einem Wissensprodukt führt, wird hier als Wissenskommunikation bezeichnet.

> Die **Wissenskommunikation** wird in der gesamten Arbeit als eine Kommunikation zwischen Menschen verstanden, die Computermedien in die Kommunikation integrieren. Sie umfasst den wechselseitigen Erwerb neuen Wissens und die Angleichung von Wissenselementen und –strukturen. Dabei kommt es zur Reflexion vorhandener Ansichten und Verständnisse, zur Revision der „mentalen Modelle" der Beteiligten (Gaßner & Schröder, 2003). Mit implizitem Wissen („tacit knowledge"), vagem Wissen oder

neuen auch unvollständigen Informationen kann bestehendes Wissen erweitert, überprüft und verändert werden.

2.5.1 Kommunikationsmedien

Wird das Medium weniger zur Übermittlung vorbereiteter Darstellungen mit asynchroner Rezeption verwendet, als vielmehr bei der synchronen Kooperation, verschiebt sich die Wissenskommunikation von einem vermittelten Austausch hin zu einem Prozess des „shared understanding" einer „practice[...] of meaning-making in the context of joint activity" (Koschmann, zitiert in Stahl, 2002, S. 1). Dabei wird mit Wissen handlungsorientiert und gestalterisch umgegangen.

Für die Wissenskommunikation sind also die verfügbaren Ausdrucksmittel relevant. Doch viele prominente Kommunikationsmedien wie Chats, E-Mails und Newsgroups reduzieren den Austausch von Wissen auf Sprache oder Texte und greifen zu kurz, wenn Ausdrucksmittel benötigt werden, um den Gegenstand selbst zu gestalten. Auch bergen sie ihre eigenen Probleme bei der Gestaltung der Kommunikation. Beispielsweise leiden Newsgroups und Diskussionsforen wegen der verwendeten Baumstruktur unter der Divergenz der Diskussionsstränge, wegen der dargestellten Historie, ist es schwer, den aktuellen Diskussionstand zu erfassen und es ergeben sich Probleme, wenn unterschiedliche Themen in einer Diskussion behandelt werden (Suthers & Xu, 2002).

Gemäß der „Activity Theory" (Engeström, Miettinen & Punamäki, 1998) werden menschliche Aktivitäten über Tools (im sehr weiten Sinn) vermittelt. Diese Werkzeuge beeinflussen einerseits die Interpretation von Aktivitäten, andererseits konstituieren sich zusammen mit ihnen die ausführbaren Aktivitäten. In Abgrenzung zur „distributed cognition" (z.B. Pea, 1993), nach der soziales/kulturelles Wissen auch in Objekten kodiert ist, geht die „Activity Theory" davon aus, dass Personen sich die Verwendung von Objekten „selbst erfinden". Für eine erfolgreiche Wissenskommunikation sollten die Beteiligten also mit den geeigneten flexiblen Mitteln ausgestattet sein, sich Wissen anzueignen, Wissen darzustellen und auszutauschen.

In diesem Zusammenhang wird im folgenden Abschnitt ein Exkurs über den Medienbegriff von Dewey in der Interpretation von Vogel (2001) eingeblendet.

> Vogel zieht den Medienbegriff nach Dewey heran, um „einen medientheoretisch erweiterten Begriff des Verstehens" (Vogel, 2001, S.137) zu entwickeln, der den Gebrauch von Medien handlungstheoretisch einbettet. Obwohl Deweys Medienbegriff speziell für die Kunstrezeption erläutert wird, stehen darin kommunikative Aspekte im Vordergrund. Kunstwerke sind dort Medien, über die der

Künstler oder die Künstlerin mit Rezipierenden kommuniziert. Die Kommunikation über das Medium wird so beschrieben, dass die produzierende Person das Werk im Hinblick auf die erwartete Rezeption anderer Personen gestaltet, welche wiederum das Werk unter dem Gesichtspunkt der erwarteten Intentionen der produzierenden Person aufnehmen. Diese Kommunikation vollzieht sich im Kontext gesellschaftlich geprägter sowie individueller Erfahrungshintergründe, beinhaltet aber eine aktive, handelnde Beteiligung aller Kommunikationspartner. Dewey macht deutlich, „ ... dass er Medien im Rahmen von Handlungen, also im Rahmen bewusster intentionaler Aktivitäten, analysiert“ (Vogel, 2001, S. 141).
Aus den Darlegungen von Dewey extrahiert Vogel die folgenden Kriterien für Medien, die nun weitgehend unabhängig davon formuliert sind, ob es sich um Kunstwerke oder andere Medien handelt.

> *M ist ein Medium, wenn gilt:*
> 1. *M wird intentional gebraucht.*
> 2. *M wird in expressiver, mitteilender Absicht verwendet.*
> 3. *M geht konstitutiv in die Akte oder Produkte ein, zu deren Erzeugung es herangezogen wird.*
> 4. *M wird*
>   a. *im Rahmen einer stabilisierten performativen Praxis oder*
>   b. *durch konstitutive Regeln sozial etabliert.*
> 5. *M erlaubt das Hervorbringen von Ausdrucksakten oder –produkten, die öffentlich wahrnehmbar sind und für ihre Rezipienten Gegenstand einer Erfahrung im Sinne einer integrierten Episode werden können.“*
>
> *(Vogel, 2001, S. 153-154)*

Ein Medium wird immer mit einer bestimmten Zielstellung eingesetzt (1). Dass z.B. ein Stein zum Hämmern eingesetzt wird, macht ihn aber noch nicht zu einem Medium. Um eine Abgrenzung vom Werkzeugbegriff zu erreichen, werden deshalb noch zwei weitere Kriterien hinzugefügt: Erst die kommunikative Zielstellung und der Verwendungskontext lassen ein Mittel zu einem Medium werden (2). Die Ausdrucksmittel wiederum prägen, was mit ihnen ausgedrückt wird. Medien gehen also konstitutiv in das Produkt mit ein (3). Beispielsweise können Steine, ohne etwas bestimmtes auszudrücken, zum Hausbau verwendet werden oder medial eingesetzt werden, indem sei beim Kirchenbau konkrete religiöse Aussagen vermitteln. Gelingende Kommunikation ist der Mechanismus der Etablierung von Medien (4a). Mediale Kommunikationsmittel sind damit weitgehend unabhängig von sprachlichen Vereinbarungen. Auch soziale Regeln wie Tabus und Umgangsformen können eine Kommunikation konstituieren (4b). Kriterium (5) bezieht sich ganz maßgeblich auf Kunstwerke als Medien. Die Ausdrucksfähigkeit der Kunst wird im Unterschied zu Sprache als eher

kulturübergreifend eingeschätzt. Sie erlaubt jeder Person einen Zugang vor dem Hintergrund individueller Erfahrungen.

Auf Grundlage dieses Medienbegriffs ist die Rolle, die man Medien beimisst, neu zu überdenken. Meistens wird ihnen eine eher repräsentationale Rolle in einem Produzenten-/Konsumentenverhältnis zugedacht bei dem die Medien ihre Interpretationsgrundlagen mitliefern müssen (z.B. statische Darstellungen von Informationen auf Webseiten). In dieser Sicht besteht kaum ein Handlungsspielraum für die Konsumentenseite. Die Kommunikation mit Hilfe von Medien ist nach Vogel ein Ausgangspunkt rationalen Verstehens bei dem die Kommunizierenden situativ und individuell eine gemeinsame Ausdrucksweise entwickeln. Mit Medien können und sollen Sachverhalte handelnd in Aktion und Reaktion begriffen werden, ohne sie erst in sprachliche Mittel zu übersetzen.

> *„ … der Blick des Rezipienten auf die vermeintliche Perspektive des Produzenten und der Blick des Produzenten auf mögliche Perspektiven der Rezipienten, und damit die kommunikative Dimension ästhetischer Erfahrung, sind auf vermittelnde Momente angewiesen, wenn sich diese Blicke nicht in purer Kontingenz verlieren sollen. Es sind genau diese vermittelnden Momente, die Dewey* Medien *nennt."*
>
> *(Vogel, 2001, S. 140)*

Die Kernidee eines Mediums besteht danach weder in der Lösung eines Transportproblems, noch in der sozialen Handlungskoordination. Vielmehr geht es um das „Zumverstehen-geben" (Vogel, 2001, S. 140).

Bei der Wissenskommunikation mit Medien müssen zwei wesentliche Aktivitäten ausgeübt werden. Das, was vermittelt werden soll, muss mit den gegebenen Mitteln ausgedrückt werden und die kommunizierten Inhalte müssen rezipiert und behandelt werden, um zu einem Ergebnis zu gelangen. Das Medium übt dabei einen normativen Einfluss aus. Die vorhandenen Mittel beeinflussen, was mit ihnen ausgedrückt werden kann. Darunter fallen sowohl die technischen Gegebenheiten des Mediums als auch die Möglichkeiten des jeweiligen Notationssystems. Der – nach Dewey ästhetische – Erfahrungshintergrund manifestiert sich sowohl in der Gestaltung der Mittel als auch in deren Rezeption.

Digitale Medien und moderne Kommunikationstechnologien eröffnen ein ganzes Spektrum an Möglichkeiten, diese Kommunikation zu gestalten. Es liegt auf der Hand, den Ansatz der kognitiven Tools mit dem der technisch ausgerichteten Kommunikationsermöglichung zu verbinden.

Die Rezeption und deren Wirkung beeinflusst maßgeblich den weiteren Verlauf der Kommunikation. Deshalb müssen in diesem Zusammenhang vor allem die Ausdrucksmittel diskutiert und experimentell untersucht werden. Den Einfluss, den Medien auf die

Wissensgenese ausüben, charakterisiere ich als aktiv oder passiv (vgl. Gaßner, Hoppe, Lingnau & Pinkwart, 2003).

> Ein **passiver Einfluss eines Mediums** entsteht schon dadurch, dass eine visuelle Sprache verwendet wird, die den Sprachumfang und die Notation bzw. die Symbole festlegt. Ebenso wirkt sich die verwendete Technologie und die Funktionalität auf die Ausdrucksmächtigkeit aus.
>
> Ein **aktiver Einfluss des Mediums** liegt dann vor, wenn es in die Gestaltung eingreifen kann, wenn es direkten Einfluss auf den Prozess nimmt, also z.B. wenn es eine Methode definiert.

Anders ausgedrückt liegt ein passiver Einfluss vor, wenn ein Mensch die Darstellung interpretiert, ein aktiver, wenn die Darstellung durch die Maschine ausgeübt wird und zu systemgesteuerten Interaktionen führt.

Der passive Einfluss wird schon durch die verwendete Notation ausgeübt. Suthers (1999A, 1999B, 1999C) erläutert, dass durch die Notation die Constraints für die Ausdrucksmöglichkeiten vorgegeben sind, also auch die Syntax der Ausdrücke. Mit „salience“ wird das Phänomen beschrieben, dass die Darstellungen bestimmte Aspekte hervorheben. Daraus resultiert, dass bestimmte Informationen daraus wieder leicht abzulesen sind, andere aber schwer. Die Darstellungen können entsprechend auch hervorheben, welche Inhalte oder Strukturen noch fehlen.

Tools, die visuelle Sprachen anbieten, können meistens auch als kognitive Tools charakterisiert werden. Kognitive Tools sollen helfen, kognitive Aufgaben wie Problemlösen, Schließen oder auch die Entscheidungsfindung zu unterstützten (Zhang, 1997; Pea, 1985). Lajoie (1993) charakterisiert diese Tools auch dadurch, dass meta-kognitive Prozesse unterstützt werden, in denen es nicht mehr nur um die konkreten Inhalte geht, sondern das eigene Handeln reflektiert und kontrolliert wird, Strategien gewählt werden, Wissen und Annahmen über sich selbst bedacht werden, auch Abstraktionen und Verallgemeinerungen gefunden werden. Ein kognitives Tool stellt die Mittel zur Externalisierung von Wissen zur Verfügung, wodurch in der Konsequenz auch das Gedächtnis entlastet wird. Visualisierte Inhalte und Daten sind für alle sichtbar und stehen damit während der ganzen Diskussion passiv zur Verfügung.

Während der Kommunikation auf Basis von „shared“ Workspaces dienen die Repräsentationen sowohl als Bezugspunkt als auch als Ausgangspunkt der Kommunikation (Suthers, 1999A). Die Nutzerinnen und Nutzer werden dazu gebracht, sich innerhalb der vorgegebenen Notation auszudrücken, wenn sie über die externen Repräsentationen

sprechen wollen. Davon wiederum sei im Standardfall auszugehen, da Personen den einfachen Bezug auf eine Darstellung einer komplexen Beschreibung eines Sachverhaltes vorziehen.

Weitere Forderungen an kognitive Tools können allerdings kaum noch durch ein passives Medium geleistet werden. Beispielsweise sollen Schlussfolgerungen erleichtert werden.

Der Übergang von einer passiven zu einer aktiven Unterstützung wird im Allgemeinen begleitet durch ein Domänenmodell, ein Logik, Constaint-Ansätze, wissensbasierte Agenten, etc. Die jeweilige Unterstützung kann sich entweder an der Domäne orientieren oder an syntaktische Eigenschaften der Darstellung zur Grundlage haben.

Als aktive Unterstützung können auch Methoden gewertet werden, die für Objekte operationale Semantiken definieren, die zustandsbasiert auf den aktuellen Kontext einer Modellierung reagieren: „computational objects to think with“ (Pinkwart et al., 2002). Die Computer-Umgebung übernimmt dort eine aktive bzw. reaktive Funktion: „the environment talks back“. In dieser Umgebung können spielerische und explorative Lernsituationen ermöglicht werden, mit denen beispielsweise der Vorgang des Formulierens und Testens von Hypothesen gut umgesetzt werden kann. Modelle, die aus Elementen visueller Sprachen aufgebaut werden, können interaktiv getestet und die Ergebnisse direkt wieder in die Kommunikation eingebracht werden (Pinkwart et al., 2002)!

2.5.2 Wissensprodukte

Mit Bezug auf den Beginn dieses Abschnittes werden folgend noch Beispiele aufgeführt, wie Kommunikationsmedien zur Erstellung von Wissensprodukten eingesetzt werden.

Ergebnisse aus Kreativphasen. Ein Beispiel für die Verwendung von visuellen Sprachen als Kommunikationsmedium sind Kreativtools, deren Einsatz dazu führen soll, neue Ideen zu entwickeln. Dies ist oft mit dem Versuch kombiniert, den Kontext eines Problems aufzuschlüsseln. Der Arbeitsprozess beim Einsatz eines Kreativtools kann deshalb sowohl den Charakter einer Wissenskonstruktion haben als auch den Ansatz verfolgen, Gedanken oder Zusammenhänge zu analysieren oder zu verstehen.

Typische Kreativtechniken basieren auf der Assoziation, der Bisoziation oder der Analogie (Lehmann, 1998). Bei der Assoziation werden vorhandene Erfahrungen miteinander verknüpft. Bei der Bisoziation werden Problemlösungen gesucht, indem willkürliche Bilder gezeigt werden. Diese sollen Ideen für die Problemlösung hervorrufen. Bei der Analogie wird versucht, Verläufe oder Lösungen in vergleichbaren Bereichen zu finden und darüber analoge Lösungen zu erarbeiten. In einem recht

anschaulichen Artikel der Zeitschrift c't (Jungbluth, 1998) werden verschiedene solcher Methoden und Systeme erläutert. Darunter fällt das Brainstorming als typisches Beispiel für eine Assoziationsmethode, worin zu einem Problem Ideen gesammelt werden. Ebenfalls als Assoziationsmethode klassifiziert ist der morphologische Kasten. Dabei handelt es sich um eine tabellarische Darstellung eines Inhaltsbereiches unter verschiedenen Aspekten.

Auch das „Mind Mapping" ist den Kreativ-Methoden zuzuordnen. Dabei werden im Standardfall ungerichtete Baumstrukturen aufgebaut, deren Bestandteile Ideen, Einfälle, Assoziationen, etc. sein können. Dieses Vorgehen ist im Wesentlichen schlagwortbasiert, da ausgefeilte Beiträge kontraproduktiv sind weil sie den Denkprozess behindern.

Kreativmethoden zielen meistens auf die Elizitation und das Hervorrufen von Ideen durch neue Kombinationen und Assoziationen. Ihren medialen Charakter erhalten die Tools vor allem durch den aktiven Einfluss der angewandten Methoden und den passiven Einfluss der Darstellung.

Texte und Hypertexte. Ein anderes Beispiel für ein Wissensprodukt sind gemeinsam erstellte Texte und Hypertexte. Das Sepia-System (Streitz, Haake, et al., 1992; Streitz, Haake, et al., 1998) beispielsweise wurde ursprünglich dazu entwickelt, um das kooperative Schreiben von Hypertexten zu unterstützen (siehe dazu Kapitel 3.2). Darin sollen verschiedene visuelle Sprachen für verschiedene Phasen der Vorbereitung verwendet werden. So kann beispielsweise in einer Sprache auf die Sammlung von Inhalten fokussiert werden, in einer anderen die Rhetorik aufbereitet werden. Das Ergebnis besteht in diesem Fall nicht im erstellten Hypertext selbst, sondern in einer ausführlichen Vorbereitung dazu. Der Hypertext kann dann in einem anderen Medium umgesetzt werden. Das System sollte vor allem als Medium für das kooperative Schreiben dienen. Dabei muss das Zusammentragen der Inhalte unterstützt werden und die verschiedenen Ansichten und Argumente in Übereinstimmung gebracht werden.

Ein typisches Problem der Produkterstellung besteht darin, dass es währenddessen oft zu Medienbrüchen kommt, wenn die Darstellungsmittel gewechselt werden. Bei Wissensprodukten in Form von digitalisierten Texten können Medienbrüche jedoch auch vermieden werden, wenn geeignete Schnittstellen implementiert sind.

Design-Produkte. Im Designbereich werden visuelle Sprachen dazu eingesetzt, Design-Entscheidungen strukturiert im Zusammenhang mit ihren Begründungen darzustellen. Man erhofft sich davon eine Rationalisierung des Design-Prozesses, da so Argumente offengelegt werden und der Design-Prozess transparenter wird. Die festgehaltenen Begründungen sollen später für Re-Designs ausgenutzt werden, um die Wiederholung

von Fehlern zu vermeiden. Dieser Ansatz, den Design-Prozess durch eine strukturierte Dokumentation zu begleiten, wird als „design rationale“ (DR) bezeichnet.

Der Anlass einer Produktneuentwicklung besteht oft darin, dass ein Vorgängerprodukt unzulänglich gestaltet war. Somit ist es wichtig, die Entwicklungshintergründe und bereits entstandene Erfahrungen und Überlegungen mit einzubeziehen. Nach McKerlie und MacLean (1994) soll DR helfen, die Gedanken und Argumente hinter den eigentlichen Design-Entscheidungen zu externalisieren.

Die meisten Ansätze im Bereich des DR erwarten aber noch mehr von den entwickelten Lösungen. Neben der objektiven Verbesserung der Arbeitsergebnisse soll auch die Externalisierung und die Kooperation verbessert werden. Lee und Lai (1996) nennen als Ziele des DR die Dokumentation, Verstehen, Debugging, Verifikation, Analyse, Erklärungen, Modifikationen und Automatisierung sowie „Re-use“, „Re-design“ und Wartung. DR erschöpft sich also nicht in der Spezifikation eines Produktes, sondern geht deutlich darüber hinaus (Moran & Carroll, 1996).

Wegen der Relevanz von Begründungen und Argumentationen im DR steht dort traditionell die Visualisierung und Repräsentation von Argumenten im Mittelpunkt. Buckingham Shum und Hammond (1994) stellen eine ganze Reihe von argument-basierten DR-Ansätzen vor. Im Zentrum ihrer Erörterung steht die Frage, wie erfolgreich diese Ansätze sind. Sie kommen jedoch zu der Feststellung, dass unter Maßgabe diverser existierender Evaluationsergebnisse dazu eigentlich kaum eine Aussage getroffen werden kann.

Die meisten bestehenden DR-Ansätze basieren entweder auf einer Studie von Kunz und Rittel (1970), die auch in die IBIS-Methode (Conklin & Begemann, 1987; vgl. auch gIBIS, Kapitel 3.2) eingeflossen ist, oder auf der Argumentfigur nach Toulmin (1958) (Lee & Lai, 1996). Unterschieden wird ein „historischer“ Ansatz bei dem die Entwicklung eines Designs entsprechend durch die Dokumentation der sich ebenfalls entwickelnden Argumente dargestellt wird und ein „psychological claim“-Ansatz bei dem Hintergründe für den Zustand eines Produktes dokumentiert werden. Drittens gibt es noch den „design space“-Ansatz, der den Versuch macht, den gesamten Raum der Möglichkeiten zu erschließen (Lee & Lai, 1996).

Die verwendeten Darstellungsmittel sind meisten Graphen, deren Knoten- und Kantentypen eine visuelle Sprache bilden, die für den Einsatz einer bestimmten Methode entwickelt wurden wie z.B. bei QOC (MacLean, Young, Belloti & Moran, 1991) und IBIS (Conklin & Begemann, 1987). Wegen dieser Fokussierung auf die visuellen Sprachen steht auch die Diskussion um die Elemente dieser Sprachen oft im Mittelpunkt

der Beiträge (Buckingham Shum und Hammond, 1994). So nimmt beispielsweise Fischer, Lemke, McCall & Morch (1996) ausführlich Stellung, inwiefern die PHI-Methode (Procedural Hierarchy of Issues) den IBIS-Ansatz verbessert, mit der nun verstärkt der Design-Raum entwickelt werden kann.

Ein wesentliches Problem solcher Begründungs- bzw. Dokumentationsansätze scheint in dem Medienbruch zu bestehen, der zwischen einer schriftlichen Auseinandersetzung und der Konstruktion des Produktes selbst besteht. Fischer, Lemke et al. (1996) verweisen auf grundlegende Probleme von Designern und Designerinnen, die abgekoppelten Argumentationsnetze konstruktiv in ein neues Design einzubeziehen, wenn kein geeigneter Bezug zu den Artefakten vorhanden ist. Design sei ein Wechsel zwischen „reflection-in-action" und „knowing-in-action" wobei die Konstruktion die Aktion und die Argumentation die Reflektion verkörpert. Die Argumentation sollte nicht immer externalisiert werden, sondern nur in „Breakdown"-Situationen. Fischer, Lemke et al. (1996) fordern deshalb, während des Design-Prozess bereits Modelle der Artefakte zu entwickeln, an denen Tests durchgeführt werden können.

Gruber und Russel (1996) extrahieren auf Basis ausführlicher Untersuchungen von Dialogen unter Designern und Designerinnen welche Informationen bei einem Produkt-Design überhaupt ausgetauscht werden: Anforderungen (an ein Design-Produkt), Struktur und Form, interne und allgemeine Funktionalität, Hypothetische Fragestellungen (z.B. „was wäre wenn ..."), Abhängigkeiten, Constraints, Entscheidungen, Alternativen, Erklärungen, Simulationen auf Modellen, Definitionen. Es zeichnet sich also ein sehr diverses Bild von Inhalten ab, die alleine durch Argumentationsdarstellungen nicht ausreichend abgebildet werden können.

Gruber und Russel (1996) zielen dementsprechend auf eine Integration unterschiedlicher unterstützender Hilfsmittel, die zusätzlich mit den eigentlichen Konstruktionswerkzeugen verknüpft sind. Von ihnen wie von Fischer, Lemke et al. (1996) wird deshalb eine Verschiebung des Schwerpunktes im DR für notwendig gehalten: von der Ausrichtung auf Argumentationsnetze hin zu deren Integration von Konstruktions-Tools. Sie fordern ein „rationale by demonstration". Für dessen Realisierung schlagen sie fünf Sichten auf die Design-Informationen vor: „Requirements, Function, Objective, Structure, Behaviour". Zwischen diesen sollen Abhängigkeiten für jeweilige Beispiele definiert werden.

Gruber und Russel (1996) verweisen aber auch noch auf diverse Umstände beim DR, die nur schwer mit einem zu formalen Ansatz vereinbar sind. Nach ihren Untersuchungen stellten sie fest, dass der größte Teil von Äußerungen bei Design-Dialogen nicht sehr

formal bzw. detailliert ist („weak explanations“). Das heißt, dass zwar „zulässige“ Designs aus formalisierten Daten generiert werden könnten, dass aber damit das, was das eigentliche Design ausmacht, nämlich vage und unvollständige, oft sehr subjektive Informationen, kaum einbezogen würden.

Weiterhin wird dort auf die Einzigartigkeit jedes Design-Prozesses verweisen. Den gesamten „Design-Raum“ darzustellen ist quasi eine unmögliche Aufgabe. Jedes Re-Design wirft wieder völlig neue Fragestellungen und Umstände auf, die vorher gar nicht abzusehen waren. Dies führt tendenziell zu der Erwartung, dass Sammlungen von Fakten und Informationen viel besser geeignet sind zur Design-Unterstützung als Sammlungen von Antworten auf antizipierte Design-Probleme.

Die genannten Hintergründe zum Bereich DR haben an vielen Stellen Bezugspunkte zu der entwickelten Diskussionsunterstützung. Auch dort wird von verteilt vorliegender Expertise ausgegangen, von der Bedeutung von Argumenten sowie den unterschiedlichen Perspektiven auf den Designprozess (Moran & Carroll, 1996). Bei Fischer, Lemke et al. (1996) wird deutlich, dass unterschiedliche Sichten sinnvoll sind, dass aber die Verknüpfung dieser das entscheidende Moment darstellt. Design-Situationen sind gekennzeichnet durch Unsicherheit, Konflikte und einmalige Fragestellungen und Problemstellungen (Fischer, Lemke et al., 1996). Die schwersten Beweggründe für Dispute und Missverständnisse sind oft „festgefahrene“ Positionen der Beteiligten (Stefik et al., 1987), die einhergehen mit nicht ausgesprochenen Voraussetzungen oder Ansichten sowie unausgesprochenen Kriterien. Nach Fischer, Lemke et al. (1996) ist Design nicht nur eine kontinuierliche Problemlösung sondern auch Problemfindung. Die Problemfindung bei der Diskussion und Wissenskonstruktion besteht z.B. in der Aufdeckung von Widersprüchen, Missverständnissen oder unterschiedlichen Interessen.

3 SYSTEME AUF BASIS VISUELLER SPRACHEN

In diesem Kapitel werden Systeme vorgestellt, die visuelle Sprachen verwenden und für kooperative Situationen ausgelegt sind. In diesem Zusammenhang werden zu Beginn die Basis-Systeme erläutert, auf denen die Entwicklung der Diskussionsunterstützung mit dem DiscBoard aufsetzt. Anschließend werden weitere Systeme erläutert, die visuelle Sprachen als Kommunikationsmedien verwenden. In diesem Kontext werden auch Systeme zur Kreativitätsunterstützung mit angesprochen, die unter Umständen nur für die Benutzung durch eine einzelne Person entwickelt wurden. Das Kapitel schließt mit einer Synopse der erläuterten Systeme. Das System DiscBoard selbst sowie der FreeStyler werden dann später im zweiten Teil dieser Arbeit vorgestellt.

3.1 Architektur der Basis-Systeme für die Diskussionsunterstützung

Das System zur Diskussionsunterstützung DiscBoard setzt auf drei verschiedenen Teilsystemen auf: MatchMaker (Hoppe & Tewissen, 1994; Tewissen, 1996; Mühlenbrock, Tewissen & Hoppe, 1997; Gaßner, Tewissen et al., 1998), Dalis (Mühlenbrock, Tewissen & Hoppe, 1997; Mühlenbrock, M. & Loesch, A., 1998; Gaßner, Tewissen et al., 1998) und CardBoard (Tewissen, 1996, Mühlenbrock, Tewissen & Hoppe, 1997; Gaßner, Tewissen et al., 1998; Hoppe, Gaßner et al., 2000), die in den folgenden Unterkapiteln erläutert werden. Während es sich bei MatchMaker und CardBoard um eine C++-Entwicklung handelt, ist das Dalis-System in Prolog implementiert. Die Kommunikation zwischen den drei beteiligten Systemen basiert in allen Fällen auf TCP/IP. MatchMaker und CardBoard sind für Windows-Plattformen entwickelt worden. Dalis ist an Quintus-Prolog gebunden, das hier unter UNIX eingesetzt wird[7]. Die Kombination aller drei Systeme zu einem Gesamtsystem bewirkt eine deutliche Zunahme der Komplexität, die für den alltäglichen Einsatz nicht unbedingt erwünscht ist. Insgesamt steht deshalb bei dieser Entwicklung nicht die Architektur im Vordergrund, sondern die Möglichkeiten, auf visuellen Sprachen zu operieren.

[7] Zum Zeitpunkt der Dalis-Entwicklung war Quintus-Prolog unter UNIX die geeignetste Prolog-Implementierung unter der Voraussetzung, dass TCP/IP verwendet werden sollte.

Die Grundfunktionalität der drei Basis-Systeme war kein Entwicklungsgegenstand dieser Arbeit, soll im Weiteren aber trotzdem kurz erläutert werden. MatchMaker ist mittlerweile reimplementiert worden in einer Java-Version (Jansen, Pinkwart & Tewissen, 2001; Jansen, 2002), die weitgehend andere Synchronisationskonzepte verfolgt. Diese neue Version wird aber erst im zweiten Fallbeispiel dieser Arbeit, dem FreeStyler, verwendet.

3.1.1 MatchMaker

MatchMaker (Hoppe & Tewissen, 1994; Tewissen, 1996; Mühlenbrock, Tewissen & Hoppe, 1997; Gaßner, Tewissen et al., 1998) organisiert die Synchronisation von Userinterface-Objekten mehrerer verteilter Applikationen. Diese Synchronisation wird im Weiteren auch als Kopplung bezeichnet.

Mit MatchMaker wird eine replizierte Architektur realisiert, in der eigenständige Applikationen temporär synchronisiert werden. Die Synchronisation beruht auf der Verteilung von Userinterface-Events, die dann jede Applikation für sich ausführt. Dafür stellt MatchMaker eine Funktionsbibliothek für Userinterface-Objekte zur Verfügung, die die Koppelungsfunktion anbieten. Prinzipiell ist es durch diese Art der Kopplung über Events möglich, verschiedenartige Objekte zu synchronisieren. Für das jeweilige Objekt muss jeweils definiert sein, wie der Event interpretiert werden soll.

Eine Replikation besteht hier in der Nachbildung der Inhaltsdaten, die in der Kooperation erarbeitet werden. Die replizierte Architektur, die durch MatchMaker aufgebaut wird, hat den Vorteil, dass die Daten, die die Arbeitsinhalte bilden, wirklich für jede Applikation vorhanden sind. So kann jede der beteiligten Applikationen beendet werden, ohne dass es bei den anderen zu einem Datenverlust kommt. Außerdem bewirkt sie eine gewisse Sicherheit im Fall von Systemsabstürzen. Die Replikation ist in einer Client/Server-Architektur realisiert. MatchMaker organisiert darin die Kommunikation und speichert dafür Daten darüber, wo und welche Objekte gekoppelt sind. Die konkreten Daten der Objekte liegen aber nur in den Applikationen, den Clients vor.

Anhand der MatchMaker-Architektur, soll noch eine weitere Eigenschaft erläutert werden und zwar die partielle Kopplung. Das bedeutet, dass nur Teile einer Applikation gekoppelt werden. So ist es beispielsweise möglich, private und öffentliche Arbeitsbereiche zu erstellen. Abbildung 5 zeigt ein Beispiel für eine partielle Kopplung. Nur die gekoppelten Bereiche sind für andere sichtbar. Bei der Diskussionsunterstützung werden gesamte Workspaces synchronisiert. Mit MatchMaker ist es möglich, mehrere Workspaces mehrerer Applikationen zu koppeln. Dies ist in Abbildung 5 durch das Paar roter und blauer Pfeile angedeutet.

Über MatchMaker verläuft außerdem die Kommunikation der Applikationen mit einem weiteren Basis-System, dem Dalis-System. Dort wird für die Diskussionsunterstützung das Diskussionsmodell verwaltet.

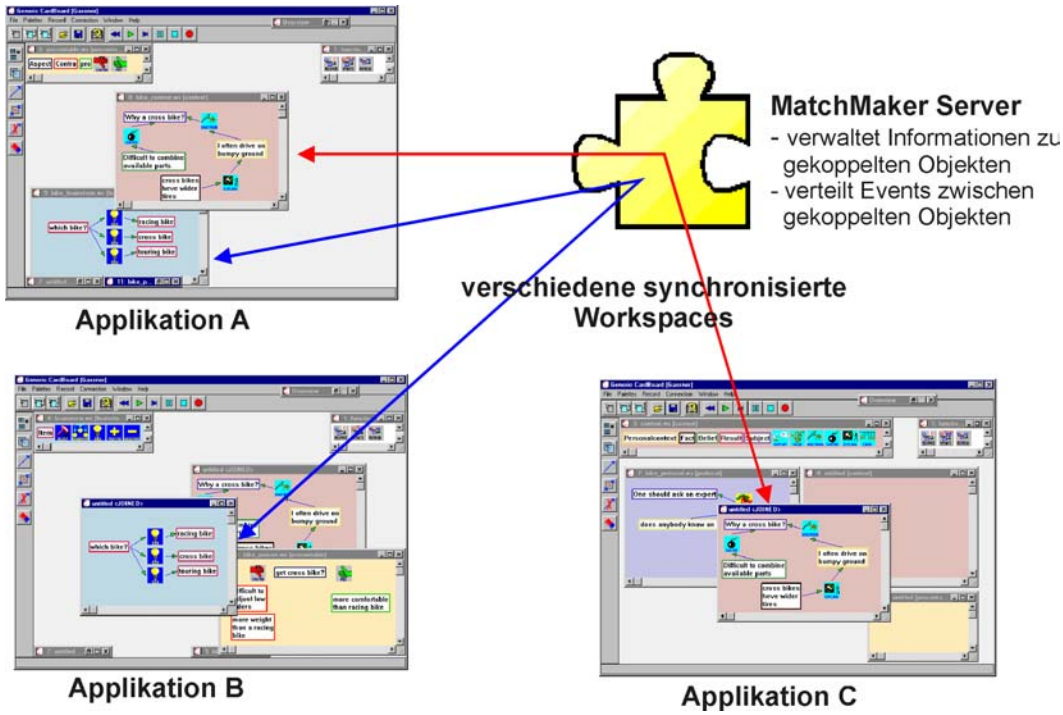


Abbildung 5

Der MatchMaker Server synchronisiert Objekte unterschiedlicher Applikationen. Im dargestellten Beispiel wird jeweils ein rosafarbener Workspace der Applikation A und C sowie ein blauer Workspace der Applikation A und B synchronisiert.

3.1.2 Dalis

Dalis ist das zweite eingebundene Teilsystem (Mühlenbrock, Tewissen & Hoppe, 1997; Mühlenbrock & Hoppe, 1999; Mühlenbrock, M. & Loesch, A., 1998; Gaßner, Tewissen et al., 1998). Es führt immer ein Monitoring der über MatchMaker angebundenen Applikationen durch. In dieser Grundfunktion hält Dalis den Ist-Zustand der Applikationen vor. Das beinhaltet die Zustände von Karten, Links und Workspaces. Jede Änderung des Zustandes eines Workspaces oder eines Objektes einer visuellen Sprache wird an Dalis übermittelt.

Zwischen den drei Teilsystemen ist eine Schnittstelle implementiert, in der Daten über die vorgegebenen Kommunikationsprimitive create, modify und delete ausgetauscht werden können. Diese Primitive beziehen sich auf die Objekte, die während der inhaltlichen Arbeit entstehen, also auf Objektinstanzen der visuellen Sprachen, sowie auf Workspaces. Beispielsweise bewirkt das Erstellen eines Objektes einer visuellen Sprache, eine create-

Mitteilung an MatchMaker. Von dort wird diese Mitteilung an alle gekoppelten Applikationen sowie an Dalis weitergeleitet.

Auf Grundlage des Ist-Zustandes, der in der Datenbasis von Dalis vorliegt, und den aktuellen create-, modify- und delete-Messages können in Dalis Reaktionen auf Zustände bzw. deren Änderung implementiert werden. Beispielsweise können Operationen in den Applikationen aus einer domänenspezifischen Modellierung in Dalis angestoßen werden. Dies bildet auch den Hintergrund, warum Dalis später in die Diskussionsunterstützung eingebunden wird, denn so soll das Diskussionsmodell zur Verfügung gestellt werden. D.h. die eigentliche Modellierung, die auch zur Unterstützung des Diskussionsprozesses verwendet wird, findet in den Dalis-Komponenten statt. Das Diskussionsmodell und die damit verbundene operationale Semantik ist wie Dalis in Prolog implementiert.

Abbildung 6 erweitert Abbildung 5 um die Darstellung der Kommunikation mit Dalis. Während die Synchronisation der Applikationen alleine von MatchMaker organisiert wird, werden create-, modify- und delete-Informationen auch zwischen MatchMaker und Dalis ausgetauscht.

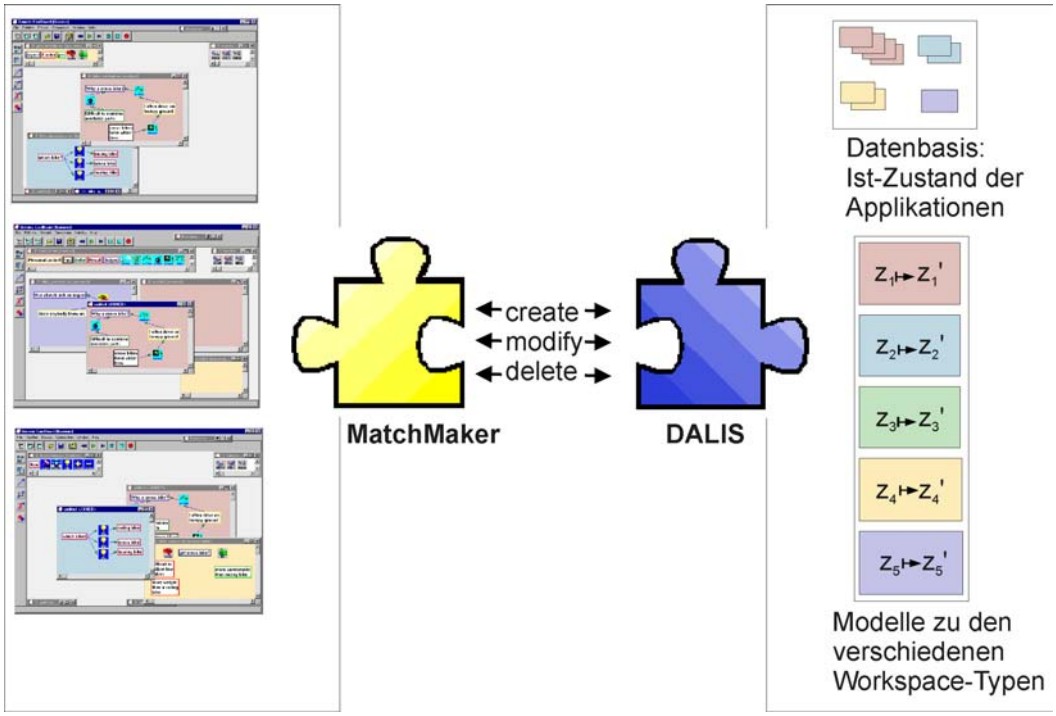


Abbildung 6

Dalis und MatchMaker kommunizieren über die standardisierten Primitiven create, modify und delete. In der Dalis-Datenbasis wird der Ist-Zustand der visualisierten Objekte abgelegt.

Die in Abbildung 6 angedeutete Dalis-seitige Modellierung ist bereits eine Vorausschau auf das Beispiel der Diskussionsunterstützung. Darin sollen die bunten Rechtecke oben rechts jeweils Gruppen von Datensätzen symbolisieren, die einen Workspace

beschreiben. So sind vier rosafarbene Rechtecke eingeblendet für die vier entsprechenden Workspaces der drei Applikationen auf der linken Seite der Abbildung. Die blauen, gelben und lilafarbenen Rechtecke symbolisieren die andere Workspaces der Applikationen. Die mit $z \rightarrow z'$ beschrifteten Rechtecke auf der rechten Seite symbolisieren Reaktionen auf den jeweiligen Zustand eines Workspaces.

3.1.3 CardBoard

Das System CardBoard wird durch seine Entwicklungsstadien hindurch in Tewissen (1996), Mühlenbrock, Tewissen und Hoppe (1997), Gaßner, Tewissen et al. (1998) und Hoppe, Gaßner et al. (2000) vorgestellt. Es folgt deshalb nur eine kurze Erläuterung.
Die Bezeichnung CardBoard bezieht sich auf das Rahmensystem, ohne eine bestimmte Einschränkung auf eine visuelle Sprache und damit auf eine Anwendung. Die spätere Spezialisierung auf die Diskussionsunterstützung wird mit der Bezeichnung DiscBoard abgegrenzt.

Das Rahmensystem CardBoard

Das System CardBoard (Abbildung 7) bildet den Rahmen für die Arbeit mit visuellen Sprachen, die man sich hier erst einmal als beschriftbare Karten und Verbindungen vorstellen kann. Die visuellen Sprachen liefern Objekte, die in Workspaces gelegt, strukturiert und bearbeitet werden.

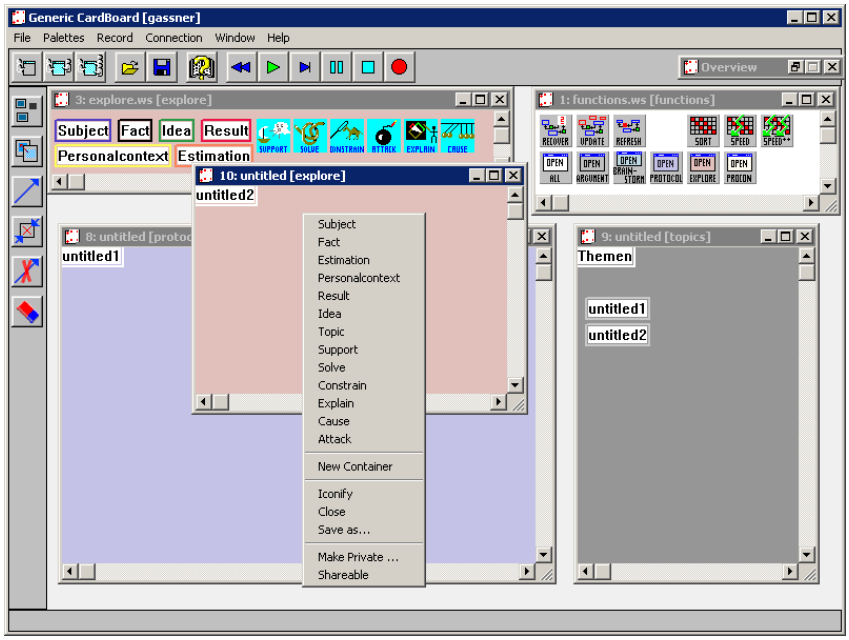


Abbildung 7

Das Rahmensystem CardBoard mit mehreren Workspaces, dem Bearbeitungs-Menü und zwei Paletten.

Die CardBoard-Oberfläche bietet zwei Standard-Paletten an und zentral einen freien Raum in dem Workspaces angelegt werden können (Abbildung 7). Über die obere, waagerechte Palette können neue Workspaces geöffnet werden (Tabelle 3). Außerdem können die Aktionen, die in den Workspaces ausgeführt werden, aufgezeichnet werden. Das CardBoard bietet dafür eine „Record&Replay“-Funktion an (Tabelle 3).

	Buttons zum Öffnen von Workspaces: „private“ (links), „shared“ (mittig), „joint“ (rechts)
	Das CardBoard bietet die Möglichkeit, die Ereignisse eines bestimmbaren Workspaces aufzuzeichnen.

Tabelle 3

Funktionen des CardBoards zum Öffnen von Workspaces und zur Steuerung des „Record&Replay“.

Das CardBoard stellt drei Arten von Workspaces zur Verfügung:

- „private“ Workspaces, auf die keine weitere Person Zugriffsrechte hat,
- „shared“ Workspaces als Kommunikationskanäle zu einer konkreten Person bzw. Applikation,
- „joint“ Workspaces als Kommunikationskanäle zwischen einer Gruppe von Personen.

Während der Arbeit können im Prinzip beliebig viele „private“ oder „shared“ Workspaces geöffnet werden. Arbeiten mehrere Personen zusammen in einem Workspace, so können sie parallel Karten und Verbindungen anlegen, verändern oder löschen. Diese Eingaben sind sofort für alle Beteiligten sichtbar. Von diesen wird deshalb ein Mindestmaß an Koordination verlangt, so dass nicht eine Person Beträge anderer verhindert.

Die linke, senkrechte Palette bietet allgemeine Funktionen zur Bearbeitung der Objekte in den Workspaces an, also zur Bearbeitung der Elemente der visuellen Sprachen. Diese Funktionen sind unabhängig von der konkreten Sprache. Sie werden in Tabelle 4 näher erläutert.

	Einzelne Karten können einer Auswahl hinzugefügt werden.		Vorgesehen: Generieren ganzer Relationen.
	Es kann definiert werden, ob Karten zwischen Workspaces gleichen Typs verschoben oder kopiert werden sollen.		Vorgesehen: Löschen von Links.
	Einzelne Links können menügesteuert generiert werden.		Löschen von Karten.

Tabelle 4

Funktionen des CardBoards zur Bearbeitung der Elemente der visuellen Sprachen.

Parametrisierung durch visuelle Sprachen

Die visuellen Sprachen sind Parameter des Rahmensystems CardBoard, das die allgemeinen Funktionen zur Nutzung der visuellen Sprachen zur Verfügung stellt. Die Sprachobjekte sollen schnell und flexibel definierbar sein. Deshalb werden die Objekte der Sprache nicht systemintern definiert, sondern extern spezifiziert. So können auch Anwender Sprachen definieren, ohne das System selbst erweitern zu müssen. Das bezieht sich allerdings nur auf die rein syntaktische Definition der Sprachen.

In einer Initialisierungsdatei wird festgelegt, welche Sprachen geladen werden sollen. Abbildung 8a zeigt den Inhalt einer solchen Datei. Jede erwähnte Sprache referenziert die Datei, in der sie definiert ist. Jeder Sprache wird außerdem ein Name zugewiesen, der später verwendet wird, um Menüs zu generieren. Abbildung 8b zeigt ein Menü, das geöffnet wird, wenn ein neuer Workspace angefordert wird. Darin werden die Namen der Sprachen aufgenommen, die in der Initialisierungsdatei aufgeführt sind.

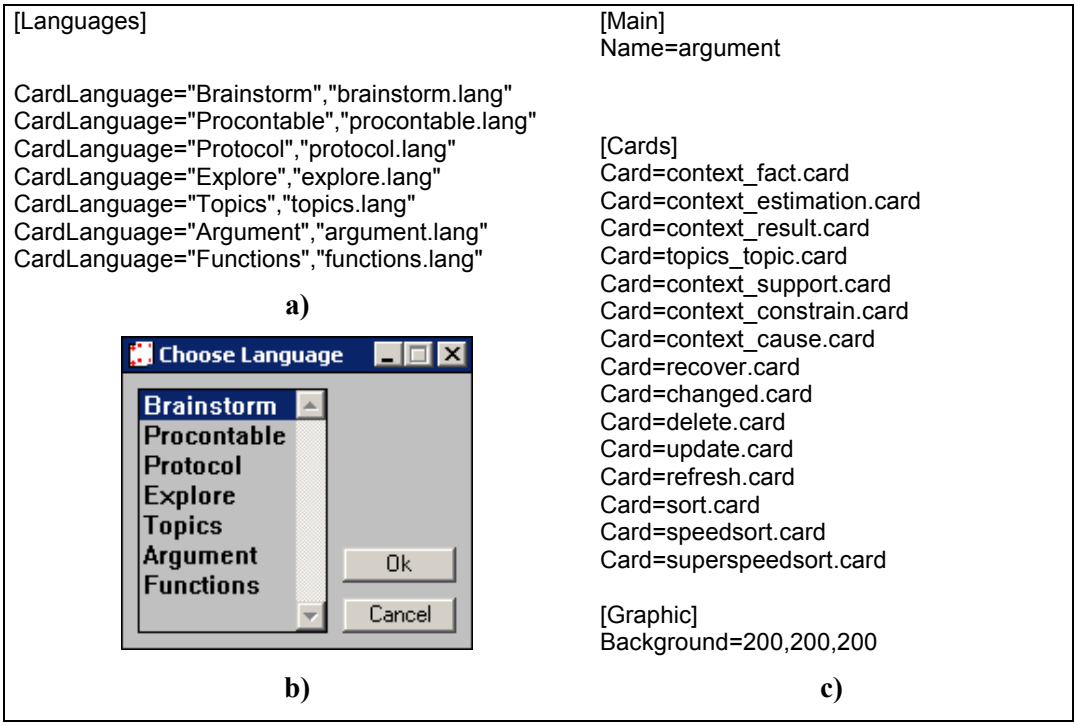


Abbildung 8

a) Das CardBoard wird durch extern definierte Sprachen parametrisiert. b) Über ein Menü wird ausgewählt, mit welcher visuellen Sprache gearbeitet werden soll. c) Die Definition einer visuellen Sprache in einer Datei.

Abbildung 8c zeigt die Definition einer visuellen Sprache. Diese Datei verweist wiederum auf die Definitionen der einzelnen verwendeten Kartentypen, die wieder in separaten Dateien definiert werden. Die Einträge des Menüs zum Erstellen von Karten in

einem Workspace werden auf Basis dieser externen Spezifikation generiert. So ein Menü ist in Abbildung 7 zu sehen.

3.1.4 Die visuellen Sprachen im CardBoard

Definition

Es wurde argumentiert, dass Netzstrukturen eine kognitionstheoretisch wie technisch geeignete Grundlage sind, um visuelle Sprachen für kognitive Tools einzusetzen. Im Rahmensystem CardBoard werden deshalb visuelle Sprachen zur Verfügung gestellt, mit denen Netzstrukturen erstellt werden können. Formal handelt es sich dabei um Graphen. Für die Nutzerinnen und Nutzer stellen die Schaubilder jedoch einfach Diagramme dar. Durch die Art und Weise ihrer Verwendung in kooperativen Arbeitsszenarien und deren flexible Bearbeitung wirken diese Diagramme als Medien in einem Kommunikationsprozess.

Die visuellen Sprachen bieten jeweils Mengen von Objekten als graphische Hilfsmittel an, um Diagramme zu erstellen. Die Primitive der visuellen Sprachen, die Objekte, werden auch als Karten bezeichnet, resultierend aus einer ursprünglich psychologischen Anwendung (Plötzner, Hoppe, Fehse, Nolte & Tewissen, 1996). Die Kartenmetapher erläutert aber gleichzeitig die Art des Einsatzes, nämlich kurze Notizen wie auf Karteikarten festzuhalten. Es existieren zwei Arten von Karten:

- **Inhaltskarten**, die als Container für freie Eingaben dienen und
- **Konnektorkarten**, mit denen die Inhaltskarten verbunden werden können.

Die angebotenen Karten werden auch als Kartentypen bezeichnet, deren Unterschied typischerweise auch in der graphischen Darstellung sichtbar ist.

Die Konnektorkarten sind Bestandteil einer Kante in einem Graphen, bzw. Bestandteil einer Relation zwischen Inhaltskarten. Um eine Kante zu vervollständigen, müssen neben einer Konnektorkarte noch die Verbindungen zu den Inhaltskarten bestimmt werden. Dafür werden in der Applikationsoberfläche zwischen einer Konnektorkarte und den in Relation stehenden Inhaltskarten Pfeile eingefügt: die Links. Durch die Stelligkeit der Relationen inklusive der Richtung der Links wird die Struktur der Graphen und damit die Syntax der Sprache bestimmt, wobei prinzipiell n-stellige Relationen dargestellt werden können (entsprechend dem Konzept eines Hypergraphen). Die Interaktion zur Erstellung von Links wird über die Konnektorkarten gesteuert. Diese definieren Anzahl, Typ und Richtung der Links für einen Kartentyp.

Über den Typ einer Karte oder auch den eines Links können den jeweiligen Objekten bestimmte operationale Semantiken zugeordnet werden.

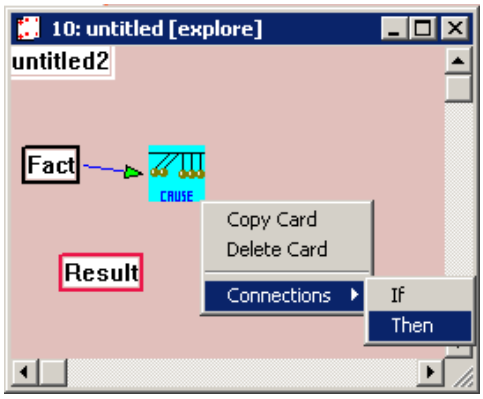
Sowohl die Inhaltskarten als auch die Konnektorkarten sind so definierbar, dass sie während der Kommunikationsprozesse editierbar sind.

Die Spezifikationen der Kartentypen befinden sich in externen Files, die beim Start des Rahmensystems CardBoard geladen werden. Im vorangegangenen Abschnitt wurde bereits erläutert, wie die Zugehörigkeit der Sprachprimitive zu einer visuellen Sprache festgelegt wird. Abbildung 9a zeigt nun ein Beispiel wie eine Karte, in diesem Fall eine Konnektorkarte, definiert wird.

```
[Description]

SemanticType=cause
MenuEntry="Cause"
Type=connector_card
Style=bitmap
Content="cause.bmp"
Shape=CS_NONE
ReadOnly=true
ShapeColor=253,253,89
Link="input","If",-1,"",0,0,0
nLink="output","Then",1,"",0,0,0
```

a)



b)

Abbildung 9

a) Die Spezifikation einer Karte wie sie in einer Datei durchgeführt wird.
b) Interaktion mit einer Konnektorkarte zur Erstellung von Links.

Über „Tags“ werden dort Typ, Aussehen und Bestandteile der Karte festgelegt und es findet sich dort auch die Unterscheidung der Konnektor- und Inhaltskarten wieder: Mit Hilfe des „Type-Tags“ wird zwischen „connector_card“ und „content_card“ unterschieden. Mit dem „Style-Tag“ wird definiert, wie der Inhalt des „Content-Tags“ zu interpretieren ist. Mit dem „Content-Tag“ kann sowohl auf eine Datei verwiesen werden, aus der der Inhalt geladen werden soll, als auch direkt ein Inhalt bestimmt werden. Über den Verweis auf eine Datei ist es möglich, Graphiken einzubinden, und zur Gestaltung von Symbolen zu verwenden. Das „Content-Tag“ könnte auch auf dynamische Komponenten verweisen, die eine spezifische Interaktion anbieten. Die Spezifikation einer Konnektorkarte beinhaltet auch die Definitionen für die Links („Link-Tag“), bestehend aus dem Namen des Links, seinem Eintrag im kontextsensitiven Popup-Menü der Konnektorkarte, seiner Richtung in Bezug auf die Konnektorkarte, dem Label des Links und seiner Farbe. Im gezeigten Beispiel sind keine Label angegeben. „Shape“ und „ShapeColor“ bestimmen das Aussehen der Karten. Insbesondere bei Inhaltskarten bilden sie die wesentlichen Merkmale an denen man den Kartentyp in der graphischen Darstellung erkennen kann.

Das „ReadOnly-Tag“ und das „Movable-Tag“ bestimmen Interaktionsmöglichkeiten auf Ebene des Interfaces. Das „SemantikType-Tag“ wird für die Zuordnung der operationalen Semantik verwendet, die von Dalis gesteuert wird.

Abbildung 9b zeigt die Interaktion mit einer Konnektorkarte, um Verbindungen zu Inhaltskarten herzustellen. Dafür bietet eine Konnektorkarte, hier die „Cause“-Karte, ein Menü an, unter dem der Eintrag „Connections“ zu den möglichen Verbindungstypen überleitet. In dem dargestellten Beispiel werden die Verbindungen „If“ und „Then“ angeboten. Mit der Wahl des Verbindungstypen wird automatisch auch die Richtung des Links zu der betroffenen Inhaltskarte festgelegt. In der Abbildung sind „Fact“ und „Result“ die zu verbindenden Inhaltskarten, deren Inhalt hier noch nicht weiter spezifiziert wurde. In der Abbildung sind beide Inhaltskarten in der Form dargestellt, in der sie generiert werden. Der „default“-Text stellt eine zusätzliche Hilfe dar, um den Typ der Karte zu erkennen. Im vorliegenden Ansatz werden mit visuellen Sprachen immer semi-formale Repräsentationen erstellt. Ein wesentliches Kriterium semi-formaler Repräsentationen besteht in der Mixtur von Computer-interpretierbaren und nicht – interpretierbaren Bestandteilen eines Ausdruckes ([dynagloss1], [dynagloss2]). Ein typisches Beispiel dafür sind Hypertexte, bei denen die Links formal sind und interpretiert werden, die Inhalte aber nicht. Auch E-Mails sind ein Beispiel für semi-formale Repräsentationen. Normalerweise haben Nutzerinnen und Nutzer keinen Einfluss auf die Interpretation der formalen Bestandteile einer Darstellung, wohl aber auf die Inhalte, also die nicht-formalen Bestandteile. Der Vorteil von semi-formalen Darstellungen besteht gerade darin, individuelle nicht formalisierte Inhaltseingaben mit formalen interpretierbaren Elementen zu verbinden.

Über die Darstellungen mit visuellen Sprachen kann auf diverse strukturorientierte und kartentyporientierte Merkmale, wie Art von Verbindungen und Muster sowie auf Ort, Nähe und Anzahl und von Karten und Verbindungen zugegriffen werden. Die so gewonnenen Daten bilden den computerinterpretierbaren bzw. formalen Bestandteil der Repräsentation. Diese Daten können genauso für Interpretationen ausgenutzt werden wie sie Auslöser für Operationen sein können. Gleichzeitig können die unformalisierten Inhalte frei mit dem System bearbeitet werden.

Die Kartenhierarchie im CardBoard

Eine Entscheidung beim CardBoard-System bestand darin, nicht nur die Inhaltskarten, sondern auch die Konnektorkarten als Karten zu implementieren. Dies sollte zu einer flexiblen Möglichkeit bei deren Gestaltung führen. Außerdem waren so Hyperlinks

graphisch gut umsetzbar. In der CardBoard-Implementierung werden entsprechend sowohl die Inhaltskarten als auch die Konnektorkarten von der gleichen Klasse abgeleitet (Abbildung 10). Die CardBoard-Implementierung stellt als übergeordnete Klasse die „BaseCard"-Klasse zu Verfügung. Davon abgeleitet wird sowohl die „ContentCard"-Klasse als auch die „ConnectorCard"-Klasse. Weiter verfeinerte Ausprägungen werden in abgeleiteten Klassen definiert.

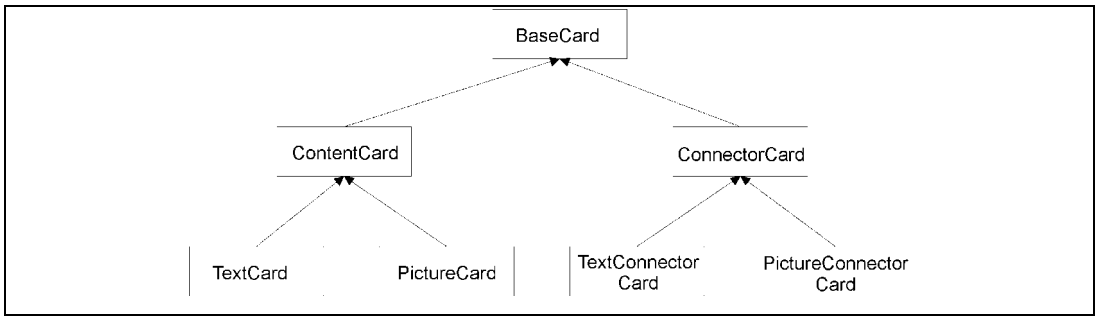


Abbildung 10

Die Klassenhierarchie der Karten im CardBoard.

Die „BaseCard"-Klasse stellt allgemeine Funktionen bereit, die die Interpretation der externen Kartenspezifikationen leisten und z.B. Drag&Drop und delete-Aktionen steuern. Die Funktion der „ContentCard"-Klasse und der „ConnectorCard"-Klasse besteht darin, auf Basis der in der Spezifikation referenzierten oder definierten Inhalte das jeweilige Oberflächenobjekt aufzubauen. Dazu gehört auch, die notwendigen Popup-Menüs für die Karten zu generieren und auf die spätere Interaktion zu reagieren. Das gilt insbesondere für die Konnektorkarte, für die die „ConnectorCard"-Klasse den Umgang mit den Links implementiert.

3.2 Systeme im Vergleich

Obwohl die „face-to-face"-Kooperation nach wie vor die üblichste Art der Zusammenarbeit ist (Lernen in Klassen, Meetings, Debatten, etc.), existieren heute wenige Systeme, die explizit für eine Diskussionsunterstützung in einem „face-to-face"-Szenario implementiert sind, die also versuchen Computer in dieses Szenario zu integrieren.

Anders als in verteilten Szenarios wird in diesem Kontext direkt deutlich, dass die Art der Hilfestellung durch Software auf die Externalisierung, Repräsentation, (Re-)Strukturierung und Erstellung von Artefakten abzielen muss und nicht darauf, die Kommunikation selbst zu ermöglichen. Das bedeutet, dass die Darstellungsmittel und Arbeitsmethoden eine entscheidende Rolle bei der Diskussionsunterstützung tragen.

Eine andere These dieser Arbeit besteht darin, dass die nachhaltige Nutzung von Diskussionsinhalten nicht ausreichend in die Systemgestaltungen mit einfließt. Dies ist in sofern fatal, als dass aus dem Kontext des Wissensmanagements abgeleitet wird, dass die nachhaltige Dokumentation einen wesentlichen Mehrwert bei der Nutzung einer Diskussionsunterstützung darstellt. Durch die nachhaltige Nutzung soll Wissen re-integriert werden, „neues" Wissen entstehen und „communities" aufgebaut werden.
Insgesamt werden dem gemäß drei Kriterien für den folgenden Systemvergleich ausgewählt:

- Nutzbarkeit im **„face-to-face"-Setting**.
- Verwendung von **visuellen Sprachen**, durch die Diskussionen durch kreative oder vielfältige Ausdrucks- und Darstellungsmöglichkeiten angereichert werden oder prozessorientierte Unterstützung bieten.
- **Nachhaltige Nutzung**, Wissensbildung.

Für den Vergleich werden exemplarisch Systeme herangezogen, die sich mindestens einem dieser Kriterien zuordnen lassen. Es zeigt sich, dass kaum ein System alle Aspekte auf sich vereinigt.
In Kapitel 1.5 wurden bereits drei Ebenen erläutert, auf denen „face-to-face"-Szenarios technisch unterstützt werden können: 1) Software/Tools 2) Realisierung des Szenarios 3) Organisatorische Einbettung. Diese drei Ebenen sollen im folgenden auch für den Systemvergleich herangezogen werden und können außerdem den oben genannten Kriterien zugeordnet werden: Die Beschreibungen der Tools werden jeweils auch die Nutzung der visuellen Sprachen verdeutlichen, zum Aspekt Realisierung des Szenarios wird neben technischen Ansätzen auch das Nutzungsszenario beschrieben und unter dem Aspekt organisatorische Einbettung wird die Art der Nutzung und Einbettung in Arbeitsabläufe geschildert. Die Systeme werden in alphabetischer Reihenfolge vorgestellt. Die Einordnung gemäß der „Zielrichtung" bezieht sich auf die in Kapitel 2 eingeführten Verwendungsarten von visuellen Sprachen.

3.2.1 Belvedere

- *Zielrichtung.* Lernunterstützung; Kommunikationsmedien und Wissensprodukte
- *Schwerpunkt.* Vermittlung wissenschaftlicher Argumentationsweise
- *Kriterien.* Nutzung visueller Sprachen, Verwendung im „face-to-face"-Szenario

Software/Tool. Das Ziel des Belvedere-Systems (Suthers, Weiner et al., 1995; Suthers, Toth & Weiner, 1997; siehe Abbildung 11) besteht darin, Schülerinnen und Schülern wissenschaftliches Argumentieren beizubringen.

Die Motivation besteht darin, dass bei jungen Studierenden häufig Probleme beobachtet werden können, wichtige Elemente aus einer umfangreichen Debatte herauszugreifen und relevante Argumentationskriterien zu finden. Hinzu kommt ungenügendes Grundlagenwissen in den diskutierten Bereichen und außerdem eine nicht ausgereifte Motivation, einen Inhaltsbereich zu diskutieren. Das Lernziel besteht darin, bestimmte Regeln der Argumentation zu vermitteln:

•Hypothesen erklären Daten •Hypothesen werden ohne Daten nicht akzeptiert •Mehrere Belege, die auf eine Hypothese hinführen sind besser als ein einzelnes gutes Datum •Man soll positive aber auch negative Belege finden •Man muss zusätzliche Belege finden, wenn zwei Hypothesen durch die gleichen Belege unterstützt werden.

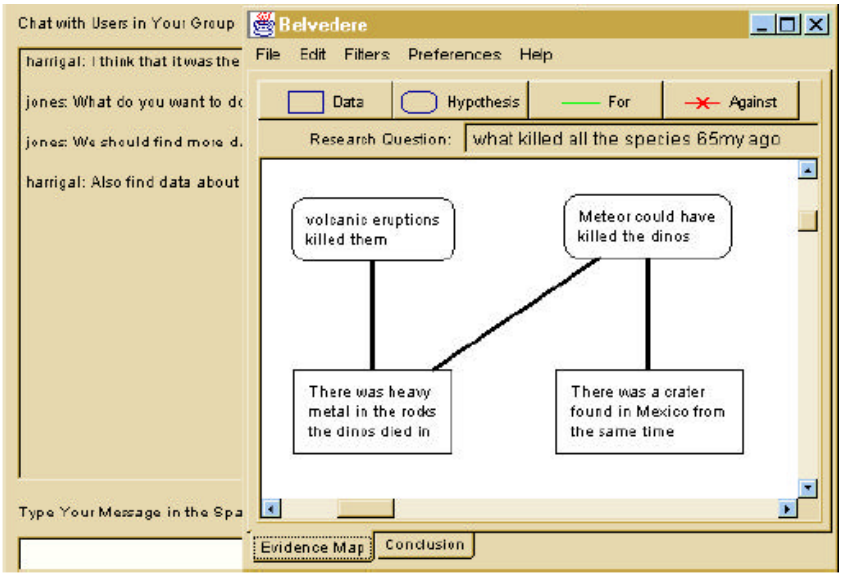


Abbildung 11

Ein „inquiry diagram“ im Belvedere-System (Suthers, 1999C).

Das Arbeitsprodukt bei Belvedere ist ein Diagramm. Eine visuelle Sprache stellt dafür Knoten und Relationen zur Verfügung, wobei sich die Knoten über ihre Form unterscheiden. Sie stellen Hypothesen oder Daten dar, die durch die Relationen zueinander in Verbindung gesetzt werden. Den Relationen können durch die Dicke der Linien sogenannte „belief levels“ zugeordnet werden. Wenn Diagramme erstellt werden, wird erwartet, dass die oben formulierten Regeln möglichst ausgeschöpft werden. Dies wird durch zwei Agenten angeleitet und unterstützt, die diese Regeln überprüfen und von Zeit zu Zeit Hilfestellungen dazu geben.

Das Belvedere-System hat bereits mehrere Entwicklungsstufen hinter sich. In der Version, die in Suthers, Toth und Weiner (1997) vorgestellt wurde, wurde vor allem auf einen großen Umfang der visuellen Sprache verzichtet. Gab es vorher drei Arten,

Diagramme aufzubauen, so gibt es darin nur noch eine. Das beruht auf Erfahrungen nach denen sich die Bedeutung der einzelnen Primitive häufig überlagerte und deshalb zu Ungenauigkeiten führte.

Realisierung des Szenarios. Belvedere 2.x (Suthers, Toth & Weiner, 1997, Suthers 1999C) stellt „shared" Workspaces für die Diagramme und ein „Chat" zur unstrukturierten Diskussion zur Verfügung. Trotz dieser scheinbaren Ausrichtung auf die örtlich verteilte Situation, ist Belvedere für den Klassenraum gedacht. Dabei arbeiten zwei oder mehr Personen miteinander, erstellen zusammen in den Workspaces ihre Diagramme, interagieren aber zusätzlich mündlich.

Organisatorische Einbettung. Der Schwerpunkt von Belvedere liegt bei der kooperativen Aufarbeitung und Ko-Konstruktion von Informationen in Form von wissenschaftlichen Argumentationsnetzen. Im Klassenraumszenario sollen weitgehend reale wissenschaftliche Probleme ergründet werden. Dazu besteht die Möglichkeit externe Materialien zu referenzieren. Auch Experimente und Analysen sollen mit in den Lernablauf integriert werden auch wenn sie nicht von dem Belvedere-System selbst begleitet werden, dass nur ein Ausschnitt des Lernens begleiten soll.

Das Belvedere-Systems wird vor allem dazu herangezogen, um zu untersuchen , ob und inwieweit Repräsentationen geeignet die Kommunikation begleiten können („representational guidance", „salience": Suthers, 1999A, 1999B, 1999C). Trotzdem wird ein technisch wie didaktisch relativ komfortables Szenario erstellt.

Weniger Anstrengungen werden darauf investiert, die jeweiligen Ergebnisse weiter zu verwerten, sie beispielsweise als Diskussionsanreiz anderen Gruppen zur Verfügung zu stellen oder beste Ergebnisse vorzuhalten. Entsprechend wird die Suche nach ergänzenden Diagrammen oder deren Integration nicht unmittelbar unterstützt oder angestrebt. Belvedere ist deshalb nicht unbedingt für länger andauernde Entwicklungen umfangreicherer Sachzusammenhänge zu verwenden. Auch bleibt unklar, wie mit größeren Diagrammen, die komplexere Domänen aufarbeiten, umgegangen wird.

3.2.2 Chips

- *Zielrichtung.* Kommunikationsmedien und Wissensprodukte
- *Schwerpunkt.* Unterstützung von Gruppenarbeit
- *Kriterien.* Nutzung visueller Sprachen, Verwendung im „face-to-face"-Szenario, Nachhaltigkeit

Software/Tool. Bei dem Chips-System (Wang & Haake, 2000; Abbildung 12) handelt es sich um ein kooperatives Hypermediasystem, dass begleitend zu Kommunikations-

situationen eingesetzt werden soll. Es baut teilweise auf dem Sepia-System auf (wird im Folgenden erläutert), erweitert dieses jedoch maßgeblich und folgt deutlich unterschiedlichen Konzepten. Chips soll genauso bezüglich individueller Anforderungen adaptierbar sein, wie in Bezug auf die Anforderungen einer Gruppe. Sowohl der gemeinsame Kommunikationsbereich („shared information space“) als auch Prozesse sollen definiert oder angepasst werden können. Damit sollen vier zentrale Anforderungen erfüllt werden: 1) Design des gemeinsamen Kommunikationsbereiches 2) Anpassung der Kooperationsmittel 3) Zugriffskontrolle 4) Anpassung des gemeinsamen Userinterfaces.

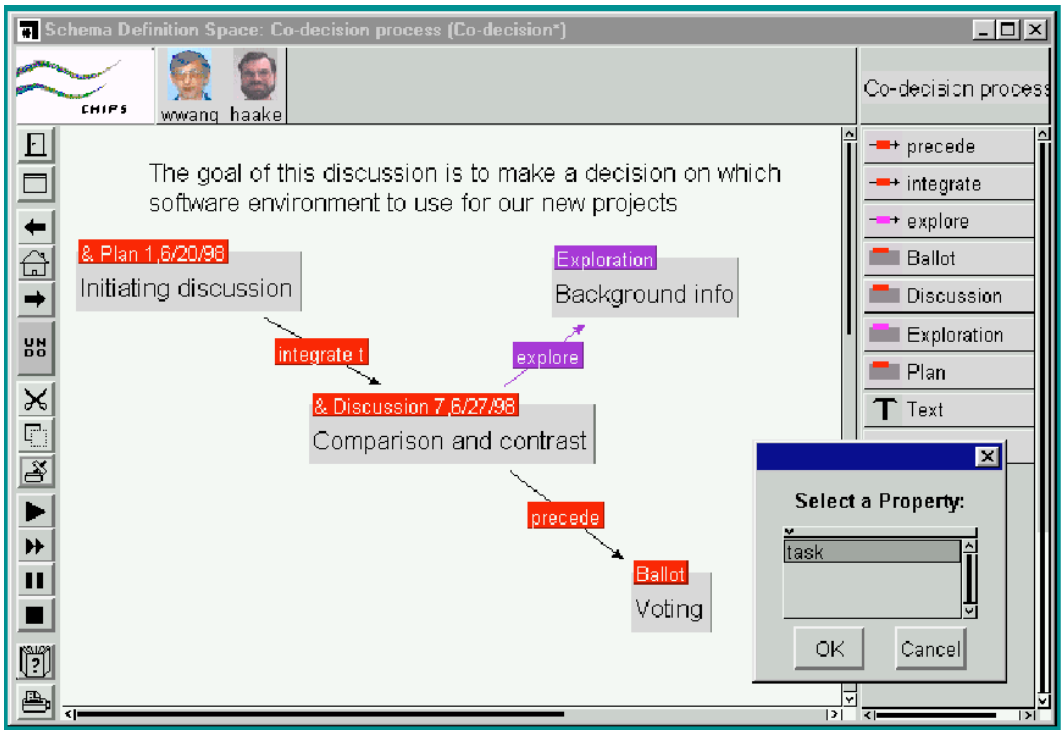


Abbildung 12

Darstellung eines Workflows in Chips. Rechts: Palette mit den Objekten der visuellen Sprache. Links: Allgemeine Funktionen.

„Activity spaces“ zeigen die einzelnen potentiell vernetzten Seiten an und stellen das adaptierbare Interface zur Verfügung. Die Seiten enthalten wiederum Knoten, Links und Multimedia-Objekte, wobei die Links an Knoten gebunden sind und auf andere Seiten verweisen: Organisationale Links unterliegen strukturellen Constraints, referenzielle Links entsprechen allgemeinen Hyperlinks. Zusätzlich können Relationen zwischen Knoten einer Seite dargestellt werden.

Einem „activity space“ wird ein Userinterface zugeordnet, über das Kontroll- und Browserfunktionen zugänglich sind und über das die Kooperationsparameter während einer Session festgelegt werden können. Organisationseinheiten (z.B. Personen oder Gruppen) können Rollen zugeordnet werden, an die Zugriffsrechte gebunden sind.

Auch Prozesse werden in den „activity spaces“ dargestellt bzw. definiert (siehe Abbildung 12). Dafür können beispielsweise Aufgabenknoten über Links verbunden werden, die um Attribute für Zeit und Transitionen erweitert werden.

Anpassbar sind die Interfaces, die eine visuelle Sprache zur Verfügung stellen, indem neue Objekte durch User definiert werden. Diese werden ebenfalls in die Palette mit aufgenommen, um so Vorlagen für bestimmte Arbeitsweisen anzulegen, die später wieder als Templates verwendet werden können. Weiterhin sind die Objekte selbst durch User anpassbar, indem ihre Attribute geändert werden.

Realisierung des Szenarios. Chips soll sowohl in synchronen wie in asynchronen, in verteilten wie „face-to-face“-Situationen Unterstützung liefern.

Organisatorische Einbettung. Ähnlich wie Sepia scheint Chips ein weitgehend geschlossenes System zu sein. Schnittstellen zu anderen „Welten“ wie beispielsweise zum Internet, XML und XSLT werden zumindest in Wang und Haake (2000) nicht thematisiert. Chips scheint deshalb in erster Linie als wissenschaftliche Versuchsumgebung entwickelt worden zu sein.

Durch die Zugriffskontrolle und die Definitionsmöglichkeiten in Bezug auf die Gruppenzusammensetzung sind komplexe Organisationsstrukturen bestimmbar. Undeutlich bleibt jedoch, wie die Erstellung der Navigationsnetze über kleine Teilaspekte hinaus unterstützt wird, inwieweit also eine Unterstützung im Sinne eines „corporate memory“ vorgesehen ist.

Die Anpassung der Interfaces an die jeweiligen Bedürfnisse ist weitgehend, gleiches gilt auch für die Prozessdefinitionen. Es ergibt sich eine anpassbare, domänenunabhängige Arbeitsumgebung. In Bezug auf eine „computational semantic“ der Objekte sind die User allerdings weitgehend auf ein Entwicklerteam angewiesen, das gewünschte Operationen implementiert.

Die Prozessunterstützung kann auch an einen Workflow bezüglich der Inhaltsknoten gebunden werden. Auch darin hebt sich dieses Tool von anderen Entwicklungen ab. Ein wenig fraglich erscheint es, ob Projektplanungsaufgaben nicht teilweise sinnvoller in tabellarischer Weise dargestellt werden könnten, als in Graphen.

3.2.3 CoLab: Cognoter/Argnoter

- *Zielrichtung.* Kommunikationsmedien und Wissensprodukte
- *Schwerpunkt.* Cognoter: Kooperatives Vorbereiten von Präsentationen; Argnoter: Argumentation zu konkurrierenden Vorschlägen
- *Kriterien.* Nutzung visueller Sprachen, Verwendung im „face-to-face“-Szenario

Software/Tool. Argnoter und Cognoter (Stefik et al., 1987) wurden als Tools des CoLab („collaborative laboratory“) vorgestellt, das bei XEROX PARC entwickelt und realisiert wurde (siehe Abbildung 13).

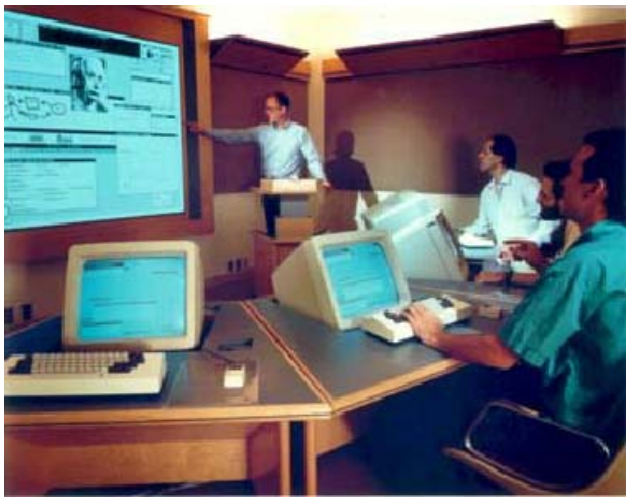


Abbildung 13

Aufnahme einer CoLab-Installation [CoLab].

Cognoter: Beim Vorbereiten von Präsentationen wird davon ausgegangen, dass zuerst Ideen gesammelt werden (Brainstorming), die dann sortiert (Organisation) und überarbeitet (Evaluation) werden müssen. Die Grundideen für eine Präsentation werden in möglichst kurzen Beiträgen als Knoten von Graphen notiert. Für diese Phase gibt es hauptsächlich Verhaltenshinweise, die Bezug nehmen auf die Brainstormingmethode. In der Organisationsphase werden gerichtete Kanten angelegt, um eine Reihenfolge zu definieren, in der die Aspekte präsentiert werden sollen. Auch Gruppierungen sind möglich, in denen verschiedene Punkte unter einem Aspekt zusammengefasst werden. Jede Ideengruppe wird in einem eigenen Fenster angezeigt. Bei der Evaluation wird die Struktur der Beiträge überarbeitet und ergänzt.

Argnoter: Vorbereitete Vorschläge z.B. für Designs oder Programmentwürfe werden von Mitgliedern einer Arbeitsgruppe eingebracht. Die Vorbereitung gehört zu den „Spielregeln“ bei der Verwendung des Tools. Es wurde festgestellt, dass in so vorbereiteten Sitzungen stärker diskutiert wurde, da die Vorschläge oft divergieren und so die Kontroversen eher in den Vordergrund treten. In der Argumentationsphase sollen die einzelnen Vorschläge durch Pro- und Kontra-Argumente abgewogen werden. Dies führt dann direkt zu einer Evaluationsphase, in der den Argumenten „beliefs“ zugeordnet werden: Eine Aussage kann von einer Person geglaubt werden, von einer anderen nicht. Daraus werden „belief sets“ gebildet, die bestimmten Standpunkten oder Ansichten zugeordnet werden können.

Realisierung des Szenarios. Wie in Abbildung 13 zu erkennen ist, gehören mehrere vernetzte Computer und eine elektronische Tafel zu diesem Szenario. Das Labor ist für die „face-to-face"-Situation ausgelegt. Gearbeitet wird in „shared" Workspaces, die über eine Datenbank synchronisiert werden. Technisch führte das zu Problemen bei der parallelen Eingabe von Text.

Organisatorische Einbettung. Die organisatorische Einbettung besteht in Form der „Spielregeln", die besagen wie die Tools verwendet werden. Insbesondere die Argnoter-Methode wurde ebenso auch ohne Tool nur mit Hilfe eines „schwarzen Brettes" praktiziert. Andere Aspekte wie beispielsweise Dokumentationen und Retrieval werden nicht problematisiert.

3.2.4 Csile

- *Zielrichtung.* Lernunterstützung; Kommunikationsmedien und Wissensprodukte
- *Schwerpunkt.* Kritische Wissenskonstruktion beim Lernen
- *Kriterien.* Nachhaltigkeit

Software/Tool. Csile (computer supported intentional learning environments) (Scardamalia & Bereiter, 1992; Scardamalia, Bereiter, Brett et al., 1992) wurde als Lernsystem für Schülergruppen entwickelt.

Die Gruppe soll mit dem System Wissen konstruieren und dabei kritisch hinterfragen. Die Arbeitsweise besteht darin, dass einzelne Personen Notizen verfassen, wie sie in Abbildung 14 zu sehen sind. Zu jeder Notiz können Grafiken oder weitere Notizen annotiert werden, allerdings können nur die Autoren bzw. Autorinnen einer Notiz diese ändern oder löschen. Sie werden jedoch benachrichtigt, wenn ein Beitrag kommentiert wird.

Für Csile 2.0 (Scardamalia & Bereiter, 1992) wurde eine Teilung in thematische Arbeitsbereiche („thematic spaces") und Arbeitsumgebungen („environments") vorgesehen. Erstere lassen eine Sortierung nach Themen zu, so dass Beiträge zu bestimmten Themen assoziierbar sind. Die Arbeitsumgebungen (Erklärungen, Funktionsweisen, Exploration, Bedeutungen) bieten bestimmte Sichten an und stellen dafür bestimmte Tools zur Verfügung. Bei der Frage nach Funktionsweisen wäre es beispielsweise sinnvoll Animationen zu integrieren. Bei der Darstellung von Bedeutungen sollen später „Concept Maps" erstellt werden.

Das System soll außerdem Automatismen und Hilfen zur Verfügung stellen, wie beispielsweise eine Rechtschreibprüfung für Schlüsselwörter und Hinweise auf angrenzende Beiträge.

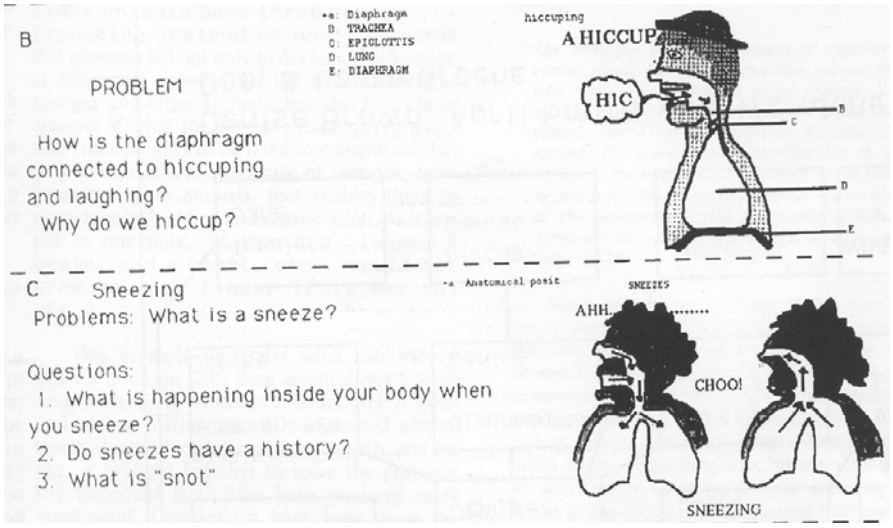


Abbildung 14

Skizzen und Annotationen im Csile-System (Scardamalia, Bereiter, Brett, et al., 1992).

Realisierung des Szenarios. Entwickelt wurde das System für eine asynchrone und verteilte Arbeitssituation. Die Notizen - Graphiken und textuelle Eingaben – werden zu einer zentralen Datenbank hinzugefügt und können von anderen annotiert werden. Diese Kooperation kann sowohl unter Schülerinnen und Schülern einer Klasse stattfinden, wie zwischen denen einer Schule, also auch mit unterschiedlichem Lernniveau und auch außerhalb des Schul-kontextes.

Durch dieses Vorgehen sollen Schülerinnen und Schüler dazu animiert werden, über die Inhalte, ohne denn Druck einer mündlichen Situation, zu reflektieren und durch die verteilte Situation sollen sie lernen, unabhängig eigene Gedanken zu entwickeln.

Organisatorische Einbettung. Der Csile-Ansatz ist ein kooperativer Ansatz zur Wissenskonstruktion. Durch ihn soll es den Lernenden, im Gegensatz zu traditionellen Lehrmethoden, erleichtert werden, eigenen Fragestellungen zu folgen („intentional learning“; Scardamalia & Bereiter, 1994). Scardamalia und Bereiter (1994) bezeichnen diesen Lernprozess als „second-order“, da man sich nicht asymptotisch einem Lernziel nähert, sondern sich die Lernziele während des Lernens ändern können sowie das Niveau der Ziele.

Angestrebt wird ein Prozess, der mit dem eines rezensierten Journals vergleichbar ist. Dieser Prozess wird zwar nicht erzwungen aber doch durch die Art der möglichen Eingaben sehr nahe gelegt. Aus dem Vergleich zum akademischen Publizieren werden auch intrinsische Motivationen abgeleitet, mit solch einem System zu arbeiten, dem Wunsch nach Anerkennung, dem Wunsch, etwas zu beeinflussen und dem Wunsch an relevanten Diskursen teilzuhaben.

Eine Prozessunterstützung findet statt in Anlehnung an diesen Publikations- und Rezensionsprozess. Zum einen müssen Beiträge „veröffentlicht“ werden. Die Lehrerin oder der Lehrer oder eine Gruppe entscheidet darüber, welche Beiträge vom „zu publizieren“ zum „publiziert“-Status wechseln. Um zu veröffentlichen, muss für den Beitrag formuliert werden, worin seine besondere Aussage liegt.
Weiterhin werden Autorinnen und Autorinnen darüber benachrichtig, wenn auf sie Bezug genommen wurden. Das heißt, dass nicht nur veröffentlichte Beiträge von anderen gelesen werden können, diese aber einen besonderen Grad der Ausarbeitung erreicht haben.
Das Szenario und die verwendete Datenbanktechnologie ist hier weit mehr als eine zufällige Zusammenführung, die die Kooperation ermöglichen soll. Die Verteiltheit der Personen wird didaktisch motiviert (verbesserte Reflexion, unabhängige Gedanken, Eigeninitiative). In der Verwendung einer Datenbank wird ein Vorteil zum Einsatz eines reinen Hypermediasystems gesehen, in dem alle Verknüpfungen zwischen Materialen explizit über Links dargestellt sind. Mit der Datenbank können über Suchen dagegen immer neue Suchkriterien aufgestellt und so immer neue Verbindungen gefunden werden.

3.2.5 Dolphin

- *Schwerpunkt.* Kommunikationsmedien und Wissensprodukte
- *Zielstellung.* Gesprächsunterstützung
- *Kriterien.* Verwendung im „face-to-face“-Szenario, Nutzung visueller Sprachen

Software/Tool. Das Dolphin-System (Streitz, Geißler et al., 1998) setzt auf der Sepia-Architektur (wird nachfolgend ebenfalls erläutert) auf und adaptiert diese für Besprechungssituationen („face-to-face“).
In einem Dolphin-Workspace (Abbildung 15, rechts) können Bilder, Texteingaben und handschriftliche Eingaben kombiniert werden. Zur Strukturierung werden Knoten angeboten, die Inhalte zu einem Objekt zusammenfassen. Sie werden einfach durch das handschriftliche Umranden von Inhalten erstellt und erscheinen daraufhin in einer grau schattierten Box.
Es existieren zwei Möglichkeiten, um Knoten zueinander in Relation zu setzen: Einerseits können Links in einem Workspace erstellt werden, andererseits können Knoten in andere Knoten gelegt werden, so dass eine „dreidimensionale“ Struktur entsteht. Aus einer abstrakteren Sicht auf die Knoten können auch Relationen zwischen Knoten unterschiedlicher „Ebenen“ angelegt werden. Die Knoten in Dolphin sind zwar nicht typisiert aber es ist möglich, Dokumente aus dem Sepia-System zu importieren, die

getypte Knoten enthalten. Gleichermaßen können Dolphin-Dokumente für Sepia exportiert werden.

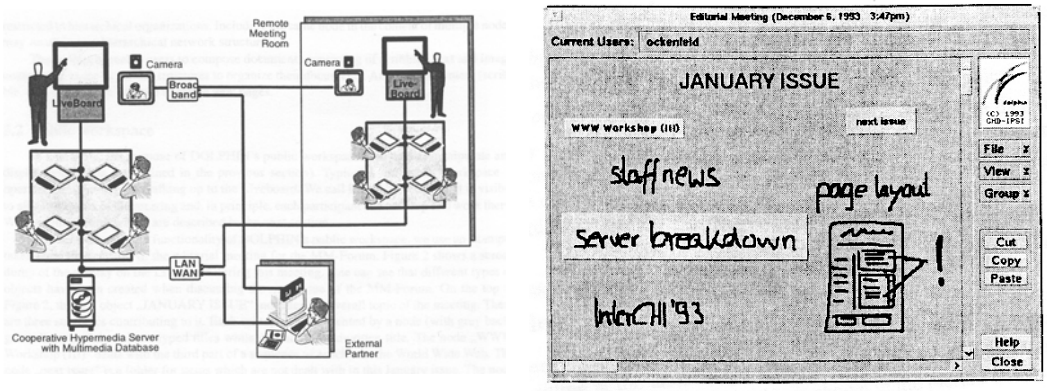


Abbildung 15

Links: Das Dolphin-Szenario; Rechts: Ein Workspace im Dolphin-System.

Realisierung des Szenarios. Mit Dolphin soll nicht nur das „face-to-face"-Szenario unterstützt werden, sondern es soll explizit ein nahtloser Übergang zwischen unterschiedlichen Arbeitsszenarien möglich sein. Dazu gehört die direkte Arbeit an der elektronischen Tafel, die „face-to-face"-Gruppenarbeit in einem Raum mit vernetzten Rechnern für die Beteiligten und einer elektronischen Tafel sowie die Möglichkeit, externe Personen bzw. Computer einzubinden oder sogar weitere Arbeitsgruppen (Abbildung 15, links).

Durch die Interoperation mit Sepia, das auch asynchrones Abreiten unterstützt, werden asynchrone Arbeitsphasen stärker mit den synchronen integriert.

Organisatorische Einbettung. Es wird davon ausgegangen, dass „meeting support" die Vielzahl von Aktivitäten, beginnend mit einer Agenda und Entwürfen, Ideen und Notizen, hin zu Arbeitsplänen und Dokumenten unterstützen muss. Dies wird jedoch nur teilweise und implizit erreicht, z.B. indem vorbereitete Inhalte mit in die Diskussion eingebracht werden können. Als Mittel stehen dafür öffentliche und private Workspaces zur Verfügung zwischen denen Inhalte transferiert werden können. Strukturiertere Inhalte können in Sepia entworfen werden.

3.2.6 gIBIS

- *Schwerpunkt.* Kommunikationsmedien und Wissensprodukte
- *Zielstellung.* Design rationale
- *Kriterien.* Nutzung visueller Sprachen, Nachhaltigkeit

Software/Tool. gIBIS (Conklin & Begemann, 1987) ist ein Hypertext-Werkzeug, mit dem Design-Überlegungen in der Gruppe unterstützt werden sollen und das gleichzeitig der Dokumentation der Begründungen dient. Dafür erlaubt es, Argumente und Standpunkte grafisch über drei verschiedene Inhaltstypen zu strukturieren: Issues, Positions, Arguments. Für diese wird ein Übersichtsgraph dargestellt, in dem die jeweiligen Inhaltsknoten durch Relationen miteinander in Verbindung gesetzt werden. Die Knoteninhalte werden in separaten Fenstern dargestellt.

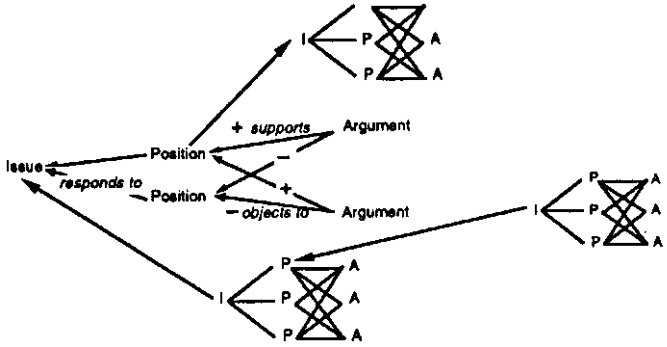


Abbildung 16

Schematische Darstellung der gIBIS-Darstellungen. „**I**ssues“ werden durch verschiedene „**P**ositions“ aufgegliedert. „**A**rguments“ nehmen dazu Stellung.

Ausgegangen wird davon, dass durch unterschiedliche Interessen jeweils andere Standpunkte eingebracht werden und diese in einen Zusammenhang für die eigentliche Design-Aufgabe gebracht werden müssen. In den Graph können und sollen Skizzen und Notizen genauso wie Design-Entscheidungen und Begründungen aufgenommen werden. Die Design-Überlegungen werden asynchron editiert, wodurch die Navigation durch die entstandenen Netze eine große Rolle spielt. Diese wird durch einen Browser unterstützt, der das Netz im Überblick abbildet, in dem jedoch nur die Struktur erkennbar ist, sowie durch einen lokalen Browser, der Teile des Netzwerkes vergrößert („zoom“), und mit den Details für die Knotentypen darstellt. Der lokale Browser stellt auch die syntaktische Korrektheit der Netze sicher.

Die Navigation wird weiterhin durch Suchen ermöglicht. Dafür werden Beispielknoten erstellt, für die dann Entsprechungen im Graph gesucht werden („query by example“, Conklin & Begemann, 1987, S. 249).

Realisierung des Szenarios. gIBIS unterstützt das asynchrone Arbeiten. Es dient dabei als Kommunikationsmedium für die Tele-Kooperation. Konkrete „face-to-face“-Diskussionen sind nicht explizit eingeplant, das heißt, dass alle Argumente über das System ausgetauscht werden müssen. Unklar bleibt, wie die Nutzerinnen und Nutzer die

Veränderungen erkennen und verfolgen können, ohne durch das gesamte Netz stöbern zu müssen.

Technisch wird die Kooperation im LAN durch „locking“ von Knoten und durch Benachrichtigungen bzgl. inhaltlichen Veränderungen unterstützt.

Organisatorische Einbettung. Die ursprüngliche Motivation, ein solches System zu entwickeln, bestand darin, den Design-Prozess selbst untersuchen zu können und weiterhin in der Frage, wie große Mengen an informellen Inhalten genutzt, indexiert oder auch wiedergefunden werden können. Mit dieser Ausrichtung kann gIBIS nicht nur als Wegbereiter für viele Systeme zum „design rationale“ gewertet werden, sondern stellt außerdem bereits viele Fragen zum Thema Wissensmanagement, das eigentlich erst 10 Jahre später aktuell wurde. Von dort stammt vor allem die Idee, visuelle Repräsentationen für die Analyse von Design-Ideen einzusetzen, sowie die Annahme, dass Ideen ein ganz zentrales Element solcher Wissensprozesse sind. Auch die Relevanz, die Wissens-entwicklung zu dokumentieren, wird dort bereits hervorgehoben.

3.2.7 MindManager

- *Schwerpunkt.* Kommunikationsmedien und Wissensprodukte
- *Zielstellung.* „Mind Mapping“
- *Kriterien.* Nutzung visueller Sprachen

Software/Tool. MindManager (Abbildung 17) von Mindjet ist vermutlich das am weitesten verbreitetste „Mind Mapping“-Tool auf dem Markt ([MindManager][8]). Das Tool dient der Ideenbildung durch Brainstorming, Vorbereitung von Meetings und durch den strukturierenden Ansatz beispielsweise auch der Projektorganisation. Die damit zu entwickelnden „Maps“ (Karten) werden im Wesentlichen in Baumstrukturen dargestellt. Nur mit einer Art von Relationen können auch einzelne Äste miteinander assoziiert werden. Der Typ eines Knotens wird meistens über ein Symbol verdeutlicht, das z.B. die Art referenzierter Medien symbolisiert. Ansonsten können auch Graphiken in den Arbeitsbereich eingefügt werden. Jede Art von Text wird in Knoten eingegeben oder durch referenzierte Medien verknüpft.

Das System übernimmt die Anordnung der Knoten im Workspace. Dies ist einerseits hilfreich, da eine gleichmäßige Verteilung der Eingaben angestrebt wird, andererseits aber auch verwirrend, wenn automatische Veränderungen den „Bildeindruck“ zerstören.

[8] Erläutert wird das System im Kontext von Kreativitätswerkzeugen auch in einem Artikel der Zeitschrift c’t (20/98, Jungbluth).

Realisierung des Szenarios. In der ursprünglichen Version war MindManager ein „single user“-System. Aktuell wurde es aber zu einem System erweitert, das die synchrone Kopplung zulässt. Dafür muss ein Konferenzserver mit installiert werden, an den sich die Applikationen anmelden müssen. Zur Kopplung muss eine Sitzung freigegeben werden, wofür auch Zugriffsrechte vergeben werden können. Der Server wartet eine Liste aller aktuellen Konferenzen, die auch permanent eingerichtet werden können.

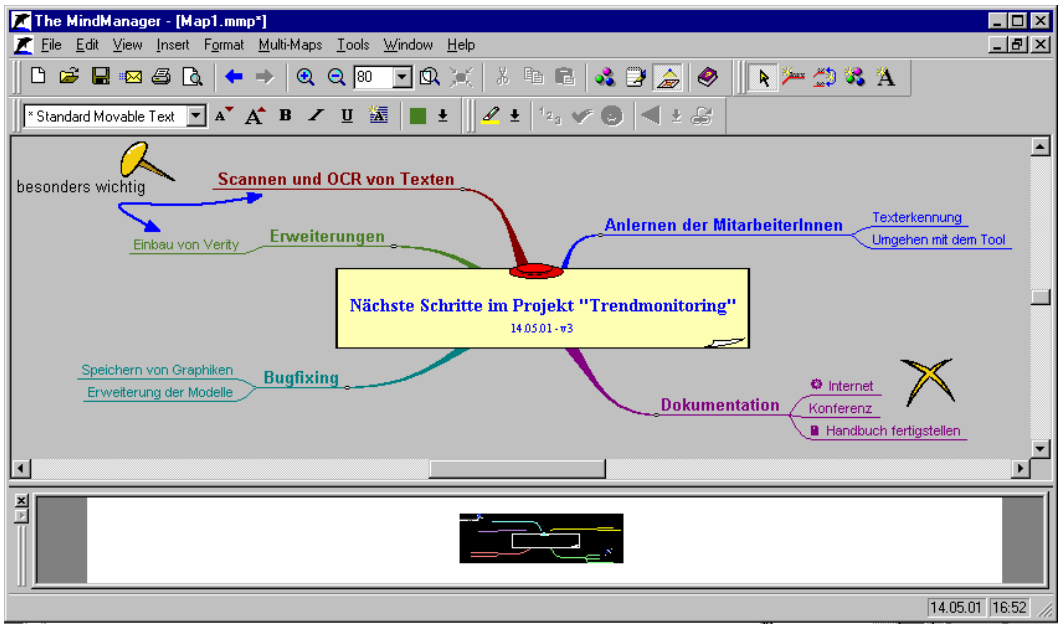


Abbildung 17

Das Interface des Tools MindManager.

Organisatorische Einbettung. Ein „Mind Mapping“-Tool kann vielseitig eingesetzt werden, von der Vorbereitung bis zur Dokumentation. Konkrete Prozesse werden hier nicht unterstützt, durch die Exportmöglichkeit zu Standardformaten wie PowerPoint, Word, Outlook und html können allerdings Medienbrüche beim Wechsel zu anderen Tools vermieden werden. Somit ist durchaus eine flexible Integration des Tools in existierende Prozesse möglich.

3.2.8 OneNote

- *Schwerpunkt.* Kommunikationsmedien und Wissensprodukte
- *Zielstellung.* Organisieren von Notizen
- *Kriterien.* Nutzung visueller Sprachen, Nachhaltigkeit

Software/Tool. OneNote von Microsoft ([OneNote]) kann hier nur als Ausblick verstanden werden, denn dabei handelt es sich um ein Tool der Office-Gruppe, das erst

Mitte 2003 auf den Markt kommen soll[9]. Das Tool stellt mehrere Strukturierungsmöglichkeiten zur Verfügung, wie sogenannte Notebooks, Files und Seiten. Auf den Seiten können Bilder, Text, Handschrift und Audio-Informationen angeordnet und verschoben werden. Inhalte können per Drag&Drop aus Web-Browsern integriert werden und behalten als Information ihre Internetseite. Als Bestandteil der Office-Familie besteht zu den Parallelprodukten eine Exportmöglichkeit. Ergebnisse können auch im Web publiziert werden.

Für die erstellten Daten werden Suchen angeboten und es wird besonderer Wert auf die Datensicherheit gelegt.

Realisierung des Szenarios. Das System ist als „single user"-System geplant und zielt vermutlich auf eine deutlich verbesserte Verwendung als Notizbuch im Vergleich zu Handhelds.

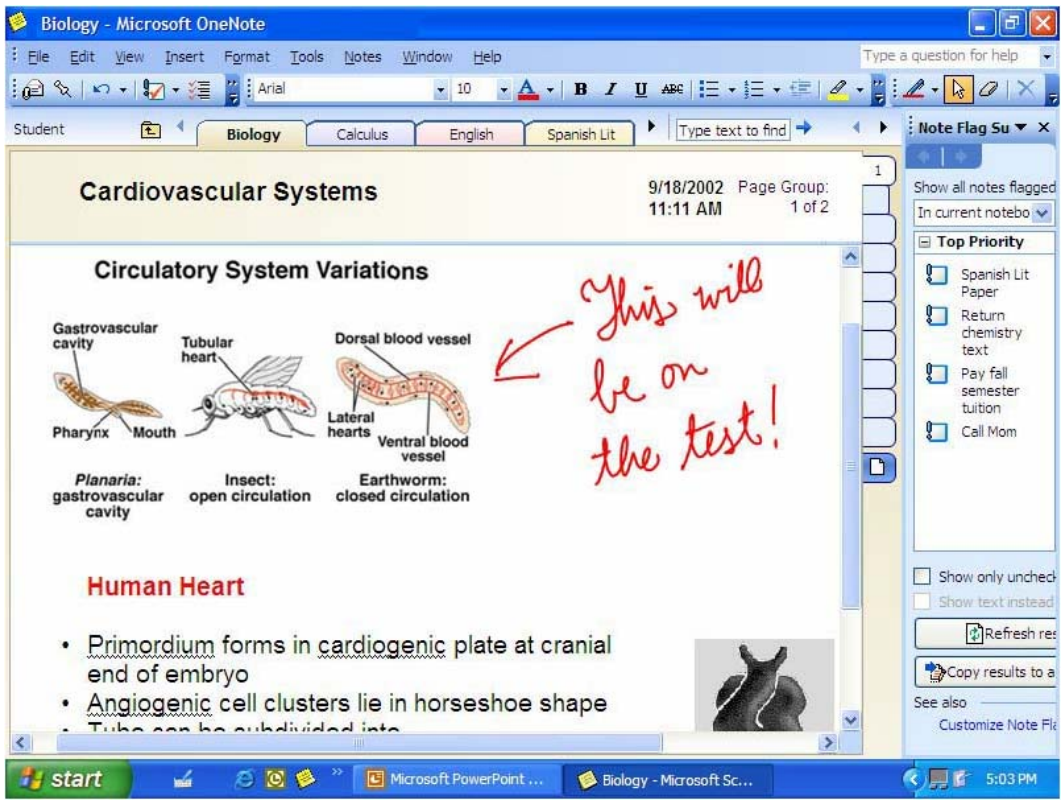


Abbildung 18

Oberfläche und Nutzungsbeispiel des Tools OneNote.

Organisatorische Einbettung. Das Szenario eröffnet komfortable Notizmöglichkeiten. Das kann weite Bereiche abdecken, angefangen vom „Mind Mapping" bis hin zur Planung von Aufgaben und Terminen. Es unterstützt vor allem den Übergang dieser

[9] Die Beschreibung stützt sich deshalb nur auf kurze Informationen aus dem Internet.

alltäglichen Notizen in ein digitales Format mit Übergang zu weiteren (Word, etc.) und zielt damit unter anderem auf die Vermeidung von Medienbrüchen. Der Schritt, die Inhaltsaufbereitungen und Datensicherungen auch kooperativ zu nutzen, wurde mit diesem Tool bisher noch nicht vollzogen.

3.2.9 Sepia

- *Schwerpunkt.* Kommunikationsmedien und Wissensprodukte
- *Zielstellung.* Planung und Erstellung von Hypertexten
- *Kriterien:* Nutzung visueller Sprachen, Nachhaltigkeit

Software/Tool. Das Sepia-System (Streitz, Haake, et al., 1992; Streitz, Haake, et al., 1998) soll Autorinnen und Autoren kognitiv bei der Erstellung von Hypertexten unterstützen (Abbildung 19).

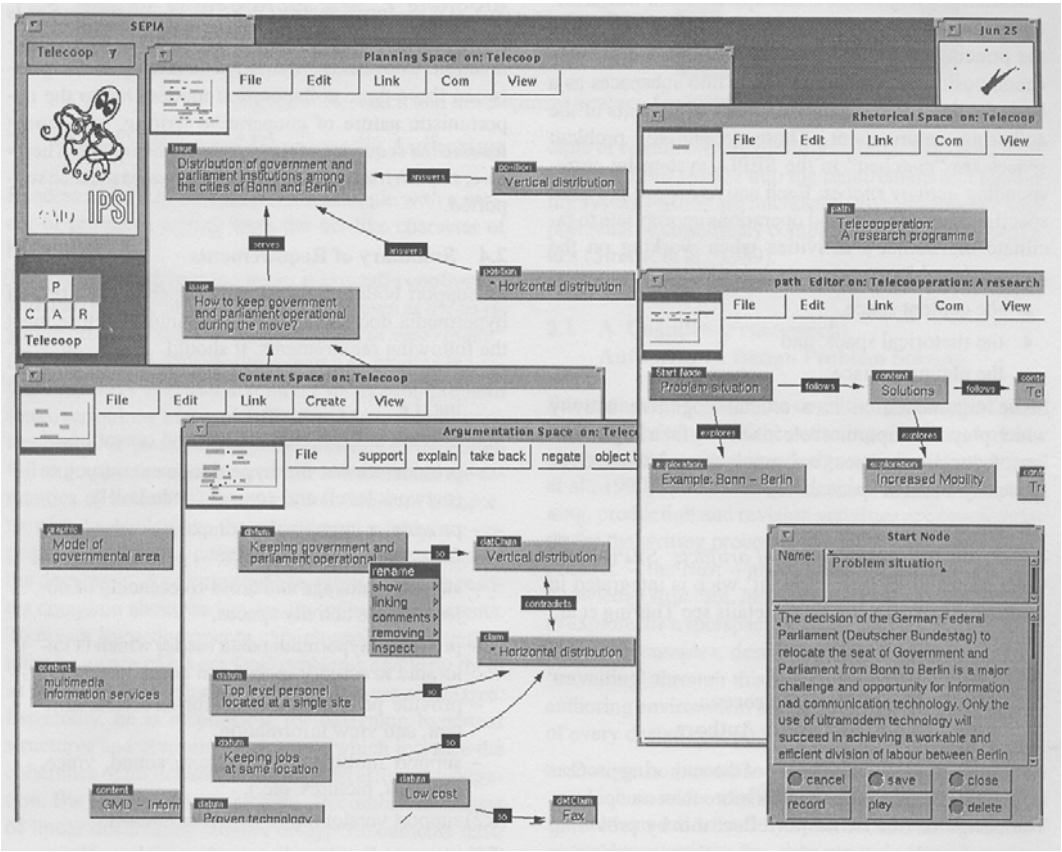


Abbildung 19

Das Sepia-System mit mehreren geöffneten „activity spaces".

Der Schreibprozess wird als Design- und Problemlöseprozess verstanden, bei dem Planungs-, Produktions- und Revisionsphasen durchlaufen werden. Bei diesem gemeinsamen Schreiben müssen vor allem Probleme auf der inhaltlichen, der rhetorischen und der Planungsebene gelöst werden, für die unterschiedliche „activity spaces" zur

Verfügung gestellt werden, ergänzt um einen zusätzlichen Argumentations-Workspace. Ein „activity space“ zeichnet sich durch eine vordefinierte visuelle Sprache aus Knoten- und Kantentypen aus. Die Knoten selbst tragen keine operationale Semantik. Wie bei Dolphin können „dreidimensionale“ Strukturen angelegt werden, indem Knoten in andere verschoben werden.

Realisierung des Szenarios. Das Sepia-System entstand aus der Überzeugung, dass große Hypertexte fast immer von Gruppen erstellt werden. Es wurde für die örtlich verteilte Kooperation entwickelt, die wiederum durch unterschiedliche Kooperationsmodi strukturiert wird. Es steht ein „single user“-Modus zur Verfügung, eine lose Kopplung und eine starke Kopplung. Diese drei Modi unterscheiden sich insbesondere dadurch, inwieweit die einzelnen Autoren und Autorinnen über die Arbeit anderer an gleichen Knoten oder Knotengruppen informiert werden und gleichzeitige Arbeit erlaubt wird, bzw. einzelne Objekte gesperrt werden.

Die Sepia-Architektur baut auf einer „Hypermedia Engine“ auf, die die anfallenden Daten, in einer „Hyperdocument“-Datenbank verwaltet. Für die einzelnen Objekte (Knoten und Links) werden darin Inhalte zusammen mit Attributen abgelegt, so dass sie auch in unterschiedlichen „activity spaces“ angezeigt werden können.

Ein kooperativer Hypermediaserver organisiert den „multi-user“-Zugriff auf Dokumente. Jeder Client beinhaltet einen „broadcast listener“ und einen „broadcast server“ für die Synchronisation der Arbeit in „shared“ Workspaces.

Organisatorische Einbettung. Ein Prozess bei der Erstellung von Hypertexten wird implizit durch die „activity spaces“ unterstützt. Eingebettet in den „rhetorical space“ ist außerdem ein „construction kit“, das dabei helfen soll, am Ende den eigentlichen Hypertext zu erstellen, um den Medienbruch zwischen der Textplanung und dem eigentlichen Hypertext zu vermeiden.

Eine Verknüpfung der Ergebnisse, der Hypertexte, im Sinne eines „corporate memory“ wird nicht unterstützt. Darin wäre es beispielsweise sinnvoll, Autoren und Autorinnen über Änderungen zu informieren, wie beispielsweise bei Csile. Undeutlich bleibt in den Beschreibungen, wie mit größeren Netzen umgegangen werden kann. Unklar bleibt auch, inwieweit das potentiell asynchrone Szenario die Erstellung der Hypertexte unterstützt, wie beispielsweise mit der dezentralen Veränderung von Inhalten umgegangen wird und wie die eigentlichen Abstimmungsprozesse dann ablaufen.

Das System ist sehr Umfangreich, an hohe technische Voraussetzungen gebunden und integriert viele Komponenten, angefangen bei den „activity spaces“ für den Autorenprozess, über die Ermöglichung kooperativen Arbeitens bis hin zur persistenten

Datenhaltung. Es scheint jedoch weniger komponentenorientiert implementiert, so dass es schwierig adaptierbar zu sein scheint. So könnten beispielsweise die Kooperationskomponenten eher eigenständig und visuelle Sprachen leichter definierbar sein.

3.3 Erfahrungen bei der Verwendung visueller Sprachen

Problemzerlegungen. In Conklin und Begemann (1987) werden in Bezug auf die Nutzung des gIBIS-Systems verschiedene Erfahrungen aufgezeigt. Dazu gehört, dass eine Schwierigkeit in der zu frühen Zergliederung von Problemen bestand. Das sei vor allem dann nicht produktiv, wenn das eigentliche Problem noch nicht gut verstanden ist. Gleiches wird auch von McKerlie und MacLean (1994) im Rahmen von QOC-Studien erwähnt. QOC, Questions-Options-Criteria (MacLean et al., 1991), ist nicht an ein System gebunden, sondern stellt ein Schema dar, nach dem Design-Probleme analysiert werden können. Sie ist dem IBIS-Schema sehr ähnlich, grenzt sich aber davon ab, als dass darin die Analyse und Reflexion durch die Designer angeregt wird, während bei der Umsetzung von IBIS in gIBIS vor allem die Dokumentation der entstehenden Ideen unterstützt wird (McKerlie &MacLean, 1994).

Löwgren (1994) analysiert den Design-Prozess weitergehend auf Basis der QOC-Methode und kommt zu einer Charakterisierung von Design-Prozessen als „evolutionär“ oder „argumentativ“. „Evolutionär“ besagt, dass sich die Argumente, die zu Entscheidungen geführt haben, zu diesen korrespondieren und jeweils den aktuellen Stand reflektieren, nicht aber die Historie und Gründe die zu Veränderungen geführt haben.

Bei „argumentativen“ Herangehen werden verschiedene Alternativen zu einem Zeitpunkt gegeneinander abgewogen werden, um daraufhin beispielsweise zu einer Entscheidung zu kommen. Im weiten Sinn könnte behauptet werden, dass gIBIS den „evolutionären“ Ansatz realisiert, während die QOC-Methode auf den analytischen Ansatz abzielt.

In der Untersuchung wurde beobachtet, dass beim „evolutionären“ Ansatz eine Tendenz besteht, vorschnell in subjektiv bevorzugte Ideen zu verfallen und andere dann nicht mehr zu verfolgen oder gar nicht erst versucht wird, andere Ideen zu finden.

Die Unterstützung der „argumentativen“ Herangehensweise, die darauf abzielt den gesamten Argumentationsraum aufzuspannen, hat den Designern insofern geholfen, als dass sie stark subjektive Entscheidungen später revidierten, da diese Inkonsistenz durch die Darstellung transparent wurde. In McKerlie und MacLean (1994) wird auch auf die Eigenart verwiesen, dass eher versucht wird, stützende Kriterien für einen Vorschlag zu finden als Gegenargumente.

Diese Erfahrungen sind insofern wichtig, als dass visuelle Sprachen generell den Zerlegungsaspekt in sich tragen, welcher gerade der Klärung von Sachverhalten dienen soll. Wenn die diskussionsbegleitende Zergliederung jedoch gerade zur Vernachlässigung von Aspekten führt, wie vielleicht in einem „evolutionären" Ansatz, so wäre das fatal.
Ein weiteres für gIBIS erkanntes Problem bestand darin, dass die Autorinnen und Autoren allein durch die Idee des Hypertextes dazu angehalten waren, die Knoteninhalte so zu formulieren, dass sie weitgehend selbsterklärend waren, denn der Kontext, der in normalen, linearen Texten eher gegeben ist, geht deutlich verloren. Leser können auf beliebigen Wegen zu diesem Knoten gelangen. Dieses Problem ist nach meiner Ansicht grundlegend mit der Idee des Hypertextes verwoben.
Einbeziehen der Visualisierung in die Gesprächssituation. Im Kapitel 4 entsteht im Ergebnis eine zyklische Strategie, in der sich Notations- und Diskussionsphasen abwechseln. Eine entsprechende Beobachtung wurde in Stefik et al. (1987) in Bezug auf das Cognoter-System beschrieben. Auch dort lösen sich reine Diskussionsphasen ab mit Phasen, in denen alle Beteiligten parallel Eingaben anfertigen über die dann später – als Gesprächsfokus – wieder diskutiert wird. Dort wird außerdem beschrieben, dass zweidimensionale Gruppierungen ein Übergangsstadium bilden, zwischen reinen Sammelphasen und Phasen, in denen Relationen eingefügt werden.
In Suthers (1999C) wird die Beobachtung gemacht, dass die externe Repräsentation in der „face-to-face"-Situation weniger dazu dient, in bzw. mit dieser Repräsentation zu argumentieren, als dazu, sie als Basis für die Argumentation zu verwenden, als Stichwortgeber. Er begründet das so, dass die Repräsentation nicht die eigentlich Kommunikation manifestiert, sondern diese *durch* das Medium stattfindet.
Einbindung in Arbeitssituationen. Bei der Verwendung von Cognoter und Argnoter zeigte sich, dass das gemeinsame Erarbeiten von Inhalten stärker konsensorientiert verläuft, als Diskurse, in die einzelne Personen bereits vorbereitete Beiträge einbringen. Dies deckt sich mit den Erfahrungen aus der in Kapitel 4 beschriebenen Vorstudie, in der auf Basis von vorbereiteten Beiträgen eher disputiert wurde und kann bewusst methodisch eingesetzt werden.
Bezüglich Csile wird vor allem darauf hingewiesen, dass Funktionalitäten immer auch einen Sinn im didaktischen Ablauf haben müssen. Dort gab es die Möglichkeit, Schlüsselwörter für Beiträge zu vergeben, um so eine Abstraktion zu erhalten. Dieser Schritt der Abstraktion wurde jedoch nicht ausgenutzt. Er war weder in den Unterrichtsprozess explizit integriert, noch in die Abläufe, die durch das System nahegelegt werden.

Ein großer Wert bei Csile bestand darin, dass die Beteiligten einen Fortschritt der Inhalte erkennen konnten und wollten. Dies ist sicherlich gerade wegen der verteilten Arbeitssituation wichtig: jede und jeder will sehen, was mit den Ergebnissen passiert. In einer „face-to-face"-Arbeitssituation ist dies anders, da der Progress der Inhalte ja gerade zusammen in der Gruppe unternommen wird.

Bei itIBIS (Conklin & Burgess-Yakemovic, 1996), einer einfachen, textbasierten Umsetzung des IBIS-Schemas, wurde allein ein Erfolg darin gesehen, dass die Gruppenmitglieder die IBIS-Methode eintrainierten. Dies führte zu produktiveren Arbeitstreffen: Es wurde danach besser erkannt, wann man sich vom eigentlichen Thema weg bewegte, und man konnte einfacher wieder auf den Ausgangspunkt zurückkommen.

Die QOC-Methode war erfolgreich, um auf die zentrale Fragestellung zu fokussieren, diese nicht aus den Augen zu lassen und konkurrierende Ideen darzustellen (McKerlie & MacLean, 1994).

Wiederverwendung. Bei der Verwendung von itIBIS (Conklin & Burgess-Yakemovic, 1996) wurde deutlich, dass die Aufschlüsselung der Design-Ideen einerseits den Designern selbst hilft, sich an diese später wieder zu erinnern. Andererseits, beim Ausscheiden einer für das Design zentralen Person aus der Arbeitsgruppe, konnte die Gruppe das Produkt auf Basis der dokumentierten Design-Entscheidungen einigermaßen problemfrei zu Ende entwickeln. Generell konnten Fragen der Art „Warum haben wir das so gemacht?" rückverfolgt werden. Im Vergleich zwischen verschiedenen Arbeitsgruppen, die mit und ohne die IBIS-Methode gearbeitet hatten, führte deren Verwendung zu einer transparenteren Aufschlüsselung der Anforderungen und Begründungen.

Da die Ergebnisse der IBIS-Methode und der QOC-Methode eher den „Design Space" darstellen, fehlt eine Darstellung der Entwicklung der Ideen und teilweise der verworfenen Ideen (McKerlie & MacLean, 1994).

3.4 Synopse

Aus den vorangegangen Kapiteln wurde deutlich, dass Visualisierungen einen breiten Einsatz beim Konstruieren von Wissen haben. Da sie außerdem sehr gut verwendet werden können, um Kommunikationssituationen durch weitere Ausdrucksmittel und Strukturelemente anzureichern, sind Visualisierung insbesondere für den Informationsaustausch in Gruppen geeignet.

Die Arbeitshypothese dieser Arbeit besteht darin, dass visuelle Sprachen dazu geeignet sind, um mit ihnen sonst flüchtige Inhalte aus diversen Arbeitssituationen zu notieren.

Deren potentiell semi-formale Darstellung bietet außerdem viele Optionen, die Inhalte Computersystemen zugänglich zu machen. Damit sind alle möglichen Funktionalitäten gemeint, die sich einerseits auf die Einbindung der Produkte in Arbeitsprozesse beziehen können, wie das Retrieval oder die Unterstützung eines Wissensflusses im Allgemeinen. Andererseits sind damit auch solche Funktionen gemeint, die die Kommunikationssituation selbst komfortabler gestalten, wie verbesserte Referenzen zwischen Beiträgen, ausgefeilte Sprachelemente, Perspektiven, Sortierungen, Feedback, etc. Diese Überführung von flüchtigem, dynamischem Wissen in eine technisch handhabbare Form soll insbesondere die Nachhaltigkeit der besprochenen Ergebnisse verbessern. Hierbei handelt es sich um ein zentrales Ziel im Wissensmanagement.
Im Verlauf dieser Arbeit entstand die Vermutung, dass zwar diverse Systeme existieren, die visuelle Sprachen in der einen oder anderen Weise verwenden, dass aber genau dieser Überführungsprozess meistens nur in Teilen bedacht wird. Das gilt genauso für die Unterstützung eines Prozesses durch das System, wie für die Einbettung des System in einen organisatorischen Ablauf. Mit einer Prozessunterstützung ist dabei nicht ein Workflowsystem gemeint, das Arbeitsvorgänge strikt vorgibt, sondern die freie Einbindung von Tools in den Arbeitsalltag, worin trotzdem wiederkehrende Abläufe stattfinden. Ein Beispiel dazu ist Projektarbeit, die von den ersten Entscheidungen und Planungen hin zu Ergebnissen verläuft und die dokumentiert werden muss.
Für den Bereich des Wissensmanagement ergäbe sich daraus ein großer Mangel in Bezug auf den Mehrwert von Systemen, die z.B. nur einzelne Arbeitsphasen unterstützen.
Die Ursachen dafür liegen natürlich meistens in einer anderen Zielstellung, die sich die jeweiligen Entwicklerinnen und Entwickler gesetzt haben. Genau deshalb möchte ich ein offenes Feld im Bereich der Systementwicklungen andeuten, das gerade für das Wissensmanagement interessant sein sollte.
In der folgenden Zusammenschau werden deshalb Kriterien für Systeme vorgestellt, die nach meiner Ansicht relevant dafür sind, ob sie einen Arbeitsprozess unterstützen, in dem gerade dynamisches Wissen in einen Zyklus der Externalisierung und Internalisierung eingebracht werden kann. Darunter fallen auch Kriterien dafür, ob die potentiellen Möglichkeiten visueller Sprachen ausgenutzt werden. Die Kriterien werden den drei Ebenen zugeordnet, auf denen Diskussionen technisch unterstützt werden können (Kapitel 1.5.2).

3.4.1 Software/Tool

1. *Domänenunabhängigkeit* ist wichtig, um ein Visualisierungshilfsmittel alltäglich verwenden zu können.
2. Unrealistisch ist die Forderung einer wirklich formalen Repräsentation, um während normaler Arbeitsabläufe Wissensdomänen zu erstellen. Sinnvoll sind dagegen Möglichkeiten, Notizen digital aber *handschriftlich* anfertigen zu können, um flexible Notiztools zu erhalten. In diesem Zusammenhang könnte auch Handschrifterkennung eingesetzt werden.
3. *„Epistemic games"* werden in Collins und Ferguson (1993) gleichermaßen als analytische wie konstruktive Vorgehen eingeführt. Das „epistemic game" fügt einer epistemischen Form die Regeln hinzu, wie mit ihr umgegangen werden soll (Auf diese Begriffe wird im Kapitel 6.1 vertieft eingegangen.). Die Tabelle ist beispielsweise eine ganz grundlegende epistemische Form, die ganz unterschiedlich eingesetzt werden kann. Zu einer epistemischen Form können also mehrere epistemische Spiele existieren. Diesen konkreten Zusammenhang einer Form mit einer Umgehensweise bezeichne ich im Weiteren auch als Methode. Verschiedene solcher strukturellen Methoden sind mit visuellen Sprachen relativ leicht umsetzbar, wie Brainstorming, Meta-Plan, Argumentationsnetze oder manche Entscheidungs-methoden, und passen typischerweise zu bestimmten Phasen in einem Arbeitsprozess. Deshalb ist es sinnvoll mehrere solcher Methoden zur Verfügung zu haben und außerdem, Inhalte von einer Methode in eine andere zu übernehmen.
4. Wegen der Forderung nach Domänenunabhängigkeit, der flexiblen Nutzung, dem Wunsch nach Methodenkatalogen und der Unterstützung eines Arbeitsprozesses wäre es sinnvoll, dass Systeme an die jeweiligen Anforderungen von Nutzergruppen *adaptiert* werden können. Denkbar wäre das, indem Nutzerinnen und Nutzern die Möglichkeit gegeben wird, die Syntax von visuellen Sprachen und die Semantik von Operationen auf deren Objekten zu definieren. Eine Engineering-orientierte Variante bestünde z.B. darin, Architekturen bereitzustellen, zu denen Sprachen und Methoden als „Plug-In" hinzugefügt werden können. Wang und Haake (2000, S. 129) zitieren vier Arten, auf die ein System adaptierbar sein kann: Es kann generische Objekte zur Verfügung stellen (flexibel), es kann parametrisierbar sein, es kann leicht integriert werden oder das System kann selbst verändert werden („tailorable"). Mittel dazu können beispielsweise „Templates" oder Scriptsprachen sein.
5. *Feedback* muss nicht auf eine konkrete Domäne Bezug nehmen. Es kann z.B. auch Rückmeldungen über die Gruppenarbeit liefern sowie Auswertungen liefern, die

unter Umständen Bezug auf angewendete Methoden nehmen. Gerade diese Auswertungen harmonieren gut mit einer semi-formalen Darstellung mit visuellen Sprachen, wenn sie nicht die konkreten Inhalte analysieren, sondern auf strukturellen oder abstrakten Eigenschaften der Sprachobjekte beruhen.

3.4.2 Realisierung des Szenarios

6. Werden Diskussionen als Szenario verwendet, um Inhalte zu externalisieren, diese flüchtigen mündlichen Beiträge zu fixieren und zu dokumentieren, um diese später wieder aufzugreifen, dann beinhaltet dies Übergänge zwischen verschiedenen Arbeitsszenarios: Es wird synchron und asynchron gearbeitet, individuell und gemeinsam, verteilt und im gleichen Raum, es finden Vorbereitungen und Nachbereitungen statt, Materialien werden verwendet und Inhalte neu einbezogen. Vor diesem Hintergrund ist es sinnvoll, unterschiedliche Szenarien zu unterstützen, bzw. die Übergänge zwischen diesen. In Bezug auf die Realisierung des Szenarios werden dazu die Kriterien *verteilt*, *„face-to-face“*, *synchron* und *asynchron* aufgenommen. Werden mehrere davon mit einbezogen, so wird das zumindest als Indiz dafür gewertet, dass Rücksicht auf komplexere Arbeitsabläufe genommen wird. (Die nächste Rubrik, „Organisatorische Einbettung“, beinhalte angrenzende Kriterien.)
7. „*Shared objects*“ sind hier sowohl im Sinne einer Bedeutung zu verstehen, die in einer Gruppe geteilt wird, als auch im technischen Sinn als Objekte in „shared“ Workspaces. In der „face-to-face“-Situation muss nach dieser Beschreibung eine gemeinsam konstruierte Bedeutung nicht an ein technisches „shared object“ gebunden sein. Die Feststellung, dass mit technischen „shared objects“ umgegangen wird, wird als Indiz dafür gewertet, dass „kollaborative“ Situationen eingeplant sind. Darunter sei hier das wirklich gemeinsame Arbeiten am gleichen Wissensprodukt zu verstehen, das zu einem inhaltlichen Austausch führt, einer Wissenskommunikation. Technische „shared objects“ sind also ein Indiz dafür, dass die gemeinsame Wissens-konstruktion unterstützt wird.

3.4.3 Organisatorische Einbettung

8. Die *Interoperabilität* zwischen den Tools und Medien ist sinnvoll, um Medienbrüche zu vermeiden, denn es muss bedacht werden, dass in einer Organisation verschiedenste Materialtypen für die unterschiedlichen Arbeitszwecke verwendet werden, die unter Umständen in eine Kooperation und Kommunikation mit einbezogen werden sollen. In gleicher Weise sollen Kommunikationen in einen

Arbeitskontext mit einbezogen werden und Ergebnisse auch außerhalb des Kommunikationskontextes zugänglich sein. Bei einem Wissensmanagement-Ansatz ist nach meiner Ansicht nicht davon auszugehen, dass jede gewünschte Funktionalität aus einen umfassenden System heraus zur Verfügung gestellt wird. Vielmehr ist von einem Zusammenspiel vieler einzelner Tools oder Komponenten auszugehen.

9. *Repositories* dienen formal als „Gruppengedächtnis“ und bilden eine Grundlage für die Wiederverwendung von Informationen. Die Ergebnisse aus den Kommunikationen sollten darin aufgenommen und auch wiedergefunden werden können. Außerdem ist es wünschenswert Beziehungen zwischen den Inhalten des Repositories herstellen zu können.
10. Um in größeren Datenbeständen Informationen wiederfinden zu können, sollte ein *Informationretrieval*-Ansatz integriert sein. Dieser ist abhängig von der verwendeten Technologie. Dieser Aspekt schließt sich also direkt an den der Interoperabilität an. Abgelegte Formate sollten beispielsweise von handelsüblichen Suchmaschinen indexierbar sein. Andere Möglichkeiten sind Verschlagwortungen und deren Verwaltung in Datenbanken oder generell Metadaten-Ansätze. Graphisch oder Hypermedia-artig aufgearbeitete Repositories können auch einen explorativen Zugang ermöglichen.
11. Durchaus abhängig von der Art der Gruppenunterstützung kann die „*notification*“ – gemeint ist die Benachrichtigung über Ereignisse in den Datenbeständen – ein sehr wichtiges Konzept der Kooperation und Wissenskonstruktion sein.
12. Ein Visualisierungs-Tool kann zwar technisch für bestimmte Arbeiten geeignet sein und auch Arbeitsprozesse unterstützen, sollte aber auch in entsprechende *nicht-technische Arbeitsprozesse* organisatorisch mit einbezogen werden.

Nach diesen Kriterien werden im Folgenden die vorgestellten Systeme eingeordnet (Tabelle 5, Tabelle 6, Tabelle 7). Die Tabelle 5 widmet sich der ersten Gruppe der Kriterien, die die Software selbst charakterisieren. Alle in Betracht gezogenen Systeme sind domänenunabhängig (Tabelle 5), was eine Voraussetzung für die allgemeine Kommunikationsunterstützung ist. 7 der 10 Systeme verbinden ihren Ansatz mit einer oder mehreren Methoden („epistemic games“). Ausschließlich das Sepia-System stellt auch einen abgestimmten Methodenkatalog zur Verfügung. Chips bietet die Möglichkeit, „epistemic forms“ zu definieren, um so gruppenspezifische Prozesse begleiten zu können, wobei die Art des Umgangs mit diesen Darstellungen selbst schon ein Ergebnis der Kooperation ist. Gerade durch diesen erheblichen Freiheitsgrad geht jedoch die Anleitung verloren, die Sepia durch abgestimmte Methoden anbietet. Das CardBoard bietet zwar die

Möglichkeit, visuelle Sprachen extern zu spezifizieren, ist in der bisher vorgestellten Form aber noch nicht an eine bestimmte Anwendung adaptiert. Das Dolphin-System wiederum verzichtet weitgehend auf Methoden und unterstützt dafür ganz wesentlich die „dreidimensionale“ Strukturierung von Inhalten. Csile bietet in der Grundfunktionalität zwar keine expliziten Strukturierungen an, die Kommunikation wird aber maßgeblich durch den Lernprozess gesteuert, in den die Systemverwendung einbezogen wird. Allerdings integrieren neuere Versionen auch schon spezifische Methoden, wie beispielsweise „Concept Mapping“.

Software/Tool	Domänen-unabhängigkeit	Handschrift-eingabe	„epistemic games“ „epistemic forms“	Adaptierbarkeit	Feedback
Belvedere	+	–	Ø	–	+
CardBoard	+	–	+	+	–
Chips	+	+	+	+	–
Cognoter/ Argnoter	+	–	Ø	–	–
Csile	+	–	–	–	–
Dolphin	+	+	–	–	–
gIBIS	+	–	Ø	–	–
MindManager	+	–	Ø	–	–
OneNote	+	+	–	–	–
Sepia	+	–	+	–	–

Tabelle 5

Eigenschaften und Funktionen der Systeme.
+: Kriterium wird erfüllt bzw. Funktionalität ist vorhanden.
o: Wird nur in Bezug auf „epistemic games“ und „epistemic forms“ vergeben und bedeutet, dass genau eine Methode unterstützt wird.
–: Kriterium wird nicht erfüllt bzw. Funktionalität ist nicht vorhanden.

Die meisten Ansätze zum „meeting support“ bzw. zur Kommunikationsunterstützung gehen davon aus, dass Arbeitsphasen durch adäquate Methoden begleitet werden sollten. Dies kann einerseits durch die Verwendung verschiedener Tools gelöst werden, an die dann aber ein hoher Anspruch in Bezug auf die Interoperabilität bestünde. Andererseits kann ein einzelnes Tools verschiedene Methoden anbieten. Dies hat den Vorzug, dass die verwendeten Objekte eher interagieren können. Offen bleibt nach meiner Ansicht, ob ein umfangreicher Methodenkatalog besser vorgegeben wird, ob generell ein adaptiver Ansatz wie im Chips-System verfolgt werden sollte, oder ob ein Mischkonzept anzustreben ist. In allen Fällen ist die Verwendung von visuellen Sprachen, für die Syntax, Operationen, Constraints, usw. von Vorteil.

Die Option, im Kommunikationsprozess Feedback zu liefern, wird außer vom Belvedere-System nicht genutzt. Denkbar sind hier beispielsweise Analysen der Kooperation (Mühlenbrock, 2001; Mühlenbrock & Hoppe, 1999; Mühlenbrock & Hoppe, 2001), die ohne das System kaum durchführbar sind, zumindest nicht so, dass das Ergebnis gleich

wieder in die Gruppe zurück reflektiert werden kann. Auch Beteiligung kann bei größeren Gruppen interessant sein, insbesondere mit Bezug zum Thema.

Tabelle 6 gibt einen Überblick, welches Szenario mit welchem System unterstützt wird. In den Spalten sind die vorher genannten Kriterien so zusammengefasst worden, dass redundante Aussagen weitgehend vermieden werden. Beispielsweise ist davon auszugehen, dass für ein Szenario mit einer örtlich verteilten Gruppe, die synchron kooperiert, „shared objects“ eine Voraussetzung der gemeinsamen Kommunikation sind (erste Spalte). In diesem Kontext meint „shared objects“ deshalb sowohl die technisch gekoppelten Objekte als auch deren Bedeutung, die aus der Kommunikation heraus entsteht.

Realisierung des Szenarios	Verteilt, Synchron, “Shared Objects”	„Face-to-face“, “Shared Objects”	Asynchron, “Shared Objects”
Belvedere	Ø	+	–
CardBoard	Ø	+	–
Chips	+	+	+
Cognoter/ Argnoter	Ø	+	–
Csile	–	–	+
Dolphin	Ø	+	–
gIBIS	–	–	+
MindManager	+	Ø	+
OneNote	–	Ø	–
Sepia	+	Ø	+

Tabelle 6

Einordnung der Systeme in Bezug auf die Szenarien, die mit ihnen realisiert werden können.
+: Es handelt sich um das mit dem Tool angestrebte Szenario.
o: Das Szenario ist technisch möglich, wird aber nach der Toolbeschreibung nicht angestrebt.
–: Das Szenario wird technisch nicht unterstützt.

In einem „face-to-face“-Szenario (zweite Spalte), kann zwar technisch betrachtet synchron oder asynchron kooperiert werden, im Fall der asynchronen Kooperation ist aber die Tatsache, sich am gleichen Ort zu befinden, selten noch relevant, es sei denn dafür, dass schnell zwischen zwei Szenarien gewechselt werden kann. Im Fall des „face-to-face“-Szenarios wird deshalb davon ausgegangen, dass synchron kooperiert wird. Die „shared objects“ können dabei sowohl über technisch gekoppelte Objekte realisiert sein als auch über einen einzelnen gemeinsamen Fokus, ohne eine technische Kopplung vorauszusetzen.

Für den Fall einer asynchronen Kooperation wird in der dritten Spalte immer auch der Umgang mit „shared objects“ vorausgesetzt, auch hier nicht nur im technischen Sinn, sondern auch im Sinn einer Wissenskommunikation. Damit wären solche kooperationsunterstützenden Systeme ausgeschlossen, die beispielsweise nur gemeinsame Kalender anbieten.

Tabelle 7 ordnet die Systeme bezüglich ihrer Funktionalitäten ein, die hier als sinnvoll für kooperatives Wissensmanagement angesehen werden. Für die Bewertungen „+“ und „Ø“ wurde fast immer präzisiert, wofür sie vergeben wurden, da die Kriterien relativ allgemein sind und durch die unterschiedlichsten Ansätze ausgefüllt werden können. Ob ein „+“ oder ein „Ø“ vergeben ist, ist eine weitgehend subjektive Einschätzung, die sich daran orientiert, welche Technologien heute üblich oder möglich sind.

Organisatorische Einbettung	Interoperabilität	Repositories	Information-retrieval	Notifikation	Reflexion über den Wissensbestand	Einbettung in nicht-technischen Arbeitspozess
Belvedere	–	Ø (Lernmaterial)	Ø (Lernmaterial)	–	–	+
CardBoard	–	–	–	–	–	–
Chips	Ø (Integration externer Medien)	Ø (Hypertext)	Ø (Navigation)	Ø (E-Mails)	+ (Referenz, Integration von Dokumenten)	+ (Definition von Prozessen)
Cognoter/ Argnoter	–	–	–	–	–	+
Csile	Ø (Integration externer Medien)	+ (Datenbank)	+ (Anfrageorientiert)	+ (autom. Benachrichtigung bei Änderung)	+ (ständige Wissenskonstruktion)	+
Dolphin	Ø (Integration externer Medien)	Ø (Hypertext)	Ø (Navigation)	–	–	–
gIBIS	–	+	+ (Navigation und Suche)	–	+ (ist Ziel der Kooperation)	Ø (im Versuchs-Settings)
MindManager	Ø (Integration externer Medien)	–	–	Ø (E-Mail)	+ (Referenz, Integration von Dokumenten)	–
OneNote	+ (Export/ Import)	Ø (privates Repository)	+ (voraussichtlich)	– (nicht kooperativ)	+ (Referenz, Integration von Dokumenten)	–
Sepia	Ø (Integration externer Medien)	Ø (Hypertext)	Ø (Navigation)	–	–	–

Tabelle 7

Einordnung der Systeme danach, wie sie organisatorische Belange unterstützen.

Die Interoperabilität beschränkt sich in den meisten Fällen auf die Integration von externen Medien, was als Minimum angesehen werden muss. Oft handelt es sich oft um große, kompakte Systeme, die Ergebnisse, die mit ihnen erstellt werden, anderen

Systemen nicht zur Verfügung stellen. Unterstützt wird meistens der Prozess zum Wissensprodukt, wobei schließlich die Produkte nur vom Tool selbst gehandhabt werden können. Eine Art der Interoperabilität kann durch Import und Export von Standardformaten erreicht werden. Im Rückgriff auf die visuellen Sprachen können an dieser Stelle sogar „intelligente“ Dokumentationsprozesse angestoßen werden, die sich die Struktur der Wissensprodukte zu eigen machen, um daraus stärker formatierte Dokumente zu generieren oder Inhalte beispielsweise im Web zu publizieren. XML bietet sich heute für solche frei gestaltbaren Dokumente an.

Standardformate sind außerdem dann nützlich, wenn auf allgemeine Suchmaschinen zurückgegriffen werden soll. Dieser Anforderungen kommt von den aufgeführten Systemen keines nach, evtl. nicht einmal OneNote, das für die „Microsoft-Welt“ entwickelt wird.

Nach meiner Ansicht begrenzt die Verwendung einer eigenen Hypertextmaschine wie in Sepia, Chips und Dolphin die Einsatzmöglichkeiten. Die Hypertexte sind nur über die Maschine zugänglich und setzen diese für eine Kooperation voraus. Es sind deshalb technische sowie unter Umständen lizenzrechtliche Probleme zu erwarten, wenn das LAN verlassen wird und im WAN kooperiert werden soll. Eine Ausnahme stellt da das Web als Hypertextplattform dar, das in den genannten Fällen aber nicht als Plattform verwendet wird.

Eine Hypertextmaschine stellt die Navigation zur Verfügung, um den Informationsraum zu durchlaufen. Dabei ist man auf die Güte und Art der Links angewiesen. Für eine Pflege eines Repositories sind aber außerdem navigationsunabhängige Zugänge, z.B. durch Suchmaschinen, zu Inhalten sinnvoll, um neue Verbindungen herstellen zu können. Darin bestand für das Csile-System der Grund, sich für eine zentrale Datenbank zu entscheiden und keinen Hypertextansatz zu verfolgen.

Notifikationen werden kaum in die Arbeitsprozess mit einbezogen. Die einzige Ausnahme bildet das Csile-System. Generell können zwar E-Mails zur Benachrichtigung verwendet werden, manchmal ist diese Funktion auch in die Systeme integriert, sie sind aber nicht unbedingt als besonderes „feature“ zu werten, wenn nicht zumindest unter bestimmten Voraussetzungen Automatisierungen angeboten werden. Benachrichtigungen können einen asynchronen Arbeitsprozess maßgeblich steuern, indem sie auf Änderungen hinweisen, einen Arbeitsbedarf ankündigen, Interessengruppen realisieren, etc.

Vom Engineering-Standpunkt besteht ein deutlicher Bedarf an flexibleren Programmarchitekturen.

4 Diskussionsunterstützung: Eine Vorstudie zur Hypothesenbildung

4.1 Zielsetzung

Das zentrale Ziel der formativen Evaluation bestand in der Generierung von Hypothesen darüber, wie das technisch angereicherte Lernszenario das Kommunikationsverhalten der Beteiligten beeinflusst. Grundlage der Beobachtungen waren zwei Applikationen: eine Whiteboard-Software (siehe Abbildung 20) und das CardBoard (siehe Abbildung 21). Das CardBoard sollte den Mittelpunkt der Evaluation bilden, da der Einfluss der visuellen Sprachen auf die Kommunikationssituation eingeschätzt werden sollte.

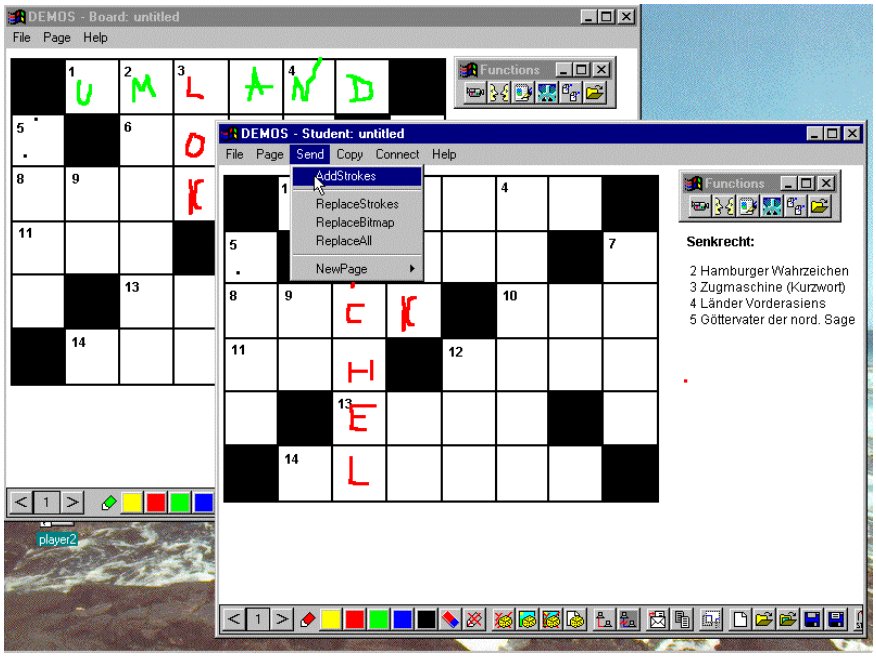


Abbildung 20

Aufwärmmaterial Kreuzworträtsel

Die Evaluation gliederte sich in zwei Teile, die Aufwärmphase und die Phase, die der eigentlichen formativen Evaluation mit dem CardBoard galt. Zur Vorbereitung wurden Stundenabläufe geplant, die die genannten Applikationen mit Hilfe von Unterrichtsmaterialien einbezogen.

Um beobachten zu können, wie die Applikationen das Gesprächsverhalten beeinflussen, sind die Materialien so vorbereitet worden, dass sie die Kommunikationssituation induzieren, das heißt dass sie den Ausgangspunkt oder die Grundlage für eine kollaborative Aufgabe bieten mussten. Die Materialien orientieren sich deshalb an der „jigsaw"-Methode (Aronson, 1978; Kagan, 1985), in der den Beteiligten unterschiedliche Informationen vorliegen, die dann in der Kooperation ausgetauscht werden.

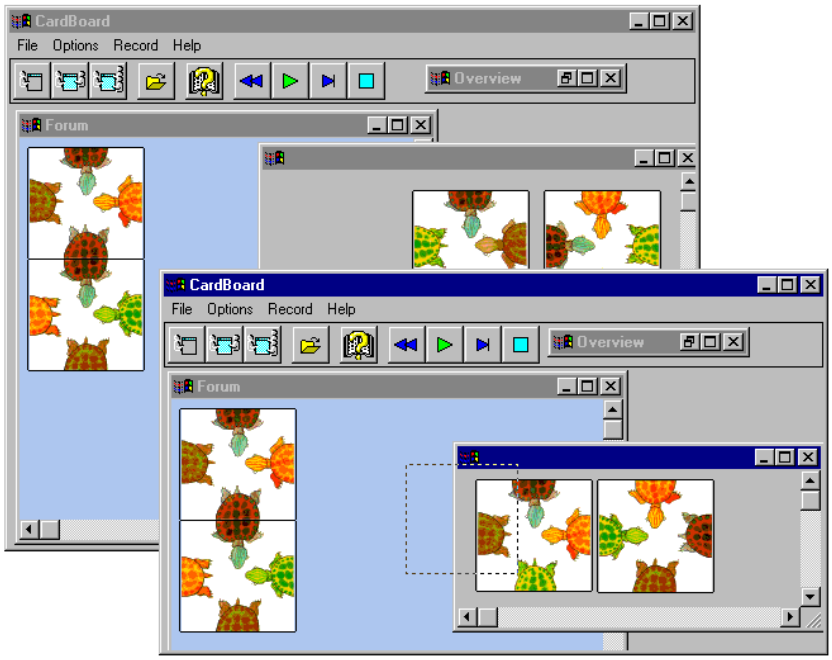


Abbildung 21

Aufwärmmaterial Schildkrötenspiel

Als Übung für das WhiteBoard sollte ein Kreuzworträtsel kooperativ gelöst werden, in dem jede beteiligte Person nur einen Teil der Fragen für die Lösungswörter erhielt. Mit einem „Schildkröten-Puzzle" wurde der Umgang mit dem CardBoard trainiert, bei dem den verschiedenen Personen in der Ausgangssituation unterschiedliche Kartenmengen zur Verfügung standen. Mit diesen Materialien konnten die Tools spielerisch erkundet werden.

Der anschließende Unterrichtsverlauf war noch einmal in zwei Phasen unterteilt. Im ersten Teil sollten sich die Probanden in die Diskussionsthematik einarbeiten. Dafür wurden für das WhiteBoard Materialen zum Thema Energiegewinnung angeboten, im CardBoard konnte ein „Energiedomino" gespielt werden. Erst anschließend sollte zum Thema Energiegewinnung mit Hilfe des CardBoards diskutiert werden. Die Unterrichtsverläufe wurden unter folgenden Gesichtspunkten durch Beobachterteams (2-3 Personen) frei protokolliert:

- Kommunikationsverhalten der Beteiligten,
- Unterstützung/Behinderung durch das System,
- Aufgabenstellung/Material,
- Moderation,
- Planung/Verlauf

Der Schwerpunkt der Beobachtung lag auf den positiven und negativen Einflüssen der Moderation und der Aufgabenstellungen. Die wesentlichen Erkenntnisse, die aus der Evaluation gewonnen wurden, basieren auf den Beobachtungsprotokollen und Anmerkungen der Beobachterteams.

4.2 Setting

Es wurde eine klassische Unterrichtssituation aufgebaut in der „face-to-face“ kommuniziert werden sollte. Das Evaluations-Setting beinhaltete eine interaktive, elektronische Tafel und Notebooks für alle Beteiligten, die miteinander vernetzt waren (vgl. Hoppe, Baloian & Zhao, 1993). Wegen ihrer geringen Höhe war mit den Notebooks der Sichtkontakt im Klassenraum nicht beeinträchtigt. Ebenfalls Bestandteil des Settings war eine selbst entwickelte, zur kooperativen Arbeit ausgelegte, WhiteBoard-Software und das bereits vorgestellte CardBoard in einer Vorgängerversion, die für die Diskussionsunterstützung adaptiert wurde. Für die Nutzung der Whiteboard-Software standen Tabletts zur Pen-basierten Freihandeingabe zur Verfügung. Die Tests wurden jeweils mit drei oder vier Probanden durchgeführt, die sich mit einer Lehrerin (Moderatorin) und einem Beobachterteam in einem Raum befanden. Alle Testgruppen wurden von der gleichen Person angeleitet. Die Probanden waren alle Studierende aus unterschiedlichen Fächern und hatten unterschiedliches Vorwissen im Umgang mit Computern.

Das CardBoard in der für die Evaluation verwendeten Form erlaubte „private“, „shared“ und „joint“ Arbeitsbereiche für die Teilnehmer und Teilnehmerinnen.

Im Vorfeld der Evaluation hatte sich herausgestellt, dass die Kombination dieser unterschiedlichen Kooperationsformen einen maßgeblichen Einfluss auf die Kommunikation ausgeübt hatten. Diese Einflüsse werden anhand von Abbildung 22 erläutert. Darin sind mögliche Kombinationen der Kooperationsmodi aufgeführt mit potentiellen Informationsflüssen zwischen den Workspaces. Die weißen Rechtecke repräsentieren darin die Workspaces der CardBoard-Instanzen und die umschließenden, grauen Rechtecke symbolisieren jeweils einen Computerbildschirm, wobei die elektronische Tafel

hervorgehoben ist („Board“). Die schwarzen Linien zeigen an welche Workspaces gekoppelt sind, ob gerade zwei („shared“) oder mehrere („joint“) Personen auf einen Workspace zugreifen können. Ist keine Linie dargestellt, so handelt es sich um einen „private“ Workspace. Die grauen Pfeile zeigen, von welchen Workspaces Daten in einen anderen transferiert werden und deuten damit Arbeitsphasen an.

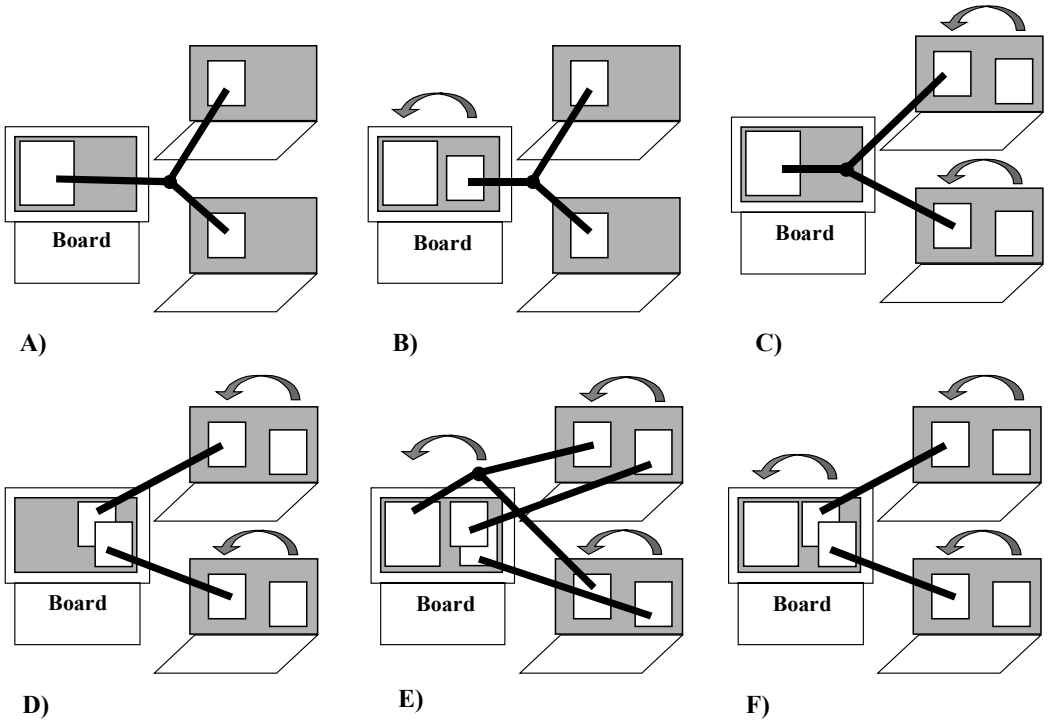


Abbildung 22

Arbeitsszenarien

Die folgende Liste nimmt Bezug auf die Fälle aus Abbildung 22:

A) *Ein „joint“ Workspace.* In dieser Situation hat keine der beteiligten Personen eine verstärkte Kontrolle über den gemeinsamen Arbeitsbereich. Sie hat sich als besonders geeignet für die ungeleitete Diskussion herausgestellt.

B) *Ein „joint“ Workspace; ein „private“ Workspace am Board.* Diese Situation kann als Lehrsituation ausgenutzt werden, in der die Lernenden gemeinsam Notizen anfertigen und begleitend oder anschließend relevante Inhalte herausgefiltert werden, die dann in den „private“ Workspace am Board übernommen werden. Charakteristisch wäre dafür ein geleitetes Unterrichtsgespräch, in dem aus der Diskussion und durch die Lehrperson beeinflusst, relevante Beiträge ausgewählt, strukturiert oder gewichtet werden.

C) *„Private“ Workspaces bei allen Beteiligten; ein „joint“ Workspace.* In dieser Situation werden Vorbereitungen aus „private“ Workspaces in einen gemeinsamen

Diskussionsbereich („joint“ Workspace) eingebracht. Dieses Setting wurde zuerst für die Evaluation bevorzugt, stellte sich dann aber als nicht geeignet für die angestrebte Diskussion heraus. Da die einzelnen Beiträge unabhängig voneinander vorbereitet wurden, gab es Probleme, sie später zu integrieren. Das bezog sich auf die Begrifflichkeit und auch die konkrete Diskussionslinie. Es passierte in den Vorversuchen, dass zwar alle Probanden ihre Meinungen vorbereitet hatten, über diese Aspekte später aber gar nicht diskutiert wurde. Letztlich wurden während der Diskussion völlig neue Beiträge formuliert. Durch die Vorbereitung war zwar jede Person gefordert, sich unabhängig Gedanken über das Diskussionsthema zu machen, was didaktisch eingesetzt werden kann, zu der erwarteten Zeitersparnis trug die Vorbereitung jedoch nicht bei. Die Erfahrungen, die zum Argnoter-System zitiert wurden (Kapitel 3.2), nach denen Vorbereitungen zu einem kontroversen Austausch führten, konnten hier nicht unmittelbar bestätigt werden.

D) *„Private“ Workspaces; „shared“ Workspaces.* Vorbereitungen aus „private“ Workspaces werden in „shared“ Workspaces kopiert und so am zentralen Board angezeigt. In diesem Beispiel dient die Kopplung der Präsentation von Einzelergebnissen.

E) *„Shared“ Workspaces; ein „joint“ Workspace.* Dieser Fall kann Fall D) zeitlich betrachtet fortsetzen, indem die präsentierten Ergebnisse noch gemeinsam überarbeitet und ggf. kombiniert werden. Ohne Fall D) vorher ist in dieser Form auch eine dyadische Vorbereitung in „shared“ Workspaces möglich mit anschließender gemeinsamer Diskussion und Integration.

F) *Individuelle „private” Workspaces und am Board; „shared“ Workspaces.* In diesem Fall wird, ähnlich wie beim geleiteten Unterrichtsgespräch, eine stark kontrollierte oder moderierte Integration von Einzelvorbereitungen herbeigeführt.

Die Arbeitsszenarien zeichnen sich vor allem dadurch aus, dass eine Lehrperson oder die Moderation, in Zusammenhang mit der öffentlichen Präsentation an der Tafel, unterschiedlich starken Einfluss auf die Ergebnisse ausüben kann.

Bei der formativen Evaluation haben wir uns für das Setting in Fall A entschieden, das der Zusammenarbeit und der Spontaneität einer offenen Diskussion Rechnung trägt. Die in diesem Szenario mögliche parallele Eingabe in das gemeinsame Forum hat sich letztlich bewährt und wurde generell positiv aufgenommen.

4.3 Verlauf

Für die Tests wurden Unterrichtsverläufe geplant, die die Nutzung der WhiteBoard-Software und des CardBoards beinhalteten. Die Testdurchläufe bestanden aus zwei vorbereitenden Phasen und den eigentlichen Testphasen (siehe Tabelle 8). Begonnen wurde mit einer technischen Einführung in die Software und deren Bedienung (10 bzw. 20 Min.). Darauf folgte eine „Aufwärmphase“ (20 Min.), in der die Probanden mit eigens vorbereiteten Materialien den Umgang mit den Applikationen übten. Dabei wurde auch Wert darauf gelegt, dass eine „face-to-face“-Kommunikation neben der Benutzung aufrecht erhalten wurde.

	Phase	Inhalt	Dauer 1 (Min)	Dauer 2 (Min)
1	Technische Einführung	1.Einführung in die technischen Devices (Pens, Tabletts, Touchpad) 2. Einführung der Software	ca. 10	20
2	Aufwärmphase	Benutzung des Kreuzworträtsels (Whiteboard) *oder* Benutzung des Schildkrötenpuzzle (CardBoard)	20	20
3	Inhaltliche Hinleitung zum Diskussions-gegenstand	Benutzung der Energieketten (CardBoard) *oder* Verwendung von Material zum Thema Energieproduktion (Whiteboard)	20	
4	Diskussion	Diskussion zur Energieproduktion *oder* Diskussion zur Universitätsreform	20	40

Tabelle 8

Zeitliche Struktur der Testdurchläufe

Nach diesen Einführungen folgte das eigentliche Unterrichtsgespräch, bei dem wiederum beide Applikationen mit einbezogen werden sollten (40 Min.). In Tabelle 8 beschreibt die Spalte „Dauer 1“ diese Planung. Die freie Diskussion während der letzten Phase wurde durch vorbereitete Materialien zum Thema Energiegewinnung eingeleitet.

Das in Abbildung 20 dargestellte Kreuzworträtsel war Bestandteil des Aufwärmmaterials und war sowohl für drei als auch für vier Personen vorbereitet worden. Alle Probanden bekamen dazu einen anderen Satz an Fragen, nach denen die Felder ausgefüllt werden mussten. Die elektronische Tafel diente als Sammelpunkt und Verteiler für die eingetragenen Begriffe. Wie zu erwarten war, mussten sich die Probanden mündlich absprechen, um die korrekten Eintragungen zu finden.

Abbildung 21 stellt das „Schildkröten-Puzzle“ dar, bei dem 9 Karten (12 Karten bei 4 Mitspielern) in einem gemeinsamen Workspace zu einem passenden Quadrat zusammengelegt werden mussten. Die Probanden hatten auch hier zu Beginn einen

unterschiedlichen Satz an „Schildkrötenkarten“ in privaten Arbeitsbereichen. Wesentliches Lernziel dieses Spieles war das Kennenlernen der privaten und öffentlichen Workspaces.

Der zweite Teil der Unterrichtsabläufe bezog sich auf das Thema Energiegewinnung. Hierfür wurden Text- und Bildmaterialien auf Whiteboard-Basis sowie ein „Energiedomino“ als visuelle Sprache für das CardBoard entwickelt, die eine spätere Diskussion zum Thema Energiegewinnung einleiten sollten. Das „Energiedomino“ stellte einzelne Karten zur Verfügung, die eine Umwandlung von einer Energieform in eine andere beschrieben: z.B. wird Licht durch eine Solarzelle in elektrische Energie umgesetzt. Diese Karten konnten aneinander gelegt werden und bildeten „Energieketten“, für die im Gespräch Beispiele gefunden werden sollten.

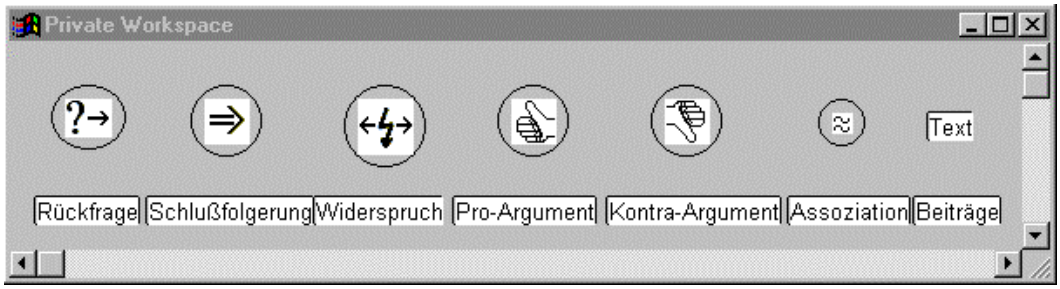


Abbildung 23

Visuelle Sprache zur Diskussionsunterstützung

Für die Diskussionsphase wurde die in Abbildung 23 dargestellte visuelle Sprache modelliert. Mit dieser Sprache konnten Rückfragen, Schlussfolgerungen und Widersprüche ausgedrückt werden. Pro- und Kontra-Argumente bezogen sich mit Hilfe des „thumb up“- und „thumb down“-Symbols auf andere Beiträge. Mit dem Assoziationszeichen konnten unspezifische Zusammenhänge ausgedrückt werden.

Die formative Evaluation beinhaltete 5 Tests, an denen insgesamt 17 Probanden teilnahmen.

Nach den ersten beiden Sitzungen zeigte sich, dass zu dem gewählten Thema ein zu großer Konsens bestand, der eine Diskussion verhinderte. Trotzdem sollte auf die Diskussion nicht verzichtet werden, da diese für die angestrebte Hypothesenbildung und damit die Zielsetzung der Evaluation notwendig war.

Aus diesem Grund wurde der Versuchsablauf noch einmal verändert und die Diskussion noch mehr in den Mittelpunkt gerückt. Dafür wurde auf die inhaltsbezogene Einführung aus dem Unterrichtsablauf verzichtet und der Diskussionsgegenstand durch einen anderen ersetzt, der stärker zur Diskussion anregte: Aus aktuellem Anlass wurde nun ein Streik

der Studierenden diskutiert. Mit diesem Thema war eine allgemeine Betroffenheit gegeben und die Probanden konnten gut motiviert werden.

Diese Änderung des Testverlaufes spiegelt sich in Tabelle 8 durch die Spalte „Dauer 2“ wider. Bei der Umstellung des Testverlaufes wurde außerdem mehr Zeit für die Erklärung der technischen Hilfsmittel eingeplant, da hier der Lernbedarf der Probanden unterschätzt worden war.

Tabelle 9 stellt genauer dar, wie die einzelnen Unterrichtssituationen bei den unterschiedlichen Testgruppen aufgebaut waren. Von Gruppe zu Gruppe wurde zwischen den Aufwärmmaterialien Kreuzworträtsel und Schildkrötenspiel gewechselt, um daraus auf auch eine Eignung der jeweiligen Applikationen Rückschlüsse ziehen zu können.

Test-gruppe	Phase 2	Phase 3	Phase 4
1	Benutzung des Schildkrötenpuzzle (CardBoard)	Verwendung von Material zum Thema Energieproduktion (Whiteboard)	Diskussion zur Energieproduktion
2	Benutzung des Kreuzworträtsels (Whiteboard)	Benutzung der Energieketten (CardBoard)	Diskussion zur Energieproduktion
3	Benutzung des Schildkrötenpuzzle (CardBoard)		Diskussion zur Universitätsreform
4	Benutzung des Kreuzworträtsels (Whiteboard)		Diskussion zur Universitätsreform
5	Benutzung des Schildkrötenpuzzle (CardBoard)		Diskussion zur Universitätsreform

Tabelle 9

Testverlauf bei den einzelnen Testgruppen

4.4 Auswertung

4.4.1 Kooperatives Editieren in den Workspaces

Koordination. Das Problem, dass mehrere Personen gleichzeitig versuchen, eine Karte oder eine Relation zu editieren, und sich so gegenseitig Eingaben überschreiben, ist in keiner der Versuchsgruppen aufgetreten. Das liegt mit großer Wahrscheinlichkeit an der relativ geringen zeitlichen „Dichte“ von Beiträgen bezogen auf die gesamte Gesprächsdauer. Als These ist auch zu formulieren, dass das Hinzufügen von Karten meistens an einen mündlichen Beitrag gebunden ist und deshalb automatisch auch koordiniert stattfindet.

Autorschaft. Da die Beteiligten an ihren Plätzen saßen und die Interaktion über die gekoppelten Workspaces stattfand, wurde nicht deutlich, wer gerade eine Eingabe machte, es sei denn, es wurde explizit gefragt. Dies war störend, da sich danach in der Kommunikation keiner an eine bestimmte Person wenden konnte, um etwas zu erwidern, sondern ins Auditorium gesprochen werden musste. Das kann Brüche im Diskussionsverlauf bewirken.

Intention. Störend war, dass Umstrukturierungen vorgenommen wurden, ohne sie mündlich zu motivieren. Dadurch war ihre Intention unklar. Im Ergebnis entstand „nur" ein neues „Bild", in das sich alle Probanden wieder neu einarbeiten mussten. Ob alle mit dieser Umstrukturierung einverstanden waren, wurde nicht ausgehandelt, was auch nicht technisch unterstützt wurde, beispielsweise durch ein qualitatives „undo" (Zusammenfassung mehrerer Schritte) zum Vergleich zweier Strukturierungsvorschläge. Automatische Strukturierungen könnten hilfreich sein, da über sie das Kriterium der Sortierung expliziert wird. Generell schien gerade bei der Restrukturierung ein besonderer Bedarf, diese zu erläutern. Diese kann also in keiner Weise parallel zu anderen Aktivitäten im Hintergrund stattfinden.

4.4.2 Verständlichkeit der Darstellung

Darstellung. Eine Relation besteht aus zwei Links und einem Symbol, das den Typ dieser Relation markiert (Ein Diskussionsergebnis wird in Abbildung 24 gezeigt.). Im CardBoard sind die Links in sich gerade. Zwei Links zusammen, die eine Relation bilden, stehen aber normalerweise in einen Winkel zueinander. Optisch wird dadurch erschwert, den Bezug der Links zu erkennen, insbesondere wenn Knoten über eine größere Entfernung verbunden werden.

Die Symbole waren nicht drehbar und widersprachen deshalb unter Umständen der Richtung einer Relation, was sich ebenfalls störend auf die Lesbarkeit der Darstellung auswirkte.

Bedeutung. Die durch die Symbole angedeutete Semantik unterschied sich trotz vorheriger Erläuterung zwischen Personen und Gruppen. Schwierigkeiten ergaben sich insbesondere bei der Verwendung des Schlussfolgerung-Symbols. Das Widerspruch-Symbol wurde sowohl als aktives Widersprechen, also als Feedback und damit Aufforderung zur Erwiderung verwendet, als auch als Feststellung einer unterschiedlichen Ansicht, also eher dokumentarisch. Die Pro- und Kontra-Argumente sind ebenfalls in einer nicht intendierten Form verwendet worden. Sie wurden z.B. als positive bzw. negative Meinungsäußerungen zum Grundthema verwendet.

Einer Gruppe muss also ausreichend Zeit gegeben werden, bis sich die Bedeutungen der Symbole aneinander angenähert haben. Daraus folgt, dass das Verstehen der Diskussionsergebnisse stärker als erwartet an die Gruppe gebunden ist, die sich die jeweilige Semantik erarbeitet hat.

Generell schien es, dass die Darstellungen nicht unbedingt selbsterklärend waren, dass die Darstellungen nicht sehr gut „lesbar“ waren, bzw. nur für die Gruppe einen Wert hatte, die sie erstellt hatte.

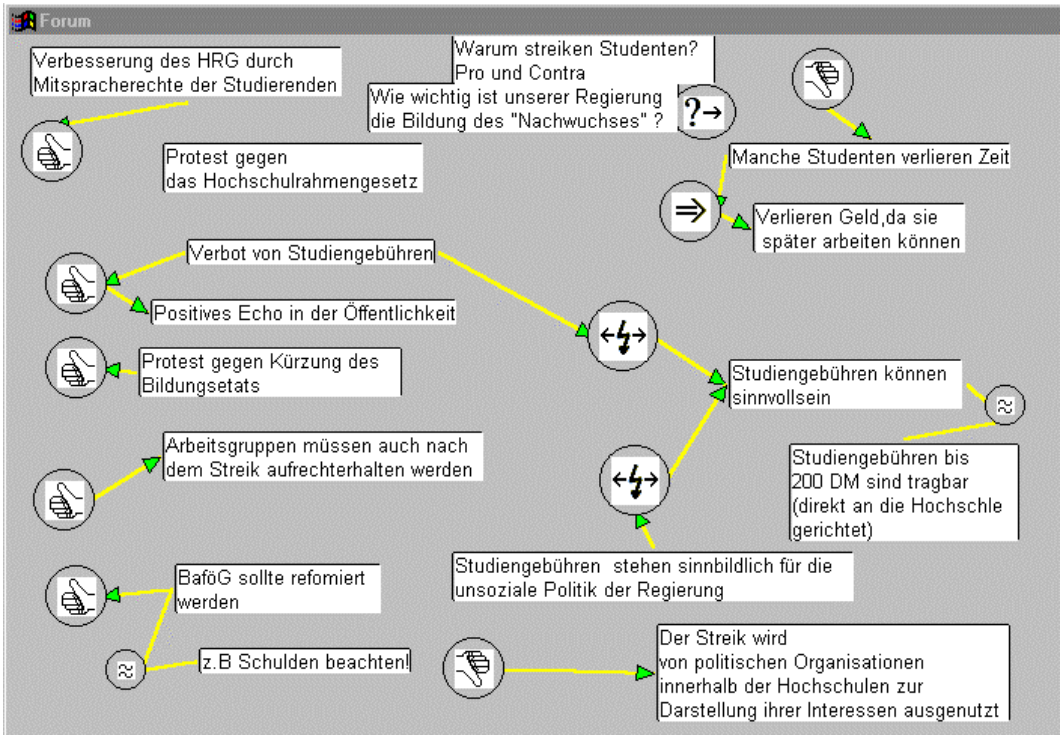


Abbildung 24

Ein Diskussionsergebnis.

Ergebnis. Ein Ziel der technischen Unterstützung besteht darin, am Ende einer Diskussion eine Ergebnisdokumentation zur Verfügung zu haben. Obwohl der entwickelte Graph als Protokoll der Diskussion akzeptiert wurde, ergab sich jedoch die Frage nach dem eigentlichen Fazit der Diskussion.

4.4.3 Einarbeitung

Die Aufwärmmaterialien stellten sich als motivierend heraus und schienen ein geeignetes Hilfsmittel zu sein, um die Software kennen zu lernen. Besonders positiv war, dass sie sich nicht auf eine bestimmte Domäne bezogen, sondern spielerisch an die Applikationen heranführten. Dadurch wurde eine Überlagerung einer inhaltlichen Fragestellung mit dem Erlernen der Handhabung vermieden. Im Vergleich zu Nutzungen in vorangegangenen

Seminaren wurden die technischen Schwierigkeiten im Umgang mit den Applikationen stark reduziert.
Es wurde deutlich, dass eine verlängerte Einarbeitungsphase verbesserte Diskussionsergebnisse bewirkte, da der Gesprächsfluss dann weit weniger gestört wurde.

4.4.4 Moderationsstrategie

Um eine angemessene Mitschrift der Beiträge zu erreichen, sind beispielsweise die folgenden Vorgehensweisen zu nennen: Sowohl eine kooperative, über die Personen verteilte Eingabe ist denkbar, als auch eine Protokollführung durch eine einzelne, dafür bestimmte Person. Letzteres widerspricht aber der Idee, das der „shared" Workspace eine gemeinsame Arbeitsgrundlage, einen Fokus und gemeinsamen Bezugspunkt darstellen soll.
Ein wirklich verteiltes Notieren als kooperative Aktivität war angestrebt, erwies sich jedoch als außerordentlich schwierig. Durch das Einbeziehen der Applikation wurde die Gesprächssituation im Vergleich zu einer traditionellen „face-to-face"-Situation komplexer. Diese Komplexität ergibt sich aus den unterschiedlichen Arten, sich nun beteiligen zu können, nämlich sowohl schriftlich als auch mündlich. Entsprechend ergeben sich in der Situation auch verschiedene Foki, auf die die Aufmerksamkeit gelenkt werden kann: Die elektronische Tafel, die Notebooks oder unterschiedliche Personen. Um mit diesen Problemen umgehen zu können, entstanden teilweise durch Gruppeninitiative, teilweise durch die Moderation, ad hoc Strategien, die sich als unterschiedlich erfolgreich erwiesen.
Eine dieser Strategien bestand darin, jeden Beitrag sofort in das gemeinsame Forum niederzuschreiben, nachdem er geäußert wurde. Für die jeweils Schreibenden war es dann aber nicht mehr möglich, auf Erwiderungen zu achten. Üblicherweise bezieht sich ein Folgebeitrag in einer Diskussion aber auf den vorangegangenen. Mit dieser Strategie konnte also das jeweilige Feedback nicht aufgenommen werden, so dass als Folge die Diskussion zusammenbrach.
Daraufhin wurde es den Beteiligten überlassen, wann sie welchen Beitrag notierten. Dabei ergab sich, dass zu wenig und teilweise nicht die relevanten Beiträge festgehalten wurden. Außerdem geschah das zu einem recht undefinierbaren Zeitpunkt. So entstand zusätzlich das Problem, dass man Bezüge zu anderen, bis dahin nur mündlichen Äußerungen, nicht darstellen kann.
Eine Lösung ergab sich in Form einer zyklischen Strategie (Abbildung 25), die sich in drei Phasen aufteilt:

1. Freie Diskussion ohne Notizen (Aushandeln der Inhalte)
2. Gleichzeitiges Niederschreiben von Beiträgen in Inhaltskarten (Akkumulation)
3. Diskussion, über die Bezüge zwischen den Beiträgen (gegenseitiges Erklären, Dissens- / Konsens-Feststellungen)

In der ersten Phase wurde frei diskutiert, ohne schriftliche Beiträge zu formulieren. Wurden in vorangegangenen Iterationen bereits Beiträge an der elektronischen Tafel erstellt, so war es erwünscht, darauf mündlich Bezug zu nehmen.

Nach einer bestimmten Zeit ging man in die zweite Phase über. Darin notierten alle Beteiligten gleichzeitig im gleichen Workspace (Forum) die Äußerungen, an die sie sich erinnerten. Durch die Zeitverzögerung zwischen mündlicher und schriftlicher Äußerung wurden die eher interessanten bzw. relevanten Beiträge gefiltert. Bestehende Beiträge konnten durch detailliertere Darstellungen ersetzt oder ergänzt werden. Durch die parallelisierte Eingabe ergab sich außerdem ein relativ geringer Zeitaufwand bei den Notizen.

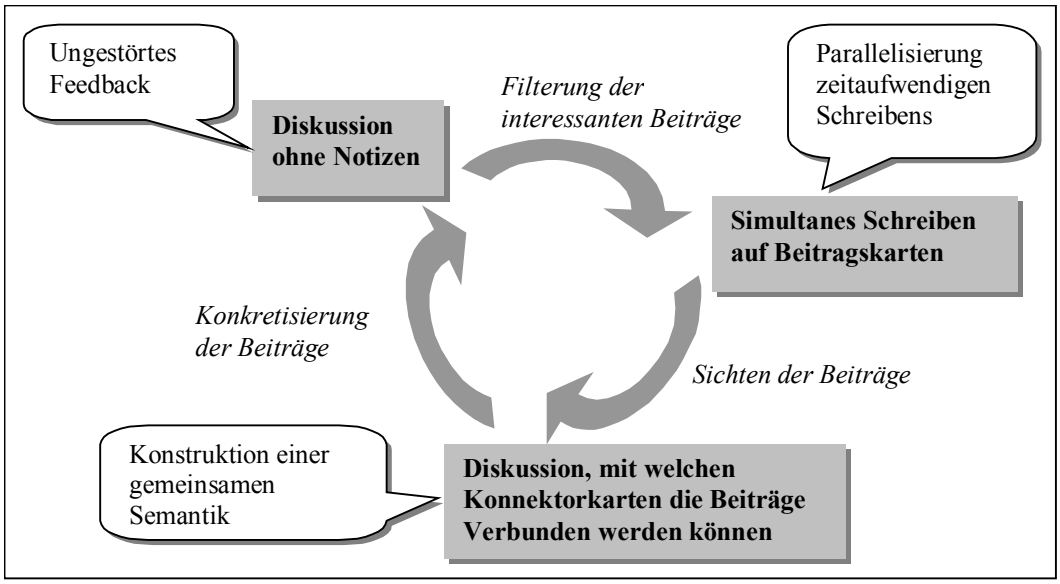


Abbildung 25

Kommunikationszyklus

Um die Beiträge gemeinsam mit der Gruppe mit Hilfe der vorhandenen Konnektorkarten in Relation zu setzen (dritte Phase), mussten die einzelnen Beiträge gelesen und unter Umständen kurz erläutert werden. Dabei wurden die Gruppenmitglieder mit den neuen Beiträgen vertraut, redundante Beiträge konnten zusammengefasst oder gelöscht werden. Mit Hilfe der Gespräche über die schriftlichen Beiträge und die Relationen erarbeitete sich die Gruppe außerdem ein gemeinsames Verständnis der Konnektorkarten. Aus dem

Gespräch ergab sich dann direkt ein eleganter Übergang zu einer neuen Iteration im Zyklus, also zu einer neuen Diskussionsphase.

Dass sich eine solche zyklische Strategie herausgebildet hat, deutet eine enge Relation zur Sicht auf kooperatives Arbeiten an, wie es in Roschelle und Teasley (1995) eingeführt wird. Dort wird die Akkumulation einer gemeinsamen Grundlage von Wissenselementen genauso als Teil der Zusammenarbeit gesehen wie das Verhandeln darüber. Auch in Baker (1995) werden Problemlöseprozesse als Aushandlungen betrachtet, dort jedoch nur in dialogischen Situationen. Es zeigte sich allerdings, dass gerade die Akzeptanz oder Ablehnung von Beiträgen in der Gruppensituation offensichtlich anders verläuft, als in der dialogischen Situation. Wirkt sie im Dialog als zentrales steuerndes Element, so wird in der Gruppensituation scheinbar erst einmal wenig Akzeptanz oder Ablehnung zu den notierten Beiträgen geäußert.

Um Akzeptanz- oder Ablehnungsäußerungen zu provozieren, hat sich das Löschen von Beiträgen als nützliche Methode ergeben. Das Löschen stellte eine explizite Ablehnung eines Beitrages dar und führte so zu erheblichem Widerspruch, wenn Beteiligte den gelöschten Beitrag unterstützten. Diese Erkenntnis steht im Widerspruch zu Lehmann (1998), der betont, dass eine Visualisierung von Inhalten auch eine Akzeptanz der Gruppe bzgl. der festgehaltenen Punkte beinhaltet. Nach Erfahrungen mit unseren Diskussionsgruppen ist dies im verteilten Szenario nicht der Fall. Hier besteht vermutlich ein entscheidender Unterschied zwischen einer verteilten Eingabe und dem Festhalten von Punkten durch einen Moderator oder eine Moderatorin.

Die Schwierigkeiten bei der Einbindung der Diskussionsunterstützung lassen vermuten, dass bei deren Verwendung ein völlig neues Kommunikationsmodell erlernt werden muss.

5 ANSÄTZE ZUR UNTERSTÜTZUNG KOOPERATIVER WISSENSELABORATION

Wissenselaboration wird hier als Prozess verstanden, in dem Informationen so aufgegriffen werden, dass sie in Entscheidungen, Planungen, Argumentationen, etc. einfließen und gleichzeitig dokumentiert werden können. Es handelt sich um einen Reflexionsprozess und damit auch um einen Lernvorgang, dessen Ergebnisse jeweils wieder in den Arbeitsprozess einbeziehbar sein sollten.

In dieser Arbeit wird die Frage verfolgt, wie die Wissenselaboration so unterstützt werden kann, dass sie sich in den alltäglichen Arbeitskontext integrieren lässt. Das wirft Fragen danach auf, unter welchen Umständen Wissen elaboriert wird, wie dieses Wissen aussieht und welche Intentionen damit verfolgt werden.

- *Szenario.* In dieser Arbeit werden Diskussionen als relevantes Szenario eingestuft, um Wissen zu entwickeln, denn sie eignen sich im besonderen Maße, um implizites Wissen zu äußern (van Joolingen, 2000). In den Diskussionen steht nicht die Abkopplung von Wissen von Personen, sondern die Externalisierung von Gesprächsinhalten im Vordergrund. Auf diese Weise sollen implizite Wissensressourcen erschlossen und in die Arbeitsabläufe eingebracht werden. Durch Argumentationen, Nachfragen und Erklärungen werden darin unmittelbar Strategien umgesetzt, die sonst für die Wissenskonstruktion und zur Reflexion eingesetzt werden (VanLehn, Jones & Chi 1992; Chi et al., 1994; Suthers, Toth & Weiner, 1997). Weiterhin eignen sich Diskussionen als „natürliche Notizsituationen“ insofern, als dass auch in herkömmlichen Diskussionen Skizzen und Notizen als Hilfsmittel verwendet werden (Lehmann, 1998). Es handelt sich also um ein bekanntes und eingeübtes Vorgehen. Ein weiteres Argument besteht in der Häufigkeit, mit der Diskussionen vorkommen, nach Bargiela-Chiappini & Harris (1997) bilden sie z.B. ca. 70% einer Manager-Tätigkeit.
- *Wissen.* Das Wissen, das in Diskussionen externalisiert wird, bezeichne ich als dynamisches Wissen: Es entsteht ad hoc, an vielen Orten und Zeitpunkten (Teamsitzungen) und hat unter Umständen keine lange Dauerhaftigkeit. Je allgemeiner verständlich eine Dokumentation sein soll, desto aufwendiger ist es, sie zu produzieren, desto weniger enthält sie aber auch dynamisches ad hoc-Wissen. Ob und

welche Bestandteile wichtig sind, zeigt sich unter Umständen erst daran, dass sie wieder verwendet werden.

Der Prozess der Wissenselaboration kann nicht mit einem Wissensenginieering-Ansatz gleichgesetzt werden, wie er zur Entwicklung wissensbasierter Systeme durchgeführt wird. Zum einen kann das Wissen im Alltag nicht so vollständig erhoben werden, zum anderen sind die Inhalte nicht in gleicher Weise an eine umgrenzte Domäne gebunden.

In dieser Arbeit wird angestrebt, die Übergänge von informellen Bestandteilen, über semi-formale hin zu vernetzten Informationsstrukturen zu unterstützen. Das Ziel besteht in einem Wissensbildungsprozess wie er in Nonaka (1994), Stahl (2000) und Borghoff und Pareschi (1998) beschrieben wird und auf der Überarbeitung und Reflexion sowie auf der Externalisierung sowie der Kommunikation von Inhalten aufbaut.

- *Intentionen.* Es werden hier zwei zentrale Intentionen bei der Unterstützung von Diskussionsprozessen gesehen: 1) Die Unterstützung eins *kreativen* Herangehens, bei dem es darum geht, offene Fragen zu klären, Probleme zu lösen, Entscheidungen zu treffen oder Inhalte zu verstehen, um auf die Qualität und die Plausibilität (vgl. Kapitel „Einleitung und Überblick“, Abschnitt „Wissenselaboration“) der Inhalte einzuwirken. 2) Die Unterstützung eines *dokumentarischen* Herangehens, in dem Ergebnisse festgehalten werden, die der Reflexion und Wiederverwendung offen stehen, um die Nachhaltigkeit der Ergebnisse zu verbessern.

Im zweiten Teil dieser Arbeit werden zwei Systementwicklungen vorgestellt, das DiscBoard und der FreeStyler, die im Schwerpunkt jeweils einer dieser Intentionen folgen.

Das DiscBoard zielt auf die Unterstützung einer konkreten, zeitlich begrenzten Diskussionssituation, in der kreativ an offenen Fragen, Entscheidungen und Problemen gearbeitet wird. Dafür werden Perspektiven eingesetzt, die in ihrer Idee den „epistemic forms“ nach Collins und Ferguson (1993) entsprechen, die alleine durch ihre Struktur bzw. Form bestimmte Hinterfragungsstrategien anleiten. Vergleichbare Ansätze finden sich in Carlson (1995) und in Scardamalia, Bereiter und Steinbach (1984).

Solche erkenntnisleitenden Strukturen werden auch in verschiedenen Systemen umgesetzt, wie beispielsweise in Sepia, Belvedere, gIBIS, Chips und in ganz einfacher Form dem MindManager (vgl. Kapitel 3.2). Die meisten Systeme setzen allerdings nur eine epistemische Form ein, um ein ganz gezieltes Ergebnis zu erhalten, wie Argumentmuster oder Bäume mit getypten Knoten. Um einen Arbeitsprozess zu unterstützen, reicht eine einzelne Darstellungsform aber selten aus. Systeme wie Sepia und Chips und als

Implementierungsansatz auch das Broca-System (Carlson, 1995) zielen deshalb auf die Unterstützung eines Prozesses durch unterschiedliche visuelle Sprachen.

Die Entwicklung des DiscBoards schließt daran an, indem auch der Übergang zwischen den Arbeitsphasen unterstützt werden soll. Darunter ist die Übernahme von Daten aus einer Arbeitsphase in eine andere zu verstehen und damit integriert eine Filterfunktion. Die angestrebten epistemischen Formen werden herangezogen, um daraus Filteroptionen für Strukturen und Sortierungen zu bestimmen.

Dieses Herangehen wird in den Kontext der folgenden Forschungsrichtungen und Ansätze eingebettet:

- Collins und Ferguson (1993) begründen den Ansatz der *epistemischen Formen* und Spiele, die an bestimmte strukturorientierte Darstellungsformen gebunden sind. Diese werden regelmäßig und theorieübergreifend zur Analyse von Phänomenen verwendet und begleiten sinnvoll wissenschaftliche Herangehensweisen.
- Jonassen und Carr (2000), Lajoie (1993) und Pea (1985) führen die Kategorie der *Mind Tools* ein, mit denen Denkprozesse unterstützt werden sollen. Diese verwenden beispielsweise strukturelle, logische, kausale oder systemische Repräsentationshilfen, mit denen die Wissenskonstruktion ermöglicht wird. Dieses konstruktive Herangehen soll zum kritischen, kreativen Denken anregen sowie komplexere Vorgehensweisen zur Problemlösung nahe legen, wie beispielsweise das Analysieren, das Synthetisieren oder die Entscheidungsfindung. Speziell für die Diskussionsunterstützung werden reflexionsorientierte Ansätze als sinnvoll erachtet, da für die Ergebnisse typischerweise keine absoluten Maßstäbe existieren, sondern es darauf ankommt, die relevanten Aspekte unter geeigneten Perspektiven und Maßgaben zu diskutieren.
- Suthers (1999A; 1999B; 1999C; 2003) führt das Konzept des „*representational bias*" ein. Der Einfluss durch die Repräsentation besteht in der Art und Weise wie Inhalte dargestellt werden und welche Gegebenheiten in den Vordergrund gestellt werden. Es ist deshalb davon auszugehen, dass für verschiedene Arbeitsziele, Aufgaben oder Probleme unterschiedliche Darstellungsmittel zur Verfügung gestellt werden sollten (vgl. auch Carlson, 1995; Cooke, 1994; Jonassen & Carr, 2000; Jüngst & Strittmatter, 1995; Streitz, Haake et al., 1998).

Der FreeStyler fokussiert auf einen Dokumentationsprozess, der aber wieder davon ausgeht, dass ein wesentlicher Teil der relevanten Informationen in Kommunikationssituationen anfällt. Allein dadurch ist er beispielsweise von Dokumenten- oder Contentmanagement-Systemen abzugrenzen, die im Wesentlichen die technischen Aspekte der Verwaltung und des Retrievals von Daten zur Verfügung stellen. Das Ziel besteht darin,

eine Lücke zwischen organisatorischen, nicht-technische Maßnahmen des Wissensmanagements und diesen technischen Ansätzen zur Datenverwaltung zu schließen.

Mit dem System FreeStyler sollen vor allem Möglichkeiten bereitgestellt werden, um ein „group memory" aufzubauen. Im Vergleich zum DiscBoard bietet er komfortablere Gestaltungsmöglichkeiten mit potentiell interaktiven Objekten. Die visuelle Sprache verbindet Objekte, die zur Strukturierung von Inhalten eingesetzt werden können, mit der Möglichkeit digitale handschriftliche Notizen anzufertigen. Der FreeStyler erlaubt deshalb einen sehr breit gefächerten Einsatz von ad hoc erstellten Notizen über die Unterstützung von Kreativsitzungen hin zur Darstellung von Inhalten, die auch für den langfristigen Gebrauch übersichtlich aufbereitet sind.

Als Hintergrund für diese Entwicklung werden die folgenden Ansätze hervorgehoben:

- Nonaka (1994) sowie Nonaka und Takeuchi (1995) beschreiben die Art und Weise sowie die Relevanz der Fortentwicklung des Wissens in einer Organisation. Danach wird Wissen zyklisch externalisiert, kombiniert, internalisiert und sozialisiert. Für die Kombination muss das Wissen explizit vorliegen, die Sozialisation, vor allem also die Verbreitung und Anwendung, basiert im wesentlichen auf dem impliziten („tacit") Wissen. Die Externalisierung und die Internalisierung stellen jeweils einen Übergang zwischen implizitem und explizitem Wissen dar.

  Diese Sicht bildet den Ausgangspunkt vieler Ansätze des Wissensmanagements, die den Austausch von Wissen und in letzter Konsequenz dessen Einfluss auf Produktionsprozesse als zentrales Element moderner Organisationsstrukturen ansehen.
- Borghoff und Pareschi (1998) greifen diesen Ansatz auf und entwickeln dazu technisch ausgerichtete Vorschläge für eine Wissensmanagement-Architektur. Darin steht der Wissensfluss als wichtigstes Ziel im Zentrum. Dieses eher abstrakte Ziel wird technisch durch drei Komponenten gestützt:
  - Wissensrepositories
  - Wissensnavigation und –konstruktion sowie Modellbildung
  - Wissensgemeinschaften, Wissenskommunikation, Kooperation

Sowohl der FreeStyler als auch das DiscBoard sind als *kognitive Tools* einzuordnen (Jonassen, 1992; Lajoie, 1993; Jonassen & Carr, 2000; Zhang, 1997; Pea, 1985) und beide verwenden semi-formale visuelle Sprachen zur Repräsentation von Inhalten. Im Sinne von Lakin (1990) werden sie als *Kommunikationsmedien* eingesetzt. Die visuellen Sprachen werden genutzt, um einen Übergang von informellen zu formaleren Beschreibungen zu finden. Das bedeutet, dass informelle Bestandteile in semi-formale Darstellungen integrierbar sein müssen. Dafür werden in beiden Systemen Container-

Objekte als Sprachprimitive zur Verfügung gestellt, in die Inhalte notiert werden können. Diese werden als Knoten in einer Graphstruktur verwaltet. Eine visuelle Sprache besteht aus einer Menge von Knoten- und Relationstypen. Diese Menge und die Möglichkeit daraus Graphen zu entwickeln, stellt die Syntax einer visuellen Sprache dar.
Gemäß der Beschreibung von Suthers (1999A) sind visuelle Sprachen repräsentationale Notationen in Form einer Menge primitiver Elemente, aus denen mit einem repräsentationalen Tool repräsentationale Artefakte konstruiert werden können, die Ausdrücke einer visuellen Sprache. Formen, Symbole und Farben werden genutzt, um den Inhalten zusätzliche Informationen beizufügen. Die Syntax der visuellen Sprachen beschreibt also tendenziell eine Symbol-Sprache, die der Strukturierung und Klassifikation von Inhalten dient.
Im Folgenden werden die Zielstellungen, Voraussetzungen und Hypothesen der beiden Systeme kurz präzisiert.

5.1 Konzepte der Systeme DiscBoard und FreeStyler

5.1.1 DiscBoard

- Es sollen unterschiedliche *visuelle Sprachen* zur Verfügung gestellt werden. Entsprechend der Idee der „*epistemic forms*“ sollen sie unterschiedliche Arbeitsphasen begleiten.
- Diese visuellen Sprachen stellen *Perspektiven* auf die notierten Inhalte zur Verfügung.
- Die nach Lehmann (1998) *für Diskussionen typischen Phasen* „Eröffnung“, „Arbeit am Problem“, „Abschluss“ sollen unter diesen Perspektiven durchlaufen werden können. Dafür wird die folgende Gruppe visueller Sprachen angeboten: Die *Protokollsprache* wird zur Diskussionsplanung und –zusammenfassung verwendet. Die *Argumentations-* , die *Pro-/Kontra-*, die *Explorations-* und die *Brainstormingsprache* dienen der kooperativen Reflexion und sind zum konkreten Umgang mit dem Diskussionsgegenstand entwickelt worden. Nach Spada und Rummel (2002) unterstützen diese Sprachen die *Mikrostruktur einer Kooperation*. Insbesondere die Argumentationssprache, die Pro-/Kontratabellen und die Brainstormingsprache zielen auf bestimmte epistemische Formen ab und lassen eine strukturierte Übersicht über Inhalte zu. Sie dienen deshalb zusätzlich einer abschließenden Darstellung.

- Die Inhalte eines Workspaces können einem Thema zugeordnet werden. *Zu jedem Thema kann genau ein Workspace pro Perspektive geöffnet* werden.
- Die Perspektiven stellen *keine abgeschlossene Auswahl* dar, sondern sind als Beispiel für einen Perspektivenansatz zu verstehen.
- Es wird davon ausgegangen, dass während einer Diskussion relativ viele Notizen anfallen können. Deshalb wird zum Überblick eine *übergeordnete Sicht* auf die Diskussion generiert, die „Topic Map". Darin fließen die vergebenen Themen ein.
- Eine Gemeinsamkeit aller visuellen Sprachen besteht darin, dass sie das *Sammeln*, *Reflektieren* und *Zusammenfassen* von Beiträgen mindestens im Vergleich zu nicht technisch angereicherten Szenarien unterstützen. Diese drei Phasen bilden nach Spada und Rummel (2002) die *Makrostruktur einer Kooperation*.
- Zwischen den Perspektiven wird eine Übernahme und Interpretation von Daten definiert, die auf einem *Diskussionsmodell* basieren. Dabei werden Inhalte von Workspaces automatisch gefiltert und in andere aufgenommen. So sollen *offene Diskussionspunkte* visuell kenntlich werden, wenn in einem Workspace unvollständige Strukturen oder unverbundene Objekte enthalten sind.
- Der *Datenfluss zwischen den Perspektiven* wird als *Abbildung* bezeichnet. Es werden sowohl einzelne Beiträge abgebildet, das heißt in eine andere Perspektive übernommen, als auch Strukturen, das sind durch Relationen verbundene Gruppen von Beiträgen.
- Perspektiven können als Workspace-übergreifende Sichten definiert werden, die Inhalte *filtern* und *sortieren*. Implementiert ist dazu eine *Erkennung und Abbildung von Strukturen*. Die konkreten Abbildungen werden im Diskussionsmodell definiert, auf denen der *Abbildungsalgorithmus* operiert.
- Die Abbildung von Beiträgen sowie Strukturen, führt zu einer *internen Bindung der Objekte*, die nun inhaltsgleich in verschiedenen Perspektiven vorliegen. Diese Bindung wird in *Knotenclustern* verwaltet.
- Die Abbildung von Strukturen führt zur Verwaltung von *Strukturmodellen*, die die Übereinstimmungen der Strukturen zwischen den Perspektiven verwalten.
- Der Datenfluss zwischen den Perspektiven soll eine Grundlage für weitere Arbeitsschritte liefern. Da die Idee der Perspektiven aber gerade bedeutet, den gleichen Themenbereich unter verschiedenen Maßgaben zu betrachten, ist nicht davon auszugehen, dass Inhalte in unterschiedlichen Perspektiven konsistent entwickelt werden und austauschbar sind. Es wird von „*divergierenden Inhalten*" ausgegangen, die explizit *nicht konsistent* zwischen den Perspektiven gehalten werden.

- Trotz der Annahme, dass die Inhalte divergieren, soll es den Nutzerinnen und Nutzern überlassen werden, den Zeitpunkt zu bestimmen, ab wann sich die Inhalte auseinander entwickeln sollen. Es wird ein Konzept der „*delayed changes*“ implementiert, das in parallelen Workspaces anzeigt, wenn Änderungen vorgenommen wurden. Diese können übernommen oder abgelehnt werden. Vom Zeitpunkt einer Ablehnung, wird die interne Bindung der Objekte aufgehoben. Die Abbildung von Strukturen, die sich in den Perspektiven unterschiedlich entwickeln können, führt zu einer relativ komplexen *Verwaltung der Strukturmodelle*.

5.1.2 FreeStyler

- Der FreeStyler soll in unterschiedlichen Arbeitsphasen einsetzbar sein: Für individuelle und gemeinsame Notizen und Visualisierungen, für Sitzungen und Präsentationen einschließlich deren Vor und -nachbereitung und zur Dokumentation. Themen- und vorgangsbezogene FreeStyler-Dokumente sollen diese Phasen begleiten. Die *Nutzungsvarianten* sollen die *Wiederverwendung* der Inhalte fördern. Ein solcher Prozess, der Argumentationen, Begründungen und die Bildung von Zusammenhängen einbezieht, wird von Stahl (2000) als „knowledge-building process“ charakterisiert.
- Angestrebt wird ein *Dokumentationsprozess*, der *kommunikative Situationen* als Ausgangspunkt vorsieht, und in dem man über eine zuerst informelle Darstellung zu einer Repräsentation mit formalen Bestandteilen gelangt, von *dynamisch-informellen* Bestandteilen, zu statischen bzw. *dauerhaften Bestandteilen*.
- Durch die themenspezifische Wiederverwendung von FreeStyler-Dokumenten wird inkrementell ein „*group memory*“ aufgebaut. Durch die gemeinsame Nutzung des Dokumentenbestand kann sich verstärkt auch eine gemeinsame Sprache entwickeln, Ideen und Beispiele stehen allen zur Verfügung.
- Durch die Wiederverwendung soll ein *Zyklus der Externalisierung, Kommunikation und Internalisierung* angestoßen werden, wie er bei Nonaka (1994) als Voraussetzung für ein Wissensmanagement angesehen wird.
- Die Externalisierung soll durch einfache Verwendung in *bekannten Situationen* unterstützt werden. *Digitale, handschriftliche Notizen* stellen in vielen Situationen eine Grundlage für die Visualisierung von Gesprächsinhalten dar. Der FreeStyler soll die Möglichkeit bieten, Inhalte sehr frei zu gestalten.

- Eine *visuelle Sprache* stellt die Mittel zur *Strukturierung und Klassifizierung* der Inhalte zur Verfügung. Containerobjekte fördern eine einfache *Standardisierung* der Darstellungen.
- Inhalte der FreeStyler-Dokumente sollen zusammen mit externen Dokumenten zu *Informationsnetzen* strukturiert werden können. Daraus entsteht eine *übergeordnete Sicht auf Dokumentenarchive*, in denen Dokumente vielfältig in Kontexte eingebettet werden können.
- FreeStyler verbindet interne und externe Links mit einer seitenorientierten Dokumentstruktur. Das bedeutet, dass unterschiedliche Strukturierungsprinzipien verfolgt werden können: 1) Es wird ein *Hypertextsystem* zur Verfügung gestellt, mit dem sogar getypte Referenzen durch die User definierbar sind. 2) Durch das Seitenformat können *lineare*, *buchartig strukturierte Inhalte* aufgebaut werden. 3) Durch *interne Links* können themenbezogene Inhalte auch innerhalb eines Dokumentes netzartig strukturiert werden.
- Die Strukturierungsmöglichkeiten ermöglichen, Inhalte *bedarfsorientiert* zu (re)strukturieren, zu gruppieren, zu verwalten und zueinander in Bezug zu setzen.
- Die *Suche und Wiederverwendung* von externen Dokumenten sowie von FreeStyler-Dokumenten wird durch eine integrierte Suchmaschine unterstützt. Suchergebnisse werden als Linklisten mit aktiven Referenzen auf Dokumente im FreeStyler dargestellt.
- FreeStyler-Dokumente können *verschlagwortet* werden. Ergebnisse von Suchen auf den Schlagworten werden als Struktogramme im FreeStyler dargestellt. Die Schlagworte werden mit dem Dokument gespeichert und stehen damit selbst weiteren Suchanfragen offen.
- Das Ablageformat von FreeStyler-Dokumenten soll *retrieval-fähig* sein. FreeStyler-Dokumente werden im *XML-Format* abgelegt und stehen u.a. durch deren interne Strukturierung auch *weiteren Verwendungen offen*.
- Suchen und Navigation sind als *handlungsorientierte Perspektiven* einzuordnen. Bei einer Suche bzw. beim Retrieval werden die Suchkriterien jeweils neu und individuell angegeben. Die jeweiligen Suchkriterien können als Perspektive gesehen werden.

5.2 Systemarchitektur

Beide Systeme stellen eine replizierte Architektur zur Verfügung, in der jede Instanz des DiscBoards bzw. FreeStylers eine eigenständige Applikation ist (vgl. Abbildung 26). Die

Kommunikation für die Kopplung von Arbeitsbereichen wird in beiden Fällen über den MatchMaker-Server realisiert. Im Falle des DiscBoards wird eine C++-Version verwendet, die zusätzlich die Kommunikation mit Dalis leistet. Für den FreeStyler wird MatchMaker TNG, eine in Java implementierte Version verwendet (Tewissen, Baloian, Hoppe & Reimberg, 2000; Jansen, Pinkwart & Tewissen, 2001; Jansen, 2002).
Die Konzepte der Kopplung sind jeweils unterschiedlich: MatchMaker TNG verwaltet Modelle gekoppelter Objekte, der C++-MatchMaker organisiert hauptsächlich die Kommunikation zwischen den Applikationen.

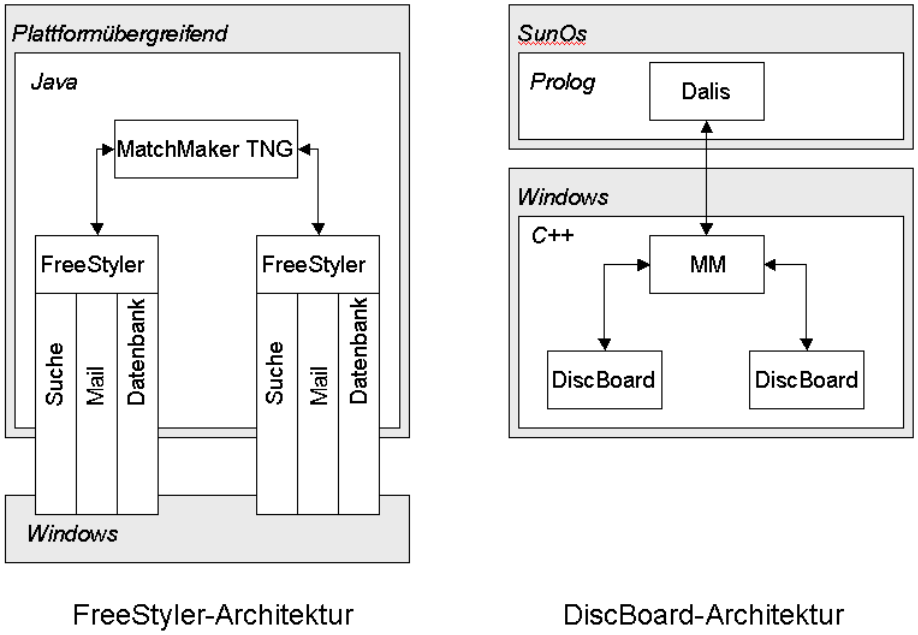


Abbildung 26

Komponenten, Plattformen und Entwicklungssprachen der zwei vorgestellten Systeme.

Die DiscBoard-Architektur, die als Test-Architektur gewertet werden muss, ist im Verhältnis komplexer, da sie unterschiedliche Plattformen einbindet. Es wird eine bestehende Architektur verwendet (vgl. Mühlenbrock, 2001), in der die Dalis-Komponente integriert ist. Der Grund, warum in dieser Architektur zwei Plattformen kombiniert wurden, bestand darin, dass Dalis in Prolog implementiert wurde und alleine Quintus-Prolog unter SunOs/Unix zu diesem Zeitpunkt die notwendige TCP/IP-Kommunikation zur Verfügung stellte.
Die zentralen Komponenten des FreeStyler sind in Java implementiert und damit weitgehend plattformunabhängig. Die externen Komponenten, die eingebundene Suchmaschine Verity und die Datenbank Access, sind zwar ebenfalls über Java-Schnittstellen angebunden, sind selbst aber plattformabhängig.

TEIL II

IMPLEMENTIERUNG ZWEIER SYSTEMBEISPIELE

6 FALLSTUDIE I: WISSENSELABORATION DURCH PERSPEKTIVEN: DAS SYSTEM DISCBOARD

Im Folgenden wird beschrieben, wie das CardBoard-System (Hoppe, Gaßner et al., 2000; Gaßner, Tewissen et al., 1998) zu einem diskussionsunterstützenden System, dem DiscBoard, erweitert wird. Die Implementierungs- und Entwicklungsarbeiten zu diesem Fallbeispiel decken die folgenden Bereiche ab:

- Erweiterungen des CardBoards, um die Diskussionsunterstützung (DiscBoard) zu realisieren.
- Modellierung der Perspektiven.
- Entwicklung eines Diskussionsmodells, das den Informationsfluss zwischen den Perspektiven definiert.
- Die technische Realisierung der Abbildung für Karten.
- Die technische Realisierung der Abbildung für Strukturen.

6.1 Perspektiven

Bevor auf die technische Realisierung eingegangen wird, wird im Folgenden das Perspektivenkonzept eingeführt.
Perspektiven strukturieren Verstehensprozesse:

> *A perspective structures a person's cognitive process by:*
>
> - *the selection of those properties of the phenomenon that are being considered (and by implication of those that are ignored). In this way the perspective influences the choice of operative cognitions.*
> - *concepts and other cognitions that are being used in the interpretation of the selected properties.*
> - *the figurative and sometimes physical standpoints influencing the choice of position with respect to the situation or phenomenon in question.*
>
> *(Nygaard & Sørgaard, 1987, S.382)*

Der Einfluss, den eine Perspektive ausüben kann, entsteht danach durch die Auswahl von Eigenschaften oder Kriterien, unter denen ein Phänomen betrachtet wird, durch die

Begrifflichkeiten, die darauf angewendet werden sowie durch individuelle Überzeugungen und außerdem durch den situativen Kontext.
Wird ein Sachverhalt bewusst unter einer Perspektive betrachtet, so geschieht das, um ihn unter bestimmten Voraussetzungen oder Zielstellungen darzustellen, in einem Kontext zu interpretieren oder zu analysieren. Wird von einem Problem angenommen, dass es problemimmanent subjektiv und interessegeleitet gelöst wird und dass Lösungen selektiv, unvollständig, ausschnitthaft, fehlerhaft, interpretierend und widersprüchlich sein können, so kann die bewusste Verwendung von Perspektiven vor allem leisten, dass die jeweiligen Kriterien der Problemlösungen offen gelegt werden.
Nygaard und Sørgaard (1987) unterscheiden drei Arten von Perspektiven. Eine Perspektivenkategorie besteht in der bewussten oder unbewussten selektiven Beschreibung oder Wahrnehmung. Eine zweite Kategorie fasst solche Perspektiven zusammen, die die Umwelt oder Inhalte interpretieren. Darunter fallen Ansichten und individuelles Verstehen, denn in beiden Fällen wird subjektives und unvollständiges Wissen als Grundlage herangezogen, also auf Basis eines Vorverständnisses neue Gegebenheiten interpretiert. Als Drittes existiert die Perspektive als Standpunkt, dies auch im Sinne eines physikalischen Standpunktes.
Stahl (2001) prägt den Begriff der „computational perspective", unter der eine Auswahl von Daten aus einer Menge verstanden wird, bei der es sich damit vor allem um eine kriteriengeleitete, filternde Sicht auf Daten handelt. Unter diese Definition fallen beispielsweise Datenbankviews (Dittrich, 1997). Eine Datenbanksicht ist eine abstrakte Relation, die auf den Basisrelationen oder auf anderen Sichten aufbaut. Mit diesen dynamischen „Fenstern" können komplizierte oder sich wiederholende Anfragen vereinfacht werden. Die Sichten wirken nicht interpretierend, sie stellen nur Ausschnitte der in Relationen verbundenen Daten kompakt dar. Auch Stahl (2001) stellt mit dem System WebGuide einen Perspektiven-Ansatz vor. Dort werden Ansichten von Personen oder Gruppen als Perspektiven hierarchisch verwaltet. Perspektiven sind in diesem System Meinungen zu einem Thema. Diese Meinungen können anderen Personen oder Gruppen zugeordnet werden. Mit so einer Zuordnung werden Mengen von Aussagen automatisch in die neue Perspektive übernommen.

6.1.1 „Epistemic forms" als Perspektiven

In dieser Arbeit wird ein Perspektiven-Ansatz gewählt, der die konstruktiven und interpretierenden Aspekte mit einem filternden Herangehen verbindet. Die Grundlage dafür bilden epistemische Formen (Collins & Ferguson, 1993), die im Folgenden erläutert

werden. Nach Collins und Ferguson (1993) sind epistemische Formen Zielstrukturen, die eine wissenschaftliche (Nach-)Forschung bzw. Untersuchung begleiten.

> *We refer to the target structures that guide scientific inquiry as epistemic forms and the set of rules and strategies that guide inquiry as epistemic games.*
> *(Collins & Ferguson, 1993, S.25)*
> *The forms and games we describe are epistemic in that they involve the construction of new knowledge. They are played to make sense of phenomena in the world.*
> *(Collins & Ferguson, 1993, S.26)*

Die epistemischen Spiele werden insofern von Theorien und Modellen abgegrenzt, als dass sie allgemeiner sind. Theorien und Modelle greifen hingegen oft auf epistemische Formen zurück und setzten sie in einer ganz konkreten Weise ein.

> *They* (Anmerkung: epistemic games) *are not simply inquiry strategies or methods; rather they involve a complex of rules, strategies, and moves associated with particular representations (i.e. epistemic forms).*
> *(Collins & Ferguson, 1993, S.26)*

Als ganz grundlegende epistemische Spiele wird das Erstellen von Listen angeführt, Vergleiche, die Ermittlung von Prozessschritten und die Analyse von Trends. Collins und Ferguson (1993) führen u.a. das sehr einfache Spiel des Anlegens von Listen aus: Die Erstellung einer Liste ist immer an eine Frage gebunden, die auf eine Menge von Antworten zielt, wie „Welche Vorteile hat die Französische Revolution für Europa gebracht?“. Für dieses Spiel bestehen typische Constraints auf den Ergebnismengen: Wert wird auf die Abdeckung gelegt, also darauf, dass die Antwort möglichst viele Aspekte enthält, auf die Unterschiedlichkeit, dass sich die Antworten nicht überdecken, die Vielzahl, denn es soll nicht nur eine Antwort in der Liste enthalten sein, auf die Kürze, denn die Antworten sollen möglichst prägnant sein und auf die Gleichartigkeit der Antworten z.B. in Bezug auf den Allgemeinheitsgrad.

Collins und Ferguson (1993) unterscheiden drei Typen von epistemischen Spielen:

- *Strukturelle Analysen.* Darin wird die „Natur“ eines Phänomens hinterfragt. Der zu untersuchende Gegenstand wird in seine Teile oder auch konstituierenden Elemente zerlegt, die beschrieben und ins Verhältnis gesetzt werden. Collins und Ferguson (1993) nennen dafür u.a. folgende Beispiele: Die räumlich Dekomposition, bei der die topologische Relation von Dingen zueinander geklärt wird, Zustandsmodelle, Vergleichen und Gegenüberstellen beispielsweise mit dem Spezialfall der Kosten/Nutzen-Analyse, die Suche nach „Primitiven“ wie den chemischen Elementen oder grundlegenden Beschreibungssprachen in der KI oder auch die einfache Hierarchisierung. Ebenfalls in diese Kategorie werden Axiomensysteme eingeordnet, wie die Peano-Axiome und die Euklid’schen Axiome in der Geometrie.

- *Funktionale Analyse.* Bei der funktionalen Analyse besteht das Ziel darin, die kausale oder funktionale Struktur von Elementen eines Systems zu beschreiben. Ein Beispiel dafür sind „Ursache/Folge"-Analysen, worin kritische Ereignisse herausgegriffen werden, um notwendige Bedingungen oder mögliche Folgen zu spezifizieren (Collins & Ferguson, 1993). Auch UND/ODER-Graphen können für die funktionale Analyse verwendet werden.
- *Prozessanalysen.* Bei der Prozessanalyse soll das dynamische Verhalten von Phänomenen analysiert werden. Beispiele sind „system dynamics"-Modelle und Beschreibungsmodelle über Constraints (Collins & Ferguson, 1993).

Diese drei Kategorien unterscheiden sich in ihrer Zielstellung, jedoch nicht unbedingt in den Mitteln, die für die Analysen eingesetzt werden. Grundlegende Darstellungen mit speziellen Tabellen oder Graphen finden sich in jeder davon.

Collins und Ferguson (1993) vertreten in ihrer Arbeit schließlich u.a. die folgenden Thesen:

> *Epistemic forms are generative frameworks with slots and constraints on filling those slots. In this respect they are like a grammar that can be expanded to fit the degree of complexity of the phenomena being analyzed. The slots can be cells in a matrix, variables in a model ,or types of curves and their parameters.*
>
> *Epistemic forms and games serve to guide inquiry by directing the inquirer as to which slots to fill and which constraints to meet in filling those slots.*
>
> *Scientific and historical theories often reflect the forms that generated them, but they usually combines a number of different forms in complex interrelationships.*
>
> *(Collins & Ferguson, 1993, S.40)*

„Epistemic games" stellen in den meisten Fällen genauso ein analytisches wie ein konstruktives Vorgehen dar. Der Unterschied zwischen dem epistemischen Spiel und der Form ist, dass beim Spiel die Regeln, Züge oder Strategien hinzukommen. Collins und Ferguson (1993) vergleichen den Unterschied mit einem Spielfeld wie bei Tic-Tac-Toe und dem Spiel selbst, durch das bestimmt wird, wie mit den Feldern umgegangen werden soll.

In dieser Arbeit stehen die „epistemic forms" Pate für den Perspektiven-Ansatz. Wie mit den Darstellungsmitteln umgegangen wird, wird dabei weitgehend den Nutzerinnen und Nutzern überlassen. Gerade dadurch soll eine sehr allgemeine Verwendungsmöglichkeit aufrecht erhalten werden.

Allerdings beschränken sich die visuellen Sprachen nicht ausschließlich auf die Formen, in denen etwas dargestellt wird. Durch die Symbole, die die Sprachen anbieten, werden bestimmte Verwendungen nahe gelegt, also wird im Sinne von Collins und Ferguson (1993) bereits ein „epistemic game" angedeutet. Beispielsweise wird eine Sprache angeboten, die später als Brainstormingsprache benannt wird, die sich an der gIBIS-

Darstellung (Conklin & Begemann, 1987) orientiert. Darin können zwar im Prinzip UND/ODER-Graphen erstellt werden, durch die Symbolik wird aber nahe gelegt, dass damit Optionen bzw. Lösungen zu Fragen oder Problemen aufgeschlüsselt werden. Eine Funktionale Analyse, zu der UND/ODER-Graphen bei Collins und Ferguson (1993) zugeordnet werden, wäre in der vorgegebenen Symbolik nicht plausibel.

6.1.2 Modellierung mit „epistemic games“

Perspektiven können für solche Probleme klärend eingesetzt werden, wo Lösungen zu komplex sind, um konsistent und vollständig erfasst zu werden. Insbesondere sind sie sinnvoll, wenn das Ergebnis offen ist, wie in Design-, Modellierungs- und Konstruktionsprozessen.

Seit einigen Jahren haben Perspektiven-Ansätze auch in der Softwareentwicklung Einzug gehalten. Insbesondere im Ansatz partizipativen Software-Engineerings wird von interessegeleiteten Zielen bei der Entwicklung ausgegangen, die schon von Beginn eines Entwicklungsprozesses mit berücksichtigt werden sollen. In diesem Prozess liefert beispielsweise die Anforderungsanalyse einen Erkenntnisgewinn über die Prozesse, die durch die geplante Software unterstützt werden sollen. Als Resultat aus den Analysen wird durch die neue Software oft ein veränderter Arbeitsablauf etabliert. Ungeeignete Anforderungsanalysen führen deshalb immer wieder zum Scheitern von Softwareprojekten. Perspektiven sollen die unterschiedlichen Interessen transparent machen, um so konstruktiv mit ihnen umgehen zu können.

- *Reference Model of Open Distributed Processing (ODP).* Das “Reference Model of Open Distributed Processing (ODP)” ([ODP]) ist ein Standard für die Entwicklung offener, verteilter Systeme, mit dem gerade der geschilderten Problematik Rechnung getragen werden soll. ODP bietet als zentrales Element einen Modellierungsansatz, der fünf verschiedene Perspektiven („Viewpoints”) anbietet: der Enterprise, Information, Computational, Engineering und Technology „Viewpoint“. Jede dieser Sichten bietet eine eigene Menge von Begriffen und graphischer Beschreibungsmöglichkeiten an. Es ergeben sich fünf unterschiedliche Modelle eines Systems, jedes ist ein Teil der Gesamtspezifikation. Die Systembeschreibung soll damit vereinfacht werden, da davon ausgegangen wird, dass eine vollständige Spezifikation nicht möglich ist. Diese wird letztlich erst durch das System selbst realisiert und liegt entsprechend nur implizit vor. Der Zusammenhang zwischen den Perspektiven wird dadurch hergestellt, das die Objekte der einzelnen „Viewpoints“ miteinander korrespondieren.

- *Unified Modelling Language (UML).* Die Unified Modelling Language (UML) (Jacobson, Booch & Rumbaugh, 1999; Quatrani, 2000) bietet nach dem hier vertretenen Verständnis ebenfalls Perspektiven für die Modellierung von Systemen an. Die Perspektiven entsprechen unterschiedlichen Diagrammtypen, die verschiedene graphische Sprachen anbieten, die jeweils für bestimmte Problemdarstellungen geeignet sind, für die sie Mengen von Beschreibungskonzepten anbieten:
  - „Use Case"-Diagramme greifen typische Nutzungsszenarien, Systemverwendungen und Zielstellungen auf. Mit ihnen können vor allem Anforderungen dargestellt werden.
  - Mit „Activity"-Diagrammen können Workflows und parallele Prozesse dargestellt werden. Sie beziehen sich nicht unbedingt auf Objekte, weshalb Aktionen, die auf Objekten durchgeführt werden, nur schlecht damit ausgedrückt werden können. Gut geeignet sind sie wiederum für die Analyse von „Use Cases".
  - UML bietet auch informationsredundante Diagrammtypen an – z.B. die Gruppe der Interaktionsdiagramme - die aufgrund ihrer unterschiedlichen Art der Visualisierung bewusst zu einem anderen Zweck eingesetzt werden können. Typischerweise schlüsselt ein Interaktionsdiagramm einen konkreten „Use Case" auf. Es verbindet ausgesuchte Objekte und Nachrichten, die zwischen diesen Objekten fließen. Beispiele für Interaktionsdiagramme sind „Sequence"- und „Collaboration"-Diagramme. Im „Sequence"-Diagramm ist es leichter, die Reihenfolge von Ereignissen zu erkennen. Sie sind geeignet wenn wenige Objekte mit einem umfangreichen Nachrichtenfluss dargestellt werden. „Collaboration"-Diagramme zeigen eher die statische Verbindung zwischen den Objekten. Sie sind deshalb besser geeignet, wenn viele Objekte relativ wenig interagieren.
  - „State"-Diagramme sollten nur Klassen verwendet werden, bei denen das Verhalten stark zustandsabhängig ist. Oft sind sie für UI-Kontrollobjekte gut einsetzbar.
  - Am bekanntesten sind vermutlich die Klassendiagramme, die schon sehr eng mit der Implementierung verwoben sind. Die Klassendiagramme können direkt zur Spezifikation von Klassen mit Elementen und Methoden sowie für die Darstellung der Vererbungen verwendet werden.

Die Diagramme greifen ineinander, indem gleiche Objekte wieder aufgenommen und in einen anderen Kontext gestellt werden können. Die Idee besteht nicht in der vollständigen Spezifikation eines Systems, sondern in der exemplarischen Darstellung ausgewählter Fälle (Fowler & Scott, 2000). Das gilt auch für die Klassendiagramme. Relevant sind typische Fälle oder besonders schwierige. Dies wird auch als "Use Case driven development" bezeichnet (Jacobson, Booch & Rumbaugh, 1999; Fowler & Scott, 2000; Quatrani, 2000).

UML wird als Modellierungssprache und nicht als Methode gesehen (Fowler & Scott, 2000), d.h. es werden explizit keine konkreten Prozesse definiert. Einen Ansatz für die Prozessgestaltung bietet der „Rational Unified Process (RUP)" (Quatrani, 2000), der zwar die Darstellungsmethoden von UML verwendet aber umgekehrt nicht durch UML präjudiziert wird.

Anzumerken ist, dass diese Diagrammtypen von den UML-Entwicklern nicht als Perspektiven bezeichnet werden. Dort werden die Designebenen als Perspektiven bezeichnet, die die jeweilige Zielstellung einer Modellierung beschreiben: die konzeptuelle Perspektive, die Spezifikationsperspektive oder die Implementierungsperspektive. Im Gegensatz zu Diagrammen als Perspektiven, die die Beschreibungskonzepte in den Vordergrund stellen, stellen die Designebenen die Modellierungsziele vor.

Auch in Diskussionen finden Prozesse der Wissenskonstruktion statt. Dort muss man ganz besonders von einer selektiven Wahrnehmung durch die „Gruppenkultur" ausgehen. Gerade eine unbefriedigende Wissensgrundlage kann Beweggrund für eine Diskussion sein. Generell können Inhalte nur in Ausschnitten während einer Diskussion aufgearbeitet werden. Trotzdem besteht natürlich der Wunsch, möglichst viele relevante Aspekte aufzugreifen. Deshalb kann sich der Einsatz von Perspektiven positiv auswirken, nämlich um Inhalte besser zu verstehen und ihre Vollständigkeit zu hinterfragen.

6.1.3 Perspektivenwechsel

Die Idee der Perspektive legt es nahe, Perspektivenwechsel zielgerichtet einzusetzen. Jonassen und Carr (2000) argumentieren, dass unterschiedliche Perspektiven das Verständnis eines Sachverhaltes verbessern können. Danach wird ein Problem oder ein Gegenstand besser in einen Kontext gestellt, wenn er explizit auf unterschiedliche Weise repräsentiert wird. So wird eher der Problemzusammenhang begriffen und nicht nur einzelne Details. Jonassen und Carr (2000) nennen diesen bewussten Einsatz unterschiedlicher Repräsentationsmittel „active learning strategies". Sie lassen jedoch die

Frage, in welchem Verhältnis diese Darstellungsformen zueinander stehen und wie diese Perspektivenwechsel angeleitet werden können, weitgehend offen. Auch Suthers (1999A, 1999B, 1999C) hebt die notwendige Passung einer Aufgabe zu einer bestimmten Darstellung und Umgangsmethoden hervor, die er als „representational bias“ bzw. „representational guidance“ beschreibt. Er sieht die Notwendigkeit, erst einmal relevante Eigenschaften von Darstellungen zu konkretisieren, um eine solche Zuordnung zu Aufgaben leisten zu können. Collins und Ferguson (1993) erläutern zwar, dass es Übergangsbedingungen zwischen den „epistemic games“ gibt, gehen jedoch nicht darauf ein, wie diese aussehen können.

Allerdings versuchen manche Systemansätze, die epistemische Formen einsetzen, gerade einen Arbeitsablauf dadurch zu unterstützen, dass epistemische Formen oder Spiele als Bestandteile eines Prozesses implementiert werden. Dies soll hier als *epistemischer Prozess* eingeführt werden.

Ein bereits in Kapitel 3.2 beschriebenes Beispiel ist das Sepia-System (Streitz, Haake, et al., 1992; Streitz, Haake, et al., 1998), in dem vier visuelle Sprachen („epistemic games“) Autorinnen und Autoren kognitiv bei der Erstellung von Hypertexten unterstützen sollen. Aus der Untersuchung von kooperativen Schreibprozessen wurden dafür typische Arbeitsschritte abgeleitet, die jeweils durch die visuellen Sprachen angeleitet werden. Die Prozessunterstützung resultiert letztlich aus der Abstimmung der Darstellungsweisen, die mit dem gewohnten Schreibprozess harmonieren sollen.

Eine Grundlage für die Unterstützung epistemischer Prozesse wird in Scardamalia, Bereiter und Steinbach (1984) gelegt. Deren Ausgangspunkt besteht in der Anleitung von Reflexionsprozessen beim Schreiben. Sie widersprechen der Annahme, dass der „innere Dialog“ mit einem gedachten Leser oder einer Leserin ein geeignetes Modell zur Beschreibung der Reflexion darstellt, die während des Schreibens stattfinden sollte. Stattdessen schlagen sie ein Konzept vor, in dem der Wechsel zwischen zwei Problemräumen, dem „content space“ und dem „rhetorical space“, herangezogen wird, um Phänomene beim Schreiben von Texten zu interpretieren. Meinungen, Fakten, Erklärungen, etc., die im „content space“ vernetzt sind, werden in den „rhetorical space“ übernommen, um sie dort in einen linearen Text zu überführen. Fragen, die sich dort aus der Strukturierung ergeben, werden als Unterziele zurück in den „content space“ übernommen. Scardamalia, Bereiter und Steinbach (1984) behaupten, dass in dieser Teilung der beiden Problemräume die Essenz der Reflexion beim Schreiben liegt.

Carlson (1995) nimmt mit dem Broca-Ansatz, worin allgemeine Schreibprozesse durch die Trennung von inhaltlichen und rhetorischen Aspekten unterstützt werden sollen, auf

den Ansatz von Scardamalia, Bereiter und Steinbach (1984) Bezug. Dafür werden sequentiell verschiedene „facilitator“ (hier interpretiert als „epistemic games“) durchlaufen: Der „Cluster-Browser“ soll das Explorieren von Daten ermöglichen. Jeweils zusammenhängende Graphen bilden die thematischen Cluster. In der „Konzept-Synthese“ sollen die vorher gefundenen Cluster weiter elaboriert werden. Dafür werden Notizkarten angelegt, die Informationsquellen miteinander verbinden und gleichzeitig in eine bestimmte Reihenfolge gebracht werden können. Mit dem „Information Threader“ wird dann versucht, zentrale Konzepte mit semantischen Netzen zueinander ins Verhältnis zu setzten. Die bis hier durchlaufen Phasen behandeln die Daten- bzw. Inhaltsebene des zu erstellenden Textes. Für die Rhetorik werden drei weitere Arbeitsschritte ausgeführt: In einem hierarchischen Planer werden die Textbestandteile von einer Graphstruktur in eine Baumstruktur umgewandelt, um eine lineare Struktur, wie sie für normale Texte verlangt wird, anzustreben. In der nächsten Phase wird die rhetorische Feinstruktur entworfen. In der letzten Phase wird noch eine Revisionsmöglichkeit angeboten.

In den Grundlagen von Broca wird ausführlich für einen Ansatz argumentiert, der verschiedene Arbeitsphasen durch verschiedene Arbeitsmethoden unterstützt. Leider bleibt unklar, inwieweit dieser Ansatz auch als Werkzeug umgesetzt wurde und ob dabei auch schon Erfahrungen dokumentiert wurden. Der technische Ansatz soll auf Objekten basieren, die durch die unterschiedlichen Phasen verändert werden.

Werden in einer Computeranwendung Perspektiven definiert, so beinhaltet das zwingend eine algorithmische Umsetzung davon, wie entweder Daten ausgewählt werden oder in welcher gegenseitigen Abhängigkeit die Perspektiven stehen sowie eine Unterstützung bei der Darstellung der Perspektiven. Die folgende Aufzählung erläutert mögliche Ausprägungen von Abhängigkeiten, die zwischen Perspektiven bestehen können:

- *Filter.* Durch einen Filter kann eine Teilmenge der Inhalte aus einer oder mehreren Perspektiven oder der Datenbasis in einer anderen Perspektive dargestellt werden. Filter sind sinnvoll, wenn auf bestimmte Aspekte fokussiert werden soll. Werden Teile von Inhalten aus mehreren Perspektiven gefiltert, so kann das als Zusammenfassung charakterisiert werden. Damit ist eine Schnittmenge gemäß korrespondierender Eigenschaften gemeint, keine Vereinigungsmenge.
- *Spezialisierung und Abstraktion.* Spezialisierungen können strukturell oder thematisch motiviert sein. Charakteristisch ist, dass ein Objekt, Element oder Teilstruktur aus einer Perspektive herausgenommen wird, um dies in einer anderen Perspektive weiter auszuarbeiten.

Werden Perspektiven durch Vererbungen gewonnen handelt es sich ebenfalls um Spezialisierungen. Die Vererbung basiert allerdings auf internen Hierarchien, die nicht unbedingt aus den externen Darstellungen erkennbar sind.

Eine Abstraktion kann als Gruppierung von Elementen zu einem neuen übergeordneten Objekt verstanden werden. Thematisch motivierte Abstraktionen können z.B. Clusterverfahren sein. Die Bedeutung des Abstraktums bliebe in diesem Fall für das System an die Extension des Clusters gebunden.

- *Sortierungen.* Sortierungen strukturieren ein bestehendes Diagramm um. Sie sind deshalb nicht unbedingt als eigenständige Perspektiven zu verstehen. Trotzdem beeinflussen die Darstellungsformen maßgeblich den Umgang mit ihnen. Sortierungen können sinnvoll mit anderen Perspektivenänderungen kombiniert werden wie z.B. Filter oder Abbildungen. Alle automatisch generierten Perspektiven definieren zwingend auch das Layout der Darstellung und damit die Sortierung.
- *Abbildungen.* Unter Abbildungen kann einerseits die homomorphe oder isomorphe Abbildung von Graphen bzw. Teilgraphen verstanden werden. Abbildungen können andererseits aber auch auf Interpretationsregeln beruhen, die nicht mehr die Struktur erhalten, auch wenn die Basis der Interpretation Strukturmuster sind. Eine solche Interpretation von Strukturen kann auch zur Generierung von Inhalten führen, die in neuen Objekten einer anderen Perspektive visualisiert werden.

Das DiscBoard setzt einen Perspektivenansatz um, der Filter, Abbildungen, Sortierungen und Abstraktionen integriert. Die einzelnen visuellen Sprachen stellen epistemische Formen zur Verfügung, wenn auch durch die Vorgabe von Symbolen bestimmte Verwendungen im Sinn von epistemischen Spielen nahe gelegt werden. Die Kombination verschiedener visueller Sprachen leitet einen epistemischen Prozess an, der aber nicht als Schreibprozess zu verstehen ist, sondern als Prozess der Wissenselaboration, der sich also im „content space“ (Scardamalia, Bereiter & Steinbach, 1984) bewegt.

Durch die Perspektiven soll der Diskussionsgegenstand unter verschiedenen Zielstellungen betrachtet werden können. Es wird deshalb insbesondere keine Konsistenz zwischen den Perspektiven verlangt. Es wird davon ausgegangen, dass sich die jeweiligen Inhalte der Perspektiven zwar aufeinander begründen und auseinander hervorgehen, dass sie sich dann aber auch additiv ergänzen. Das reflektiert die Idee, dass ein Diskussionsgegenstand meistens viel zu umfangreich ist, um ihn vollständig zu erarbeiten. Vielmehr soll er sporadisch für einige relevante Aspekte aufgearbeitet werden.

Zwischen den Perspektiven im DiscBoard wurden Abhängigkeiten implementiert. Durch das Angebot von Perspektiven, die mit einer spezifischen Auswahl bestehender Daten

automatisch gefüllt werden, sollen Diskussionsprozesse angeregt und unterstützt werden. Mit dem DiscBoard wird damit ein pragmatischer Ansatz zur Strukturierung von Gesprächsinhalten implementiert.

6.2 Erweiterungen des CardBoards

6.2.1 „Joint“ Workspaces

Aufgrund des bereits vorgestellten Diskussionsszenarios, dass als „face-to-face“-Szenario eingeführt wurde, gehe ich davon aus, dass in der Gruppe entweder über eine Projektionswand von allen Beteiligten in einer gemeinsamen Applikation gearbeitet wird oder dass „joint“ Workspaces verwendet werden. Diese dienen dann einerseits der komfortablen Eingabe vom Platz, um das Notierte unmittelbar an der Projektionsfläche sichtbar zu machen. Andererseits liegt der Vorteil eines Settings mit „joint“ Workspaces, in dem allen Beteiligten eine eigene Applikation zur Verfügung steht, in der individuellen Kombinationsmöglichkeit privater und öffentlicher Eingaben.

Ob auch „shared“ Workspaces erwünscht sind, hängt nach meiner Überzeugung von der Art des Gespräches und der Aufgabenstellung ab. Stellt man sich eine politische Debatte vor, in der Absprachen zwischen verschiedenen Beteiligten während der Diskussion getroffen werden könnten, wären Zwiegespräche u.U. sinnvoll. In weniger taktischen, sondern mehr konsensorientierten Sitzungen, beispielsweise in Projektgruppensitzungen, sind geheime Bestandteile weniger sinnvoll oder auch unerwünscht. Für die Aufwertung der „joint“ Workspaces gegenüber den „shared“ Workspaces bei der Diskussion spricht auch, dass die Koordination von möglicherweise mehreren schriftlichen Diskussionssträngen sowie der verbalen Kommunikation einschließlich der jeweiligen Interaktion zwischen Personen und mit dem System ausgesprochen kompliziert ist. Dass diese Situation, in der die Interaktion mit dem System mit dem Gespräch koordiniert werden muss, auch ohne parallele schriftliche Gesprächsstränge schwierig ist, konnte schon in der Vorstudie beobachtet werden (siehe Kapitel 4).

Konzeptuell ist für jeden Workspace-Typ, anders ausgedrückt für jede visuelle Sprache, nur ein „joint“ Workspace vorgesehen, um Verwirrungen zu vermeiden. An diesen Workspace kann sich jede Applikation anschließen und diesen auch wieder verlassen. „Joint“ Workspaces erlauben den Zugriff einer beliebigen Anzahl von Diskussionsbeteiligten auf den gleichen Datenraum. Die „joint“ Workspaces präsentieren sich damit den Beteiligten als Arbeitsbereiche, die immer zugreifbar sind und die den aktuellen Diskussionsstand zeigen. In diesem Konzept ist es möglich, dass nur eine Person in einem

„joint“ Workspace arbeitet. Das ist dann die erste Person in diesem Arbeitsbereich, die damit ihre Bereitschaft zur Kommunikation kundtut und dadurch für andere erreichbar wird. Das „Joinen“ stellt sich für die Beteiligten nicht direkt als Aufnahme eines Kommunikationsprozesses dar, sondern eher als Zugriff auf eine virtuelle Tafel auf der auch andere arbeiten könnten.

Dieses Konzept widerspricht in gewisser Weise, der im MatchMaker implementierten Idee der Synchronisation. MatchMaker stellt als Basis für die Synchronisation gerade keine zentralen „Datenobjekte“ zur Verfügung, die einer solchen virtuellen Tafel entsprechen könnten. Synchronisiert werden die Userinterface-Events. Der Zustand eines „joint“ Workspaces wird also nicht direkt im MatchMaker verwaltet, sondern liegt nur in der Applikation selbst als Information vor.

Für das skizzierte „join“-Konzept muss deshalb das Problem gelöst werden, dass eine beliebige Applikation, die ein „join“ ausführt, eine beliebige andere finden muss, von der die relevanten Daten übernommen werden können. Dieser Instanzierungsprozess darf selbst noch nicht im synchronisierten Modus stattfinden.

Jede Applikation verwaltet dafür eine Liste ihrer eigenen „joint“ Workspaces. Löst eine Applikation ein „join“ für einen bestimmten Workspace-Typ aus, so wird eine beliebige andere Applikation gesucht, die bereits einen „joint“ Workspace dieses Typs generiert hat. Mit diesem wird dann die Kopplung durchgeführt. Ein neuer „joint“ Workspace wird erst generiert, wenn in allen anderen Applikationen keine geeignete Partner-Applikation zum Koppeln gefunden wird. Auf diese Weise wird sichergestellt, dass tatsächlich nur ein „joint“ Workspace zu einem Workspace-Typ existiert. Nachdem eine geeignete Partner-Applikation gefunden wurde, wird der neue Workspace mit den bereits vorliegenden Daten initialisiert. Wird ein „joint“ Workspace geschlossen, so besteht die Verbindung für die anderen Beteiligten weiter. Dieser Prozess des „join“ ist in Abbildung 27 dargestellt.

Links sind zwei Applikationen in Form ihrer aktuellen Benutzeroberfläche symbolisiert. In Applikation 1 (oben links) sind zwei „joint“ Workspaces geöffnet, die intern in einer Liste verwaltet werden, die in der Abbildung ebenfalls dargestellt ist. In der unteren Applikation sind drei „joint“ Workspaces geöffnet. Auf der rechten Seite befindet sich eine dritte Applikation in drei Varianten, die sich aus einer zeitlichen Entwicklung ergeben. Diese Applikation hat ein „join“ für den in der Abbildung blau dargestellten Workspace ausgelöst. Dafür wird in Schritt 1) ein geeigneter Partner gesucht, wobei in diesem Fall beide anderen Applikationen in Frage kämen.

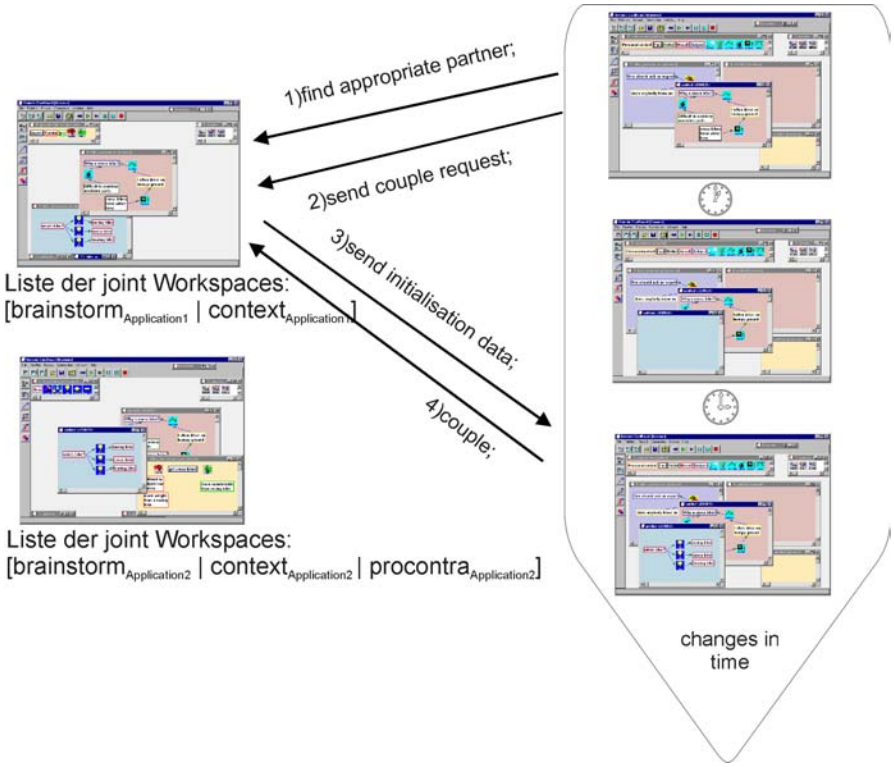


Abbildung 27

Wird von einer Applikation aus ein „join“ durchgeführt, so wird zuerst eine geeignete Partner-Applikation gesucht. Auf die Kopplungsanfrage werden die Initialisierungsdaten zurückgesendet. Erst am Ende wird das eigentliche Koppeln durchgeführt.

Nachdem jedoch in diesem Beispiel die obere ausgewählt wurde, wird an diese ein „couple request“ gesendet (Schritt 2). In Folge davon kann der Workspace selbst bereits aufgebaut werden, hat jedoch noch keine Daten (Bild rechts, mittig). In der nächsten Phase sendet die angefragte Applikation die Initialisierungsdaten (Schritt 3). Erst danach wird die eigentliche Synchronisation vorgenommen (Schritt 4).

Dieses eher umständliche Kopplungskonzept für das „join“ ist als ein Erfahrungshintergrund zu werten, der in die Neukonzeption des MatchMaker TNG (Tewissen, Baloian et al., 2000; Jansen, Pinkwart & Tewissen, 2001; Jansen, 2002) eingeflossen ist. Dort werden gerade Objektmodelle vorgehalten, zusätzlich aber auch Veränderungen in den Modellen an alle betroffenen Applikationen propagiert. Daraus ergibt sich eine Kombination eines zustandsorientierten und eventorientierten Ansatzes.

Trotz der beschriebenen und implementierten Erweiterungen des „join“-Konzeptes, das ohne die Dalis-Anbindung voll funktionsfähig ist, wurden im Verlauf der Entwicklung der Diskussionsunterstützung weitere Adaptionen notwendig, um das „join“ mit den für die Diskussion verwalteten Daten in Dalis kompatibel zu gestalten.

Wegen der in den Folgekapiteln noch eingehend vorgestellten Perspektiven müsste von Dalis eine vollständige persistente Datenhaltung implementiert werden. Jegliches Öffnen eines Workspaces müsste dann auf diese Daten zugreifen. Das „join“ würde dann einen allgemeiner, „öffentlicher“ Zugriff sein, während die „private“ Workspaces an einen konkreten Autor oder Autorin gebunden wären. Das bisher realisierte „join“-Konzept, würde so nicht mehr benötigt werden, da Dalis eine zentrale Verwaltung der Objekt-zustände zur Verfügung stellen würde.
Weder die Autorschaft, noch die vollständige Persistenz der Daten in Dalis sollte aber im Rahmen dieser Arbeit umgesetzt werden, die sich zentral mit den Perspektivenwechseln beschäftigt. Außerdem hätte das weitergehende Konsequenzen nach sich gezogen: Das Interface zwischen den Teilsystemen MatchMaker, CardBoard und Dalis hätte erweitert werden müssen, um diese zusätzliche Information der Autorschaft überhaupt kommunizieren zu können.
Eine vereinfachende Idee bestünde darin, immer alle Personen ausschließlich in „joint“ Workspaces arbeiten zu lassen. Es wären so zwar keine privaten Daten möglich, dafür könnte auf die Autorschaft verzichtet werden.

6.2.2 Paletten

Die Aufgabe der Paletten besteht vor allem darin, den Nutzerinnen und Nutzern die Relationen und Kartentypen einer visuellen Sprache zu präsentieren. Einerseits kann so auf einen Blick das Spektrum der jeweiligen visuellen Sprache erfasst werden und die Kartentypen können leichter gegeneinander abgewogen werden, um einen geeigneten herauszugreifen. Andererseits können so Karten durch einfaches Drag & Drop aus der Palette generiert werden.
Paletten sind selbst Workspaces, jedoch nicht als Arbeitsbereiche vorgesehen. Die CardBoard-Erweiterung besteht darin, dass für die in der Initialisierungsdatei frei anzugebenen Sprachen die Paletten-Definitionen automatisch geladen werden. Abbildung 28 zeigt wie diese Palettendateien referenziert werden.
Außerdem wird durch das System sichergestellt, dass sich immer die Palette im Vordergrund befindet, die zum gerade bearbeiteten Workspace-Typ passt.
Der ebenfalls in Abbildung 28 aufgeführte Eintrag [InitFunctionsPalette] bezieht sich noch auf eine bestimmte Art von Palette, die Funktionspalette. Auf diese wird in Kapitel 6.4 weiter eingegangen. Diese Palette enthält spezielle Karten, die der Steuerung der Interaktion mit Dalis dienen.

```
[Initpaletten]
Init="protocol.ws","protocol"
Init="explore.ws","explore"
Init="brainstorm.ws","brainstorm"
Init="topics.ws","topics"
Init="argument.ws","argument"
Init="procontable.ws","procontable"

[InitFunctionsPalette]
Init="functions.ws","functions"
```

Abbildung 28

Definitionen, welche Paletten geladen werden sollen.

6.3 Die visuellen Sprachen für die Diskussionsunterstützung: Die Perspektiven

Im Folgenden werden die im System umgesetzten visuellen Sprachen vorgestellt, die Perspektiven. Die Grundlage für deren Auswahl sind Vorgängerversionen und einzelne Tests[10], die zu Veränderungen der jeweiligen Objektmengen führten, die jedoch weitergehend evaluiert werden sollten.

Die Sprachen sind so gestaltet, dass aus ihren primitiven Objekten jeweils Graphen oder graphische Darstellungen entwickelt werden können, die auf Basis ihrer „intendierten" symbolischen Bedeutung dafür geeignet sind, bestimmte Strukturen zu entwickeln oder auch ganz frei Notizen anzufertigen. Bei argumentativen Diskursen können beispielsweise Argumentstrukturen aufgebaut werden. Tabellen können für Pro-/Kontradebatten herangezogen werden. Ein UND/ODER-Graph kann für die Diskussion und Abwägung von Alternativen entwickelt werden. Ein Protokollbereich dient der Zusammenfassung und Planung eines Gespräches. Ein weiterer Bereich dient der Exploration von Themen. Damit passen die einzelnen Sprachen zu verschiedenen Phasen der Diskussion.

Durch die strukturierte Darstellung können insbesondere die Argumentationen, die Tabellen und die UND/ODER-Graphen zur Wissenselizitation herangezogen werden, da sie Lücken sichtbar machen. Dies ist eine Eigenschaft der visuellen Darstellung (Boxtel & Veerman, 2001). Viele Wissenselizitationsmethoden basieren auf bestimmten Strukturierungsverfahren. Sortierungen, Clustering oder Bewertungen sollen Zugehörigkeiten

[10] Als Test ist die im Kapitel „Diskussionsunterstützung: Eine Vorstudie zur Hypothesenbildung" erläuterte Studie eingeflossen sowie Vorläufer-Versionen zu dieser Studie, eine mehrtägige eher protokoll-orientierte Mitschrift einer Projekt-Sitzung, die mit dem CardBoard durchgeführt wurde und die Verwendung des CardBoards bei der Darstellung konkreter Inhalte.

und Abhängigkeiten verdeutlichen (Cooke, 1994). Tabellen sind besonders gut geeignet, um auf unausgefüllte Felder zu verweisen. Ein Beispiel dafür ist das „repertory grid"-Verfahren nach Kelly (1970), das z.B. von Gaines und Shaw (1993) schon einmal in eine umfangreiche visuelle Sprache umgesetzt wurde.

Suthers (1999B) versucht Vorteile verschiedener diagrammatischer Darstellungsformen herauszuarbeiten. Er beschreibt, dass Tabellen dazu führen, die Relationen zwischen den Objekten vollständig zu definieren. Dagegen verleiten explizite Relationen in Graphen offenbar zu singulären Verbindungen. Der Vorteil besteht darin, dass die Relationen in diesem Fall als eigenes Objekt greifbar sind. Sie stellen deshalb in der Gesprächssituation einen einfachen, optischen Bezugspunkt in der Visualisierung dar, auf den referenziert werden kann. Sowohl die Graph- als auch die Tabellenform kann in unterschiedlichen Kontexten sinnvoll sein. Suthers (1999A, 1999B) benennt diese Passung zwischen Aufgabe und Darstellungsmittel als „salience"[11]. Das bedeutet, dass die hervorspringenden Eigenschaften einer Darstellung mit der Aufgabenstellung übereinstimmen sollten.

In den folgenden visuellen Sprachen wird für die Pro/Kontradarstellung eine tabellarische Darstellung gewählt, die Schwerpunkte und Lücken verdeutlicht. Dagegen werden Argumentationen als Graphstrukturen repräsentiert. Das bietet sich für die Argumentationen in Anlehnung an Toulmin (1958) an, da hier nur spezielle Relationen angeboten werden und eine vollständige Verbindung der Argumentbestandteile gerade nicht erwünscht ist. In der visuellen Sprache zur Exploration sollen die Relationen vorerst nur punktuell aufgeführt werden. Ein tabellarisches Herangehen würde aus meiner Sicht diese Phase zu stark organisieren und den explorativen Charakter aufheben.

6.3.1 Allgemein zur Gestaltung der visuellen Sprachen

Generell muss der Umfang der visuellen Sprachen, also die Anzahl und auch die Art der Relationen und Beitragstypen, abgewogen werden. Eine zu große Ausdrucksmächtigkeit kann schnell unübersichtlich wirken und zu redundanter Ausdrucksweise führen (Suthers, Toth und Weiner, 1997). Das heißt, dass verschiedene Sprachelemente für gleiche Beitragstypen herangezogen werden. Je nachdem, welche Inhalte mit ihnen dargestellt werden sollen, müssen die Beitrags- und Relationstypen an das Problem angepasst werden. Im vorliegenden Ansatz zur Diskussionsunterstützung kann durch die Vielfalt an Perspektiven der Umfang der einzelnen Sprachen reduziert werden, um so eine Anpassung an eine ganz konkrete Arbeitsphase bzw. Anwendungsmethode zu erreichen.

[11] Übersetzung: Hevorspringen

Die Gestaltung der Relationen stand im Spannungsfeld zwischen der technischen Umsetzbarkeit und Ausdrucksmächtigkeit einerseits und der benutzungsfreundlichen Darstellung andererseits.

In Bezug auf die Ausdrucksmächtigkeit stellen die verwendeten visuellen Sprachen umfangreiche Mittel zur Verfügung, da wie bereits erwähnt, n-stellige Relationen darstellbar sind, die zur Entwicklung von Hypergraphen führen. Es zeigte sich jedoch in der Versuchsphase, dass mehrstellige Relationen nur schwer lesbar waren. Zugunsten der Übersichtlichkeit ist deshalb darauf verzichtet worden. Eine n-stellige Relation würde durch n Links visualisiert, die eine Konnektorkarte mit einer entsprechenden Anzahl von Inhaltskarten verbindet.

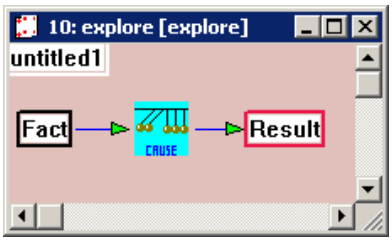


Abbildung 29

Eine Relation bestehen aus zwei Inhaltskarten („Fact“ und „Result“) und einer Karte, der Konnektorkarte, die den Typ der Relation symbolisiert, die über zwei Links mit den Inhaltskarten verbunden ist.

Weiterhin bestand eine Frage darin, wie die Konnektorkarten gestaltet werden sollten. Für die Diskussionsunterstützung wurden die Konnektorkarten letztendlich durchgängig mit festen Symbolen ausgestattet, wie es in Abbildung 29 zu sehen ist. Es bestand zusätzlich die Option, solche Konnektorkarten zu verwenden, die wie die Inhaltskarten, frei beschriftet werden können. Außerdem hätte es den Nutzerinnen und Nutzern überlassen werden können, selbst ausgewählte Symbole zu verwenden. In dem hier gewählten Vorgehen sollten jedoch die Symbole jeweils einen Hinweis auf den Bedeutungszusammenhang in dem sie verwendet werden geben („intendierte Semantik“). Es soll gerade ausgeschlossen werden, dass immer neue Symbole ein „gemeinsame Sprache“ in der Gruppe verhindern.

Auch die Art der Symbole hat sich als relevant herausgestellt. In der Versuchsphase beinhaltete die dort verwendete Sprache Symbole, die optisch gerichtet waren, wie z.B. einen Folgerungspfeil, der von links nach rechts zeigte. Diese Richtung wirkte verwirrend, da sie teilweise entgegengesetzt zu den interaktiv hergestellten Links war. Deshalb ist auf solche gerichteten Symbole für die Relationen verzichtet worden.

In den ersten Entwicklungen des CardBoards wurde die Entscheidung getroffen, die Typen der Inhaltskarten durch Farben zu kennzeichnen. Diese Entscheidung ist

rückblickend zu kritisieren. Es zeigten sich zwei Probleme: Das erste bestand darin, dass anhand der Farben nur relativ wenige trennende Kategorien definiert werden können. Kategorien wären rot, blau, gelb, grau... Die Farben sind jedoch schwierig zu unterscheiden, wenn sie dichter beisammen liegen. Außerdem zeigte sich, dass die Bedeutung der Farben nur sehr schwer zu erlernen ist. Dagegen können Formen sehr vielfältig und auch bedeutungstragend definiert werden. Die Form selbst ist leicht als Kategorie zu verstehen wie eine Wolkenform, ein Warndreieck, etc. Diese Verbesserung ist in die zeitlich nachfolgende Entwicklung des FreeStylers eingeflossen.

6.3.2 Die visuelle Sprache zur Diskussionsstrukturierung (Protokoll)

In der Gruppe der visuellen Sprachen für die Diskussionsunterstützung ist eine Sprache für die Diskussionsstrukturierung zuständig. Diese Sprache wird als Protokollsprache bezeichnet. Sie ist jedoch gerade nicht dazu gedacht, die Diskussion im Sinne eines klassischen Protokolls zu begleiten. Sie soll vor allem die Planung und Strukturierung des Gesprächs unterstützen und ist deshalb im Wesentlichen für die Anfangsphase einer Diskussion gedacht. Darin werden Besprechungspunkte, erste Zusammenhänge und Bemerkungen festgehalten. Es werden Beiträge wie Fragen, Aspekte und Ideen erst einmal gesammelt. Das Notieren der Beiträge soll schnell gehen und die eigentliche Diskussion dazu später geführt werden. Deshalb stellt diese Sprache nur eine kleine Auswahl an Beitragskategorien und Relationstypen zur Verfügung, denn sowohl die Klassifikation der Beiträge als auch deren Strukturierung mit Relationen ist im Allgemeinen anspruchsvoll und zeitaufwendig. Eine einzige allgemeine Relation kann dazu verwendet werden, um die Beiträge in Verbindung zu setzen.

Natürlich kann die Liste der zu behandelnden Besprechungspunkte auch während der Diskussion leicht ergänzt oder Punkte wieder gestrichen werden. Zu diesen Ergänzungen gehören später die Diskussionsergebnisse, die mit in das Protokoll integriert werden.

Die Tabelle 10 stellt in der linken Spalte die visuellen Objekte der Protokollsprache dar. In der rechten Spalte werden die Kategorien erläutert.

In Abbildung 30 ist links ein Schema für die Verwendung der Protokollsprache angedeutet. In ihr können Listen von „Tasks“ erstellt werden, die selbst schon auf Teilprobleme verweisen können. Dies könnten selbst wieder „Tasks“ sein oder auch „Questions“. Ein empfohlenes Strukturierungsmittel ist hier die Einrückstruktur. Außerdem kann die Relation verwendet werden, um Zusammenhänge aufzuzeigen. Auf der rechten Seite der Abbildung wird beispielsweise ein Zusammenhang „Task“-übergreifend markiert. Die

„Contribution“- und „Question“-Karten stellen im Wesentlichen ad hoc-Beiträge zur ersten „Task“ dar.

Task	Die Kategorie „Task“ ist zum Festhalten von Problemen oder Arbeitspaketen und grundlegenden Fragestellungen gedacht.
Question	Konkrete, detailliertere Fragen können mit der „Question“-Kategorie von den Tasks unterschieden werden.
Contribution	Auf Karten der „Contribution“-Kategorie können beliebige andere Beiträge notiert werden.
Idea	Mit Karten der „Idea“-Kategorie können Ideen notiert werden. Diese können natürlich auch außerhalb des gerade diskutierten Kontextes liegen.
RELATED	Mit dieser Relation können Beiträge assoziiert werden.

Tabelle 10

Menge und Bedeutung der Beitrags- und Relations-Typen der Protokollsprache

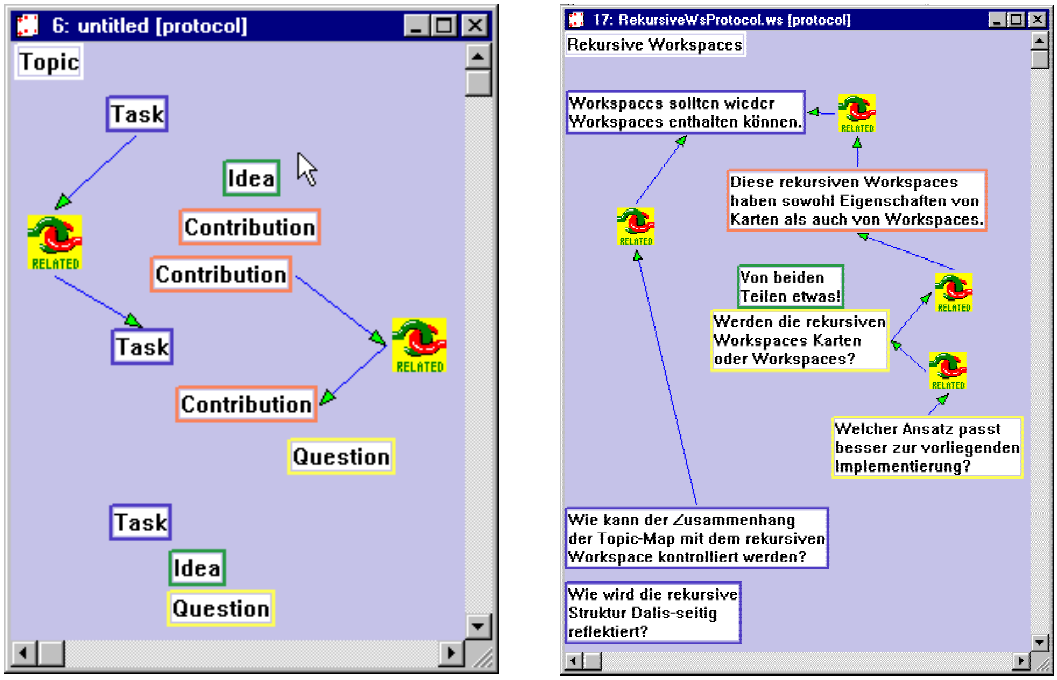


Abbildung 30

Schema (links) und Beispiel (rechts) für die Verwendung der Protokollsprache.

6.3.3 Die visuelle Sprache zum Explorieren

Die Explorationssprache dient der Vor- wie Nachbereitung anderer Arbeitsphasen. Aus der Diskussionsanfangsphase werde Beiträge aufgegriffen und exploriert. Einzelne Aspekte sollen dann später in spezifischeren Darstellungen, wie Tabellen, weiter diskutiert werden. Mit der Explorationssprache sollen Argumente darstellbar sein, sowie ein Themenfeld exploriert werden können. Zu einem Thema werden in einem

Arbeitsbereich Fakten gesammelt, Einschränkungen und Bedingungen aufgeführt sowie Hintergründe, Zusammenhänge und Meinungen dargestellt.

Je nach Bedarf können hier sehr detailliert Themengebiete entwickelt werden. Es ist davon auszugehen, dass die Strukturierung von Inhalten mit dieser visuellen Sprache arbeits- bzw. zeitaufwendig ist. Durch die größere Menge an Relationen können umfangreiche Strukturen aufgebaut werden, die der Reflexion des Gegenstandes dienen.

Verzichtet wurde allerdings – wie vorher in dieser Arbeit schon diskutiert - auf die Darstellung von „moves“ wie sie in linguistischen Analysen meist verwendet werden. Dort interpretieren „moves“ die linguistische Bedeutung zwischen Beiträgen in Gesprächen, meistens in Dialogen. Beispiele dazu wären „Re-initiative“, „Gegen-Initiative“ oder „Reaktion“ (Franke, 1990). Auch die „Answer“-Relation im Sepia-System wäre darunter einzuordnen. Hier sollen die Beiträge im Wesentlichen auf inhaltlicher Ebenen miteinander in Verbindung gebracht werden.

Die Suche nach geeigneten Primitiven gestaltete sich jedoch schwierig. Gruppen von Primitiven werden auch in anderen Ansätzen verwendet, wie zum kooperativen Schreiben (Sepia), zum wissenschaftlichen Argumentieren (Belvedere) oder auch für didaktische Zusammenhänge (Baloian Tataryan, 1997); aus dem linguistischen Umfeld stammen Diskursanalysen (Levin & Moore, 1977; Pilkington & Mallen, 1996; Baker, 1995).

Mit der Explorationssprache sollten möglichst viele Arten von Zusammenhängen ausgedrückt werden. Versucht wurde, die in anderen Systemen verwendeten Primitiven aufzugreifen, zu gruppieren und allgemeinere Bezeichnungen dafür zu finden. Tabelle 11 stellt diese Sammlung von Primitiven dar, die in den genannten Systemen verwendet werden sowie eigene Ergänzungen. In der ersten Zeile werden diese Primitive allgemeiner kategorisiert, um daraus wiederum Beitragkategorien für die Explorations-sprache abzuleiten.

Allgemeine Beschreibung	Ausgangspunkt	Gegebenheiten	Vages/ Abstimmbares	Persönliche Wertigkeiten/ Vorlieben/ Umstände	Ergebnisse
Mögliche Primitive	Problems Tasks Questions	Constraints Facts Option Thesis Premise	Hypothesis Belief Opinion Advantage Disadvantage	Interest Desire Preference Advantage Disadvantage	Thesis Solution Compromise Suggestion Result Hypothesis
Abgeleitete Beitrags-Kategorie	Subject	Fact	Estimation	Personalcontext	Result

Tabelle 11

Beitragskategorien für die Explorationssprache

Neben den Beitragskategorien in Tabelle 11 werden in Tabelle 12 die verwendeten Relationen erläutert.

SUPPORT	Ein Beitrag stützt einen anderen.
SOLVE	Ein Beitrag stellt eine Lösung für ein Problem dar.
CONSTRAIN	Ein Beitrag beschreibt, warum etwas nur unter bestimmten Bedingungen eintritt, zutrifft, etc.
CAUSE	Folgern; Eine Begebenheit führt zu einer weiteren.
EXPLAIN	Erklären-Warum, Begründen, Rechtfertigen
ATTACK	Beiträge die andere in Frage stellen aufgrund von Indizien, Annahmen, die eine Gegenposition darstellen oder einen logischen Widerspruch aufzeigen.

Tabelle 12

Bedeutung der Relations-Typen der Explorationssprache.

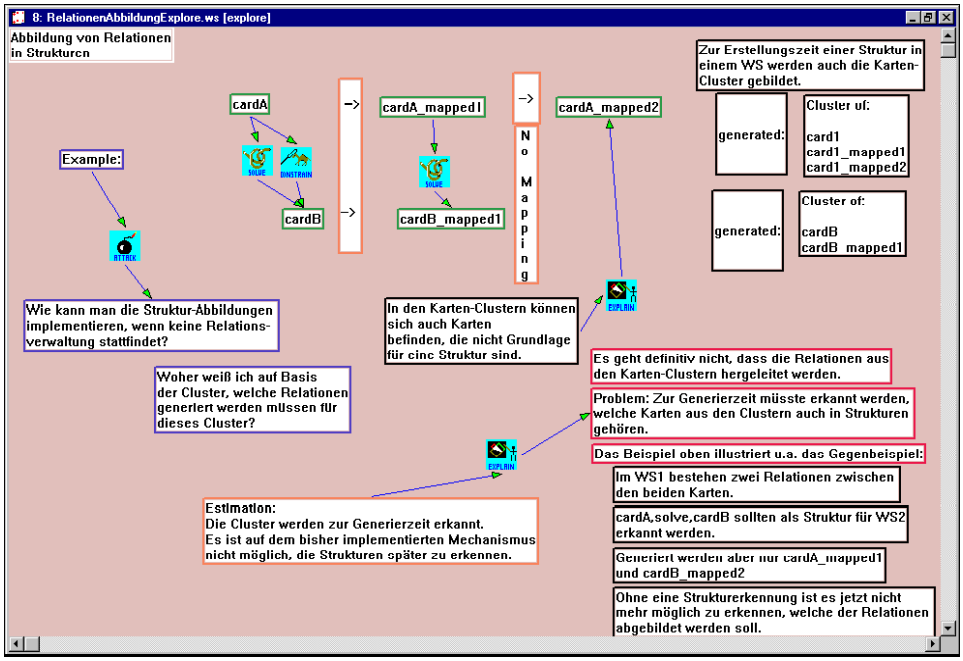


Abbildung 31

Beispiel 1 für die Verwendung der Explorationssprache.

Abbildung 31 und Abbildung 32 zeigen Beispiele für die Arbeitsweise in einem Explorations-Workspace. Inhaltlich befassen sie sich mit Designfragestellungen zum Entwurf der Diskussionsunterstützung. Im ersten Beispiel (Abbildung 31) werden Karten u.a. als Auflistungen (rechts unten) und als Container für selbstdefinierte Symbole bzw. optisch strukturierende Elemente verwendet. Gemeint sind damit z.B. die Pfeile, die ohne

weiteren Text in einer Karte enthalten sind und in diesem Beispiel die Abbildung von Karten zwischen Workspaces andeuten.

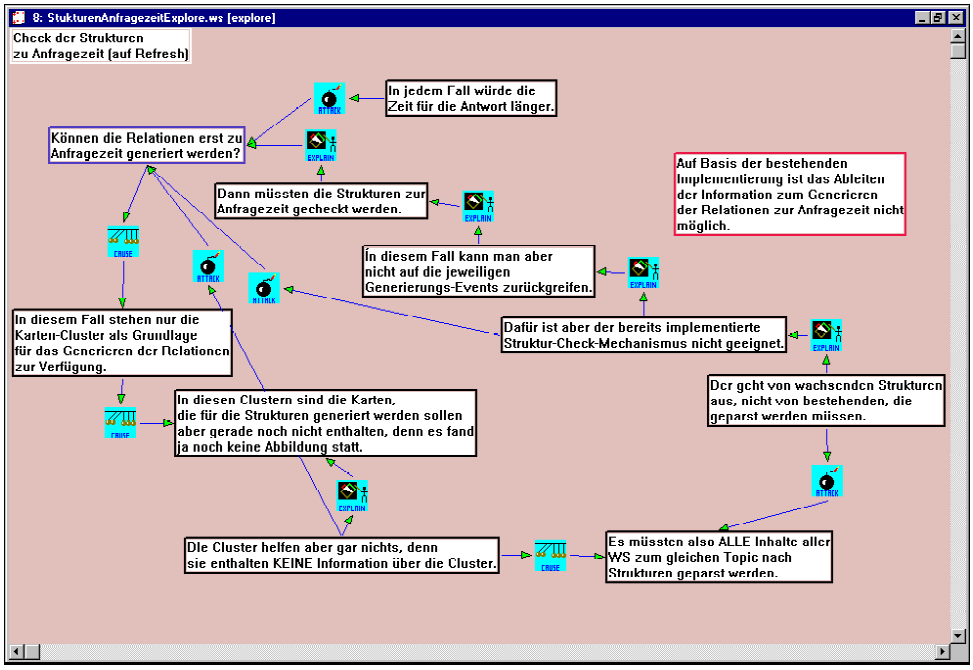


Abbildung 32

Beispiel 2 für die Verwendung der Explorationssprache.

Auch bei solchen optisch oder inhaltlich strukturierenden Symbolen können die Kartenkategorien sinnvoll verwendet werden. In diesem Fall wurden die Pfeile in „Estimation"-Karten eingetragen. So kann eine Unsicherheit in dieser Einschätzung ausgedrückt werden.

Das andere Beispiel (Abbildung 32) zeigt zwei Begründungsketten, die sich in der unteren rechten Ecke mit einem gemeinsamen Bezug schließen, der so auf ein ganz zentrales Problem bei dem dort erörterten Design-Vorschlag verweist.

6.3.4 Die visuelle Sprache für das Brainstorming

Die Brainstormingsprache lehnt sich an den IBIS-Ansatz (Conklin & Begemann, 1987) an. Sie stellt an UND/ODER-Graphen angelehnte Hilfsmittel zur Verfügung, Ideen und Alternativen festzuhalten. Mit ihr können Optionen zu Fragestellungen in mehreren Baumebenen aufgeführt werden. Diese können dann positiv oder negativ bewertet und mit Kriterien versehen werden. Durch Markierungskarten können Entscheidungen und Präferenzen deutlich gemacht werden.

Für das eigentliche Brainstorming werden die vorgegebenen Relationen jedoch erst einmal gar nicht verwendet. Hier werden die jeweiligen Ideen der beteiligten Gruppenmitglieder einfach auf Karten notiert. Dazu wird möglichst immer die gleiche Karten-

kategorie verwendet, um während des Brainstorms nicht über die Kategorie reflektieren zu müssen. Je nachdem worüber das Brainstorming durchgeführt wird, kann dies entweder ganz allgemein die „Idea"-Karte sein oder bereits die „Option"-Karte. Im Grunde könnte sogar über Begründungen ein Brainstorm durchgeführt werden. Dem eigentlichen Brainstorming kann dann die Strukturierung der Vorschläge in dem UND/ODER-Graph folgen.

Die Brainstormingsprache ist deshalb weniger als Hilfe für das Brainstorming selbst zu sehen, als dass es eine Möglichkeit bietet, die Ergebnisse nachfolgend zu strukturieren und zu bewerten. Die Brainstorming-Methode könnte prinzipiell mit allen visuellen Sprachen des DiscBoard-Systems durchgeführt werden.

Task	Die Ausgangsfragestellung.
Idea	Allgemeine Ideen.
Option	Zum Festhalten von alternativen Vorschlägen zu einer übergeordneten Frage.
Reason	Ein Grund kann sowohl eine Bewertung als auch Kriterien erläutern.
	Die „Idea"-Relation stellt eine ODER-Verbindung zu anderen Ideen auf der gleichen Ebene dar.
	Die „Ideagroup"-Relation stellt eine UND-Verbindung zu anderen Ideen auf der gleichen Ebene dar. Die Zugehörigkeit mehrerer Optionen oder Ideen kann momentan jedoch nur durch Nähe der verwendeten Relationskarten visualisiert werden.
	Diese Relation signalisiert eine positive Einstufung einer Idee.
	Diese Relation signalisiert eine negative Einstufung einer Idee.
	Mit dieser Relation können Kriterien dargestellt werden, unter denen Ideen verwirklicht werden können, Voraussetzungen, etc.
	Zwei Ideen können gegeneinander bewertet werden.
	Endgültige Entscheidungen werden markiert.

Tabelle 13

Bedeutung der Beitrags- und Relationstypen der Brainstormingsprache.

In Tabelle 13 werden die Kategorien und Relationen der Brainstormingsprache erläutert. Abbildung 33 zeigt ein Schema der Brainstormingstrukturen. Von der Baumwurzel, der „Task", verzweigen sich die Optionen. Kriterien und Gewichte beziehen sich auf die einzelnen Optionen. Falls eine Auswahl getroffen wurde, kann diese durch die „Select"-Karte hervorgehoben werden. In Abbildung 34 wird die Bewertung der drei Optionen dargestellt. Dabei gewichten die Begründungen teilweise mehrere Optionen. Die mittlere

Option verweist auf eine weitere „Task“, die dafür gelöst werden müsste, was die übergeordnete Option in diesem Fall negativ gewichtet.

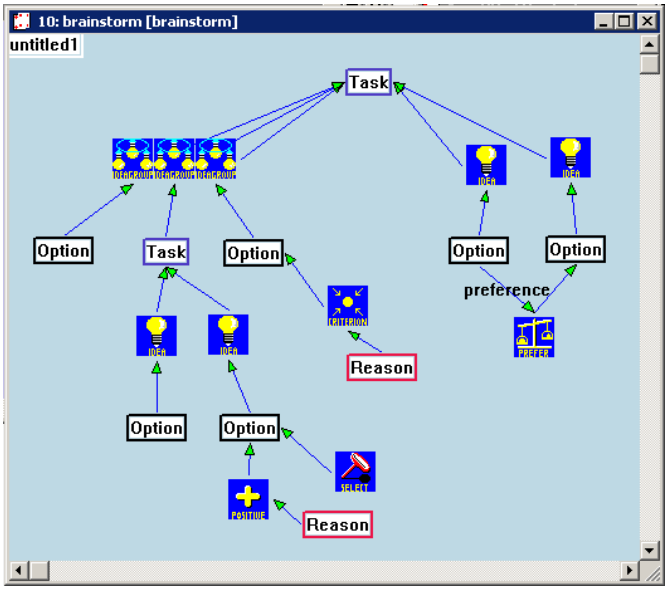


Abbildung 33

Schema für die Verwendung der Brainstormingsprache.

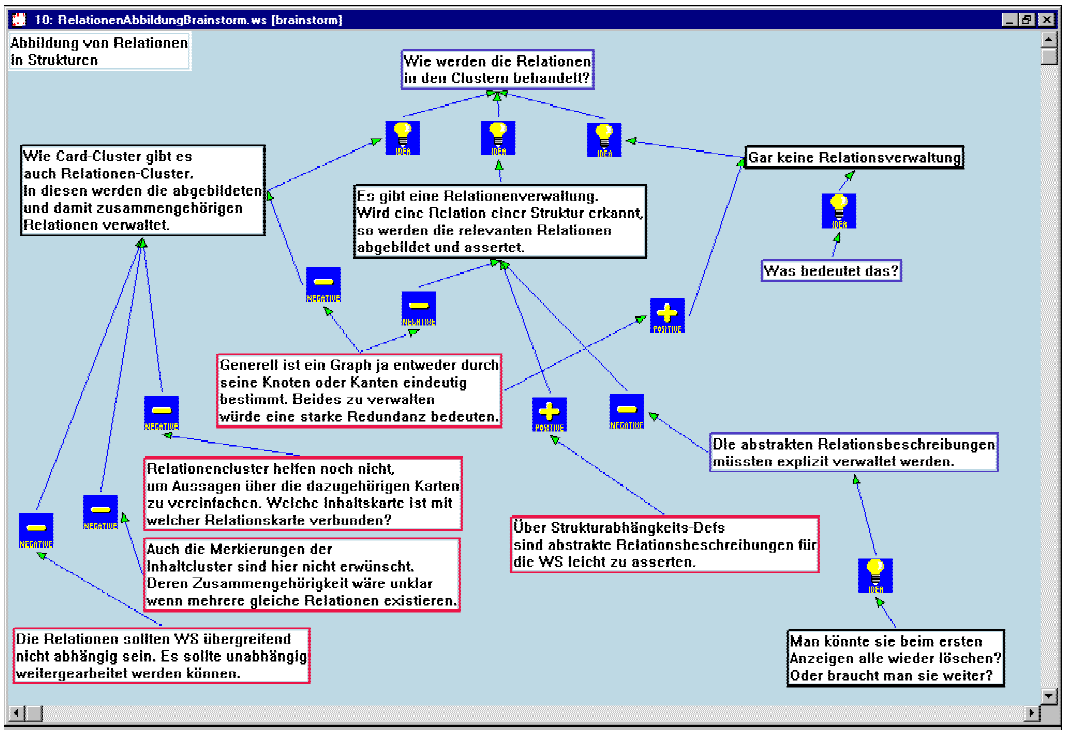


Abbildung 34

Beispiel für die Verwendung der Brainstormingsprache.

6.3.5 Die „Topic Map“

Die Diskussionsunterstützung soll es ermöglichen, punktuell Themen aufzuarbeiten, diese später weiter zu vertiefen und sie in ein Verhältnis zu setzen. Die Zuordnung von Themen zu Diskussionssträngen ist in zweierlei Hinsicht sinnvoll. Erstens können während der Diskussion verschiedene Aspekte jeweils Themen („Topics“) zugeordnet werden und so die Diskussion und die Inhalte strukturiert werden. Zweitens kann so ein Themengraph aufgebaut werden, die „Topic Map“[12], der sowohl die erarbeitete Wissensstruktur darstellt als auch einen Zugriff auf die jeweiligen Diskussionsstränge erlaubt.

Durch die Untergliederung von Problemen in Teilprobleme (Dekomposition) kann ein Problemlösungsprozess positiv beeinflusst werden. Im Allgemeinen dient sie dem Verständnis der Problemzusammenhänge und reduziert die Komplexität.

Im Diskussionsunterstützungssystem wird jeder Workspace mittels einer „Topic“-Karte einem Thema zugeordnet. In den vorangegangenen Abbildungen war diese jeweils oben links in den Workspaces zu erkennen (siehe auch Abbildung 35 und Abbildung 36). Konzeptuell muss jeder Arbeitsbereich genau einem Topic zugeordnet sein. Das heißt, dass jede visuelle Sprache auch die „Topic“-Karte als Kartentyp enthalten muss.

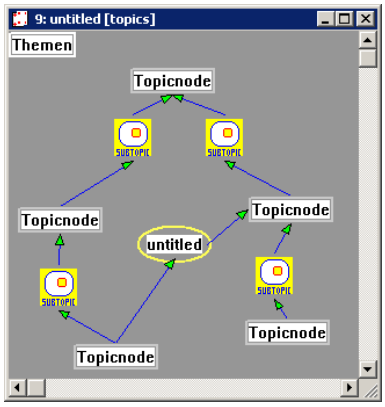


Abbildung 35

Schema für eine „Topic Map“.

Wegen der geforderten Zuordnung von einem „Topic“ zu einem Workspace unterliegt die „Topic“-Karte einer besonderen Kontrolle. Sie wird nicht frei zur Verfügung gestellt, sondern automatisch generiert. Nur der Inhalt muss von den Nutzerinnen und Nutzern hinzugefügt werden. Jeder generierte „Topic“ wird automatisch in die „Topic Map“ übernommen. Diese Übernahme basiert auf einem Abbildungsmechanismus, der in

[12] Diese entstehenden „Topic Maps“ sind nicht mit dem 1999 verabschiedeten ISO/IEC Standard 13250 über Topic Maps zu verwechseln. Diese können als Ontologien verstanden werden, die auf Ressourcen im Web verweisen, die entsprechend durch Topics klassifiziert worden sind.

Kapitel 6.6 vertieft vorgestellt wird. Zu jedem „Topic“ können die verschiedenen Perspektiven herangezogen werden. Der „Topic“ wird jedoch nur ein Mal in der „Topic Map“ erwähnt.

Die „Topic Map“ ist damit eine abstrahierte Sicht auf die Themen. In sie werden alle Themen, die in der Diskussion behandelt werden, automatisch integriert. Dementsprechend gibt es nur eine „Topic Map“ in der Diskussionsunterstützung.

Begrifflich ist es für die „Topic Map“ schwierig, die normalen Beitragskarten von dem „Topic“ zu unterscheiden, dem die „Topic Map“ selbst zugeordnet ist. Da es nur eine „Topic Map“ geben soll, ist deren „Topic“ einem Defaultwert zugeordnet („Themen“), wie es in Abbildung 35 und Abbildung 36 zu sehen ist.

Die restliche visuelle Sprache für die „Topic Map“ ist nur aus wenigen Kartentypen zusammengesetzt, mit denen Themen zueinander ins Verhältnis gesetzt werden können. Sie enthält ausschließlich einen Beitragskartentyp, den „Topicnode“, eine frei editierbare Relation und die Teilthema-Relation. Zwischen den Topics dürfen beliebig viele Relationen hergestellt werden (siehe Tabelle 14 und Abbildung 35, die die „Topic Map“ schematisch darstellt). Mit Hilfe der „Topicnode“-Karte können nicht nur solche Themen aufgeführt werden, die bereits in den anderen Perspektiven behandelt wurden, sondern auch neue eingebunden werden.

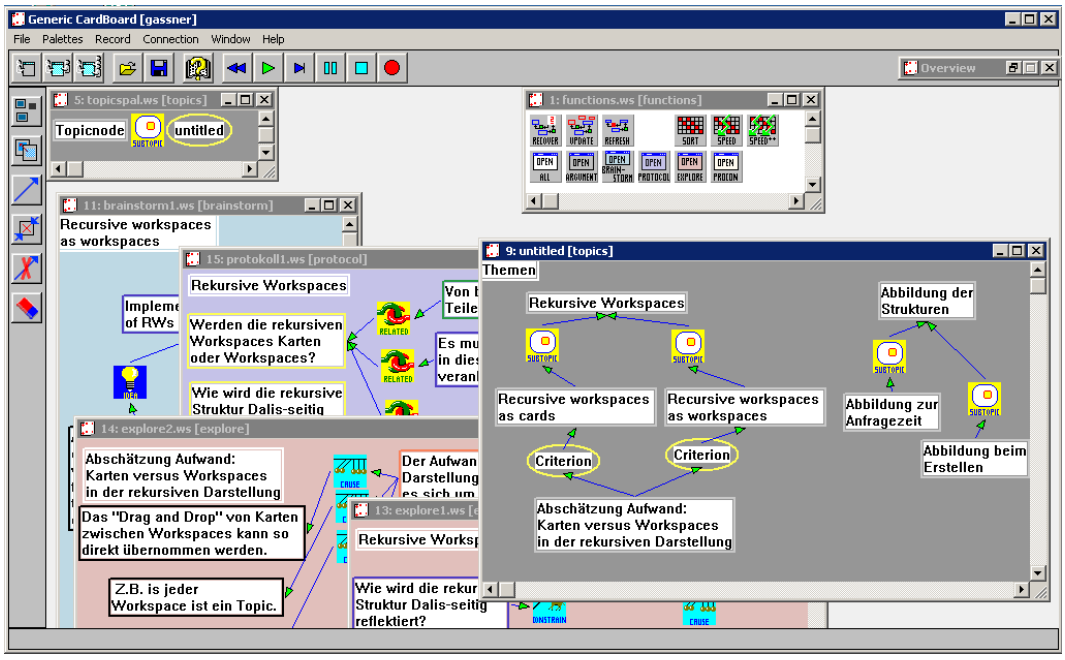


Abbildung 36

Beispiel einer „Topic Map“. Es werden existierende Topics mit neuen vermischt.

Die „Topic Map“ dient nicht nur der Übersicht über die behandelten Themen, sondern wird auch zur Arbeit mit den Themen herangezogen. Über die „Topic Map“ können

Workspaces zu einem bestimmten Thema und unter einer bestimmten Perspektive geöffnet werden. Dazu werden Funktionskarten verwendet, die im Kapitel 6.4 erläutert werden. Die Interaktion mit den „Topic“-Karten der verschiedenen Perspektiven und der „Topicnode“-Karten der „Topic Map“ unterliegt intern verschiedenen zusätzlichen Kontrollen.

	Im einzigen Beitragskartentyp werden die Themen festgehalten.
	Mit dieser Karte können beliebige Relationen zwischen Themen definiert werden.
	Es ist davon auszugehen, das die Teilthema-Relation besonders häufig genutzt wird. Dafür wird eine eigene Karte zur Verfügung gestellt.

Tabelle 14

Bedeutung der Beitrags- und Relations-Typen der Brainstormingsprache.

6.3.6 Die visuelle Sprache zum Argumentieren

In Diskussionen sind Argumente wichtige Bestandteile. Werden sie klar herausgearbeitet, so steuern sie nicht nur den Diskussionsverlauf, sondern sind gleichzeitig auch ein wichtiges Resultat einer Diskussion. Für die Diskussionsunterstützung wird deshalb davon ausgegangen, dass es sinnvoll ist, die entstandenen Argumente gesondert zu sammeln. Dadurch, dass sie als Teilstrukturen automatisch aus anderen Darstellungen herausgegriffen werden, werden sie u.U. überhaupt erst sichtbar. Zusammengefasst werden jeweils die Argumente zu einem „Topic“, die dafür aus anderen Perspektiven gefiltert werden müssen. Der Vorgang des Filterns wird ausführlich in den Kapiteln 6.7 und 6.8 beschrieben. Hier soll zur Erläuterung nur erwähnt werden, dass die im Folgenden definierten Argumentstrukturen automatisch vom System in verschiedenen Perspektiven erkannt, herausgegriffen und in der Argumentperspektive gesammelt werden. Zwischen den Workspaces wird also eine Relation definiert, auf deren Grundlage zu einem benutzerdefinierten Zeitpunkt die Argumente gefiltert werden. Die gefilterten Argumente können dann wieder frei strukturiert werden.

Argumentstrukturen werden typischerweise aus linguistischer Sicht definiert. Es gibt jedoch keine allgemein gültige Definition. Meißner (1994) weist auf die unterschiedliche Verwendung des Terminus „Argument“ hin. Einerseits stehe es für eine „Schlussfigur“, andererseits für „Sprechhandlungssequenzen [...], die als wesentliches Merkmal ein strittiges Moment aufweisen“ (Meißner, 1994, S. 25). Die Schlussfigur verweist auf die inhaltliche Struktur eines Argumentes. Es wird davon ausgegangen, dass ein Argument auf eine bestimmte Art und Weise strukturiert sein muss, z.B. muss verdeutlicht werden, wovon ausgegangen wird, und was man daraus ableitet. Stellvertretend für diesen Ansatz

ist die Arbeit von Toulmin (1958), die sich außerdem stark an logischen Aussagestrukturen orientiert.

Der Ansatz, das Argument als Sprechhandlungssequenz zu sehen, geht davon aus, dass bei einem Argument bestimmte Aussagen aufeinander folgen. Klein, mit der Theorie der konklusiven Sprechhandlungen (zitiert durch Meißner, 1994), vertritt diese zweite Auslegung. Die möglichen Sequenzen sind dabei Erklären-Warum, Begründen, Rechtfertigen, Folgern, welche sich wiederum aus Sprechakten zusammensetzen.

Für die angestrebte Diskussionsunterstützung soll die Toulmin'sche Schlussfigur Pate stehen. Dafür sprechen zwei Gründe: Erstens ist sie strukturorientiert und passt sich deshalb besser in den strukturorientierten Ansatz der Diskussionsunterstützung ein. Zweitens kann diese Struktur im Wesentlichen auch ohne Sprechakte nachgebildet werden. Das heißt, dass die Bestandteile der Struktur nicht als Sprechakte begriffen werden müssen.

Trotzdem sollen die von Klein beschriebenen Sequenzen Erklären-Warum, Begründen, Rechtfertigen und Folgern nicht völlig vernachlässigt werden. Auf der rechten Seite von Abbildung 37 ist jeweils ein Beispiel für diese vier Sequenzen aufgeführt, wie sie zumindest in einem Explorations-Workspace dargestellt werden können, wenn sie auch bisher nicht für den Argumentations-Workspace gefiltert werden.

Erklären-Warum ist „das Explizieren des Zustandekommens eines Sachverhalts" (Klein zitiert durch Meißner, 1994, S. 25). Dafür steht das erste Beispiel in der Abbildung rechts oben. Die Sequenz wird durch eine „Explain"-Karte ausgedrückt, die in diesem Fall ein Faktum als Voraussetzung angibt und auf ein Resultat verweist. „Rechtfertigen ist das Stützen des Anspruchs auf nichtnegative oder positive Bewertung ... eines Subjektes, ... eines Sachverhaltes, ... oder des Zustandekommens eines Sachverhaltes ..." (Klein zitiert durch Meißner, 1994, S. 26). Dafür steht das zweite Beispiel, das ein Resultat, das nicht erwünscht ist, durch einen persönlichen Umstand erklärt. Folgern „... soll das Ziehen einer Konsequenz aus Gegebenem bezeichnen ..." (Klein zitiert durch Meißner, 1994, S. 26). Hierfür steht das dritte Beispiel, bei dem ein Faktum die Voraussetzung für eine Einschätzung ist. „Begründen ist das Stützen von ... Ansprüchen in der Wahrheitsdimension ..." (Klein zitiert durch Meißner, 1994, S. 26). Diese Kategorie wird hier so interpretiert, dass ein gegebener Sachverhalt durch persönliche Einschätzungen begründet wird.

Da es sich um Darstellungen im Explorations-Workspace handelt, sind diese Beispiele als kurzer Exkurs anzusehen, der im Kontext der Argumentdarstellungen als Erläuterung

dienen soll. Sicherlich wünschenswert ist es, den Argumentfilter später um diese Argumentarten zu erweitern.

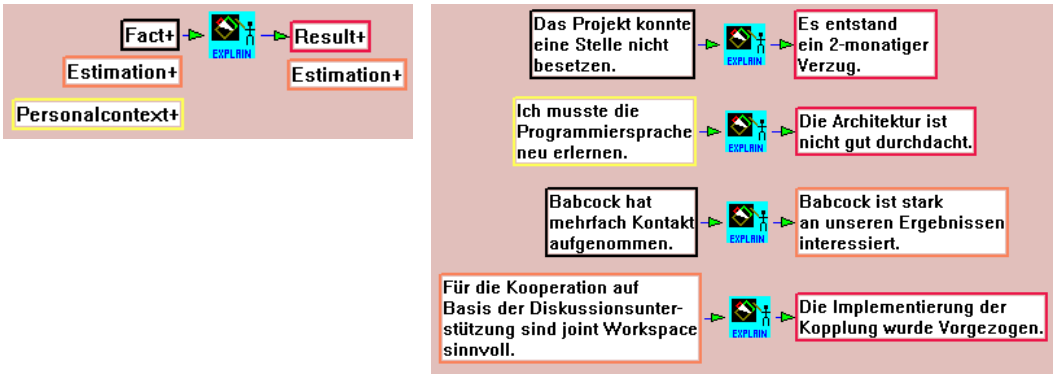


Abbildung 37

Die Argument-Sequenzen nach Klein werden im Explorations-Workspace alleine mit der „Explain"-Karte ausgedrückt (links). Auf der rechten Seite sind Beispiele dafür (folgern, rechtfertigen, begründen, erklären).

In Abbildung 38 wird die Argumentstruktur nach Toulmin dargestellt. Das Prinzip des Toulmin'schen Arguments ist eine Schlussfolgerung aufgrund von Daten, die in das Argument einbezogen werden. Darin besteht ein deutlicher Unterschied zu der Sicht nach Klein, der nicht nur Daten bzw. Fakten als Prämisse zulässt. In diesem Sinne kann die Toulmin'sche Schlussfigur auch als strenge Grundlage für wissenschaftliches Schließen interpretiert werden, wie es beispielsweise im Belvedere-System (siehe Kapitel 3.2) umgesetzt ist. Schlüsse aus nicht sicheren Voraussetzungen sind hingegen im Alltag üblich, wie es auch im Klein'schen Ansatz betont wird.

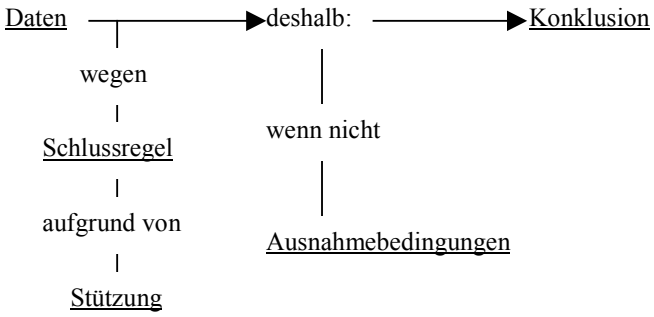


Abbildung 38

Die Argumentstruktur nach Toulmin (1958)

Neben den einfachen Daten, die nach Toulmin in ein Argument einfließen, können noch zusätzliche Prämissen eingebracht werden, die selbst wieder Daten sein können oder auch in weiteren Schlussregeln aufgeschlüsselt werden. Ein dritter Bestandteil sind die Ausnahmenbedingungen, unter denen das Argument nicht zutreffen würde. Toulmin problematisiert, dass die stützenden Daten sowie die Ausnahmebedingungen in den

meisten Argumentationen nicht explizit genannt werden. Das kann daran liegen, dass es schlicht überflüssig ist, sie zu nennen, weil diese Bestandteile entweder durch den Kontext oder das Vorwissen der Diskussionsteilnehmerinnen und -teilnehmer bekannt sind. Oft liegen hierin aber zentrale Ursachen für Missverständnisse oder falsche Argumentationen. Diese Tatsache kann auch als Hintergrund dienen, dass Visualisierungen zu sachlicheren Argumentationen führen können, wenn fehlende Bestandteile deutlich werden und fehlerhafte Aussagen zur Diskussion stehen.

Für die Diskussionsunterstützung wurde diese Toulmin'sche Struktur in eine Darstellung mit einer visuellen Sprache übertragen. Abbildung 39 (links) stellt die allgemeine Struktur eines Arguments in einem Argument-Workspace vor. Dabei sind die Kartentypen und deren Position in der Struktur relevant. Die Darstellung ist so zu lesen, dass bei nebeneinander oder untereinander stehenden Karten immer die eine oder die andere Karte verwendet werden kann. Das „+"-Zeichen in den Karten besagt, dass genau eine Karte davon in einer zulässigen Struktur an dieser Stelle vorhanden sein muss. Das „*"-Zeichen besagt, dass beliebig viele Karten dieser Art mit der entsprechenden Relation an der genannten Position in dem Argument auftreten können.

Abbildung 39

Allgemeine Argumentstruktur nach Toulmin in einem Argument-Workspace (links). Beispiel für eine Folgerung nach Toulmin (rechts).

Als Voraussetzung des Arguments wird die „Fact"-Karte verwendet, die durch die „Cause"-Relation auf die Konklusion verweist. Diese kann dann alternativ als Resultat oder Einschätzung kategorisiert sein. Die „Cause"-Relation macht den Kern des Arguments aus, der jeweils nur aus einer Voraussetzungs- und eine Konklusionskarte besteht. Diese Struktur kann insofern als Einschränkung bzgl. der Toulmin'schen gesehen werden, als dass sie jeweils nur eine Voraussetzungs- und eine Konklusionskarte vorsieht. Damit soll erst einmal eine überschaubare Grundlage für das Filtern der Argumente geschaffen werden. Um diese Einschränkung zu umgehen, könnten natürlich umfangreichere Inhalte in die Karten eingefügt werden. Der unterstützende Teil wird durch die „Support"- oder „Cause"-Karte in Verbindung mit weiteren Fakten oder

Einschätzungen aufgeführt. Einschränkungen werden mit der „Constrain“-Karte beigefügt. Eigentlich bezieht sich die Einschränkung des Arguments, die durch die „Constrain“-Karte ausgedrückt wird, auf den Schluss selbst, müsste also auf die „Cause“-Karte verweisen. Dies lässt sich aber weit schwieriger formalisieren, weshalb ich mich für diese Vereinfachung entschieden habe.
Das Beispiel auf der rechten Seite der gleichen Abbildung stellt ein mögliches Argument dar. Thematisch bezieht es sich auf das Systemdesign für rekursive Workspaces mit der Frage, ob diese eher als Karten oder eher als Workspaces entwickelt werden sollten. Ausgehend von der Annahme, diese auch als Workspaces zu modellieren, wird gefolgert, dass dann genau eine zentrale Eigenschaft der Karten verloren geht, nämlich sichtbar über Kanten miteinander in Relation gesetzt werden zu können, was aber auch für rekursive Workspaces wünschenswert wäre. Diese Folgerung wird relativiert, indem die Option aufgezeigt wird, dass die Workspaces sowohl von der Klasse Karte als auch von der Klasse Workspace sein könnten.
Die visuelle Sprache zur Darstellung der Argumente ist eine Teilmenge aus der Explorationssprache (Fact, Result, Estimation, Support, Cause, Constrain). Das liegt daran, dass in der Explorationssprache Argumente bereits darstellbar sein sollen. Das bedeutet aber nicht, dass sich Argumente in anderen Perspektiven nicht anders präsentieren. Z.B. werden auch bestimmte Strukturen der Brainstormingperspektive als Argumente erkannt, die dort sehr unterschiedlich aussehen und erst interpretiert werden müssen, um sie in die Argumentperspektive aufnehmen zu können. Auch handelt es sich im Falle des Filterns von Argumenten aus einem Explorations-Workspace nicht um eine triviale Übernahme. Für eine weiterführende Diskussion wird jedoch auf das Kapitel 6.7 verwiesen.

6.3.7 Die visuelle Sprache für die Pro-/Kontratabellen

Tabellen können oft gut dazu genutzt werden, um Lücken sichtbar zu machen. In der Pro-/Kontraperspektive können Tabellen dargestellt werden in denen Pro- und Kontra-argumente gegenübergestellt werden. Jede der Tabellen soll einem strittigen Aspekt gewidmet werden.
Wie für die Argumente wird auch für diese visuelle Sprache davon ausgegangen, dass Inhalte für die Tabellen aus den Explorations-Workspaces automatisiert übernommen werden können und eine Grundlage für weitere Diskussionen bieten. Auch hier kann nach dem Filtern frei mit den Tabellen gearbeitet werden. Wie dieses Filtern aussieht wird erst später, als Beispiel der Strukturabbildung erläutert (Kapitel 6.7 und Kapitel 6.8).

Abbildung 40 zeigt das Schema einer Pro-/Kontratabelle. Intendiert sind zweispaltige Tabellen, die durch die jeweilige „Pro“- bzw. „Kontra“-Konnektorkarte übertitelt sind. Für die Darstellung werden einfache Formatierungskarten in Form von vertikalen und horizontalen Linien angeboten. Über der Tabelle kann jeweils der strittige Aspekt angegeben werden.

Aspect	Überschrift für eine Tabelle: Der Aspekt, der durch Pro- und Kontraargumente erörtert werden soll.
Contra	Container für Kontrabeiträge.
pro	Container für stützende Beiträge.
	Formatierungskarten.
PRO	Spaltenmarkierung der stützenden Argumente.
CONTRA	Spaltenmarkierung der Kontraargumente.

Tabelle 15

Die Elemente der visuellen Sprache für die Pro-/Kontraperspektive.

Bei der freien Arbeit ist die Platzierung ausreichend, um optisch eine Zusammengehörigkeit der Argumente zur Tabelle bzw. zum behandelten Aspekt zu verdeutlichen. Die automatisch generierten Tabellen sehen jedoch jeweils Relationen zwischen Argumenten und Aspekten vor, um die Zusammengehörigkeit auch systemseitig explizit zu gestalten.

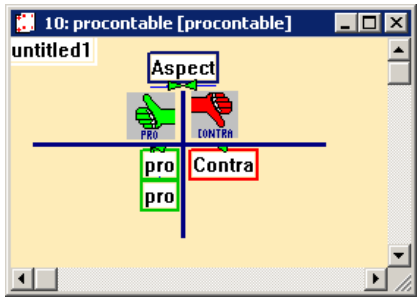


Abbildung 40

Schema der Tabellendarstellung.

6.4 Die Funktionspalette

In Kapitel 3.1 sind die drei Basis-Systeme für die Diskussionsunterstützung vorgestellt worden. Insbesondere das Kommunikationsinterface zwischen Dalis und dem CardBoard – vermittelt durch MatchMaker – wird für die Diskussionsunterstützung intensiv genutzt.

Im gleichen Kapitel wurde auch erläutert, dass das Interface im Wesentlichen auf den drei Nachrichten „create“, „delete“ und „modify“ basiert, über die mit zusätzlichen Parametern Informationen zwischen den Teilsystemen kommuniziert werden.
Diese Schnittstelle führt insofern zu Problemen, als dass sie sich ausschließlich auf Interaktionen auf den Objekten der visuellen Sprachen bezieht (Ausnahme bilden Daten über die Workspace selbst). Im Falle der für die Diskussionsunterstützung verwendeten Architektur wird ein Diskussionsmodell (Kapitel 6.8) in Dalis verwaltet. Dieses Modell steuert mehr als nur unmittelbare Reaktionen auf Karteninteraktionen in den Workspaces. Es hält eine Workspace-übergreifende Datenbasis vor, auf die auch beim Öffnen und schließen von Workspaces sowie für Sortierungen zugegriffen werden muss. Dies sollte an sich durch einen Dialog gesteuert werden. Auf Basis des vorliegenden Interfaces müssen solche Interaktionen jedoch auf Kartenoperationen abgebildet werden. Das bedeutet, dass Interaktionen, die auf die Daten in Dalis rekurrieren, immer in Form von Aktionen auf Karten ausgedrückt werden müssen, damit dies an Dalis übermittelt werden kann.
Um das zu erreichen, wurde eine Funktionspalette eingeführt. Die Funktionspalette ist selbst ein Workspace-Typ, für den eine eigene visuelle Sprache vorliegt. Die Karten werden jedoch nicht zum Aufbau von Graphen verwendet. Wird eine Karte dieser Palette in einen Workspace gezogen, so wird an Dalis eine entsprechende „create“-Mitteilung gesendet. Auf Basis des Kartentyps kann in Dalis eine vordefinierte Reaktion ausgelöst werden. Eine Rückmeldung von Dalis kann auf Kartenzustände und deren Vorhandensein Einfluss nehmen sowie auf die Workspaces selbst.
Abbildung 41 zeigt diese Palette. Die Karten darin bilden funktional drei Gruppen (Tabelle 16)

1. Verwaltung von Zustandsmarkierungen
2. Sortierungen
3. Funktionen zum Öffnen von Workspaces

Auf die Verwaltung der Zustandsmarkierungen wird vertieft in Kapitel 6.6 eingegangen und den Sortierungen widmet sich Kapitel 6.9, die Funktionen zum Öffnen werden im Folgenden erläutert:
Die Idee der Diskussionsunterstützung führt dazu, dass die Daten zu den Karten in den Workspaces persistent sein sollten, auch wenn diese während der Diskussion geschlossen werden. In dieser Situation liegen die wirklichen Objekte nicht mehr im DiscBoard vor und die Daten werden nur noch über Dalis vorgehalten. Es muss also ermöglicht werden,

die Daten in Dalis wieder anzufordern, wenn ein bestimmter Workspace erneut geöffnet werden soll. Dazu müssen zwei Aspekte gelöst werden:

- Der Workspace muss ausgewählt werden können und
- diese Auswahl muss, wie oben erläutert, über Aktivitäten auf Karten implementiert werden.

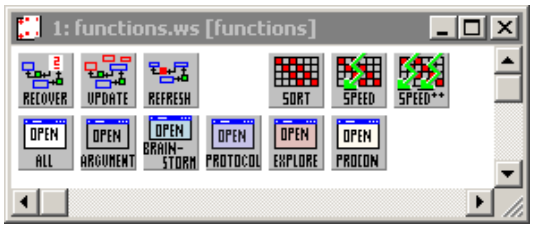


Abbildung 41

Die Funktionspalette.

RECOVER UPDATE REFRESH	Verwaltung von Zustandsmarkierungen
SORT SPEED SPEED**	Sortierungen
OPEN ALL, OPEN ARGUMENT, OPEN BRAIN-STORM, OPEN EXPLORE, OPEN PROTOCOL, OPEN PROCON	Funktionen zum Öffnen von Workspaces

Tabelle 16

Die Elemente der visuellen Sprache für die Funktionspalette.

Die Auswahl des Workspaces wird in Zusammenhang mit der „Topic Map“ realisiert (Kapitel 6.3.5). Die „Topic Map“ zeigt alle „Topics“ an, zu denen Workspaces bearbeitet wurden. Die Auswahl eines Workspaces erfolgt durch die Wahl des gewünschten „Topics“ in der „Topic Map“. Dazu wird eine Funktionskarte auf diesen „Topic“ gezogen. Um die Perspektive bestimmen zu können, unter der das Thema bearbeitet werden soll, steht für jede Perspektive eine eigene Funktionskarte zur Verfügung. Wird eine Funktionskarte zum Öffnen auf den entsprechenden „Topic“ gelegt, so bewirkt das, dass die Daten dazu aus Dalis angefordert werden. Auf Basis der Daten in Dalis wird dann ermittelt, welche Karte überdeckt wurde. Der Inhalt dieser Karte bestimmt dann das gewünschte Thema. Aus dem Typ der „Open“-Karte wird der Typ des angeforderten Workspaces erkannt, also die Perspektive. Daraufhin werden die angeforderten Daten übermittelt und der Workspace in der entsprechenden Sicht erzeugt.

Für dieses Vorgehen muss garantiert werden, dass jedes bis dahin bearbeitete Thema in der „Topic Map“ enthalten ist. Um das zu erreichen wird jedem Workspace erst einmal ein generierter (bedeutungsleerer) „Topic“ zugeordnet, der später angepasst werden kann. Diese „Topics“ werden automatisch in die „Topic Map“ übernommen, worin zusätzlich auch direkt neue Themen angelegt werden können.

6.5 Implementierungskonzept der multi-Perspektivik

6.5.1 Anforderungen und Ziele

Das DiscBoard implementiert einen Perspektivenansatz, der den Übergang zwischen den Perspektiven und den damit einher gehenden Arbeitsphasen unterstützt. Die eigentliche Perspektivik ergibt sich aus den verwendeten visuellen Sprachen und den darin darstellbaren Zusammenhängen und Beitragsarten, also den Darstellungsmitteln und nicht den konkreten Inhalten. Jeder Workspace der DiscBoard-Applikation ist einer visuellen Sprache zugeordnet und bietet damit eine bestimmte Perspektive an.

Die einzelnen Perspektiven stellen epistemische Formen zur Verfügung im Sinne von Collins und Ferguson (1993). Sie begleiten Diskussionsphasen gleichermaßen wie Diskussionsstränge, da den einzelnen Arbeitsbereichen Themen zugeordnet werden. Aus der Verwendung unterschiedlicher Perspektiven ergeben sich Perspektivenwechsel. Dieser Ansatz folgt damit den Forderungen aus Suthers und Xu (2002), dass es möglich sein muss, Artefakte – hier die einzelnen Diagramme – auf natürliche Art und Weise zu wechseln. Auch dort wird davon ausgegangen, dass dieser Wechsel des Artefaktes zu einem neuen Diskussionsstrang führt und dies explizit kenntlich gemacht werden sollte.

Mit dem hier vorgestellten Perspektivenansatz soll vor allem der Übergang zwischen den Perspektiven unterstützt werden. Die Übernahme, die hier nicht nur einzelne Beiträge, sondern auch zusammenhängende Strukturen betreffen kann, wird im weiteren als Abbildung bezeichnet. Geleistet wird dabei nicht nur die Duplizierung von Elementen, sondern auch eine Generierung von Knoten sowie Relationen.

Der Implementierung der Abbildungen liegen im Wesentlichen die folgenden Annahmen und Voraussetzungen zugrunde:

- Das systemgesteuerte Generieren von Knoten und Strukturen wirkt motivierend und koordinierend auf die Diskussion.
  - Die generierten Objekte bieten einen Ausgangspunkt für die weiterführende Elaboration. Sie dienen als Einstiegspunkt insbesondere dann, wenn durch besondere Strukturierungen Lücken in den Inhalten deutlich werden.
  - Automatisch übernommene Beiträge oder Strukturen haben eine Koordinationsfunktion bei der Diskussion, da durch sie deutlich werden kann, an welchen Punkten zusätzliche Diskussionen sinnvoll wären.
  - Eine Grundlage für weiterführende Diskussionen entsteht insbesondere dadurch, dass Strukturen, die aus anderen Perspektiven extrahiert wurden, automatisch

angeordnet werden. Es werden dadurch Muster angeboten, die strukturelle Aspekte hervorheben.

- Durch die Vorgaben soll verhindert werden, dass relevante Aspekte übersehen werden.
- Durch die Abbildung der Karten soll Zeit und unnötiger Aufwand gespart werden, denn es ist davon auszugehen, dass sonst mühsam die Objekte herausgesucht werden müssen, die noch nicht diskutiert wurden. Gerade in größeren Diskussionen, in denen gar nicht alle Arbeitsbereiche zu überblicken sind, ist das ohne Systemunterstützung beinahe unmöglich. Durch das Löschen der Systemvorschläge können diese übrigens leicht ignoriert werden. Das entspricht der Designidee, nämlich den Prozess nicht vorzuschreiben sondern flexibel anzuregen.

- Die Inhalte der Perspektiven entwickeln sich unterschiedlich, sie divergieren.
  - Im Sinne einer „representational guidance“ wird davon ausgegangen, dass sich die unterschiedlichen Perspektiven dazu eignen, verschiedene Inhalte hervorzulocken und zu entsprechend spezifischen Ergebnissen führen. Da die Perspektiven so gestaltet sind, dass es nicht sinnvoll ist, jeden Aspekt unter allen Perspektiven zu besprechen, ergänzen sich die Inhalte der Perspektiven eher, als dass sie sich überlagern. Entsprechend gibt es keine „Integrationsperspektive“, die alle Inhalte aus allen anderen Perspektiven konsistent zusammenfasst.
  - Die Übernahme von Inhalten aus anderen Perspektiven, stellt eine Diskussions*grundlage* dar. Es kommt zwar zu einer vorübergehenden Übereinstimmung von Teilen der inhaltlichen Beiträge zwischen den Perspektiven aber im Anschluss können sich die Inhalte unterschiedlich entwickeln. Im Verlauf der weiteren Diskussion divergieren die Inhalte in den Perspektiven.
  - Solange Inhalte von Knoten, die abgebildet wurden, noch übereinstimmen, wird diese Zusammengehörigkeit vom System verwaltet. Diese Zusammengehörigkeit muss bei der Nutzung einfach aufgehoben werden können.
- Es wird davon ausgegangen, das nach Themen diskutiert wird.
  - Die Zuordnung erörterter Inhalte zu Themen soll die inhaltliche Strukturierung der Diskussion forcieren.
  - Das teilweise Generieren einer „Topic Map“ aus den Themen, die einzelnen Workspaces und damit auch Gesprächsphasen zugeordnet werden, soll die Übersicht über die Diskussion verbessern.

  - Zu einem „Topic“ unter einer Perspektive kann nur ein Workspace geöffnet werden. Alle Beiträge dazu sollen in diesem einen Workspace platziert werden. Werden mehrere Workspaces mit der gleichen visuellen Sprache geöffnet, sind diese zwingend unterschiedlichen Themen zuzuordnen.
- Um einen Arbeitsprozess anregen und unterstützen zu können, muss Wissen über die Diskussionsabläufe im System repräsentiert werden. Das Diskussionsmodell definiert eine konkrete Annahme über Übergänge zwischen Diskussionsphasen.
  - Für die Diskussion wird davon ausgegangen, dass es zwar unterschiedliche Diskussionsphasen gibt, diese aber keiner definierten Reihenfolge unterliegen.
  - Das Diskussionsmodell definiert, welche Kartentypen oder Strukturen in andere Perspektiven abgebildet werden sollen.
  - Die Abbildungen der Karten und Strukturen beziehen sich auf die Typen der Karten und Relationen, nicht auf deren Inhalt. Das Diskussionsmodell definiert die Abbildungen in allgemeinen Regelschemata, die auf die „semantische Idee“, den Typ der Sprachprimitive, Bezug nehmen.

6.5.2 Konzept der Implementierung

Bereits vorgestellt wurden die visuellen Sprachen, die die Perspektiven bilden. Das Konzept zur Verwendung multipler Perspektiven sieht vor, dass diese nicht einfach nur abwechselnd verwendet werden können, sondern dass teilweise Informationen zwischen ihnen transferiert werden, die dabei potentiell strukturellen (topologischen) Veränderungen und geänderten Ordnungskriterien unterliegen. Abbildung 42 stellt diesen Vorgang schematisch dar: Für die Daten des grafischen Editors wird eine jeweils aktuelle Zustandsbeschreibung protokolliert. Auf Basis dieser Daten arbeiten die Produktionsregeln im Diskussionsmodell, die daraus Informationen ableiten und diese wieder an den Editor übermitteln.

Der Editor einschließlich der konkreten visuellen Sprachen ist Bestandteil des DiscBoards. Die Zustandsbeschreibung und das Diskussionsmodell werden in Dalis verwaltet. Das Diskussionsmodell besteht aus den Definitionen für die Abbildungen von Strukturen und Beiträgen zwischen den Perspektiven. Die Zustandsbeschreibung bildet die Datenbasis, die den Diskussionsstand vorhält. Dieser bezieht sich auf alle erarbeiteten Inhalte einer Diskussion, also auch die Daten für Karten und Strukturen, die gerade nicht angezeigt werden aber bereits erarbeitet wurden.

Das Zustandsmonitoring der aktuell angezeigten Daten ist eine Grundfunktion von Dalis und in Mühlenbrock (2001) ausführlich beschrieben.

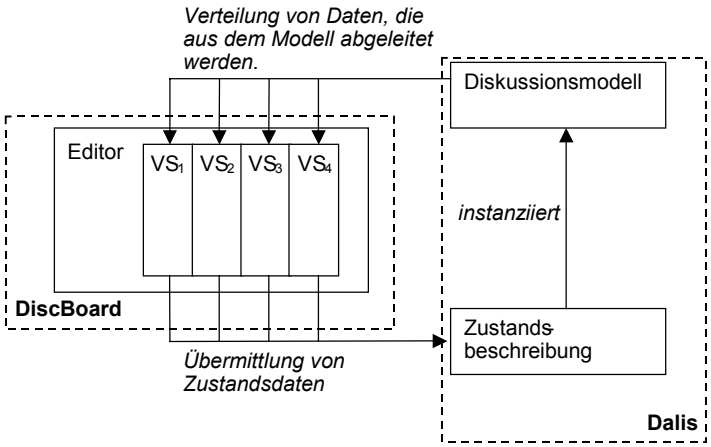


Abbildung 42

Datentransfer zwischen verschiedenen Systemkomponenten.

Struktur- und Knotenmodelle

Für die Diskussionsunterstützung müssen Informationen über die Übereinstimmungen zwischen den Perspektiven verwaltet werden. Diese Übereinstimmungen entstehen durch definierte Abbildungen zwischen den Perspektiven, die auf Systeminitiative hin vorgenommen werden. Abgebildet werden einzelne Beiträge und Teile der Diagramme, im Weiteren als Strukturen bezeichnet, die durch die Nutzerinnen und Nutzer im Verlauf einer Diskussion erstellt werden.

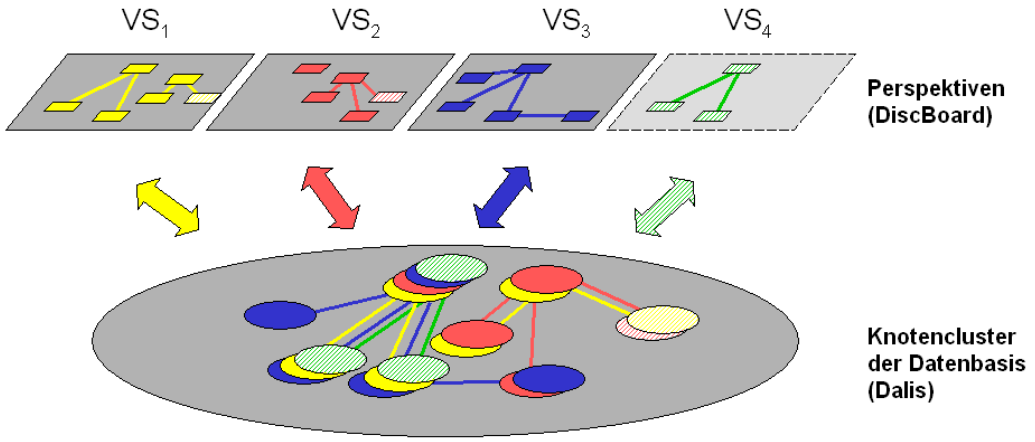


Abbildung 43

Über die Perspektiven wird die Datenbasis aus der DiscBoard-Applikation heraus modifiziert.

Abbildung 43 stellt schematisch die Trennung zwischen der Datenbasis auf Seite von Dalis und den Perspektiven im DiscBoard dar. Vier Workspaces, die unterschiedliche Perspektiven VS_1-VS_4 symbolisieren, sind dort in der oberen Ebene rautenförmig

dargestellt. Durch die Farben wird angedeutet, dass die Perspektiven jeweils Ausschnitte aus der Datenbasis zeigen.
Bei der Abbildung einzelner Beitragskarten werden die Inhalte dieser Karten mit in die neue Perspektive übernommen. Solange diese Inhalte nicht geändert werden, wird zwischen diesen Karten eine Abhängigkeit verwaltet: Dafür werden als Knotencluster bezeichnete Datensätze erstellt und gewartet.

> **Knotencluster** sind Datensätze der Datenbasis in Dalis mit denen die Zusammengehörigkeit von Knoten verwaltet wird. Alle Knoten eines Knotenclusters haben den gleichen Inhalt aber im Allgemeinen einen anderen Typ. Zu jeder Perspektive darf sich nur ein Knoten in einem Knotencluster befinden (siehe auch Folgekapitel 6.6).

Abbildung 43 symbolisiert auf der unteren Ebene solche Knotencluster für die Perspektiven. Jeweils die übereinander gelegten kleinen Ovale sind zusammen als Cluster zu verstehen. Farblich gekennzeichnet ist, aus welchen der vier Perspektiven Karten zu diesem Cluster gehören. Die durchbrochen umrandete Perspektive VS_4 mit den gemusterten Objekten deutet an, dass eine Perspektive nicht unbedingt angezeigt wird, trotzdem aber die Daten für die Knotencluster in der Datenbasis vorgehalten werden.
Die Abbildungen beziehen sich nicht nur auf nur einzelne Beitragskarten, sondern auch auf Strukturen. Beim Abbildungsvorgang werden dazu Strukturmodelle erstellt und in die Datenbasis aufgenommen.

> **Strukturmodelle** verwalten die Informationen zu erkannten Teilstrukturen. Sie setzten sich zusammen aus Kantentypen und Knotenclustern, die durch diese Kantentypen verbunden werden und der jeweiligen Perspektive, für die diese Information zutreffen soll.

Die Informationen, die in Dalis über die Abhängigkeiten von Karten und Strukturen verwaltet werden, müssen unabhängig von den konkreten visuellen Objekten sein, da Workspaces geschlossen werden können und so diese konkreten Objekte nicht mehr existieren. Deshalb wird hier auch von Strukturmodellen bzw. Knotenmodellen gesprochen. Über sie muss eindeutig definiert sein, welche Karten und Relationen zu erstellen sind, wenn ein bestimmter Workspace angefordert wird.
Die Datensätze können grob in drei Gruppen unterteilt werden (Abbildung 44): diejenigen, die die Modellinformationen, also die Abhängigkeiten verwalten (innerhalb der Raute in der Datenbasis), die, die die Referenz zu Oberflächenobjekten beifügen

sobald diese existieren (hinterlegt durch den gelben Pfeil) und drittens allgemeine Information, die beispielsweise Zugriffe auf Daten beschleunigen.

Abbildung 44 stellt die konkret verwendeten Prädikate vor. Zentral sind die Prädikate *cluster* und *structure* für die Modellinformationen der Strukturen und Knotencluster. Die Verbindung zur Visualisierung wird durch die Prädikate *in_structure*, *map* und *thema* geleistet. Sie verwalten für jede Karte, ob sie zu einer Struktur oder zu einem Cluster gehört und zu welchen. Den existierenden Workspaces werden Themen zugeordnet.

Die Prädikate *hash_topics* und *topic2clusters* sind für die Verwaltung von „Topic"-Informationen zuständig. Die „Topics" gehen nur in Form eines Identifikators in andere Fakten der Datenbasis ein, um so deren Änderung auf eine Stelle zu begrenzen (*hash_topics*). Nur aus Effizienzgründen existiert das Prädikat *topic2clusters*, dass für jedes Thema verwaltet, in welchen Knotenclustern es erwähnt wird. So müssen beim Generieren von Karten zu einem Thema nicht alle Knotencluster durchsucht werden, sondern es können die relevanten zielgerichtet angesprochen werden. Prinzipiell ist es möglich, dass sich ein Knotencluster über unterschiedliche Themen erstreckt. Insbesondere wird diese Option für die Übernahme der „Topics" in die „Topic Map" verwendet.

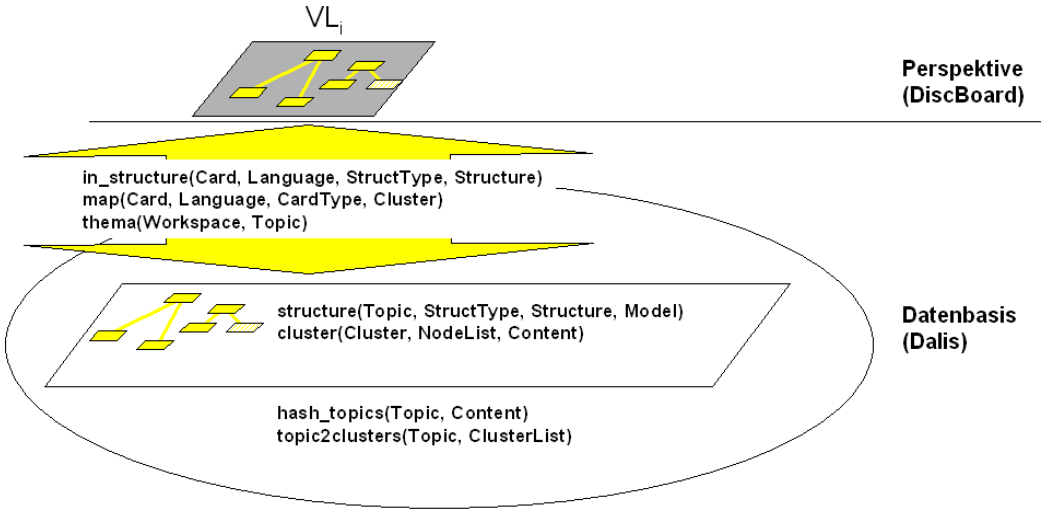


Abbildung 44

Einträge der Datenbasis.

User-gesteuerte Inhaltspropagierung („delayed changes")

Werden Karten abgebildet, so besteht eine Abhängigkeit der Inhalte zwischen den Karten, die auseinander hervorgegangen sind. Wenn nun der Inhalt einer Karte, für die ein solcher Zusammenhang zu anderen besteht, unter einer Perspektive geändert wird, so bestehen

zwei Möglichkeiten, damit umzugehen. Entweder werden die Inhalte sofort angeglichen oder später, unter der Entscheidung der Nutzerinnen und Nutzer. Für das DiscBoard wurde entschieden, dass diese Änderungen nicht sofort in die dazugehörigen Karten propagiert werden. Ein Grund dafür ist die Annahme, dass trotz der Abhängigkeit die Karten in den verschiedenen Perspektiven in unterschiedliche Kontexte eingebettet sind. Eine Inhaltsänderung kann also wegen der Perspektivänderung zustande kommen. Dies bedeutet dann nicht unbedingt, dass die vorangegangenen Inhalte selbst nicht mehr relevant sind. Deshalb wurde die Entscheidung getroffen, den Nutzerinnen und Nutzern die Änderungen offen zu lassen. Dafür werden die Änderungen in Form von Zustandsmarkierungen angezeigt. Dieses Vorgehen wird als „delayed changes“ bezeichnet.
Es sind aber nicht nur die Änderungen von Karten durch diese Design-Entscheidung betroffen, sondern auch das initiale Generieren von Karten. Würden diese neuen Karten gemäß der Abbildungsvorschriften sofort auch unter den anderen Perspektiven angezeigt werden, so würde dies dort relativ unmotiviert das „optische Erscheinungsbild“ maßgeblich beeinflussen. Es ist davon auszugehen, dass das zur Verwirrung der Nutzerinnen und Nutzer führt. Auch hier wird deshalb nach dem Prinzip der „delayed changes“ vorgegangen, das heißt, dass zusätzliche Karten explizit angefordert werden müssen.
Dieses Prinzip wird als besonders wichtig für den Fall von Kartenlöschungen angesehen. Automatisches Löschen führt zum Verdacht, dass Inhalte verloren gegangen sind, was beim automatischen Aktualisieren auch wirklich geschehen kann. Löschungen von Workspace-Inhalten finden deshalb ebenfalls zu einem von Usern definierten Zeitpunkt statt und werden ebenfalls durch „Marker“ angezeigt.
Das Konzept des „delayed deletion“ wird auch in Stefik et al. (1987) eingesetzt. Dort wird argumentiert, dass widersprüchliche Inhalte gerade auch eine Diskussion anregen können und erst einmal nicht gelöscht werden sollten.

Verwendung struktureller statt räumlicher Eigenschaften

In Ansätzen zum formalen Schließen auf visuellen Darstellungen wird oft davon ausgegangen, dass der Ort der Objekte und deren räumliche Relation zueinander (z.B. neben, über, berührt, etc.) eine Rolle spielen und dass deshalb solche Informationen mit in die formale Repräsentation des Zustandes aufgenommen werden müssen.
Wang und Lee (1993) und Wang, Lee und Zeevat (1995) argumentieren, dass diese räumlichen Informationen nur dann sinnvoll verwendet werden können, wenn in diese bereits Wissen über die Domäne einfließt, in der Schlussfolgerungen angestellt werden sollen. Das liegt insbesondere daran, dass bestimmte Relationen nur für bestimmte

Domänen gelten. Beispielsweise kann ein Flaschenzugproblem visualisiert werden und die Objekte, aus denen die konkrete Kombination aus Flaschenzügen zusammengesetzt werden, können als Seile mit den jeweiligen Verbindung, Rollen und Gewichte repräsentiert werden (Beispiel aus Larkin und Simon, 1987). Eine anderes Beispiel besteht in der diagrammatischen Darstellung von Syllogismen (Wang, Lee & Zeevat, 1995; Stenning & Oberlander, 1995). Aber auch allgemeinere Eigenschaften wie „Nähe" oder „Farbe" mit potentiellen Schlussfolgerungen müssen im Allgemeinen anwendungsabhängig interpretiert werden.

Insofern spielen beim visuellen Schließen nicht unbedingt allgemeine räumliche Parameter eine Rolle, sondern domänenspezifische Aspekte einer zweidimensionalen Darstellung.

Im Zusammenspiel von Dalis und DiscBoard werden die Zustandsdaten an Dalis übermittelt und dort überwacht. Diese Zustandsdaten enthalten auch die Objektkoordinaten und –dimensionen (Abbildung 45). Auf dieser Basis können dort auch Schlüsse implementiert werden, die diese räumlichen Daten ausnutzen. Im Rahmen der Diskussionsunterstützung wird aus diesen Informationen bisher jedoch noch keine Bedeutung abgeleitet. Ausgenutzt werden sie nur für die automatische Anordnung der Objekte. Insofern können die Abbildungen als topologische bzw. strukturelle Schlüsse charakterisiert werden!

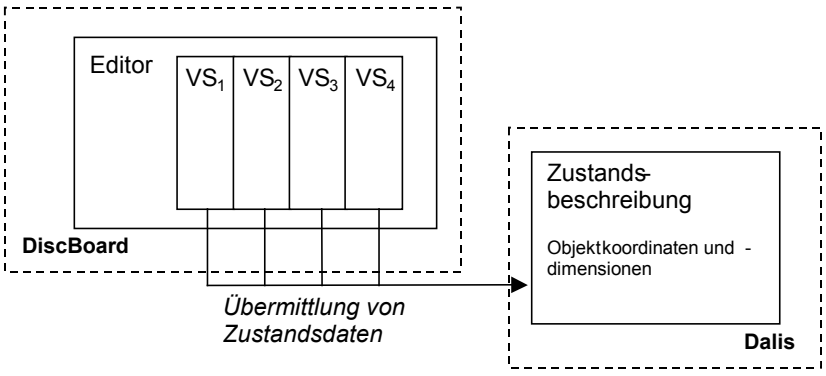


Abbildung 45

In den Zustandsdaten in Dalis werden auch Objektkoordinaten und –dimensionen vorgehalten.

Lokales Parsen: Vermeidung globalen Parsens

Unter globalem Parsen ist hier zu verstehen, dass die gesamte Darstellung überprüft bzw. geparst wird. Die Ziele eines solchen globalen Parsens können beispielsweise darin bestehen, Schlussfolgerungen aus dieser Darstellung zu ziehen oder sie in eine andere

Darstellung zu transformieren. Das globale Parsen ist im Normalfall allerdings sehr zeit- und rechenaufwändig und soll für die Diskussionsunterstützung vermieden werden.
Ein pragmatischer Grund dafür liegt auch in der Art, wie diese Strukturen entstehen. Sie werden allmählich, während der Arbeit, entwickelt. Beispielsweise kann ein Argument mehrere Voraussetzungen haben, die erst nach und nach hinzugefügt werden. Das heißt, dass nach sehr vielen Interaktionen im Workspace eine Strukturüberprüfung durchgeführt werden muss, die deshalb nicht übermäßig aufwändig sein sollte.
Der zentrale Grund für das lokale Parsen liegt jedoch darin, dass die bedeutungstragenden Strukturen gerade als Teile der Gesamtdarstellung definiert sind. Da es sich bei den bedeutungstragenden Strukturen, um verbundene Graphen handelt, kann eine neue Struktur oder eine Erweiterung einer bestehenden immer nur entstehen, wenn eine Relation eingefügt wird. Wird ein solcher Teil identifiziert, ausgehend von einer Benutzerinteraktion, braucht die restliche Darstellung nicht mehr geprüft werden. Das lokale Parsen geht deshalb von einem Objekt aus, konkret einem Link als Bestandteil einer Relation, um ein verändertes Diagramm auf die Entstehung einer Struktur zu untersuchen.
Die Ausdrücke der visuellen Sprache brauchen nicht als Ganzes geparst werden, da sie in der Diskussionsunterstützung im Prinzip immer als korrekt anzusehen sind, denn alle Objekte der visuellen Sprache dürfen ohne Einschränkungen verwendet werden.

Klasse der erkennbaren Strukturen

Für die zu erkennenden Strukturen können in Bezug auf die Anzahl von Karten und Relationen keine eindeutigen Definitionen angegeben werden, wohl aber für die Art und Weise, wie diese Strukturen gebildet werden. Im Rahmen dieser Arbeit wurde ein Algorithmus zum lokalen Parsen entwickelt, der Strukturen erkennen kann, die auf die folgende Art und Weise aufgebaut sind: Diese Strukturen teilen sich auf in einen Strukturkern, das ist die minimale Struktur, und in Erweiterungen. Die Art des Strukturkerns sowie die Art der einzelnen Erweiterungen wird jeweils als Einheit beschrieben. Der Aufbau der gesamten Struktur findet dann nach und nach statt. Darauf basiert ein inkrementelles Vorgehen bei der Erkennung und Wartung von Strukturen.
Die Strukturarten, die damit erkannt werden können, werden im Folgenden erst einmal unabhängig von einer konkreten visuellen Sprache erläutert. Die Beschreibung der konkreten Strukturen folgt in den nächsten Kapiteln.

> **Ausdruck in einer visuellen Sprachen.** Sei VS eine visuelle Sprache mit einer Menge $Label_{N_VS}$ von Knotenarten und einer Menge $Label_{E_VS}$ von

Kantenarten. Die Funktionen φ_{N_VS} und φ_{E_VS} liefern zu jedem Knoten bzw. jeder Kante das entsprechende Label. Sei weiterhin N_{VS} die Menge der Knoten, E_{VS} die Menge von Kanten eines Graphs mit $E_{VS} \subseteq N_{VS} \times N_{VS}$. Ein Graph ist definiert als

$$G_{VS} = (N_{VS}, E_{VS}, Label_{N_VS}, Label_{E_VS}, \varphi_{N_VS}, \varphi_{E_VS})$$

und wird als Ausdruck in einer visuellen Sprache bezeichnet.

Die Definition für einen Ausdruck in einer visuellen Sprache soll verdeutlichen, dass im hier entwickelten Ansatz alle Ausdrücke in allen visuellen Sprachen als Graphen repräsentiert sind. Die Bezeichnung „Label“ wird hier neu eingeführt und entspricht dem Typ oder der Art von Kanten und Knoten, wie sie im Vorfeld verwendet wurden. Während „Typ“ eher auf eine Bedeutung verweist, verdeutlich „Label“, dass in dieser und den folgenden Definitionen vor allem auf die Syntax Bezug genommen wird und den Knoten und Kanten ein Name zugeordnet wird.

Erkennbare Struktur. Eine Instanz einer erkennbaren Struktur besteht aus genau einem Strukturkern und beliebig vielen Extensionen. Der Strukturkern sowie alle Arten von Extensionen müssen jeweils für sich als Einheit definiert werden. Eine erkennbare Struktur ist immer ein echter oder unechter Teilgraph eines Ausdruckes einer visuellen Sprache, kann sich also nicht über mehrere visuelle Sprachen erstrecken. Es gilt immer:

$N_{VS_Struktur} \subseteq N_{VS}$, $E_{VS_Struktur} \subseteq E_{VS}$,
$Label_{N_VS_Struktur} \subseteq Label_{N_VS}$, $Label_{E_VS_Struktur} \subseteq Label_{E_VS}$.

Im Folgenden werden die Bildungsvorschriften für Strukturkerne und Extensionen sowie deren Kombination anhand von kurzen Graphgrammatiken erläutert.
Abbildung 46 stellt die Grammatik für Strukturkerne vor, Abbildung 48 die Grammatik für die Extensionen sowie deren Kombination mit den Kernen.
Die in Abbildung 46 dargestellten Produktionen 1-3 für Strukturkerne verdeutlichen, dass diese kaum beschränkt sind, sondern beinahe beliebige Graphen zu einer visuellen Sprache sein können. Die einzige Einschränkung besteht darin, dass der „kleinste Strukturkern“ aus mindestens zwei Knoten und einer Kante bestehen muss. Regel 1 stellt diese Beschränkung dar, indem aus dem Startsymbol G_{Kern} gleich ein solcher Kern hervorgeht. Regel 2 ermöglicht die Erweiterung der Strukturen um weitere Knoten, die entweder über eine ausgehende oder eine eingehende Kante mit dem Ausgangsgraphen

verbunden sein müssen. Regel 3 ermöglicht, die aus Regel 1 und 2 entstehenden Ketten aus Knoten und Kanten, auch untereinander zu verbinden.

Als Einbettungsregel für die hier eingeführte Graphgrammatik gilt, dass alle Kantenverbindungen, die vorher zu einem Knoten bestanden haben, nach der Ersetzung wieder hergestellt werden.

Auch wenn für jeden Strukturkern eine Grammatik angegeben werden kann, so werden die Kerne sowie die Extensionen später im Diskussionsmodell nicht über eine solche Grammatik definiert, sondern als Menge von Relationen- und Knotenarten. Der Grund besteht vor allem darin, dass die tatsächlich anfallenden Strukturen, denen eine Bedeutung zuerkannt wird, sehr einfach sind. Beispielsweise kann die Grammatik für den Strukturkern einer Argumentstruktur wie in Abbildung 47 angegeben werden.

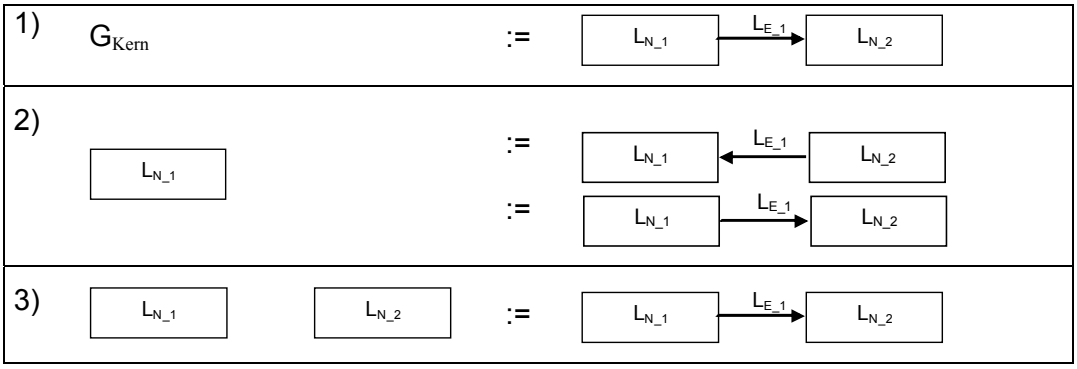


Abbildung 46

Eine Graphgrammatik für Strukturkerne, in der zwischen erweiterbaren und nicht erweiterbaren Knoten unterschieden wird. Ansonsten ist jeder gerichtete Graph ein zulässiger Strukturkern.

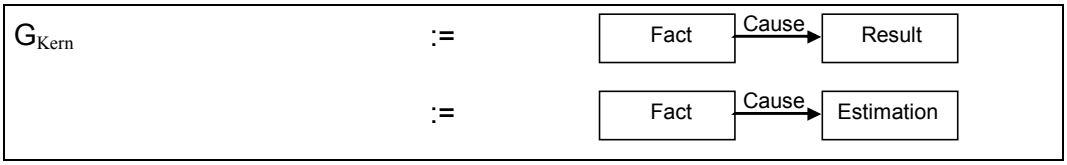


Abbildung 47

Grammatik für die Bildung zweier Ausprägungen des Strukturkernes für eine Argumentstruktur.

Abbildung 48 beinhaltet die Erweiterungsregeln des Kerns. Relevant ist darin die Unterscheidung in die rechteckigen und ovalen Knoten. Die rechteckigen Knoten sind noch Bestandteil des Strukturkerns, während die ovalen Bestandteil der Extension sind. Die durchbrochen gezeichneten Knoten des Kerns sind explizit ausgezeichnete Knoten, an die die Extensionen angebunden werden können. Somit beschreiben die Regeln 4 und 5, wie Extensionen an den Kern angebunden werden können und die Regeln 6 und 7 beschreiben den Aufbau der Extensionen selbst. Regeln 6 und 7 entsprechen den Regeln

2 und 3, über die wiederum allgemein ein gerichteter Graph aufgebaut werden kann. Sie sind hier hauptsächlich aus Gründen der Vollständigkeit noch einmal aufgeführt und zeigen, dass auch die Extensionen in sich keinen Beschränkungen unterliegen.

Im Prinzip kann jeder Knoten eines Strukturkernes erweitert werden. Welche konkret erweitert werden dürfen, wird über die Regeln für die Extensionen definiert (Regel 4). Deshalb befindet sich diese Regel in der Grammatik für die Extensionen und nicht in dem Teil für die Kerne. Die Erweiterbarkeit bezieht sich ausschließlich auf den Typ des zu erweiternden Knotens, spezifiziert durch das jeweilige Label. Für eine konkrete Struktur müssen die jeweiligen Label konkretisiert werden.

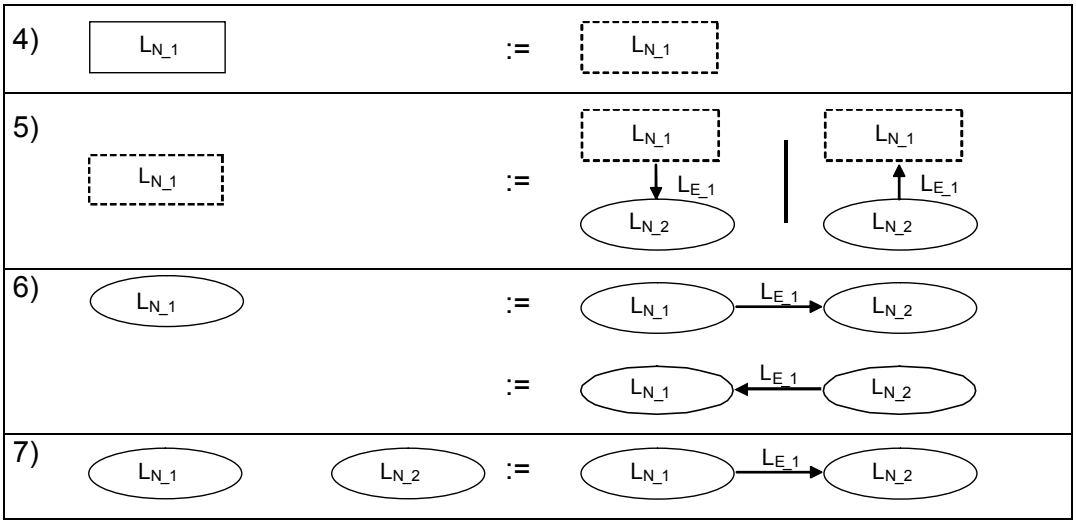


Abbildung 48

Die Grammatik für die Extensionen mit der Anbindung an die Strukturkerne.

Zu bemerken ist, dass Regel 5 mehrfach angewendet werden kann. Es können also sowohl mehrere Extensionen mit einem Knoten des Kerns verbunden werden, als auch eine Extension über mehrere Relationen.

Dadurch, dass nur die Strukturkerne erweitert werden können, denn in der Grammatik besteht keine Erweiterungsregel für die Extensionen, werden keine rekursiven Erweiterungsmöglichkeiten erlaubt. Das heißt, dass keine Extensionen von Extensionen definiert werden können. Die Extensionen können zwar als Klasse einen beliebigen Graph bilden, werden dann aber als „Ganzes“ betrachtet. Das heißt auch, dass der Parser keine rekursiven Extensionen erkennt. Der Grund dafür besteht in der konkreten Nutzung für die Diskussionsunterstützung, die keine solchen komplexen Strukturen verwendet.

Abbildung 49 zeigt ein Beispiel für Grammatikregeln für die Argumentstruktur. Darin wird das Beispiel aus Abbildung 47 aufgegriffen und der„Result“-Knoten des Argumentkerns erweitert.

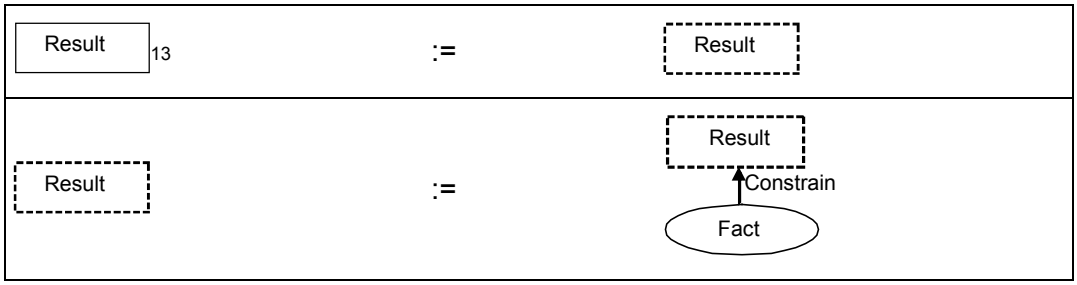


Abbildung 49

Extension des Strukturkerns der Argumentstruktur über einen Knoten vom Typ „Result“.

Abbildung 50 veranschaulicht mögliche Topologien, die mit der Implementierung zum lokalen Parsen erkannt werden könnten, hier auch komplexere als solche, die für die Diskussionsunterstützung definiert sind. Die Strukturkerne sind jeweils rot hervorgehoben. In den Beispielen a)-c) sind nur solche Extensionen enthalten, die jeweils aus einer Relation bestehen. Der aus zwei Knoten bestehende Strukturkern in Beispiel a) wurde mit 4 Extensionen erweitert.

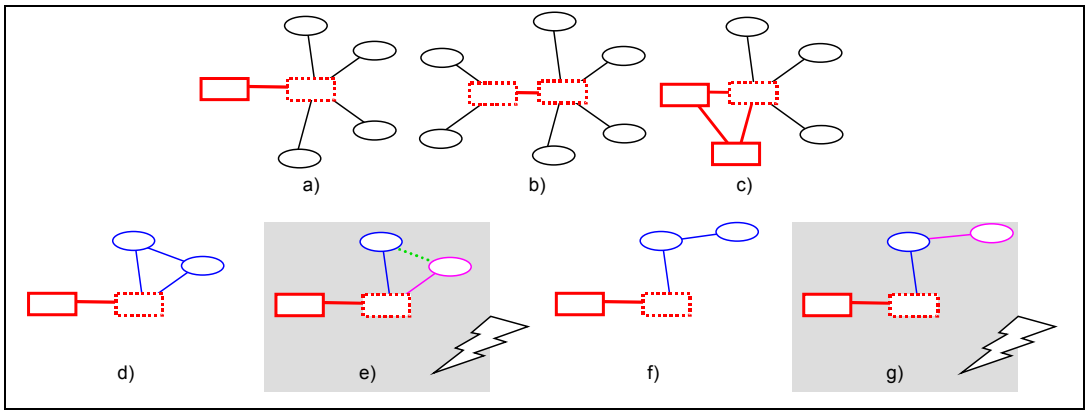


Abbildung 50

Mögliche Topologien für Strukturen, die mit der Implementierung zum lokalen Parsen erkannt werden könnten. Die roten Bestandteile symbolisieren jeweils den Strukturkern, die anderen die Extensionen.

Beispiel c) beinhaltet einen Kern, der aus drei Knoten und drei Relationen besteht. Beispiel d) zeigt eine Extension (blau), die mit dem Kern über zwei Relationen verbunden ist. Nicht definierbar ist die Struktur aus Beispiel e)! Darin ist ein Knoten des Kerns durch zwei Extensionen erweitert worden (angedeutet durch unterschiedliche Farben). Eine Verbindung dieser Extensionen (grün, durchbrochen) kann im vorliegenden Ansatz nicht definiert und nicht erkannt werden. Beispiel f) zeigt ein Beispiel mit einer Extension, die aus zwei Knoten und zwei Relationen besteht (blau). Im Gegensatz kann

[13] Werden im Strukturkern mehrere Knoten gleichen Typs eingebunden und ist außerdem die Position des Knotens in der Topologie relevant ist, so müssen die linken Seiten dieser Regeln den Strukturkern darstellen.

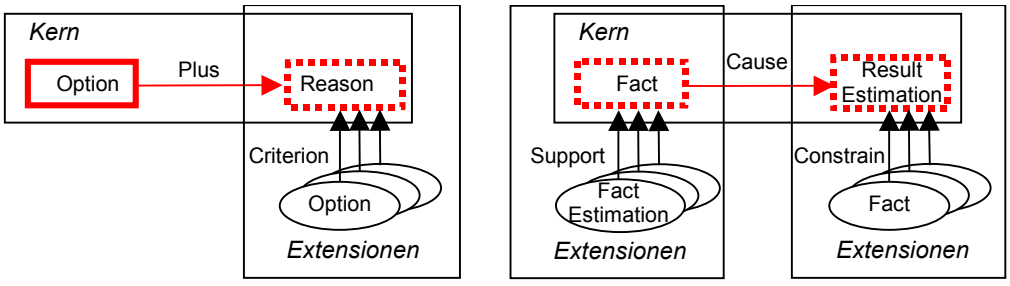
das Beispiel g) nicht definiert und erkannt werden. Der Graph ist zwar topologisch identisch zu f) aber es werden, angedeutet wieder durch die unterschiedlichen Farben, zwei Ausdrücke aneinander gereiht.

Multiple Perspektiven: Die Abbildungen

Die Diskussionsunterstützung operiert auf den oben beschriebenen lokalen Strukturen, indem diese zwischen den Perspektiven abgebildet werden. Im vorangegangenen Abschnitt wurde der Aufbau der erkennbaren Strukturen erläutert. Bei einer Abbildung von Strukturen ist nun außerdem der Typ dieser Strukturen relevant, da pro visueller Sprache mehrere Strukturtypen spezifiziert werden können.

> Ein **Strukturtyp** T ist durch eine bestimmte Topologie charakterisiert, die in einer Strukturgrammatik ausgedrückt werden kann. Für eine visuelle Sprache VS darf ein Strukturtyp T nur über genau eine Bildungsvorschrift definiert sein. Ist ein Strukturtyp T für mehrere visuelle Sprachen VS_i definiert, so sind die Topologien bzw. deren Bildungsvorschriften unabhängig voneinander.

Die Argumentstruktur ist ein Beispiel für einen Strukturtyp. Sie ist für die Explorationssprache und die Brainstormingsprache definiert, in denen sich das Argument jeweils in einer anderen Topologie darstellt. Abbildung 51 stellt diesen Strukturtyp dar und greift dazu die Darstellungsform aus Abbildung 50 auf. Die Form und Farbe der Knoten bezieht sich entsprechend nicht auf die Typen der Objekte der visuellen Sprache, sondern kennzeichnet ihre Zugehörigkeit zur Extension oder zum Kern.



Der Strukturtyp „Argument“ in der Brainstormingsprache

Der Strukturtyp „Argument“ in der Explorationssprache

Abbildung 51

Der Strukturtyp „Argument“.

Die Abbildung einer Struktur unterteilt sich in zwei Phasen:

- Erkennen einer Struktur und deren Typs.

- Abbildung der Struktur auf Ebene der internen Modelldarstellungen.

Die spätere Darstellung der abgebildeten Strukturen ist nicht mehr Bestandteil der eigentlichen Abbildung. Ein Strukturtyp muss explizit nur für die visuellen Sprachen definiert sein, aus denen Strukturen diesen Typs abgebildet werden sollen. Nur dort müssen die Strukturen identifiziert werden. In der Zielsprache einer Abbildung sind die Strukturtypen nur implizit definiert, indem die Modelle einem Typ zugeordnet werden. Bei einer konkreten Abbildung wird die neu generierte Struktur dem gleichen Strukturtyp zugeordnet, der bereits für das Urbild zutraf. Der Strukturtyp wird also bei einer Abbildung beibehalten.

Abbildungen von Strukturen werden inkrementell durchgeführt. Das bedeutet einerseits, dass sie schrittweise durchgeführt werden, beginnend mit dem Strukturkern. Werden dann Extensionen erkannt werden diese jeweils in einem weiteren Schritt zusätzlich abgebildet. Andererseits bedeutet inkrementelles Abbilden hier auch, dass das schrittweise Erkennen von Bestandteilen einer Struktur zur deren Ergänzung führt und nicht die gesamte Struktur neu generiert wird. Die neuen Bestandteile werden dem bestehenden Modell hinzugefügt.

Abbildung eines Strukturkerns. Sei S_{VSx_T} ein Strukturkern in einer visuellen Sprache VS_x zu einem Strukturtyp T, dann führt eine Abbildung dieser Struktur in eine visuelle Sprache VS_y zur Generierung, das heißt zum Anlegen eines Modells einer Struktur gleichen Typs T in der visuellen Sprache VS_y, wobei $x \neq y$ gelten muss:

$$a: S_{VSx_T} \rightarrow S_{VSy_T}$$

Modell einer Struktur. Eine Relation r ist in einem Strukturmodell beschrieben durch den Typ zweier Knoten, $Label_{N1_VS}$ und $Label_{N2_VS}$, eine Funktion c: $Label_{N_VS} \rightarrow$ Content, die den Inhalt der Knoten liefert und den Typ der Relation: $r_{VS} = (Label_{N1_VS}, Label_{N2_VS}, c_{N_VS}, Label_{E_VS})$. Sei T ein Strukturtyp sowie VS_x und VS_y die visuellen Sprachen der Ausgangs- und Zielsprache einer Abbildung, dann besteht ein Strukturmodell M aus den Teilmodellen $M_{VSx_T} = \{r_{1_VSx}, ..., r_{n_VSx}\}$ und $M_{VSy_T} = \{r_{1_VSy} ..., r_{m_VSy}\}$.

Abbildung einer Extension. Die Abbildung einer Extension führt zur Erweiterung der Menge der Relationen, die im Modell M einer Struktur enthalten sind: Sei M_{VSx_T} das Teilmodell einer Struktur S_{VSx_T} in der Sprache VS_x und M_{VSy_T} das Teilmodell der selben Struktur in der Sprache

VS_y, dann führt die Abbildung der Extension a: $S_{VSx_T} \rightarrow S_{VSy_T}$ zu einer Erweiterung der Modelle $M_{VSx_T} \cup \{r_{j_VSx}, ..., r_{k_VSx}\}$ und $M_{VSy_T} \cup \{r_{s_VSy}, ..., r_{t_VSy}\}$.

Es ist möglich, dass die Abbildungen zu nicht strukturgleichen Bildern führen.
Ob eine Struktur neu entstanden ist, wird immer bei einer Erstellung eines Links untersucht, der potentiell eine Relation vervollständigt. Dabei ist es im Prinzip möglich, dass eine entstandene Relation eine Grundlage für mehrere Strukturen darstellt oder auch, dass sie gleichzeitig einen neuen Strukturkern darstellt und für eine bestehende Struktur eine Extension bildet oder Extension für unterschiedliche Strukturen ist. Es müssen in diesem Fall mehrere Abbildungen durchgeführt werden.

Mehrfachabbildungen. Wird ein Teilgraph eines Ausdruckes einer visuellen Sprache VS_x als Kern oder Extension unterschiedlicher Strukturtypen $T_{1\text{-}m}$ erkannt, so werden alle Abbildungen a_i: $S_{VSx_Ti} \rightarrow S_{VSy_Ti}$ durchgeführt, für alle i mit $1 \leq i \leq m$.

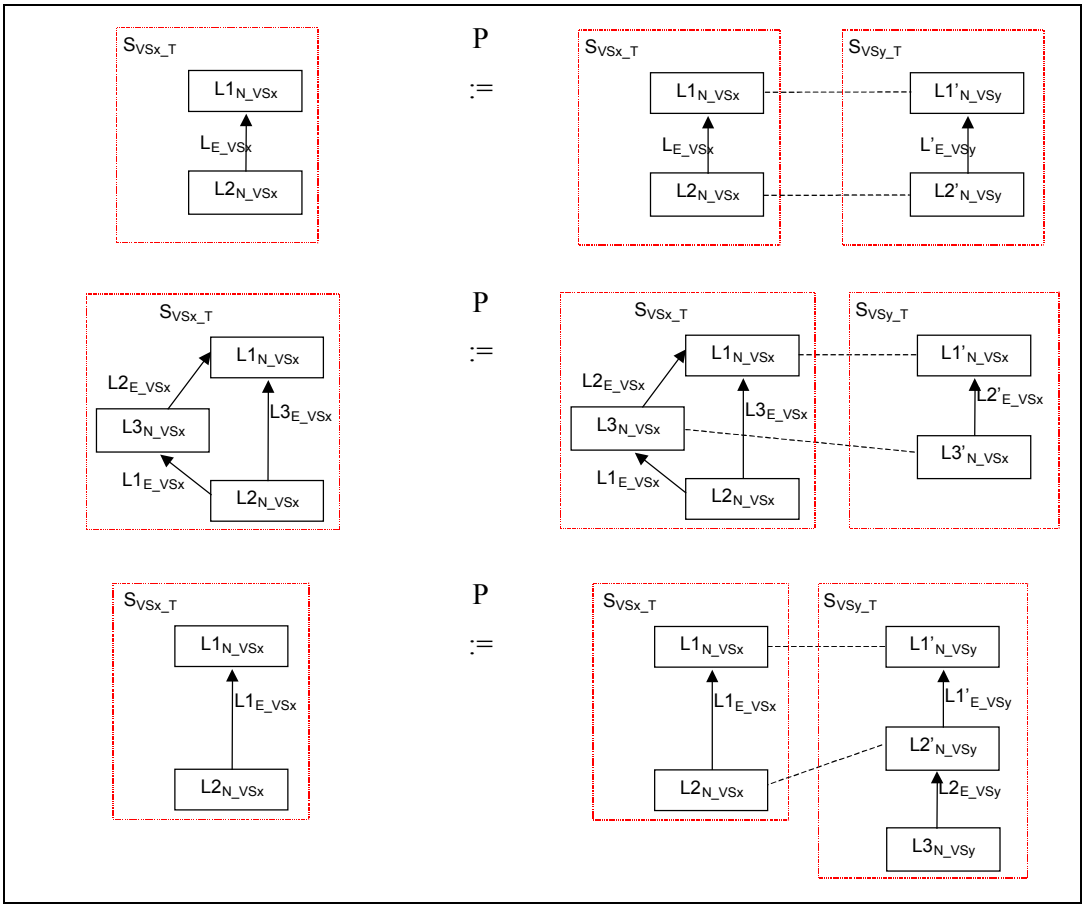


Abbildung 52

Beispiele für Produktionen, die eine Abbildung von Strukturen zwischen visuellen Sprachen darstellen.

Im Folgenden wird diese Abbildungen ebenfalls abstrahiert mit Hilfe von Graphgrammatiken beschreiben. Wie auch bei Rekers (1994) führt das zu verschiedenen Problemen. Graphgrammatiken eignen sich eigentlich nicht dazu, um multiple Perspektiven zu beschreiben, denn sie sagen nur etwas über Transformationen aus. Der Ansatz der Abbildungen führt aber zum Generieren neuer Graphen in anderen Perspektiven, die an sich nicht miteinander in Verbindung stehen. Dieses Generieren könnte zwar auch als graphgrammatische Produktionsregel notiert werden, ohne Einbettungsregeln anzugeben. Das würde aber eine Erweiterung bereits generierter Graphen ausschließen, da die Zusammengehörigkeit der Knoten oder auch Kanten verloren ginge.
Die gewählte Darstellung der Abbildungen lehnt sich an die von Rekers (1994) an. Darin wird das Generieren von Graphen auf die Transformation zurückgeführt. Dafür werden ausgezeichnete Kanten mit erstellt, die letztlich die Verbindung der Graphen verdeutlichen.
Sei $VS_{1\text{-}m}$ eine Menge von visuellen Sprachen und sei S_{VSx_T} eine Struktur zum Strukturtyp T in der Sprache VS_x aus dieser Menge $VS_{1\text{-}m}$. Dann definiert eine Produktion P die Erweiterung des Graphen um eine Struktur S_{VSy_T}. In Abbildung 52 sind Beispiele für solche Produktionen dargestellt, wobei explizit Kanten generiert werden, wenn Knoten auseinander hervorgehen (durchbrochene Kanten). Hier ist diese Verbindung als Zugehörigkeit zu einem Knotencluster zu interpretieren. Die rot und durchbrochen gezeichneten Rechtecke markieren die Teilgraphen, sind aber kein Bestandteil der Graphgrammatik.

6.6 Abbildung von Karten: Knotencluster

Um den Diskussionsprozess zu unterstützen, werden Karten automatisch von einem Workspace in einen anderen abgebildet. Als Ergebnis dieser Abbildung wird ein Knotencluster generiert, das das Modell der nun abhängigen Knoten bildet. Dafür wird in einem ersten Schritt auf Basis des Diskussionsmodells festgestellt, ob überhaupt eine Abbildung stattfinden soll. Dann wird das Diskussionsmodell nach allen zutreffenden Abbildungsvorschriften durchsucht, in denen auch definiert wird, in welche Perspektiven abgebildet werden soll. Darauf folgt die eigentliche Generierung eines Knotenclusters.
Abbildung 53 macht die Zusammengehörigkeit von Knoten zwischen den Perspektiven deutlich. Dargestellt sind die Perspektiven VS_1, VS_2 und VS_3. Unter jeder Perspektive können unterschiedliche Diagramme entstehen. Nur wenn eine Abbildung ausgeführt wird, führt das dazu, dass ein Knotencluster angelegt wird. Diese Abbildung muss nicht

von einer Perspektive in alle anderen geschehen. So sind in Knotencluster A und B beispielsweise drei Karten aus drei Perspektiven referenziert, während Knotencluster C und D nur zwei Karten referenzieren. In den Knotenclustern ist ausschließlich die Information zu Karten enthalten, keine Information zu Relationen. Die Abbildung führt nicht unmittelbar auch zum Anzeigen der Oberflächenobjekte. Erst durch ein „Refresh“ unter einer Perspektive werden auf Grundlage des Knotenclusters die erforderlichen Karten generiert.

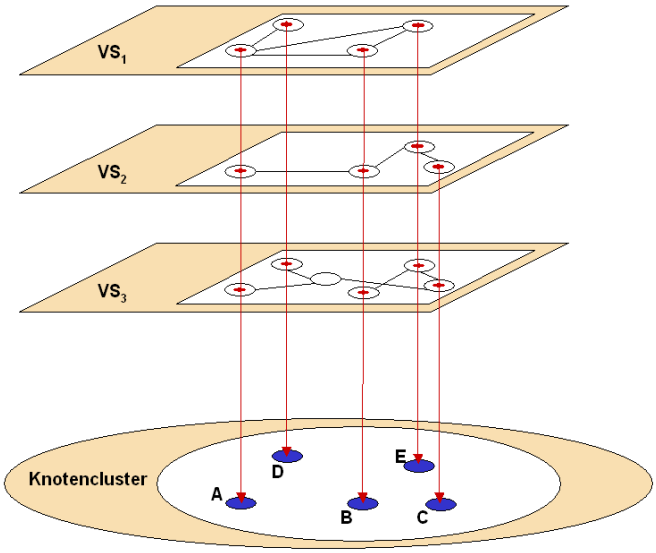


Abbildung 53

Abhängige Knoten werden intern in Knotenclustern verwaltet.

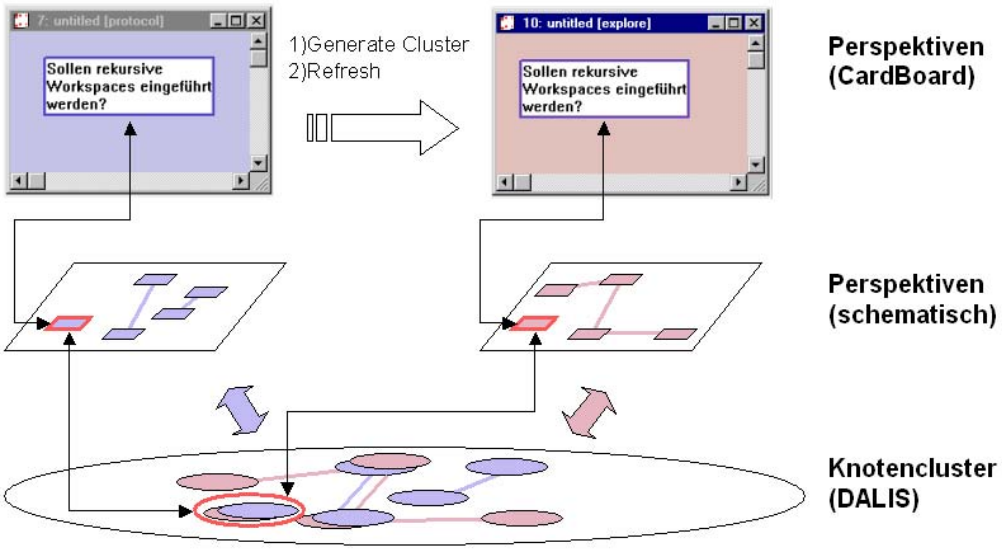


Abbildung 54

Beispiel für die Generierung neuer Karten.

Abbildung 54 zeigt dazu ein Beispiel: Symbolisiert werden dort wieder die Knotencluster in Dalis und die Perspektiven. Darüber ist jeweils ein konkreter Workspace zu erkennen wie er im DiscBoard erscheint. In dem Beispiel wird versucht, einen Ablauf zu verdeutlichen. Ein Beitrag, der links oben in der Protokollperspektive erstellt worden ist, führt zu einer Abbildung dieser Karte in die Explorationsperspektive oben auf der rechten Seite.

Aus der Protokollperspektive werden Beiträge, die als „Task“ ausgewiesen sind, automatisch in eine Karte vom Typ „Subject“ der Explorationsperspektive übernommen. Dafür wird das hervorgehobene Knotencluster erstellt und in die Datenbasis von Dalis mit aufgenommen.

In die nach dem „Refresh“ generierte Karte wird der Inhalt der Karte übernommen, die als Urbild der Abbildung diente. Dabei handelt es sich um eine zentrale Designentscheidung, diese Übereinstimmung explizit zu verwalten. Dadurch soll erreicht werden, dass Änderungen, die in einem Workspace vorgenommen werden, auch in die dazugehörigen Perspektiven propagiert werden können. In so einer Verwaltung von Inhalten und Zusammenhängen besteht ein Vorteil der Computernutzung zu nicht-digitalisierten Mitschriften, in denen dies kaum leisten in.

Die Cluster werden als Fakten in der Prolog-Datenbasis abgelegt[14]:

```
cluster(? Cluster, + Nodes, ? Content).
```

Die Cluster-Information beinhaltet die eindeutige Id des Clusters (*Cluster*), eine Liste dazugehöriger Knotenmodelle (*Nodes*) und den Inhalt, den diese Karten besitzen sollen (*Content*). Da davon ausgegangen wird, dass Knoten nur so lange in Abhängigkeit stehen, wie sie den gleichen Inhalt haben, steht der Inhalt *Content* stellvertretend für alle Karten eines Clusters. Die Cluster verwalten prinzipiell keine direkte Referenz zu Oberflächenobjekten. D.h. es wird in den Clustern weder eine Referenz zu Karten-Ids verwaltet, noch zu Workspace-Ids.

Bekannt sein muss die visuelle Sprache, der Kartentyp und das Thema. Aus diesen Informationen kann ein Oberflächenobjekt generiert werden. Auch ein Zustand *State* einer Karte wird verwaltet, der aber erst weiter hinten in diesem Kapitel erläutert wird. Entsprechend setzt sich die Knotenliste *Nodes* eines Clusters zusammen:

[14] In der Implementierung taucht das Prädikat *cluster* nicht direkt auf, sondern wird nur als Term in *db(cluster)* verwendet. Aus Gründen der Übersichtlichkeit wird im weiteren auf die Darstellung des Prädikates *db* verzichtet. Dies gilt entsprechend für alle Fakten, die in die Datenbasis eingetragen werden. Formal ergeben sich daraus formal kaum Unterschied. Für den Zugriff auf die Fakten der Datenbasis bot sich in der Implementierung jedoch eine Vereinheitlichung der Prädikate an.

```
cluster(Cluster, [ (Language1, Type1, Topic1, State1) | Nodes], Content).
```

Dadurch, dass der „Topic“ für jede Karte im Cluster mit aufgenommen wird, können auch themenübergreifende Cluster definiert werden. Ausgenutzt wird das für das Generieren der „Topic Map“.

6.6.1 Das Generieren der Knotencluster

Wird eine Karte in einem Workspace angelegt, so wird automatisch geprüft, ob dafür ein Cluster angelegt werden muss. Das geschieht auf Grundlage des Diskussionsmodells, in dem die Abbildungsdefinitionen zwischen den Kartentypen enthalten sind. Diese Abbildungen werden mit dem Prolog-Prädikat *translate* definiert.

```
translate (– Language1, – Type1, – Topic1, + Language2, + Type2, + Topic2).
```

Dieses Prädikat bildet die Vorschrift, eine Karte vom Typ $Type_1$, mit der visuellen Sprache $Language_1$ und dem Thema $Topic_1$ in eine Sprache $Language_2$, mit dem Kartentyp $Type_2$, in das Thema $Topic_2$ abzubilden. Die Abbildungsvorschriften beziehen sich jeweils auf einen einzelnen Kartentyp mit Bezug auf eine visuelle Sprache. Soll ein Cluster erstellt werden, müssen deshalb alle zutreffenden Abbildungsvorschriften ermittelt werden, da für jede Abbildung in eine Perspektive eine eigene Vorschrift definiert ist. *get_translation_list* erstellt eine Liste dieser gültigen Abbildungen und erstellt daraus Informationen, die später in das Cluster übernommen werden.

```
get_translation_list(– Language1, – Type1, – Topic1, + Nodes):-
        setof((   Language2, Type2, Topic2, default),
                translate(Language1, Type1, Topic1, Language2, Type2, Topic2),
                + List),
        append([ (Language1, Type1, Topic1, default) ], List, + Nodes), !.
```

Mit dem Prädikat *generate_cluster* wird für die resultierende Liste (*List*) das Cluster erstellt und in der Datenbasis abgelegt. Existiert ein Oberflächenobjekt, also eine Karte, zu der Information im Cluster, so wird eine Referenz (*map*) zwischen der visualisierten Karte und dem in der Datenbasis repräsentierte Cluster erstellt. Zum Zeitpunkt der Cluster-Erstellung wird diese Referenz deshalb nur für die die Abbildung auslösende Karte angelegt, da die entsprechenden Oberflächenobjekte nicht sofort mit generiert werden.

```
generate_cluster(– Language, – Type, – Card, – Topic, – Content):-
        get_translation_list(Language, Type, Topic, + Nodes),
        unique_db_id(+ Cluster),
        …
        assert( map(Card, Language, Type, Cluster) ),
        assert( cluster(Cluster, Nodes, Content) ), !.
```

Als Gegenpol zu der Prozedur der Erstellung der Cluster muss auch das Generieren von Karten aus den Clustern implementiert werden. Um die generierten Cluster anzuzeigen steht den Nutzerinnen und Nutzern die „Refresh“-Karte aus der Funktionspalette zur Verfügung. Wird diese Karte in einen Workspace gelegt, so wird dessen Inhalt auf Grundlage der Datenbasis erweitert. Tatsächlich handelt es sich dabei um eine monotone Operation, denn durch sie kann die Anzahl der Karten in einem Workspace nur erweitert werden oder gleich bleiben. Bestehende Karten fallen durch diese Operation in keinem Fall weg und werden auch nicht verändert.
Für das Generieren der Karten werden auf Basis eines gewünschten Themas und einer Perspektive die relevanten Cluster ermittelt. Dabei handelt es sich um einen einfachen Zugriff auf eine Information der Datenbasis, da aus Effizienzgründen für jedes Thema verwaltet wird, in welchen Clustern es vorkommt.

```
refresh(_, _, – Topic, – Language):-
        …
        topic2clusters(Topic, + Clusters),
        refresh_card_cluster(_, _, Clusters, Language),
        ….
```

Das Prädikat *refresh_card_cluster* führt dazu, aus den Clustern der Datenbasis die Oberflächenobjekte zu generieren. Rekursiv wird die Liste der Cluster durchlaufen. Mit der jeweiligen Id kann das konkrete Cluster aus der Datenbasis angefragt werden. Dieses liefert dann die Daten, um die Oberflächenobjekte zu generieren.

```
refresh_card_cluster(_, _, [], _) :- !.

refresh_card_cluster(_, – Workspace, – [ Cluster | Clusters ], – Language) :-
        cluster(Cluster, + Nodes, + Content),
        create_cards(_, Workspace, Cluster, Nodes, Language, Content),
        refresh_card_cluster(_, Workspace, Clusters, Language), !.
```

Dieses Generieren der Oberflächenobjekte wird durch das Prädikat *create_cards* implementiert, in dem die folgenden Fälle unterschieden werden:

Der erste Fall verhindert durch die Abfrage *map*, die nur zutrifft, wenn das Oberflächenobjekt schon existiert, dass Karten, die bereits angezeigt sind, durch ein weiteres „Refresh“ doppelt generiert werden.

```
create_cards(_, – Workspace, – Cluster, – [ (Language, Type, _, _) | Nodes],
                – Language, – Content):-
        map(_, Language, Type, Cluster),
        create_cards(_, Workspace, Cluster, Nodes, Language, Content), !.
```

Einen Sonderfall stellt die Behandlung der „Topic“-Karten dar, da sie immer in die linke, obere Ecke positioniert werden sollen.

```
create_cards(_, – Workspace, – Cluster, – [ (Language, topic, _, State) | _],
                – Language, – Content):-
        create_card(_, type(topic), pos(0, 0), ws(Workspace), content(Content), id(Card)),
        assert( map(Card, Language, topic, Cluster)), !.
```

Das Prädikat *create_card* ist bereits Bestandteil der Schnittstelle zwischen MatchMaker und Dalis und muss für eine konkrete CardBoard-Applikation, wie hier die Diskussionsunterstützung, nicht implementiert werden. Wird eine relevante Information in der Knotenliste gefunden und *create_card* aufgerufen, so terminiert die Rekursion von *create_cards*, da pro Cluster für eine vorgegebene Sprache immer nur ein relevanter Eintrag in der Liste vorhanden ist.
Der dritte Fall von *create_cards* ist der, der zum standardmäßigen Generieren einer Karte führt. Hier wird auch der Zustand relevant, der für die Karten mit verwaltet wird, der dazu führen kann, dass neu zu generierte Karten sofort auch mit einer Zustands-markierung versehen werden. Eine solche Markierung besteht selbst aus einer Konnektorkarte und einem Link auf den Beitrag. Diese Zustandsmarkierungen werden für das Konzept der „delayed changes“ verwendet.

```
create_cards(_, – Workspace, – Cluster, – [ (Language, Type, _, State) | _],
                – Language, – Content):-
        get_position(X, Y),
        create_card(_, type(Type), pos(X, Y), ws(Workspace), content(Content), id(Card)),
        mark(_, Card, Workspace, X, Y, Type, State),
        assert( map(Card, Language, Type, Cluster) ), !.
```

Als Designentscheidung wurde festgelegt, dass die Kartenmodelle in den Clustern keine Information über deren Platzierung verwalten sollen. Dem liegt die Annahme zugrunde, dass nur in ganz wenigen Fällen davon ausgegangen werden kann, dass eine Karte an der

gleichen Position in einem Workspace generiert werden soll an der auch ihr Urbild sich befand. Dies führt dazu, dass beim Generieren von Karten auch eine Position vorgegeben werden muss. *get_position* bezieht sich also nicht auf die Position des Urbildes, sondern stellt eine Schnittstelle zu Sortierungsberechnungen dar.
Die letzten Fälle von *create_cards* bilden die Suche einer relevanten Information in der Knotenliste und die Terminationsbedingung.

```
create_cards(_, – Workspace, – Cluster, – [ (_, _, _, _) | Nodes], – Language, – Content):-
        create_cards(_, Workspace, Cluster, Nodes, Language, Content), !.

create_cards(_, _, _, [], _, _):-!.
```

6.6.2 Zustandsmarkierungen

Bisher wurden schon einige Male die Zustandsmarkierungen angesprochen. Den Hintergrund für diese Markierungen bildet das folgende Problem: Implementiert wird eine Abbildung von Karten zwischen Perspektiven. In den beschriebenen Knotenclustern werden dafür die Abhängigkeiten verwaltet. Eine Abhängigkeit bedeutet, dass wenn an einer Karte, die aus einem Knotencluster referenziert wird, der Inhalt geändert wird, wird erst einmal davon ausgegangen, dass diese Änderung potentiell auch für die anderen Karten des Clusters wichtig ist. Beispielsweise könnte eine Karte in der Explorationsperspektive, die eine Prämisse für ein Argument darstellt, in die Argumentperspektive übernommen werden. Wird an der einen Karte etwas geändert, so scheint es plausibel, diese Änderung auch in die andere Karte zu übernehmen. Jedoch sind auch Fälle denkbar, in denen das gerade nicht gewünscht ist. So kann eine übernommene Prämisse in der Argumentperspektive verfeinert werden, behält aber ihren Sinn in der Explorationsperspektive in der ursprünglichen Form. Auch ist es möglich, dass eine Karte an einer Stelle ihre Relevanz verliert und gelöscht wird, an anderer Stelle aber noch wichtige Verknüpfungen bestehen. Würden Inhalte unmittelbar propagiert, so gingen vorangegangene Inhalte verloren.
Das Problem, ob Zustände eingeführt werden sollten oder nicht, unterlag einer längeren Diskussion wie die Erwartung der Nutzerinnen und Nutzer in diesem Fall sein könnte. Diese Überlegungen hatten letztlich eine weit höhere Relevanz, als es auf den ersten Blick scheint, denn aus ihnen begründete sich letztlich die gesamte Pragmatik der Entwicklung: Es sollte ein Tool demonstriert werden, das die Diskussion unter verschiedenen Perspektiven motiviert!

Das bedeutet in Bezug auf die Karten, dass zwar durch die automatische Generierung eine Grundlage und Arbeitserleichterung für weitere Diskussionen erzielt werden sollte, sich aber die Workspace-Inhalte und damit auch die Karteninhalte in den unterschiedlichen Perspektiven „auseinander entwickeln“ können. Diese Idee wird später auch für die abgebildeten Strukturen aufrecht erhalten. Würden automatisch die Inhalte propagiert werden, wäre dieser Ansatz nicht möglich, da dadurch eine Kongruenz der Perspektiven implementiert würde, eine sehr strenge Gleichartigkeit der Inhalte auch in unterschiedlichen Kontexten!

Diese Übereinstimmung der Inhalte der unterschiedlichen Perspektiven soll aber gerade nicht angestrebt werden. Trotzdem soll die Abhängigkeit der Karteninhalte solange wie möglich zur Verfügung gestellt werden, um den Diskussionsfluss zu unterstützen und anzuregen.

Außerdem wurde davon ausgegangen, dass das sofortige Anzeigen von Änderungen die Nutzerinnen und Nutzer verwirren würde, da diese unmotiviert und als „Flimmern“ wahrgenommen würden. Warum sich was wo geändert hätte, wäre sehr untransparent.

Die Zustände sind ein Kompromiss, der einerseits erlaubt, Änderungen zu sehen aber andererseits ein lokales, örtlich beschränktes Arbeiten unter einer Perspektive ermöglicht. Änderungen, die gerade nicht relevant sind, müssen auch nicht sofort mit einbezogen werden. Mit diesen explizit dargestellten Zuständen ist es den Nutzerinnen und Nutzern möglich, zu entscheiden, ob Änderungen übernommen werden oder nicht. Solange die Änderungen übernommen werden, besteht die Abhängigkeit weiter. Wird die Änderung nicht übernommen, so wird die Abhängigkeit aufgelöst.

Daraus ergeben sich zwei Kartenzustände bzw. Markierungen wie sie in Abbildung 55 dargestellt sind:

1. Anzeigen, dass eine Karte aus dem Cluster gelöscht wurde.
2. Anzeigen, dass eine Karte aus dem Cluster modifiziert wurde.

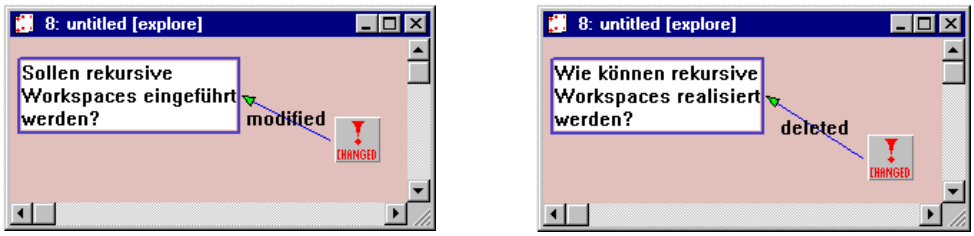


Abbildung 55

Markierung von Kartenzuständen.

Für die Darstellung der Zustände werden selbst wieder Karten verwendet. Um diese Zustände zu entfernen, werden in der Funktionspalette zwei Funktionskarten angeboten:

„Update“ und „Recover“. Mit der „Update“-Karte wird der neue Zustand übernommen, mit der „Recover“-Karte bleibt der alte Zustand bestehen und die Abhängigkeit wird aufgelöst. Die beiden Funktionskarten können jeweils in einen Workspace oder auf eine konkrete Zustandskarte gelegt werden. Dies verändert deren Aktionsrahmen, der sich entweder auf den gesamten Workspace oder nur auf eine konkrete Zustandsmarkierung bezieht.

An einem Beispiel wird in Abbildung 56 die interne Darstellung der Knotencluster den Oberflächenobjekten gegenübergestellt. Ausgangspunkt ist auf der linken Seite die Abbildung einer Karte aus der Protokollperspektive in die Explorationsperspektive. Beide Karten haben erst einmal den gleichen Inhalt (Diese Situation entspricht der, die schon in Abbildung 54 beschrieben wurde.). Darunter sind die Fakten aufgeführt, die diese Situation in der Datenbasis beschreiben. Dazu gehört zum einen das Cluster selbst, das unten dargestellt ist, und zum anderen die Referenzen der Oberflächenobjekte auf dieses Cluster.

Abbildung 56

Oberflächenobjekte und interne Darstellung zu einem Knotencluster vor und nach einer Modifizierung eines Inhaltes.

Auf der rechten Seite von Abbildung 56 wird nun das Resultat einer Änderung der Karte in der Protokollperspektive gezeigt. Die korrespondierende Karte im Explorations-Workspace behält vorerst den alten Inhalt. Allerdings wird die Zustandsänderung durch die „changed“-Markierung angezeigt. Diese Änderung in der Protokollperspektive führt bereits zur Änderung des Inhalts im Knotencluster. Der alte Inhalt ist nun nur noch über die Oberflächenrepräsentation der Karte im Explorations-Workspace zugänglich. Der entsprechende Knoten in der Knotenliste wird durch den Zustand *modification* gekennzeichnet.

Je nachdem, ob eine Löschung oder eine Modifizierung durchgeführt wurde und ob danach ein „Recover“ oder „Update“ angestoßen wird, ergeben sich diverse Fälle wie die Cluster jeweils gewartet werden müssen. Diese Fälle werden im Folgenden in der Form vorgestellt, dass der jeweilige Fall kurz beschrieben wird und danach, wie in Abbildung 56, die Fakten der Datenbasis aufgeführt werden wie sie vor und nach einer Aktivität aussehen.

Modifizieren von Inhalten

Fall 1: Zustandsmarkierungen nach Änderungen in einem Knotencluster

In diesem Fall sind alle Karten eines Clusters als Oberflächenobjekte angezeigt (Abbildung 57, linke Spalte. Darin symbolisieren die weißen Rechtecke Karten in den größeren farbig dargestellten Rechtecken, die für Workspaces stehen.). Es befinden sich Karten aus mehreren Perspektiven - hier drei - in einem Cluster. Vor der Aktion ist keine davon mit einer Zustandsmarkierung versehen. Bei einer Modifikation einer der Karten, hier von *Card*$_1$, ändern sich die Cluster-Daten.

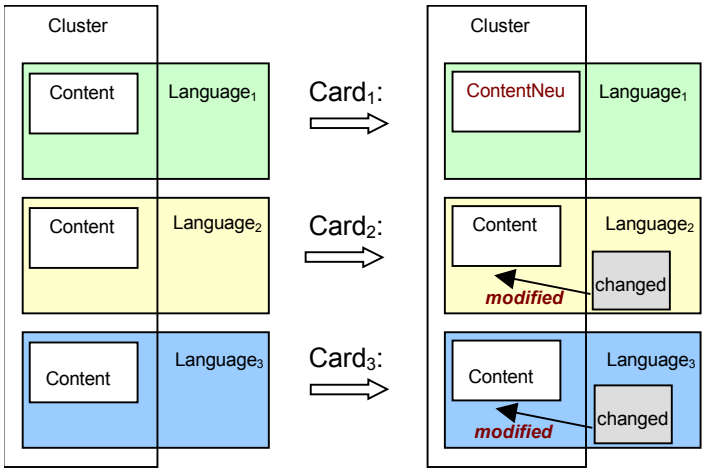


Abbildung 57

Zustandsmarkierungen nach Änderungen in einem Knotencluster.

map(Card$_1$, Language$_1$, Type$_1$, Cluster).		bleibt
map(Card$_2$, Language$_2$, Type$_2$, Cluster).		bleibt
map(Card$_3$, Language$_3$, Type$_3$, Cluster).		bleibt
cluster(Cluster, [(Language$_1$, Type$_1$, Topic$_1$, default), (Language$_2$, Type$_2$, Topic$_2$, default), (Language$_3$, Type$_3$, Topic$_3$, default)], Content).	→	cluster(Cluster, [(Language$_1$, Type$_1$, Topic$_1$, default), (Language$_2$, Type$_2$, Topic$_2$, modification), (Language$_3$, Type$_3$, Topic$_3$, modification)], ContentNeu).

Der neue Inhalt *ContentNeu* wird direkt in das Cluster übernommen. Die Informationen über den vorangegangenen Inhalt bestehen nur noch über die anderen unveränderten Oberflächenobjekte *Card*$_2$ und *Card*$_3$. Zu diesen werden außerdem die entsprechenden

Zustandsmarkierungen im Workspace hinzugefügt. Die folgende Gegenüberstellung zeigt Teile der Datenbasis vor und nach der Inhaltsveränderung einer Karte.

Fall 2: Weitergehende Inhaltsänderung

In Folge von Fall 1 kann die geänderte Karte, also $Card_1$, weiter geändert werden, ohne dass die angezeigten und intern repräsentierten Zustandsänderungen aufgelöst worden sind. Für die anderen angezeigten Karten, die bereits durch einen Modifizierungsmarker gekennzeichnet sind, ändert sich dabei nichts. Sie behalten weiter ihren alten Inhalt. Im Cluster ändert sich jedoch die Inhaltsinformation. In diesem Fall wurde die Entscheidung getroffen, die Zwischenstadien der Inhalte nicht mit zu verwalten. $Content_1$ steht nach einer so durchgeführten Änderung nicht mehr zur Verfügung.

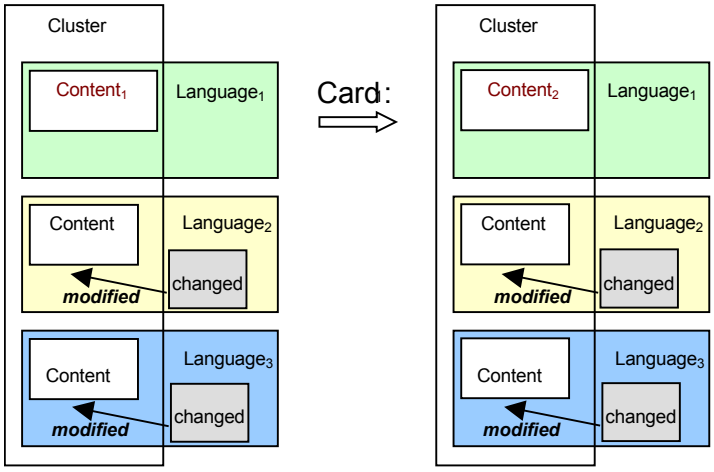


Abbildung 58

Weitergehende Inhaltsänderung.

map(Card$_1$, Language$_1$, Type$_1$, Cluster).		bleibt
map(Card$_2$, Language$_2$, Type$_2$, Cluster).		bleibt
map(Card$_3$, Language$_3$, Type$_3$, Cluster).		bleibt
cluster(Cluster, [(Language$_1$, Type$_1$, Topic$_1$, default), (Language$_2$, Type$_2$, Topic$_2$, modification), (Language$_3$, Type$_3$, Topic$_3$, modification)], Content$_1$).	→	**cluster**(Cluster, [(Language$_1$, Type$_1$, Topic$_1$, default), (Language$_2$, Type$_2$, Topic$_2$, modification), (Language$_3$, Type$_3$, Topic$_3$, modification)], Content$_2$).

Fall 3: Veränderung einer Karte, die schon eine Zustandsmarkierung trägt

In Folge von Fall 1 kann es außerdem dazu kommen, dass eine Karte, die die Zustandsmarkierung *modification* trägt, selbst modifiziert wird, ohne vorher auf die Markierung einzugehen, d.h. sie zu akzeptieren oder abzulehnen. Dies entspricht zwar nicht der intendierten Nutzung, kann aber auch nicht ausgeschlossen werden.

Es wird davon ausgegangen, dass in diesem Fall die angebotene Modifikation keine Rolle mehr spielt. Das bedeutet, dass die neue Änderung den aktuellen Stand anzeigt, die

Zustandsmarkierung gelöscht wird und der modifizierte Knoten aus dem Cluster herausgenommen wird. Im Beispiel wird *Card*$_3$ geändert.

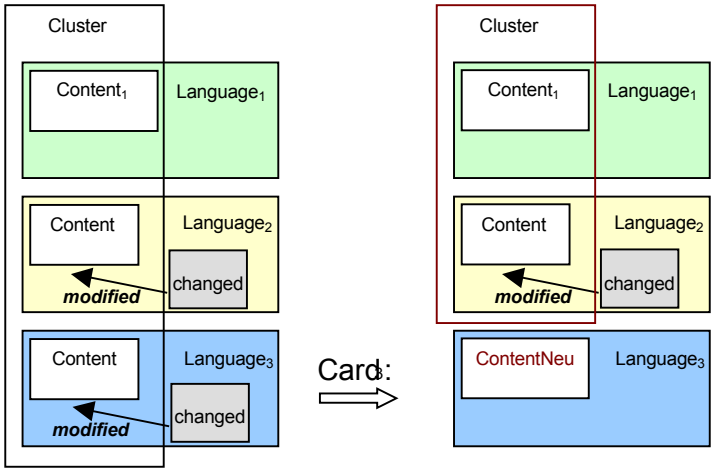


Abbildung 59

Veränderung einer Karte, die schon eine Zustandsmarkierung trägt.

map(Card$_1$, Language$_1$, Type$_1$, Cluster).		bleibt
map(Card$_2$, Language$_2$, Type$_2$, Cluster).		bleibt
map(Card$_3$, Language$_3$, Type$_3$, Cluster).	→	entfernt
cluster(Cluster, [(Language$_1$, Type$_1$, Topic$_1$, default), (Language$_2$, Type$_2$, Topic$_2$, modification), (Language$_3$, Type$_3$, Topic$_3$, modification)], Content$_1$).	→	cluster(Cluster, [(Language$_1$, Type$_1$, Topic$_1$, default), (Language$_2$, Type$_2$, Topic$_2$, modification)] Content$_1$).

Fall 4: Änderung einer Karte, ohne dass alle Karten des Cluster angezeigt werden

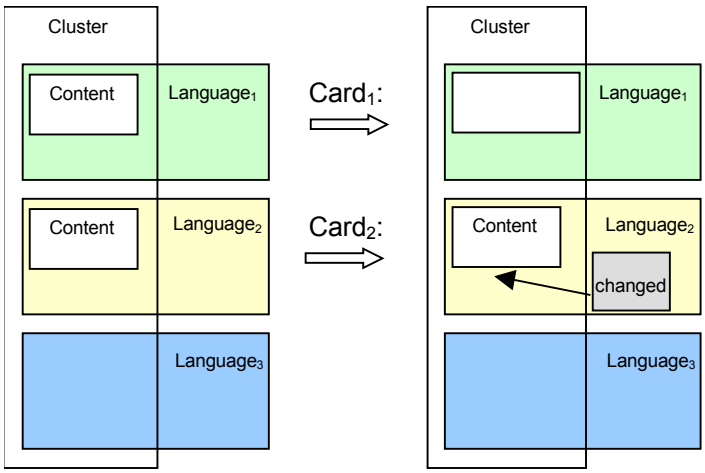


Abbildung 60

Änderung einer Karte, ohne dass alle Karten des Cluster angezeigt werden.

Es ist möglich, dass nicht alle Karten eines Clusters als Oberflächenobjekte angezeigt sind. Für die angezeigten Objekte wird verfahren wie oben beschrieben. Bei den nicht

angezeigten werden keine Zustände außer des *default* eingetragen. Dies ist einerseits eine Designentscheidung, bei der davon ausgegangen wurde, dass es verwirrt, wenn beim erstmaligen Anzeigen einer Karte gleich eine Zustandmarkierung mit erscheinen würde. Aber auch programmiertechnisch harmonierte das nicht mit dem gewählten Ansatz, in dem ja alte Inhalte nur als Inhalte der dargestellten Oberflächenobjekten vorhanden sind. Würde man so vorgehen, müsste eigentlich die gesamte Historie eines Oberflächenobjektes verwaltet werden, um sie Nutzerinnen und Nutzern zur Auswahl zu stellen. Im Beispiel ist $Card_3$ nicht angezeigt, $Card_1$ wird verändert.

map(Card$_1$, Language$_1$, Type$_1$, Cluster).		bleibt
map(Card$_2$, Language$_2$, Type$_2$, Cluster).		bleibt
cluster(Cluster, [(Language$_1$, Type$_1$, Topic$_1$, default), (Language$_2$, Type$_2$, Topic$_2$, default), (Language$_3$, Type$_3$, Topic$_3$, default)], Content).	→	**cluster**(Cluster, [(Language$_1$, Type$_1$, Topic$_1$, default), (Language$_2$, Type$_2$, Topic$_2$, modification), (Language$_3$, Type$_3$, Topic$_3$, default)], ContentNeu).

Löschen von Karten

Fall 5: Löschen einer Karte aus einem Knotencluster

So, wie Karten, die in Clustern verwaltet werden, modifiziert werden können, können sie auch gelöscht werden. Im folgenden Beispiel sind drei Karten eines Clusters angezeigt. $Card_1$ wird gelöscht. Im Cluster wird daraufhin die Löschinformation aufgenommen. Die anderen dazugehörigen angezeigten Oberflächenobjekte werden durch den entsprechenden Zustand markiert. Der Cluster-Eintrag für $Card_1$ wird aus der Liste gelöscht und die Information *map(...)* wird aus der Datenbasis gelöscht.

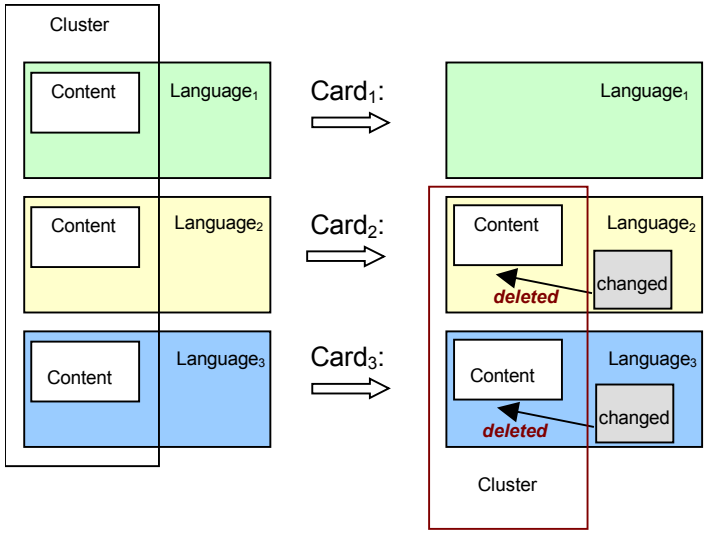


Abbildung 61

Löschen einer Karte aus einem Knotencluster.

map(Card$_1$, Language$_1$, Type$_1$, Cluster).	→	entfernt
map(Card$_2$, Language$_2$, Type$_2$, Cluster).		bleibt
map(Card$_3$, Language$_3$, Type$_3$, Cluster).		bleibt
cluster(Cluster, [(Language$_1$, Type$_1$, Topic$_1$, default), (Language$_2$, Type$_2$, Topic$_2$, default), (Language$_3$, Type$_3$, Topic$_3$, default)], Content).	→	cluster(Cluster, [(Language$_2$, Type$_2$, Topic$_2$, deletion), (Language$_3$, Type$_3$, Topic$_3$, deletion)], Content).

Fall 6: Löschen einer Karte, wenn nicht alle Karten angezeigt sind

Ist eine Karte im Cluster enthalten aber nicht angezeigt, so wird beim Löschen einer anderen Karte aus dem Cluster intern keine Löschinformation gespeichert. In diesem Beispiel ist *Card$_3$* nicht angezeigt, *Card$_1$* wird als Oberflächenobjekt gelöscht und daraufhin die dazugehörige Cluster-Information. Da *Card$_2$* angezeigt ist, erhält es auch die „Deletion"-Information im Cluster. *Card$_3$* ist jedoch nicht angezeigt und behält die „Default"-Information.

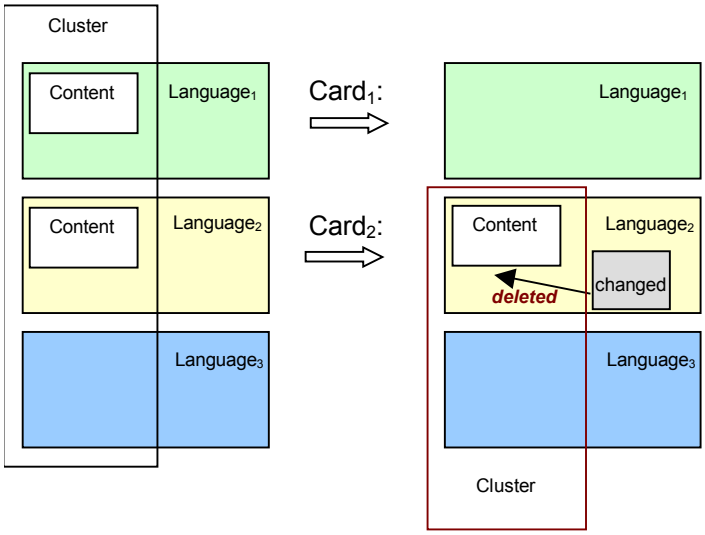


Abbildung 62

Löschen einer Karte, wenn nicht alle Karten angezeigt sind.

map(Card$_1$, Language$_1$, Type$_1$, Cluster).	→	entfernt
map(Card$_2$, Language$_2$, Type$_2$, Cluster).		bleibt
cluster(Cluster, [(Language$_1$, Type$_1$, Topic$_1$, default), (Language$_2$, Type$_2$, Topic$_2$, default), (Language$_3$, Type$_3$, Topic$_3$, default)], Content).	→	cluster(Cluster, [(Language$_2$, Type$_2$, Topic$_2$, deletion), (Language$_3$, Type$_3$, Topic$_3$, default)], Content).

Auflösen von Zustandsmarkierungen

Um Markierungskarten wieder zu entfernen, sind zwei Funktionskarten aus der Funktionspalette vorgesehen: die „Update"-Karte und die „Recover"-Karte. Die folgenden Fälle beschreiben deren Wirkung auf die Zustandsmarkierungen. Alle

Aktionen, die durch die Funktionskarten ausgelöst werden, beziehen sich ausschließlich auf den Workspace, in den sie gelegt wurden, bzw. auf die Zustandsmarkierung, auf die die Karte gelegt wurde. Außerdem beschränken sie sich immer auf die angezeigten Karten.

Fall 7: „Update“ einer „Deleted“-Markierung

Ein „Update“ hebt eine „Deletion“-Information so auf, dass die markierte Karte zusammen mit der Markierung gelöscht wird. Nach der Aktualisierung werden die Daten auch aus dem Cluster gelöscht. In diesem Beispiel wird das „Update“ im Workspace von *Card*$_2$ ausgeführt. Vorher war eine dritte Karte des Clusters, *Card*$_1$, gelöscht worden.

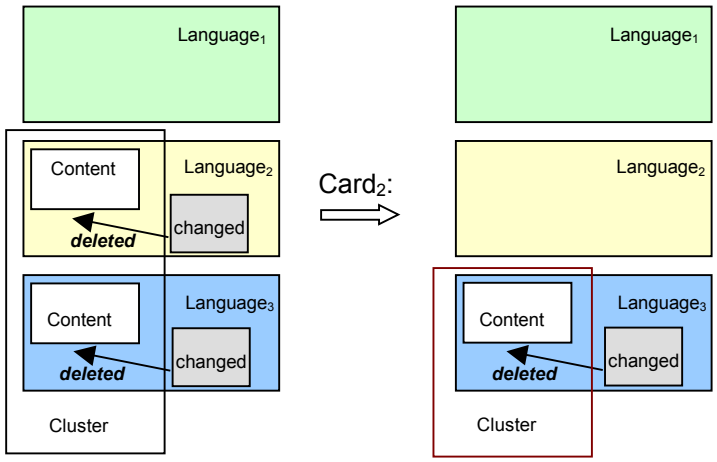


Abbildung 63

Bestätigen einer „Deleted“-Markierung.

map(Card$_2$, Language$_2$, Type$_2$, Cluster).	→	entfernt
map(Card$_3$, Language$_3$, Type$_3$, Cluster).		bleibt
cluster(Cluster, [(Language$_2$, Type$_2$, Topic$_2$, deletion), (Language$_3$, Type$_3$, Topic$_3$, deletion)], Content).	→	**cluster**(Cluster, [(Language$_3$, Type$_3$, Topic$_3$, deletion)], Content).

Fall 8: „Update“ einer Modifizierung

Ein „Update“ hebt eine „Modification“-Information so auf, dass die Markierung gelöscht wird und der Inhalt der Karte an die im Cluster gespeicherte Information angepasst wird. Im Beispiel war *Card*$_1$ modifiziert worden. Das „Update“ wird nun im Workspace von *Card*$_2$ durchgeführt.

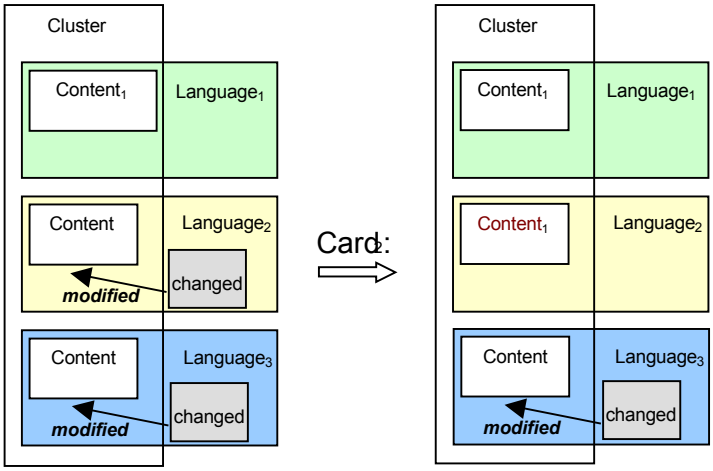


Abbildung 64

Bestätigen einer Modifizierung.

map(Card$_1$, Language$_1$, Type$_1$, Cluster).	bleibt
map(Card$_2$, Language$_2$, Type$_2$, Cluster).	bleibt
map(Card$_3$, Language$_3$, Type$_3$, Cluster).	bleibt
cluster(Cluster, [(Language$_1$, Type$_1$, Topic$_1$, default), (Language$_2$, Type$_2$, Topic$_2$, modification), (Language$_3$, Type$_3$, Topic$_3$, modification)], Content$_1$).	→ **cluster**(Cluster, [(Language$_1$, Type$_1$, Topic$_1$, default), (Language$_2$, Type$_2$, Topic$_2$, default), (Language$_3$, Type$_3$, Topic$_3$, modification)], Content$_1$).

Fall 9: „Recover“ einer „Deleted“-Markierung

Ein „Recover“ hebt eine „Deletion“-Information so auf, dass die Markierungskarte gelöscht wird, die Karte mit dem alten Inhalt bestehen bleibt und aus dem Cluster herausfällt. In der dargestellten Situation war *Card$_1$* gelöscht worden. Für den Workspace von *Card$_2$* wird ein „Recover“ ausgeführt.

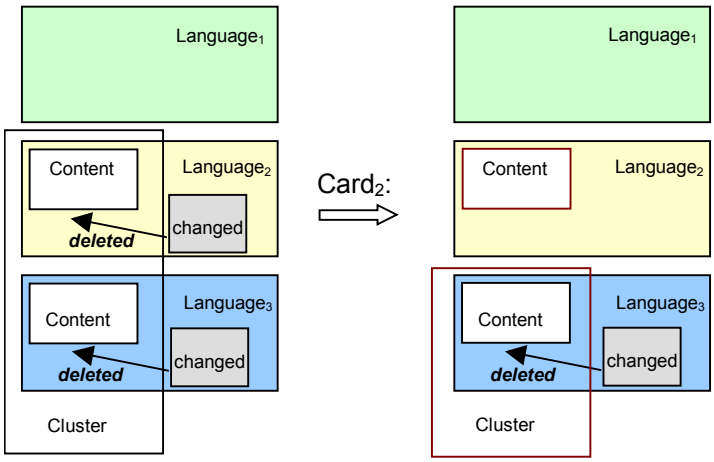


Abbildung 65

„Recover” einer Karte mit einer „Deleted“-Markierung.

Da die *map(...)*-Information nur für Cluster-Karten gespeichert wird, wird sie für $Card_2$ als Resultat ebenfalls gelöscht. Da $Card_2$ durch das „Recover“ aus dem Cluster herausgenommen wird, verändert sich auch das Cluster. Das resultierende Cluster unterscheidet sich nicht von dem, das in Fall 7 entstanden ist. Der Unterschied besteht nur darin, dass die Karte $Card_2$ bestehen bleibt, allerdings ohne weiter zu diesem Cluster zu gehören.

map(Card$_2$, Language$_2$, Type$_2$, Cluster).	→	entfernt
map(Card$_3$, Language$_3$, Type$_3$, Cluster).		bleibt
cluster(Cluster, [(Language$_2$, Type$_2$, Topic$_2$, deletion), (Language$_3$, Type$_3$, Topic$_3$, deletion)], Content).	→	**cluster**(Cluster, [(Language$_3$, Type$_3$, Topic$_3$, deletion)], Content).

Fall 10: „Recover“ einer „Modified“-Markierung

Ein „Recover“ hebt eine „Modification“-Information so auf, dass die Markierung gelöscht wird, und die Karte nicht länger zum Cluster gehört. Wie im vorangegangenen Fall wird entsprechend auch die *map(...)*-Information gelöscht, auch wenn die Karte als Oberflächenobjekt weiter bestehen bleibt. In diesem Beispiel wird das „Recover“ auf den Workspace von $Card_2$ ausgeführt. Vorher war $Card_1$ modifiziert worden.

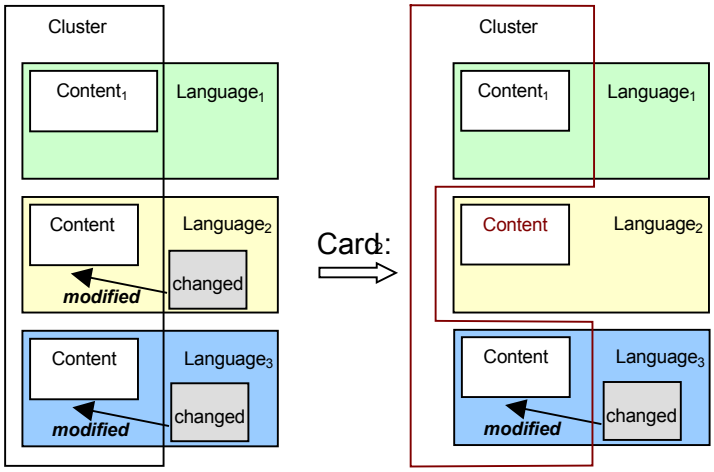


Abbildung 66

„Recover” einer Karte mit einer „Modified“-Markierung.

map(Card$_1$, Language$_1$, Type$_1$, Cluster).		bleibt
map(Card$_2$, Language$_2$, Type$_2$, Cluster).	→	entfernt
map(Card$_3$, Language$_3$, Type$_3$, Cluster).		bleibt
cluster(Cluster, [(Language$_1$, Type$_1$, Topic$_1$, default), (Language$_2$, Type$_2$, Topic$_2$, modification), (Language$_3$, Type$_3$, Topic$_3$, modification)], Content$_1$).	→	**cluster**(Cluster, [(Language$_1$, Type$_1$, Topic$_1$, default), (Language$_3$, Type$_3$, Topic$_3$, modification)], Content$_1$).

Fall 11:

Da die Markierungskarten für die Zustände selbst Karten sind, können diese auch direkt von den Nutzerinnen und Nutzern gelöscht werden. Entsprechend muss auch für diese Löschaktion intern eine Interpretation vorhanden sein.

- Das Löschen eines „Deleted"-Zustandes wird intern als „Recover" auf eine Karte interpretiert, die eine „Deleted"-Markierung hat. Das entspricht Fall 9.
- Das Löschen eines „Modified"-Zustandes wird ebenfalls als „Recover" interpretiert. Das entspricht Fall 10.

Schließen von Workspaces

Fall 12:

Wird ein Workspace geschlossen, in dem Zustandsmarkierungen vorlagen, so werden sie bisher nach einem erneuten Öffnen wieder mit hergestellt. Im Falle einer „Deletion"-Markierung kann das sinnvoll durchgeführt werden, da der Inhalt, der im Cluster verwaltet wird, auch den relevanten Stand des Karteninhalts reflektiert. Er entspricht dem ursprünglichen Inhalt bevor eine korrespondierende Karte aus dem Cluster gelöscht wurde.
Im Falle einer „Modification"-Markierung geht jedoch Information verloren, denn auf den ursprüngliche Inhalt vor der Änderung konnte nur über das Oberflächenobjekt selbst zugegriffen werden, das aber durch das Schließen des Workspaces nicht mehr besteht, denn in der Cluster-Beschreibung liegt schon der modifizierte Zustand vor. Eine verbesserte Strategie ist dafür noch nicht implementiert. Es kommt jedoch zu keinen echten Inkonsistenzen. Durch die Funktionskarten „Recover" und „Update" kann die generierte Markierung aufgehoben werden. In einem Fall fällt die Inhaltskarte aus dem Cluster, im anderen bleibt sie darin bestehen.
Eine Lösung dafür würde in eine allgemeinere Realisierung persistenter Daten für die Diskussionsunterstützung fallen.

6.7 Abbildung von Strukturen

Wie vorangegangen für einzelne Karten beschrieben, so ist es auch möglich, mehrere Karten, die miteinander in Relation stehen, gleichzeitig abzubilden. Diese Zusammenhänge werden Strukturen genannt. Es soll weitere Diskussionen zu bestimmten Aspekten anregen. Durch die Abbildung von Strukturen ergibt sich jedoch die zusätzliche Möglichkeit, diese räumlich zu gestalten, sie zu sortieren, und eine definierte Auswahl

von Strukturen zu treffen. Das heißt, dass Daten aus einem Kontext herausgefiltert werden, zusammengefasst werden können und potentiell bei der Abbildung verändert werden. Diese Veränderung wird als Interpretation gewertet.
In den folgenden Unterkapiteln werden die Strukturen detailliert eingeführt. Dazu gehört, wie sie sich zusammensetzen, wie sie erkannt werden und wie der Abbildungsprozess realisiert ist. Die konkreten Abbildungsvorschriften werden erst im nächsten Kapitel, das das Diskussionsmodell beischreibt, erläutert.

6.7.1 Strukturen

Ausdrücke in den hier verwendeten visuellen Sprachen sind Graphen, in denen die Inhaltskarten die Knoten darstellen. Eine Graph-Kante ist die Verbindung zwischen zwei Inhaltskarten, die aus einer Konnektorkarte und zwei Links besteht.

> Eine **Struktur** ist eine Menge von Inhaltskarten und Konnektorkarten, in der keine „unverbundenen“ Karten enthalten sind. Sie ist ein Subgraph eines Ausdruckes in einer visuellen Sprache, jedoch kein induzierter Graph, da nicht alle Kanten zwischen den durch die Struktur bestimmten Inhaltskarten auch Teil der Struktur sein müssen.

Die Implementierung der Strukturabbildungen setzt Multigraphen voraus. Das heißt, dass keine zwei gleichen Kanten zwischen zwei Inhaltskarten bestehen dürfen, wohl aber mehrere unterschiedliche Kanten.
Die Abbildung einer Struktur ist mehrstufig.

- Zuerst einmal muss erkannt werden, dass eine gültige Struktur entstanden ist.
- Für die Beitragskarten, die in dieser Struktur enthalten sind, werden Knotencluster generiert, sofern dafür nicht schon welche bestehen.
- Für die Struktur wird ein Modell angelegt, über das die Typen der Relationen verwaltet werden, die zu einer Struktur gehören und zwar bezogen auf die jeweilige Perspektive, sowie zwischen welchen Knotenclustern sie sich befinden.

Die Strukturen, die für die Diskussionsunterstützung definiert sind, bestehen nicht aus einer festen Anzahl von Knoten und Kanten. Sie wachsen oder schrumpfen während der Diskussion. Allein Knoten- und Kantentypen mit ihrer Konstellation zueinander machen eine Struktur aus. Um diesen Sachverhalt implementieren zu können, werden Strukturen über ihre kleinsten sinntragenden Teilstrukturen definiert und darüber wie diese zueinander positioniert sein müssen. Außerdem wird festgelegt, welche die minimale Ausprägung einer Struktur ist.

Die minimale Ausprägung einer Struktur wird als **Strukturkern** bezeichnet.

Extensionen sind Teilstrukturen, die ohne den Kern keine gültige Struktur darstellen.

Eine Struktur besitzt dementsprechend kein eindeutiges Muster, sondern kann als Bildungsvorschrift verstanden werden. Strukturen sind so festgelegt, dass sie aus einem Kern bestehen und möglichen Erweiterungen, den Extensionen.

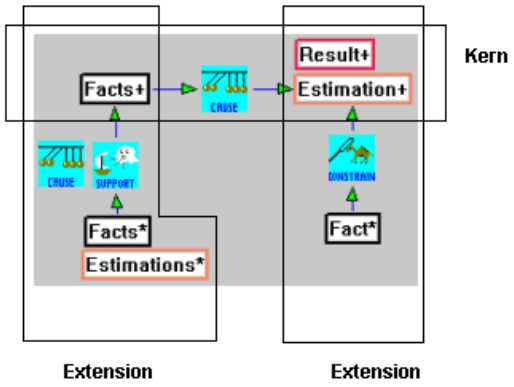


Abbildung 67

Die Teilstrukturen eines Arguments.

Beispielsweise besteht die minimale Struktur eines Arguments, wie es für die Diskussionsunterstützung modelliert ist, in dem Begründungszusammenhang einer Bedingung und dem Fazit, das daraus gezogen wird. Dies ist in Abbildung 67 zu sehen, die den Kern der Struktur darstellt, der aus einem Faktum, einer „Cause"-Relation und einem Resultat („Result") oder einer Einschätzung („Estimation") besteht. Die anderen Bestandteile, die Extensionen, sind optional und können als Teilstrukturen in beliebiger Anzahl vorkommen.

Um die Komplexität für die Strukturerkennung einzuschränken, wurde die Anzahl der Links für diese Anwendung durchgängig auf zwei Stück pro Konnektorkarte eingeschränkt. Dies ist jedoch bei der Verwendung nicht unbedingt als Nachteil anzusehen, da wie bereits geschildert, schon in der Vorstudie deutlich wurde, dass die Lesbarkeit von Hypergraphen mit mehrstelligen Relationen nicht sehr gut war.

Im Diskussionsmodell werden die Strukturen definiert. „Constraint"-artige Prolog-Regeln legen die jeweiligen Kerne und Extensionen fest. Im Vorgriff auf das Kapitel 6.8 zum Diskussionsmodell wird hier kurz das Schema dieser „Constraints" erläutert.

Die Strukturkerne für die Diskussionsunterstützung bestehen alle nur aus einer Relation. Dies ist eine Beschränkung, die von der konkreten Anwendung ausgeht, nicht aber vom implementierten Abbildungsvorgang. Eine Definition eines Kerns gibt für eine gegebene

visuelle Sprache und einen konkreten Strukturtyp, wie z.B. *argument*, vor, welche Bedingungen für die konkreten Oberflächenobjekte gelten müssen.

```
core(language, type, – View) iff condition(– View)
```

Dafür wird ein Ausschnitt des Zustands eines Workspace überprüft, ob dieser die angegebenen Bedingungen erfüllt. Diese Zustandsbeschreibung beschreibt Relationen, die wieder aus drei Informationen kombiniert sind: dem Typ einer Konnektorkarte und zwei mit ihr verbundenen Inhaltskarten. Jeweils eine Relation ist im Term *view_rel* beschrieben, der außerdem eine Richtung der Relation ausdrückt, die durch die Oberflächenobjekte vorgegeben ist: Eine Graphkante verweist von der *Condition* auf die *Conclusion*:

```
core(language, type, [ view_rel(Connector, Condition, Conclusion)]):-
        has_one( relation(types1, Connector,
                          types2, Condition,
                          types3, Conclusion)).
```

Das *has_one*-Prädikat definiert die Typen der Karten von Teilstrukturen und ebenfalls die Richtung der Relationen zwischen diesen Karten. $types_{1\text{-}3}$ steht für Typlisten. Deren Festlegung ist wesentlicher Bestandteil des Diskussionsmodells.
Mit Hilfe des *has_one*-Prädikates, im Zusammenspiel mit den Instanzierungen des Terms *view_rel*, kann überprüft werden, ob eine Relation, die gerade im DiscBoard hergestellt wurde, zur Entstehung eines neuen Strukturkernes geführt hat. Das folgende Beispiel stellt die Definition eines Argumentkernes (*argument*) in der Explorationsperspektive (*explore*) vor.

```
core(explore, argument, [view_rel(Connector, Condition, Conclusion)]):-
        has_one( relation([cause], Connector,
                          [fact], Condition,
                          [result, estimation], Conclusion)).
```

Die Erweiterungsdefinitionen der Kernstrukturen werden durch das Prädikat *extension* definiert. Eine Extension ist erst dann Teil einer Struktur, wenn der Kern dazu bereits besteht. Entsprechend wird in *extension* sowohl die Extension selbst definiert, wie es schon für den Kern durchgeführt wurde (*has_one*), als auch die Konstellation zwischen Extension und Kern vorgegeben.

```
extension(language, type, + Structure, view_rel(– Connector1, ? Condition1, ? Conclusion1)):-
        relation(– Connector1, _, ? Condition1, ? Conclusion1),
        has_one( relation(types1, – Connector1,
                          types2, – Condition1,
                          types3, – Conclusion1)),
        core(language, type, [view_rel(+ Connector2, + Condition2, – Conclusion1)]),
        \==(– Condition1, – Condition2),
        in_structure(– Conclusion1, language, type, + Structure).
```

Beim Aufruf von *extension* sind im Term *view_rel* immer nur die Konnektor-Id (*Connector*$_1$) und eines der beiden anderen Argumente instanziert. Durch das Prädikat *relation* werden die freien Variablen so belegt, dass Bedingung und Konklusion korrekt zugeordnet werden. Nach dem Typ-Check in *has_one* wird definiert, für welchen Kern die Extension gilt. Übereinstimmen muss die Perspektive (*language*) und der Strukturtyp (*type*). Im dargestellten Fall wird vorgegeben, dass die Konklusion des Kerns mit der der Extension übereinstimmen muss. In *core* wird dann ermittelt, ob zu dieser als syntaktisch korrekt erkannten Extension bereits ein Kern vorliegt. Durch die Bedingung der Ungleichheit der zwei Vorbedingungen wird ausgeschlossen, dass für die Extension und den Kern die gleiche Relation ermittelt wird.
Dass die Extension und der Kern im gleichen Workspace vorliegen, wird indirekt dadurch sichergestellt, dass die *Condition*- oder *Conclusion*-Variablen von Kern und Extension zumindest an einem Punkt übereinstimmen. Dies muss immer möglich sein, da sonst kein zusammenhängender Graph vorläge. Die letzte Anfrage (*in_structure*) ermittelt die Struktur-Id für die Extension. Auch hier ist kurz ein Beispiel für die Argumentstruktur zu nennen:

```
extension(explore, argument, Structure, view_rel(Connector1, Condition1, Conclusion1)):-
        relation(Connector1, _, Condition1, Conclusion1),
        has_one( relation([support], Connector1,
                          [fact, estimation], Condition1,
                          [fact], Conclusion1)),
        core(explore, argument, [view_rel(_, Conclusion1, _)]),
        \==(Condition1, Conclusion1),
        in_structure(Conclusion1, explore, argument, Structure).
```

In diesem Beispiel ist eine Extension definiert, die die Vorbedingung eines Argumentkernes erweitert. Die Konklusion der Extension und die Bedingung für den Kern müssen übereinstimmen.

Für diese Art und Weise der Strukturdefinitionen werden im Folgenden verschiedene Beispiele angegeben, die allerdings keinen Bezug zum konkreten Diskussionsmodell nehmen. Es werden allgemeine Möglichkeiten für Strukturbildungen vorgestellt. In den Beispielen stellen die Quadrate Inhaltskarten dar, die Kreise Konnektorkarten.

Beispiel 1 in Abbildung 68 stellt das Definitionsschema für eine Struktur dar, in der die Extension an die Bedingung eines Kerns gebunden ist. Außerdem weist die Relation vom Kern weg. Diese Struktur wird ausschließlich durch die Position der Parameter *D*, *A*, *E* bzw. *B*, *A*, *C* (für dieses Beispiel) bestimmt.

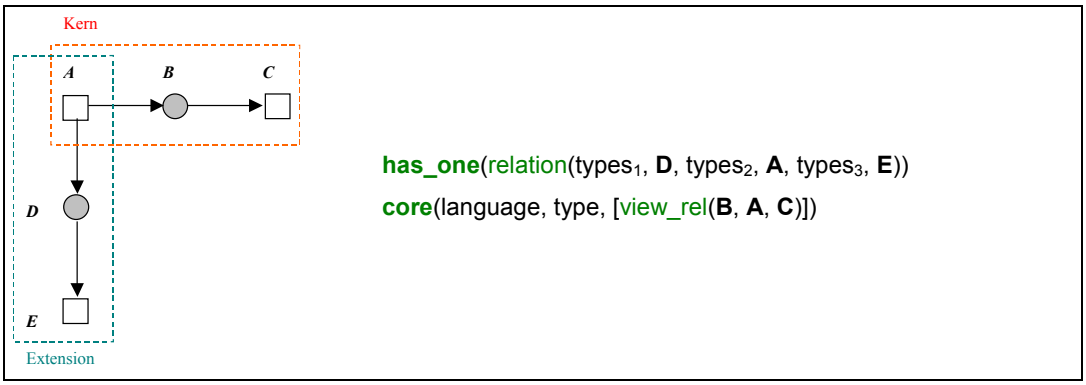


Abbildung 68

Strukturdefinition, Beispiel 1.

Beispiel 2 in Abbildung 69 stellt ein ähnlichen Fall vor, nur dass die Extension an die Konklusion des Kerns gebunden ist. Im Vergleich zwischen Abbildung 69 und Abbildung 70 kann gut erkannt werden, wie die Vertauschung der Variablen *E* und *C* die Richtung der Extensionsrelation verändert.

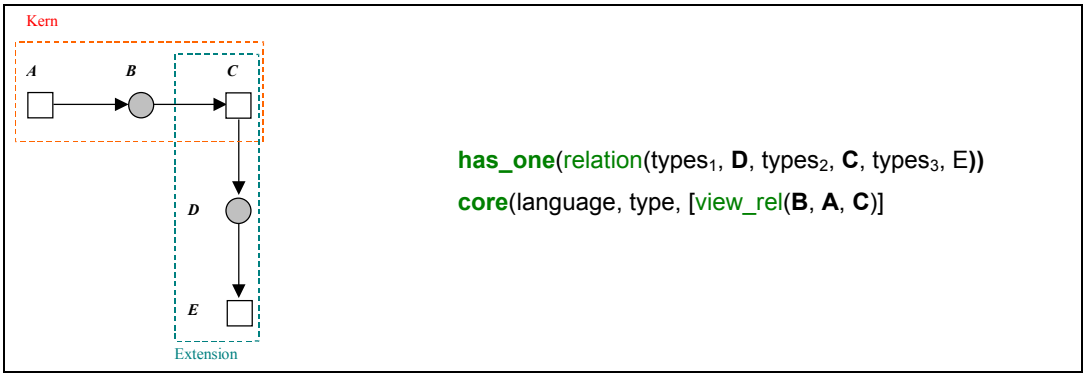


Abbildung 69

Strukturdefinition, Beispiel 2.

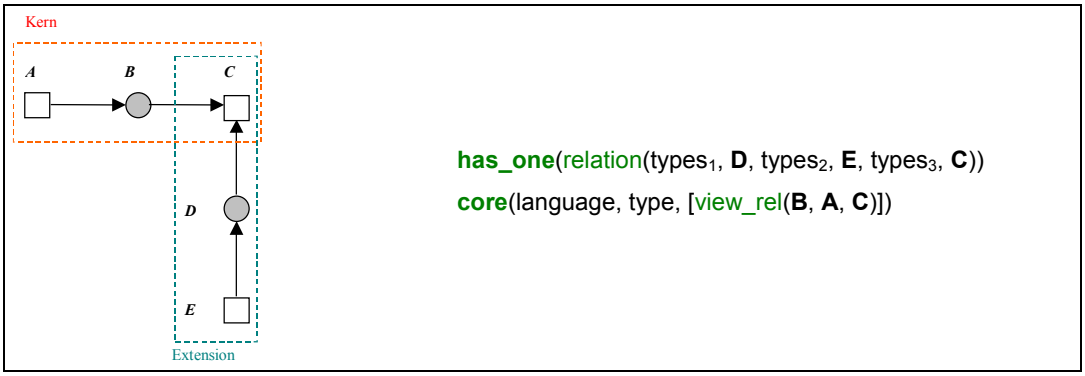


Abbildung 70

Strukturdefinition, Beispiel 3.

Abbildung 71 zeigt eine Struktur mit einem umfangreicheren Kern als Teilstruktur. Für diesen wird deshalb eine Liste von Relationen angegeben.

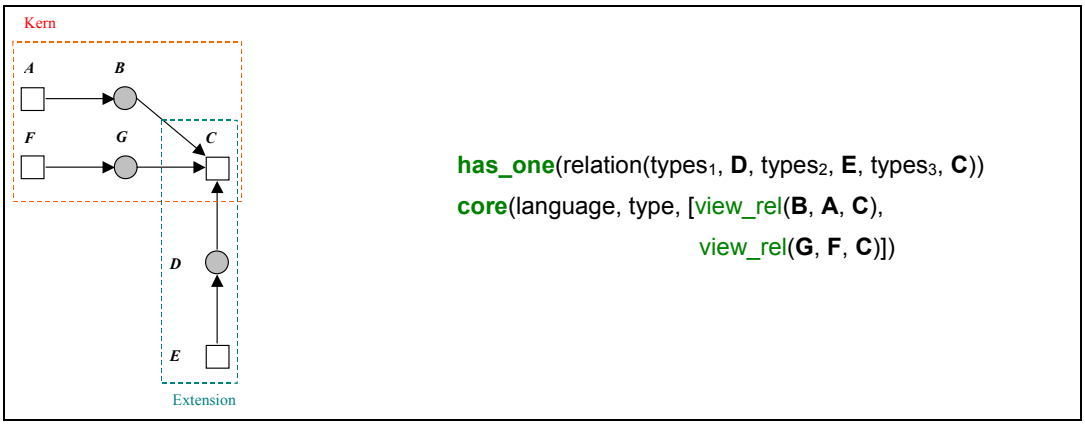


Abbildung 71

Strukturdefinition, Beispiel 4.

6.7.2 Das Strukturmodell

Wie für die Karten die Knotencluster implementiert worden sind, wird bei Abbildungen für die Strukturen ein Strukturmodell aufgebaut, das die Zugehörigkeit der Strukturen zwischen den Perspektiven verwaltet. Der Datenbasiseintrag beinhaltet die Informationen über das Thema, zu dem die Struktur generiert wurde, den Typ der generierten Struktur, den Struktur-Identifikator und das eigentliche Modell. Aus dieser Repräsentation kann die visuelle Darstellung generiert werden.

structure(**?** Topic, **?** Type, **?** Structure, **?** Model).

Das Modell besteht aus einer Liste von Tupeln, die für jede Perspektive die Graphstrukturen verwaltet.

model(**+** [(Language$_1$, Relations$_1$) | Perspectives])

Da Strukturen von einer Perspektive in die andere abgebildet werden, darf jede Sprache nur in einem der Tupel eines Strukturmodells genannt werden. Jede im Modell festgehaltene Graph-Kante, die im Term *model_rel* zusammengefasst ist, besteht wieder aus drei Teilen, dem Typ der Konnektorkarte und zwei Ids für die verbundenen Knotencluster.

$$\textbf{model_rel}(+\ \text{Type}, +\ \text{Cluster}_1, +\ \text{Cluster}_2)$$

In den Strukturmodellen kann nicht auf die konkreten Oberflächenobjekte verwiesen werden, da nicht gewährleistet ist, dass diese immer gerade angezeigt sind. So werden die Relationen als Verbindungen zwischen Knotenclustern definiert. Das bedeutet, dass jede Karte, die sich in einer Struktur befindet, sich außerdem in einem Knotencluster befinden muss. Auch im Term model_rel ist wieder eine Richtung der beschriebenen Kante kodiert, die wie vorher

Kodiert ist außerdem eine Richtung der Kanten. Sie verläuft immer von der ersten Cluster-Nennung in Richtung der zweiten. In detaillierter Form stellt sich die Modell-information wie folgt dar:

$$\begin{aligned}&\textbf{model}(\ [\\ &\quad (\text{Language}_1,\ \ [\text{model_rel}(\text{Type}_{11}, \text{Cluster}_1, \text{Cluster}_2) \mid \text{Relations}]\ ),\\ &\qquad\qquad\qquad\qquad :\\ &\quad (\text{Language}_m,\ \ [\text{model_rel}(\text{Type}_{m1}, \text{Cluster}_{m1}, \text{Cluster}_{m2}) \mid \text{Relations}]\ )\\ &\ ]\ )\end{aligned}$$

6.7.3 Strukturüberprüfung

Strukturen werden zur Erstellungszeit erkannt und abgebildet, nicht erst dann, wenn eine Perspektive angefordert wird. Dadurch reduziert sich die Zeit, um einen Workspace zu einem bestimmten „Topic“ aufzubauen, wenn er benötigt wird. Wichtiger jedoch ist, dass sich der Aufwand reduziert, um die Strukturen zu identifizieren. Das liegt daran, dass eine neue Struktur oder eine Erweiterung einer bestehenden immer nur dann entstehen kann, wenn ein neuer Link als Oberflächenobjekt hergestellt wird. Dadurch kann von diesem ausgehend eine lokale Prüfung im Graphen durchgeführt werden. Müssten die Strukturen unabhängig von dieser Information, wo ein Link neu entstanden ist, erkannt werden, müsste eine globale Prüfung nicht nur eines Workspaces stattfinden, sondern aller Perspektiven, die zu dem aktuellen „Topic“ gehören, um alle relevanten Strukturen zu erkennen. Das Prädikat *structure_test* implementiert die Strukturüberprüfung.

```
structure_test(– Topic, – Language, – Connector, – Card):-
        relation(Connector, Card, + Condition, + Conclusion),
        extension_test(Topic, Language, Connector, Condition, Conclusion),
        core_test(Topic, Language, Connector, Condition, Conclusion).
```

structure_test bezieht sich auf eine konkrete Perspektive (*Language*) und einen bestimmten „Topic“. Der Ausgangspunkt der Strukturüberprüfung, ein aktuell erstellter Link, drückt sich in den Parametern *Connector* und *Card* aus.

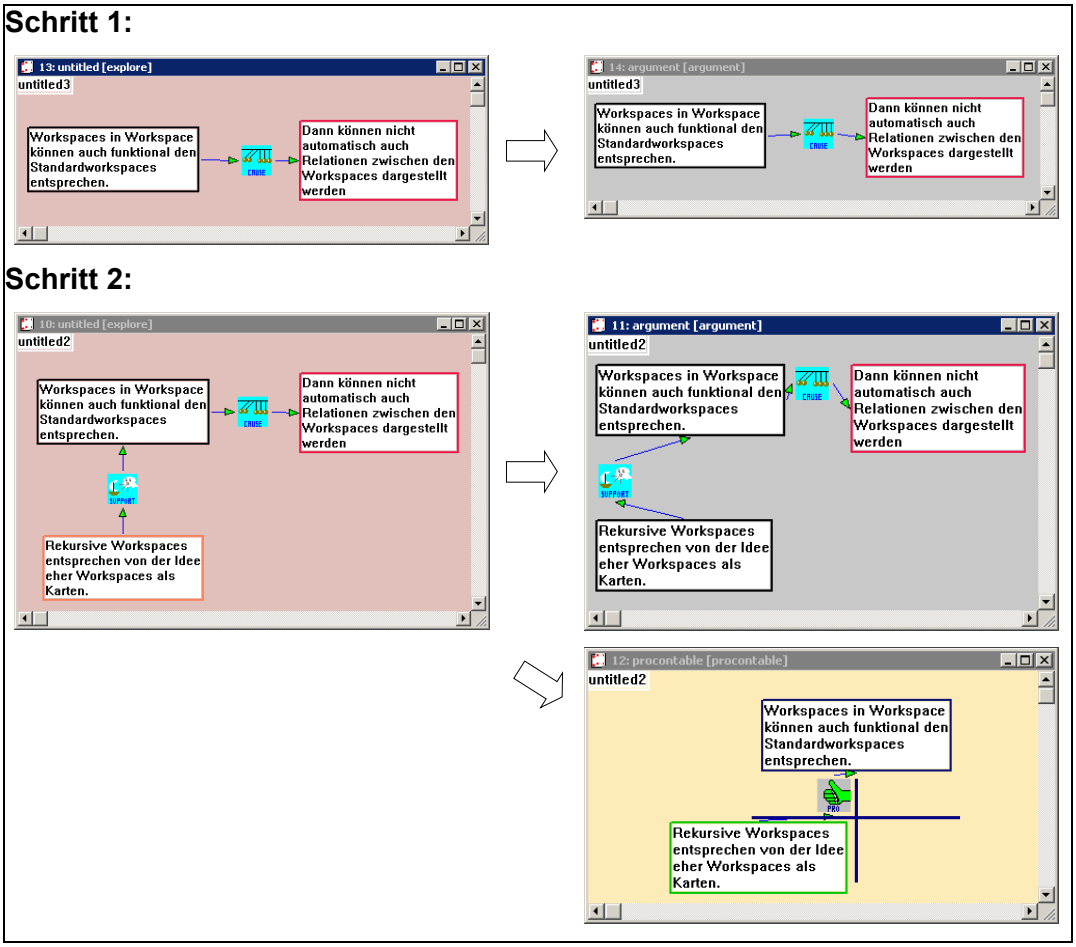


Abbildung 72

Schritt 1: Inhaltskarten sind mit der „Cause“-Relation verbunden (links). Sie werden als Argumentkern erkannt und in die Argumentperspektive abgebildet (rechts).
Schritt 2: Links: Eine weitere Inhaltskarte wird hinzugefügt und mit einer „Support“-Relation verbunden. Rechts oben: Diese Darstellung wird als Erweiterung für das bestehende Argument erkannt und in die Argumentperspektive übernommen. Rechts unten: Gleichzeitig wird ein Strukturkern für die Darstellung als Pro-/Kontratabelle erkannt.

Zu Beginn wird sicher gestellt, dass der erstellte Link auch Teil einer vollständigen Relation ist. Zusätzlich wird die Richtung, in der die Relation verläuft, herausgefunden und mit den entsprechend benannten Variablen unifiziert. Danach wird untersucht, ob es sich um eine Extension bzw. einen Strukturkern handelt. Die Reihenfolge ist dabei weitgehend unwichtig, da beides unabhängig voneinander zutreffen kann. Eine Relation

kann gleichzeitig für eine oder mehrere Strukturen eine Erweiterung sein und den Kern neuer Strukturen bilden.

Ein Beispiel dazu ist in Abbildung 72 dargestellt. Dort sind in der Explorationsperspektive (links) zwei Beiträge in einem ersten Schritt über eine „Cause"-Relation ins Verhältnis gesetzt worden. Dies wird als Argumentstruktur erkannt und in die Argumentperspektive übernommen (rechts oben). In einem zweiten Schritt wurde eine zusätzliche Inhaltskarte über die „Support"-Relation hinzugefügt (links). Dieser Zusatz wird als Extension für das bestehende Argument erkannt und dem Argument der Argumentperspektive hinzugefügt (rechts). Außerdem stellt diese Erweiterung, die in Bezug auf das Argument eine Extension bildet, auch einen neuen Kern für eine Tabelle dar. Die daraus zu entwickelnde Tabellendarstellung ist rechts unten in der Abbildung zu sehen.

```
extension_test(– Topic, – Language, – Connector, – Condition, – Conclusion):-
        extension       (Language, Type, + Structure,
                         view_rel(Connector, Condition, Conclusion)),
        extension_map   (Topic, Language, Type, – Structure,
                         [view_rel(Connector, Condition, Conclusion)]),
        fail.

extension_test(_, _, _, _, _):- !.
```

extension_test erzwingt ein Backtracking über alle im Diskussionsmodell definierten Extensionen, weil eine Erweiterung gleichzeitig mehrere Strukturen erweitern könnte. In den Argumenten spezifiziert ist der „Topic", die visuelle Sprache und die Identifikatoren der Karten, die in der visuellen Darstellung die Extension ausmachen: die Konnektorkarte *Connector* und die Inhaltskarten *Condition* und *Conclusion*. Wird eine Extension gefunden, wird dabei auch der Typ der Struktur spezifiziert (*Type*) und der konkrete Struktur-Identifikator ermittelt. Im zweiten Schritt wird die eigentliche Abbildung durchgeführt (*extension_map*), die zur Veränderung des internen Modells führt. Die Abbildungsvorschrift ist Bestandteil des Diskussionsmodells (siehe Kapitel 6.8) und wird dort weiter erläutert.

Unabhängig davon, ob in *extension_test* schon Extensionen gefunden wurden, werden in *core_test* nun alle zutreffenden Strukturkerne (*core*) gesucht.

```
core_test(– Topic, – Language, – Connector, – Condition, – Conclusion):-
        core(Language, + Type, [view_rel(Connector, Condition, Conclusion)]),
        core_extensions(Topic, Language, Type, view_rel(Connector, Condition, Conclusion)),
        fail.

core_test(_, _, _, _, _, _):- !.
```

Wird ein Kern festgestellt, so kann es außerdem sein, dass für diesen bereits Erweiterungen bestehen, die vorher noch keine eigenständigen Strukturen darstellten, die nun aber ebenfalls mit in das Modell mit aufgenommen werden müssen. Dafür ist das Prädikat *core_extensions* zuständig.

```
core_extensions (_,– Language, – Type, view_rel(– Connector, – Condition, – Conclusion)):-
        extension(Language, Type, _, view_rel(Connector, Condition, Conclusion)), !.

core_extensions (– Topic, – Language, – Type,
                        view_rel(– Connector, – Condition, – Conclusion)):-
        structure_map(– Topic, – Language, – Type, – Structure,
                        [view_rel(– Connector, – Condition, – Conclusion)]), !,
        add_existing_extensions(– Topic, – Language, – Type, – Structure), !.
```

In *core_extensions* muss außerdem das Problem gelöst werden, dass Extensionen und Kerne gleich aussehen können. Abbildung 73 zeigt links so einen Fall. Darin bildet eine „Support"- und eine „Attack"-Relation die Ausgangsinformation für eine Tabellenstruktur (rechts). Beide Relationen können sowohl einen Kern als auch eine Extension bilden. Welche Kategorie zutrifft ist rein zeitlich definiert: Die erste zulässige Relation bildet auch den Kern, alle weiteren bilden Extensionen.

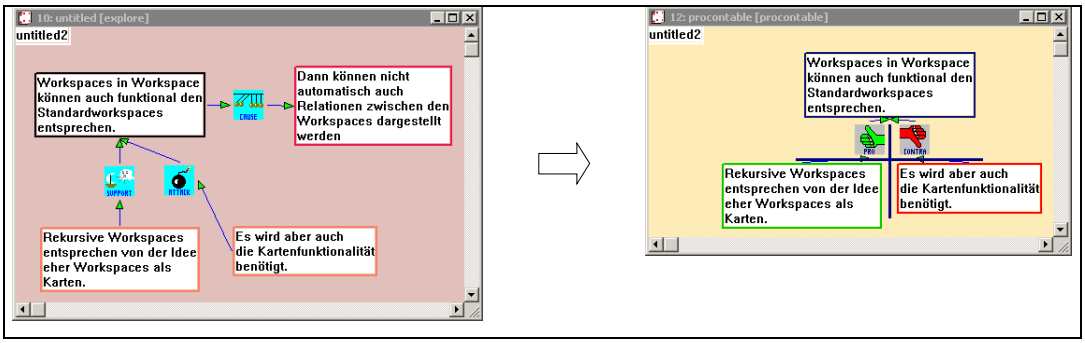


Abbildung 73

Erweiterung einer Tabellenstruktur in Anschluss an Abbildung 72.

Weil Extensionen und Kerne gleich aussehen können, muss in der ersten Regel von *core_extensions* wieder geprüft werden, ob der gefundene Kern aus *core_test* nicht schon

als Extension behandelt worden ist. Der fehlerhafte Fall, der ansonsten auftreten könnte, wäre der folgende:
Unter der Voraussetzung, dass Extension und Kern identisch aussähen, würde zuerst in *extension_test* eine Extension zu einem Kern gefunden, hier die „Attack"-Relation, die erst als Zweites dazu gekommen ist. In *core_test* würde die gleiche Relation auch als Kern erkannt werden. Ohne die erste Regel in *core_extensions* würde nun dieser erkannte Kern zu einem neuen Strukturmodell in der Datenbasis führen. Schon das wäre nicht wünschenswert. Außerdem würde aber noch in *add_existing_extensions* der bereits bestehende Kern als Erweiterung des neu entdeckten Kerns klassifiziert. So hätte man die gleiche Struktur zwei Mal erkannt und der Datenbasis hinzugefügt. In der Tabellendarstellung würde sich das in der doppelten Darstellung der gleichen Tabelle äußern.
Im Prinzip wird dadurch eine Priorität bzgl. Kern und Extension festgelegt, eine Relation, wenn möglich, als Erweiterung, nicht als neue Struktur zu identifizieren.
In der zweiten Regel von *core_extensions* findet die eigentliche Abbildung des Strukturkerns statt (*structure_map*). *add_existing_extensions* erweitert die neuen Modelle um die Modellteile bereits bestehender Erweiterungen.
Dieser Fall, dass für neue Kerne bereits Extensionen existieren, die davor aber noch nicht als solche relevant waren, kann im Prinzip auch für die Extensionen zutreffen. D.h. auch wenn eine Extension erkannt wird, könnten dafür bereits Erweiterungen bestehen, die erst durch das „Verbindungsstück" mit relevant werden. Da aber alle für die Diskussions-unterstützung definierten Strukturen nicht so aufgebaut sind, ist diese Option nicht realisiert worden.

6.7.4 Strukturabbildungen

Genauso, wie die Strukturerkennung mit veränderbaren Strukturen umgehen muss, muss dies auch die Strukturabbildung berücksichtigen. Wie die Strukturüberprüfung finden die Strukturabbildungen sukzessive statt. Das heißt, dass zuerst die Strukturkerne erkannt werden müssen, aus denen das initiale Strukturmodell generiert wird. Alle später hinzukommenden Teile sind Extensionen. Sie führen zur Ergänzung vorhandener Strukturmodelle.
Auch für die Strukturen wird das Konzept der „delayed changes" ausgenutzt. Das bedeutet, dass für die Inhaltskarten, die Bestandteil einer erkannten Struktur sind, ebenfalls Knotencluster erstellt werden. Die Abbildung der Strukturen beinhaltet damit die Abbildung der beteiligten Karten. Im Gegensatz zur Abbildung einzelner Karten, die

sofort beim Erstellen abgebildet werden, können die Karten für Strukturen aber erst abgebildet werden, wenn die Struktur entstanden ist, nicht wenn die Karte erstellt wird. Typischerweise wird für die Karten aus den Strukturen deshalb erst zum Zeitpunkt der Strukturabbildung ein Knotencluster generiert.

Das Prädikat *structure_map* implementiert die Abbildung der Strukturkerne. Diese Abbildungsregeln sind Teil des Diskussionsmodells und werden im gleichnamigen Kapitel 6.8 noch mit verschiedenen konkreten Beispielen vorgestellt. Hier soll lediglich das Prinzip einer Abbildung erläutert werden.

structure_map(– Topic, + language, + type, + Structure, – Relations)

Eine Definition einer Strukturabbildung generiert für eine festgelegte Perspektive (*language*) und einen Strukturtyp (*type*) ein neues Modell für eine Struktur. Generiert wird dafür auch ein Identifikator (*Structure*). Als Parameter der aktuellen Situation wird dafür der gerade bearbeitete „Topic“ mit übergeben und die Identifikatoren von Karten in Relationen, die hier verkürzt als *Relations* benannt sind. *structure_map* generiert das neue Modelle und fügt es direkt der Datenbasis hinzu.

Die Relationen, die auf die konkreten Oberflächenobjekte verweisen, stellen sich in der bereits bekannten Form dar, in der an erster Position die Konnektorkarte referenziert wird, danach die zwei Inhaltskarten einer Relation, einschließlich der Zuordnung einer Richtung für die Relation.

[**view_rel**(Connector, Condition, Conclusion) | Relations]

Für alle Karten, die in diesen Relationen referenziert werden, muss ein Knotencluster erstellt werden. Dafür ist das Prädikat *adapt_cluster* zuständig.

adapt_cluster(+ language, + type, – Structure, + mapDefinition, – Condition, + Cluster)

Dessen Funktion besteht vor allem darin, für eine Karte ein neues Cluster (*Cluster*) zu erstellen und dabei zu überprüfen, ob nicht bereits eines bestanden hat. Die zweite wichtige Aufgabe besteht darin, die Abbildung für die Karten zu definieren, deren Abbildung aus den Strukturen heraus angestoßen wird.

Ein allgemein gehaltenes Beispiel für *structure_map* könnte wie folgt aussehen:

```
structure_map(Topic, language1, type, Structure, [view_rel(Connector, Condition, Conclusion)]):-
        unique_db_id    (Structure),
        adapt_cluster    (language1, type, Structure,
                         [(language2 ,cardType2, Topic, default)], Condition, Cluster1),
```

```
adapt_cluster   (language1, type, Structure,
                [(language2, cardType3, Topic, default)], Conclusion, Cluster2),
assert          (structure(Topic, Type, Structure,
                model   ( [ (language1, [model_rel(connector1,Cluster1,Cluster2)]) ,
                          (language2, [model_rel(connector2, Cluster1, Cluster2) ] )
                ])), !.
```

Im Bedingungsteil ist die Sprache (*language1*) und der Typ der Struktur definiert (*type*), für den diese Abbildung auszuführen ist. *unique_db_id(Structure)* generiert einen eindeutigen Identifikator für die neue Struktur. Danach wird für die Knoten, die in der Abbildung beteiligt sind *adapt_cluster* ausgeführt. Schließlich wird das eigentliche Strukturmodell der Datenbasis hinzugefügt. Für jede Perspektive, unter der die Struktur existieren soll, werden die Typen der Relation und deren Zugehörigkeit zu den Knotenclustern aufgeführt. So kann perspektivenübergreifend das Verhältnis zwischen den Strukturen bestimmt werden.

In einer Abbildung kann es auch dazu kommen, dass zusätzliche Knoten generiert werden sollen (ein Beispiel dazu wird in Kapitel 6.8.2 vorgestellt). Dafür kann mit *new_cluster* auch direkt ein neues Cluster erstellt werden.

Die blau gekennzeichneten Konstanten sind Bestandteile, die das Diskussionsmodell ausmachen. Sie werden nicht generiert, sondern festgelegt.

Wie die Strukturmodelle die Visualisierungen beschreiben wird kurz anhand von Abbildung 74 und Abbildung 75 erläutert. Darin ist jeweils oben ein mögliches Strukturmodell angegeben. In der Grafik sind die zwei Perspektiven (*language1* und *language2*) symbolisiert. Die Rechtecke stellen Inhaltskarten dar, die Kreise Konnektorkarten. Die schwarzen, durchbrochenen Linien deuten an, dass diese zwei Kartenobjekte der unterschiedlichen Perspektiven zum gleichen Knotencluster gehören. Das Beispiel in Abbildung 74 stellt eine Umkehrung der Relationsrichtung zwischen den Perspektiven dar.

```
model( [    (language1, [model_rel(type1, cluster1, cluster2)]),
            (language2, [model_rel(type2, cluster2, cluster1)])    ])
```

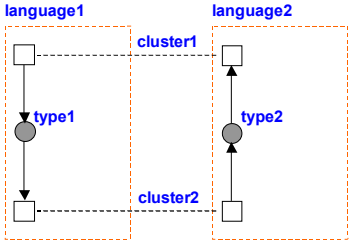


Abbildung 74

Beispiele für eine Strukturabbildung.

```
model( [    (language1, [model_rel(type1, cluster1,cluster2)]) ,
            (language2, [model_rel(type2, cluster3, cluster1),
                         model_rel(type3, cluster2, cluster3)])    ])
```

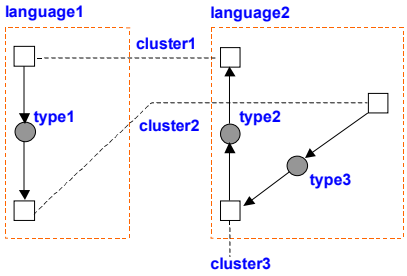


Abbildung 75

Beispiel für eine Strukturabbildung.

In Abbildung 75 wird ein Beispiel vorgestellt, in dem die Strukturen der verschiedenen Perspektiven deutlicher unterschieden sind. Es existieren drei Knotencluster, von denen eines nur eine Karte beinhaltet (*cluster3*). In einer Perspektive gehört nur eine Relation zur Struktur (*language1*), in der anderen zwei Relationen (*language2*). Die Modelldarstellung beinhaltet keine Annahmen über die Ähnlichkeit der Strukturen in den Perspektiven. Pro Perspektive werden die zugehörigen Relationen aufgeführt, die jeweils nicht auf die Karten, sondern auf die Knotencluster verweisen.

Entsprechend zu den Abbildungen der Strukturkerne müssen auch die Abbildungen der Extensionen definiert werden. Das Prädikat *extension_map* entspricht dem bereits vorgestellten Prädikat *structure_map* und es besitzt die gleichen Parameter.

```
extension_map(– Topic, + language, + type, + Structure, – Relations)
```

Auch in *extension_map* werden die Abbildungen der Strukturteile mit dem Generieren von Knotenclustern verbunden. Dazu wird ebenfalls die Prozedur *adapt_cluster* eingesetzt. Im Prädikat *extension_map* muss das bestehende Modell erweitert werden. Dafür wird die bestehende Eintragung aus der Datenbasis gelöscht und in veränderter Form wieder hinzugefügt.

6.7.5 Löschen in Strukturen

In Strukturen zu löschen bedeutet, dass Inhaltskarten oder Konnektorkarten gelöscht werden. In beiden Fällen müssen die Strukturmodelle angepasst werden. Für jede Veränderung der Struktur in den Workspaces werden die Einträge *model_rel* im Modell angepasst, beim Löschen also aus dem Modell herausgenommen.

Das Löschen von Karten verwendet wieder das Konzept der „delayed changes“. Werden in einem Fenster Karten gelöscht, wird in dem anderen die Löschung aber nicht akzeptiert, so entwickeln sich die Strukturen auseinander.
Das Löschen von Konnektorkarten führt direkt zur Änderung im Strukturmodell. Eine Löschung einer Konnektorkarte betrifft immer nur die Perspektive, aus der die Löschung durchgeführt wurde.

6.8 Das Diskussionsmodell

Das Diskussionsmodell beinhaltet die diskussionsspezifischen Abbildungen. Darunter fallen die Abbildungsvorschriften der Inhaltskarten und die der Strukturen.

6.8.1 Abbildungsdefinitionen für Inhaltskarten

Die Abbildungen der Karten zwischen den Perspektiven sind in dem Prädikat *translate* im Diskussionsmodell definiert.

```
translate (– Language1, – Type1, – Topic1, + Language2, + Type2, + Topic2).
```

Das Prädikat liest sich von links nach rechts: Die Karte vom Typ *Type1* aus der visuellen Sprache *Language1* wird abgebildet in eine Karte vom Typ *Type2* einer Sprache *Language2*. Das Thema kann dabei mit definiert werden. Eine Abbildung zwischen zwei Perspektiven stellt sich im Standardfall dar wie z.B.:

```
translate(explore, subject, Topic, protocol, task, Topic).
```

Dabei wird eine Inhaltskarte vom Typ „Subject“ aus der Sprache „Explore“ abgebildet in eine Karte vom Typ „Task“ der visuellen Sprache „Protocol“. Das Thema bleibt dabei identisch.

Abbildungen aus der Protokollperspektive

Ausgehend von der Annahme, dass die Protokollperspektive auch zur Planung einer Diskussion verwendet werden soll, werden Grundfragestellungen („Task“), Teilfragen („Question“) und einfache Beiträge („Contribution“), die im Verlauf fallen, in die Explorationsperspektive übernommen. Es wird davon ausgegangen, dass alle diese Beiträge in einer anschließenden vertiefenden Diskussion wieder aufgegriffen werden sollten.

```
%% protocol to explore
        translate(protocol, task, Topic, explore, subject, Topic).
```

```
        translate(protocol, question, Topic, explore, subject, Topic).
        translate(protocol, contribution, Topic, explore, estimation, Topic).
        translate(protocol, topic, Topic, explore, topic, Topic).
```

Diese Vorschriften, die den Arbeitsprozess unterstützen sollen und durch den typischen Ablauf einer Diskussion motiviert sind, nehmen keinen Einfluss auf den „Topic“. Die nachfolgenden Vorschriften, die zur Verwaltung der „Topics“ notwending sind, definieren den aktuellen „Topic“ auch für die anderen Perspektiven. Die „Topic“-Karten bilden somit für jedes behandelte Thema jeweils ein Cluster. Auf diese Weise kann die Themenbenennung perspektivenübergreifend koordiniert werden.

```
%% protocol to argument
        translate(protocol, topic, Topic, argument, topic, Topic).
%% protocol to brainstorm
        translate(protocol, topic, Topic, brainstorm, topic, Topic).
%% protocol to procontable
        translate(protocol, topic, Topic, procontable, topic, Topic).
%% protocol to topics
        translate(protocol, topic, Topic, topics, topicnode, themen).
```

Für die Abbildung in die „Topic Map“ verändert sich das Thema und der Knotentyp. Die vorher in den Themen-Workspaces als „Topic“-Karte eingeblendeten Themen werden in der „Topic Map“ als „Topicnode“-Karten repräsentiert und werden dort in die Themenübersicht einbezogen. Da die „Topic Map“ ein vordefiniertes Thema (*themen*) besitzt, ist dies in der Abbildungsvorschrift angegeben.

Abbildungen aus der Explorationsperspektive

Die Explorationsperspektive kann in vieler Hinsicht als zentrale Perspektive angesehen werden. Es wird davon ausgegangen, dass die meisten Themen hauptsächlich darin aufgearbeitet werden. Sie bildet zwar die Grundlage für viele Strukturabbildungen, einzelne Karten werden bisher aber nur zurück in die Protokollperspektive propagiert, wo sie einer Zusammenfassung der Diskussion dienen. Dazu gehören Ergebnisse („Result“) und neue Aspekte, die erst während der Exploration entstanden sind („Subject“).

```
%% explore to protocol
        translate(explore, subject, Topic, protocol, task, Topic).
        translate(explore, result, Topic, protocol, contribution, Topic).
        translate(explore, topic, Topic, protocol, topic, Topic).
%% explore to argument
        translate(explore, topic, Topic, argument, topic, Topic).
%% explore to brainstorm
        translate(explore, topic, Topic, brainstorm, topic, Topic).
```

```
%% explore to procontable
        translate(explore, topic, Topic, procontable, topic, Topic).
%% explore to topics
        translate(explore, topic, Topic, topics, topicnode, themen).
```

Abbildungen aus der Brainstormingperspektive

Wiederum mit dem Ziel, entstandene Inhalte noch einmal in der Protokollperspektive aufzugreifen, werden aus der Brainstormingperspektive Gründe und Optionen aufgegriffen.

```
%% brainstorm to protocol
        translate(brainstorm, option, Topic, protocol, idea, Topic).
        translate(brainstorm, reason, Topic, protocol, contribution, Topic).
        translate(brainstorm, topic, Topic, protocol, topic, Topic).
%% brainstorm to argument
        translate(brainstorm, topic, Topic, argument, topic, Topic).
%% brainstorm to explore
        translate(brainstorm, topic, Topic, explore, topic, Topic).
%% brainstorm to procontable
        translate(brainstorm, topic, Topic, procontable, topic, Topic).
%% brainstorm to topics
        translate(brainstorm, topic, Topic, topics, topicnode, themen).
```

Abbildungen aus der Argumentperspektive

Aus der Argumentperspektive finden keine Kartenabbildungen statt, außer der Verwaltung der „Topics“. Die Argumentperspektive stellt selbst eine fokussierte Auswertung dar.

```
%% argument to protocol
        translate(argument, topic, Topic, protocol, topic, Topic).
%% argument to brainstorm
        translate(argument, topic, Topic, brainstorm, topic, Topic).
%% argument to procontable
        translate(argument, topic, Topic, procontable, topic, Topic).
%% argument to explore
        translate(argument, topic, Topic, explore, topic, Topic).
%% argument to topics
        translate(argument, topic, Topic, topics, topicnode, themen).
```

Abbildungen aus der Pro/Kontraperspektive

Wie die Argumentperspektive dient auch die Pro/Kontraperspektive der übersichtlichen Zusammenfassung. Deshalb werden auch hier keine Einzelaspekte weiter propagiert.

```
%% procontable to protocol
        translate(procontable, topic, Topic, protocol, topic, Topic).
%% procontable to brainstorm
        translate(procontable, topic, Topic, brainstorm, topic, Topic).
%% procontable to argument
        translate(procontable, topic, Topic, argument, topic, Topic).
%% procontable to explore
        translate(procontable, topic, Topic, explore, topic, Topic).
%% procontable to topics
        translate(procontable, topic, Topic, topics, topicnode, themen).
```

Abbildungen aus der Topicperspektive

Eine Möglichkeit besteht darin, zwischen Perspektiven abzubilden, jedoch das Thema dabei nicht zu beeinflussen. Dies stellt den Standardfall dar, bei dem typischerweise, ausgehend von einer Karte, nicht in alle möglichen Perspektiven abgebildet wird, sondern nur in eine kleine Auswahl, für die eine Bedeutung in einem Arbeitsprozess gesehen wird.

Die Behandlung der „Topics“ unterscheidet sich davon. Sie ordnen den Inhalten der Perspektiven Themen zu und strukturieren diese damit semantisch. Die Behandlung der „Topics“ unterliegt einer besonderen Kontrolle, da sichergestellt werden muss, dass jeweils nur eine Themenkarte in einem Workspace enthalten ist. Die „Topic“-Karten in den Perspektiven stellen eine Verbindung zu der übergeordneten „Topic Map“ dar. Die „Topic Map“ ist aber selbst wieder eine Perspektive und ist deshalb selbst einem Thema zugeordnet. Da es nur eine dieser Übersichten geben darf, ist dieses Thema festgelegt als *themen*. Die Inhaltskarten der visuellen Sprache der „Topic Map“ müssen davon unterschieden werden und werden als „Topicnode“-Karten bezeichnet. Wird ein neuer Workspace geöffnet, so wird dieser einem neuen Thema in Form einer generierten „Topic“-Karte zugeordnet, die in der „Topic Map“ als „Topicnode“-Karte aufgenommen wird.

Anders herum muss ebenfalls ein Cluster für ein Thema erstellt werden, wenn direkt in der „Topic Map“ ein Thema definiert wird. Die frei Variable *Topic*, die in den Abbildungsvorschriften definiert ist, wird bei der Erstellung des Themas instanziert.

```
%% topics to protocol
        translate(topics, topicnode, themen, protocol, topic, Topic).
%% topics to explore
        translate(topics, topicnode, themen, explore, topic, Topic).
%% topics to argument
        translate(topics, topicnode, themen, argument, topic, Topic).
```

```
%% topics to procontable
        translate(topics, topicnode, themen, procontable, topic, Topic).
%% topics to brainstorm
        translate(topics, topicnode, themen, brainstorm, topic, Topic).
```

6.8.2 Abbildungsdefinitionen für Strukturen

Neben den Abbildungsdefinitionen für einzelne Inhaltskarten werden auch die Abbildungen für die Strukturen im Diskussionsmodell definiert. Für das Diskussionsmodell werden zwei als relevant eingestufte Arten von Strukturen herausgegriffen:

- Tabellen
- Argumente

Für diese Darstellungen existieren die beiden bereits erläuterten Perspektiven, die Pro-/Kontraperspektive und die Argumentperspektive. Die Abbildungsdefinitionen für diese Strukturen beinhalten die Informationsfilter, um die dafür geeigneten Informationen aus anderen, unter Umständen mehreren, Perspektiven herauszugreifen und die Generatoren, die daraus Strukturmodelle generieren. Die folgenden Beispiele stellen die implementierten Strukturabbildungen dar.

Filtern von Argumenten aus der Explorationsperspektive

In der Argumentperspektive werden die Argumente der Explorations- und Brainstorming-perspektive zusammengefasst, die sich dort jeweils unterschiedlich darstellen. Wegen der Übereinstimmung der Sprachobjekte der Argumentationssprache mit einem Teil der Explorationssprache beschränkt sich das Filtern der Argumente aus der Explorationsperspektive auf das Erkennen derselben in größeren Strukturen, deren Selektion und deren darauffolgende Rekonstruktion in der Argumentperspektive. Abbildung 76 greift noch einmal auf, wie Argumente in der Explorationsperspektive dargestellt werden.

Abbildung 76

Schematische Darstellung von Argumenten in der Explorationsperspektive. Links: Schema eines Argumentkerns. Rechts: Kern mit Erweiterungen.

Abbildung 77 zeigt dazu ein größeres Beispiel[15]. Darin ist auf den ersten Blick nicht zu erkennen, an welcher Stelle sich Argumente herausgebildet haben. Inhaltlich werden Aspekte zum Thema Staatshaushalt aufgeführt.

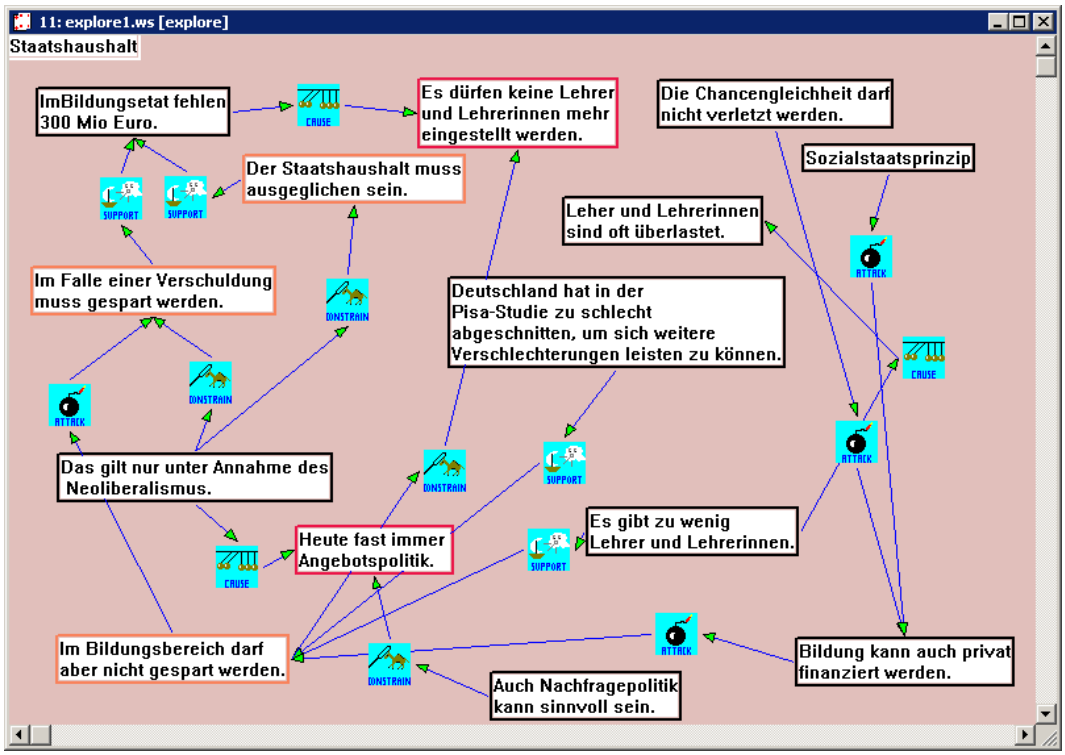


Abbildung 77

Beispiel für einen Explorations-Workspace, der verschiedene Argumente enthält, die in einen größeren Kontext gesetzt sind.

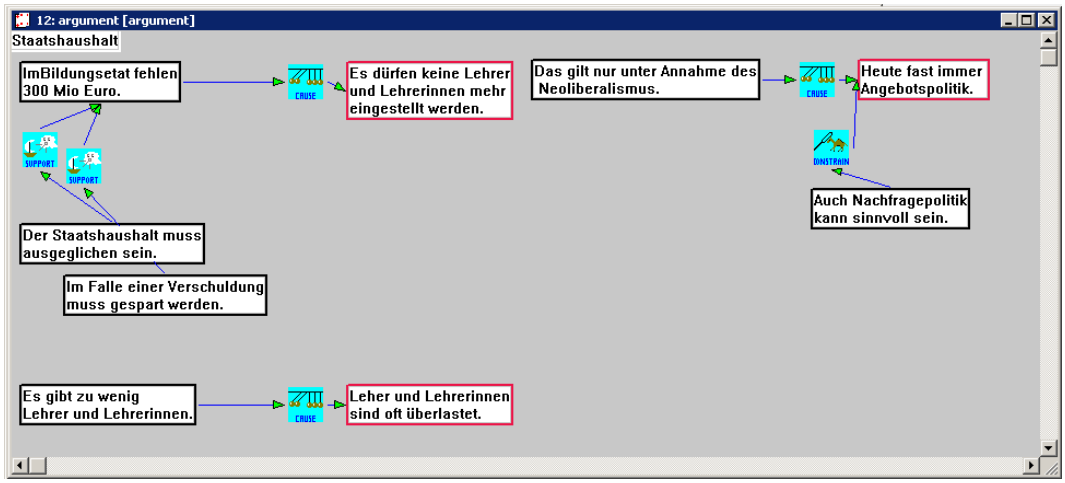


Abbildung 78

Der Argument-Workspace basiert auf den Informationen, die in Abbildung 77 dargestellt sind.

[15] Das Beispiel lehnt sich an einer Darstellung von W. Harst an [harst].

Abbildung 78 zeigt das Resultat nach dem Filtern der Argumente in einem Argument-Workspace. Die Bestandteile der Argumente haben sich wegen der gleichen Sprachkonstrukte der beiden visuellen Sprachen nicht verändert. Sie wurden jedoch aus dem Kontext herausgegriffen. Es wird deutlich, dass auch ein einfaches Filtern von Teilstrukturen zur Übersicht beitragen kann.

Im weiteren Verlauf können sich die Strukturen in den beiden Perspektiven unterschiedlich entwickeln. Alle Erweiterungen aus der Explorationsperspektive werden, soweit möglich, in die Argumentperspektive propagiert. Die umgekehrte Richtung ist nicht definiert, so dass die Darstellung in der Argumentperspektive letztlich zur verfeinerten Darstellung der Explorationsperspektive wird. Diese Tatsache ist nicht durch den technischen Ansatz determiniert. Es könnten durchaus Regeln für die Abbildung in die entgegengesetzte Richtung aufgenommen werden. Dies nicht zu tun resultiert aus der Annahme, dass die Verfeinerung unter einer anderen Perspektive gerade sinnvoll ist und nicht notwendig in die allgemeine Erörterung der Explorationsperspektive übernommen werden braucht.

Die folgenden Regeln sind für die Abbildung des Strukturkernes definiert. Da deren Prinzip bereits ausführlich erläutert wurde, werden Teile der folgenden Abbildungen in verkürzter Form aufgeführt und an entsprechenden Stellen durch ein „*“ gekennzeichnet.

```
core(explore, argument, [view_rel(Connector, Condition, Conclusion)]):-
        has_one(relation([cause], Connector, [fact], Condition, [result, estimation], Conclusion)).
```

Als Kern ist eine „Cause“-Relation deklariert, die eine „Fact“-Karte im Bedingungsteil mit einer „Result“- oder „Estimation“-Karte in der Konklusion verbindet. *explore* steht für die visuelle Sprache, für die dieser Kern definiert ist und *argument* für den Strukturtyp.

```
structure_map(Topic, explore, argument, Structure, [view_rel(_, Condition, Conclusion)]):-
        …
        unique(+ Structure),
        adapt_cluster( (argument, fact, Topic, default), Condition, Cluster1), *
        adapt_cluster( (argument, result, Topic, default), Conclusion, Cluster2),  *
        assert(     structure(Topic, argument, Structure,
                            model( [  (explore,    [model_rel( cause, Cluster1, Cluster2) ]) ,
                                      (argument, [model_rel( cause, Cluster1, Cluster2) ] ) ]
        ))),
        !.
```

Die Vorschrift für die Abbildung des Strukturkerns bewirkt, dass zwei neue Knotencluster angelegt werden, die für die Argumentperspektive auf eine „Fact"-Karte (*Cluster1*) und eine „Result"-Karte (*Cluster2*) verweisen. In das Strukturmodell wird für beide betroffenen Perspektiven jeweils eine „Cause"-Relation aufgenommen, die auf *Cluster1* und *Cluster2* referenziert.
Als Informationsgrundlage für die Argumentstruktur sind für die Explorationsperspektive außerdem die folgenden Extensionen festegelegt.

```
extension(explore, argument, Structure, view_rel(Connector, Condition1, Conclusion1)):-
        ...
        has_one(relation(       [support], Connector,
                                [fact, estimation], Condition1,
                                [fact], Conclusion1)),
        core(explore, argument, [view_rel(_, Conclusion1, _)]),
        ...
        in_structure(Conclusion1, explore, argument, Structure).

extension(explore, argument, Structure, view_rel(Connector, Condition1, Conclusion1)):-
        ...
        has_one(relation(       [constrain], Connector,
                                [fact], Condition1,
                                [result, estimation], Conclusion1)),
        core(explore, argument, [view_rel(_, Condition2, Conclusion1)]),
        ...
        in_structure(Conclusion1, explore, argument, Structure).
```

Definiert sind zwei unterschiedliche Erweiterungsmuster mit jeweils zwei möglichen Ausprägungen. Der erste „Constraint" deklariert Extensionen für den Bedingungsteil eines Arguments, der zweite für die Konklusion. Die Kartentypen der Erweiterungen werden in *has_one* vorgegeben, die Strukturdefinition ergibt sich aus dem Zusammenhang der Parameter aus dem Term *view_rel*. Der erste „Constraint" fordert, dass die Konklusion (*Conclusion1*) aus der Extension, der Bedingung des Argumentkernes entsprechen muss. Der zweite „Constraint" fordert, dass die Konklusion (*Conclusion1*) aus der Extension, der Konklusion des Argumentkernes entsprechen muss. *in_structure* stellt letztlich sicher, dass für die Extension tatsächlich schon eine Struktur existiert.
Der Extensionsüberprüfung folgt deren Abbildung und Aufnahme in das Strukturmodell.

```
extension_map(Topic, explore, argument, Structure, [view_rel(Connector, Condition, _)]) :-
        …
        retract( structure(Structure, + Model) ), *
        find_model_rel(Model, argument, model_rel( cause, _, Cluster3) ),
        adapt_cluster( (argument, fact, Topic, default), Condition, Cluster1), *
        adapt_model(Model, + NewModel,
                        [ (explore,     [model_rel( constrain, Cluster1, Cluster3)]),
                          (argument,  [model_rel( constrain, Cluster1, Cluster3)]) ]),
        assert( structure(Topic, argument, Structure, NewModel)), !.

extension_map(Topic, explore, argument, Structure, [view_rel(Connector, Condition, _)]) :-
        …
        retract( structure(Structure, + Model) ), *
        find_model_rel(Model, argument, model_rel( cause, Cluster2, _) ),
        adapt_cluster( (argument, fact, Topic, default), Condition, Cluster1),
        adapt_model(Model, + NewModel,
                        [ (explore,     [model_rel( support, Cluster1,Cluster2)]),
                          (argument,  [model_rel( support, Cluster1, Cluster2)]) ]),
        assert( structure(Topic, argument, Structure, NewModel)), !.
```

extension_map greift auf ein bestehendes Modell zu. Darin wird die Relation gesucht, mit der die Extension verbunden werden soll (*find_model_rel*). Für die neuen Karten der Extension, hier nur *Condition*, wird in *adapt_cluster* ein neues Knotencluster angelegt (*Cluster1*). Die neuen Relationen werden im Modell für die beiden betroffenen visuellen Sprachen, *explore* und *argument*, eingetragen.

Filtern von Argumenten aus der Brainstormingperspektive

Bei der Diskussionsunterstützung ist es nicht nur in der Explorationssprache, sondern auch in der Brainstormingsprache möglich, Argumente zu formulieren, die gemäß der Toulmin'schen Argumentstruktur interpretiert werden können.

Anders als bei der Explorationssprache ist die Brainstormingsprache jedoch grundverschieden von der Argumentsprache. Das heißt, dass eine wirkliche Interpretation der Karten mit ihrer Stellung innerhalb der Struktur stattfinden muss. Dafür vergegenwärtigt Abbildung 79 noch einmal die Idee der Strukturen in der Brainstormingsprache. Ausgehend von einer allgemeineren Frage („Task") werden Optionen aufgeführt („Option1" und „Option2"), die positiv und negativ gewichtet werden können. Auch Kriterien für diese Optionen können formuliert werden. Die Sterne in den Karten bedeuten, dass beliebig viele Wertungen hinzugefügt werden können.

In Abbildung 80 ist links eine Argument in der Brainstormingperspektive umrandet dargestellt. Die positiven Bewertungen können als stützende Bestandteile eines Argumentes verstanden werden. Die Kriterien dienen als Einschränkungen. Eine „Option"-Karte kann damit als Ausgangspunkt eines Argumentes verwendet werden. Für ein Argument fehlt dann jedoch noch die Konklusion, also die Frage danach, was daraus folgt, wenn eine Option verfolgt würde. Daraus könnten z.B. weitere Handlungsnotwendigkeiten erwachsen.

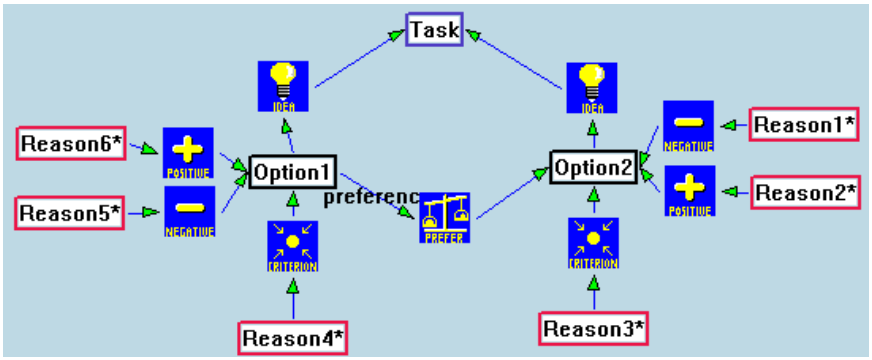


Abbildung 79

Idee der Strukturen in der Brainstormingsprache.

Abbildung 80 zeigt rechts, welche Lösung in diesem Ansatz zur Diskussionsunterstützung angestrebt wird. Und zwar wird an der Stelle der Folgerung in dem abgebildeten Argument eine leere Karte erzeugt. Diese Karte verweist auf eine potentielle Lücke bei der Argumentation. Dadurch wird hinterfragt, zu welchen Konsequenzen eine Option führen könnte.

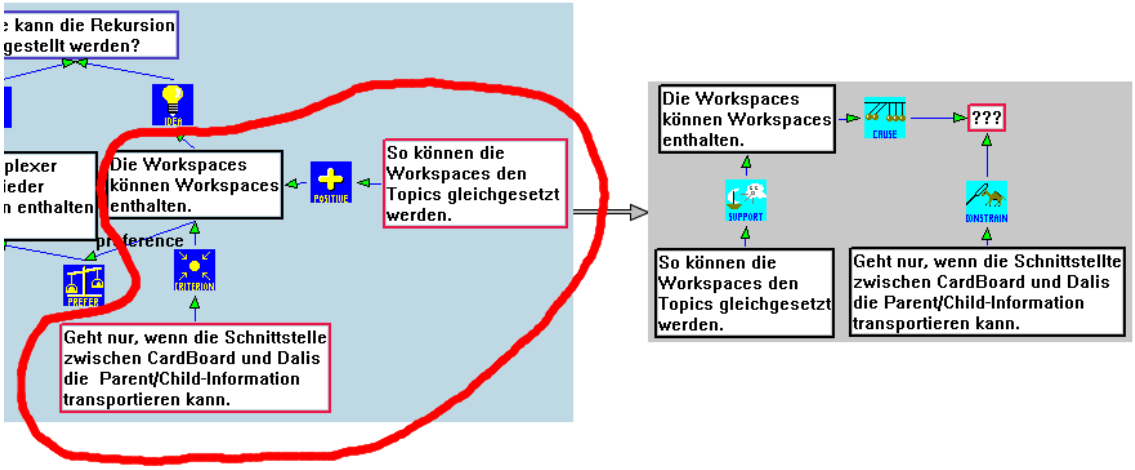


Abbildung 80

Beispiel für die Struktur eines Argumentes in der Brainstormingperspektive und die dafür implementierte Abbildung.

Negative Einschätzungen einer Option werden nicht in die Argumentdarstellung übernommen. Das liegt daran, dass die Toulmin'schen Darstellung keine Erörterung vorsieht. Zwar sind Einschränkungen der Argumentation vorgesehen aber nicht der Widerspruch. Sinnvoll ist es, die positiven und negativen Einschätzungen einer Option zusätzlich in einen Pro-/Kontra-Workspace abzubilden. Diese Vorschriften sind für diese Arbeit aber nicht mehr realisiert worden.

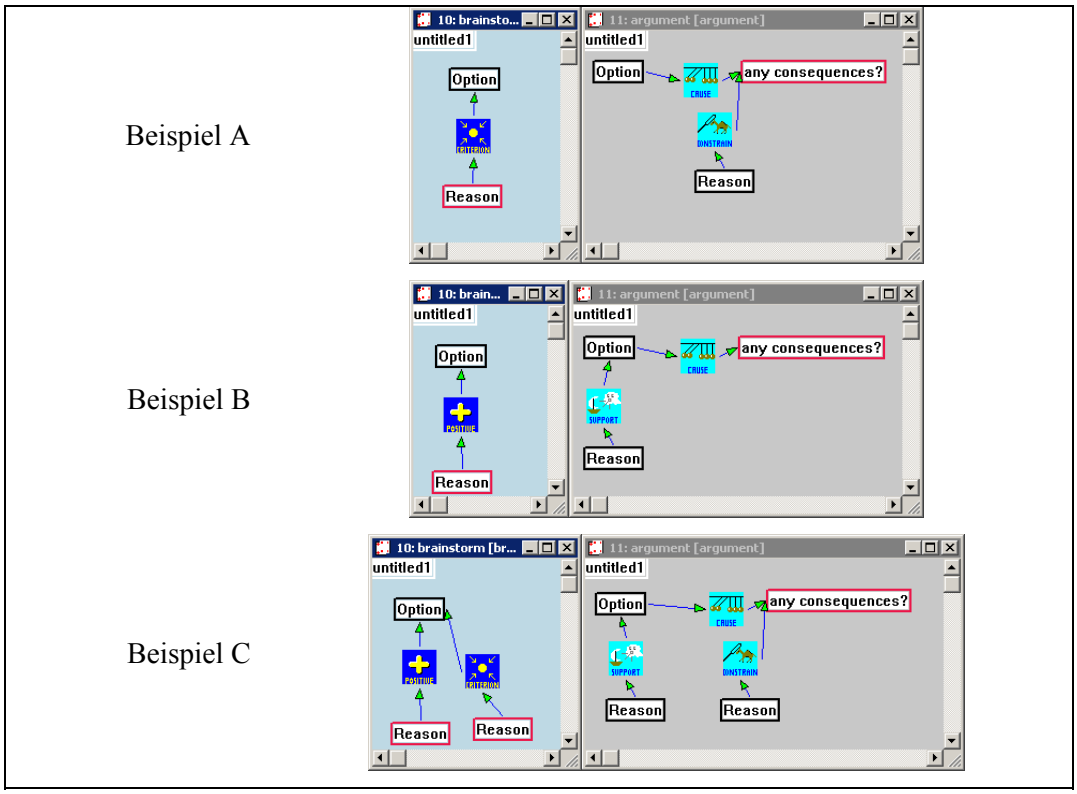


Abbildung 81

Schematische Darstellung der Abbildung von Strukturkernen von Argumenten aus der Brainstormingperspektive in die Argumentperspektive (Beispiel A und B). Erweiterungen der Argumente (Beispiel C).

Im Unterschied zu den Argumenten der Explorationsperspektive können Argumente in der Brainstormingperspektive durch zwei unterschiedliche Kerne begründet werden. Diese beiden Möglichkeiten werden in Abbildung 81, Beispiel A und B, verdeutlicht. Ein Kern besteht aus einer Konnektorkarte vom Typ „Criterion", die eine Begründung mit einer Option in Verbindung setzt. Der andere verbindet die gleichen Typen von Inhaltskarten mit der „Plus"-Relation. Da jeweils die Konklusion eines Argumentes generiert wird, reicht eine „Criterion"- oder „Plus"-Relation für das Generieren einer Argumentstruktur in der Argumentperspektive aus. Entsprechend sind für die Brainstormingperspektive zwei Kernstrukturen definiert:

```
core(brainstorm, argument, [view_rel(Connector, Condition, Conclusion)]):-
        has_one(relation([criterion], Connector, [reason], Condition, [option], Conclusion)).

core(brainstorm, argument, [view_rel(Connector, Condition, Conclusion)]):-
        has_one(relation([plus], Connector, [reason], Condition, [option], Conclusion)).
```

Die gleichen Strukturen, die als Kerne erkannt werden müssen, sind in der Erweiterung auch als Extensionen zulässig:

```
extension(brainstorm, argument, _, view_rel(Connector, Condition1, Conclusion1)):-
        ...
        has_one(relation(       [plus], Connector,
                                [reason], Condition1,
                                [option], Conclusion1)),
        core(brainstorm, argument, [view_rel(_, Condition2, Conclusion1)]),
        ...

extension(brainstorm, argument, _, view_rel(Connector, Condition1, Conclusion1)):-
        ...
        has_one(relation(       [criterion], Connector,
                                [reason], Condition1,
                                [option], Conclusion1)),
        core(brainstorm, argument, [view_rel(_, Condition2, Conclusion1)]),
        ...
```

Bei der Abbildung der Kernstrukturen wird die zusätzliche Karte für den Konklusionsteil des Arguments generiert (*new_cluster*). Für diese neue Karte wird auch ein Inhalt vorgegeben. Er besteht in der Frage „any consequences?“, die den Zweck dieser Karte erklärt und zur Reflexion anregen soll.

```
structure_map(Topic, brainstorm, argument, Structure, [view_rel(_, Condition, Conclusion)]):-
        …
        unique(+ Structure),
        adapt_cluster( (argument, fact, Topic, default), Condition, +Cluster1), *
        adapt_cluster( (argument, fact, Topic, default), Conclusion, +Cluster2), *
        new_cluster("any consequences?", (argument, result, Topic, default)], + Cluster3), *
        assert( structure(Topic, argument, Structure,
                        model( [    (brainstorm,    [model_rel( criterion, Cluster1, Cluster2) ]) ,
                                    (argument,      [model_rel( constrain, Cluster1, Cluster3),
                                                     model_rel( cause, Cluster2, Cluster3)] )] ))), !.
```

```
structure_map(Topic, brainstorm, argument, Structure, [view_rel(_, Condition, Conclusion)]):-
        …
        unique(+ Structure),
        adapt_cluster( (argument, fact, Topic, default), Condition, + Cluster1), *
        adapt_cluster( (argument, fact, Topic, default), Conclusion, + Cluster2), *
        new_cluster("any consequences?", (argument, result, Topic, default)], + Cluster3), *
        assert( structure(Topic, argument, Structure,
                        model( [   (brainstorm,   [model_rel( plus, Cluster1, Cluster2) ]) ,
                                   (argument,     [model_rel( support, Cluster1, Cluster2),
                                                   model_rel( cause, Cluster2, Cluster3)] )] ))), !.
```

Eine Schwierigkeit bei den Argumenten der Brainstormingperspektive liegt darin, dass die Extensionen und die Strukturkerne syntaktisch gleich sind. Aus diesem Grunde muss vor der Abbildung immer eindeutig ermittelt werden, ob eine Extension eben eine Erweiterung oder der Kern eines neuen Arguments darstellt. Diese Überprüfungen wurden im Kapitel 6.7.3 erläutert. Die Erweiterung einer Struktur um eine Extension hat dabei Vorrang vor dem Generieren eines neuen Strukturkernes. Die Abbildungsdefinitionen für die Extensionen lauten wie folgt:

```
extension_map(Topic, brainstorm, argument, Structure, [view_rel(Connector, Condition, _)]) :-
        …
        retract( structure(Structure, + Model) ), *
        find_model_rel(Model, argument, model_rel( cause, Cluster2, _) ),
        adapt_cluster( (brainstorm, fact, Topic, default), Condition, Cluster1), *
        adapt_model(Model, + NewModel,
                        [  (brainstorm,   [model_rel( plus, Cluster1, Cluster2)]),
                           (argument,     [model_rel( support, Cluster1, Cluster2)]) ]),
        assert( structure(Topic, argument, Structure, NewModel)), !.


extension_map(Topic, brainstorm, argument, Structure, [view_rel(Connector, Condition, _)]) :-
        …
        retract( structure(Structure, + Model) ), *
        find_model_rel(Model, argument, model_rel( cause, Cluster2, Cluster3) ),
        adapt_cluster( (brainstorm, fact, Topic, default), Condition, Cluster1), *
        adapt_model(Model, + NewModel,
                        [  (brainstorm,   [model_rel( criterion, Cluster1, Cluster2)]),
                           (argument,     [model_rel( constrain, Cluster1, Cluster3)]) ]),
        assert( structure(Topic, argument, Structure, NewModel)), !.
```

Filtern von Tabellen aus der Explorationsperspektive

Die Pro-/Kontraperspektive stellt Informationen in einer einfachen Tabellenstruktur dar. Das Ziel besteht darin, auf die Weise zu erkennen, welche Aspekte noch ungenügend besprochen wurden. Die Informationen dafür entstehen in der Explorationsperspektive. Wird in der Explorationsperspektive irgendein Beitrag angegriffen („Attack“) oder unterstützt („Support“) wird daraus sofort gefolgert, dass dort eine weiterführende Einschätzung sinnvoll wäre und eine Tabellenstruktur angelegt. Im Gegensatz zur „Attack“-Relation kann die „Support“-Relation in dem beschriebenen Diskussionsmodell Bestandteil mehrerer Strukturen sein. Die selbe Relation könnte sowohl als Teil eines Arguments erkannt werden als auch in eine Tabelle übernommen werden.

Es wäre auch sinnvoll, Informationen aus der Brainstormingperspektive in die Tabellendarstellung zu überführen. Anbieten würden sich dafür die „Plus“- und „Minus“-Relationen. Das Diskussionsmodell wurde aber bisher noch nicht um diese Regeln erweitert.

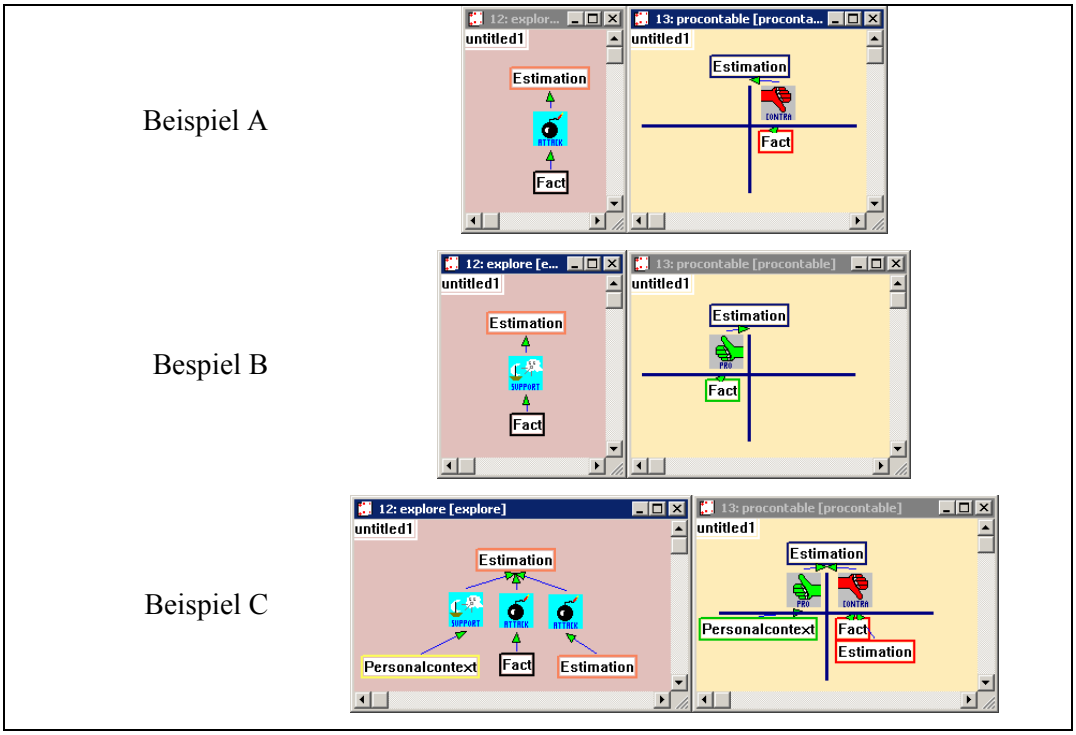


Abbildung 82

Schematische Darstellung der Abbildung der Strukturkerne aus der Explorationsperspektive in die Pro-/Kontraperspektive (Beispiel A und B). Abbildung von Erweiterungen (Beispie C).

Wie auch im vorangegangenen Beispiel gibt es zwei unterschiedliche Strukturen in der Explorationsperspektive, die zur Generierung einer initialen Tabelle führen. In Abbildung 82 sind diese Strukturkerne in Beispiel A und B dargestellt zusammen mit dem Resultat einer Abbildung in der Pro-/Kontraperspektive auf der rechten Seite. Anders als bei der

Argumentabbildung werden bei dieser Abbildung kaum Einschränkungen für die Typen der Inhaltskarten angegeben, das heißt, dass die Liste der zulässigen Typen recht lang ist.

```
core(explore, procon, [view_rel(Connector, Condition, Conclusion)]):-
        has_one(relation(   [support], Connector,
                            [fact,subject,idea,personalcontext,result,estimation], Condition,
                            [fact,subject,idea,personalcontext,result,estimation], Conclusion)).


core(explore, procon, [view_rel(Connector, Condition, Conclusion)]):-
        has_one(relation(   [attack], Connector,
                            [fact,subject,idea,personalcontext,result,estimation], Condition,
                            [fact,subject,idea,personalcontext,result,estimation], Conclusion)).
```

Wie schon für die Brainstormingperspektive erläutert, sind auch in diesem Fall die Extensionen strukturell identisch zu den Strukturkernen:

```
extension(explore, procon, _, view_rel(Connector, Condition1, Conclusion1)):-
        ...
        has_one(relation(       [attack], Connector,
                                [fact,subject,idea,personalcontext,result,estimation], Condition1,
                                [fact,subject,idea,personalcontext,result,estimation], Conclusion1)),
        core(explore, procon, [view_rel(_, Condition2, Conclusion1)]),
        ...

extension(explore, procon, _, view_rel(Connector, Condition1, Conclusion1)):-
        ...
        has_one(relation(       [support], Connector,
                                [fact,subject,idea,personalcontext,result,estimation], Condition1,
                                [fact,subject,idea,personalcontext,result,estimation], Conclusion1)),
        core(explore, procon, [view_rel(_, Condition2, Conclusion1)]),
        ...
```

Die Struktur, die in der Explorationsperspektive mit den Erweiterungen entsteht, ist sternförmig (vgl. Abbildung 82, Beispiel C). Der Mittelpunkt des Sternes bildet den Aspekt, der in der Tabellendarstellung die Überschrift bildet. Die Beiträge, die in der Explorationsperspektive über eine „Support“-Karte mit diesem Aspekt verbunden sind, werden in der Tabellendarstellung über eine „Proheader“-Karte mit der Tabellenüberschrift verbunden. Die Verbindungen über eine „Attack“-Karte werden in die „Conheader“-Relation abgebildet. Die Abbildung des Strukturkernes wird wie folgt angegeben:

```
structure_map(Topic, explore, procon, Structure, [view_rel(_, Condition, Conclusion)]):-
        …
        unique(+ Structure),
        adapt_cluster( (procontable, pro, Topic, default), Condition, + Cluster1), *
        adapt_cluster( (procontable, aspect, Topic, default), Conclusion, + Cluster2), *
        assert( structure(Topic, argument, Structure,
                        model( [   (explore,       [model_rel( support, Cluster1, Cluster2) ]),
                                   (procontable,  [model_rel( proheader, Cluster1, Cluster2)])]
        ))), !.

structure_map(Topic, explore, procon, Structure, [view_rel(_, Condition, Conclusion)]):-
        …
        unique(+ Structure),
        adapt_cluster( (procontable, con, Topic, default), Condition, + Cluster1), *
        adapt_cluster( (procontable, aspect, Topic, default), Conclusion, + Cluster2), *
        assert( structure(Topic, argument, Structure,
                        model( [   (explore,       [model_rel( attack, Cluster1, Cluster2) ]),
                                   (procontable,  [model_rel( conheader, Cluster1, Cluster2)])]
        ))), !.
```

Die neue Extension wird im Strukturmodell zu dem Cluster verbunden, in dem der Aspekt, also die Tabellenüberschrift, verwaltet wird (*Cluster3*).

```
extension_map(Topic, explore, procon, Structure, [view_rel(Connector, Condition, _)]) :-
        …
        retract( structure(Structure, + Model) ), *
        find_model_rel(Model, explore, model_rel( _, _, Cluster3) ),
        adapt_cluster( (explore, procon, Topic, default), Condition, Cluster1), *
        adapt_model(Model, + NewModel,
                        [ (explore,     [model_rel( attack, Cluster1, Cluster3)]),
                          (procon,      [model_rel( conheader, Cluster1, Cluster3)]) ]),
        assert( structure(Topic, argument, Structure, NewModel)), !.


extension_map(Topic, explore, procon, Structure, [view_rel(Connector, Condition, _)]) :-
        …
        retract( structure(Structure, + Model) ), *
        find_model_rel(Model, explore, model_rel( _, _, Cluster3) ),
        adapt_cluster( (explore, procon, Topic, default), Condition, Cluster1), *
        adapt_model(Model, + NewModel,
                        [ (explore,     [model_rel( support, Cluster1, Cluster3)]),
                          (procon,      [model_rel( proheader, Cluster1, Cluster3)]) ]),
        assert( structure(Topic, argument, Structure, NewModel)), !.
```

Die Darstellung einer Tabelle birgt ein besonderes Problem in dem implementierten Abbildungsansatz. Dieser geht immer von der expliziten Darstellung der Relationen aus, die im Modell verwaltet werden. Die Visualisierung der Relationen in einer Tabelle ist jedoch erst einmal nicht plausibel, da schon das Tabellenformat der Übersichtlichkeit dient und die expliziten Relationen ersetzt. Die Zugehörigkeit eines Beitrages zu einer Tabelle ist durch seine Positionierung optisch eindeutig bestimmt. Es wäre im Prinzip möglich, diese Nähe intern als Relation zu modellieren.

Was optisch leicht zu sehen ist, ist allerdings formal unter Umständen nur schwierig definierbar, insbesondere wenn die Tabellen sehr eng stehen und insbesondere dann, wenn gar nicht davon ausgegangen werden kann, dass Karten überhaupt immer korrekt und sinnvoll platziert sind. Karten können ja auch vorübergehend hierhin und dorthin geschoben werden, beispielsweise um Platz zu schaffen.

Für die Tabellendarstellung muss deshalb entschieden werden, ob die Beibehaltung der expliziten Relationen bei einer Abbildung von einem Workspace in einen anderen für diesen Fall aufgegeben werden soll. Dem widerspricht, dass die Tabellen nicht nur einmalig aus der Datenbasis generiert werden, sondern auch ergänzbar sein sollen. Wenn also eine Struktur in der Explorationsperspektive ergänzt wird, soll diese Ergänzung auch mit in die richtige Tabelle aufgenommen werden. Dazu muss aber bekannt sein zu welcher Tabelle und wo sich diese befindet. Diese Information ist ohne explizite Relationen aus der visuellen Darstellung kaum deduzierbar, denn es können kaum sinnvolle Annahmen darüber gemacht werden, wie „gut“ oder „schlecht“ die Karten durch Nutzerinnen und Nutzer platziert werden.

Die Lösung besteht in einem Trick. Die Zugehörigkeit der Beiträge zu den Tabellen wird auch in dieser Perspektive durch explizite Relationen ausgedrückt. Damit ist die Zugehörigkeit der Karten zu den Aspekten eindeutig. Auch die Positionierung ist dadurch frei gestaltbar, da die Zugehörigkeit über die Relationen wieder zugreifbar ist. Der Trick besteht darin, das alleine durch die Anordnung der Karten die Tabellen so aufgebaut werden, dass die Relationen kaum sichtbar sind und deshalb auch kaum stören. Dazu gehört, dass die Konnektorkarten, die die Überschriften der Tabellenspalten bilden, alle übereinandergelegt werden, so dass nur eine sichtbar ist. Die Beiträge werden eng aneinandergelegt, so dass die Linklinien verdeckt werden (Abbildung 84). In Abbildung 83 ist ein Beispiel für eine Explorationsperspektive dargestellt. Es dient als Grundlage für die generierten Tabellen in Abbildung 84.

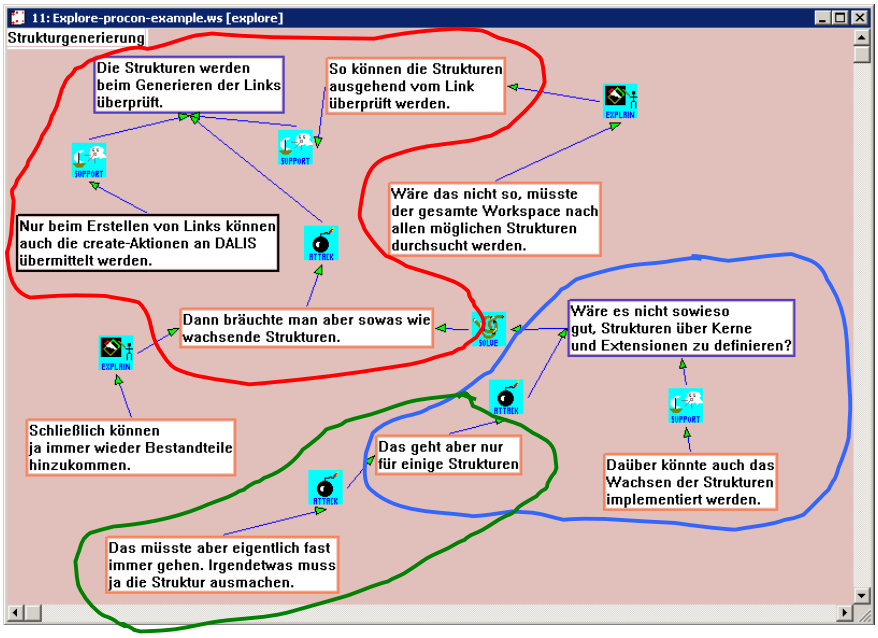


Abbildung 83

Der Explorations-Workspace stellt die Grundlage für die Tabellendarstellung in Abbildung 84 dar.

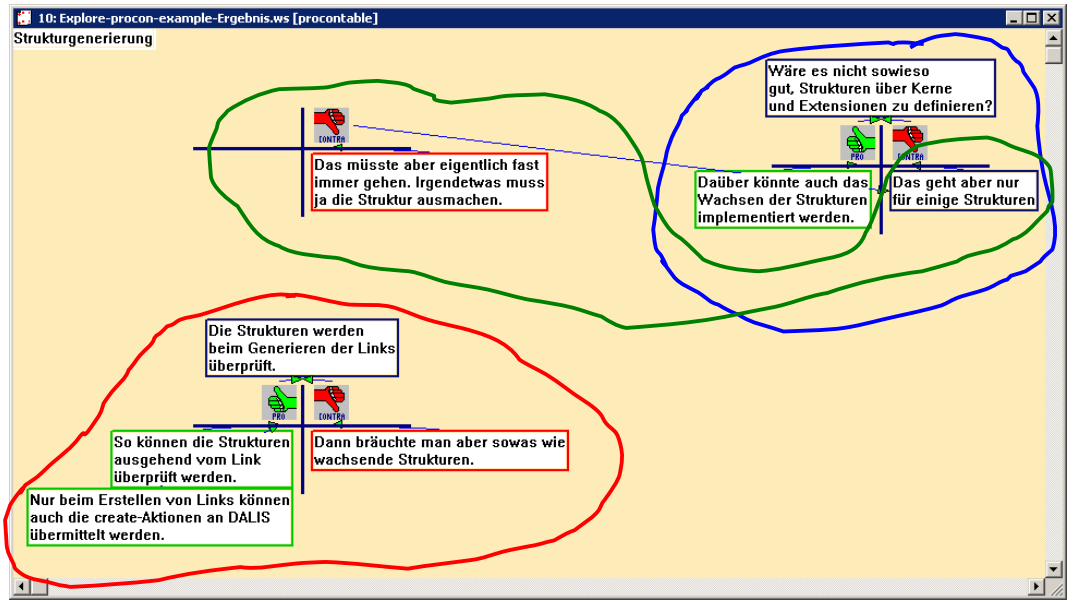


Abbildung 84

Beispiel einer Pro-/Kontradarstellung, die aus dem Explorations-Workspace in Abbildung 83 generiert wurde.

Die Tabellen, die aus den Strukturen in der Explorationsperspektive hervorgegangen sind, sind in beiden Abbildungen mit den gleichen farblichen Umrandungen markiert.

Aus der Darstellung in Abbildung 84 wird noch ein weiteres Problem bei der Generierung der Tabellen deutlich. In der Graph-Repräsentation eines Explorations-Workspaces

können die Karten mehrfach mit anderen Karten verbunden werden. In der Tabellendarstellung würde das dazu führen, dass eine Karte mehrfach dargestellt werden müsste, um z.B. ein Argument für mehrere Aspekte zu bilden. Diese Mehrfachdarstellung von Karten in einem Workspace ist aber nicht zugelassen, da sie dem implementierten Prinzip der Knotencluster widerspricht.
Auch für die Tabellen musste sichergestellt werden, dass jede Karte nur genau ein Mal im Arbeitsbereich vorliegt. Das wird durch die Sortierung so unterstützt, dass eine Karte wenn sie mehrfach referenziert wird, immer wieder an den „neuesten" Platz der Sortierung gelegt wird: sie wandert. An dieser Stelle bewährt sich die vorangegangene Entscheidung, die Relationen doch anzuzeigen. Diese können hier ausgenutzt werden, um Zugehörigkeiten von Karten tabellenübergreifend darzustellen. So ein Fall ist in Abbildung 84 für die Tabellenüberschrift zu erkennen (links oben), die gleichzeitig als Kontraargument in einer anderen Tabelle (rechts oben) aufgeführt ist.

6.9 Sortierungen

Die bisher beschriebenen Informationsfilter und Abbildungsvorschriften machen keine Angaben zum Layout der Darstellung. Für manche Darstellungen ist die Übersichtlichkeit aber notwendig, um überhaupt einen Vorteil aus der Visualisierung ziehen zu können. Dies betrifft alle jene Workspace-Typen, die von einer strukturierten Darstellung ausgehen, bzw. deren Strukturen einem bestimmten Muster folgen. Das betrifft die Argumentperspektive und die Pro-/Kontraperspektive.
Für die dort verwendeten Muster können keine allgemeinen Platzierungsalgorithmen verwendet werden, wie sie z.B. für die Darstellung von Bäumen, gleichmäßige Verteilungen oder möglichst wenige Überschneidungen von Kanten, angewendet werden. Eine Platzierung der Karten muss sich an der Strukturidee der Perspektive orientieren und zusätzlich deren graphischen Mittel berücksichtigen. Beispielsweise werden die Tabellen auch über Inhaltskarten formatiert.
Für die einzelnen Perspektiven werden Layouts implementiert, in denen die Anordnung einzelner Strukturen definiert ist. Im Falle der Argumentperspektive sind das die einzelnen Argumente, bei der Pro-/Kontraperspektive die einzelnen Tabellen. Zusätzlich wurde ein Layout implementiert, das die Verteilung der einzelnen Strukturen im Workspace realisiert. Es gibt also zwei Arten von Layouts:

1. Layouts für die jeweilige Perspektive, die die relative Platzierung der Karten der einzelnen Strukturen berechen.
2. Layouts, die die absolute Platzierung der Strukturen im Workspace berechnen.

Um die Perspektivenlayouts jeweils kompatibel zu dem allgemeineren Layouts realisieren zu können, implementieren sie eine an Java-Interfaces angelehnte Gruppe von Prolog-Prädikaten.

```
init(_, _, + language)
get_structures(_, Structures, + language)
check_size_optimum(Structures, + language, + Wgrid, + Hgrid)
check_size_heuristical(Structures, + language, + Wgrid, + Hgrid)
check_size_predefined(_, + language, + wgrid, + hgrid)
print_grid(_, _, + language, Xc, Yc, Structure)
```

Angeboten wird die Prozedur *init*, die den Workspace in einen Zustand überführt, in dem die darauffolgende Sortierung durchgeführt werden kann. Darunter fällt das Löschen von Karten, die alleine zur Formatierung dienen, wie die senkrechten und vertikalen Linien der Tabellen. *get_structures* liefert die vorhandenen Einzelstrukturen in Listenform.
Die drei alternativen *check_size*-Prädikate berechnen die Fläche, die zur Darstellung der Strukturen notwendig sind. In einem Fall berechnen sie eine optimierte Darstellung (*check_size_optimal*), in der Abstände weder zu groß, noch zu klein werden sollten, und sich die Karten nicht überdecken. Dafür müssen alle Strukturen untersucht werden. In der heuristischen Herangehensweise (*check_size_heuristical*) wird eine Struktur zufällig herausgegriffen und für diese die benötigte Fläche ermittelt. Diese Sortierung kann dazu führen, dass die Strukturen dann ihre zugeordnete Fläche überschreiten. *check_size_predefined* bietet eine feste Wertvergabe ohne Berechnungen, die in den meisten Fällen nicht befriedigend sein wird aber durchaus eine Grundstruktur vorgeben kann. Die zwei letzten Varianten ermöglichen es, den Aufwand für die Sortierung zu verringern, der im Allgemeinen recht hoch ist. Alle drei *check_size*-Prädikate liefern als Rückgabewerte die Dimensionen einer Fläche, die im übergeordneten Layout verwendet werden, um das Sortierungsgitter zu bestimmen.
print_grid platziert nach der Berechnung des Gitters die einzelnen Argumente und Tabellen in die Gitterfelder.
Angesteuert werden diese unterschiedlichen Sortierungen durch Karten der Funktionspalette (Tabelle 17).

	check_size_optimum
	check_size_heuristical
	check_size_predefined

Tabelle 17

Drei Funktionskarten leiten unterschiedliche Sortierungsgenauigkeiten ein.

Die Sortierungsprädikate des übergeordneten, perspektivenunabhängigen Layouts lösen zuerst die Initialisierung des zu sortierenden Workspaces aus und fragen dann die zu platzierenden Strukturen an. Für diese wird die erforderliche Größe bestimmt und auf dieser Basis das Gitterlayout berechnet. Dies bezieht zusätzlich noch definierte Abstände zwischen den Strukturen ein und die mögliche oder definierte Anzahl von Reihen und Spalten.

```
sort_o/h/p(_, _, - Language) :-
        init (_, _, - Language),
        get_structures (_,+ Structures, - Language),
        check_size_o/h/p(- Structures, - Language, + Wgrid, + Hgrid),
        get_layout(+ Gridlayout, gridlayout(- Wgrid, -  Hgrid) ),
        …
        arrange(_, _, - Language, - Structures, - Gridlayout).
```

6.10 Zusammenfassung

Mit dem DiscBoard ist ein System entwickelt worden, das Perspektiven über das Medium visueller Sprachen zur Verfügung stellt. Diese Perspektiven wurden als epistemische Formen (Collins & Ferguson, 1993) eingeführt. Der Ansatz wurde in den Kontext von Mind Tools gestellt (Jonassen & Carr, 2000; Lajoie, 1993; Pea, 1985), mit denen Denkprozesse angeregt und die Wissenskonstruktion unterstützt werden soll. Es ist davon ausgegangen worden, dass für verschiedene Arbeitsziele, Aufgaben oder Probleme unterschiedliche Darstellungsmittel geeignet sind (vgl. Suthers, 1999A; 1999B; 1999C; 2003; Carlson, 1995; Cooke, 1994; Jonassen & Carr, 2000; Jüngst & Strittmatter, 1995; Streitz, Haake et al., 1998).

Um eine abgestimmte Arbeit mit den Perspektiven zu ermöglichen, wurde eine Übernahme von Inhalten zwischen den Perspektiven definiert, die alleine auf die Objekttypen und strukturelle Informationen zugreift.

Durch diesen Ansatz werden im wesentlichen zwei Ziele verfolgt:

1. Die Beeinflussung der Externalisierung durch epistemische Formen.
2. Die Unterstützung eines Diskussionsprozesses durch die Abbildungen zwischen den Perspektiven.

6.10.1 Die Perspektiven

Für die Diskussionsunterstützung wurden die folgenden visuellen Sprachen als Perspektiven umgesetzt:

- Die Diskussionsstrukturierung (Protokollsprache).
  Hauptsächlich für die Anfangsphase gedacht, zur Strukturierung der geplanten Inhalte, Sammeln von Besprechungspunkten, Fragen, Aspekten, Ideen, Zusammenhängen, für schnelle Notizen mit wenigen Beitragstypen. Zusätzlich für die Aufnahme von Resultaten als Ergebnis der Diskussion.
- Die Explorationssprache.
  Zur Vor- und Nachbereitung anderer Arbeitsphasen, zur Exploration von Beiträgen, Darstellung von Hintergründen, Zusammenhängen, Meinungen. Erwartet werden umfangreichere Diagramme.
- Die Brainstormingsprache.
  Es können UND/ODER-Graphen entwickelt werden. Ideen bzw. Fragen oder Probleme werden über Alternativen, Optionen, Vorschläge aufgeschlüsselt. Darauf bezieht sich auch der Name der visuellen Sprache. Sinnvoll ist es, diese Darstellung durch einen echten Brainstormingprozess einzuleiten, der vorgegebene Relationen erst einmal gar nicht verwendet. Die Brainstormingsprache ist weniger als Hilfe für das Brainstorming selbst zu sehen, als dass eine Möglichkeit geboten wird, die Ergebnisse nachfolgend zu strukturieren. Am Schluss können Entscheidungen und Präferenzen verdeutlicht werden.
- Die Argumentsprache.
  Darstellung von Argumenten angelehnt an die Argumentstruktur, wie sie in Toulmin (1958) vorgestellt wird. Argumente können frei erstellt werden. Automatisch werden Argumente aus der Brainstorming- sowie aus der Explorationsperspektive gefiltert. Als Arbeitshypothese wird den Darstellungen in der Argumentsprache ein dokumentarischer Wert zugeschrieben.
- Die Pro-/Kontrasprache.
  Darstellung einer einfachen Tabellenstruktur zur Auflistung von Pro- und Kontraaspekten. Es wird davon ausgegangen, dass unvollständige Tabellen eine

Reflexion motivieren. Auch für diese Sprache wird ein besonderer dokumentarischer Wert angenommen. Inhalte für die Tabellen werden automatisch aus der Explorationsperspektive gefiltert.

- Die „Topic Map".
  Übersicht über behandelte Themen, die unter den einzelnen Perspektiven angegeben werden können.

Im Sinne von Collins und Ferguson (1993) stellen die Sprachen die epistemischen Formen bereit, wie sie in Tabelle 18 aufgelistet sind.

Visuelle Sprache	**Epistemische Form**
Diskussionsstrukturierung (Protokollsprache)	Liste
Explorationssprache	Gerichteter Graph
Brainstormingsprache	UND/ODER-Graphen
Argumentsprache	Argumentstruktur nach Toulmin
Pro-/Kontrasprache	Tabelle
„Topic Map"	Gerichteter Graph und Hierarchie

Tabelle 18

Gegenüberstellung der verwendeten visuellen Sprachen zu den darin angestrebten epistemischen Formen („target epistemic form": Collins und Ferguson, 1993, S. 28).

6.10.2 Unterstützung eines Diskussionsprozesses

Ergänzend zu den Perspektiven, die für bestimmte Arbeitsphasen entwickelt wurden, wurde der Diskussionsprozesses unterstützt, indem der Übergang zwischen den Perspektiven erleichtert wurde. Dafür sind Regeln, die Abbildungsvorschriften, definiert worden.

Die Entwicklung der Abbildungen basierte auf den folgenden Hypothesen und Voraussetzungen:

- Geht man von epistemischen Formen aus und drückt diese mit visuellen Sprachen der hier verwendeten Art aus, so sollte es vielfach möglich sein, dafür Strukturmuster als Klasse von Graphen zu definieren. Als Arbeitshypothese wurde davon ausgegangen, dass es möglich ist, über strukturelle Information relevante Teile zu filtern.
- Durch generierte Strukturen, die bestimmten Anordnungen unterliegen, kann die Diskussion motiviert werden. Diese Motivation besteht darin,
  - dass unvollständige Strukturen offensichtlich werden,
  - dass Zusammenfassungen bestimmter Strukturen eine vergleichende Diskussion erleichtern,

  - dass die offensichtliche Verwendung bestimmter Strukturen dazu anregt, diese zu erlernen.
- Es kann und sollte keine eindeutige Reihenfolge definiert werden, in der zwischen diesen Perspektiven gewechselt wird. Deshalb müssen die Abbildungen zu jedem Zeitpunkt korrekt funktionieren, auch wenn Urbild und Bild frei veränderbar sind.
- Da die Perspektiven grundsätzlich für unterschiedliche Arbeitsphasen gedacht sind, ist davon auszugehen, dass die Inhalte in den Arbeitsbereichen nicht deckungsgleich sind, sondern sich ergänzen. Die Abbildung unterstützt deshalb nur den Übergang zwischen den Phasen und bietet einen Ausgangspunkt weiterer Arbeitsschritte. Für die auseinander hervorgegangenen Strukturen bedeutet das, dass sie einerseits abgeglichen werden müssen, solange noch eine Übereinstimmung vorliegt, dass sie ansonsten aber ohne Beschränkungen weiter entwickelt werden können.

Unter diesen Voraussetzungen wurden drei Aspekte realisiert:

- Die Verwaltung der Übereinstimmungen zwischen den Strukturen und Karten der Perspektiven.
- Der Vorgang der Abbildung.
- Die Abbildungsvorschriften.

Verwaltung der Übereinstimmungen. Knotencluster verwalten die Übereinstimmung von Karten zwischen den Perspektiven. Abbildung 85 stellt unten den Datensatz dar, der für ein Knotencluster generiert wird.

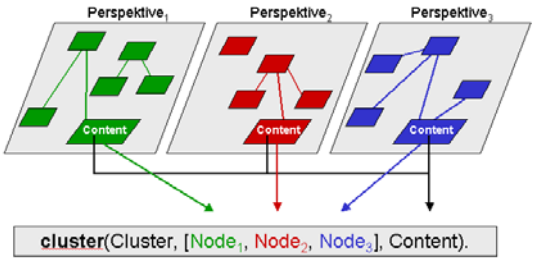


Abbildung 85

Informationen, die in einem Knotencluster verwaltet werden.

Wird eine Struktur abgebildet, so werden für die darin einbezogenen Karten Knotencluster und für die Strukturen Modelle generiert (siehe Abbildung 86). In den Strukturmodellen werden die Strukturen über deren Kantentypen beschrieben, die Knotencluster verbinden.

Eine Übereinstimmung von Strukturen oder Karten zwischen den Perspektiven besteht nach einer Abbildung solange, bis sie durch Nutzerinnen oder Nutzer explizit aufgehoben wird („delayed changes“).

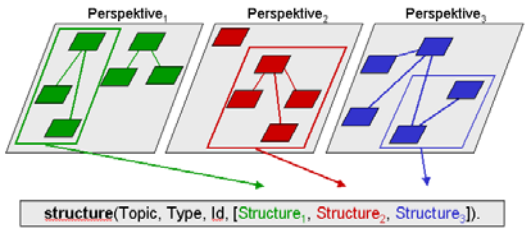


Abbildung 86

Informationen zu den Abhängigkeiten von Strukturen, die aus einer Abbildung hervorgegangen sind.

Abbildung. Eine Abbildung besteht jeweils aus

- einer Auswahl von Informationen, also einem Filter,
- dem Anlegen der Modell, potentiell einschließlich einer Interpretation der Strukturen und Kartentypen
- sowie einer sortierten Darstellung.

Die Filter beziehen sich ausschließlich auf Strukturen, dass sind Teilgraphen der Darstellungen mit den visuellen Sprachen. Sie reagieren dynamisch auf die Fertigstellung von Relationen in den Diagrammen, worauf jeweils untersucht wird, ob eine neue, relevante Struktur entstanden ist oder eine bestehende erweitert wurde. Für dieses Vorgehen werde die Strukturen intern definiert über:

- eine minimale Ausprägung (Kern)
- sowie mögliche Erweiterungen (Extensionen).

Geparst wird das Diagramm ausgehend von einer entstandenen Relation nur so lange, bis eine Neuentstehung oder Erweiterung erkannt ist, bzw. bis erkannt wird, dass dies nicht der Fall ist (lokales Parsen).

Interpretiert werden die gefilterten Strukturen insofern, als dass sie unter der neuen Perspektive im Normalfall anderen Kartentypen und anderen Relationstypen zugeordnet werden. Außerdem kann sich die Topologie der Strukturen ändern. Die Interpretation wird durch die Abbildungsvorschrift spezifiziert.

Abbildungsvorschriften. Die Abbildungsvorschriften für die Karten und Strukturen bilden das Diskussionsmodell. Vor der Abbildung einer Struktur, die zur Generierung des Strukturmodells führt, muss die Struktur, die das Urbild der Abbildung darstellt erkannt und klassifiziert worden sein. Das heißt der Typ der abzubildenden Struktur muss erkannt werden (z.B. Argument). In der Abbildungsvorschrift wird die resultierende Struktur definiert und generiert.

6.11 Diskussion

„Epistemic games", „Epistemic forms". Ein epistemisches Spiel wird nach Collins und Ferguson (1993) durch vier Charakteristika bestimmt: durch die Eingangsbedingung, durch die möglichen Aktionen, durch die Übergangsmöglichkeiten zu anderen epistemischen Spielen und die angestrebte Darstellungsform.

- Die Eingangsbedingung stellt eine Aussage darüber dar, wann ein bestimmtes Vorgehen (Spiel) gewählt werden sollte, also über die Art des Problems. Die Abbildung zwischen den Perspektiven kann als explizite Visualisierung einer Eingangsbedingung interpretiert werden: Liegen generierte Inhalte vor, die aber noch nicht weiter diskutiert wurden, sollte unter dieser Perspektive weiter gearbeitet werden.
- Die Abbildungsvorschriften definieren Übergangsbedingungen zwischen den epistemischen Spielen.
- Die angestrebte Form wird einerseits durch die Syntax der visuellen Sprache spezifiziert, andererseits soll sie durch die Bedeutung der verwendeten Symbole präzisiert werden.
- Die möglichen Aktionen während des Arbeitens unter einer Perspektive („inquiry") hängen spezifisch mit dem konkreten „epistemic game" zusammen. Einfache Aktionen, die beispielsweise typisch für die Arbeit mit Listen sind (Collins & Ferguson, 1993), wie das Hinzufügen von Einträgen, die Kombination von Einträgen, die Aufteilung von Einträgen und das Löschen von Einträgen zielen auf die Restrukturierung und werden generell durch die visuellen Sprachen gut unterstützt.

Divergierende Perspektiven. Grundlegend für einen Lernprozess sind nach Collins und Ferguson (1993) Änderungen der Fragestellung. Sie kennzeichnen, dass ein Problem anders verstanden wird und andere Aspekte wichtig werden. Auch diese Art von Aktionen sind mit dem DiscBoard möglich. Diese inhaltlichen Änderungen von Karten, führen im vorliegenden Ansatz zum Divergieren der Inhalte unter den verschiedenen Perspektiven.

Werden solche Änderungen als Indiz für einen Lernprozess gesehen, so bestätigt das nach meiner Ansicht den Ansatz divergierender Perspektiven, denn einerseits muss die Änderung möglich sein, andererseits kann nicht davon ausgegangen werden, dass sich die Frage unter einer anderen Perspektive gleichermaßen entwickelt. Es ist deshalb sinnvoll, diese Entscheidung den Nutzerinnen und Nutzern zu überlassen, was durch das Konzept der „delayed changes" unterstützt wird. In diesem Sinn können die Markierungen, die

eine Änderung in einem korrespondierenden Workspace anzeigen, auch als zusätzlicher Anreiz zur Diskussion bzw. Reflexion gewertet werden.

Für das DiscBoard muss allerdings kritisiert werden, dass es den Vergleich zwischen einem alten und dem neuen Inhalt nicht unterstützt. Beide Bestandteile müssten parallel betrachtet werden können, um sie gegeneinander abzuwägen. Erst dann wäre eine geeignete Reflexion möglich.

Das Interface. Die Kommunikation zwischen den Teilsystemen C++-MatchMaker, Dalis und DiscBoard ist vereinheitlicht worden (vgl. Mühlenbrock, 2001) und verwendet die drei Sprachpimitive create, modify und delete, um Änderungen in den Arbeitsbereichen zu übermitteln.

Dieses Interfaces stellte sich als relativ unflexibel heraus, wenn zusätzliche Informationen zu denen versendet werden sollen, die in den Parametern vorgesehen sind. Im Rahmen der Überlegungen, ob eine rekursive Gestaltung der Workspace-Struktur ermöglicht werden sollten, war beispielsweise die Übermittlung der „Parent"-Information nicht unmittelbar möglich.

Weiterhin bezieht sich die Verwendungsidee der Sprachprimitive auf die Karten in den Workspaces. Andere Ereignisse, wie Menü-Auswahlen, können damit generell nicht übermittelt werden. Wenn, wie es in der Diskussionsunterstützung durchgeführt wurde, ein Modell in Dalis hinterlegt wird, das auf Ereignisse reagieren soll, so müssen diese immer auf Karteninteraktionen abgebildet werden. Das führte im DiscBoard zur Entwicklung der Funktionspaletten. Dieses Vorgehen ist für Nutzerinnen und Nutzer ungewohnt. Außerdem ist es aufwändig in der Programmierung. Komplexere Interaktionen bzw. Dialogstrukturen sind damit nur umständlich zu implementieren.

Auch der Zustand der Applikation, wie Variablenbelegungen, kann nur schwer als Kontext für die Workspace-Interaktion mit in Betracht gezogen werden.

Wird Dalis ganz allgemein für die Verwaltung einer Datenbasis verwendet, wäre ein gefächertes Spektrum für Anfragen sinnvoll.

Parsing. Für die Abbildungen wurde der Ansatz des lokalen Parsens gewählt. Dafür wird jeweils ein Inkrement in einem Diagramm daraufhin untersucht, ob es eine bestehende Struktur erweitert oder die Grundlage für eine neue bildet. Dieses Erkennen von Strukturen wird dadurch vereinfacht, dass die Gruppe der erkennbaren Strukturen eingeschränkt wird, insbesondere werden keine rekursiven Strukturdefinitionen unterstützt. Dadurch wird auch ein Problem im Umgang mit Zyklen vermieden. Die Definitionen für die Syntax bedeutungstragender Strukturen ist zweischrittig in die Erkennung eines Kernes und einer Extension untergliedert. Diese Bestandteile können

selbst jedoch umfangreicher sein. Ohne solche Einschränkungen wäre das Parsen von Graphen oft zu aufwändig, um es in einer sinnvollen Zeit durchzuführen. Die Einschränkungen für die definierbaren Strukturen werden speziell für den Kontext der Diskussionsunterstützung als gering eingeschätzt, da sich bisher kein Bedarf ergeben hat, komplexeren Strukturen eine Bedeutung zuzuweisen. Das inkrementelle Vorgehen bei den Abbildungen in Verbindung mit der Verwaltung der Strukturmodelle hat sich als gut durchführbar erwiesen.

7 Fallstudie II: Wissenselaboration durch Dokumentationsprozesse: Das System FreeStyler

Der FreeStyler (Hoppe & Gaßner, 2002) wurde ursprünglich dazu entwickelt, organisationales Lernen zu unterstützen. Angestrebt wurde eine „low scale"-Lösung, die möglichst keine umfangreichen Anschaffungen, Prozessänderungen und Technologien notwendig machte, denn es war davon auszugehen, dass eine Investition in technologisch ausgerichtete Lösungen die Zielgruppe finanziell und in Bezug auf die Wartung überfordern würde. Die Anschaffung großer Systeme wie LotusNotes oder SAP erschien außerdem erheblich überdimensioniert für die erwarteten zu unterstützenden Prozesse.

Das Ziel bestand nicht darin, die Arbeitsprozesse umfangreich zu verändern oder sogar durch die Tools zu organisieren. Wie bei anderen parallelen Entwicklungen (z.B. Hoppe, Lingnau et al., 2000; [nimis]) sollte die Technologie nicht im Vordergrund stehen, das System leicht zu bedienen sein und eingeübte Prozesse nicht behindert werden.

Als Vorausschau auf die nächsten Abschnitte folgt eine kurze Beschreibung der zentralen Ziele, die mit der Entwicklung des FreeStylers vorfolgt wurden:

Dokumentationsprozess. Der FreeStyler wurde entwickelt, um einerseits komfortable Visualisierungen in „face-to-face"-Diskussionen zu ermöglichen, andererseits die daraus entstandenen Ergebnisse direkt in einen Dokumentationsprozess einfließen zu lassen. Der FreeStyler dient deshalb ganz besonders als Hilfsmittel, um von gesprochener Sprache, über eine zuerst informelle Darstellung, zu einer Repräsentation mit formalen Bestandteilen zu gelangen. Eine wichtige Bedingung für die Dokumentation bestand darin, dass externe Materialen mit einbezogen werden können, um die Verwaltung von Fachartikeln, Briefwechseln, etc. zu ermöglichen.

„Re-Use" durch Varianz der Nutzungsszenarien. Entwickelt wurde der FreeStyler zur Begleitung eines Arbeitsprozesses, der zu großen Teilen aus

- Telefongesprächen,
- Sitzungen,
- Consulting und
- Dokumentation

besteht. In diesem Kontext kann er sowohl zur Vorbereitung wie Nachbereitung von Sitzungen und Präsentationen verwendet werden. Aus der Nutzungsvarianz soll sich implizit eine Wiederverwendung von Dokumenten ergeben. Für so einen Ansatz eines „knowledge-building process", der die Argumentationen, Begründungen und die Bildung von Zusammenhängen einbezieht, spricht auch Stahl (2000)

Visualisierung und Strukturierung. Der FreeStyler soll die Möglichkeit bieten, Inhalte sehr frei zu modellieren. FreeStyler verbindet interne und externe Links mit einer seitenorientierten Dokumentstruktur, die besonders für Dokumentationen von Themenbereichen als geeignet erachtet wurde. Das bedeutet, dass mit dem FreeStyler ein Hypertextsystem zur Verfügung gestellt wird, mit dem sogar getypte Referenzen durch die User definierbar sind. Durch die Kombination mit der Seitenstruktur können gleichzeitig stärker in sich geschlossene, trotzdem strukturierbare Kontexte dargestellt werden und dies mit den Eigenschaften eines Hypertextsystem verbunden werden.

In der neueren Version können außerdem Paletten verwendet werden, die jeweils visuelle Sprachen anbieten. Darin enthalten sind in sich abgestimmte Objektmengen, die für konkrete Anwendungen entwickelt sind.

„Group Memory". Angestrebt wurde ein Tool, mit dem ein „group memory" inkrementell aufgebaut werden kann. Da mehrere Personen gemeinsam die gleichen Dokumente nutzen, bildet der Dokumentenbestand ein „corporate memory". Durch die gemeinsame Nutzung kann sich verstärkt auch eine gemeinsame Sprache entwickeln, Ideen und Beispiele stehen allen zur Verfügung.

Retrieval und „Re-Use". Ein großer Nachteil marktüblicher Kreativtools ist ihr sehr spezifisches Ablageformat und ihre ungenügende Integrierbarkeit in andere Medien oder Systeme. Das liegt u.a. daran, dass deren Speicherformate mit allgemein verwendeten Suchmaschinen oft nicht indexiert werden und deshalb kaum retrievalfähig sind. Durch die Ablage der FreeStyler-Dateien im XML-Format wird erreicht, dass ein Großteil der Inhalte mit handelsüblichen Suchmaschinen durchsucht werden kann und damit retrievalfähig ist. Das bedeutet auch, dass der FreeStyler auch in Systemarchitekturen einbezogen werden kann, die den FreeStyler nur als Komponente verwenden. Für ein Szenario, in dem der FreeStyler auch zur Organisation der Datenbestände eingesetzt wird, stellt er zusätzlich Möglichkeiten der Verschlagwortung zur Verfügung und integriert selbst eine Suchmaschine für die Indexierung von Datenbeständen. Ergebnisse von Suchen auf den Schlagworten oder dem Index werden dann im FreeStyler selbst dargestellt und zwar u.a. als Links auf die referenzierten Dokumente.

Mit dieser Ausrichtung werden Probleme ins Zentrum gestellt, die auch in Slavin et al. (2001) als zentrale Herausforderungen beim Wissensmanagement angesehen werden:

- Verbesserung der Kommunikation über unstrukturierte Probleme
- Aufnahme der Informationen parallel zu Arbeitsprozessen („real time capture“)
- Integration hybrider Materialien
- Wiederverwendbarkeit über ein „group memory“
- Formattransformationen

7.1 Wissensmanagement: Das Projekt Trendmonitoring

In Zusammenarbeit der „Transferstelle Hochschule-Praxis“ (*THP*), dem Fachgebiet „Kooperative und lernunterstützende Systeme“ (*Collide*) der Universität Duisburg sowie der „Mannesmann Pilot Entwicklung“ (*mpe*) wurde aus dem Projekt „Verbesserung des Innovationstransfers durch die Optimierung des transferorientierten Informations- und Wissensmanagements“ ein Ansatz zum Wissensmanagement für die *THP* umgesetzt.
Dokumentation, Suche und Wiederverwendung des „Know What“ waren Gegenstand dieses Projektes und sollten in verschiedene Arbeitsabläufe der *THP* integriert werden. Der Schwerpunkt lag bei der Erstellung von Informationsnetzen unter Einbindung von Kreativtechnologien.
Als Projektpartner hatte *mpe* ihre bisher erprobten Ansätze zum Trendmonitoring in das Projekt eingebracht. Das von ihr praktizierte Innovationsmanagement beruhte u. a. auf einem Trendmonitoring, das eine tiefgehende Beobachtung technischer Entwicklungen über Fachliteratur sowie eine Analyse der Marktsituation in Bezug auf Produkttrends ermöglichte.
Wie auch die *mpe* stellt die Transferstelle einen „Gatekeeper“ relevanter Informationen dar und fungiert als „Trend-Scout“ auf dem Gebiet technologischer und wirtschaftlicher Entwicklungen. Beide sind auf der Suche nach Partnern für ihre Organisationen, und beide sind bemüht, konkrete Projekte mit solchen Partnern zu initiieren.
Im Rahmen des Projektes war es eine entscheidende Aufgabe, die aktuelle Organisation der Transferstelle Hochschule Praxis (*THP*) zu analysieren. Der Schwerpunkt dieser Analyse bestand einerseits in der momentanen Organisation des Wissensmanagements und andererseits in der Frage, wie Trendmonitoring im gegebenen Kontext transparent praktiziert werden kann. Im Zentrum stand dabei die technische Unterstützung der bisherigen und geplanten Abläufe. Das angestrebte Trendmonitoring umfasste im Wesentlichen zwei Schwerpunktprobleme:

- Wie kann es geschafft werden, die alltäglich anfallenden Informationen so festzuhalten und operationalisierbar zu gestalten, dass aus ihnen Trends reflektiert werden können?
- Welche Schnittstellen können geschaffen werden, damit bedarfsgerechte Informationen zu Trends kundengerecht weitergegeben werden können?

7.1.1 Ansatz

Es wurde ein Ansatz zum Wissensmanagement verfolgt, der die alltäglichen Arbeiten wie das Anfertigen von Notizen, Visualisierungen und Skizzen in den Mittelpunkt stellt. Die dabei entstehenden Repräsentationen setzen unterschiedlichste Medien in Form von Informationsobjekten zueinander in Verbindung. Informationen können auf diese Art und Weise integriert werden, was sonst über die Grenzen der einzelnen Medien hinaus selten unterstützt wird. Die Darstellungen können im Verlauf der anfallenden Arbeiten verfeinert und verändert werden, hauptsächlich werden sie wachsen. Daraus entstehen Trendkontexte (Kapitel 7.1.2).

Das Externalisieren von komplexen Wissenszusammenhängen ist generell äußerst schwierig und stellt sich in den meisten Wissensmanagementprojekten als Flaschenhals dar. Das Problem, wie man Personen motiviert, Wissen zu externalisieren und welches eine geeignete Ebene dafür ist, hat bis heute keine generelle Antwort erhalten und bleibt vermutlich prinzipiell an die konkreten Arbeitszusammenhänge gebunden.

Im Projekt vertraten wir die Meinung, dass die Externalisierung sinnvoll an Kommunikationsprozesse gekoppelt werden kann. Dass es sich bei diesen Externalisierungen nicht um formale Darstellungen handeln muss, sondern um semi-formale oder auch digitale handschriftliche Notizen und Skizzen, war in diesem Projekt selbstverständlich. Formalisierungen sind einerseits zu zeitaufwendig, andererseits war auch die Art des Wissens, das gehandhabt werden sollte, gar nicht dazu geeignet.

Werden dagegen die Notizen, die sowieso während eines Treffens oder bei einem Telefongespräch anfallen, gleich mit entsprechenden Tools und Devices angefertigt, also digitalisiert, so sind sie leichter wiederzuverwenden. Mit dem entwickelten Tool – FreeStyler – können bei Bedarf „Concept Maps“ angefertigt werden und „Mind Maps“. Argumentationen können dargelegt werden und vordefinierte Beitragskategorien verwendet werden. Durch die zusätzliche Möglichkeit, handschriftliche Eingaben durchzuführen, kann auch der Grad an Formalisierung deutlich variiert werden.

Ein weiterer Schwerpunkt liegt beim FreeStyler auf Retrieval-Möglichkeiten und auch auf der Verbreitung von Informationen durch E-Mail-Optionen.

7.1.2 Trendkontexte

Um ein Trendmonitoring betreiben zu können, musste geklärt werden, was ein Trend im Arbeitskontext der *THP* ist. Ein Trend ist ganz allgemein eine Grundrichtung einer Entwicklung (Duden). Eine solche Entwicklung kann beispielsweise aus bestehenden Datensätzen nachgezeichnet werden (Z.B. kann anhand von Daten die Zunahme beim Verkauf von Mobiltelefonen in den 90iger Jahren dargestellt werden.). Dies setzt aber voraus, dass umfangreiche und bereits dafür strukturierte Datensätze bestehen.
Das setzt eine gezielte Datenerfassung voraus, in die wiederum die Hypothesen für Entwicklungen schon einfließen müssen. Aber gerade diese Hypothesen zu finden, ist Aufgabe der *THP*. Generell ist die Darstellung bereits bekannter Trends durch Entwicklungsdaten ein eher rückblickend analytischer Vorgang, der in der *THP* nicht angestrebt werden sollte.
Das Problem der Trenderkennung liegt für die *THP* eher darin, Trends zu formulieren. Diese Trends müssen sowohl in die Universität hinein- wie aus ihr hinausgetragen werden und in Übereinstimmung mit den bestehenden Nachfragen gebracht werden. D.h. man muss sich an existierenden Kunden und Forschungsthemen orientieren.
Ein Trend ergibt sich für die *THP* aus der Menge relevanter Anfragen: dem Bedarf. Als transferorientierte Stelle muss sie zu diesen Bedarfen konkrete Kontakte und Informationen aufbauen. Daraus ergeben sich vielfältige Zusammenhänge die zu einer verwobenen Struktur führen, dem Trendkontext. Eine solche Struktur wird im Rahmen des Projektes auch verkürzt als Trend bezeichnet.

7.1.3 Analyse der Arbeitsorganisation

Die Aufgaben der *THP*

In der *THP* werden drei zentrale Aufgabenbereiche verfolgt, die im Folgenden aufgegriffen und erläutert werden:

- Akquisition
- Beratung und Service
- Dokumentation

Akquisition

Die Akquisition beinhaltet die Projektpartner-Akquise und die -Vermittlung. Darin inbegriffen ist die Verbreitung der Informationen zum gesamten Serviceangebot der *THP* u.a. in Form von Öffentlichkeitsarbeit und z.B. Wirtschaftstreffs, die Ausrichtung und Begleitung von Seminaren, Workshops, Symposien, Messen und Anwenderforen.

Weiterhin wird begleitend zum täglichen Arbeitsprozess „Know How“ bzw. „Know What” akquiriert. Es werden Journale und Tageszeitungen rezipiert, zielgerichtet Hintergrundmaterial gesammelt für Gutachten, etc.

Beratung und Service

Die *THP* erbringt sowohl auf Anfrage als auch zum allgemeinen Service Auskünfte, Hilfestellungen (z.B. Existenzgründungen, Begleitung von Veranstaltungen), Analysen, Tests und Gutachten zu Themen, die sich im Bereich der Forschungskompetenz der Universität befinden. Dieser Service kann sowohl intern als auch extern angefragt werden. Veranstaltungen wie Forschungstage und Messen werden ausgerichtet oder begleitet.

Beratung und Service war der für das Projekt zentrale Aspekt. Beratungen folgen meistens einem vergleichbaren Ablauf. Dieser besteht aus einer Abfolge von Kontaktaufnahmen, die zur Materialaufbereitung und –nachbereitung führen und den eigentlichen Treffen und Absprachen. Die dabei anfallenden Materialien sollten eine Grundlage für die Beobachtung von Trends bilden.

Dokumentation

Die Dokumentation bildet die Grundlage sowohl für die Beratungen und Services als auch für die Akquisition. Sie bildet im Prinzip ein eigenständiges Aufgabengebiet, wenn sie bisher auch nicht als eigenständiger, festgelegter Vorgang in den Arbeitsablauf integriert ist.

Die Dokumentation gliedert sich in die Dokumentation von Kontakten, Absprache- bzw. Meeting-Ergebnissen und Recherchen. Letztere können ohne konkreten Beratungsbedarf durchgeführt werden, um die Entwicklung von Bereichen zu verfolgen und zu belegen.

Zu den Ergebnisdokumentationen gehört u.a. auch der Verlauf einer Beratung. Dokumentiert werden bisher eher statische Daten zu zwei Bereichen:

- *Die Universität und ihre Forschungsschwerpunkte.* Fachbereiche und Fachgebiete, Grundlagenforschung, Industrienahe Forschung, Entwicklungskapazitäten, Dienstleistungsangebote, Verwertungsangebote
- *Firmen.* Ansprechpartner, Kernkompetenzen, Forschungs- und Entwicklungsaktivitäten, Brancheninformationen, Entwicklungstendenzen

Zentraler Arbeitsablauf

Im Bereich Beratung und Service, inklusive der Dokumentation, sahen wir im Gegensatz zur Akquise eindeutige Unterstützungsmöglichkeiten und Bedarf.

Es zeigte sich, dass Beratung und Service in den meisten Fällen einem generellen Ablaufschema folgen. Dieses besteht prinzipiell aus einer Abfolge von Kontaktufnahmen, die zur Materialaufbereitung für weitere Treffen führt (Abbildung 87).

Wir gehen davon aus, dass ein Meeting einen zentralen Stellenwert beim Herausarbeiten von Trends hat. Durch den Austausch zwischen den Personen werden oft kreative Prozesse angestoßen, Ideen entwickelt und Entscheidungen getroffen, die relevante Beiträge für die Trends sein können. Insbesondere sind kurzfristig wichtige Notizen mit längerfristig interessanten Informationen verwoben, die nicht unbedingt voneinander getrennt werden sollen oder können. In den Kommunikationsprozessen kommen alle Möglichen Szenarien vor, wie „face-to-face“, Telefongespräche und E-Mail-Austausch. Treffen finden zwar auch in Räumen der *THP* statt, vermehrt aber an anderen Orten in der Universität oder bei Firmen, Messen, usw.

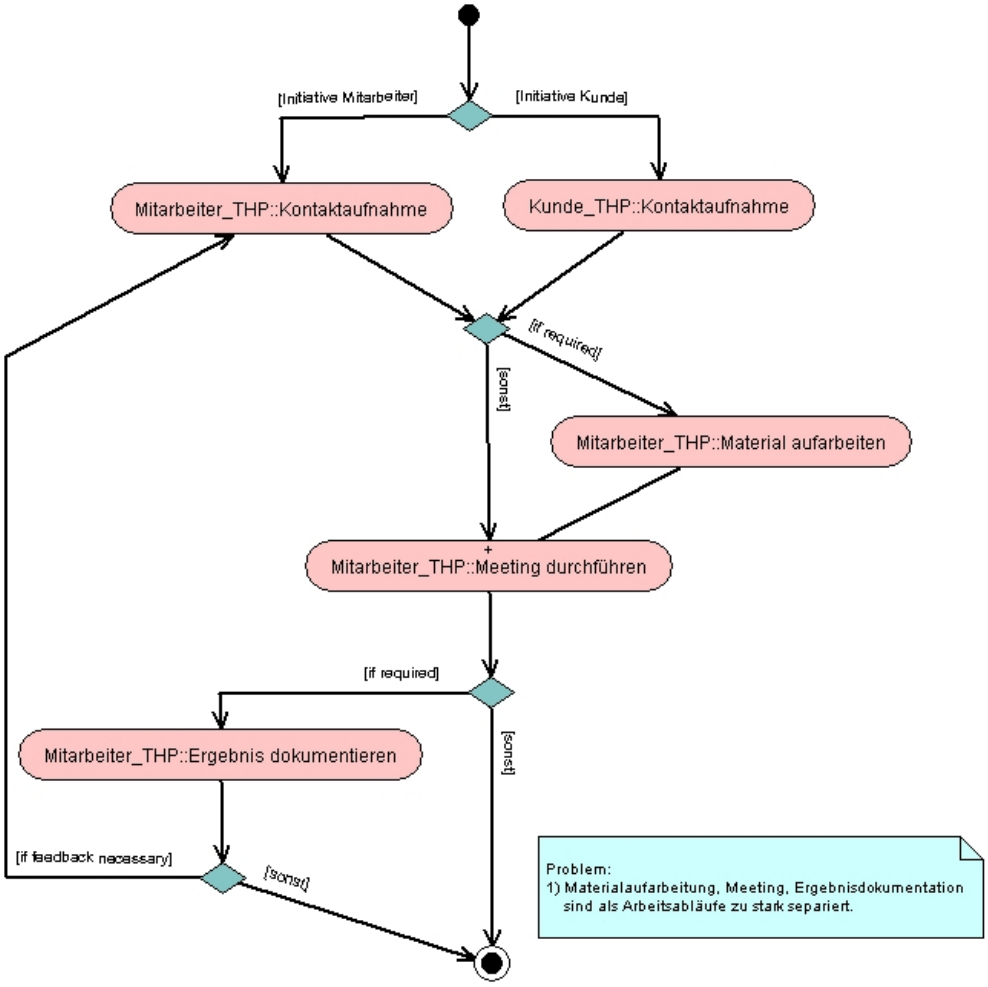


Abbildung 87

UML-Aktivitätsdiagramm zum Ablauf von Beratungen und Services.

Ein spezielles Problem von Treffen im Allgemeinen und Telefonmeetings ganz besonders besteht darin, dass viele Aktivitäten parallel verlaufen. Die Abbildung 88 stellt das Maximum solcher paralleler Aktivitäten bei einem Telefongespräch dar, einschließlich der verwendeten Tools, wie es bei der *THP* auftreten kann.

Beim einem Anruf wird typischerweise versucht, den bisherigen Vorgang abzurufen. Diese Vorgänge sind bisher, sofern überhaupt, in dem System Teamworks abgelegt. Prinzipiell werden zwei Arten von Notizen angelegt: Kundennotizen und inhaltliche Notizen. Für erstere wird ein vorgefertigtes Formular verwendet, das dann im Akten-ablagesystem verwaltet wird. Weitere Notizen werden entweder damit zusammen abgelegt oder auf zusätzlichen Zetteln.

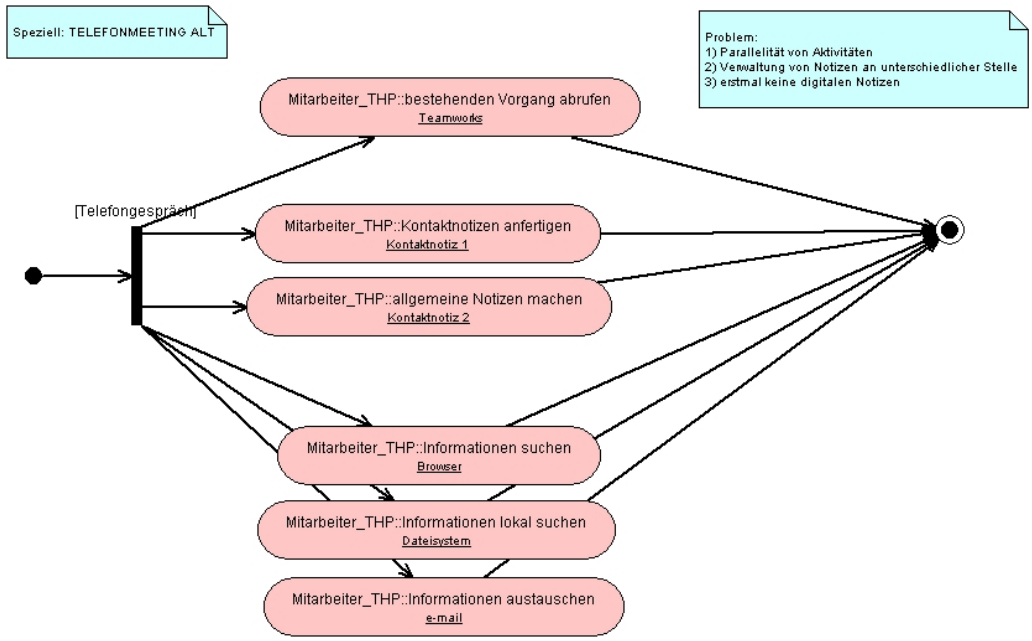


Abbildung 88

Das UML-Aktivitätsdiagramm zeigt die Aktionen, die u. U. während eines Telefonates verfolgt werden müssen.

Die Suche nach Informationen findet einerseits im Internet über einen Browser statt, andererseits über Bookmarks oder im Dateisystem. Dies ist jeweils stark an die individuellen Erinnerungen gebunden, um Suchkriterien zu definieren oder dezentral abgelegte Bookmarks zu verwenden. In vielen Fällen ist die E-Mail das wichtigste Mittel beim Transfer der Inhalte.

Suche und Dokumentation

Um Material für Kundenkontakte aufzuarbeiten, muss einerseits recherchiert werden und müssen andererseits, bei Bedarf, die recherchierten Ergebnisse gebündelt und präsentiert

werden. Bestandteile der Materialaufarbeitung sind also einerseits die Suche und andererseits die Bündelung der gefundenen Informationen.

Aktuell basiert die Recherche und Dokumentation auf den folgenden Ablagestrukturen:

- Teamworks: ein System zur Verwaltung von Kalenderdaten, Projektablauf-Informationen und assoziierten Inhalten. Dieses System wird im Schwerpunkt zur Abstimmung von Terminen verwendet und zur Verwaltung von Kundendaten. Die Möglichkeit, darüber auch Dokumente zu verwalten, wird kaum genutzt.
- Dateistruktur: Im Wesentlichen besteht eine individuelle Ablage von Dokumenten.
- Klassische Akten: Darin werden handgeschrieben Kontaktnotizen und Dokumente abgelegt aber nicht bibliothekarisch verwaltet.

Erweiterte Recherchen beziehen selbstverständlich das Internet und beispielsweise klassische Bibliotheken mit ein. Aus der bestehenen Organisation ergeben sich die folgenden Schwierigkeiten:

- Zwischen den einzelnen Ablagestrukturen bestehen keine Referenzen. Auch in den elektronischen Fassungen sind die Daten nicht verschlagwortet. Zwischen Meeting, Recherche und Dokumentation kommt es zu Medienbrüchen, die besonders eine spätere Suche erschweren.
- Weiterhin existieren keine abgesprochenen Ablagerichtlinien, was dazu führt, dass letztlich jede Mitarbeiterin und jeder Mitarbeiter hauptsächlich auf die eigenen Daten zugreift. Es handelt sich deshalb um einen hochgradig internalisierten Wissensbestand. Die inhaltsbezogenen Synergieeffekte sind deshalb eher als gering einzuschätzen.
- Da die Recherchen selbst nicht unterstützt werden, kann es zu einem wiederholten hohen Aufwand bei der Suche nach Dokumenten kommen.
- Dadurch, dass Anfragen und Ergebnisse dazu nicht dokumentiert werden und damit nur umständlich reflektiert werden können, ist jegliche Auswertung von Entwicklungen schwierig. Der Kontext in dem Einzelinformationen auftreten wird so nicht dokumentiert und ist nur lückenhaft zwischen den Personen transportierbar.

Wissen

Dynamisches Wissen macht einen großen Bestandteil des Wissens aus, das durch die *THP* wiederum vermittelt werden muss (Abbildung 89). Die bedarfsorientierte, individuelle Beratung verlangt, dass kundenorientiert jeweils neue Wissenszusammenhänge herausgearbeitet werden. Dabei werden verschiedenste Medien eingesetzt und in einen Kontext gebracht. Integriert werden oft auch nicht-öffentliche Informationen, die

beispielsweise über persönliche Firmenkontakte bekannt wurden. Diese können zwar häufig informativ eingebracht werden, dürfen jedoch nicht allgemein zugänglich gemacht werden.

Für dieses spezielle Consulting ergibt sich, dass die Repräsentation der Ergebnisse kaum mit HTML-Seiten geleistet werden kann. Dies würde zu einem zu hohen Arbeitsaufwand bei der Repräsentation führen, da die relevanten Informationen sich häufig ändern, also sehr dynamisch sind. Außerdem sollen die Ergebnisse in den meisten Fällen gerade nicht direkt veröffentlicht werden. Auch die Expertise zur Erstellung solcher Pages ist im gegebenen Arbeitsumfeld nicht für jede Person voraussetzbar. Eine dezentrale Nutzung und Erweiterung muss aber für alle gewährleistet sein.

Das dargestellte Wissen muss außerdem für einen unkomplizierten Austausch geeignet sein und trotz der Dynamik der Daten müssen Hintergründe und Fakten in den Kommunikationsprozess aufgenommen werden können.

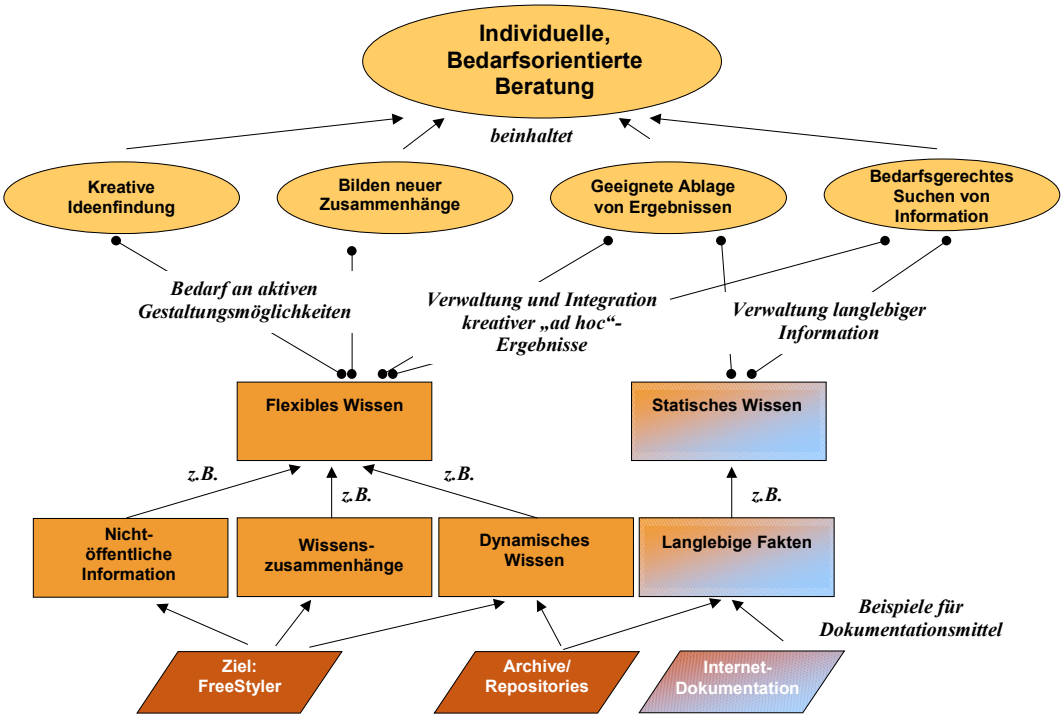


Abbildung 89

Flexibles Wissen statt statische Fakten: Relevantes Wissen für die Transferstelle der Universität Duisburg.

Zusammenfassung

Organisation

- Der Vorgang der Dokumentation ist zwar zentraler Bestandteil der Arbeit, findet aber in den Arbeitsabläufen als eigenständiges Arbeitspaket kaum Niederschlag.

- Wissenskommunikation ist nicht in die Arbeitsabläufe integriert.

Wissenskommunikation

- Es entstehen geringe Synergieeffekte bei der inhaltlichen Dokumentation, da diese stark individuell ausgerichtet ist und auch die Arbeitsthemen individuell organisiert werden.

Externalisierung

- Die Dokumentation ist zwischen den Mitarbeiterinnen und Mitarbeitern inhaltlich kaum abgestimmt.
- Reflexion und Strukturierung der Kundenbedarfe ist auf Basis der bestehenden Dokumentationsstrukturen schlecht durchführbar.
- Daraus ergibt sich ein hochgradig internalisierter Wissensbestand und individuelles Wissen und Annahmen über Trends.

Wissensstruktur

- Für Kontakte werden teilweise Verweise zu Dokumenten verwaltet, die in diesem Kontext wichtig waren.
- Außer auf der Ebene grob thematisch unterteilter Akten gibt es keine inhaltliche Bündelung oder Referenzierung zwischen Materialien.

Wissensressourcen

- Bisher anzufertigende Kontaktnotizen in nicht-digitaler Form werden von den meisten Mitarbeitern und Mitarbeiterinnen wegen des davon erwarteten geringen Nutzens nicht angefertigt.
- Nicht-digitalisierte Dokumente (z.B. Zeitschriftenartikel, Kontaktnotizen) werden nicht digitalisiert. Die Suche und Verbreitung der Texte stößt deshalb auf Hindernisse.
- Die Suchen in internen Archiven sind nicht unterstützt. Ablagestrukturen sind zu großen Teilen in Papierform.

7.1.4 Systemanforderungen

Der auch in diesem Fall diskussionsorientierte Ansatz geht davon aus, dass die Assoziationen und Begrifflichkeiten zu den Trends in den alltäglichen Gesprächen anfallen. Deshalb sollen mit einem Notiz-Tool kommunikative Situationen angereichert werden. Das Notiz-Tool soll sich dazu eignen, handschriftliche sowie strukturelle Information abzulegen.

Der vorgestellte Arbeitsablauf, der Meetings und Materialaufbereitungen beinhaltet, soll so unterstützt werden, dass die Etappen des Consultings begleitet werden (Abbildung 90).

Verfolgt wird der Ansatz, dass ein einzelnes Dokument pro Trend durch alle diese Phasen verwendet, erweitert und überarbeitet wird.

Kontaktaufnahme. Für die meistens telefonischen Kontaktaufnahmen muss es möglich sein, währenddessen Notizen und Kontaktinformationen anzufertigen. Verlangt wird eine „einhändige“ Eingabe, da keine „headsets“ eingesetzt werden.

Materialaufbereitung. Anschließend soll mit diesen Notizen eine strukturierte Vorbereitung für weiterführende Treffen möglich sein. Dazu gehören Verweise auf zusätzliche Informationen und archivierte Materialien. Auch die Restrukturierung der vorangegangen Notizen ist dafür notwendig.

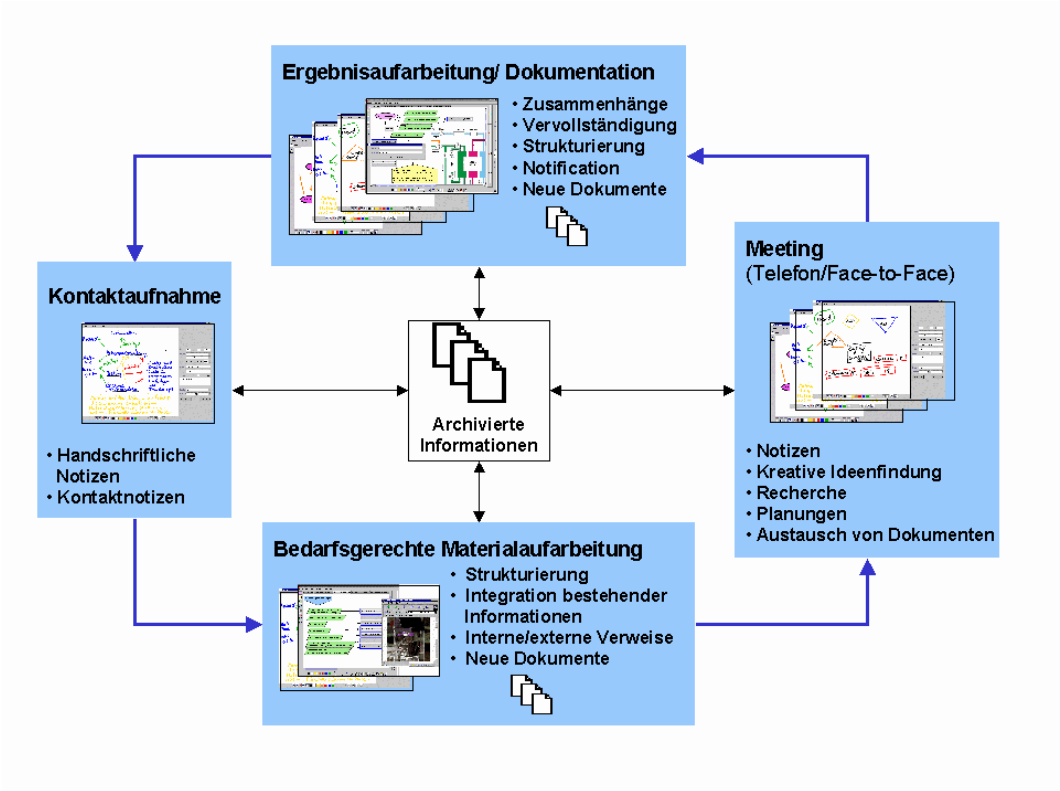


Abbildung 90

Angestrebter Dokumentenfluss im Projekt „Trendmonitoring“.

Sitzung. Während eines „face-to-face“-Treffens soll das Tool zur Visualisierung verwendet werden. Generell dient es zur Externalisierung und Strukturierung von Informationen. In der Sitzung sind Skizzen und Notizen ein besonderes Hilfsmittel, um Planungen abzustimmen und den Kontext zu fixieren. Unmittelbare Recherchen nach geeigneten Materialien können den Informationsaustausch unterstützen.

Ergebnisaufbereitung/Dokumentation. Die Nachbereitung einer Sitzung dient in erster Linie dazu, sich die Planungen in Erinnerung zu rufen und anstehende Aufgaben zu bearbeiten. Diese Phase kann auch für eine detailliertere Vernetzung der Inhalte genutzt

werden. Involvierte Personen müssen benachrichtigt werden. Gerade dazu kann es notwendig sein, einerseits die Darstellung selbsterklärend aufzubereiten, andererseits Teilaspekte aus den Visualisierungen herauszugreifen.

Nach der Klassifikation von WIPER ([wiper]) ist die Methode des Wissensmanagements hier sowol als aktivierungs- als auch als transferorientiert einzuschätzen. Ersteres zielt auf ein aktivierendes Ideenmanagement z.B. zur Offenlegung und Dokumentation von Wissen, letzteres auf rasche Verbreitung neuer adäquater Wissensbestände. Das neue Wissen der Transferstelle soll in der Verknüpfung einzelner Informationseinheiten bestehen. Außerdem wird die Digitalisierung und der Zugriff (Recherche, Integration, weniger Medienbrüche) unterstützt.

Basierend auf der oben referierten Analyse ([wiper]) wurden an das unterstützende System die folgenden Anforderungen gestellt:

- Externalisierung von Wissen
  - Ideen, Entscheidungen, Begriffe, Informationen sollen Informationsnetze bilden.
  - Digitale, handschriftliche Notizen, die auch in Sitzungen erstellt werden können, sind Elemente der Informationsnetze.
- Kommunikation von Wissen
  - Technisch angereicherte kooperative Sitzungen sollen möglich sein.
  - Visualisierung der Gesprächsinhalte.
  - Bedarfsorientierte Bündelung und (Re-)Strukturieren von Informationen.
- Weiterentwicklung der Wissensressourcen
  - Konstruktion von Informationsnetzen zur Visualisierung und Dokumentation.
  - Digitale Textarchive
  - Durchsuchbare & verschlagwortete Notizen.

7.1.5 Der FreeStyler Version 1

Interface

Die Oberfläche der Applikation (Abbildung 91) teilt sich in zwei Teile. Im linken Teil werden die konkreten Inhalte strukturiert, der rechte Teil dient der Verschlagwortung und Verwaltung. Die Verschlagwortung ist jedoch optional. Um Mappen zu erstellen stehen diverse Bedienungsfelder (Toolbars) zur Verfügung. Als Mappen werden die Dateien, die mit FreeStyler erstellt werden, bezeichnet. Die Mappen werden in XML-Dateien gespeichert. Alle Inhalte, Bilder, Handschrift und Einträge, werden zusammen verwaltet.

Mit den Buttons am linken Rand wird die visuelle Sprache zur Verfügung gestellt. Mit den damit erstellbaren Objekten können Einträge und Verbindungen erstellt werden. Die

untere Button-Leiste dient der Verwaltung der Freihandeingabe und der einzelnen Seiten. Dazu gehört die Farbgebung der Freihandeingabe und Löschmöglichkeiten. Unten befinden sich auch Möglichkeiten, um neue Seiten einzufügen, zu kopieren oder zu löschen. Jede einzelne Seite besteht aus vier Ebenen. Diese können einzeln ausgeblendet und in ihrer Reihenfolge verändert werden. In dem Leerraum in der Mitte werden die Inhalte entwickelt.

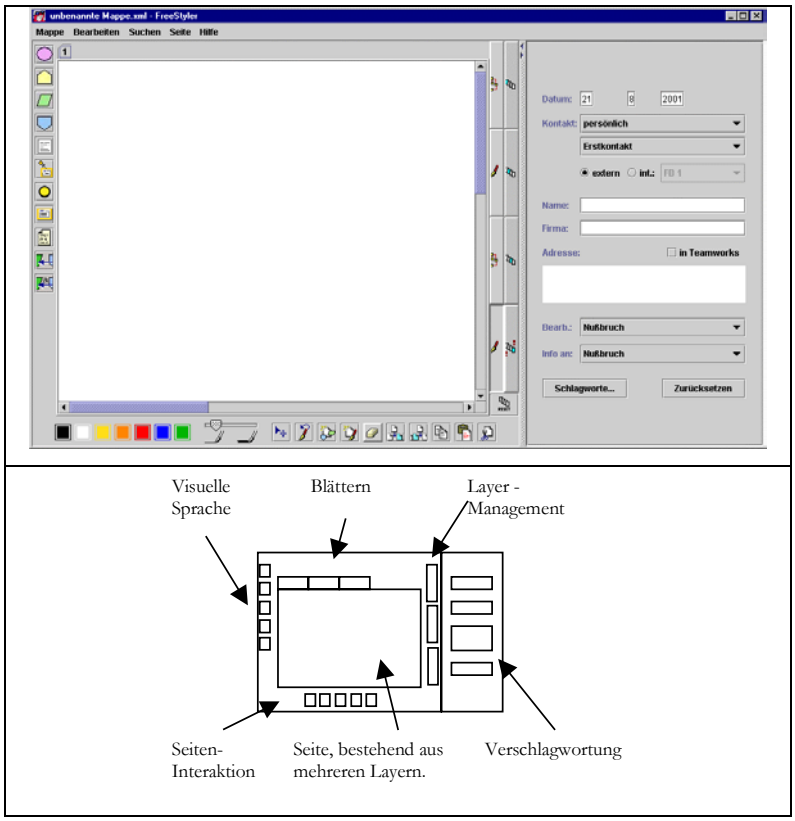


Abbildung 91

Das Interface des FreeStyler.

Auf der rechten Seite der Applikation existieren diverse Eingabemöglichkeiten für allgemeine Informationen zu der gerade erstellten Mappe. Es kann Name und Adresse einer Kontaktperson angegeben und der jeweilige Bearbeiter festgehalten werden. Außerdem wird über den Button „Schlagwort“ die Definition der assoziierten Schlagworte unterstützt.

Charakteristische Elemente

Visuelle Sprache

Durch die Vorgabe einer Gruppe von Eintragstypen, der visuellen Sprache, ist es möglich, die Klassifikation von Beiträgen zu vereinheitlichen, sofern die Typen adäquat

verwendet werden. Damit soll die Strukturierung der Inhalte gefördert werden und die Rezeption unterstützt. Dies ist besonders in der Gruppensituation wichtig, in der die Mappen von unterschiedlichen Personen verstanden werden sollen. Die Sprache stellt vier Eintragstypen zur Verfügung, die durch Farbe und Form eine einfache Wiedererkennung ermöglichen und für bestimmte Inhalte vorgesehen sind: Trend, Idee, Assoziation und Entscheidung. Diese Kategorien wurden als besonders relevant für den angestrebten Verwendungszweck angesehen.

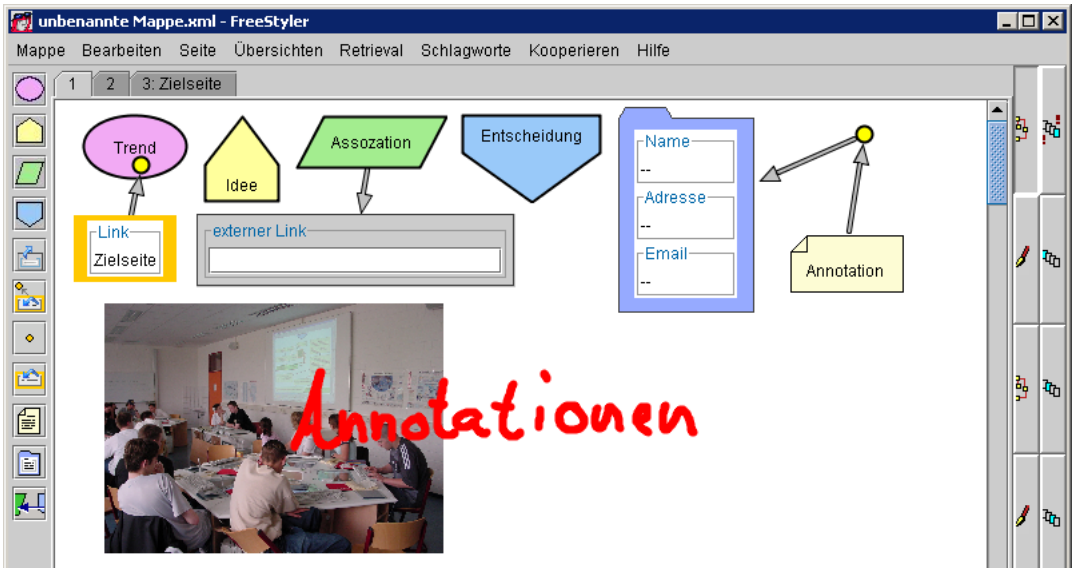


Abbildung 92

Die Elemente der visuellen Sprache.

Weiterhin existiert ein interner Link (orangener Rahmen) und ein externer Link, mit dem beliebige Dokumente und URLs referenziert werden können. Zur Strukturierung einer Seite stehen einfache Relation zur Verfügung. In eine Art Karteikarte kann Name, Adresse und E-Mail zu Kontakten eingegeben werden. Ein Annotations-Eintrag steht unbestimmten Eingaben offen. Ein kleiner gelber Kreis kann verwendet werden, um auf einen „Ort“ in der Darstellung hinzuweisen und diesen zu kommentieren. Bilder, handschriftliche Notizen und Skizzen können beliebig eingefügt und verschoben werden.

Inhaltsobjekte

Sowohl die Einträge und Relationen zwischen den Einträgen als auch die handschriftlichen Eingaben werden hier als Inhaltsobjekte bezeichnet. Deren entscheidende Eigenschaft ist, dass sie ein explizites Modell besitzen. Auf diesem Modell kann operiert werden, wodurch beispielsweise die Kopplung von Applikationen ermöglicht wird (Tewissen, Baloian et al., 2000).

Das Modell ist außerdem Grundlage des Speichervorgangs mit der Java-DOM-Schnittstelle. Mit Hilfe des Modells kann also das Objekt mit seinem Zustand persistent gespeichert werden.
Prinzipiell können über die Inhaltsobjekte deren Interaktionen kontextabhängig bestimmt werden.
Das zugrundeliegende Modell stellt auch einen wesentlichen Unterschied von FreeStyler zu allgemeinen Grafik-Tools dar. Über allgemeine Speicherformate (gif, jpeg, bmp,...) sind die einzelnen Objekte später nicht mehr zugreifbar und bei der Verwendung kann dann nicht mehr mit den Objekten interagiert werden.

Interne Links (Contentobjekt)

Interne Links sind durch die User definierbar. Ihr Inhalt stellt eine Sprungadresse innerhalb der aktuellen Mappe dar. Diese Sprungadressen werden für Seiten definiert. Jeder Seite können mehrere dieser Sprungadressen zugeordnet werden. Die Adressen haben dabei die doppelte Funktion, gleichzeitig als Einträge auf den Seitenreitern eingeblendet zu werden, wodurch sie auch als Überschriften eingesetzt werden können. In den Reitern werden alle Sprungadressen einer Seite aufgenommen, für einen Link reicht es jedoch aus, einen dieser Begriffe zu benennen.

Hyperlinks (Contentobjekt): Interaktion mit externen Browsern

Ein Eintragstyp ist für Hyperlinks vorgesehen, die auf Dateien verweisen oder Web-Sites (URLs). Insbesondere können andere XML-Dokumente aufgenommen werden, um so z.B. die FreeStyler-Mappen selbst zu verlinken. Durch Anklicken des Linkeintrages wird die referenzierte Datei aufgerufen und im entsprechenden Browser angezeigt. So können über die Hyperlinks externe Dokumente in einen inhaltlichen Kontext eingebunden werden. Realisiert wird der Aufruf der Browser durch die run.dll, die Bestandteil von Microsoft Windows ist, wodurch allerdings in Bezug auf diese Funktionalität eine Festlegung auf dieses Betriebssystem stattfindet.
In Verbindung mit anderen Eintragstypen kann der Typ oder die Idee eines Hyperlinks frei durch die User angegeben werden, ohne zwischen einem Editor- und Arbeitsmodus wechseln zu müssen. Das geht über den Freiheitsgrad bei der Erstellung von Web-Seiten hinaus. Darin können nur „html-kundige“ überhaupt Links definieren. Links sind darin Texte oder Graphiken, die selbst den Bezug des Links zu der Linkadresse verdeutlichen müssen. Beispielsweise dynamische Navigationskriterien können darin kaum unterstützt werden.

Integration einer Suchmaschine

Der FreeStyler interagiert mit einer Suchmaschine, Vertiy ([verity]), mit der Dokumente indexiert werden können. Der Index wird dann für die Volltextsuche in den Dokumenten eingesetzt. Verity sucht über allen üblichen Dateiformaten wie doc, .pdf, .xml, .txt, etc.
Verity ist über das Java Native Interface eingebunden. Dazu wurde eine Dynamik Link Library (DLL) implementiert, die die Funktionen des Verity-API nutzt. Über diese Schnittstelle wird die Indexierung von Dateien organisiert und werden Suchanfragen ermöglicht. Die Rückgabe beschränkt sich bisher auf Mengen von Dokumenten, die in Hyperlink-Einträge abgebildet werden. Über Referenzen sind die Dateien dann wieder direkt zugreifbar.

Meta-Daten

Einer Mappe können explizit Meta-Daten angefügt werden. Neben dem allgemeinen Adressteil auf der rechten Seite der Applikation können von dort auch Schlagworte vergeben werden. Bestehende Schlagworte werden dazu in einer Liste angezeigt, aus der eine Auswahl getroffen werden kann. Natürlich können auch neue Schlagworte definiert werden.

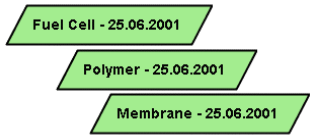
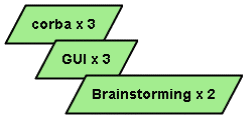
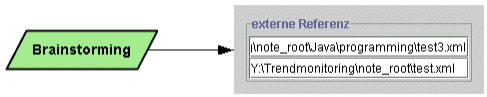
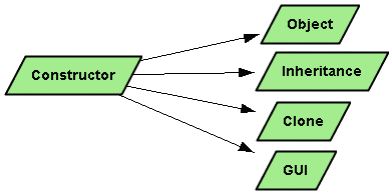
Anfrage	Ausgabebeispiel
Finde die N neuesten Schlagworte.	
Finde die N häufigsten Schlagworte.	
Finde Dokumente, in denen Schlagwort X vorkommt.	
Mit welchen Schlagworten wurde Schlagwort X kombiniert?	

Tabelle 19

Anfragen an die Datenbank für die Schlagworte und die Ausgabe in Form von Einträgen in die Mappen.

Die Schlagworte werden in einer Datenbank (Access) verwaltet, in der derzeit drei Tabellen angelegt sind:

- Die Verbindung der Dokumenten-Ids mit ihrem Dateinamen und ihrem Dateipfad.
- Die Verbindung eines Schlagwortes mit seiner Id und dem Erstellungsdatum.
- Die Verbindung der Schlagwort-Ids und der Dokument-Ids.

An diese Datenbank können bisher die in Tabelle 19 aufgeführten, vorgefertigten Anfragen gestellt werden, die Java-intern über das SQL-interface realisiert sind. Die Ausgaben finden jeweils in Form von Einträgen in die Mappe statt.

Die Entscheidung, in der *THP* kein Dokumentenmanagement-System einzusetzen, führt in diesem Fall dazu, dass zwar die Schlagworte in einer Datenbank verwaltet werden, nicht aber die Dokumente. Das führt zu der Schwierigkeit, die Referenzen auf Dateien, die in der Datenbank zu den Schlagworten verwaltet werden, konsistent zum tatsächlichen Dokumentenbestand zu halten. Derzeit besteht keine Komponente, die die Dateien und deren Pfade konsistent zum Filesystem abgleicht. Aufgrund der im XML-Format abgespeicherten Meta-Daten wäre jedoch prinzipiell ein Filesystem-konsistenter Abgleich der Datenbank möglich. Denkbar wäre auch eine Lösung durch kontrollierte Zugriffe auf Dokumente, die sofort eine Änderung in der Datenbank bewirken.

Durch die Datenbank ergibt sich im Gegensatz zur integrierten Suchmaschine ein wesentlicher Vorteil: Es können auch Dateien und Schlagworte verwaltet werden, die eben nicht mehr existieren. Daraus können interpretierende Analysen unterstützt werden, wie beispielsweise seit wann welche Schlagworte nicht mehr vergeben wurden oder auch in welchen Zeiträumen welche Schlagworte besonders häufig verwendet wurden.

Neben diesen expliziten Schlagworten sind außerdem alle im Modell des Objektes aufgeführten Daten, die zur Generierung des Objektes notwendig sind, auch im gespeicherten XML-Format enthalten. Diese Daten sind damit ebenfalls durchsuchbar bzw. indexierbar, was mit der integrierten Suchmaschine Verity realisiert wird.

Seiten

Die Mappen sind in Seiten gegliedert, denn es wurde davon ausgegangen, dass Seiten eine gewohnte Strukturierungsmethode sind. Sie erlauben genauso das fortlaufende („lineare“) Schreiben wie grundlegende Gruppierungen über die Seiten, z.B. thematisch. Durch diese Gliederung wird keine zusätzliche Struktur zwingend gefordert.

In einem reinen Hypertext müsste jeder Knoten ausgiebig mit Links zu anderen Knoten in Verbindung gesetzt werden, um ihn geeignet einbinden zu können. Dies setzt ein hohes

Maß an Strukturierung voraus, von dem wir für die in diesem Projekt vorliegenden Arbeitsszenarien nicht ausgehen konnten.
Seiten können „unbegrenzt“ generiert werden. Jede Seite ist durch einen Reiter gekennzeichnet, der gleichzeitig die Funktion einer Überschrift erfüllen kann. Diese Überschriften werden als Sprungadresse für interne Links verwendet. Beim Design wurde ein Vorteil darin gesehen, mit diesen Reitern direkt zwischen Seiten wechseln zu können. Ein anderer Ansatz wäre das Blättern oder ein Wechsel zu einer Übersichtsseite. Eine Übersichtsseite könnte zusätzlich zur Verfügung gestellt werden.

Layer

Jede Seite besteht aus mehreren Layern. Dabei handelt es sich im Prinzip um durchsichtige, übereinander gelegte Schichten. Die Reihenfolge der Schichten kann zur Bearbeitung gewechselt werden. Insbesondere für Präsentationen ist die Möglichkeit sinnvoll, einzelne Schichten auszublenden. Es stehen zwei Arten von Layerobjekten zur Verfügung: Layer zur Eingabe von Handschriften und Skizzen, deren Objekte die einzelnen Striche (Strokes) sind und Layer, die die Graph-Struktur verwalten, die hinter den Einträgen und Verbindungen steht.
Das Tool FreeStyler setzt hier auf Komponenten auf, die im Rahmen des NIMIS-Projektes (Hoppe, Lingnau et al., 2000; [nimis]) entwickelt wurden. Übernommen wurde u.a. die Graph-Klasse, die die Datenstruktur zur Verwaltung der Einträge zur Verfügung stellt. Das Objekt, das die Handschrifteingabe erlaubt, ist für den FreeStyler von einer Pixel-orientierten Version zu einer Strich-orientierten Version reimplementiert worden.
Jede FreeStyler-Seite bietet genau vier Layer an: Der unterste bietet eine Graphstruktur, danach folgt ein Handschriftlayer und darauf je noch ein Graph und eine Freihand-eingabe. Die Idee dieser Reihenfolge besteht darin, zu jeder Graphstruktur eine Freihand-Annotationsebene zu bieten. Eine Umfangreichere Schichtung stellte sich als sehr kompliziert im Umgang heraus.
Die angedeutete Annotation ist bisher rein visuell und an die Repräsentation im Browser gekoppelt. Auch wenn hier eine Idee der Freihandeingabe die der Annotation der Einträge ist, so findet sich diese Zusammengehörigkeit bisher nicht in den Modellen wieder.

Filter

Mit Filtern werden bestimmte Typen von Inhaltsobjekten aus bestehenden Seiten herausgegriffen und als Kopie in einer neu generierten Seite zusammengefasst. So entstehen Übersichten über eine bestimmte Art von Information. Beispielsweise wird das Filtern von Einträgen unterstützt, in denen Adressen aufgenommen werden. So entsteht

eine Kontaktübersicht zu einem Thema. Eine weitere sinnvolle Übersicht besteht in der Zusammenfassung aller Einträge, in denen Trends benannt werden.

XML Format

Die Mappen werden im XML-Format abgelegt. Für den Speicher- und Ladeprozess wird die Java-DOM-Schnittstelle verwendet.
Gespeichert werden darin die Modelle der Einträge und ihrer Verbindungen. Die Schlagworte werden unter entsprechenden Schlagwort-Tags ebenfalls im XML-File abgelegt.
Alle in den Mappen gespeicherten Daten stehen so der Suche und Indexierung über Verity offen. Werden Freihandeingaben gespeichert, so werden diese allerdings in Striche („strokes“) unterteilt und so gespeichert. Es wurde bisher keine Handschrifterkennung integriert.

Gruppeninteraktion

Neben Meetings mit Kunden und vereinzelten Abstimmungsgesprächen wird in der *THP* hauptsächlich asynchron kooperiert. Suchfunktionen nach Schlagworten und Inhalten können dies unterstützen, da so immer wieder Änderungen gefunden und in die Arbeit einbezogen werden können. Zusätzlich wird Mailfunktionalität integriert.

Abbildung 93

Der Maildialog im FreeStyler.

Über den Maildialog (Abbildung 93) können die erarbeiteten FreeStyler-Seiten als Bilder versendet werden. Auf diese Weise ist es möglich, Sitzungsnotizen schnell und unkompliziert auch an solche Personen zu verschicken, die den FreeStyler nicht installiert haben. Definierbar ist, welche Seiten in welcher Größe versendet werden sollen. Ein Text

kann dieser E-Mail beigefügt werden. Auch die XML-Datei soll über diesen Dialog mit als Attachment versant werden können.

Angestrebt war, ein Konzept für die automatische Versendung von Seiten zu entwickeln. Als Grundlage hätte die Zuordnung zu Interessengruppen dienen können, was in der relativ kleinen Arbeitsgruppe der *THP* leicht möglich wäre.

Abbildung 94 stellt dar, mit welchen externen Komponenten der FreeStyler kommuniziert. Dazu gehören die Zugriffe auf das Filesystem, die Suchmaschine Verity, die Schlagwortverwaltung in der Access-Datenbank und die Einbindung von Mailfunktionen.

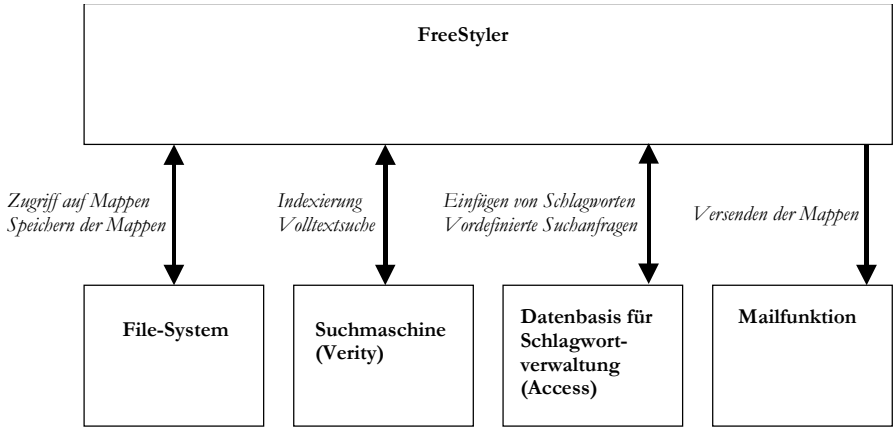


Abbildung 94

Datenflüsse bei der Interaktion des FreeStylers mit anderen Komponenten.

7.1.6 Beispiel für die Entwicklung eines Trendkontextes

Die folgenden Abbildungen sollen verdeutlichen, welche Informationen auf welche Art während unterschiedlicher Arbeitsphasen zu Trendkontexten zusammengestellt werden können. Das Beispiel „Brennstoffzelle“ wurde auf Anraten der *THP* ausgewählt, da sie in diesem Bereich oft Anfragen erhält und in der Universität Duisburg-Essen dazu auch Forschungsgruppen arbeiten. Das Beispiel ist nachgestellt, jedoch an den momentanen Abläufen orientiert.

Abbildung 95 stellt Notizen dar, wie sie während eines Telefongespräches bei einer ersten Kontaktaufnahme entstehen. Ein Industrievertreter erkundigt sich darin nach Forschungsgruppen zum Thema Brennstoffzellen. Typischerweise wird nach Referenzprojekten und Ergebnissen gefragt, insbesondere nach Veröffentlichungen und anderen schriftlichen Informationen. Die grün umrandeten Stichpunkte markieren bereits Punkte, die offen sind, zu denen später recherchiert werden muss.

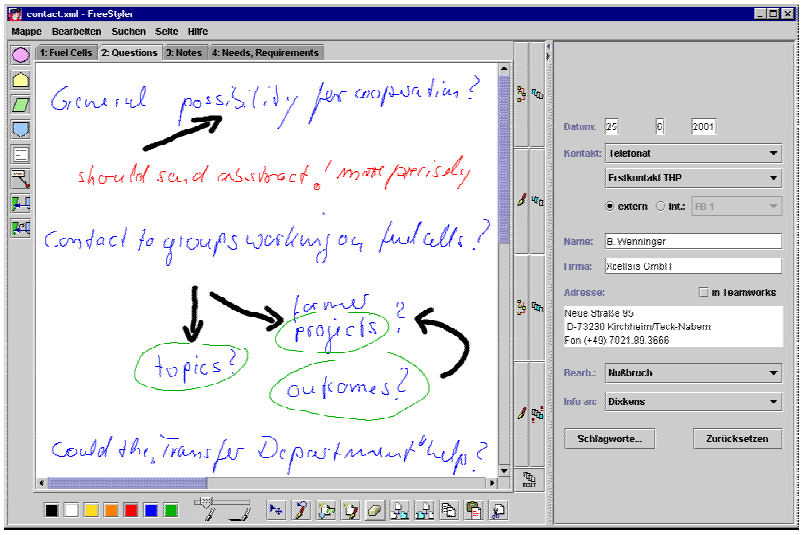


Abbildung 95

Typische Notizen während eines Telefongespräches im FreeStyler notiert.

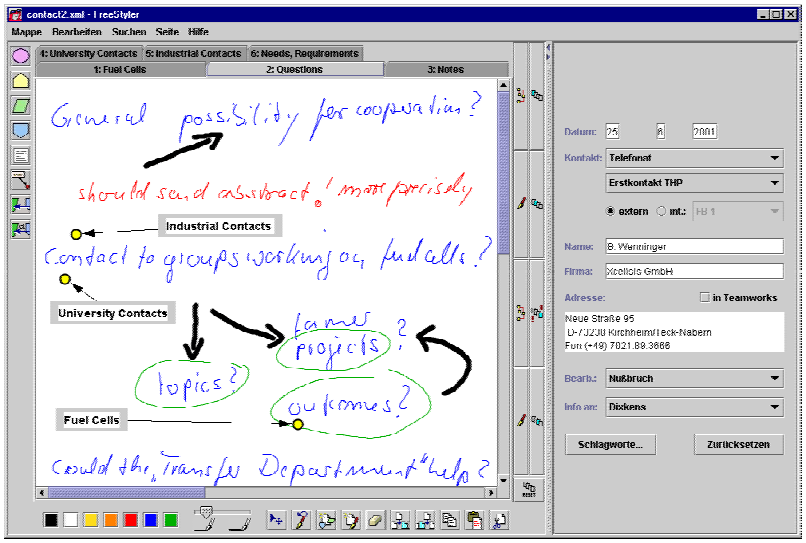


Abbildung 96

Erste Überarbeitungen nach einem Telefonat.

Auf der rechten Seite der Applikation können Notizen zur Kontaktperson eingetragen werden, die im gespeicherten Dokument später mit enthalten und damit wieder durchsuchbar sind. Die Handschrifteingabe mit einem digitalen Pen-Input ermöglicht in dieser Situation die einhändige Notiz während des Telefonierens.

Abbildung 96 zeigt einen ersten Überarbeitungsschritt. Die rechteckigen Einträge stellen dokumentinterne Links dar (in der ersten Version), zu denen die „gelben Punkte“ verwendet werden, um den Bezug des Links zu verdeutlichen. Die einzelnen Seiten können durch Begriffe übertitelt werden, die in den Reitern erscheinen.

Entspricht der Text eines Links einem Begriff einer Seitenüberschrift, so wird dies erkannt und beim Anklicken des Links der Sprung auf die entsprechende Seite vollzogen. Von nun an entwickelt sich die erste Kontaktnotiz zu einem Dokument, der Mappe, in der bestimmte Informationen wie Kontakte oder Hintergrundwissen zusammengefasst sind.

Abbildung 97 zeigt eine Art von Liste zu Forschungszentren. Verbunden ist jeweils der Name der Institution mit der dazugehörigen URL. Beim Anklicken dieses Verweises wird ein Web-Browser mit der angegebenen Adresse geöffnet, wie es in dem rechten Fenster in der Abbildung zu sehen ist. Die Verweise in den Einträgen sind nicht auf URLs beschränkt. Es können beliebige Pfade zu Dateien angegeben werden. Der jeweils adäquate Browser wird dazu automatisch ausgewählt.

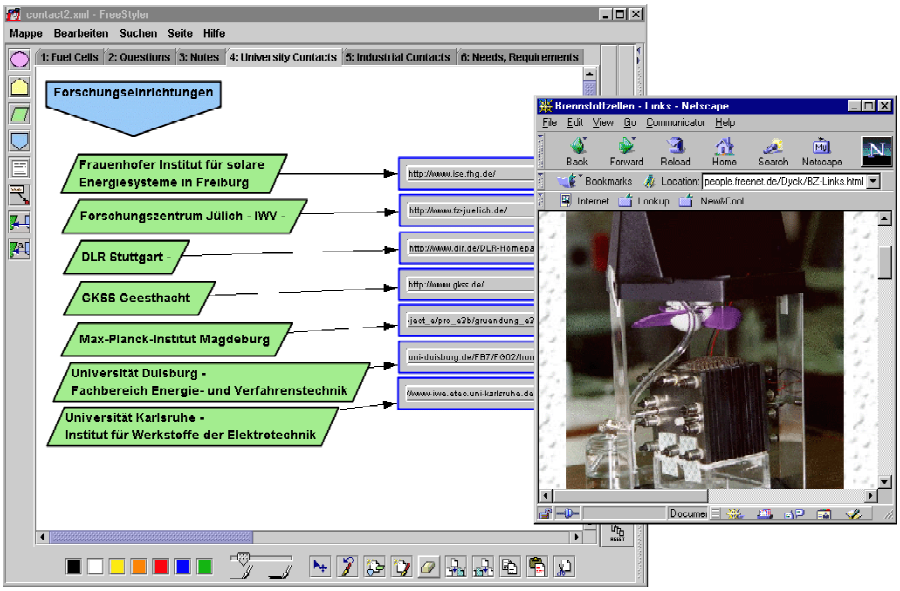


Abbildung 97

Verweise zu externen Dokumenten.

Diese Art der Darstellung kann auch gut als Sitzungsvorbereitung verwendet werden. In der Sitzung kann dann sehr schnell auf die jeweiligen Seiten zugegriffen werden, um die Inhalte dort mit in das Gespräch einzubeziehen. Das gilt insbesondere dann, wenn es bei der Sitzung nicht um die Präsentation von Ergebnissen geht, sondern die Vorbereitung eine Grundlage für die Arbeit mit Materialien bieten soll, die in die Sitzung einfließen. So kann eine Bandbreite von Informationen vermittelt werden, ohne alle Informationen direkt zu integrieren, denn eine zu spezielle Vorauswahl kann auch kontraproduktiv sein. Während des Treffens können dann ungeeignete Links gelöscht oder auch für den konkreten Partner eine besondere Auswahl angefertigt werden.

Abbildung 98 kombiniert diverse Eintragstypen mit denen Hintergründe zum Thema Brennstoffzelle zusammengetragen wurden. Links oben ist die Brennstoffzelle durch den

ovalen Eintrag als Trend markiert. Dazu in Relation werden unterschiedliche Arten von Brennstoffzellen aufgeführt, die jeweils auf einer anderen Technik beruhen. In einer Liste sind diverse externe Dokumente zum Thema referenziert. Eine Grafik rechts unten stellt die Funktion der Brennstoffzelle dar.

Im Vordergrund ist ein Dialog geöffnet. Mit diesem kann über die angebundene Suchmaschine Verity in bestehenden Textarchiven nach Schlagworten gesucht werden. Die Rückgabe wird wieder in einer Liste dargestellt, in der die gefundenen Dokumente referenziert werden. Eine solche Liste ist hinter dem Dialog zu erkennen.

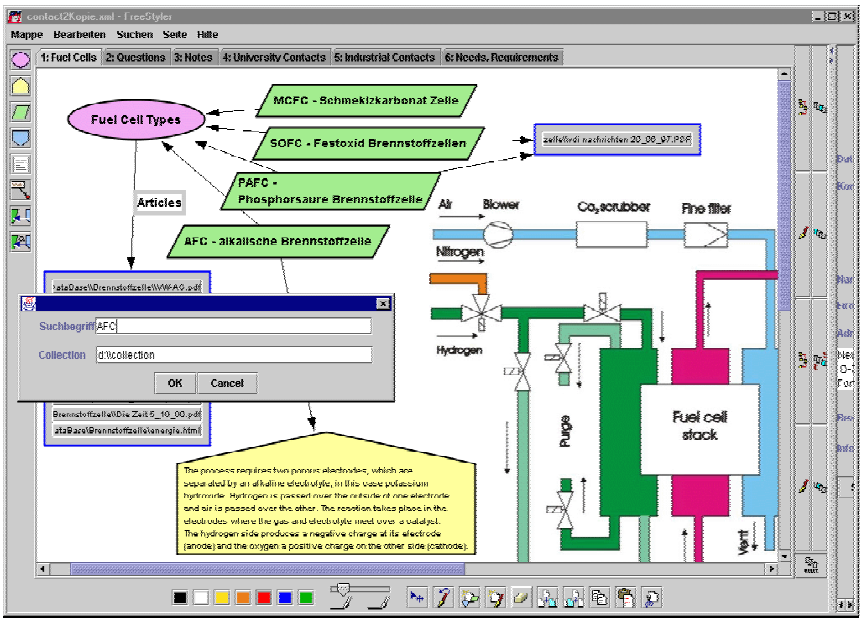


Abbildung 98

Sammlung von Hintergrundwissen zu Brennstoffzellen.

In Abbildung 99 wird die mögliche Verwendung des FreeStyler zum „Concept Mapping“ dargestellt. Begriffe und Erklärungen werden zueinander in Relation gesetzt. In der Abbildung wird auch die Kombination der Einträge mit handschriftlichen Notizen deutlich. Im Zusammenhang mit der „Concept Map“ können so z.B. Fragen notiert werden.

Im Vordergrund wird ein weiterer Dialog gezeigt. Dieser bietet die Möglichkeit, vorgefertigte Typen von Anfragen an die Schlagwort-Datenbank zu stellen. Beispiele für solche Anfragen sind: „Gib mir alle Dokumente, die mit Schlagwort X verschlagwortet wurden.“ „Gib mir die 5 neuesten Schlagworte.“ Die Ausgabe dieser Anfragen wird ebenfalls in Einträgen dargestellt. Verweist die Antwort auf Dokumente, so werden dafür die entsprechenden Links generiert (vgl. Tabelle 19).

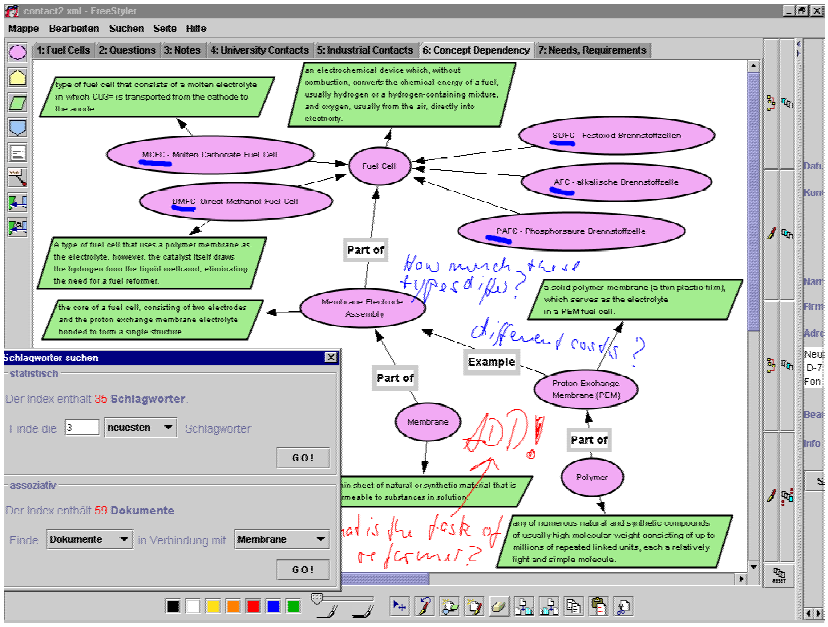


Abbildung 99

Begrifflichkeiten zum Thema Brennstoffzelle wurden in einer Art “Concept Map” dargestellt.

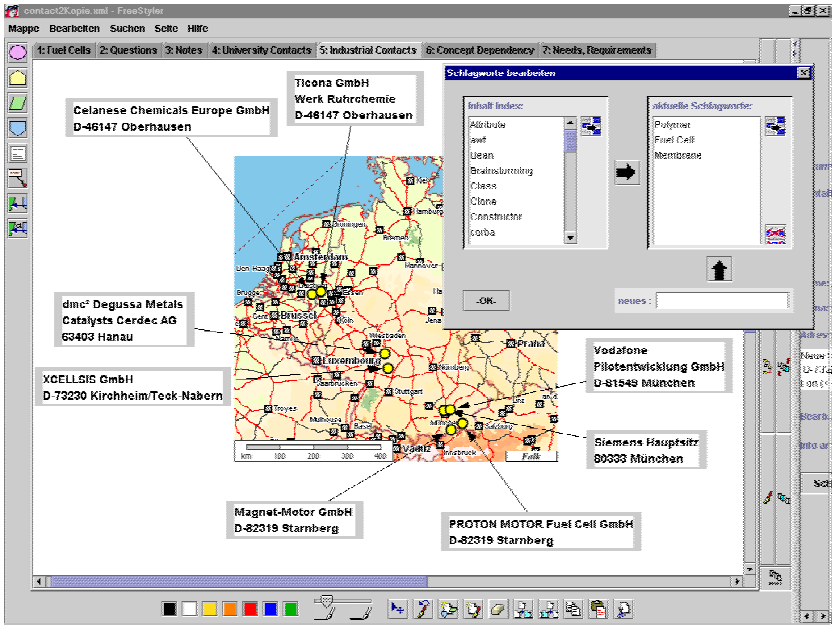


Abbildung 100

Grafische Einträge können z.B. mit einfachen, visuellen Anmerkungen versehen werden. Die gesamte Mappe kann verschlagwortet werden.

Abbildung 100 zeigt eine einfache aber übersichtliche Visualisierung von Kontakten auf Basis einer Landkarte, die durch die Layer der einzelnen Seiten realisiert werden kann. Auch in dieser Abbildung ist ein Dialog angezeigt. Mit diesem können Schlagworte für die Mappe vergeben werden. Die linke Seite des Dialoges zeigt eine Liste aller bisher verwendeten Schlagworte, aus der eine Auswahl getroffen werden kann. Die Liste läuft

alphabetisch mit der Eingabe eines neuen Schlagwortes mit, um zu jedem Zeitpunkt ähnliche Worte zu erkennen. So sollen vor allem doppelte Schlagworte wegen nur leicht differierender Schreibweise vermieden werden.

7.2 FreeStyler Version 2

Das entstandene Tool wurde auch nach Abschluss des referierten Projektes weiter entwickelt und auch an andere Bedingungen adaptiert. Während der Weiterentwicklung hat sich auch das Interface des FreeStylers verändert, was in den folgenden Abschnitten deutlich wird. Im Folgenden wird der neue Entwicklungszustand kurz dargestellt und der des FreeStylers im Deutschunterricht etwas ausführlicher beschrieben.

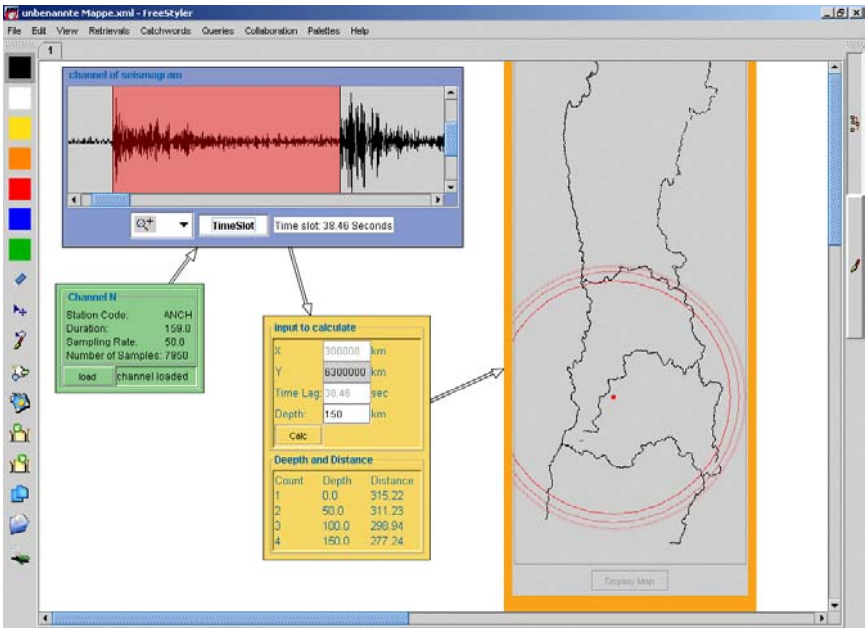


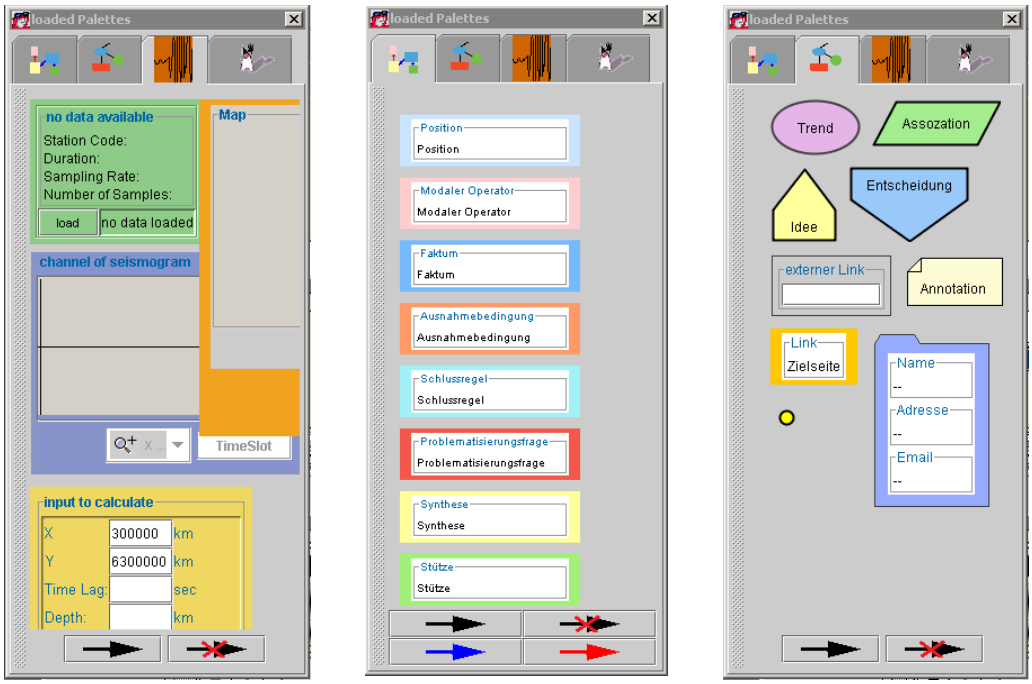
Abbildung 101

Datenauswertung zur Bestimmung von Erdbebenepizentren.

Zusätzlich zum Szenario „Wissensmanagement“ wurde und wird der FreeStyler unter der Maßgabe verändert, flexibel in unterschiedlichen Situationen einsetzbar zu sein. Dazu gehört die Unterstützung von Evaluationen, die beispielsweise im psychologischen Kontext durchgeführt werden. Ein Schwerpunkt liegt in der Möglichkeit, ihn im Lehr-/Lernbereich einzusetzen.

Eine zentrale Änderung liegt in der Verwendung von Paletten für die visuellen Sprachen anstatt einer Toolbar, die an einen bestimmten Einsatz adaptiert ist. Neue visuelle Sprachen können so sehr einfach zum Tool hinzugefügt werden. Für die Implementierung der Paletten wurde das Konzept eines Tools übernommen, das parallel dazu im Rahmen

anderer Projekte am Lehrstuhl entwickelt wird (Cool Modes: Pinkwart, Hoppe & Gaßner, 2001; Pinkwart, Hoppe et al., 2002). Abbildung 101 und Abbildung 102 zeigen das neue Interface und drei Paletten dazu. Im Folgenden werden kurz verschiedene aktuelle Einsätze des FreeStylers beschrieben, darunter etwas ausführlicher dessen Einsatz im Deutschunterricht.



a) b) c)

Abbildung 102

Drei Paletten des FreeStylers: a) Eine Palette zur Analyse von Epizentren. b) Ein Palette zum Aufbau von Argumentationsnetzen. c) Die bereits erläuterte Palette zum Wissensmanagement.

Analyse von Erbeben. Im EU-Projekt Coldex (Collaborative learning and distributed experimentation: IST 2001-32327; [Coldex]) wurde und wird eine Palette entwickelt[16], deren Objekte zur Bestimmung von Erbebenzentren verwendet werden können. Kooperiert wird mit Messstationen in Chile, um so mit realen Daten arbeiten zu können. Abbildung 101 zeigt einen Analyseablauf, wie von aufgezeichneten Erdschwingungen das Epizentrum eines Erdbebens ermittelt werden kann. Dieser Ablauf folgt den Pfeilen zwischen den Einträgen. Diese Relationen besitzen intern die Funktion des Datentransfers zwischen den Visualisierungen. Begonnen wird mit dem Laden der Schwingungsdaten,

[16] Diese Palette wird von S. Buschmann am Lehrstuhl entwickelt, der als Student der Physik das entsprechende fachliche „Know How“ einbringt. Die Implementierung wird voraussichtlich Bestandteil seines Studienabschlusses sein.

über die die Abstände zwischen den beiden ersten Erdbebenwellen bestimmt werden Es folgt die Eingrenzung des Epizentrums, das in einer Karte kreisförmig umrissen wird.

Argumentation. Die Objekte der Argumentationspalette stellen mögliche Elemente eines Argumentes dar. Eingesetzt werden soll die Palette an der Universität Dresden, am Lehrstuhl von Prof. Koerndle im Rahmen einer Diplomarbeit im Bereich Psychologie[17]. In der Evaluation sollen zwei Tools verglichen werden, FreeStyler und Interwise, in Bezug auf die Strukturiertheit von Diskussionsergebnissen.

7.2.1 Einsatz des FreeStylers im Deutschunterricht

Im Rahmen des EU-Projektes SEED (IST-2000-25214; [SEED]) wurde der FreeStyler während des Deutschunterrichts in einer 10ten Klasse, am Rheinkamp-Gymnasium Moers, eingesetzt. Das Ziel dieses Einsatzes bestand wesentlich im Test der technischen Funktionsfähigkeit der Software bzw. der replizierten Architektur. Bei der Entwicklung kopplungsfähiger Applikationen für den Gebrauch in mittelgroßen Gruppen besteht das Problem, ausreichend komplexe Testszenarien aufzubauen. Viele technische Probleme treten erst bei größeren Datenmengen und zeitlich dichten Interaktionen auf. Ein solches Testszenario bindet jedoch viele Personen und kann innerhalb einer kleinen Entwicklergruppe nur schwierig aufgebaut werden.

Unter Beobachtung stand weiterhin auch, inwieweit die Funktionalität der Software für dieses Thema ausgenutzt werden konnte. Gerade ein Deutschunterricht zu unterstützen sollte außerdem ein Unterrichtsbeispiel dafür aufzeigen, wie mit einem solchen Ansatz auch nicht-technische Inhalte von der Software-Verwendung profitieren können.

Die technischen Voraussetzungen des Gymnasiums waren überdurchschnittlich günstig. Dort existierte ein Klassenraum, der mit ca. 20 vernetzten Computern ausgestattet war (Abbildung 103). Diese waren in drei Tischkombinationen jeweils kreisförmig angeordnet, so dass an diesen Tischen auch Gruppenarbeit möglich war und „face-to-face“ kommuniziert werden konnte, da Sichtkontakt bestand. Die Lehrperson, in diesem Fall eine Lehrerin, hatte einen speziell ausgestatteten Computer, über den u.a. ein Videoprojektor angesteuert wurde.

Der Unterrichtsgegenstand war das Stück „Der Besuch der alten Dame“ (Dürrenmatt, 1980), das im Vorfeld bereits Gegenstand des Unterrichts war.

Setting. Der Test wurde mit einer Schulklasse im 10ten Schuljahr durchgeführt. Die Klasse bestand aus über 20 Personen, die für den Unterricht in Arbeitsgruppen von 2-3 Personen aufgeteilt wurden. Diese Arbeitsgruppen teilten sich jeweils einen Computer,

[17] Diese Diplomarbeit wird von Frau J. Krist durchgeführt, die auch die Palette entworfen hat.

wobei es nicht verhindert wurde, wenn freie Rechner mit einbezogen wurden, so dass sich die Gruppen aus Eigeninitiative teilweise noch aufteilten. Jeder Computer wurde für den Test mit einem Tablett zur Pen-basierten Eingabe ausgestattet.

Abbildung 103

Der Computerklassenraum am Rheinkamp-Gymnasium Moers.

Die Probanden waren mit dem Klassenraum vertraut aber nicht mit dem FreeStyler und nicht mit der Idee einer replizierten Architektur, in der sich die Applikationen temporär koppeln lassen. Es ist aber davon auszugehen, dass alle mit der Verwendung von Chats vertraut waren und damit mit der Idee der verteilten Kommunikation.

Bei der Durchführung der Gruppenarbeiten waren außer den Schülerinnen und Schülern und der Lehrerin noch vier weitere Personen aus dem Entwicklerteam anwesend. Davon leisteten drei Hilfestellung bei technischen Problemen sowie bei Fragen zur Nutzung der Applikation. Die vierte Person war ausschließlich für die Beobachtung zuständig. Für inhaltliche Fragen stand ausschließlich die Lehrerin zur Verfügung.

Versuchsablauf. Der Test erstreckte sich über 4 Schulstunden.

1. Stunde	15 Minuten: Erläuterung der Bedienung des FreeStylers im Tool selbst unter Einsatz des Videoprojektors. Vorstellung einer vorbereiteten FreeStyler-Mappe zum Thema „Auge“. Kurze Anleitung zur Pen-basierten Eingabe.30 Minuten: Die Schüler sollen anhand einer einfachen Aufgabenstellung mit dem FreeStyler vertraut werden. Dafür wird eine Aufgabenstellung als FreeStyler-Mappe vorbereitet, die wiederum mit dem FreeStyler gelöst werden muss. Am Ende der Arbeitsphase sollen die Schüler über den MatchMaker-Server koppeln.
2./3. Stunde	Aufgabenstellung zum Stück „Der Besuch der alten Dame“. Von der Lehrerin waren Gruppenaufgaben im FreeStyler vorbereitet worden. Darin sollten die Handlungen verschiedener Charaktere des Stückes verteidigt bzw. angeklagt werden: Polizist,

	Bürgermeister, Lehrer, Pfarrer, die alte Dame selbst. Das Buch stand als Material zur Verfügung. Die Aufgabe bestand in der Erstellung einer strukturierten Darstellung der Aspekte, die in eine Verteidigung oder Anklage mit aufgenommen wurden. Allen Schülerinnen und Schülern war bekannt, dass die Ergebnisse anschließend der Klasse vorgestellt werden sollten.
4. Stunde	Präsentation aller Gruppenergebnisse unter Nutzung der Kopplungsfunktion des FreeStyler. Die Dramaturgie der Präsentation bestand vor allem darin, dass die zueinander passenden Gruppen, also Anklage und Verteidigung einer Charaktere, ihre Argumente gegeneinander anbringen sollten.

Tabelle 20

Testplanung für den Deutschunterricht.

Beobachtungen. Am Schluss konnten alle Gruppen die Aufgabenstellung in der intendierten Art und Weise lösen (Beispiele: Abbildung 104 und Abbildung 105). Nach Aussagen der Lehrerin konnten qualitative Unterschiede der Ergebnisse in den direkten Zusammenhang mit dem inhaltlichen Verständnis des Stückes gesetzt werden. Nach ihrer Aussage war es außerdem erkennbar, dass sich gerade schwächere Schülerinnen und Schüler mehr als sonst engagiert haben.

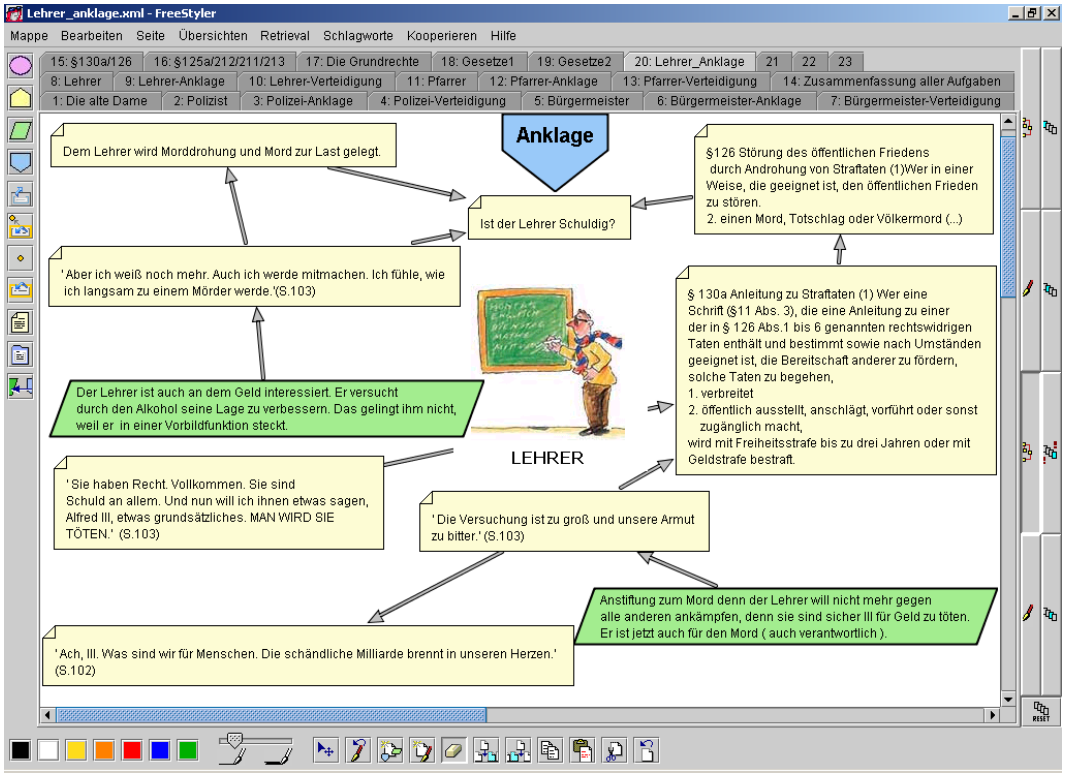


Abbildung 104

Ergebnis der Gruppe „Lehreranklage“.

In der festgesetzten Zeit war es auch möglich, dass wirklich alle Gruppen zur Präsentation ihrer Ergebnisse kamen. Nach Aussage der Lehrerin bestand ein Vorteil in der Verwendung der Software insofern, als dass die Vorbereitung relativ unkompliziert

war und die Funktionalität zu der Aufgabenstellung passte, die bereits geplant war bevor der Einsatz des FreeStylers feststand. Die Möglichkeit, dass alle Gruppen auch zur Präsentation kamen, wurde als ungewöhnlich eingeschätzt und ausgesprochen positiv aufgenommen. Nur durch diese Möglichkeit wurde direkt deutlich, dass ein großes Defizit bei den Präsentationsfähigkeiten der Schülerinnen und Schülern bestand. Dieser Mangel wurde als besonders negativ eingeschätzt, da darin eine überall geforderte Fähigkeit für den beruflichen Alltag besteht.

Während des Test wurde auch der Umgang der Schülerinnen und Schüler mit dem Tool beobachtet: Während der Einführung wurde aufmerksam und ruhig zugehört. Die Arbeit mit dem FreeStyler bereitete den Schülerinnen und Schülern keine großen Probleme. Kleinere technische Fragen und Probleme behinderten nicht und konnten meistens schnell durch die drei Betreuer gelöst werden.

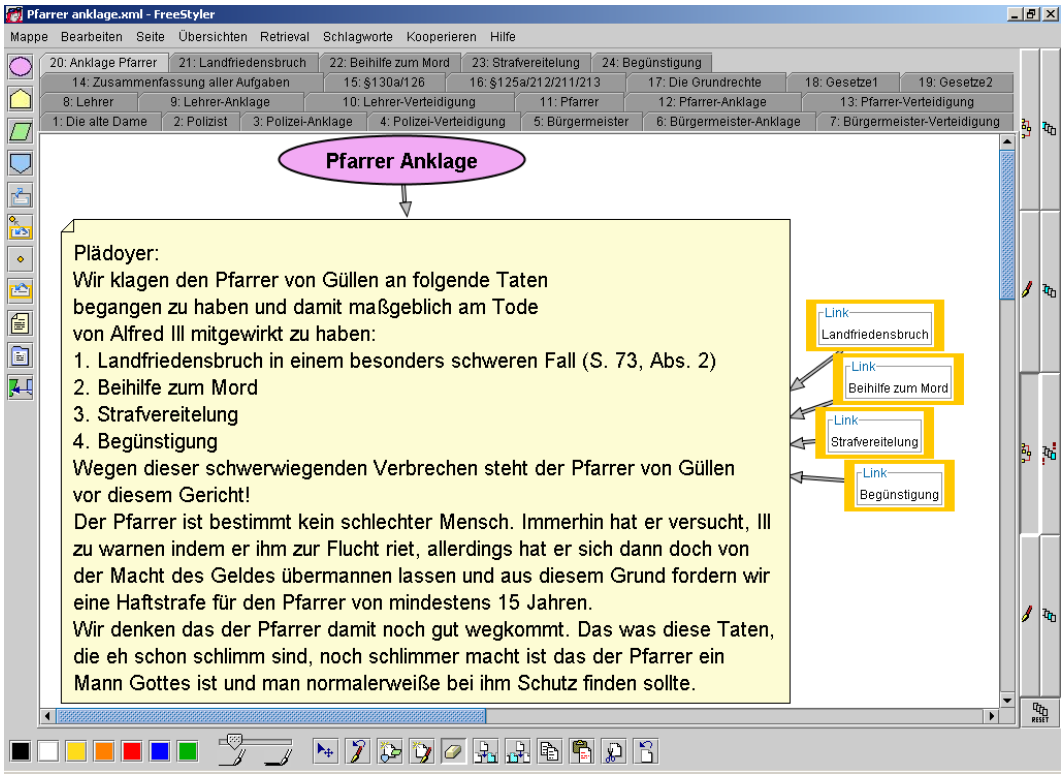


Abbildung 105

Ergebnis der Gruppe „Pfarreranklage“ mit Links zu weiteren Seiten.

Nach der Erläuterung der Kopplung aber noch während der Einführung, stellten wohl beinahe alle Probanden Unsinn an und übermalten sich gegenseitig die Ergebnisse. Dies hat sich als positiv für die weiteren Arbeiten herausgestellt, denn so waren alle betroffen, da sie sahen, wie ihre eigenen Ergebnisse verschwanden. In der eigentlichen Unterrichtssituation traten diese gegenseitigen Behinderungen dann nur noch ein einziges Mal auf.

Spannend war zu beobachten, dass die Kopplung, die vorher nur funktional eingeführt worden war, sofort aufgegriffen worden ist. Die Schülerinnen und Schüler begannen von sich aus, diese gezielt dazu einzusetzen, bei ihren „Gegenspielern“ zu spicken. Gerade in dieser Aufgabenstellung muss das als besonders konstruktiv gewertet werden, da das dazu führen kann, auch Argumente der anderen Gruppen mit einzubeziehen.
Die Zuverlässigkeit des technischen Systems war zufriedenstellend. Es kam zu verschiedenen Programmabstürzen und einigen Rechnerabstürzen. Leider überlagerte sich die Instabilität der vorhandenen Computer mit den Problemen des Tools. So waren die Computerabstürze im Wesentlichen auf 2-3 Rechner beschränkt, was gegen das Tool als Ursache spricht. Toolabstürze schienen vermehrt im gekoppelten Status aufzutreten, was aber nicht überwacht werden konnte.

7.2.2 Einordnung und Verwendung

Insbesondere über die Möglichkeit der Gestaltung von Paletten kann der FreeStyler mit unterschiedlichen Tools verglichen werden.
Von der Funktion sehr ähnlich erscheint das System OneNote (vgl. Kapitel 3.2). Darin scheinen allerdings keine zusammenhängenden visuellen Sprachen vorgesehen zu sein, so dass OneNote auf die Notizfunktion eingeschränkt ist. Interaktive Objekte können nach der vorliegenden Beschreibung nicht definiert werden. Das führt dazu, dass das Tool nicht für konkrete Anwendungen spezialisiert werden kann.
Ein „Mind Mapping“-Tool wie der MindManager kann über den FreeStyler leicht erstellt werden. Über die Objekttypen können auch Constraints definiert werden, so dass bei Bedarf nur bestimmte Objekttypen miteinander verbunden werden können. Der Vorteil des FreeStylers liegt in diesem Vergleich darin, dass unterschiedliche Methoden unterstützt werden könnten.
Bei der Verwendung des FreeStylers wurde einige Male kritisiert, dass keine „perfekten“ Layouts entstehen, wie das beispielsweise durch die Verwendung von PowerPoint angestrebt wird. In der Abgrenzung dazu besteht der Vorteil des FreeStylers darin, dass er nicht zwischen einem Präsentationsmodus und der Erstellung unterscheidet. Das heißt die Änderung der Darstellung ist gleichzeitig die Änderung der Präsentation. Darin besteht gerade ein Ziel des FreeStylers, in Sitzungen nicht zwischen der Präsentation und der „Arbeitsfläche“ zu unterscheiden.
Der FreeStyler soll vor allem die Kommunikation bereichern. So können während einer Besprechung z.B. leicht Annotationen vorgenommen werden. Der Ziel, das mit dem FreeStyler verfolgt wird, besteht eher darin, den Arbeitsprozess zu unterstützen,

beispielsweise Dokumentationen zu generieren. Dies kann auf Basis der FreeStyler-Objekte definiert werden, aus denen bestimmte XML-Dokumentformate generiert werden. Diese Möglichkeit ergibt sich auch innerhalb eines FreeStyler-Dokuments, so dass Filter definiert oder Historien aufgezeichnet werden können.
Im Vergleich mit verschiedenen Editoren sind diese vor allem wenig dazu geeignet, Objekte frei zu platzieren. Das ist wiederum ein Vorteil von Grafik-Tools, die aber wieder recht unflexibel sind, wenn während einer Besprechung schnelle Skizzen erstellt werden sollen. In beiden Fällen werden außerdem keine interaktiven Objekte eingebunden, mit denen später auch Methoden ausgeführt werden können, mit Ausnahme von aktiven Links.
Der FreeStyler verbindet die Option der methodischen Unterstützung, wie es zu anderen Systemen im Kapitel 3.2 vorgestellt wurde (z.B. Sepia, Belvedere), mit einem flexiblen Editor. Die Kopplung ermöglicht dabei die Unterstützung diverser Arbeitsszenarios.

7.3 Zusammenfassung

Der FreeStyler ist ebenfalls zur Diskussionsunterstützung entwickelt worden, vor allem mit dem Ziel der nachhaltigen Gestaltung von Diskussionsergebnissen. Angestrebt wird eine themenorientierte Gruppierung von Materialien, die durch die Wiederverwendung in unterschiedlichen Arbeitsszenarien an Struktur und Umfang zunimmt. Der Diskussionsprozess wird hier langfristig gesehen und nicht zwingend auf „face-to-face“-Situationen bezogen.
Im Hintergrund stehen Ansätze zum Wissensmanagement, die die Wissensentwicklung als zentral begreifen (Nonaka, 1994; Nonaka & Takeuchi, 1995; Borghoff & Pareschi, 1998).
Der FreeStyler folgt diesen Ansätzen, adaptiert sie allerdings an die Anforderungen, die als Ergebnis einer Analyse des Arbeitsprozesses der Transferstelle Hochschule Praxis (*THP*) der Universität Duisburg-Essen, Zielgruppe des FreeStylers, konkretisiert wurden. Unabhängig von den funktionalen Anforderungen bestanden folgende Voraussetzungen:

- Es musste eine kostensparende Lösung gefunden werden.
- Die Lösung musste für Computerlaien wartbar sein.
- Die Lösung musste für Computerlaien verwendbar sein.
- Die Entwicklung einer Lösung war zeitlich stark begrenzt.
- Es handelt sich um eine sehr kleine Arbeitsgruppe.

Wissenszyklus. Ein Wissenszyklus wurde darüber unterstützt, dass der FreeStyler in diversen unterschiedlichen Arbeitsszenarien eingesetzt werden kann:

- Bei der Kontaktaufnahme, speziell beim Telefonieren, zur Erstellung handschriftlicher Notizen.
- Bei der strukturierten Vorbereitung von Sitzungen oder Treffen mit der Möglichkeit auf externe Dokumente zu verweisen.
- Bei Sitzungen zur Präsentation der vorbereiteten Inhalte mit einfachem Zugriff auf externe Materialien, zur Visualisierung der aktuell anfallenden Kommentare, für Notizen, zur Strukturierung zukünftiger Arbeiten, zum Dokumentieren der Sitzungen.
- Bei der Nachbereitung von Sitzungen, indem angeforderte Informationen integriert werden.
- Zur themenspezifischen Aufbereitung und Dokumentation, potentiell kooperativ oder als Einzelarbeit.

Diese Arbeitsszenarien integrieren kooperative Phasen genauso wie Einzelarbeitsphasen, Internalisierung genauso wie Externalisierung sowie die Kombination und Restrukturierung. Die handschriftlichen Eingaben ermöglichen ein großes Spektrum kreativer Herangehensweisen.

Wissensnavigation und –konstruktion sowie Modellbildung. Der FreeStyler verwendet wie das DiscBoard visuelle Sprachen. Für die *THP* wurde eine spezielle Sprache entwickelt, die bezogen auf die FreeStyler-Dokumente u.a. die dokumentinterne Navigation erlaubt sowie Referenzen auf externe Dokumente ermöglicht. Diese Referenzen können jeweils benannt und damit in ihrem Bezug verdeutlicht werden. Dies ist auch für Computerlaien einfach möglich.

Daraus ergibt sich die Möglichkeit, themenspezifische FreeStyler-Dokumente zu erstellen und die interne Navigation zu nutzen, sowie in diese themenspezifischen Darstellungen externe Dokumente zu integrieren. Gleichwohl ist es möglich, den FreeStyler zu einer übergeordneten Strukturierung von Dokumenten zu verwenden. Diese dokumentationsorientierte Herangehensweise entspricht einem großen Anteil der alltäglichen Arbeitsprozesse der *THP*.

Die Erstellung solcher FreeStyler-Dokumente muss als konstruktiver Prozess gewertet werden, in dem Zusammenhänge erst durch die Arbeit mit dem Tool thematisiert werden.

Wissensgemeinschaften, Wissenskommunikation, Kooperation. Die Wissenskommunikation findet im Normalfall in Sitzungen bzw. Gesprächen statt. Dazu stellt der FreeStyler die Möglichkeit der Visualisierung zur Verfügung. Die Wissensgemeinschaft

definiert sich hier maßgeblich über die Nutzung eines gemeinsamen Repositories, das ein „group memory“ bildet.

Wissensrepositories. Als Grundlage für den Bezug auf externe Dokumente müssen diese in digitaler Form vorliegen. Diese Digitalisierung ist in den Arbeitsprozess der *THP* aufgenommen worden. Auch die FreeStyler-Dokumente sind Bestandteil dieses Repositories. Durch den Aufbau eines Repositories soll die Wiederverwendung der Dokumente verbessert werden. Unterstützt wird das durch die Integration einer Suchmaschine.

Aus oben genannten Kostengründen sollte als Repository keine aufwändige Dokumentenverwaltungssoftware angeschafft werden. Das Repository beschränkt sich deshalb auf einen definierten Bereich des Filesystems, über dessen Struktur sich die Arbeitsgruppe abstimmt.

7.4 Diskussion

Entwicklung eines Repositories. Das Repository, das in der *THP* aufgebaut wurde und noch wird, wird nicht durch gesonderte Software strukturiert:

Auf ein Dokumentenmanagement-System (DMS) wurde aus verschiedenen Gründen verzichtet, die hier kurz referiert werden. DMS gehen im Allgemeinen davon aus, dass die gesamte Dokumentenbearbeitung mit diesem System abgedeckt wird, um die Konsistenz der Daten gewährleisten zu können. Das geht teilweise so weit, dass sogar eigene Viewer für die unterschiedlichen Dateiformate angeboten werden. Die Dokumente werden in angeschlossenen Datenbanken abgelegt und sind dadurch nur noch über das System zugreifbar. Daraus entstehen Probleme, wenn diese Dokumente zusätzlich in andere Verwaltungs- oder Zugangsstrukturen integriert werden sollen. Als Beispiel sind Links auf die Dateien zu nennen, die über Web-Seiten angeboten werden.

Für diese Anforderungen müssen dann immer Schnittstellen verwendet werden. Diese Schnittstellen wurden zum Design-Zeitpunkt von den meisten DMS nicht ausreichend unterstützt. Gleiches gilt für Schnittstellen, um von anderen Systemen (z.B. Java-Applikationen) beispielsweise auf Schlagworte zugreifen zu können.

Ein weiteres Problem, das u.a. Beweggrund der *mpe* war, sich an dem Projekt zu beteiligen, besteht darin, die Dokumente zu einem beliebigen Zeitpunkt aus der Datenbank zu exportieren. Das wird oft nicht gut unterstützt und führt zu erheblichen Komplikationen, wenn das Produkt gewechselt werden soll.

DMS, die solchen Anforderungen entsprechen, sind teuer und es fallen meistens zusätzliche Entwicklungskosten für die Systemadaptierung an. Wegen der Kosten, der teilweise unflexiblen Gestaltung, meist unzureichend vorhandener APIs und oft unzureichenden Möglichkeiten bei der Verschlagwortung und der Volltextsuche wurde von der Anschaffung eines DMS Abstand genommen.

Generell bestand aber trotzdem ein Bedarf an der dokumentenorientierten Verwaltung, da dies den Artikeln, Fachbeiträgen, etc. entspricht, die in der *THP* anfallen. Dies konnte beispielsweise nicht unmittelbar mit dem Konzept einer XML-Datenbank in Übereinstimmung gebracht werden. Es wurde deshalb pragmatisch entschieden, einen abgegrenzten Bereich des Filesystems für die Verwaltung der Dokumente zu reservieren. Dort ist gerade die Umstrukturierung und Hierarchisierung von Themenbereichen leicht möglich, außerdem ist es insbesondere in der kleinen Arbeitsgruppe der *THP* möglich, Zugangsrechte und Schreib-/Leserechte sogar individuell zu vergeben.

Der Umgang damit wurde durch die folgenden Komponenten unterstützt:

- Eine Volltextsuche mit Hilfe der Suchmaschine Verity.
- Die Verschlagwortung der Dokumente und deren Verwaltung in der Datenbank Access.
- Die Speicherung der Schlagworte in den Dokumenten selbst.

Es wurde entschieden, die Suchmaschine Verity einzubinden. Diese Entscheidung war wesentlich davon beeinflusst, das System skalierbar zu gestalten, denn potentiell sollte die Möglichkeit offen gehalten werden, das System auch bei dem Projektpartner *mpe* einzusetzen. Da Verity jedoch ebenfalls kostenintensiv und lizenzgebunden ist, kann die entsprechende Komponente des FreeStylers nur sehr eingeschränkt genutzt werden, da sie anderen Anwenderinnen und Anwendern nicht zur Verfügung steht.

Trotzdem hat sich die Integration von Suchergebnissen in die Visualisierung als sehr sinnvoll herausgestellt, weshalb im Weiteren ein Konzept angestrebt werden sollte, das ohne kommerzielle Software auskommt. Die Funktionen, die Verity zur Verfügung stellt, sind außerdem sehr umfangreich und konnten nicht annähernd ausgenutzt werden.

Die Einbindung der Datenbank Access war in Bezug auf die Finanzierung und Verfügbarkeit weniger problematisch, da sie zumindest unter Windows vielen Personen vorliegt. Dass sie trotzdem wenig genutzt wird, scheint im Wesentlichen mit der fehlenden organisatorischen Einbindung zusammenzuhängen. Allerdings gibt es auch konzeptuelle Probleme mit der Anbindung der Datenbank: Die Dokumente des Repositories, sind schwierig zu verfolgen (verschieben, umbenennen). Werden Referenzen auf Dokumente in die Datenbank aufgenommen, so führt das schnell zu

unkorrekten Inhalten. Für dieses Problem müsste ein Konzept für den Abgleich entwickelt werden.

In Bezug auf das Löschen von Dokumenten ist wiederum ein Vorteil in diesem Ansatz zu sehen, da so auch Dokumentinformationen in der Datenbank verbleiben können, zu denen die Dokumente nicht mehr existieren. So könnten in Bezug auf Trends auch Entwicklungen abgeleitet werden.

Der Überarbeitungsprozess. Der Dokumentationsprozess, der durch den FreeStyler ermöglicht wird, soll kooperativ stattfinden, das heißt, dass die Dokumente sowohl synchron als auch asynchron bearbeitet werden. An dieser Stelle müsste einerseits evaluiert werden, ob unabgesprochene Änderungen in einer Gruppe akzeptiert werden. Es ist aber wahrscheinlich, das die Veränderungen zumindest nachvollziehbar gestaltet werden müssen.

Möglichkeiten dazu bestehen in der Erstellung einer Historie oder in der Einführung einer Autorschaft mit Zugriffsrechten, die zwar anderen Kommentare erlaubt aber von den Autoren oder Autorinnen eine Zustimmung zu Änderungen benötigt. Da explizit mehrere Personen kooperativ an den Dokumenten arbeiten, müsste sich eine solche Zustimmung und Autorschaft auf Objekte nicht auf Dokumente beziehen.

Eine reine Versionsverwaltung bei Änderungen erscheint auf den ersten Blick nicht ausreichend, da es möglich sein sollte, die verschiedenen Änderungen in einem Dokument zu sehen, um sie so konstruktiv integrieren oder bearbeiten zu können.

8 ZUSAMMENFASSUNG UND AUSBLICK

8.1 Zentrale Aspekte

Wissensmanagement

Wissen und wichtige Informationen zu verwalten, hat sich zu einem der zentralen Themen der Informationsgesellschaft entwickelt. Besondere Aktualität hat es im Zusammenhang mit Web-Technologien erfahren, über die Inhalte im gewaltigen Umfang miteinander vernetzt sind. Mit diesen Technologien steigt einerseits der Bedarf, Inhalte wiederzuverwenden und neu zu kombinieren. Andererseits wird deutlich, dass diese Medien sehr statisch sind, das heißt, dass mehr oder weniger fixe Informationen zur Rezeption aufbereitet werden. Solche statischen Medien, sind nicht ausreichend für die Wissenskommunikation geeignet.

Web-Inhalte zu erstellen ist für Computer-Laien oft zu schwer, für andere oft zu aufwändig, als dass sie allgemein für Wissensdarstellungen verwendet werden könnten. Außerdem sind Web-Technologien bisher noch nicht so handhabbar, dass Wissen schnell, dynamisch und kooperativ aufbereitet werden kann. Wissenskommunikation findet bestenfalls über eigens dafür vorgesehene Kommunikationskanäle, wie Newsgroups oder Chats statt, die dann wiederum wenig Bezug auf Materialien nehmen und kaum die Erstellung gemeinsamer Wissensprodukte erlauben.

Insbesondere in wissensintensiven Arbeitsbereichen hat sich jedoch ein Bedarf an Wissensmanagement entwickelt, der nicht unbedingt geeignet mit den bestehen Möglichkeiten gedeckt werden kann. Der Bedarf im Wissensmanagement zielt ab auf:

- die Verbreitung von Informationen und Wissen,
- den Zugriff auf Informationen und Wissen,
- die Erhebung von Informationen und Wissen und
- die Darstellung von Informationen und Wissen.

Erreicht werden soll damit häufig

- die Abkopplung des Wissens von einzelnen Personen,
- die Einschätzbarkeit des vorhandenen Wissens in Bezug auf Umfang und Wert,
- die Dokumentation des Wissens

- und damit die Wiederverwendbarkeit des Wissens, beispielsweise in Designprozessen.

Dynamisches Wissen

Wie diese Ziele erreicht werden können wird viel diskutiert. Zu Beginn dieser Arbeit wurden Widersprüche und Unzulänglichkeiten beim Wissensmanagement angesprochen, die darin bestehen, dass weitgehend unklar ist, wie geeignetes Wissen überhaupt erhoben werden kann, wie es insbesondere integriert in den alltäglichen Arbeitsprozess erhoben werden soll, wie es dargestellt werden soll und welche Ausschnitte dargestellt werden sollen. In der Konsequenz wird deshalb tendenziell auf allgemeine und langlebige Fakten fokussiert, wie Personendaten und –profile, Angebotspaletten etc., und das dynamische Wissen wenig beachtet. Das steht insofern im Widerspruch zum Wissensmanagement, als dass oft gerade kurzlebiges Wissen oder auch schwer darstellbares Erfahrungswissen expliziert werden soll, da gerade davon ein Mehrwert erwartet wird. In dieser Arbeit wurde deshalb danach gefragt,

- wie dieses dynamische Wissen erhoben und dokumentiert werden kann,
- wie es möglich ist, Wissenselaboration sowie Ergebnisdokumentation in den Arbeitsalltag zu integrieren,
- wie Wissenselaboration durchgeführt werden kann und
- mit welchen Mitteln sie unterstützt werden kann.

Wissenselaboration

Der Prozess, beginnend bei der Externalisierung von Wissen, über eher informelle Darstellungen zu formaleren oder stärker strukturierten Repräsentationen hin zur Dokumentation wurde in dieser Arbeit als Prozess der Wissenselaboration bezeichnet. Die Reflexion, die für diese Überarbeitungsschritte notwendig ist, unterteilt sich immer wieder in Phasen der Externalisierung, der Internalisierung und der synchronen sowie asynchronen kooperativen Überarbeitung der Inhalte. Bei einer dynamischen und zyklischen Wissenselaboration müssen die Darstellungen für die schnelle Überarbeitung geeignet sein.

Es wurden zwei grundlegende Anforderungen bei der Unterstützung der Wissenselaboration herausgearbeitet und beleuchtet:

- Die Unterstützung eines *kreativen* Herangehens, das die Plausibilität der Ergebnisse verbessern soll.
- Die Unterstützung eines *dokumentarischen* Herangehens, das die Nachhaltigkeit der Ergebnisse verbessern soll.

Diskussionen

Es wurde dafür argumentiert, dass Diskussionen ein geeignetes Szenario sind, um die Wissenselaboration anzustoßen, sofern es darin zur Wissenskommunikation kommt. Die Gründe, die dafür sprechen sind die Folgenden:

- In Diskussionen werden implizit Vorgehensweisen gewählt, die mit Lernverfahren verglichen werden können, die also den Erkenntnisgewinn fördern.
- Es handelt sich um einen reflexionsorientierten Prozess.
- Es ist eine gewohnte Option, in Diskussionen Externalisierungen anzufertigen.
- Diskussionen müssen nicht künstlich in einen Arbeitsablauf integriert werden, sondern werden bereits zu einem hohen Prozentsatz praktiziert.

Trotzdem war festzustellen, dass sich bisher Sitzungsunterstützungssysteme kaum etablieren konnten. Dafür werden u.a. die folgenden Gründe verantwortlich gemacht, die allerdings nicht statistisch abgesichert sind:

- Die bestehenden Systeme kommen hauptsächlich aus dem Universitätskontext und haben selten den Produktstatus erreicht.
- Es handelt sich vielfach um sehr große und geschlossene Systeme, deren Materialien und unzureichende Schnittstellen sich meistens schlecht in bestehende Organisationsstrukturen oder -systeme integrieren lassen.
- Eine schlechte Integration führt typischerweise zu Medienbrüchen und zu einem zu geringen Mehrwert oder erschwert sogar Arbeitsprozesse, die eigentlich durch die Sitzungsunterstützung erleichtert werden sollen.
- In einen Dokumentationsprozess müssen individuelle Notizen, Vor- und Nachbereitungen einbezogen werden. Auch hier sind Medienwechsel hinderlich.
- In einem Dokumentationsprozess sollten alle Phasen gleich zu digitalisierten Ergebnissen führen, um eine doppelte Verwaltung von Notizen (herkömmlich und digital) zu vermeiden. Dazu müssen die Eingabemedien leicht transportabel sein (geringe Größe und Gewicht), wie beispielsweise „Handhelds“, und die Daten zum allgemeinen Dokumentationsansatz interoperabel gestaltet sein.
- Im Falle einer digitalen Notizverwaltung fällt der dauerhaften Datenverwaltung ein hohes Gewicht zu (Aufwärtskompatibilität, Wartung, Standards). Dies steht aus organisationalen Gründen oft im Konflikt zur Verwendung von Universitätssystemen.
- Vielerorts wird die Bedeutung von Diskussionen unterschätzt, so dass nicht in das sinnvolle Equipment für ein Diskussionsszenario investiert wird, obwohl darunter als minimale Voraussetzung nur eine elektronische Tafel zu verstehen ist, die in einer einfachen Variante nicht übermäßig kostspielig ist.

- Trotz dieser einfachen Anreicherung der herkömmlichen Diskussionssituation beeinflusst eine Externalisierung den Kommunikationsprozess und muss eingeübt werden z.B. in Bezug darauf, an welchen Stellen Externalisierungen angebracht sind, wie sich die Beteiligten darauf beziehen, wer wann Notizen anfertigt und wie die Notizen zugänglich sind. Eine Fehleinschätzung davon kann die Verwendung entsprechender Technologien im Ansatz beenden.

Diskussionsunterstützung wurde in dieser Arbeit auf die Mittel und Methoden bezogen, die zum Externalisieren, Strukturieren und Dokumentieren der Gesprächsinhalte verwendet werden; der Einfluss auf die Inhalte ist davon explizit ausgenommen worden.

Diskussionsunterstützung

- Es wird davon ausgegangen, dass sich visuelle Sprachen eignen zur:
  - Externalisierung,
  - Strukturierung,
  - semi-formalen Darstellung,
  - Restrukturierung,
  - Umsetzung epistemischer Formen.
- Weiterhin wird davon ausgegangen, dass Externalisierungen
  - grundlegender Bestandteil der kooperativen Wissenselaboration sind,
  - je nach epistemischer Form oder epistemischem Spiel zur Reflexion geeignet sind,
  - als Kommunikationsmedien zu verstehen sind.

8.2 Diskussion

Tabelle 21 und Tabelle 22 ordnen die Systeme FreeStyler und DiscBoard den Eigenschaften zu, die bereits in Kapitel 3.4 zum Systemvergleich herangezogen wurden. Die augenscheinlichen Unterschiede beider Systeme spiegeln sich wesentlich in der Flexibilität und Art der Ausdrucksmittel, die die Systeme bieten. Während das DiscBoard die Arbeitweisen stark strukturiert, lässt der FreeStyler weitgehend offen wie er verwendet wird. Diese Flexibilität macht das Tool durchlässig für unterschiedlichste Arbeitskontexte, insbesondere solche, die keine strukturierten Arbeitsmethoden voraussetzen.

Welche visuelle Sprache auch immer verwendet wird, besteht doch oft der Bedarf über die Repräsentation zu reden, sich über Vorgehensweisen abzustimmen oder generelle

Kommentare zu notieren. Dazu werden oft übergeordnete Ausdrucksweisen benötigt oder verwendet, die nicht in der gerade verwendeten visuellen Sprache formulierbar sind.

Software/Tool	„epistemic games“ „epistemic forms“	Adaptier-barkeit	Feedback	Handschrift-liche Eingaben oder freier Text	Flexibilität der Ausdrucks-mittel
DiscBoard	+	–	–	–	Ø
FreeStyler	Ø	–	–	+	+
Belvedere	Ø	–	+	–	–
Chips	+	+	–	+	+
Cognoter/Argnoter	Ø	–	–	–	–
Csile	–	–	–	+	Ø
Dolphin	–	–	–	+	+
gIBIS	Ø	–	–	–	–
MindManager	Ø	–	–	–	–
OneNote	–	–	–	+	–
Sepia	+	–	–	–	Ø

Tabelle 21

Eigenschaften der Tools.

Bei der Entwicklung der Tools wurde die These verfolgt, dass das externalisierte Wissen von eher informellen Darstellungen hin zu stärker strukturierten Darstellungen entwickelt wird. Der FreeStyler bietet dazu die Möglichkeit der handschriftlichen Eingabe, bei der keinerlei methodisches Wissen oder Herangehen vorausgesetzt wird. Dies scheint ein wichtiges Kriterium dafür zu sein, ob ein Tool überhaupt in einer Kommunikations-situation eingesetzt wird. Der Druck, der durch methodische Herangehensweisen ausgeübt wird, überlastet offenbar schon viele Gesprächssituationen.

Eine solche Belastung kann aus der Untersuchung von Suthers (1999A, 1999B, 1999C) abgeleitet werden, nach der externe Darstellungen nicht so sehr die Ergebnisse von Kommunikationssituationen darstellen, als dass sie als Stimulus dafür verwendet werden. Dieser Stimulus geht dann unter Umständen weniger von einer gut strukturierten Darstellung aus als von Inkonsistenzen, die dann mündlich geklärt werden. Das könnte bedeuten, dass das gemeinsame, ergebnisorientierte Strukturieren der Intention einer Diskussion widerspricht und diese deshalb zu stark behindert.

Allerdings hat es sich auch gezeigt, dass eine Kommunikationssituation, die durch technische Hilfsmittel angereichert ist, neu erlernt werden kann, dass es insbesondere Regeln im Umgang mit den Tools geben muss. Hilfreich können dabei explizite Strategien sein, wie sie sich in der Vorstudie (Kapitel 4.4) herausstellten. Danach ist es sinnvoll, Diskussionsphasen, Reflexionsphasen und Notizphasen explizit voneinander zu trennen und dies unter Umständen auch anzuleiten.

Um diese Aspekte in die Tabelle zu integrieren, wurde als zusätzliche Eigenschaft mit aufgenommen, wie flexibel die Ausdrucksmittel sind. Diese Kategorie müsste allerdings

noch weitergehend präzisiert werde. Eine Aspekt dazu besteht mindestens darin, sowohl unstrukturiert als auch strukturiert mit einem Tool umgehen zu können, ohne Beiträge sofort klassifizieren zu müssen und ohne sie sofort in eine gegebene Struktur einbinden zu müssen. Unter der Flexibilität der Ausdrucksmittel ist eine Beurteilung darüber zu verstehen, ob ein solcher Übergang von informellen zu strukturierten Darstellungen unterstützt wird, ob also weniger formale Ausdrucksweisen und stärker strukturierte Darstellungen gleichermaßen möglich sind.

Danach bestätigt sich eine Tendenz, dass entweder visuelle Sprachen direkt zur Strukturierung eingesetzt werden (Belvedere, Sepia, MindManager, gIBIS), oder dass eher auf die flexible Eingabe Wert gelegt wird (OneNote). Die Systeme Chips und Dolphin erlauben eine Integration von freiem Text (Chips) und Handschrift (Dolphin) zusammen mit anderen Strukturen. Im Csile-System manifestiert sich dieser Übergang weniger in den Darstellungsmitteln als im Überarbeitungsprozess.

In Tabelle 22 wird die Ausrichtung des FreeStylers in Richtung auf eine langfristige Dokumentation und den Aufbaus eines Repositories deutlich. Die positiven Bewertungen sind allerdings insofern einzuschränken, als dass die jeweiligen Eigenschaften nur in einem begrenztem Ausmaß unterstützt werden. Die Interoperabilität wird einerseits über die Verwendung von Java begründet, dass viele Schnittstellen bietet, um beispielsweise Datenbanken einzubinden, XML-Ansätze unterstützt sowie die Implementierung verteilter Systeme ermöglicht.

Die Kombination, Volltextsuchen und Verschlagwortungen in Zusammenhang mit einer einfachen Dateistruktur zu verbinden, wird als Ansatz für den Aufbau eines Repositories gewertet. Notifikationen wurden im FreeStyler zwar ermöglicht, indem Inhalte direkt und leicht verschickt werden können. Die Usability müsste dafür jedoch deutlich verbessert werden. Insbesondere die Interoperation mit anderen Tools, die beispielsweise zur Adressverwaltung eingesetzt werden, müsste hier unterstützt werden.

Als besonders positiv und in dieser Form sonst kaum vertreten, ist die Möglichkeit herauszuheben, auf sehr einfache Weise Hyperlinks auf Dokumente anzulegen. Es ergibt sich daraus auch für Computer-Laien die Option, Dokumente bzw. Daten aus dem Repository, anderen Archiven bzw. dem Web zusammenzutragen und bedarfsorientiert einzubringen.

Die Einbettung in den nicht-technischen Arbeitsprozess bezieht sich darauf, dass der FreeStyler in unterschiedlichen Arbeitskontexten (z.B. Vorbereitung, Sitzung, Nachbereitung, etc.) eingesetzt werden kann. Es werden damit viele Situationen in der

Transferstelle Hochschule Praxis (*THP*) abgedeckt, in denen eine Externalisierung oder Visualisierung sinnvoll ist.

Organisatorische Einbettung	Interoperabilität	Repositories	Information-retrieval	Notifikation	Reflexion über den Wissensbestand	Einbettung in nicht-technischen Arbeitspozess
DiscBoard	–	–	–	–	–	+
FreeStyler	+	+	+	+	+	+

Tabelle 22

Organisatorische Einbettung der Tools.

Das DiscBoard sollte in erster Linie die konkrete Wissenskonstruktion unterstützen. In diesem Sinn ist die Reflexion zwar die Zielrichtung des Tools, es wird aber nicht unterstützt, externe Daten zu integrieren.
Die Adaption der visuellen Sprachen an die Diskussionssituation ist eine deutliche Orientierung an einem nicht-technischen Arbeitsprozess und wird deshalb positiv bewertet.

8.3 Ausblick

Aus den vorangegangenen Darstellungen ergeben sich drei Tendenzen für nächste Entwicklungsschritte:

- Integration der Konzepte des DiscBoards und des FreeStylers.
- Ausweitung der „epistemic forms“ auf „epistemic games“, also auch eine Anleitung konkreter Methoden.
- Verbesserte Unterstützung des Dokumentationsprozesses, z.B. durch automatisches Generieren von Webpages aus den Darstellungen.

Der FreeStyler bietet das Potential, über die Paletten verschiedene visuelle Sprachen anzubieten. Mit geringem Aufwand können die Paletten aus dem DiscBoard übertragen werden. Damit wäre eine Umstellung von farblich kodierten Objekten zu solchen verbunden, die vor allem Umrisse (verbunden mit Symbolen und Farben) verwenden, um die Art der Nutzung zu verdeutlichen. Flexible Möglichkeiten zum Design der Kanten existieren bereits.
Über die Paletten können auch konkrete Methoden angeleitet werden, indem Feedback über die Verwendung der Objekte angeboten wird.
Handschriftliche Eingaben werden bereits unterstützt und könnten so direkt mit den visuellen Sprachen kombiniert werden.

Interessant wäre auch die Integration weiterer Perspektiven. Dazu könnte die Interpretation der konstruierten Daten in Bezug auf eine Interessenslage gehören. In einer einfachen Variante könnten darin Autoren und Autorinnen in Bezug zu Topics gesetzt werden, evtl. sogar mit einer quantitativen Gewichtung.

Für die organisatorische Einbettung ist unbedingt eine Überarbeitung der Verschlagwortung notwendig, bzw. vorangehend die Entwicklung eines verbesserten Konzeptes dazu. Die Suchmaschine Verity sollte wenn möglich durch eine „public domain"-Lösung ersetzt werden.

Die Integration des DiscBoard-Ansatzes mit dem FreeStyler würde allerdings auch verschiedene Probleme nach sich ziehen:

Das Seitenformat des FreeStylers hat sich besonders für Mitschriften als sinnvoll, wenn nicht sogar notwendig erwiesen. Die Workspace- und „Topic"-Organisation im DiscBoard müsste darauf übertragen werden, was jedoch nicht unmittelbar möglich ist.

Die Filter und Abbildungen im DiscBoard basieren wesentlich auf dem Monitoring der Dalis-Komponente. Bei einer Übertragung dieses Ansatzes auf den FreeStyler müsste ein geeignetes Format für die Regeln (ehem. Prolog) entworfen werden, auf dem die Filter und Abbildungsvorschriften agieren können. Generell müsste ein gezielter Zugriff auf Daten nach Workspaces, Typ, Autor, etc. möglich sein. Der MatchMaker TNG hält diese Daten bereits weitgehend vor.

9 LITERATURVERZEICHNIS

AKEHURST, D.H. (2000). An OO visual language definition approach supporting multiple views. In *Proceedings of IEEE International Symposium on Visual Languages* (S. 57-59), Seattle, Washington. http://www.cs.ukc.ac.uk/pubs/2000/1097/

ARONSON, E. (1978). *The Jigsaw Classroom*. Beverly Hills, CA: Sage.

BAKER, M. J. (1995). Negotiation in collaborative problem-solving dialogues. In Beun, R.-J., Baker, M.J. & Reiner, M. (Hrsg.), *Dialogue and instruction (NATO ASI Series, Vol. 142)* (S. 39-55). Berlin Heidelberg: Springer-Verlag.

BAKER, M., HANSEN, T., JOINTER, R. & TRAUM, D. (1999). The role of grounding in collaborative learning tasks. In Dillenbourg, P. (Hrsg.), *Collaborative learning: Cognitive and computational approaches* (S. 31-63). Elsevier Science Publishers B.V.

BAKER, M.J. & LUND, K. (1996). Flexibly structuring the interaction in a CSCL environment. In Brna, P., Paiva, A. & Self, J. (Hrsg.), *Proceedings of the European Conference on Artificial Intelligence in Education* (S. 401-407). Lissabon (Portugal), September 1996.

BAKER, M., VRIES, DE, E., LUND, K. & QUINGNARD, M. (2001). Computer-mediated epistemic interactions for co-constructing notions: Lessons learned from a five-year research program. In Dillenbourg, P., Eurelings, A. & Hakkarainen, K. (Hrsg.), *Proceedings of European Conference on Computer-Supported Collaborative Learning* (S. 89-96). University of Maastricht, Netherlands.

BALOIAN TATARYAN, N.-A. (1997). *Strukturierte Erstellung und kooperative Nutzung von Instruktionsmaterial in einem computerintegrierten Klassenraum.* Dissertation am Fachbereich Mathematik der Gerhard-Mercator-Universität – Gesamthochschule Duisburg.

BARGIELA-CHIAPPINI, F. & HARRIS, S.J. (1997). *Managing Language, The discourse of corporate meetings*. Amsterdam: John Benjamins Publishing Company.

BERND, H., HIPPCHEN, T., JÜNGST, K.-L. & STRITTMATTER, P. (2000). Durcharbeiten von Begriffsstrukturdarstellungen in unterrichtlichen und computergestützten Lernumgebungen. In Mandl, H. & Fischer, F. (Hrsg.), *Wissen sichtbar machen. Wissensmanagement mit Mapping-Techniken.* Göttingen: Hogrefe.

BINKELMANN, P. (1993). *Entwicklung der Gruppenarbeit in Deutschland: Stand und Perspektiven*. Frankfurt: Campus Verlag.

BOLLEN, L. (2001). *Integration einer visuellen Modellierungssprache in eine kooperative Diskussionsumgebung für den naturwissenschaftlichen Unterricht.* Unpublished Master Thesis, University of Duisburg, Dept. of Mathematics / Computer Science.

BORGHOFF, U.M. & PARESCHI, R. (1998). *Information technology for knowledge management*. Berlin, Heidelberg, New York: Springer Verlag.

BORMANN, E.G. (1968). *Discussion and group methods* (second edition). New York: Harper & Row, Publishers, Inc.

BOTTONI, P., COSTABILE, M.F., LEVALDI, S. & MUSSIO, P. (1995). Formalising visual languages. In Haarslev, V. (Hrsg.), *Proceedings of 11th International IEEE Symposium on Visual Languages* (S.43-52). Los Alamitos, California: IEEE Computer Society Press.
http://www.computer.org/conferences/vl95/talks/T6.html

BOXTEL, VAN, C. & VEERMAN, A. (2001). Diagram-mediated collaborative Learning; Diagrams as tools to provoke and support elaboration and argumentation. In Dillenbourg, P., Eurelings, A. & Hakkarainen, K. (Hrsg.), *Proceedings of European Conference on Computer-Supported Collaborative Learning,,* (S. 131-138). University of Maastricht, Netherlands.

BROMME, R., NÜCKLES, M. & RAMBOW, R. (1999). Adaptivity and anticipation in expert-laypeople communication. In Brennan, S.E., Giboin, A. & Traum, D. (Hrsg.), *Psychological models of communication in collaborative systems. AAAI Fall Symposion Series* (S. 12-24). Menlo Park, CA: AAAI.

BUCKINGHAM SHUM, S. & HAMMOND, N. (1994). Argumentation-based design rationale: What use at what cost?. *International Journal Human-Computer Studies*, *40*, 603-652.

CARLSON, P. A. (1995). Broca: A computerized environment for mediating scientific reasoning through writing. *Journal of Universal Computer Science*, *1(8)*, 571-590.

CHI, M.T.H., DE LEEUW, N., CHIU, M.-H. & LAVANCHER, C. (1994). Eliciting self-explanations improves understanding. *Cognitive Science*, *18*, 439-477.

COLLINS, A. & FERGUSON, W. (1993). Epistemic forms and epistemic games: structures and strategies to guide inquiry. *Educational Psychology*, *28(1)*, 25-42.

CONKLIN, J. & BEGEMANN, M.L. (1987). gIBIS: A hypertext tool for team design deliberation. In *Proceedings of Hypertext'87* (S. 247-251). Chapel Hill (North Carolina).

CONKLIN, J. & BURGESS-YAKEMOVIC, K.C. (1996). A process-oriented approach to design rationale. In Moran, T.P. & Carroll, J.M. (Hrsg.), *Design rationale; concepts, techniques, and use* (S. 393-427). Mahwah (New Jersey): Lawrence Erlbaum Associates.

COOKE, N., J. (1994). Varieties of knowledge elicitation techniques. *International Journal Human-Computer Studies*, *41*, 801-849.

COULOURIS, G., DOLLIMORE, J. & KINDBERG, T. (2001). *Distributed systems, concepts and design* (third edition). Harlow, England: Pearson Education Limited. First and second edition: London, New York: Addison-Wesley.

CRIMI, C., GUERCIO, A., NOTA, G., PACINI, G., TORTORA, G. & TUCCI, M. (1991). Relation grammars and their application to multi-dimensional languages. *Journal of Visual Languages and Computing*, *2*, 233-346.

DEHNBOSTEL, P. (1995). Bedeutungszuwachs des Lernens im Arbeitsprozeß, Regulierungsbedarf oder Deregulierungsnotwendigkeit beruflicher Weiterbildung?. In Dobischat, R. (Hrsg.), *Berufliche Weiterbildung als freier Markt?*. Berlin: Edition Sigma, Rainer Bohn Verlag.

DITTRICH, K. (1997). Datenbanksysteme. In Rechenberg, P. & Pomberger, G. (Hrsg.), *Handbuch der Informatik*. München, Wien: Carl Hanser Verlag.

DOURISH, P. & BELLOTTI, V. (1992). Awareness and coordination in shared work spaces. In Turner, J. & Kraut, R. (Hrsg.), *Proceedings ACM Conference Computer-Supported Cooperative Work* (S. 107-114). Toronto, Canada.

DÜRRENMATT, F. (1980). Der Besuch der alten Dame. Zürich: Diogenes.

ECKERT, A. (2000). Die Netzwerk-Elaborierungs-Technik. In Mandl, H. & Fischer, F. (Hrsg.), *Wissen sichtbar machen. Wissensmanagement mit Mapping-Techniken*. Göttingen: Hogrefe.

ENGESTRÖM, Y., MIETTINEN, R. & PUNAMÄKI, R.-L. (1998). *Perspectives on activity theory.* Cambridge: Cambridge University Press.

FISCHER, F., GRÄSEL, C., KITTEL, A. & MANDL, H. (1996). Entwicklung und Untersuchung eines computerbasierten Mappingverfahrens zur Strukturierung komplexer Information. *Psychologie in Erziehung und Unterrich*, *4/43*, S.266-280.

FISCHER, G., LEMKE, A.C., MCCALL, R. & MORCH, A.I. (1996). Making Argumentation Serve Design. In Moran, T. P. & Carroll, J. M. (Hrsg.), *Design rationale; concepts, techniques, and use* (S. 267-293). Mahwah (New Jersey): Lawrence Erlbaum Associates.

FORRESTER, J. W. (1968). *Principles of Systems*. Waltham, MA (USA): Pegasus Communications.

FOWLER, M. & SCOTT, K. (2000). *UML distilled, A brief guide to the standard object modeling language (second edition)*. Object technology series. Reading, Massachusetts: Addison Wesley.

FRANKE, W. (1990). *Elementare Dialogstrukturen; Darstellung, Analyse, Diskussion*. Tübingen: Max Niemeyer Verlag.

GAINES, B.R. & SHAW, M.L.G. (1993). Knowledge acquisition tools based on personal construct psychology. *The knowledge engineering review*, *8*, 49-85.

GAßNER, K. & HOPPE, H.U. (2000). Visuelle Sprachen als Grundlage kooperativer Diskussionsprozesse. In Mandl, H. & Fischer, F. (Hrsg.), *Wissen sichtbar machen, Wissensmanagement mit Mappingtechniken* (S. 93–118). Göttingen: Hogrefe.

GAßNER, K., HOPPE, H.U., LINGNAU, A. & PINKWART, N. (2003). Handlungsorientierte Kommunikationsmedien als „mind tools“. *Künstliche Intelligenz*, *2/2003*, 42-47.

GAßNER, K., TEWISSEN, F., MÜHLENBROCK, M., LOESCH, A. & HOPPE, H.U. (1998). Intelligently supported collaborative learning environments based on visual languages: a generic approach. In Darses, F. & Zaraté, P. (Hrsg.), *Proceedings of 3rd International Conference on the Design of Cooperative Systems* (S. 47-55). Cannes (Frankreich).

GAßNER, K. & SCHRÖDER, O. (2003). Wissensmodellierung und Wissenskommunikation in Lernszenarien. *Künstliche Intelligenz*, *1/2003*, 5-11.

GRUBER, T.R. & RUSSEL, D.M. (1996). Generative design rationale: Beyond the record and replay paradigm. In Moran, T. P. & Carroll, J. M. (Hrsg.), *Design rationale; concepts, techniques, and use* (S. 323-349). Mahwah (New Jersey): Lawrence Erlbaum Associates.

GULLY, H.E. (1968). *Discussion, conference, and group process* (second edition). New York: Holt, Rinehart and Winston, Inc.

HATANO, G. & INAGAKI, K. (1991). Sharing cognition through collective comprehension activity. In Resnick, L.B., Levine, J.M. & Teasley, S.D. (Hrsg.), *Perspectives on socially shared cognition* (S. 331-348). Washington, DC: American Psychological Association.

HOFFMANN, M. (2002). Vorhersagen und Optionen darstellen – Wie prospektive Mechanismen Zukunftsawareness fördern. In Herczeg, M., Prinz, W. & Oberquelle, H. (Hrsg.), *Proceedings of Mensch & Computer 2002, vom interaktiven Werkzeug zu kooperativen Arbeits- und Lernwelten* (S.245-254). Stuttgart: Teubner

HOPPE, H. U., BALOIAN, N. & ZHAO, J. (1993). Computer support for teacher-centered classroom interaction. In *Proceedings of the International Conference on Computers in Education* (S. 211-217). Taipei (Taiwan), Dezember 1993.

HOPPE, H.U. & GAßNER, K. (2002). Integrating collaborative concept mapping tools with group memory and retrieval functions. In G. Stahl (Hrsg.), *Computer support for collaborative learning: foundations for a CSCL community (Proceedings of CSCL-2002)* (S. 716-725). Boulder (USA), Januar 2002 (distrib. Lawrence Erlbaum).

HOPPE, H.U., GAßNER, K., MÜHLENBROCK, M. & TEWISSEN, F. (2000). Distributed visual language environments for cooperation and learning: Applications and intelligent support. *Group Decision and Negotiation*, *9(3)*, 205-220.

HOPPE, H.U., LINGNAU, A., MACHADO, I., PAIVA, A., PRADA, R. & TEWISSEN, F. (2000). Supporting collaborative activities in computer integrated classrooms - the NIMIS approach. In *Proceedings of 6th International Workshop on Groupware (CRIWG 2000)* (S. 94-103). Madeira (Portugal), IEEE CS Press.

HOPPE, H.U. & TEWISSEN, F. (1994). Gruppen-Lernumgebungen mit computer-gestützter Kommunikation. In Hoppe, H.U. & Luther, W. (Hrsg.), *Neue Lehr-/Lernsysteme aus informatischer und fachdidaktischer Sicht, Schriftenreihe des Fachbereiches Mathematik* (S. 1-10). Universität Duisburg.

JACOBSON, I., BOOCH, G. & RUMBAUGH, J. (1999). *The unified software development process*. Reading, Massachusetts: Addison Wesley.

JANSEN, M. (2002). *Ein Rahmensystem für die Protokollierung und Rücknahme von Benutzeraktionen in replizierten Umgebungen*. Master Thesis, University of Duisburg, Dept. of Mathematics / Computer Science.

JANSEN, M., PINKWART, N. & TEWISSEN, F. (2001). MatchMaker - Flexible Synchronisation von Java-Applikationen. GI-Workshopwoche „Lernen-Lehren-Wissen-Adaptivität". *Forschungsbericht Nr. 763, Universität Dortmund, Oktober 2001*.

JOHNSON, D.W. & JOHNSON, R.T. (1985). The internal dynamics of cooperative learning groups. In Slavin, R., Shara, S., Kagan, S., Lazarowitz, R.H., Webb, C. & Schmuck, R. (Hrsg.), *Learning to cooperate, cooperating to learn* (S. 103-124). New York, London: Plenum Press.

JONASSEN, D.H. (1992). What are cognitive tools?. In Kommers, P.A.M., Jonassen, D.H. & Mayes, J.T. (Hrsg.), *Cognitive tools for learning*. Berlin: Springer Verlag.

JONASSEN, D.H. & CARR, C.S. (2000). Mindtools: Affording multiple knowledge representations for learning. In Lajoie, S. P. (Hrsg.), *Computers as cognitive tools.* Mahwah, New Jersey: Lawrence Erlbaum Associates.

JÜNGST, K.L. & STRITTMATTER, P. (1995). Wissensstrukturdarstellung: Theoretische Ansätze und praktische Relevanz. *Unterrichtswissenschaft*, *3*, 194-207.

JUNGBLUTH, V. (1998). Ideenmaschinen, Kreativitätswerkzeuge im Vergleich. *c't*, *20/98*, 142-147.

KAGAN, S. (1985). Dimensions of cooperative classroom structures. In Slavin, R., Shara, S., Kagan, S., Lazarowitz, R.H., Webb, C. & Schmuck, R. (Hrsg.), *Learning to cooperate, cooperating to learn* (S. 67-96). New York, London: Plenum Press.

KAY, A. (1998). The computer "Revolution" hasn't happened yet!. Talk at Conference on Information Technology in Higher Education, Orlando, Florida. *http://www.educause.edu/conference/e98/webcast98.html*

KELLY, G.A. (1970). A brief introduction to personal construct theory. In Bannister, D. (Hrsg.), *Perspectives in personal construct theory* (S. 1-29). London: Academic Press.

KREMER, R. (1998). Visual languages for knowledge representation. In *Online Proceedings of the Eleventh Workshop on Knowledge Acquisition, Modeling and Management*. Banff (Canada). *http://ksi.cpsc.ucalgary.ca/KAW/KAW98/kremer*

KRUSE, C.-H. (1995). Das „lernende Unternehmen" als strategische Antwort auf den Strukturwandel. In Diepold, P. (Hrsg.), *Lernen im Aufbruch, Strukturwandel und Weiterbildung in Europa (Anstöße, Band 12)*. Frankfurt am Main: Gesellschaft zur Förderung Arbeitsorientierter Forschung und Bildung.

KUNZ, W. & RITTEL, H. (1970). Issues as Elements of Information Systems. Report S-78-2, Institut für Grundlagen der Planung, Universität Stuttgart. *http://www.igp.uni-stuttgart.de/*

LAJOIE, S.P. (1993). Computer environments as cognitive tools for enhancing learning. In Lajoie, S.P. & Derry, S.J. (Hrsg.), *Computers as cognitive tools*. Hillsdale: Lawrence Erlbaum Associates.

LAKIN, F. (1990). Visual languages for cooperation: A performing medium approach to systems for cooperative work. In J. Galegher, R. Kraut & C. Egido (Hrsg.), *Intellectual teamwork* (S. 453-488). Hillsdale (USA): Lawrence Erlbaum.

LARKIN, J.H. & SIMON, H.A. (1987). Why a diagram is (sometimes) worth ten thousand words. *Cognitive Science*, *11(1)*, 65-99.

LARSON, J.R., CHRISTENSEN, C., FRANZ, T.M. & ABBOTT, A.S. (1998). Diagnosing groups: The pooling, management, and impact of shared and unshared case information in team-based medical decision making. *Journal of Personality and Social Psychology*, *75(1)*, 93-108.

LEE, J. & LAI, K.-Y. (1996). What's in design rationale? In Moran, T.P. & Carroll, J.M. (Hrsg.), *Design rationale; concepts, techniques, and use* (S. 267-293). Mahwah (New Jersey): Lawrence Erlbaum Associates.

LEHMANN, G. (1998). Grundlagen der Kommunikation; Die Moderation. In Lehmann, G. & Reinert, G.-B. (Hrsg.), *Gestalten kommunikativer Situationen, Band 2*. Frankfurt am Main: Peter Lang, Europäischer Verlag der Wissenschaften.

LEVIN, J.A. & MOORE, J.A. (1977). Dialogue-Games: Metacommunication structures for natural language interaction. *Cognitive Science*, *1(4)*, 395-420.

LÖWGREN, J. (1994). Empirical foundations for design rationale as user-interface design support. In Oppermann, R., Bagnara, S. & Benyon, D. (Hrsg.), *Proceedings of 7th European Conference on Cognitive Ergonomics* (S. 305-310). St. Augustin (Germany).

LOHSE, G.L., BIOLS, K., WALKER, N. & RUETER, H.H. (1994). A classification of visual representations. *Communications of the ACM*, *37(12)*, 36-49.

MACLEAN, A., YOUNG, R. M., BELLOTTI, V. & MORAN, T. (1991). Questions, options, and criteria: elements of design space analysis. *Human-Computer Interaction*, *6(3&4)*, 201-250.

MANDL, H. & FISCHER, F. (2000). Mapping-Techniken und Begriffsnetze in Lern- und Kooperationsprozessen. In Mandl, H. & Fischer, F. (Hrsg.), *Wissen sichtbar machen. Wissensmanagement mit Mapping-Techniken*. Göttingen: Hogrefe.

MCGRATH, J.E. (1991). Time, interaction, and performance (TIP): A theory of groups. *Small Group Research*, 22 (2), 147-174.

MCKERLIE, D. & MACLEAN, A. (1994). Reasoning with design rationale: Practical experience with design space analysis. In *Design Studies*, *15(2)*, 214-226.

MEACK, H. (1990). *Das Zielbezogene Gespräch* (zweite Auflage). Düsseldorf: VDI-Verlag GmbH.

MEIßNER, I. (1994). *Argumentation in natürlicher Sprache*. Europäische Hochschulschriften. Frankfurt am Main: Peter Lang, Europäischer Verlag der Wissenschaften.

MICHALSKI, R. (1986). Understanding the nature of learning. In Michalski, R. and Carbonell, J (Hrsg.), *Machine Learning – An artificial intelligence approach (VolumeII)*. Los Altos, Californien: Morgan Kaufman.

MORAN, T.P. & CARROLL, J.M. (1996). Overview of design rationale. In Moran, T.P. & Carroll, J.M. (Hrsg.), *Design rationale; concepts, techniques, and use* (S. 1-19). Mahwah (New Jersey): Lawrence Erlbaum Associates.

MÜHLENBROCK, M. (2001). *Action-based collaboration analysis for group learning*. Berlin, Amsterdam: Infix.

MÜHLENBROCK, M. & HOPPE, H.U. (1999). Computer supported interaction analysis of group problem solving. In J. Roschelle & Hoadley, C. (Hrsg.), *Proceedings of the Conference on Computer Supported Collaborative Learning (CSCL-99)* (S. 398-405). Palo Alto, CA.

MÜHLENBROCK, M. & HOPPE, H.U. (2001). A collaboration monitor for shared workspaces. In Moore, J.D., Redfield, C.L., & Johnson, W.L. (Hrsg.), *Proceedings of the International Conference on Artificial Intelligence in Education (AIED-2001*) (S. 154-165). Amsterdam: IOS Press.

MÜHLENBROCK, M. & LOESCH, A. (1998). Eine Architektur zur Unterstützung kollaborativen Lernens in offenen Umgebungen. In Schröder, O. (Hrsg.), *Proceedings zum Workshop der 22. Deutschen Jahrestagung für Künstliche Intelligenz 1998, Intelligente Unterstützung in offenen Lehr-/Lernsystemen, TZI Bericht Nr. 8 der Universität Bremen*. Bremen (Germany), September 1998.

MÜHLENBROCK, M., TEWISSEN, F. & HOPPE, H.U. (1997). A framework system for intelligent support in open distributed learning environments. In du Boulay, B. & Mizoguchi, R. (Hrsg.), *Artificial intelligence in education: Knowledge and media in learning systems* (S. 191-198). Amsterdam (The Netherlands): IOS Press.

MURRAY, B.S. & MCDAID, E. (1993). Visualizing and representing knowledge for the end user: a review. *International Journal Man-Machine Studies, Special Issue knowledge based cooperation*, *38 (1)*, 23-49.

MYERS, B.A. (1990). Taxonomies of visual programming and program visualization. *Journal of Visual Languages and Computing*, *1*, 97-123.

NONAKA, I. (1994). A Dynamic Theory of Organisational Knowledge Creation. *Organization Science*, *5 (1)*, 14-37.

NONAKA, I. & TAKEUCHI, H. (1995). *The knowledge creating company: How Japanese companies create the dynamics of innovation*. New York: Oxford University Press.

NYGAARD, K. & SØRGAARD, P. (1987). The perspective concept in informatics. In Bjernes, G., Ehn, P. & Kyng, M. (Hrsg.), *Computers and democracy* (S. 371-391). Aldershot, England: Avebury.

OLSON, G.M., OLSON, J.S., STORRØSTEN, M., CARTER, M., HERBSLEB, J. & RUETER, H. (1996). The structure of activity during design meetings. In Moran, T.P. & Carroll, J.M. (Hrsg.), *Design rationale; concepts, techniques, and use* (S. 217-325). Mahwah (New Jersey): Lawrence Erlbaum Associates.

PAIVA, A., MACHADO, I. & PRADA, R. (2001). Heroes, villains, magicians, …: dramatics personae in a virtual story creation environment. In *Proceedings of the International Conference on Intelligent User Interfaces* (S. 129-136). Santa Fe, New Mexico.

PALINSCAR, A.S. & BROWN, A.L. (1989). Reciprocal teaching for comprehension-fostering and monitoring activities. *Cognition and instruction*, *1*, 117-175.

PEA, R.D. (1985). Beyond amplification: Using the computer to reorganize mental functioning. *Educational Psychologist*, *20 (4)*, 167-182.

PEA, R.D. (1993). Practices of distributed intelligence and designs for education. In Gavriel, S. (Hrsg.), *Distributed cognitions* (S. 47-87). New York: Cambridge University Press.

PEDLER, M., BOYDELL, T. & BURGOYNE, J. (1996). Auf dem Weg zum lernenden Unternehmen. In Sattelberger, T. (Hrsg.), *Die Lernende Organisation, Konzepte für eine neue Qualität der Unternehmensentwicklung* (S. 57-65). Wiesbaden: Gabler.

PERKINS, D.N. (1993). Person-plus: a distributed view of thinking and learning. In Salomon, G. (Hrsg.), *Distributed cognition* (S. 88-110). New York: Cambridge University Press.

PILKINGTON, R. & MALLEN, C. (1996). Dialogue Games to support reasoning and reflection in diagnostic tasks. In *Proceedings of the European Conference on Artificial Intelligence in Education* (S. 213-219). Lisbon (Portugal).

PINKWART, N., HOPPE, H.U. & GAßNER, K. (2001). Integration of domain-specific elements into visual language based collaborative environments. In Borges, M.R.S., Haake, J.M. & Hoppe, H.U. (Hrsg.), *Proceedings of 7th International Workshop on Groupware (CRIWG 2001)* (S. 142-147). IEEE Computer Society: Los Alamitos, California.

PINKWART, N., HOPPE, H.U., BOLLEN, L. & FUHLROTT, E. (2002). Group-oriented modelling tools with heterogeneous semantics. In Cerri, S.A., Gouardères, G. & Paraguacu, F. (Hrsg.), *Lecture Notes in Computer Science 2363, Intelligent Tutoring Systems* (S. 21-30). Berlin: Springer Verlag.

PLÖTZNER, R., HOPPE, H.U., FEHSE, E., NOLTE, C. & TEWISSEN, F. (1996). Model-based design of activity spaces for collaborative problem solving and learning. In *Proceedings of the European Conference on Artificial Intelligence in Education* (S. 372-378). Lisbon (Portugal).

QUATRANI, T. (2000). *Visual modeling with Rational Rose 2000 and UML*. Reading, Masachusetts: Addison Wesley.

REIMANN, P. & ZUMBACH, J. (2001). Design, Diskurs und Reflexion als zentrale Elemente virtueller Seminare. In Hesse, F. & Friedrich, F. (Hrsg.), *Partizipation und Interaktion im virtuellen Seminar* (S. 135-163). München: Waxmann.

REKERS, J. (1994). On the use of graph grammars for defining the syntax of graphical languages. Colloquium on graph transformation and its application in computer science. Palma de Mallorca, Spain, March 1994 oder *technical report 94-11*, Leiden University.
http://www.liacs.nl/TechRep/1994/tr94-11.html

RESNICK, L.B. (1991). Shared cognition: thinking as social practice. In Resnick, L.B., Levine, J.M. & Teasley, S.D. (Hrsg.), *Perspectives on socially shared cognition* (S. 1-20). Washington, DC: American Psychological Association.

RITTER, A. & ZINK, K.J. (1994). Differenzierte Kleingruppenkonzepte als wesentlicher Bestandteil eines umfassenden, integrierenden Qualitätsmanagements (im Sinne von TQM). In Zink, K.J. (Hrsg.), *Qualität als Management-Aufgabe* (S. 245-273). Landsberg/Lech: Verlag Moderne Industrie. 3.Auflage.

ROBERTSON, J., GOOD, J. & PAIN, H. (1998). BetterBlether: The design and evaluation of a discussion tool for education. *International Journal of Artificial Intelligence in Education*, *9*, 219-236.

ROSCHELLE, J. & TEASLEY, S. (1995). The construction of shared knowledge in collaborative problem solving. In O'Malley (Hrsg.), *Computer supported collaborative learning (NATO ASI Series, Vol. 128)* (S. 69-96). Berlin Heidelberg: Springer Verlag.

SATTELBERGER, T. (1996). Personalentwicklung neuer Qualität durch Renaissance helfender Beziehungen. In Sattelberger, T. (Hrsg.), *Die Lernende Organisation, Konzepte für eine neue Qualität der Unternehmensentwicklung* (S. 207-227). Wiesbaden: Gabler.

SCARDAMALIA, M. & BEREITER, C. (1992). An architecture for collaborative knowledge building. In De Corte, E., Linn, M.C., Mandl, H. & Verschaffel, L. (Hrsg.), *Computer-Based learning environments and problem solving (NATO ASI Series, Vol. 84)* (S. 41-66). Berlin Heidelberg: Springer Verlag.

SCARDAMALIA, M. & BEREITER, C. (1994). Computer support for knowledge-building communities. *The Journal of the Learning Sciences, 3(3*), 265-283.

SCARDAMALIA, M., BEREITER, C., BRETT, C., BURTIS, P.J., CALHOUN, C. & SMITH LEA, N. (1992). Educational applications of a networked communal database. *Interactive Learning Environments*, *2(1)*, 45-71.

SCARDAMALIA, M., BEREITER, C. & STEINBACH, R. (1984). Teachability of reflective processes in written composition. *Cognitive Science*, *8*, 173-190.

SCHEELE, B. & GROEBEN, N. (1988). *Dialog-Konsens-Methoden zur Rekonstruktion subjektiver Theorien.* Tübingen: Francke Verlag.

SCHMUCK, R. (1985). Learning to cooperate, cooperating to learn: Basis concepts. In Slavin, R., Shara, S., Kagan, S., Lazarowitz, R.H., Webb, C. & Schmuck, R. (Hrsg.), *Learning to cooperate, cooperating to learn* (S. 1-4). New York, London: Plenum Press.

SIMON, H. (1983). Why should machines learn? In Michalski, R., Carbonell, J. & Mitchell, T. (Hrsg.), *Machine learning: An artificial intelligence approach* (S. 25-38). Tioga, Palo Alto, CA.

SLAVIN, A., BUCKINGHAM SHUM, S., SIERHUIS, M., CONKLIN, J., ZIMMERMANN, B., PALUS, C., DRATH, W., HORTH, D., DOMINGUE, J., MOTTA, E. & LI, G. (2001). Compendium: Making meetings into konwledge events. *Technical Report of the Knowledge Media Institute, KMI-TR-103*.

http://kmi.open.ac.uk/publications/ oder http://kmi.open.ac.uk/tr/papers/kmi-tr-102.pdf

SPADA, H. & RUMMEL, N. (2002). Modellieren als Methode zur Strukturierung netzbasierter Zusammenarbeit. Vortrag auf dem 43. Kongress der Deutschen Gesellschaft für Psychologie, Humboldt-Universität zu Berlin, 22.-26. September 2002.

STACHOWIAK, H. (1973). *Allgemeine Modelltheorie*. Wien: Springer Verlag.

STAHL, G. (2000). A model of collaborative knowledge-building. In Fishman, B. & O'Connor-Divelbiss (Hrsg.), *Proceedings of the Fourth International Conference of the Learning Sciences* (S. 70-77). Mahwah, NJ: Erlbaum.

STAHL, G. (2001). WEBGUIDE: Guiding collaborative learning on the web with perspectives. *Journal of Interactive Media in Education*, 2001(1).

STAHL, G, (2002). Introduction: foundations for a CSCL community. In G. Stahl (Hrsg.), *Computer support for collaborative learning: foundations for a CSCL Community* (*Proceedings of CSCL 2002)* (S. 1-2). Boulder (USA), Januar 2002 (Lawrence Erlbaum).

STASSER, G., STEWART, D.D. & WITTENBAUM, G.M. (1995). Expert roles and information exchange during discussion: The importance of knowing who knows what. *Journal of Experimental Social Psychology*, *31*, 244-265.

STASSER, G. & TITUS, W. (1985). Pooling of unshared information in group decision making: biased information sampling during disucussion. *Journal of Personality and Social Psychology*, *48(6)*, 1467-1478.

STEFIK, M., FOSTER, G., BOBROW, D.G., KAHN, K., LANNING, S. & SUCHMAN, S. (1987). Beyond the chalkboard: Computer support for collaboration and problem solving in meetings. *Communications of the ACM*, *30(1)*, 32-47.

STENNING, K. & OBERLANDER, J. (1995). A cognitive theory of graphical and liguistic reasoning: logic an implementation. *Cognitive Science*, 19(1), 97-140.

STREITZ, N., GEIßLER, J.,HAAKE, J. & HOL, J. (1994). Dolphin: Integrated meeting support across Liveboards, local and remote desktop environments. In Frankhauser, P & Ockenfeld, M. (Hrsg.), *Integrated publication and information systems.* (S. 57-74). Darmstadt: GMD IPSI.

STREITZ, N., HAAKE, J., HANNEMANN, J., LEMKE, A., SCHULER, W., SCHÜTT, H. & THÜRING, M. (1992). Sepia: A cooperative hypermedia authoring environment. In *Proceedings of the 4th ACM Conference on Hypertext(ECHT 92)* (S. 11-22). Milano (Italy).

STREITZ, N., HAAKE, J., HANNEMANN, J., LEMKE, A., SCHULER, W., SCHÜTT, H. & THÜRING, M. (1998). Sepia: A cooperative hypermedia authoring environment. In Frankhauser, P & Ockenfeld, M. (Hrsg.), *Integrated publication and information systems* (S. 19-39). Darmstadt: GMD IPSI.

STREITZ, N., TANDLER, P. MÜLLER-TOMFELDE, C. & KONOMI, S. (2001). Roomware: Towards the next generation of human-computer interaction based on an integrated design of real and virtual worlds. In Carroll, J.A. (Hrsg.), *Human-computer interaction in the new millenium* (S. 551-576). Addison-Wesley.

SUTHERS, D.D. (1999A). Representational bias as guidance for learning interactions: A research agenda. In Lajoie, S.P. & Vivet, M. (Hrsg.), *Artificial intelligence in education* (S. 121-128). Amsterdam: IOS Press.

SUTHERS, D.D. (1999B). Effects of alternate representations of evidential relations on collaborative learning discourse. In J. Roschelle & Hoadley, C. (Hrsg.), *Proceedings of the Conference on Computer Supported Collaborative Learning (CSCL 99)* (S. 611-622). Palo Alto, CA.

SUTHERS, D.D. (1999C). Representational support for collaborative inquiry. In *Proceedings of the Hawai'i International Conference on the System Sciences (HICSS) (S. 1076-1077)*, January 5-8, 1999, Maui, Hawai'i.

SUTHERS, D.D. (2003). The influence of knowledge modeling on the communication process. *Künstliche Intelligenz*, *1/2003*, 31-34.

SUTHERS, D., TOTH, E.E. & WEINER, A. (1997). An integrated approach to implementing collaborative inquiry in the classroom. In *Proceedings of Computer Supported Collaborative Learning*. Toronto (Canada).

SUTHERS, D., WEINER, A., CONNELLY, J. & PAOLUCCI, M. (1995). Belvedere: Engaging students in critical discussion of science and public policy issues. In Greer, J. (Hrsg.), *Proceedings of the 7th World Conference on Artificial Intelligence in Education* (S. 266-273). Whashington DC (USA).

SUTHERS, D. & XU, J. (2002). Kukakuka: An online environment für artifact-centered discourse. *Education Track of the 11th World Wide Web Conference (WWW 2002)* (S. 472-480). Honolulu, May 7-11.

TEWISSEN, F. (1996). Begriffsnetze als Basis für ein System zur kooperativen Lösung physikalischer Aufgabenstellungen. *Master Thesis, University of Duisburg, Dept. of Mathematics / Computer Science*.

TEWISSEN, F., BALOIAN, N., HOPPE, U. H. & REIMBERG, E. (2000). "MatchMaker" Synchronising Objects in Replicated Software-Architectures. *In Proc. of 6th International Workshop on Groupware, CRIWG 2000*, Madeira (Portugal), 18 - 20 October 2000, IEEE CS Press.

TEWISSEN, F. LINGNAU, A. & HOPPE, H.U. (2000). "Today's Talking Typewriter" – Supporting early literacy in a classroom environment. In Gauthier, G., Frasson, C. & VanLehn, K. (Hrsg.), *Proceedings of 5th International Conference on Intelligent Tutoring Systems* (S. 252-261), Montréal (Canada). Berlin, Heidelberg: Springer Verlag.

TEWISSEN, F., LINGNAU, A., HOPPE, H.U., MANNHAUPT, G. & NISCHK, D. (2001). Collaborative writing in a computer-integrated classroom for early learning. In Dillenbourg, P., Eurelings, A., & Hakkarainen, K. (Hrsg.), *Proceedings of the European Conference on Computer-Supported Collaborative Learning (EuroCSCL 2001)* (S. 593-600). Maastricht.

TOULMIN, S. (1958). *The uses of argument*. Cambridge: Cambridge University Press.

VAN JOOLINGEN, W. (2000). Designing for collaborative discovery learning. In Gauthier, G., Frasson, C. & VanLehn, K. (Hrsg), *Proceedings of the 5th International Conference on Intelligent Tutoring Systems* (S. 202-211). Berlin Heidelberg: Springer.

VANLEHN, K., JONES, R.M., & CHI, M.T.H. (1992). A model of the self-explanation effect. *The Journal of the Learning Sciences*, *2*, 1-59.

VERSTEEGEN, G (2000). Was ich nicht weiß... . *iX Magazin für professionelle Informationstechnik*, *7*, 100-104.

VOGEL, M. (2001). *Medien der Vernunft*. Frankfurt: Suhrkamp.

WANG, W. & HAAKE, J.M. (2000). Tailoring groupware: The cooperative hypermedia approach. *Journal of the Cooperative Computing, special issue on tailorable systems and cooperative work in computer supported cooperative work*, *9(1)*, 123-146.

WANG, D. & LEE, J.R. (1993). Visual reasoning: its formal semantics and applications. *Journal of Visual Languages and Computing*, *4*, 327-356.

WANG, D., LEE, J. & ZEEVAT, H. (1995). Reasoning with diagrammatic representations. In Glasgow, J., Hari Narayanan, N. & Chandrasekaran, B. (Hrsg.), *Diagrammatic reasoning, cognitive and computational perspectives*. Menlo Park: AAAI Press/MIT Press.

WEBER, S. & SCHUMANN, M. (2000). Das Concept Mapping Software Tool (COMASOTO) zur Diagnose strukturellen Wissens. In Mandl, H. & Fischer, F. (Hrsg.), *Wissen sichtbar machen. Wissensmanagement mit Mappping-Techniken*. Göttingen: Hogrefe.

ZHANG, J. (1997). The nature of external presentations in problem solving. *Cognitive Science*, *21(2)*, 179-217.

ZUMBACH, J., MÜHLENBROCK, M., JANSEN, M., REIMANN, P. & HOPPE, H.U. (2002). Multi-dimensional tracking in virtual Learning teams. An exploratory study. In G. Stahl (Hrsg.), *Computer Support for Collaborative Learning: Foundations for a CSCL Community (Proceedings of CSCL 2002)* (S. 650-651). Boulder (USA), Januar 2002 (distrib. Lawrence Erlbaum).

10 LINK-VERZEICHNIS

[bscw] http://bscw.gmd.de/

[CoLab] http://www2.parc.com/istl/members/stefik/colab.htm

[Coldex] http://www.coldex.info/

[dynagloss1] http://seed.cs.colorado.edu/dynagloss.makeRedefinePage.fcgi$URLinc=0&Term=Semi-formal Representations

[dynagloss2] http://seed.cs.colorado.edu/LivingBook.DisplayNode.fcgi$URLinc=12&doc=LivingBook&node=Node23

[habanero] http://www.isrl.uiuc.edu/isaac/Habanero/

[harst] http://www.webforumdeutsch.de/webforum/argument/harst_1.pdf

[JSDT] http://java.sun.com/products/java-media/jsdt/

[JavaSpaces] http://java.sun.com/products/javaspaces/

[KM-methoden] http://www.artm-friends.at/am/ol-site/lo-lo.htm

[MindManager] http://www.mindjet.ch/guided_tour/maps.htm

[nimis] http://www.collide.info/Projects/nimis/

[ODP] http://community-ml.org/RM-ODP/rm-index.html

[OneNote] http://www.microsoft.com/office/onenote/overview.asp

[rational] http://www.rational-software.de/

[SEED] http://ilios.cti.gr/seed/

[skyrme48] http://www.skyrme.com/updates/u48_f2.htm

[skyrme49] http://www.skyrme.com/updates/u49_f1.htm

[skyrme51] http://www.skyrme.com/updates/u51_f3.htm

[verity] http://www.verity.com

[visimap] http://www.team-success.de/ord-neu/visimap/vm0.htm

[wiper] http://www.wiper.de/index.htm