

# **Interaktion und Transparenz im Kontext computergestützter kooperativer Verwaltungsumgebungen**

Vom Fachbereich Informatik  
der Technischen Universität Darmstadt  
genehmigte

Dissertation

zur Erlangung des akademischen Grades  
eines Doktor-Ingenieurs (Dr.-Ing.)

von

Dipl.-Inform. Ralph Peters

geboren in Mönchengladbach

Referent: Prof. Dr.-Ing. José L. Encarnaçao, TU Darmstadt  
Koreferent: Prof. Dr. rer. nat. Armin B. Cremers, Universität Bonn

Tag der Einreichung 15. Juni 1999  
Tag der mündlichen Prüfung 15. Oktober 1999

Darmstädter Dissertation  
D17

## Vorwort

Die Arbeit entstand im Zuge verschiedener Forschungs- und Industrieprojekte während meiner Tätigkeit am Fraunhofer-Institut für Graphische Datenverarbeitung in Darmstadt. Wesentliche Teile der konzeptionellen Arbeiten habe ich im Fraunhofer Eigenforschungsprojekt „Computerunterstützte kooperative Arbeitsprozesse“ erstellt. Ergänzende Impulse und Entwicklungen kamen aus den Projekten „CSCW Tools“ (Siemens), „Mensch-Maschine Schnittstellen im multimedialen Verwaltungsbüro“ (DeTeBerkom) und „Virtue“ (Deutsche Telekom), sowie aus Diplom- und Studienarbeiten.

Mein Dank gilt zunächst Prof. Encarnacao, der mir immer wieder den richtigen Weg zeigte. Auch möchte ich mich bei Prof Cremers herzlich für die Übernahme des Koreferenten bedanken.

In meiner Zeit am Fraunhofer-IGD habe ich zahlreiche Kollegen kennen und schätzen gelernt. Besonders bedanke ich mich bei den Kollegen der A9 für Ihre Unterstützung. Namentlich sind dies Martin Bernhard, Hartmut Chodura, Martin Einhoff, Andreas Gräff, Martin Rausch, Manfred Schendel, Norbert Schiffner, unser AL Stefan Noll und natürlich Conny Kurkowski. Eine solche Arbeit ist auch nicht ohne die zahlreichen Diplomanten, Studienarbeiter oder HiWis möglich. Erwähnt seien hier Axel, Christian, Dana, Frank, Magga, Oliver und Valentina. Als sozusagen externe Gutachterin noch einen Dank an Tatjana.

Holger Kress und Christian Neuss danke ich für die zahlreichen Diskussionen und Abende. Hier habe ich immer wieder Anregungen und Hilfen für meine Arbeit gefunden und ohne Sie hätte es in den gemeinsamen Jahren am Fraunhofer-IGD nur halb soviel Spaß gemacht.

Zuletzt möchte ich mich bei meiner inzwischen recht groß gewordenen Familie bedanken. Meiner Mutter und meinem Vater vor allen Dingen dafür, daß Sie es mir überhaupt ermöglicht haben, diesen Weg einzuschlagen. Und meiner Frau, daß Sie die Arbeit tatsächlich gelesen hat.

## Inhaltsverzeichnis

<b>1 Einleitung</b>	<b>1</b>
<b>1.1 Motivation und Problemstellung</b>	<b>1</b>
<b>1.2 Existierende Ansätze</b>	<b>3</b>
<b>1.3 Zielsetzung und Vorgehensweise</b>	<b>5</b>
<b>1.4 Wissenschaftliche Ergebnisse</b>	<b>7</b>
<b>2 Technologischer Überblick</b>	<b>9</b>
<b>2.1 CSCW und Groupware</b>	<b>9</b>
2.1.1 Einführung	9
2.1.2 Rückblick	10
2.1.3 Aware — Unaware	11
<b>2.2 Telekooperation</b>	<b>12</b>
2.2.1 Einführung	12
2.2.2 Videokonferenzsysteme	12
2.2.3 Whiteboards	13
2.2.4 Gruppendeditoren	14
2.2.5 Verteilung von Standard Applikationen	15
2.2.6 E-Mail	16
<b>2.3 Konferenzumgebungen</b>	<b>17</b>
2.3.1 Einführung	17
2.3.2 Der BERKOM Multimedia Collaboration Dienst	17
2.3.3 Der Standard T.120	19
<b>2.4 Computergestützte Vorgangsbearbeitung</b>	<b>21</b>
2.4.1 Einführung	21
2.4.2 Workflow Management	22
2.4.3 Das Referenzmodell der Workflow Management Coalition	23
2.4.4 Dokumentenmanagement und Archivierung	26

---

<b>2.5 Software Agenten</b>	<b>28</b>
2.5.1 Einführung	28
2.5.2 Architekturen von Agentensystemen	28
2.5.3 Knowledge Query and Manipulation Language	31
2.5.4 Die Agentenplattform ASAP	32
<b>2.6 Transparenz in Groupware</b>	<b>34</b>
2.6.1 Einführung	34
2.6.2 Mikro-Level-Transparenz	35
2.6.3 Makro-Level-Transparenz	36
<b>2.7 Zusammenfassung</b>	<b>38</b>
<b>3 Transparenz in kooperativen Verwaltungsumgebungen</b>	<b>41</b>
<b>3.1 Motivation</b>	<b>41</b>
3.1.1 Software-Ergonomie als treibende Kraft	41
3.1.2 Einsatzgebiete	42
3.1.3 Transparenzbezug	45
<b>3.2 Analyse der Benutzeranforderungen</b>	<b>46</b>
3.2.1 Einstieg	46
3.2.2 Szenario aus dem Büroalltag	47
3.2.3 Analyse	48
3.2.4 Schlußfolgerungen	51
<b>3.3 Technisches Transparenzmodell</b>	<b>52</b>
3.3.1 Modell der Transparenzdatensätze	52
3.3.2 Geographische Verteilung von Transparenzdatensätzen	54
3.3.3 Quellen der Transparenzdatensätze	54
<b>3.4 Zusammenfassung</b>	<b>55</b>
<b>4 Konzeption des Referenzmodells</b>	<b>57</b>
<b>4.1 Einleitung</b>	<b>57</b>
<b>4.2 Anforderungsprofil</b>	<b>58</b>
4.2.1 Integration unterschiedlicher Groupware-Komponenten	58
4.2.2 Software-ergonomische Integration	58
4.2.3 Transparenz innerhalb des Gruppenraums	59

---

<b>4.3 Spezifikation des Referenzmodells</b>	<b>59</b>
4.3.1 Grobkonzeption	59
4.3.2 Transparenzdatensätze im Referenzmodell	61
4.3.3 Basisarchitektur des Agentenverbundes	64
4.3.4 Workflow-Agent	64
4.3.5 Telekooperations-Agent	67
4.3.6 Ressourcen- und Terminkalender-Agent	70
4.3.7 Administrator-Agent	73
4.3.8 Weitere Groupware Agenten	73
<b>4.4 Entscheidungsfindung – Autonomie für Agenten</b>	<b>74</b>
4.4.1 Einführung	74
4.4.2 Exkurs Fuzzy-Logik	75
4.4.3 Fuzzy-Logik-basierte Entscheidungsfindung	80
<b>4.5 Zusammenfassung</b>	<b>92</b>
<b>5 Mikro-Level-Transparenz oder Der Fokus des Interesses</b>	<b>95</b>
<b>5.1 Motivation</b>	<b>95</b>
<b>5.2 Definitionen</b>	<b>97</b>
5.2.1 Aktive und passive Interaktion	97
5.2.2 Granularität	97
<b>5.3 Konzeption zur Interaktion</b>	<b>99</b>
5.3.1 Transparente Annotationen	99
5.3.2 Semantische Verknüpfung	100
5.3.3 Dynamische Annotationen	105
<b>5.4 Integration in das Referenzmodell</b>	<b>106</b>
5.4.1 Anforderungsprofil	106
5.4.2 Systembezogene Referenzarchitektur im Referenzmodell	108
5.4.3 Anbindung der Annotation-Engine	109
5.4.4 Compound Entities	110
5.4.5 Arbeitsablauf innerhalb der Referenzarchitektur	111
<b>5.5 Zusammenfassung</b>	<b>114</b>

---

<b>6 Realisierungen</b>	<b>117</b>
<b>6.1 Einleitung</b>	<b>117</b>
<b>6.2 Mensch-Maschine-Interaktion im multimedialen Verwaltungsbüro</b>	<b>117</b>
<b>6.3 CrystalPad</b>	<b>119</b>
6.3.1 Übersicht	119
6.3.2 Agieren und Annotieren	122
6.3.3 Semantische Verknüpfung	123
<b>6.4 Virtue</b>	<b>124</b>
6.4.1 Übersicht	124
6.4.2 Basis-Agent	124
6.4.3 Telekooperations-Agent	125
6.4.4 Workflow-Agent	126
6.4.5 CTI-Agent	126
6.4.6 Termin-Agent	127
<b>7 Systemvergleich</b>	<b>129</b>
<b>7.1 Einleitung</b>	<b>129</b>
<b>7.2 Analyse und Vergleich</b>	<b>129</b>
7.2.1 POLITEAM	129
7.2.2 WoTel	131
7.2.3 WooRKS-UTUCS	133
7.2.4 POLIWORK	134
<b>7.3 Zusammenfassung und Beurteilung</b>	<b>136</b>
<b>8 Zusammenfassung und Ausblick</b>	<b>139</b>
<b>8.1 Zusammenfassung der Ergebnisse</b>	<b>139</b>
<b>8.2 Stand der Entwicklungen</b>	<b>141</b>
<b>8.3 Ausblick auf weitere Entwicklungen</b>	<b>142</b>
<b>Abbildungsverzeichnis</b>	<b>145</b>
<b>Literaturverzeichnis</b>	<b>149</b>

# Kapitel 1

## Einleitung

### 1.1 Motivation und Problemstellung

Mit der Rezession Anfang der 90er Jahre kam die Erkenntnis, daß althergebrachte und anscheinend bis dahin bewährte Organisationsstrukturen und -modelle nicht mehr den neuen Anforderungen genügen [80]. Diese Sichtweise hat sich inzwischen auf die öffentliche Verwaltung ausgedehnt. Bürgernähe, Leistungsorientierung und schlanke Verwaltung sind nun Schlagwörter, die auch hier Einzug halten. “Der schlanke Staat, das heißt höhere Effizienz, Flexibilität und Qualität bei gleichzeitiger Kostensenkung im gesamten öffentlichen Sektor, ist aufgrund der internationalen Wettbewerbssituation zwingend notwendig und ohne Informatik und ohne moderne Informations- und Kommunikationstechnik nicht realisierbar” [24] (S.27).

IN einer Studie zum Informationsverbund Berlin-Bonn (IVBB) [28] vom Bundesministerium des Innern werden kooperationsrelevante Aktivitäten der Verwaltung charakterisiert und Anforderungen an die Realisierung spezifiziert. Hintergrund ist der Beschluß des Deutschen Bundestages zur Vollendung der Deutschen Einheit, der zu einer Verteilung der Verfassungsorgane zwischen Berlin und Bonn führt. Die damit verbundene zunehmende Dezentralisierung der Ämter bewirkt, daß Kommunikation und Kooperation, zum Erreichen eines gemeinsamen Aufgabenziels auch über große geographische Entfernungen hinweg, ein zentraler Aspekt wird. Im Zuge der Europäischen Union wird die Kooperation der Verwaltungsämter auch international erforderlich sein.

Die Vielzahl an möglichen Szenarien bedarf einer intensiven computerbasierten Unterstützung auf breiter Ebene. Nach den Studien [87] und [93] zeichnen sich diesbezüglich klare Trends in Richtung netzbasierter und integrierter Gesamtsysteme ab. Ziel muß es sein, den Anwender in seiner Arbeit, allein oder im Team, optimal zu unterstützen. Der Nutzen einer integrierten, computergestützten Verwaltungsumgebung ergibt sich durch deren Vollständigkeit. Medienbrüche werden vermieden, der Anwender kann in seinem Arbeitsumfeld durchgängig agieren. Eine Steigerung der Produktivität und die Verbesserung der Prozeßqualität werden durch Reduzierung von Durchlaufzeiten und Fehlerquellen erreicht.

Neben Standard-Applikationen aus dem Büro, wie Textverarbeitungs- und Tabellenkalkulations-Systemen, haben als Basiswerkzeuge für die verteilte Verwaltung vor allen Dingen Anwendungen aus dem Umfeld der computerunterstützten Gruppenarbeit (*computer supported cooperative work* – CSCW) auf sich aufmerksam gemacht. Hierunter fallen unter anderem Workflow-Management- und Telekooperations-Systeme.

Workflow-Management-Systeme (WFMS) stellen Werkzeuge zur Verfügung, um Büroprozesse zu modellieren und computergestützte Vorgangsbearbeitung zu realisieren [86]. Teilaktivitäten eines festgelegten Prozesses werden vom System überwacht, gesteuert und mit den notwendigen Dokumenten versorgt. Die Workflow Management Coalition (WfMC) [161] legt derzeit als wesentlichen Meilenstein auf dem Weg zu einem Standard in diesem Bereich ein Workflow-Referenzmodell fest. Das Referenzmodell definiert ein generelles Architekturrahmenwerk für weitere Spezifikationen der WfMC. Als Kern fungiert in dem Modell die Workflow-Engine, die das ausführende Organ des Systems ist. Es werden Schnittstellen für den In- und Export von Prozeßdefinitionen, für die Interaktion mit Client-Applikationen und Aktivitätslistenhändlern, für die Inbezugnahme externer Software und Applikationen, für die Interoperabilität zwischen unterschiedlichen Workflow-Systemen und für die Administration und das Monitoren der Workflow-Engine festgelegt. Abschnitt 2.4 ab Seite 21 wird die computergestützte Vorgangsbearbeitung und das Referenzmodell der WfMC genauer beleuchtet. An dieser Stelle wird auch auf die mit WfMC eng verknüpfte Thematik des Dokumentenmanagements eingegangen.

Telekooperations-Systeme fördern durch den Einsatz unterschiedlicher Kommunikationstechniken Kooperation, bei der die Kooperationspartner zeitlich oder räumlich getrennt sind. Etablierte Techniken hierzu sind Audio/Videokonferenz, *whiteboard*, Gruppeneditor, *shared application* oder E-Mail. Interessant für die Verwaltung sind Konferenzumgebungen, die unterschiedliche Aspekte der Telekooperation integrieren. Der Standard T.120 definiert dazu Kommunikation mittels unterschiedlichster Formen von Daten und telematischen Medien zwischen zwei oder mehr Teilnehmern. Das T.120 Protokoll [68] umfaßt den Aufbau von Konferenzen oder Konferenzgruppen, Mechanismen zum Austausch von Identifikation und Informationen zu den teilnehmenden Elementen und flexible Möglichkeiten zur Erstellung von Kommunikationswegen. Auf die Einzelheiten des Standards geht Abschnitt 2.3.3 ab Seite 19 ein. Weitere Bemerkungen zu den Themen Telekooperation und Konferenzumgebungen finden sich in den Abschnitten 2.2 und 2.3.

Problematisch bei der Einführung von CSCW-Systemen ist die zunehmende Komplexität der Systeme gepaart mit einem potentiell als Computer- und Telekommunikations-Laien einzustufenden Anwender. Das tatsächliche Anwendungspotential einer integrierten Umgebung wird im Endeffekt durch den Nutzer bestimmt. Die eigentlich zu erledigende Arbeit wird mit Standard-Anwendungen wie Textverarbeitungs- oder Tabellenkalkulations-Systemen durchgeführt. Daraus ergibt sich zwangsweise, daß andere Systemteile, die Vorgangsautomatisierung oder Telekooperation ermöglichen, aus Sicht des Anwenders zunächst nebensächlich sind. In CSCW-Umgebungen muß insbesondere eine software-ergonomische Gestaltung der Benutzungsschnittstelle eine einfache Bedienung der komplexen Strukturen erlauben [10]. Transparenz macht dabei den Nutzen einer Funktion nachvollziehbar und verdeutlicht dem Anwender die Vorgänge. Nutzungsbezogene oder *Makro-Level* Transparenz kann den Mangel an Erwartungskonformität, der durch die gegenseitige Beeinflussung der verschiedenen Nutzer bedingt ist, kompensieren [63]. Die Wichtigkeit der Schaffung von Transparenz wird deutlich, wenn man bedenkt, daß Transparenz bei über der Hälfte aller kooperativen Aktivitäten im Büroalltag relevant ist (siehe Abschnitt 3.2 „Analyse der Benutzeranforderungen“). Da es sich bei den transparent gemachten Sachverhalten unter Umständen um persönliche Daten handelt, wirft dies natürlich sofort die Frage nach Daten-



schutz und dem Recht auf informationelle und kommunikative Selbstbestimmung auf (vgl. [130]). Transparenz ohne Überwachung ist hier der entscheidende Unterschied.

Bei der Einführung von CSCW-Techniken muß weiterhin besonderer Wert auf die Unterstützung der synchronen Kooperation gelegt werden. Dem Benutzer sind Audio/Video-Konferenzsituationen aus dem täglichen Leben in ähnlicher Form (Telefon, informelle Diskussion) bekannt, und es ist anzunehmen, daß diese schnell adaptiert werden können. Die Arbeit mit synchronen CSCW-Werkzeugen wie etwa *Whiteboards* oder *Sharing* Komponenten dagegen ist neu. Innerhalb einer Konferenzsituation muß sich die Unterstützung auf den Fokus des Interesses der Anwendergruppe konzentrieren. Unter synchroner kooperativer Interaktion mit dem Fokus des Interesses (FDI) wird der aktive (Manipulation der den Fokus repräsentierenden Entität) und passive (Annotation der den Fokus repräsentierenden Entität) Umgang mit dem aktuellen Diskussionspunkt verstanden (siehe Abschnitt 5.2). Synchroner kooperativer Interaktion mit dem FDI kann dabei als wesentliches Teilgebiet der Mikro-Level-Transparenz (siehe Abschnitt 2.6.2) angesehen werden.

Mitglieder eines Team müssen gezielt bei der Lösung von Problemen unterstützt werden. Gemeinsam kann in einem Arbeitsbereich diskutiert und aktiv am aktuell zu bearbeitenden Objekt manipuliert werden. Gruppenmitgliedern muß dazu die Möglichkeit gegeben werden, Dokumente gleichzeitig zu bearbeiten und zu annotieren. Dabei darf eine räumliche Distanz zwischen den einzelnen Mitgliedern kein Hindernis sein. Für die Bearbeitung der Dokumente sollte dem Anwender die Wahl der Applikation überlassen werden. Dies bedeutet, daß synchrone kooperative Interaktion mit dem FDI mit unterschiedlichsten Anwendungen ermöglicht werden muß.

Es ergibt sich somit die Nachfrage nach einer software-ergonomischen, den Anwender in seiner Arbeit unterstützenden, Gesamtumgebung. Die wesentlichen Anforderungen und Problemstellungen aus nutzerorientierter und technischer Sicht sind dabei:

- Software-ergonomische Integration der heterogener CSCW-Werkzeuge auf der Bedienungsebene
- Transparenz im Makro- und Mikro-Level-Bereich innerhalb des Gruppenraums zum Nutzen jedes Einzelnen der Gruppe (und damit nicht zur Überwachung oder zum Nachteil einzelner)
- Passive und aktive Interaktion mit dem Fokus des Interesses innerhalb einer synchronen kooperativen Sitzung

## 1.2 Existierende Ansätze

Aus Motivation und Problemstellung wird deutlich, daß das Umfeld dieser Arbeit durch die Anforderungen, die sich aus den neuen Formen der Arbeitsorganisation innerhalb der Verwaltungen ergeben, und den innovativen Möglichkeiten der modernen Computertechnik charakterisiert ist. Dieser Problematik haben sich auch andere Projekte und Arbeiten angenommen, von denen eine Auswahl im folgenden beschrieben wird:

- Ausgehend von der aus dem Beschluß des Deutschen Bundestages zur Vollendung der Deutschen Einheit entstehenden Problematik der Verteilung der Verfassungsorgane zwischen Berlin und Bonn unterstützt die Bundesregierung das Verbundprojekt POLIKOM. Die standortverteilte Kooperation von Verwaltungen soll in den vier Teilprojekten POLITeam, POLIFlow, POLIWork und POLIVest anhand praxisnaher Experimente untersucht werden. Die Zusammensetzung der Verbundprojekte spiegelt dies wider: Jeweils Industrievertreter als Konsortialführer bestreiten zusammen mit wissenschaftlichen Einrichtungen und einem oder mehreren öffentlichen Ämtern als Anwender ein Projekt. Die Projekte sind für diese Arbeit wichtig, da hier zum ersten Mal in großem Stil die Einführung von Workflow- und Telekooperationstechniken in der Verwaltung erprobt wird.

Ziel ist die praxisgerechte Umsetzung des Beschlusses des Deutschen Bundestages mit heute verfügbaren Techniken. Die Projekte konnten ihre erste Phase allesamt erfolgreich abschließen, was zeigt, daß trotz der meist eingefahrenen Strukturen in der Verwaltung der Einsatz moderner Technologien möglich ist. Eingesetzt wurden kommerzielle Produkte, die speziell an die Bedürfnisse des jeweiligen Anwenders angepaßt wurden. Im Projekt POLITeam ist eine elektronische Laufmappe, ein gemeinsamer Arbeitsbereich und ein Videokonferenz-System eingeführt worden. Neben dem Workflow System LinkWorks® der Firma Digital wird hier das Telekooperations-System ProShare® von Intel eingesetzt. Herauszuheben ist, daß ein eigens hierfür entwickelter Mechanismus mit Hilfe von Strichcodes die Zusammenführung von Papierdokumenten mit elektronischen Gegenständen ermöglicht. Ein weiterer Aspekt von POLITeam ist die Entwicklung eines Ereignisdienstes.

Die Bearbeitung von Gebäude- und Einrichtungsplänen ist eines der Hauptziele des Projektes POLIFlow. Herausragend ist hier der Einsatz von *application sharing*, um Diskussionen an CAD-Plänen von Gebäuden vornehmen zu können. Das Projekt POLIWork hat in der ersten Phase eine Evaluierung der planenden Ministerialverwaltung vorgenommen, deren Ergebnisse der zweiten Phase als Grundlage dient. Im Projekt POLIVest werden innovative Telekooperationstechniken bei Beratungsverfahren im Bundesrat und Baugenehmigungsverfahren im Rhein-Sieg-Kreis realisiert. Zum Einsatz kommt hier die breite Produktpalette der Firmen SNI, SIETEC und SIEMENS. Ein besonderer Augenmerk wird hier auf die Sicherheitsaspekte innerhalb der Verwaltung geworfen.

Im Kapitel 7 "Systemvergleich" wird nochmals auf die POLIKOM Projekte eingegangen.

- Im Bereich des World Wide Web (WWW) [21] haben sich einige Projekte plaziert, um vor allen Dingen die immer größer werdende Dokumentenflut zu verwalten. Die Integration des WWW in diese Arbeit wird im Ausblick auf weitere Entwicklungen (siehe Abschnitt 8.3) bedacht. An dieser Stelle sollen Entwicklungen betrachtet werden, die sich speziell mit dem Bereich "Annotationen" beschäftigen, da sie eine besondere Rolle bei der passiven Interaktion mit dem Fokus des Interesses spielen (siehe Abschnitt 5.3).

Alle Ansätze haben gemeinsam, daß grundsätzlich Annotationen als *Link* (Anker) in ein Web-Dokument eingebracht werden (siehe z.B. [19] oder [129]). Ein solcher Anker wird durch eine optische Hervorhebung (z. B. Unterstreichung oder Farbwechsel) oder ein Symbol im Dokument kenntlich gemacht [125]. Annotationen und Dokument bilden zusammen ein (neues) Hypermedia-Dokument. Die Verwaltung der Annotationen steht im Vordergrund der wissenschaftlichen Untersuchungen und Experimente auf diesem Gebiet.

Das am Computer Science Department der Stanford University entwickelte *ComMentor* [129] bietet eine Plattform, um Anmerkungen zu WWW-Dokumenten zu verwalten. Wesentlicher Wert wurde auf die Strukturierung der Anmerkungen gelegt. Interessant ist dabei die strikte Trennung von Dokument und Anmerkungen auf unterschiedliche Web-Server. Durch Verwendung von Meta-Informationen kann die Zuordnung rekonstruiert werden. Nach dem Laden eines Dokumentes (von einem beliebigen Server) erfragt der Browser bei einem speziellen Annotations-Server die dazugehörigen Anmerkungen. Anhand der eindeutigen Identifikation des Dokumentes (URL – *Unified Resource Locator*) werden mit Hilfe der dazugehörigen Meta-Information die entsprechenden Anmerkungen zusammengestellt und an den Browser zurückgeliefert. Das System verlangt dann spezielle Eigenschaften des Browsers zum Darstellen von Dokument und Annotationen. Mit speziellen Mechanismen müssen diese zusammengeführt werden.

WWW-basierte Systeme können insbesondere nicht dem Anspruch der synchronen Interaktion mit dem Fokus des Interesses gerecht werden. Auch liegen ihnen nicht Standard Büroapplikationen wie Textverarbeitungs- oder Tabellenkalkulations-Systeme zu Grunde, sondern Dokumentendarstellungskomponenten (Browser), was einen Einsatz in einer kooperativen Verwaltungsumgebung nur bedingt ermöglicht. Die Verwendung von Meta-Information zur Zuordnung von Annotation und Dokument wird aber im hier vorgestellten Lösungsansatz aufgegriffen und zur Erstellung von *compound entities* verwendet (siehe Abschnitt 5.4.4).

- Für den Bereich Makro-Level- oder auch informelle Transparenz sind für den personenbezogenen Bereich fast ausschließlich Video-basierte Systeme entwickelt worden (siehe etwa [36], [71] oder [136]). Diese Systeme verwenden Videokameras, um die beteiligten Personen einer Gruppe oder einer Institution an ihrem Arbeitsplatz aufzunehmen und diese Bilder über Netzwerke den anderen zugänglich zu machen. Untersucht werden soll hier vor allen Dingen, inwieweit der Einsatz solcher Systeme Einfluß auf zwischenmenschliche Kommunikations- und Interaktionsverhalten hat und ob eine Verbesserung der Kollaboration zu erreichen ist. Außerordentlich fraglich ist allerdings die Verwendung dieser Systeme außerhalb von Forschungseinrichtungen. Der Aspekt der Überwachung ist hier sehr stark gegeben. Andere Systeme gehen sogar einen Schritt weiter und überwachen die Tastatureingabe am Rechner. Ausgewertet wird hier zwar meist nur die Tatsache, daß der Rechner bedient wird, aber auch hier ist ein reeller Einsatz kaum denkbar.

Im Abschnitt 2.6.3 wird der Begriff Makro-Level-Transparenz weiter ausgeführt. Um eine Möglichkeit zur Umsetzung der theoretischen Überlegungen zur Transparenz in ein Realisierungskonzept zu erhalten, wird in Abschnitt 3.3 ein technisches Modell für Transparenz entwickelt, welches in Abschnitt 4.3.2 verfeinert wird.

### 1.3 Zielsetzung und Vorgehensweise

Das Ziel dieser Arbeit ist die Definition eines Referenzmodells für eine integrierte Umgebung zur Unterstützung der Belange einer Verwaltung, die durch nutzungsbezogene Transparenz eine software-ergonomische Bedienung und Nutzung erlaubt. Die Umgebung soll durch den Einsatz von CSCW-Techniken sowohl gruppenweite Aktionen als auch die ein-

zelle Person in ihrer Arbeit unterstützen. Die Einbeziehung der Anforderungen aus Anwendersicht ist ein zentraler Aspekt. Gruppenweite Transparenz ohne Überwachung und die Vereinfachung der Bedienung der komplexen Software-Bausteine sollen eine ergonomische Benutzerführung sicherstellen. Die Unterstützung von asynchronen Tätigkeiten und synchronen Interaktionen muß im Hinblick auf die potentielle Nutzergruppe gewährleistet sein. Vorhandene standardisierte und damit allgemein anerkannte Architekturen sollen im Einklang mit dem Referenzmodell stehen. Das Referenzmodell soll weiterhin als Basis für die Entwicklung einer zur Realisierung geeigneten Referenzarchitektur dienen.

Um diese Zielsetzung umzusetzen, wird zunächst ein Blick auf bereits existierende Techniken (Kapitel 2) geworfen und untersucht, inwieweit sie für das hier zu entwerfende Referenzmodell verwendbar sind. Eine Analyse der Benutzeranforderungen wird in Kapitel 3 anhand von Szenarien aus dem Büroalltag Aufschluß über die Relevanz der hier vorgestellten Problemstellungen bezüglich Transparenz im Gruppenraum geben. Aus den gemachten theoretischen Überlegungen wird ein technisches Modell zu Transparenz entwickelt, welches für eine spätere Umsetzung im Referenzmodell verwendet wird.

Grundlage für das Referenzmodell, welches in Kapitel 4 spezifiziert wird, ist der Einsatz von Software Agenten als integrativer Bestandteil. Die Heterogenität der für eine kooperative Verwaltungsumgebung zu verwendenden asynchronen und synchronen Basistechniken (Workflow, Telekooperation) zwingt zu einem Umdenken von einer statischen Lösung zu dynamisch flexiblen Ansätzen. Die Definition von direkten Schnittstellen oder Konvertern zwischen diesen Basistechniken ist nicht sinnvoll, da deren Aufgabengebiete und Informationseinheiten offensichtlich zu konträr sind. Autonome und anpassungsfähige Einheiten, die in der Lage sind, selbständig entsprechend der gerade vorherrschenden Situation Entscheidungen treffen zu können, sind hier von Vorteil. Software Agenten erfüllen diese Voraussetzungen.

Dieser Technologiebereich nimmt zunehmend mehr Einfluß auf die Informationsgesellschaft. Software Agenten werden derzeit in hunderten von Applikationen eingesetzt, um Probleme aus unterschiedlichsten Bereichen wie Büromanagement oder elektronischer Handel zu lösen. Zwei Marktstudien sagen aus, daß Software Agenten das Computer Paradigma der nächsten 10 Jahre sein werden. Janca [73] behauptet, daß bis zum Jahr 2000 jede wichtige Applikation eine Art Agentenfunktionalität besitzen wird. Guilfoyle und Warner [60] schätzen ab dann den Marktanteil von Agentenbasierter Software auf über 1.2 Milliarden Pfund pro Jahr. Wooldridge und Jennings [158] geben einen Überblick aktueller Entwicklungen auf diesem Gebiet. Software Agenten zeichnen sich durch Autonomie, Reaktions- und Entscheidungsfähigkeit, Glaubwürdigkeit und Sozialfähigkeit aus (siehe Abschnitt 2.5). Mit KQML [46] hat sich eine Agentensprache etabliert, die die notwendige Kommunikation zwischen Agenten standardisiert.

Software Agenten eignen sich weiterhin für die Erfüllung der ergonomischen Anforderungen. Agenten nehmen ihre Umwelt wahr und reagieren auf Veränderungen in ihr. Sie sind damit in der Lage, sich an den Nutzer anzupassen und ihn individuell zu unterstützen. Durch Interaktion mit anderen Agenten und Menschen können auf Makro-Level-Ebene gruppentransparente Aktionen realisiert werden. Die Autonomie und Glaubwürdigkeit der

Agenten stellt dabei sicher, daß Überwachung nicht stattfindet. Eine automatisierte Kontrolle und Koordination wird dadurch möglich, daß Agenten auch ohne die direkte Intervention von Menschen in einer zielgerichteten Art die Initiative ergreifen können. Eine vereinfachte Benutzerführung kann somit erreicht werden. Diese wird durch den Einsatz eines auf Fuzzy-Logic basierenden Moduls weiter erhöht. Die Verwendung von linguistischen Variablen erlaubt hier die Umsetzung natürlichsprachlicher Benutzeranforderungen in exakte Anweisungen an die Software Agenten.

Die Unterstützung von Mikro-Level-Transparenz ist Gegenstand des Kapitels 5. Durch den Einsatz von Techniken zur synchronen kooperativen Interaktion mit dem Fokus des Interesses werden die Mitglieder eines Team gezielt bei der Verwendung von CSCW-Systemen unterstützt. Aktive und passive Interaktion wird definiert und ein Lösungsschema zur Integration vorgestellt. Resultierend wird daraus eine systembezogene Referenzarchitektur für eine kooperative Verwaltungsumgebung innerhalb des Referenzmodells erstellt. Die Umsetzung der Architektur in reale Prototypen wird in Kapitel 6 beschrieben. Es werden zwei Entwicklungen vorgestellt, die als Nachweis der Realisierbarkeit dienen.

In Kapitel 7 wird ein Vergleich der in dieser Arbeit vorgestellten Lösungen mit anderen bekannten Ansätzen durchgeführt. Eine Zusammenfassung der Arbeit und einem Ausblick auf zukünftige Entwicklungsmöglichkeiten gibt abschließend Kapitel 8.

## 1.4 Wissenschaftliche Ergebnisse

Wesentlicher Aspekt der Arbeit ist die Schaffung eines vielseitigen Modells, um die unterschiedlichen Belange einer Verwaltung befriedigen zu können. Das Spektrum des entwickelten Referenzmodells löst daher einen Großteil der in der Motivation angegebenen Problematiken. Es werden dabei neue Erkenntnisse zur Spezifikation agentenbasierter Systeme, dem Design software-ergonomischer Benutzungsoberflächen und innovativer Integrationsmöglichkeiten komplexer Systeme gewonnen.

Die wissenschaftlichen Ergebnisse dieser Arbeit können zusammengefaßt werden durch:

- Definition und Umsetzung eines Referenzmodells einer kooperativen, verteilten Verwaltungsumgebung zur Integration heterogener CSCW-Techniken
- Definition eines Transparenz-Modells zur technischen Umsetzung von Transparenz
- Umsetzung von Transparenz ohne Überwachung in einer CSCW-Umgebung durch den Einsatz von Software Agenten
- Definition von linguistischen Variablen zur Umsetzung natürlichsprachlicher Benutzeranforderungen
- Spezifikation und Implementierung von Fuzzy-Technologie zur Erlangung selbständiger Entscheidungen bei Software Agenten
- Spezifikation und Implementierung von Anreicherung von Standard-Applikationen mit CSCW-Funktionalität

- Definition und Integration von passiver und aktiver Interaktion mit dem Fokus des Interesses innerhalb einer synchronen kooperativen Sitzung
- Entwicklung von graphischen Annotationen in einer applikationsunabhängigen Form in einer CSCW-Umgebung
- Definition und Implementierung von semantischer Verknüpfung zwischen Annotation und annotierter Entität

## Kapitel 2

# Technologischer Überblick

## 2.1 CSCW und Groupware

### 2.1.1 Einführung

Mit *Computer Supported Cooperative Work* (CSCW) wird ein interdisziplinärer Forschungszweig bezeichnet, der sich sowohl mit theoretischen und psychologischen Aspekten von Gruppenarbeit und Gruppendynamik als auch mit Möglichkeiten zur technologischen Umsetzung der Theorien in computerbasierte Systeme beschäftigt.

Das Ziel des Forschungsbereiches CSCW liegt darin, „technologische Möglichkeiten so einzusetzen, daß sie kooperatives Arbeiten verschiedener Individuen und Teams in Formen und Kontexten ermöglichen, die traditionell nicht oder [nur] unter erheblich schwereren Bedingungen möglich waren“ [42] (S.96). Dabei ist unter CSCW keine eigenständige Technologie zu verstehen, sondern vielmehr die Integration unterschiedlicher Verfahren, die im Zusammenhang das kooperative Arbeiten von Individuen unterstützen [41]. Weder Netzwerk- noch digitale Audio-/Videotechnologie kann als eigenständige Technologie Kooperation zwischen unterschiedlichen Benutzern ermöglichen, auch wenn sie als Basistechnologien anzusehen sind.

Die Entwicklung von CSCW-Systemen erfordert konsequenterweise genauso Wissen aus den Technologiebereichen Netzwerke, verteiltes Datenbankmanagement, digitale Audio- und Video-Komponenten, Design von Benutzungsschnittstellen oder verteilte Systemarchitekturen wie auch ein Verständnis für die Anforderungen aus den Anwendungsbereichen wie etwa CAD/CAM, Büro und Verwaltung oder Medizin. Die Entwicklung erfordert damit interdisziplinäre Annäherungen bei der Definition der Systeme und integrierte Lösungen, die die unterschiedlichen Ansätze und Komponenten einschließen [59].

Die so entstandenen Software Lösungen werden im allgemeinen mit Groupware bezeichnet. Ellis definiert in [38] Groupware als computerbasierte Systeme, die Gruppen von Personen, die einer gemeinsamen Aufgabe oder einem gemeinsamen Ziel verpflichtet sind, unterstützen und eine Schnittstelle zu einer gemeinsamen Umgebung bieten.

Groupware beschreibt daher mehr eine technologieorientierte Sicht mit der Diskussion von Softwareprodukten, während CSCW ein Forschungsgebiet beschreibt, das neben der techno-

logischen Sicht einen erheblichen Schwerpunkt auf das Wesen der Gruppenarbeit an sich legt. Die folgende kurze Beschreibung der Historie ist daher dem Begriff CSCW gewidmet.

### 2.1.2 Rückblick

Auch wenn der Begriff CSCW im Jahre 1984 durch einen vom Massachusetts Institute of Technology (MIT) und Digital Equipment Cooperation (DEC) abgehaltenen Workshop mit dem Titel „Computer-Supported Cooperative Work“ geprägt wurde, gehen die Ursprünge von CSCW in die 60iger Jahre zurück.

In [44] beschreibt Douglas Engelbart seine Pioniertätigkeit am Augment Research Centre. Sein NLS System (später in AUGMENT umbenannt) beinhaltet nicht nur Werkzeuge für asynchrone und verteilte Kommunikation, sondern auch erste Ansätze für synchrone Telekonferenzen. Andere frühe Umgebungen, wie CoLab von Xerox Parc [139], ein Meeting-Raum System, The Coordinator [157], ein elektronisches Mail System für organisierten Informationsaustausch, oder NoteCards [146], ein kooperatives, hypertext-basiertes System zur Strukturierung von Ideen, sollten an dieser Stelle erwähnt werden. Wichtige Voraussetzung für CSCW waren Netzwerke und Mehrbenutzer-Betriebssysteme. Eines der ersten Netzwerke, das ARPANET [44], wurde im Augment Research Centre für das NLS System verwendet. Mit der Integration der damaligen Großrechner in ein Netzwerk wurde die Basis für computerunterstützte Gruppenarbeit – das Austauschen von Nachrichten zwischen Computern – geschaffen.

Johansen [75] beschrieb 1988 in seiner zwei-dimensionalen *time-place* Matrix die heute allgemein anerkannte Kategorisierung von CSCW-Systemen in vier Gruppen (Abbildung 1).

	gleiche Zeit	untersch. Zeit
gleicher Ort	Interaktion Angesicht zu Angesicht	Asynchrone Interaktion
unterschiedlicher Ort	Synchrone verteilte Interaktion	Asynchrone verteilte Interaktion

Abbildung 1: Die Johansen time-place Matrix (nach [75])

Die erste Gruppe umfaßt Systeme, die Teamarbeit unterstützen, die am selben Ort zur gleichen Zeit stattfindet. Darunter fallen Group Decision Support Systems (GDSS), die die Entscheidungsfindung in einer Sitzung erleichtern sollen sowie Präsentationswerkzeuge eines Electronic Meeting Room. Ein solcher Raum wird speziell mit einer elektronischen Leinwand (z. B. dem LiveBoard von PARC [39]) und vernetzten Arbeitsplatzrechnern ausgestattet. Diese sind in die Sitzungstische integriert und der Inhalt der Bildschirme kann auf die Leinwand projiziert werden. Ein elektronischer “Zeigestock” für alle Teilnehmer unterstützt die Diskussion zusätzlich.



Die zweite und dritte Gruppe besteht aus Systemen für kooperative Arbeit zu unterschiedlichen Zeiten. Dies ist die Kategorie der asynchronen CSCW-Systeme, wobei die Unterscheidung „gleicher Ort / verschiedener Ort“ kaum noch relevant ist. Fast alle Systeme bieten heutzutage die Möglichkeit des entfernten Zugriffs. Dieser Bereich beinhaltet Groupware wie Elektronische Mail Systeme, Projekt- und Kalender-Management Systeme, Hypertext-Systeme, Blackboards, kooperative Editoren (soweit sie keine synchrone Bearbeitung zulassen), und allgemein Systeme, die den Zugriff auf gemeinsame Informationen zulassen. Die Organisation und die Kontrolle des Zugriffs auf die gemeinsam geteilten Informationen ist eine der Hauptaufgaben dieser Systeme. Insbesondere fallen auch Workflow-Systeme in diesen Bereich. Aufgrund ihrer Mächtigkeit werden sie in der Literatur allerdings oft eigenständig genannt und nicht als Groupware bezeichnet.

Der letzte Quadrant umfaßt die große Gruppe der synchronen, verteilten CSCW-Systeme. Hierunter fallen Konferenz-Management Systeme, Audio-/Video-Systeme, synchrone kooperative Editoren, *Shared Viewing* und *Shared Pointer* Systeme und *Whiteboards*. Letztere ermöglichen allgemein das synchrone interaktive kooperative Arbeiten an vernetzten Rechnern, unabhängig davon, wo sich die Teilnehmer einer so arbeitenden Gruppe befinden.

In den folgenden Abschnitten dieses Kapitels werden die Technologien und Funktionen dieser Systeme detailliert beschrieben. Ein Überblick über die Historie sowie interessante Blickwinkel zum Thema CSCW finden sich in [15], [38], [59], [128] und [142]. Zuvor soll allerdings auf die Begriffe *aware* und *unaware* eingegangen werden, da sie eine weitere wichtige Unterscheidungsmöglichkeit von CSCW-Systemen bieten.

### 2.1.3 Aware — Unaware

Neben der *time-place* Matrix von Johansen unterscheidet die CSCW-Forschung zwischen *collaboration unaware* und *collaboration aware* Software. Programme, die konzipiert sind, um zu einem Zeitpunkt nur von einer Person benutzt zu werden, werden als *collaboration unaware* bezeichnet. Diese Gruppe umfaßt den Großteil der heute zur Verfügung stehenden Software. Beispiele hierfür sind etwa Textverarbeitungs-, Tabellenkalkulations- oder CAD-Programme.

*Collaboration aware* Software ist hingegen speziell mit CSCW-Methoden, die in besonderer Weise Gruppenarbeit unterstützen, ausgestattet. Viele dieser Methoden wurden im betreffenden Programm erstmalig implementiert. Ein Beispiel hierfür ist ein Gruppeneditor, in dem mehrere Anwender gleichzeitig einen Text bearbeiten können.

Problematisch an kommerzieller *collaboration aware* Software ist, daß sie meist aufwendiger und damit in der Tendenz teurer, fehleranfälliger und — durch längere Entwicklungszeiten — unmoderner als vergleichbare *collaboration unaware* Software ist. Daher zeigte sich frühzeitig die Tendenz, durch spezielle (*collaboration aware*) Software, CSCW-Funktionalität in Standard-Software einzubringen. Hier haben sich die sogenannten *application sharing* Programme durchgesetzt (siehe dazu Abschnitt 2.2.5).

## 2.2 Telekooperation

### 2.2.1 Einführung

Telekooperation ist ein Teilgebiet der computergestützten Gruppenarbeit (CSCW). Diese Technologie fördert Kooperation, bei der die Kooperationspartner zeitlich oder räumlich getrennt sind. Unterschiedliche Kommunikationstechniken werden dabei zur Überbrückung der Grenzen von Raum und Zeit eingesetzt. Telekooperation schließt jedoch im Gegensatz zur Telekommunikation die kommunizierenden Personen, ihre Handlungen sowie die Ziele und Zwecke ihrer Handlungen mit ein [58].

Telekooperation umfaßt den dritten und vierten Quadranten der Johansen *time-place* Matrix. Die in diesem Abschnitt vorgestellten Techniken Videokonferenz, *Whiteboard*, Gruppeneeditor und *Shared Application* sind vor allen Dingen für den Einsatz bei synchroner Telekooperation geeignet. Als Beispiel für asynchrone Telekooperation wird die Vorstellung von E-Mail den Abschnitt beenden.

### 2.2.2 Videokonferenzsysteme

Haupteinsatzgebiet für Videokonferenzsysteme ist die Durchführung einer Besprechung, ohne die örtliche Präsenz aller Teilnehmer vorauszusetzen. Innerhalb dieser „elektronischen Konferenz“ wird die Kommunikation der Teilnehmer über Audio- und Videoverbindungen realisiert.

Die ersten Videokonferenzen fanden in eigens dafür vorgesehenen Videokonferenzstudios statt. Diese Studios waren mit verschiedenen Kameras, die je nach Bedarf auf Personen, Tafeln, Flipcharts oder Dokumente gerichtet wurden, mehreren Monitoren, Lautsprechern und Mikrofonen sowie einem Kontrollpult ausgestattet. Vom Kontrollpult aus konnten die Bilder verschiedener Kameras zusammengestellt und an den oder die Empfänger gesendet werden. Aufgrund der Ausstattung ermöglichen diese Studios nicht nur die Kommunikation zwischen einzelnen Personen, sondern auch zwischen Personengruppen. Die Bildübertragung in klassischen Videokonferenzstudios erfolgte über das vermittelte Breitbandnetz (VBN) [126].

Die hohen Kosten und die geringe Flexibilität führten dazu, daß mobile Videokonferenzsysteme entstanden. Diese Systeme bestehen in der Regel aus einem großen Monitor, einer Videokamera, einem Lautsprecher und einem Mikrofon, die über ein Keyboard gesteuert werden können. Etwa zeitgleich entstanden die ersten Desktop-Konferenzsysteme (z. B. IVS [148], XTV [9], JVTOS [61]) die Videokonferenzen vom Arbeitsplatzrechner aus ermöglichen. Um auch bei geringeren Bandbreiten, etwa über das öffentliche ISDN-Netz, Videokonferenzen zu ermöglichen und eine plattformübergreifende Zusammenarbeit zu gewährleisten, haben sich Kompressionsverfahren und Standards etabliert. Erwähnt sei hier die H.320 Reihe [67].

### 2.2.3 Whiteboards

Neben Audio-/Video-Konferenzsystemen haben sich im Gebiet Telekooperation sogenannte *Whiteboards* etabliert. Ein Whiteboard ist ein System innerhalb einer verteilten Netzwerkumgebung zum kooperativen Annotieren. Mehrere geographisch verteilte Anwender können an einer Konferenz teilnehmen. Der Schwerpunkt liegt dabei im Anbringen von skizzenhaften Anmerkungen, basierend auf ein in den Hintergrund des Whiteboards geladenes Rasterbild. Dies kann beispielsweise eine Seite eines Textdokumentes oder eine CAD-Zeichnung sein. Während einer Sitzung läuft auf jedem Arbeitsplatzrechner der einzelnen Konferenzteilnehmer eine Instanz des Whiteboards.

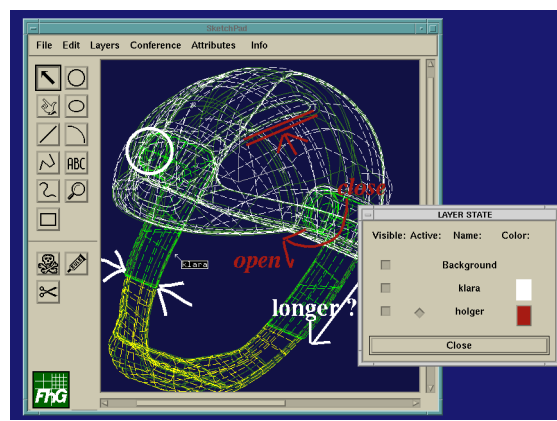


Abbildung 2: Benutzungsoberfläche des Distributed SketchPad

In den Anfängen der CSCW-Ära wurden in diesem Bereich einfache verteilte Zeichenoberflächen entwickelt. Mit VideoDraw [134] konnten zwei Designer Skizzen auf individuellen Zeichenoberflächen anbringen, die mit Hilfe von Videotechnik so kombiniert wurden, daß für die Designer die Illusion entstand, es handle sich um eine einzige gemeinsame Oberfläche. Ähnlich realisierte die TeamWorkStation [70] mit Videoüberlagerungstechniken eine verteilte Desktop-Oberfläche, die von jedem Konferenzteilnehmer eingesehen und mit Skizzen versehen werden konnte. Zusätzlich besaß jeder einen ihm zugeordneten elektronischen Zeiger (*Telepointer*). Das Videobild überlagerte den Computerbildschirm und erlaubte das manuelle Annotieren einer laufenden Applikation. Andere Systeme, wie GroupSketch [54] in seinen Anfängen, verwendeten vektorbasierte Bildschirme. Erst später verfügten Whiteboards über echte Telepointer, erlaubten das simultane Skizzieren (etwa ROCOCO SketchPad [134]) und konnten ein Rasterbild in den Hintergrund laden und allen Partnern an ihren Bildschirmen anzeigen (XGroupSketch [55], CaveDraw [88]). Abbildung 2 zeigt die Benutzungsoberfläche von Distributed SketchPad [99], einem voll objektorientierten Whiteboard mit einem speziellen Verfahren zur Generierung der Annotationen. Jeder Teilnehmer erhält ein oder mehrere elektronische Folien (*Layer*), auf denen seine Anmerkungen in Form von Annotationsobjekten in einer ihm zugeordneten Farbe aufgetragen werden. Diese Folien werden im Whiteboard übereinander gelegt. Die unterste Folie ist für das Hintergrundbild vorgesehen. Jede Annotation ist einer Folie zugeordnet

und kann nur vom Besitzer dieser Folie manipuliert werden. Zusätzlich besitzt jeder Partner einen Telepointer, um auf seinen Fokus des Interesses zu zeigen.

Die prinzipielle Architektur eines Whiteboards besteht aus mehreren Whiteboard-Clients, die über ein Netzwerk mit dem Whiteboard-Server verbunden sind. Dieser Ansatz ist in Abbildung 3 wiedergegeben. Jeder Anwender hat auf seinem Arbeitsplatzrechner einen Whiteboard-Client laufen, der die Benutzungsschnittstelle darstellt. Skizziert ein Benutzer eine neue Anmerkung, wird eine elektronische Nachricht vom Client generiert und an den Server versendet (1.). Dieser speichert die neue Annotation (2.) und schickt dann die Nachricht weiter an alle anderen Clients (3.). Die Nachricht wird bei diesen verarbeitet und die Annotation wird angezeigt (4.). Kommt ein neuer Client zu einer laufenden Sitzung hinzu (5.+6.), schickt ihm der Server die gespeicherten Annotationen inklusive des Hintergrundbildes sowie Informationen zu den anderen Teilnehmern, so daß der neue Teilnehmer sofort auf dem aktuellen Stand ist (7.+8.). Zusätzlich multiplext der Server die Daten des Neuankömmlings an die bereits angemeldeten Clients (3.).

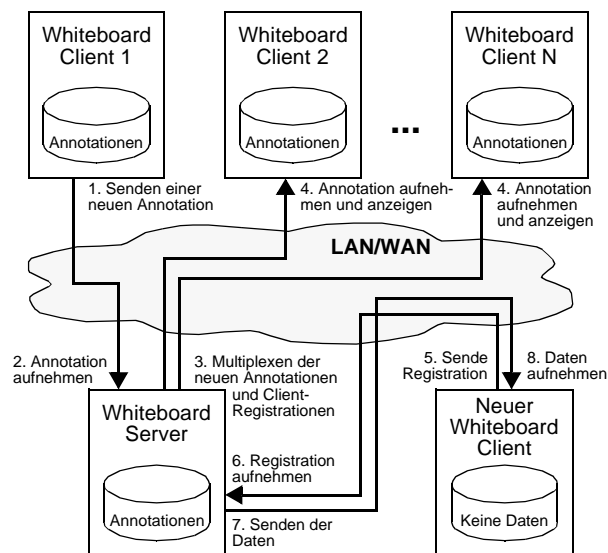


Abbildung 3: Prinzipielle Architektur eines Whiteboards

## 2.2.4 Gruppeneditoren

Gruppeneditoren sollten an dieser Stelle ebenfalls erwähnt werden. Sie erlauben einem Autorenteam gemeinsam ein Dokument zu editieren und zu verwalten. Für den asynchronen Fall muß ein solches System die zu unterschiedlichen Zeitpunkten gemachten Zugriffe auf ein Dokument synchronisieren und Überschneidungen unterbinden. Diese Funktionalität wird von Mechanismen, die ähnlich oder gleich den von konkurrierenden Datenbankzugriffen sind, geregelt. In einem einfachen Beispiel betreffen Transaktionen und Sperren nur einzelne Dateien, etwa dann, wenn die Autoren entscheiden, jedes Kapitel in einer einzelnen Datei unterzubringen. In einem etwas komplexeren Anwendungsfall - existierende Lösungen wie ShrEdit [135] oder CoMedia [132] unterstützen diese Variante - können

Autoren einzelne Paragraphen für sich zum Lesen oder Schreiben markieren. Die Transaktionsmechanismen der Gruppeneditoren stellen wie in Datenbanken sicher, daß die vier ACID Eigenschaften für jeden Entität des Dokumentes gewährleistet sind: *atomicity*, *consistency*, *isolation* und *durability* (siehe [40] für eine eingehende Besprechung von Datenbankproblematiken, [37] oder [98] für eine Diskussion über *concurrency control* in Groupware-Systemen). Diese Eigenschaften werden von den Systemen auf den synchronen Fall, in dem mehrere Autoren gleichzeitig am Dokument arbeiten, übertragen. Einige Systeme wie CoMedia unterstützen zusätzlich WYSIWIS (*What You See Is What I See*) Eigenschaften und bieten die Möglichkeit, Anmerkungen in Form von Text oder Graphiken am Dokument anbringen zu können.

Neben den Datenbankproblematiken des konkurrierenden Zugriffs auf gemeinsame Daten müssen WYSIWIS Gruppeneditoren zusätzlich die Auswirkungen solcher Zugriffe auf der Benutzungsschnittstelle beachten. Transaktionsprozesse müssen nicht nur das System, sondern auch den Anwender berücksichtigen. Unterschiedliche Mechanismen für konkurrierende Zugriffe haben daher unterschiedliche Auswirkungen auf die Benutzungsschnittstelle und wie sie sich dem Anwender darstellen [56].

### 2.2.5 Verteilung von Standard Applikationen

Um Dokumentenverarbeitungs-Systeme oder auch andere Standard-Applikationen (*single user applications*) kooperativ nutzen zu können, hat sich die Technologie des *sharing* durchgesetzt. Ein kurzer Exkurs soll *sharing* als Basistechnologie im Groupware-Bereich einführen.

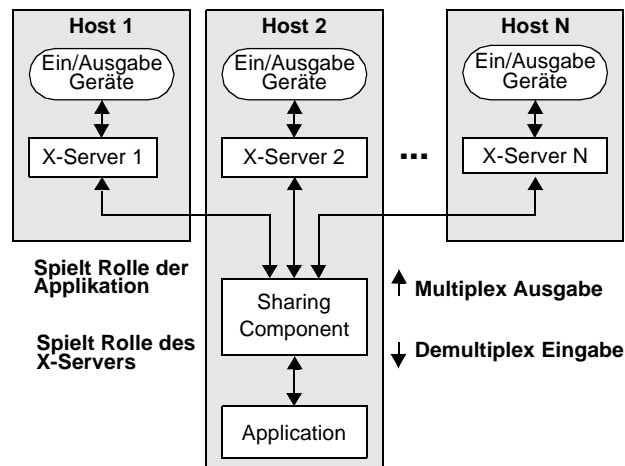


Abbildung 4: Konzeptueller Aufbau einer Sharing Komponente

Moderne *sharing*-Komponenten erlauben sogenanntes *joint viewing* und *joint editing*. Die graphische Ausgabe einer Standard-Applikation wird in einem Netzwerk simultan an mehrere Rechner verteilt und dargestellt, so daß die Konferenzteilnehmer die gleiche Sicht auf beispielsweise ein Dokument haben. Die Eingabe an die Applikation wird durch einen *floor-control* Mechanismus geregelt. Ein Autor muß den *floor* besitzen, um das Recht auf

Eingabe zu haben. Es gibt meist unterschiedliche Regelungen für die Übergabe des Rechtes auf Eingabe. Explizite Verfahren erwarten eine konkrete Anfrage eines Nutzers, der eine Eingabe machen will und entsprechend eine Freigabe vom Besitzer des *floor*s. Implizite Verfahren geben den *floor* sofort ohne Nachfrage beim Besitzer weiter. Dieses Verfahren eignet sich besonders für *brainstorming*. Das Lehrer-Schüler Szenario stellt beispielsweise ein Mischverfahren dar. Der Lehrer bekommt immer implizit das Eingaberecht, ein Schüler muß es explizit erfragen. Beispiele für *sharing*-Komponenten sind Shared X [11], GroupX [95] oder XTV [9], einen Überblick geben [76] und [89].

Die Idee des *sharing* besteht zunächst darin, die heutzutage übliche Graphikchnittstelle, die die Ausgabe auf dem Endgerät realisiert, zu ersetzen (siehe Abbildung 4). Die *Sharing*-Komponente simuliert dann applikationsseitig das Protokoll der Schnittstelle und multiplext die Ausgabe an die eigentlichen Schnittstellen auf den angeschlossenen Rechnern. Hier können spezielle Prozesse notwendig sein (der X-Server des X11 Window Systems leistet dies bereits), um die Verarbeitung zu unterstützen. Auf der anderen Seite werden die Eingaben der Rechner in der *sharing*-Komponente demultiplext und durch einen speziellen Mechanismus, der vom *floor-controlling* gesteuert wird, an die Standard-Applikation weitergeleitet.

## 2.2.6 E-Mail

E-Mail dient der asynchronen Weiterleitung und Verteilung von Nachrichten. Die CCITT Empfehlung X.400 bietet eine Möglichkeit zur Realisierung von E-Mail. Eine Nachricht wird vom Ersteller zum Empfänger über eine (beliebige) Anzahl von *Message Transfer Agents* (MTA) verschickt. Diese MTAs speichern eine Nachricht solange, bis sie sie erfolgreich an den nächsten MTA weiterleiten konnten. Die Verwaltung der Nachrichten am Sende- bzw. am Empfangsort wird durch einen Mailserver (MS) geregelt. Informationen zum Standard X.400 finden sich beispielsweise in [122].

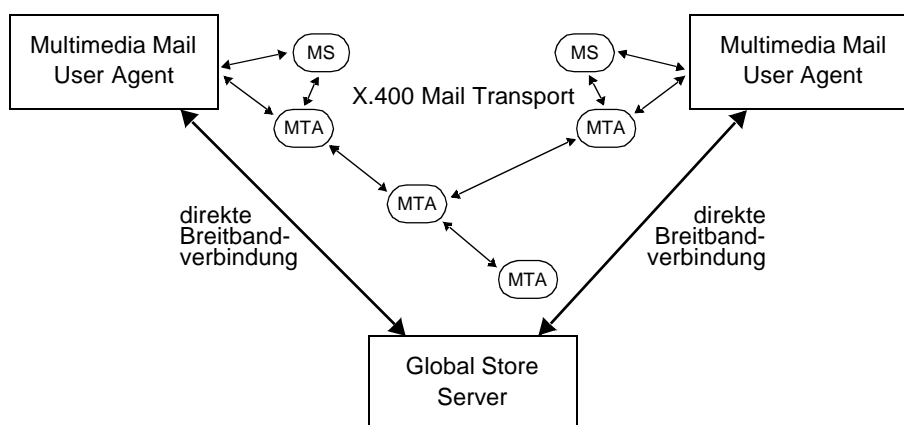


Abbildung 5: Das BERKOM Multimedia Mail Szenario (nach [97])

Auf dieser Empfehlung basiert auch das in Abbildung 5 dargestellte Multimedia Mail (MMM) Szenario des BERKOM Multimedia Teleservice Projektes der DeTeBerkom

[23][97]. Mit einer E-Mail Nachricht können hier Verweise auf externe multimediale Objekte mitgeführt werden. Jeder Anwender besitzt einen MMM *User Agent*, der Nachricht und Verweis zu einem gemeinsamen multimedialen Dokument zusammensetzt. Dazu löst er den Verweis auf, baut eine Verbindung zu dem dort angegebenen externen Server (*Global Store*) auf und erfragt die multimedialen Daten. Die Übertragung der Daten erfolgt nach dem Szenario der DeTeBerkom über eine breitbandige direkte Verbindung.

Verschiedene Hersteller wie etwa Lotus Notes oder Microsoft haben ihre eigenen E-Mail-Systeme. Fast alle bieten aber einen Übergang zum Internet-Standard SNMP (siehe etwa [138]) an. Hierdurch wird, letztendlich, die Möglichkeit zum weltweiten Austausch von elektronischen Nachrichten auch zwischen heterogenen Lösungen geschaffen.

## 2.3 Konferenzumgebungen

### 2.3.1 Einführung

Konferenzumgebungen vereinigen unterschiedliche Groupware-Technologien, wie Video-konferenz, Shared Application oder Whiteboard zu einem integrierten System. Wesentlich ist, daß ein oder mehrere zentrale Verwaltungskomponenten das Management zwischen den einzelnen Bestandteilen der Umgebung übernehmen. Meist wird hier auch die Verwaltung der Teilnehmer administriert.

Die folgenden beiden Abschnitte beschreiben den Aufbau zweier unterschiedlicher Konferenzumgebungen: der BERKOM MMC Service und der Standard T.120 der International Telecommunication Union (ITU).

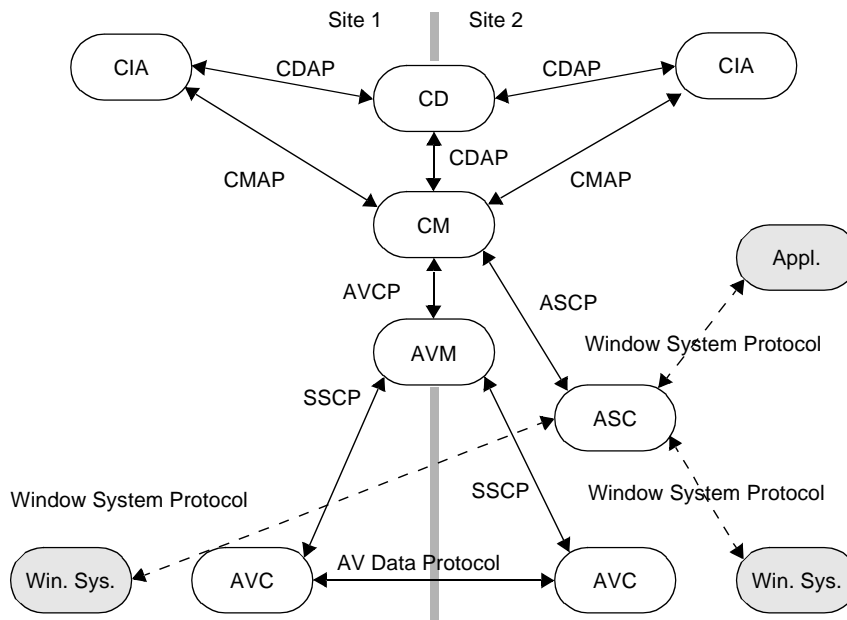
### 2.3.2 Der BERKOM Multimedia Collaboration Dienst

Die Basisarchitektur des BERKOM Multimedia Collaboration Services (MMC) [Alt93] ist in Abbildung 6 dargestellt. Der MMC Service ist Teil der DeTeBerkom multimedia Teleservices. Die Umgebung setzt sich aus folgenden Komponenten zusammen:

- Konferenzmanager
- Konferenzdatenverwaltung
- Benutzungsoberfläche
- Audio-/Video-Komponente
- Sharing-Komponente

Der Konferenzmanager (CM) startet und verwaltet Konferenzen. Die Konferenzdatenverwaltung (Conference Directory, CD) hält dazu alle für eine Konferenz notwendigen Daten wie Teilnehmer und aktuell laufende Konferenzen bereit. Die Benutzungsoberfläche wird durch den Conference Interface Agent (CIA) realisiert. Ein Benutzer meldet hierüber den Wunsch für eine neue Konferenz an. Die Teilnehmer werden eingeladen und eine Konferenz wird etabliert. Die dazu notwendigen Nachrichten werden über die entsprechenden

Protokolle CMAP (Conference Manager Access Protocol) und CDAP (Conference Directory Access Protocol) ausgetauscht.



**Abbildung 6: Basisarchitektur eines CSCW-Systems am Beispiel des BERKOM Multimedia Collaboration Services MMC (aus [Alt93])**

Um eine Diskussionsgrundlage in die Sitzung einbringen zu können, kann mit der Application Sharing Component (ASC) die Ausgabe einer Applikation auf die Bildschirme der einzelnen Konferenzpartner verteilt werden. Die Kontrolle des Eingaberechtes (*floor-controlling*) obliegt zwar der Sharing-Komponente, wird aber durch den Konferenzmanager mitverwaltet. CM und ASC tauschen über das Application Sharing Control Protocol (ASCP) die entsprechenden Anweisungen aus.

Die Audio-/Video-Übertragung wird durch den Audio Visual Manager (AVM) und die Audio Visual Component (AVC), die sich über das Single Source Control Protocol (SSCP) austauschen, ermöglicht. Dabei koordiniert der AVM die Etablierung der Videokonferenz, während die AVC die Einrichtung der Verbindung zwischen den Audio-/Video-Endgeräten und die Übertragung der entsprechenden Ströme übernimmt. Diese Übertragung erfolgt über einen separaten Kanal.

Das BERKOM-Konsortium, bestehend aus unterschiedlichen Industriefirmen und wissenschaftlichen Einrichtungen, definierte neben der Basisarchitektur die Protokollelemente der oben angesprochenen Protokolle. Die Ausprägungen der einzelnen Komponenten können so von unterschiedlicher Seite implementiert werden. Die Interoperabilität wird durch die Protokolle gewährleistet.



### 2.3.3 Der Standard T.120

Das T.120 Protokoll [68] definiert einen Standard zur Kommunikation mittels unterschiedlichster Formen von Daten und telematischen Medien zwischen zwei oder mehr Teilnehmern sowie zum Administrieren einer Konferenzumgebung. Der T.120 bietet dazu Multipunkt-Datenkommunikationsdienste an, die für alle Formen multimedialer Daten adäquat sind. Weiterhin ist das Management von echtzeit Sprach- und Videoübertragungen, die auf separaten Datenkanälen außerhalb der T.120 Protokoll Serie gesendet werden, definiert.

Das T.120 Protokoll umfaßt

- die Unterstützung zum Aufbau von Konferenzen oder Konferenzgruppen von Netzwerkelementen (etwa Konferenzterminals)
- Mechanismen zur Identifikation der teilnehmenden Elemente, zur umfassenden Information über die Elemente und zum Austausch der Fähigkeiten der Elemente
- Flexible Mechanismen zur Erstellung von Kommunikationswegen zwischen jeder möglichen Kombination dieser Elemente

Dabei kann das T.120 Protokoll simultan mehrere Sitzungen leiten. Jedes Terminal kann demnach gleichzeitig an mehreren Sitzungen teilnehmen. Die Kommunikationswege sind dabei frei definierbar, seien es Punkt-zu-Punkt, Sternförmige oder Multipunkt-Verbindungen. Letztere basieren auf dem Konzept von *Multipoint-Control-Units* (MCU). Abbildung 7 zeigt das Konzept der Multipunkt-Verbindungen. Einzelne Terminals sind zu Klastern verbunden und an einer MCU angeschlossen. Die MCUs sind wiederum miteinander verbunden. Vorteil dieser Architektur ist die deutliche Reduzierung des Datenstroms im gesamten Netzwerkverbund. Das T.120 Konsortium sieht diese Technologie als wesentlich für eine Verbreitung des Standards an.

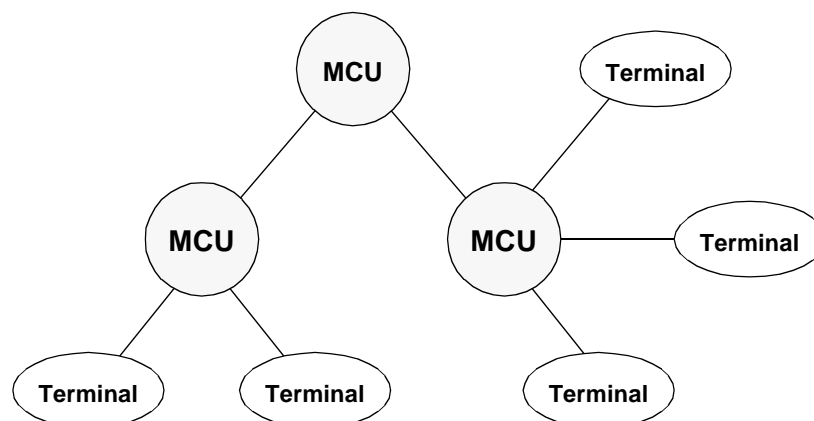


Abbildung 7: Multipunkt-Verbindungen im T.120

Abbildung 8 zeigt schematisch den Aufbau des T.120 Systemmodells. Die Netzwerkspezifischen Protokolle ermöglichen einen einfachen und offenen Zugriff auf standardisierte und nicht standardisierte Protokolle. Unterstützt werden zum Beispiel PSTN (*Public Switched Telephone Network*), ISDN (*Integrated Services Digital Network*) und PSDN (*Packet Switched Data Networking*).

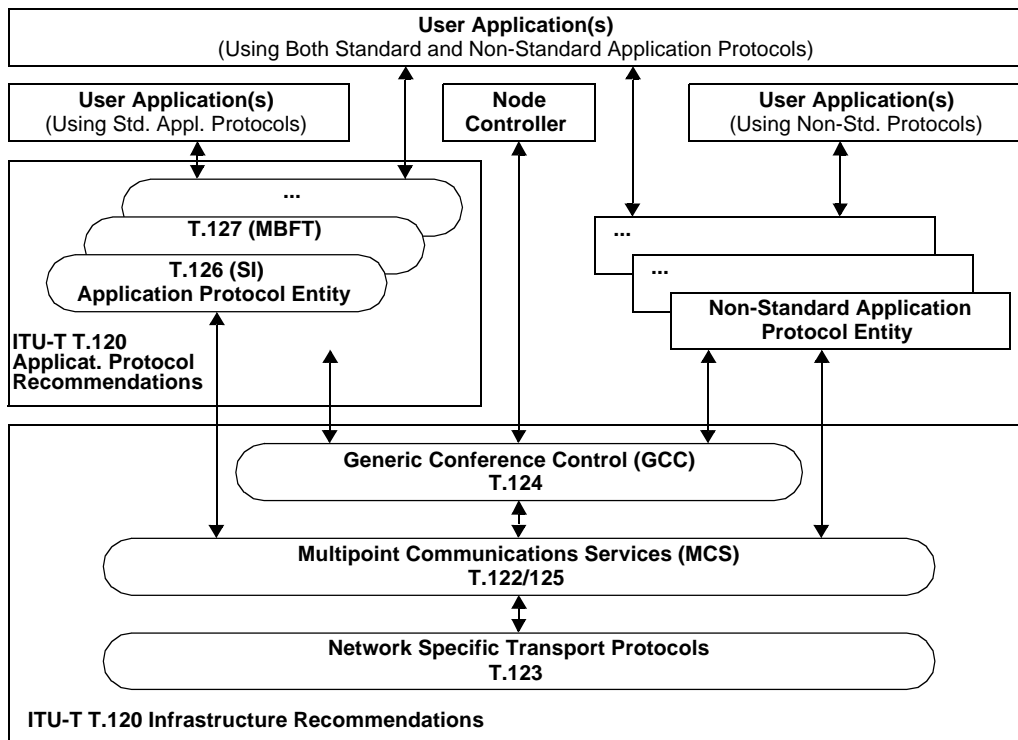
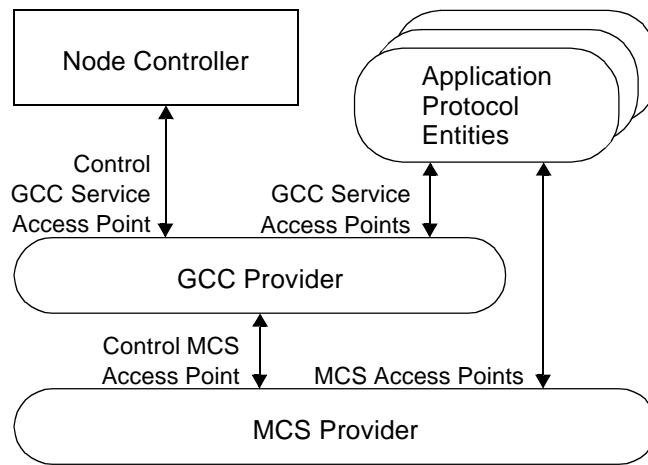


Abbildung 8: Das T.120 Systemmodell (aus [68])

Der Multipoint-Kommunikationsdienst (T.122 / T.125) verwendet das T.123 Protokoll und etabliert einen generellen Multipoint-Dienst, der verbindungsorientiert ist. Das oben angesprochene Konzept der MCUs wird hier realisiert. Auf diesen Dienst greifen sowohl außerhalb des Standards liegende Protokollentitäten als auch der generische Konferenz Manager (T.124) und die anwendungsorientierten Protokolle des T.120 zu. Das Aufsetzen und Verwalten einer Multipoint-Konferenz wird durch den GCC (*Generic Conference Control*) ermöglicht. Als Schnittstelle zum GCC existiert der *Control GCC Services Access Point*. Der *Node-Controller* steuert die GCC-Komponente über diesen Dienstbringerpunkt an und übernimmt damit die Administration einer Konferenz an einem Terminalendpunkt oder einer MCU (siehe Abbildung 9). Der Dienstzugangspunkt bietet sowohl Schnittstellen zum Starten, Beenden und zum Beitreten oder Verlassen einer Konferenz als auch zum Abfragen des Statuses von gerade laufenden Konferenzen.



**Abbildung 9: Service Access Points der T.120 Generic Conference Control Komponente (aus [69])**

Applikationen sind nicht Bestandteil des T.120. Applikationen, die den T.120 verwenden, können als *multipoint aware* bezeichnet werden (vergleiche mit Abschnitt 2.1.3 "Aware — Unaware" auf Seite 11). Der T.120 bietet dazu den GCC und den MCS an. Die *Application Protocols* des T.120 erweitern diese beiden Dienste zum Kommunikationsmanagement und zur Kommunikation selbst um:

- Austausch von Standbildern und Annotationen (*Whiteboard*)
- Verteilen von Standard Applikationen (*Shared Application*)
- Fax (alle in T.126 – *Still Image* – definiert)
- Dateiaustausch (T.127 – *Multipoint File Transfer*)
- Kontrolle von Audio-/Videostreamen (T.128 – *Audio-Visual Control*)

Mit dem T.121 (*Generic Application Template*) wurde zusätzlich ein Schema definiert, mit dem es dem Anwendungsprogrammierer leicht gemacht werden soll, die *Application Protocols* in seine Anwendung einzubinden.

## 2.4 Computergestützte Vorgangsbearbeitung

### 2.4.1 Einführung

Workflow Management als Baustein von CSCW, auch wenn die Ursprünge eher in der Büroautomation oder der integrierten Verarbeitung von Kontroll- und Steuerungsvorgängen im CIM-Bereich als im CSCW-Bereich liegen, ermöglicht die computergestützte Vorgangsbearbeitung. Der Begriff *Workflow Management* bezeichnet "eine Methode, um existierende Büroprozesse zu beschreiben und ihre Effizienz und Effektivität zu optimieren. Es stellt rechnergestützte Werkzeuge zur Verfügung, um Büroprozesse zu modellieren, zu

simulieren und zu realisieren, und es bietet das Potential, um die Informationsverarbeitung in die Organisation eines Unternehmens zu integrieren" [86].

Dies beinhaltet die Festlegung, Überwachung und zeitliche Steuerung einzelner Teilaktivitäten eines Geschäftsprozesses sowie die Bereitstellung und den Austausch dabei nötiger Dokumente. Ein Geschäftsprozeß oder Geschäftsvorfall bezeichnet dabei "eine Abfolge von Tätigkeiten, Aktivitäten und Verrichtungen, die der Erzeugung eines Produkts oder einer Dienstleistung dienen. Als Workflow Prozeß bezeichnet man jenen Bestandteil eines Geschäftsprozesses, der mit Hilfe von DV-Systemen automatisiert wurde" [80] (S.108).

Im folgenden werden die Inhalte von Workflow Management näher beleuchtet, bevor das Referenzmodell der Workflow Management Coalition, welches als erster Ansatz einer Standardisierung in diesem Bereich gelten kann, beschrieben wird.

### 2.4.2 Workflow Management

Charakteristisch für Workflow Management ist die Unterstützung der Modellierung von Geschäftsprozessen, die kontrollierte, systemgesteuerte Ausführung von Verarbeitungseinheiten und die Verwaltung von Informationen über vergangene, gegenwärtige und zukünftige Aufgaben [92]. Geschäftsprozesse lassen sich nach Formalisierbarkeit (fest strukturierte, teilstrukturierte, unstrukturierte Vorgänge), Auftretensart (einmalige, regelmäßige, zyklische Vorgänge) und Auftretenshäufigkeit (minütlich, stündlich, täglich, häufig, selten) unterscheiden.

Workflow Management unterstützt besonders regelmäßig wiederkehrende Aufgaben, die stark strukturiert sind, d.h. die

- eine Reihe von Aktivitäten in einer bestimmten Reihenfolge bzw. parallel zueinander umfassen,
- immer wieder in der gleichen oder ähnlichen Form auftreten,
- mehrere Personen bzw. Sachbearbeiter involvieren und
- einem starken Koordinationsbedarf unterliegen.

Vorgänge, die unregelmäßig auftreten und einen sehr geringen Strukturierungsgrad aufweisen, werden als Ad-hoc-Workflows bezeichnet.

Vor dem Einsatz eines Workflow Management Systems steht zunächst die Analyse der im Unternehmen vorhandenen Geschäftsprozesse. Jeder Geschäftsprozeß muß anschließend in eine Menge von Bearbeitungsschritten zerlegt werden. Zu jedem Arbeitsschritt gehören Kompetenzen und die Zuständigkeiten der Aufgabenträger. Ein Arbeitsschritt wird an einem Arbeitsobjekt, etwa einem Dokument, ausgeführt. Am Ende eines Arbeitsschritts steht ein Ergebnis, das an einen Termin gebunden sein kann. Zwischen verschiedenen Arbeitsschritten können zudem Abhängigkeiten bezüglich der Reihenfolge bestehen.

Mit dem nun verfügbaren Wissen über die Geschäftsprozesse erfolgt die Definition von Vorgangstypen für die Abbildung der Geschäftsprozesse in einem Workflow Management

System. Die so erhaltenen Vorgangstypen werden im Zuge der Modellierung als Vorgangsmodell oder Prozeßmodell abgelegt. Eine Simulation kann helfen, Vorgangsmodelle auf ihre logische Richtigkeit zu prüfen. Erst nachdem diese Schritte erfolgt sind, kann durch die Ablaufsteuerung das operative Geschäft unterstützt werden.

Die Definition der Vorgangstypen beinhaltet auch eine Zuordnung der Akteure zu jedem einzelnen Vorgangsschritt. Diese wird bei den meisten Systemen nicht direkt zu den Organisationsmitgliedern vorgenommen, sondern über ein Rollenkonzept. Das Rollenkonzept sieht eine Zuordnung jedes Vorgangsschrittes zu einer Rolle und von einer oder mehreren Rollen zu jedem Organisationsmitglied vor.

Der Übergang von Prozeßmodellierung zu operativem Workflow und die Rückführung von Analyseergebnissen wird derzeit nur unzureichend unterstützt. Einen Modell für einen ganzheitlichen Workflow Management Zyklus findet sich beispielsweise in [81]. Das Referenzmodell der Workflow Management Coalition bietet ebenfalls erste Ansätze.

### **2.4.3 Das Referenzmodell der Workflow Management Coalition**

#### **2.4.3.1 Überblick zum Standard**

Trotz der relativ alten Thematik und einer großen Produktvielfalt existieren derzeit keine fundamentalen Modelle oder Architekturen, die als Standard bezeichnet werden können [72]. Problematisch ist, daß die Produkte sehr unterschiedlich in Umfang und Eigenschaften sind und teilweise ihre eigenen "Standards" definieren. Zudem gibt es „zahlreiche E-Mail-Systeme am Markt, die sich mit Workflow Management-Funktionalität rühmen, diese aber nur in einem sehr reduzierten Umfang bereitstellen können“ [79] (S.105). Dieser Zustand schreckt viele potentielle Anwender ab, die abwarten wollen, bis sich klarere Verhältnisse abzeichnen. Dieser Problematik hat sich die Workflow Management Coalition [161] verschrieben.

Die Workflow Management Coalition (WfMC) ist eine Gruppe von Firmen aus dem Workflow-Umfeld, die derzeit als wesentlichen Meilenstein auf dem Weg zu einem Standard ein Workflow-Referenzmodell verfassen. Das Referenzmodell definiert ein generelles Architekturrahmenwerk für die weiteren Spezifikationen der WfMC. Es identifiziert fünf wesentliche Schnittstellen an der verarbeitenden Einheit eines Workflow Systems, der Workflow-Engine:

- Den In- und Export von Prozeßdefinitionen (Interface 1)
- Die Interaktion mit Client-Applikationen und Arbeitslistenhändlern (Interface 2)
- Die Inbezugnahme externer Software und Applikationen (Interface 3)
- Die Interoperabilität zwischen unterschiedlichen Workflow-Systemen (Interface 4)
- Die Administration und das Monitoren der Workflow-Engine (Interface 5)

Abbildung 10 zeigt das Referenzmodell mit den fünf Schnittstellen, deren Spezifikation die Hauptaufgabe der WfMC ist.

Die WfMC unterscheidet in zwei wesentliche Phasen, der *Buildtime* und der *Runtime*. In die *Buildtime* fällt die Geschäftsprozeßanalyse durch Modellierungs- und Definitionswerkzeuge. Der Übergang zur *Runtime* wird durch die Prozeßdefinition realisiert. Hierzu korrespondiert das Interface 1 der WfMC. Innerhalb der *Runtime* werden wiederum zwei Teilbereiche identifiziert. Die Prozeßinstanziierung und die Prozeßkontrolle, was den Kernbereichen der Workflow-Engine entspricht, und die Interaktion mit dem Anwender (Interface 2), mit externen Applikationen (Interface 3) und mit anderen Workflow-Engins (Interface 4).

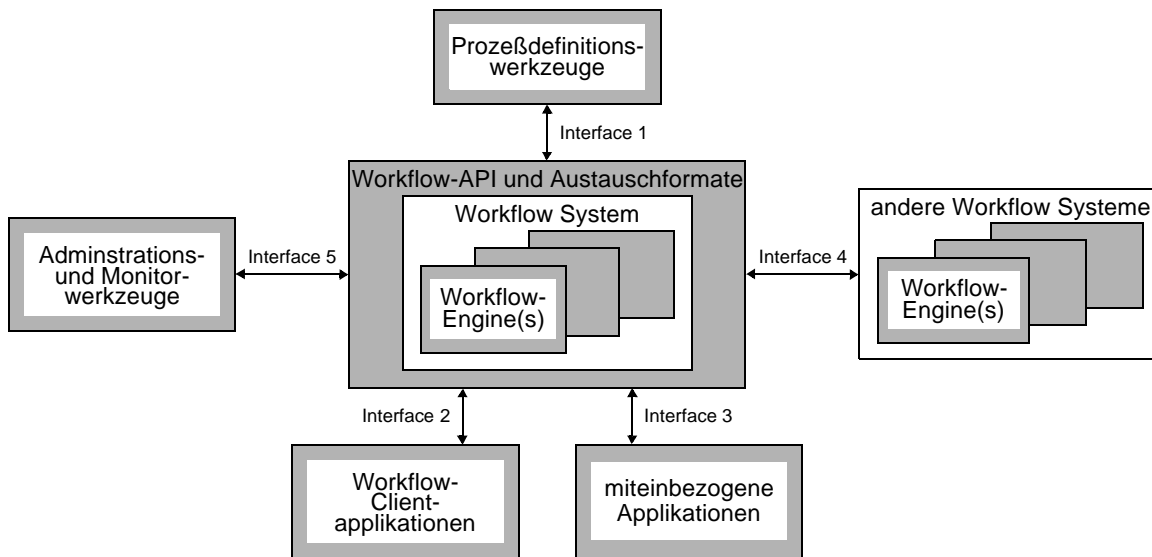


Abbildung 10: Das WfMC Referenzmodell (nach [161])

Jedes Interface definiert eine Schnittstelle, die durch ein *Application Programmers Interface* (API) und Austauschformate definiert wird. Die Terminologie wird von der WfMC in einem Metamodell beschrieben, welches Inhalt des nächsten Abschnitts ist. Der darauf folgende Abschnitt zeigt, wie die Interaktion mit der Workflow-Engine durch das Interface 2 ermöglicht wird.

### 2.4.3.2 Basisterminologien der WfMC

Die WfMC hat ein Metamodell zur Prozeßdefinition, welches eine Menge von Basisterminologien und ihre Beziehungen definiert. Abbildung 11 zeigt diese Relationen. Der linke Arm ordnet einem Geschäftsprozeß eine entsprechende Definition, bestehend aus Sub-Prozessen und Aktivitäten, zu. Der rechte Arm zeigt die Entitäten innerhalb des Workflow-Systems, das den Prozeß verwaltet und automatisiert. Hauptbestandteil ist die Prozeßinstanz, die den jeweiligen Geschäftsprozeß, spezifiziert durch die Prozeßdefinition, im Workflow-System repräsentiert. Sie beinhaltet, korrespondierend zur Definition, Instanzen der Aktivitäten. Eine Aktivität besteht aus einer oder mehreren Aufgaben (Arbeitseinheiten) und bezieht auch die u. U. dazugehörigen Applikationen oder Werkzeuge, mit der die Aufgabe bzw. die Aktivität bewältigt werden soll, mit ein. Das spezielle an einer Arbeitseinheit ist, daß sie genau einem Workflow-Teilnehmer zugeordnet ist. In der Regel ist dies eine

Person, die als sie selbst oder als menschliche Vertretung einer Rolle (z. B. Sachbearbeiter oder Stelle) auftritt. Aber auch eine maschinenbasierte Ressource, etwa ein intelligenter Software Agent, könnte konzeptionell als Workflow-Teilnehmer angesehen werden.

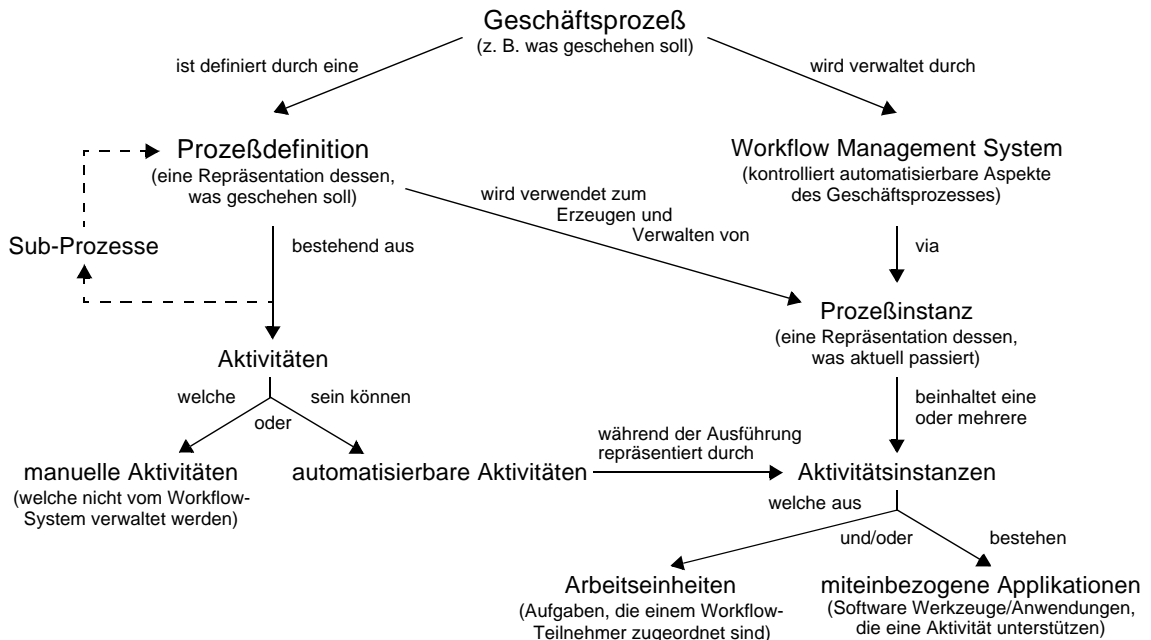


Abbildung 11: Relationen der Basisterminologien im Metamodell der WfMC (nach [161])

Die umgekehrte Relation wird durch Arbeitslisten beschrieben. Eine Arbeitsliste enthält alle Arbeitseinheiten, die einem Workflow-Teilnehmer (zu einem bestimmten Zeitpunkt) zugeordnet sind. Eine Arbeitsliste wird vom Arbeitslisten-Verwalter (*handler*) administriert.

Der Verwalter erfragt die einzelnen Arbeitseinheiten für einen Teilnehmer beim Workflow-System und verwendet dabei das Interface 2 der WfMC. Die Liste wird dann im allgemeinen dem Anwender dargestellt. Die Ergebnisse einer Interaktion mit dem Anwender, etwa die Fertigstellung einer Arbeitseinheit, werden dann wieder an das Workflow-System gemeldet.

### 2.4.3.3 Das Workflow Client Application Interface

Im Modell der WfMC wird die Interaktion zwischen einer Klientapplikation und dem Workflow-System durch das Konzept der Arbeitslisten – die Liste der Arbeitseinheiten, die einem bestimmten Teilnehmer zugeordnet sind – geregelt. Eine Arbeitsliste enthält Arbeitseinheiten von Aktivitäten einer oder mehrerer Prozessinstanzen. Der Austausch und die Manipulation von Arbeitseinheiten wird durch eine Vielzahl von Schnittstellenfunktionen, die im Interface 2 “Workflow Client Application Interface” [162] definiert sind, ermöglicht.

Die Aktivierung einer einzelnen Arbeitseinheit, etwa das Starten einer externen Applikation und deren Versorgung mit entsprechenden Dokumenten, kann entweder von der Clientapplikation oder vom Nutzer gesteuert werden. Teile der einer Arbeitseinheit oder Aktivität zugeordneten Daten können dazu notwendig sein. Verweise auf die von externen Applikationen benötigten Daten (etwa das aktuell zu bearbeitende Dokument) können in den Arbeitslisten als Attribut abgelegt werden.

Das Interface 2 bietet neben Operationen zur Verwaltung von Arbeitslisten Möglichkeiten zur Erfragung und Manipulation von Prozeß- und Aktivitätsinstanzen. Zusammengefaßt beinhaltet das Interface 2 im wesentlichen die folgenden Operationen:

- Sitzungsaufbau – Auf- bzw. Abbau der Verbindung zwischen den beteiligten Systemen
- Workflow-Definition – Abfragen von Workflow-Prozeßdefinitionen und deren Attributen
- Prozeßkontrolle und -status – Kreieren, Starten und Stoppen von Prozeßinstanzen; Suspendieren und Wiederaufnehmen individueller Prozeßinstanzen; Forcieren von Statuswechseln innerhalb einer Prozeß- oder Aktivitätsinstanz; Zuordnen von Attributen zu individuellen Prozeß- oder Aktivitätsinstanzen; Abfragen von Attributen und Status mit speziellen Filterkriterien
- Arbeitslistenkontrolle – Abfragen von Arbeitslisten mit speziellen Filterkriterien; Abfragen von Arbeitseinheiten; Notifikation über Selektion, Neuordnung, Fertigstellung von individuellen Arbeitseinheiten; Zuordnung und Abfrage von Arbeitseinheitsattributen
- Prozeßmanipulation – Veränderung des operationalen Zustands einer Workflow-Prozeßdefinition bzw. -instanz; Statusveränderungen von Prozeß- oder Aktivitätsinstanzen eines bestimmten Typs; Attributzuordnung zu Prozeß- oder Aktivitätsinstanzen eines bestimmten Typs; Terminierung von Prozeßinstanzen (diese Operationen sollten laut Empfehlung der WfMC nur ausgezeichneten Nutzern erlaubt werden)
- Datenverwaltung – Abfragen und Abspeichern von Workflow- oder Applikationsdaten

#### **2.4.4 Dokumentenmanagement und Archivierung**

Dokumentenmanagement befaßt sich mit dem Einlesen, Drucken, Erstellen, Weiterleiten, Löschen, Archivieren und Wiederfinden von Dokumenten. Die wesentlichen Probleme, die bei der herkömmlichen Arbeit mit Papier entstehen, wie Verteilungs-, Archivierungs-, Verwaltungs- und Bearbeitungsprobleme sollen durch den Einsatz von Dokumentenmanagement- und Archivierungs-Systemen ausgeräumt werden. Dabei sind vor allem Kostenfaktoren die treibende Kraft für den Einsatz. Die Kosten für Ablage, Wiederbeschaffung, Kopien und Lagerung können zwischen 25% und 90% gesenkt werden. Weitere Vorteile sind eine Steigerung der Produktivität durch einen Anstieg der Transaktionen pro Mitarbeiter und eine Verbesserung der Prozeßqualität durch die Reduzierung von Fehlerquellen und Medienbrüchen und einer Verbesserung bei Informationsbeschaffung, Zugänglichkeit und Dokumentensicherheit [149]. Die Komponenten eines Dokumentenmanagement-Systems sind in Abbildung 12 dargestellt.



Ähnlich dem Dokumentenmanagement repräsentiert, speichert und organisiert Informationsmanagement Wissen (Information) zum Zwecke der Wiedergewinnung. Wegen seiner Popularität sei an dieser Stelle das World Wide Web (WWW) [21] als eines der meistbenutzten Systeme zur Informationsbeschaffung und quasi Standard erwähnt. Dokumente können hier in sehr einfacher Weise durch *Hyperlinks* miteinander verknüpft werden. Dies erlaubt Anmerkungen, die durch ein weiteres Dokument repräsentiert und an das Original durch *Hyperlinks* gebunden werden. Mit InterNote [29] wurde bereits 1989 ein WWW-ähnliches System vorgestellt, daß Dokumente mit Hilfe von *Links* miteinander verknüpft und durch den gleichen Mechanismus Anmerkungen unterstützt. Der kommerzielle WWW-Browser Netscape bietet diese Möglichkeit jetzt auf dem Internet an.

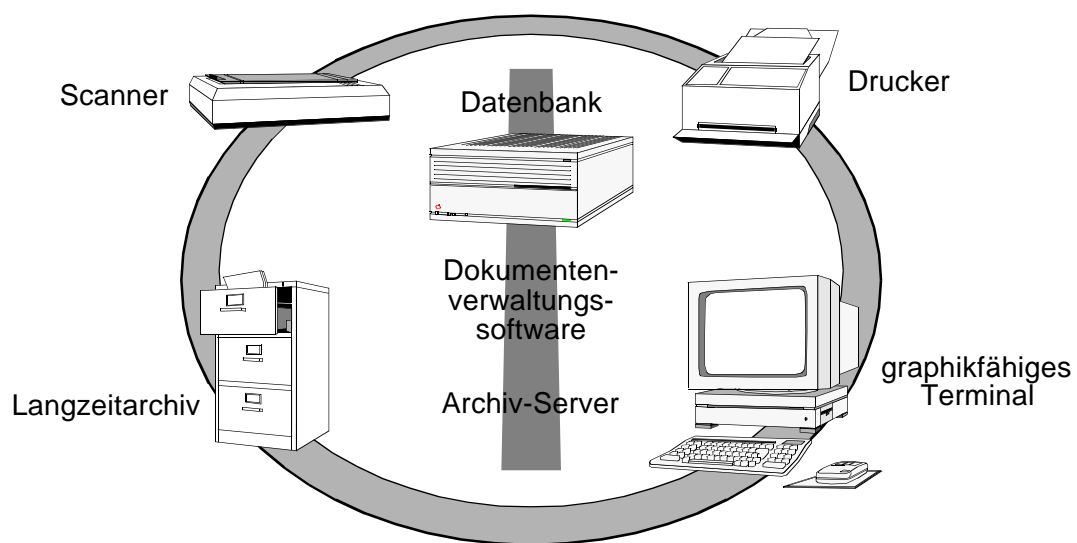


Abbildung 12: Architektur eines Dokumenten-Management-Systems (nach [149])

Dokumentenmanagement (oder Informationsmanagement) ist oft mit Workflow-Systemen gekoppelt anzutreffen. Zumindest bieten die meisten kommerziellen Workflow-Systeme ein Dokumentenmanagement und umgekehrt Dokumentenmanagement-Systeme meist die Möglichkeit zur Weiterleitung von Dokumenten an (auch wenn dies nicht als operativer Workflow angesehen werden kann).

Dies spiegeln auch die Ziele der Document Management Association (DMA) wider. Neben der Definition von gemeinsamen, nicht-proprietären Schnittstellen für Applikationen, Bibliothek-Systemen und Desktop-Anwendungen, sieht sie die Sicherstellung der Kompatibilität zu Standards, die von der Workflow Management Coalition (WfMC) entwickelt wurden, als wesentlich an. Allgemein ist der Trend einer Annäherung von Dokumenten- und Workflow-Management hier zu erkennen.

## 2.5 Software Agenten

### 2.5.1 Einführung

Auf die Frage “Was ist ein Software Agent?” gibt es derzeit keine einheitliche Antwort. Allgemein wird oft von einem Agenten als jemand, der im Auftrag eines anderen eine Aufgabe erledigt, gesprochen. Wooldridge und Jennings [158] charakterisieren Agenten über deren potentielle Attribute. Sie bezeichnen Agenten als Soft- (oder auch Hard-)ware Computersysteme, die zumindest die folgenden Eigenschaften besitzen:

- **Autonomie:** Agenten operieren ohne die direkte Intervention von Menschen oder anderen Entitäten und haben ein gewisses Maß an Kontrolle über ihre Aktivitäten und ihren Status.
- **Sozialfähigkeit:** Agenten interagieren mit anderen Agenten (und u. U. Menschen) mit Hilfe einer Art Agenten-Kommunikations-Sprache.
- **Reaktionsfähigkeit:** Agenten nehmen ihre Umwelt (was die physikalische Welt, eine virtuelle Umgebung, ein Benutzer durch eine Benutzungsschnittstelle, andere Agenten, das Internet oder eine Kombination dieser sein könnte) wahr und reagieren auf Veränderungen in ihr.
- **Pro-Aktionsfähigkeit:** Agenten reagieren dabei nicht einfach nur auf eine Veränderung der Umgebung, sondern sie sind fähig, dies in einer gewissen zielgerichteten Art zu tun, in dem sie die Initiative ergreifen.

Weitere Anforderungen an Agenten können beispielsweise sein: Glaubwürdigkeit – ein Agent muß (aufgrund seines Wissen) die “Wahrheit” sagen – oder Verpflichtung – Agenten, die einen Dienst anbieten, verpflichten sich, ihn auf Anfrage auch auszuführen [52].

Zwei wichtige Aspekte sollen nun in den folgenden beiden Abschnitten genauer beleuchtet werden. Zum einen, wie organisieren sich Agenten untereinander und welche allgemeinen Strukturen bilden sie dabei. Abschnitt 2.5.2 gibt dazu einen Überblick über unterschiedliche Architekturen von Agentensystemen. Zum anderen soll eine Möglichkeit aufgezeigt werden, wie in allgemeiner Form Informationen ausgetauscht werden können. Abschnitt 2.5.3 stellt dazu die *Knowledge Query and Manipulation Language* (KQML) vor. Im letzten Abschnitt dieses Kapitels wird dann als Beispiel für eine Agentenplattform ASAP vorgestellt. ASAP bietet eine Basis für Agenten und ermöglicht die einfache Realisierung von Multi-Agenten-Systemen.

### 2.5.2 Architekturen von Agentensystemen

#### 2.5.2.1 Mobilität von Agenten

Grundsätzlich sollte unterschieden werden, ob Agenten mobil oder statisch sind. Mobile Agenten “bewegen” sich über Rechnergrenzen hinweg zu anderen Rechnern, sammeln dort die von ihrem Auftraggeber gewünschten Informationen und kehren zurück. Eine kurze Einführung hierzu wird im folgenden Abschnitt “Mobile Agenten” gegeben.

Statische Agenten hingegen bleiben innerhalb ihrer Rechengrenze und kommunizieren mit anderen Agenten in einem "Frage-und-Antwort-Spiel", um die benötigten Informationen zu erhalten. Abschnitt 2.5.2.3 beschreibt Mechanismen, die in einer solchen Multi-Agenten-Welt zum Tragen kommen. Grundsätzlich können die dort gemachten Aussagen zu statischen Agenten auf mobile Agenten übertragen werden. Ein Multi-Agenten-Szenario kann weiterhin natürlich sowohl statische als auch mobile Agenten beherbergen.

### 2.5.2.2 Mobile Agenten

Basis für mobile Agenten ist eine Umgebung, die Transaktionen zwischen Software-Applikationen über ein Netzwerk ermöglicht. Beispiele hierfür sind *Telescript* von der Firma General Magic, *ToolTalk* von SUNSoft oder *JAVA* von SUN. In der Umgebung wird es Agenten ermöglicht, über das Netzwerk zu "laufen" und dabei sowohl Daten als auch Ausführungscode mitzuführen. Auf der Zielmaschine wird dann der Code innerhalb der Umgebung durch einen entsprechenden Interpreter ausgeführt und die Ergebnisse den bereits bestehenden Daten hinzugefügt. Wichtig ist dabei, daß die Umgebung nur sichere, nicht "zerstörerische" Zugriffe auf die lokale Rechnerumgebung zuläßt (zur Thematik Sicherheit bei mobilen Agenten siehe etwa [131]). Ist die Ausführung auf einer Maschine beendet, "wandert" der mobile Agent weiter, um weitere Informationen zu sammeln, oder er kehrt zum Auftraggeber zurück.

Interessant sind mobile Agenten vor allem im Internet und World Wide Web Bereich. Das Sammeln spezifischer Informationen soll erleichtert werden. *Webhound* beispielsweise stellt dem WWW-Benutzer einen persönlichen Dokumenten-Filter-Agent zur Verfügung, der auf Basis einer Analyse der Vorlieben des Benutzers regelmäßig eine Empfehlung für neue WWW-Seiten gibt [153]. *Bargain Finder* ist in der Lage, zu einem beliebig eingegebenen CD-Titel das preisgünstigste Angebot bei den im Internet vertretenen Anbietern zu ermitteln [17]. Eine eingehendere Diskussion zu mobilen Agenten ist beispielsweise in [90] zu finden.

### 2.5.2.3 Statische Agenten

Statische Agenten bleiben im Gegensatz zu mobilen Agenten in ihrer Rechnerumgebung. Um Informationen zu erhalten und Aktionen ausführen zu können, ist ein statischer Agent auf die Dienste anderer Agenten angewiesen und muß mit ihnen kommunizieren. Welche Dienste angeboten werden und wie dies ein Agent in Erfahrung bringt wird durch unterschiedliche Konzepte abgebildet. Agenten können beispielsweise eine Ausschreibung machen, die darauf eingehende Angebote sammeln, sie sondieren und dann die Entscheidung treffen, welcher Angebotssteller den Auftrag zum Ausführen des spezifischen Dienstes erhält. Umgekehrt können Agenten mit ihren Fähigkeiten werben, dabei aber auch gleichzeitig ihre Bedürfnisse angeben. Andere Agenten können diese Informationen dann nutzen, um ihre eigenen Aktivitäten zu koordinieren [52]. Problematisch ist dabei der Kommunikations- und Koordinationsaufwand. Wird eine direkte Kommunikation gewählt, in der jeder Agent selbstverantwortlich die Koordination übernimmt, ist der Aufwand sehr hoch.

Durch den Einsatz spezieller Agenten, sogenannter Facilitatoren, und der Bildung von Föderationen kann der Aufwand zum Aushandeln stark reduziert werden [52]. Eine Gruppe von Agenten (etwa alle Agenten, die sich auf einem Rechner befinden) schließen sich zu einer Föderation zusammen, die nach außen hin durch einen Facilitator vertreten wird. Dieser kommuniziert mit anderen Föderationen über deren Facilitatoren. Ein Facilitator kennt dazu alle Fähigkeiten und Bedürfnisse seiner Föderation. Abbildung 13 zeigt ein föderatives Agentensystem graphisch.

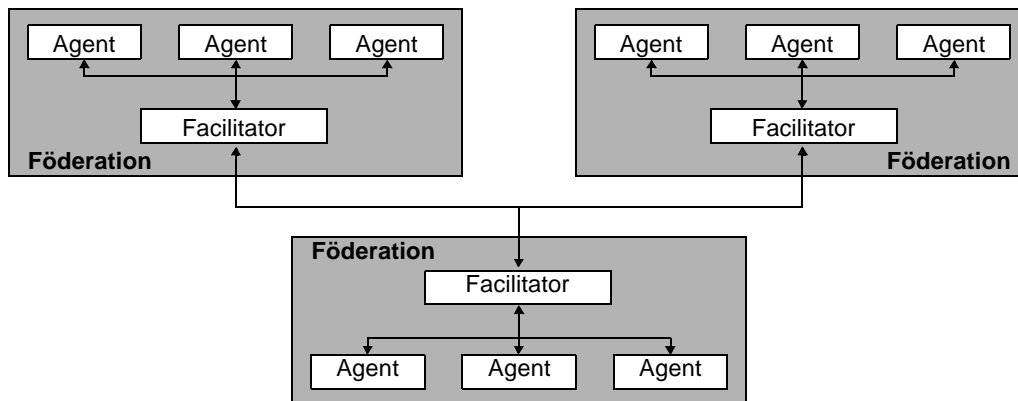


Abbildung 13: Föderatives Agentensystem

Das Konzept der Facilitatoren soll noch etwas eingehender diskutiert werden. Ein streng föderatives System hat den Nachteil, daß alle Aktionen über den Facilitator laufen müssen und damit durch die Kapazität und die Fähigkeiten des Facilitators beschränkt ist. Weicht man das Konzept auf und erlaubt daneben auch die direkte Kommunikation zwischen Agenten, ergeben sich vielseitige Möglichkeiten zum Einsatz von Facilitatoren. Die folgenden Beispiele dafür sind [45] entnommen und in Abbildung 14 dargestellt. Die in Klammern angegebenen Ausdrücke, sogenannte *performatives*, sind ein Vorgriff auf die Agentensprache KQML, die im nächsten Abschnitt beschrieben wird.

1. Ist ein Agent sich über die Dienste eines anderen Agenten bewußt und benötigt sie, kann ein einfaches Frage - Antwort (*ask - tell*) Protokoll verwendet werden, ohne den Facilitator zu verwenden.
2. Ein Agent interessiert sich für eine bestimmte Information oder Informationstyp und meldet dieses Bedürfnis beim Facilitator an (*subscribe(ask)*). Ein anderer Agent bietet diese Information dem Facilitator an (*tell*). Der Facilitator teilt diese Information automatisch dem ersten Agenten mit (*tell*).
3. Ein Agent beauftragt den Facilitator eine bestimmte Information zu besorgen (*broker(ask)*). Andere Agenten teilen dem Facilitator mit, daß sie bestimmte Informationen anbieten (*advertise(ask)*). Der Facilitator bestimmt den "richtigen" Agenten zur Beantwortung der Frage des ersten Agenten, fragt (*ask*), bekommt die Antwort vom gefragten Agenten (*tell*) und teilt sie dem ersten Agenten mit (*tell*).

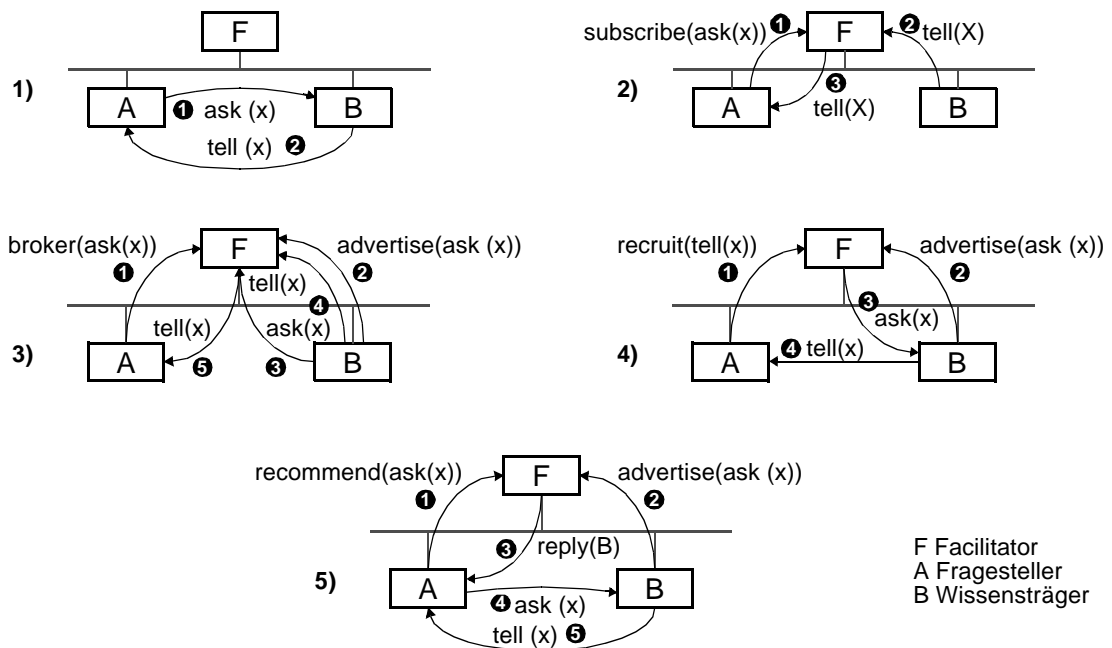


Abbildung 14: Facilitatoren in Multi-Agenten-Systemen (nach [45])

- Als Alternative kann die Antwort auch direkt vom gefragten Agenten an den Fragenden gesendet werden. Der Facilitator dient hier nur zum rekrutieren (*recruit(tell)*) des Beantworters.
- Der Facilitator dient als Vermittler und liefert auf eine entsprechende Anfrage (*recommend(ask)*) einen Vorschlag für eine (oder mehrere) Agenten, die die Frage beantworten können (*reply*). Der Fragesteller kann dann durch das Frage - Antwort (*ask - tell*) Protokoll direkt mit dem entsprechenden Agenten kommunizieren.

### 2.5.3 Knowledge Query and Manipulation Language

Die Agentensprache KQML – *Knowledge Query and Manipulation Language* – wurde im Rahmen des *Knowledge-Sharing-Effort* Projektes an der Universität von Maryland entwickelt [45]. KQML basiert auf sprachtheoretischen Ansätzen und beinhaltet ein Nachrichtenformat und ein Nachrichtenprotokoll. KQML-Nachrichten sind dabei unabhängig vom Inhalt, den sie transportieren. Der Inhalt, meist eine Information in einer *content language*, wird vielmehr in einen Kontext gestellt, um so die Interpretation zu erleichtern.

KQML eignet sich für die Kommunikation zwischen autonomen Agenten. Autonom bedeutet in diesem Zusammenhang, daß Agenten unterschiedliche, sogar konkurrierende Aufträge besitzen können. KQML-Nachrichten zwingen daher den Sender einer Nachricht, genau den Kontext des Inhalts zu spezifizieren. Dies erlaubt dem Empfänger die Wahl einer entsprechend angemessenen Aktion.

Eine KQML-Nachricht wird als *performative* bezeichnet, einem Begriff aus der Sprachtheorie. Ein *performative* wird durch eine Lisp ähnliche ASCII-Zeichenfolge in Polnisch-Umgekehrter-Notation (PPN) angegeben. Diese Notation wurde gewählt, weil sie für Menschen

leicht zu verstehen ist und damit der “Sinn” eines *performatives* fast selbsterklärend ist. Einzelne Ausdrücke in KQML werden als Wertepaar angegeben. Die Syntax eines *performatives* kann in PPN-Notation wie folgt angegeben werden:

```
word | string
<performatives> ::= <word> { :<word> <expression>}*)
<expression> ::= <word> | <quotation> | <string> |
                 (<word> { :<word> <expression>}*)
<quotation> ::= `<expression> | `<comma-expression>
<comma-expression> ::= <expression> > | ,<comma-expression>
```

In [46] werden eine Vielzahl von vordefinierten *performatives* angegeben. Sie wurden definiert, um einen Grundstock an Nachrichten für bestimmte, allgemein notwendige, Aktivitäten einer Software-Agenten-Gemeinde zu schaffen. Ein Beispiel: Gegeben seien die 2 Agenten A1, persönlicher Agent einer Person, und A2, Agent einer Fluggesellschaft. Agent A1 möchte für seinen Anwender den Preis für einen Flug von London nach Berlin herausfinden. Dazu schickt er eine Nachricht an Agent A2:

```
(ASK
:CONTENT (Preis (Flug London Berlin) ?Preis)
:RECEIVER A2
:LANGUAGE prolog
:ONTOLOGY Flugticket)
```

Agent A2 erkennt anhand der Ontologie den Kontext, schaut in seiner Wissensbasis den Preis nach und sendet die Antwort zurück zum Agenten A1:

```
(TELL
:CONTENT (Preis DM 500)
:RECEIVER A1
:LANGUAGE prolog
:ONTOLOGY Flugticket)
```

Die Schlüsselwörter CONTENT, RECEIVER, LANGUAGE, ONTOLOGY und weitere sind in ihrer Bedeutung vordefiniert. Zusammen mit dem Namen eines *performatives*, etwa TELL, ergibt sich eine vom Sinn (aber nicht vom Inhalt) her eindeutig interpretierbare Nachricht. Die entsprechenden Definitionen finden sich in [46]. Ein Vorschlag für eine erweiterte Version ist [45].

#### 2.5.4 Die Agentenplattform ASAP

Als Beispiel für eine Agentenplattform, wird an dieser Stelle ASAP – A Simple Agent Plattform – vorgestellt. ASAP [104] ist eine auf JAVA und KQML aufbauende Umgebung, die vorgefertigte Agenten-Templates anbietet und so die einfache und unkomplizierte Entwicklung von Software-Agenten ermöglicht.

Zur Laufzeit bedienen sich die Agenten der Dienste der ASAP-Umgebung. Ein sogenannter *conditioner* kann von Agenten verwendet werden, um bei Veränderungen der Rechnerumgebung oder beim Eintreten bestimmter Ereignisse informiert zu werden. Ein weiterer

wichtiger Aspekt ist die erweiterte Kommunikationsmöglichkeit: ASAP greift die Idee von *integrated networking* auf und erlaubt Kommunikation über unterschiedliche Netzwerke.

ASAP realisiert das Facilitator Modell Nummer 5 aus Abbildung 14. Der Facilitator ist selbst Agent und Bestandteil der Plattform. Um einen passenden Agenten für eine Anfrage empfehlen zu können (*recommend(ask)*), ist ein spezieller Algorithmus entwickelt worden. Dieser vergleicht die Wertepaare der registrierten *performatives (advertise(ask))* mit dem der Frage und vergibt nach einem bestimmten System Punkte. Der Agent mit der höchsten Punktzahl wird dem Fragesteller empfohlen.

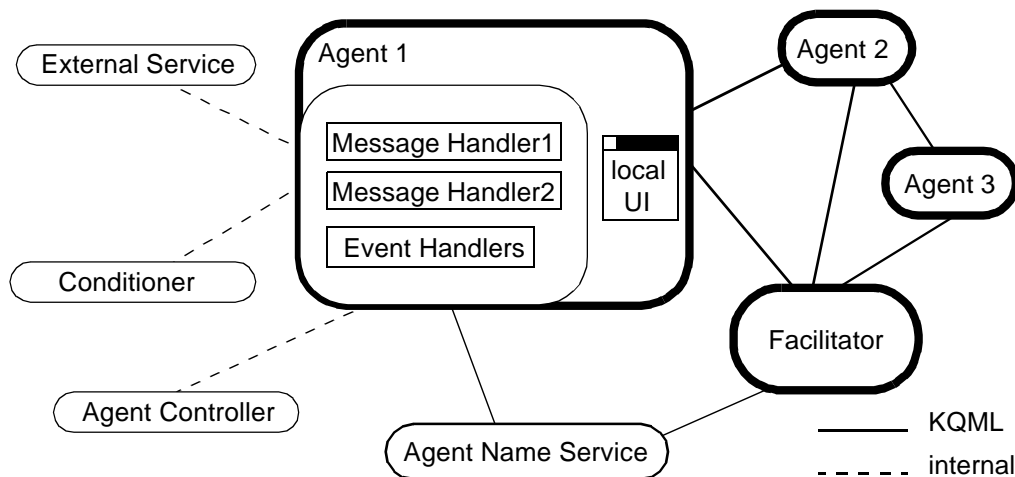


Abbildung 15: Die Architektur von ASAP

Abbildung 15 zeigt die Architektur von ASAP. Neben den bereits erwähnten Komponenten existiert der Dienst *external service*, der eine allgemeine Schnittstelle zu externen Programmen bietet, der *agent controller*, der die Aufgabe eines Aufsehers über die Agenten hat, und der *agent name service*, der für die interne Auflösung von Agentennamen und Rechneradressen zuständig ist.

In [104] werden mit ASAP entwickelte Agenten eingesetzt, um in einer virtuellen Umgebung agieren zu können. Sie repräsentieren dort Objekte einer virtuellen Szene und lösen spezielle Aufgaben:

- Ein Avatar-Agent begleitet eine reelle Person durch die virtuelle Szene und übernimmt die Interaktion mit anderen Agenten.
- Ein Tür-Agent gibt Auskunft über die Möglichkeiten im nächsten Raum, fragt nach dem Zugangscodex und berechnet ein Eintrittsgeld.
- Ein Wand-Agent dient als eine Art Zeitung und gibt Informationen und Neuigkeiten weiter.

Weitere geplante Agenten in der Umgebung sind beispielsweise Wächter, Reporter und Bibliothekare.

Neben ASAP existieren eine Reihe anderer Agentenplattformen. Sie alle wollen die einfache Entwicklung von Agenten – mobil und/oder statisch – ermöglichen. Als Beispiele seien hier JAT [48], die *Aglets Workbench* [85] und *Telescript* [156] genannt.

## 2.6 Transparenz in Groupware

### 2.6.1 Einführung

In der Literatur, wie etwa in [63], [65] oder [66], finden sich theoretische Ansätze aus dem Bereich der Gestaltungsanforderungen zu ergonomischer Software, die sich auch mit dem Thema Transparenz<sup>1</sup> beschäftigen. Der Begriff Transparenz wird dort verfeinert zu

- funktionaler Transparenz
- nutzungsbezogener Transparenz.

Funktionale Transparenz verdeutlicht die Wirkungsweise einer Funktion. Sie macht deren „tun“ für den Anwender nachvollziehbar. Realisiert wird funktionale Transparenz im allgemeinen durch sogenannte „Hilfefunktionen“, die die Funktionalität verständlich machen.

Für diese Arbeit ist die zweite Art der Transparenz, die nutzungsbezogene Transparenz, wichtig. Nutzungsbezogene Transparenz macht den Nutzen einer Funktion nachvollziehbar und verdeutlicht dem Anwender die Vorgänge im System. Dies bezieht sich sowohl auf den Zustand des Systems selbst, als auch auf den Zustand einer kooperativ (synchron wie auch asynchron) zu erledigenden Aufgabe. „Dabei kompensiert nutzungsbezogene Transparenz den Mangel an Erwartungskonformität bezüglich des aktuellen Systemzustandes, der durch die gegenseitige Beeinflussung der verschiedenen Groupwarenutzer bedingt ist ([63] S. 131)“. Ein Beispiel: Der Initiator einer Telekonferenz kann nachvollziehen, wenn sein Kommunikationsversuch scheitert, weil der Partner zur Zeit nicht gestört werden will. Das transparente System stellt die dazu notwendigen Informationen zur Verfügung.

Grundsätzlich kann unterschieden werden, ob sich Transparenz auf eine gerade aktiv stattfindende kooperative Sitzung oder auf passive Daten und Informationen über eine Gruppe oder eine gemeinsame Aufgabe bezieht. Vertegaal [151] spricht hier von *micro-level* und *macro-level awareness*<sup>2</sup>. Gaver [51] unterscheidet nach *Focused Collaboration* und *Serendipitous*<sup>3</sup> *Communication* und faßt beide durch *general awareness* zusammen.

- 
1. Transparenz, wie sie etwa bei einem transparenten Zugriff auf eine verteilte Datenbank verstanden wird, ist an dieser Stelle nicht gemeint. Im Gegensatz zu dieser Art von „Transparenz“, die etwas verbirgt, also „unsichtbar“ macht, ist es das Anliegen dieser Arbeit, dem Anwender oder Benutzer etwas zu verdeutlichen, also „sichtbar“ zu machen.
  2. Im Englischen wird korrespondierend zu nutzungsbezogener Transparenz (im folgenden nur „Transparenz“) der Begriff *awareness* verwendet.
  3. *Serendipitous*: hier „zufällig“, „nicht geplant“.



### 2.6.2 Mikro-Level-Transparenz

Mikro-Level-Transparenz bezieht sich auf eine auf den Arbeitsbereich beschränkte Tätigkeit, in der eine Gruppe von Personen aktiv an einem gemeinsamen Ziel arbeitet. Gutwin und Greenberg [62] führen dazu den Begriff *workspace awareness* ein. Unter *workspace* versteht man einen gemeinsamen Arbeitsbereich, zu dem die Mitglieder einer kollaborierenden Gruppe gleichsam Zugriff haben. Ein solcher Arbeitsbereich beinhaltet neben den zu bearbeitenden Dokumenten Werkzeuge, die zum einen die Bearbeitung der Dokumente zum anderen die Kooperation zwischen den Arbeitsgruppenmitgliedern ermöglichen. Hierunter fallen auf der einen Seite Textverarbeitungs- und Tabellenkalkulations-Systeme auf der anderen Seite Workflow-Systeme, *whiteboards*, *application sharing* Systeme, E-mail und Telekonferenz-Systeme (siehe dazu auch Abschnitte 2.1 bis 2.4).

Die Fragen, mit denen sich Greenberg und Gutwin beschäftigen betreffen im wesentlichen die Problematik *Was* passiert bzw. *Was* wird aktuell im Gruppenraum getan und *Wo* wird es durchgeführt. Die folgende Tabelle 1 faßt das konzeptuelle Rahmenwerk von Greenberg und Gutwin zusammen:

Element	Fragestellung
Präsenz	Wer arbeitet im Arbeitsbereich?
Lokation	Wo wird gearbeitet?
Aktionslevel	Wie aktiv sind die Teilnehmer?
Aktionen	Was wird getan? Was sind die aktuellen Aktivitäten und Aufgaben?
Intentionen	Was wird als nächstes getan? Wo wird dies stattfinden?
Änderungen	Welche Veränderungen finden wo statt?
Objekte	Welche Objekte werden verwendet?
Bereich	Wie weit kann ein Teilnehmer sehen? Wie weit reicht der Zugriff eines Teilnehmers?
Fähigkeiten	Was kann ein Teilnehmer tun?
Einfluß	Wo kann ein Teilnehmer Änderungen vornehmen?
Erwartungen	Was wird ein Teilnehmer als nächstes tun?

**Tabelle 1: Elemente der workspace awareness nach [62]**

Vertegaal [151] ergänzt das Rahmenwerk um einen Aufmerksamkeitsfaktor und rückt damit das aktuell bearbeitete Objekt (Objekt im weitesten Sinne) mehr in den Fokus. In die-

sem Zusammenhang wird auch vom *Point of Interest* (POI) [112] oder der synchronen Interaktion mit dem Fokus des Interesses gesprochen.

### 2.6.3 Makro-Level-Transparenz

Im Gegensatz zu Mikro-Level- beschreibt Makro-Level Transparenz oder *informal awareness* [62] eine Form der Transparenz, in der Meta-Informationen über das Team oder eine gemeinsam zu erledigende Aufgabe dem Anwender außerhalb einer Sitzung sichtbar gemacht wird. Es sollen Fragen beantwortet werden wie: Welcher Kooperationspartner ist für eine Kommunikation bereit? Womit ist er beschäftigt? Welchen Status hat ein Dokument? Wo befindet sich ein Dokument?

Es kann damit nach objekt- und personenbezogener Transparenz unterschieden werden. Ein Beispiel für eine objektbezogenes System ist der Ereignisdienst des POLIKOM Projektes [49][50]. Der Ereignisdienst erlaubt eine automatische Benachrichtigung bei bestimmten Ereignissen, etwa bei zeitversetzter Bearbeitung von Objekten. Der Schwerpunkt liegt dabei auf der Nachvollziehbarkeit von Ereignissen und der Möglichkeit zum Wiedereinstieg, etwa nach Abwesenheit. Es wird in diesem Zusammenhang auch von asynchroner Gruppenwahrnehmung gesprochen. Zur Wahrung der Vertraulichkeit von Ereignissen kann ein Benutzer Ereignisse, die er selbst auslöst, mit einer Vertraulichkeitsintensität belegen. Hierdurch wird festgelegt wie unzugänglich Ereignisse gegenüber anderen sind. Eine Ereignishistorie speichert alle Ereignisse permanent und erlaubt so, bereits vergangene Aktivitäten nachzuvollziehen und somit einen schnelleren Wiedereinstieg in das Geschehen.

Workflow-Management Systeme erlauben ebenfalls Transparenz auf Objektebene. Die Historie eines Vorgangs wird anhand der bereits ausgeführten Aktionen angezeigt. Diese Information wird zumeist angereichert mit einem Zeitstempel und einer Personenangabe. Der Anwender ist hierdurch in der Lage herauszufinden, wann und von wem eine Aktion durchgeführt worden ist. Zusätzlich wird oftmals die Versionierung von Dokumenten in Workflow- und Dokumentenmanagement-Systeme unterstützt. Forschungsprototypen, wie etwa das System DIVA [136], gehen einen Schritt weiter und erlauben beispielsweise anhand eines Versionsgraphen, die genaue Historie eines Dokumentes nachzuvollziehen.

Unter UNIX-Betriebssystemen existieren standardgemäß einige Funktionen, die die aktuellen Benutzer eines Rechners oder eines Rechnerverbundes anzeigen und damit Makro-Level-Transparenz im personenbezogenen Bereich schaffen. Der Befehl *finger* gibt detaillierte Informationen über einen Benutzer bzw. alle Benutzer eines Rechners aus. Dazu gehören:

- die Benutzererkennung
- der Name des Benutzers
- das letzte Login-Datum inklusive Uhrzeit
- die *Idle Time*, die angibt wie lange ein Benutzer bestimmte Kernprozesse (*shell*) nicht mehr verwendet hat
- ob ungelesene Mail existiert und wann der Benutzer das letzte mal Mail gelesen hat

Diese Funktionalität machen sich einige Systeme zu Nutze, um *Macro-Level awareness* im personenbezogenen Bereich zu erzeugen. Das Programm Xhtalk beispielsweise schaut regelmäßig nach, ob bestimmte Personen an ihren Arbeitsplatzrechnern „eingeloggt“ sind. Diese Personen werden dann im Xhtalk-Fenster mit ihrer *Idle-Time* angezeigt. Der Informationsgehalt der Benutzungsoberfläche von Xhtalk (siehe Abbildung 16) wird durch die Verwendung von Farben erhöht. Personen mit hoher *Idle-Zeit* werden in einer anderen Farbe dargestellt als Personen mit niedriger *Idle-Zeit*. Für jeden Benutzer, der in dem Xhtalk-Fenster erscheint, kann ein Informationsprofil abgerufen werden. Diese Informationen werden vom *finger*-Befehl bezogen. Mit der rechten Maustaste kann der Benutzer einen Dialog mit der entsprechenden Person auf Textbasis (*talk*) öffnen.

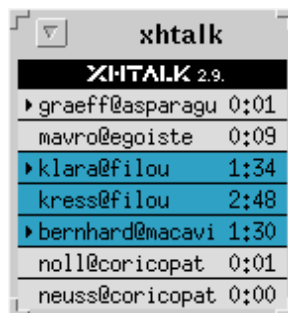


Abbildung 16: Benutzungsoberfläche von Xhtalk

Die Personen, die „beobachtet“ werden sollen und die Rechner, auf denen Xhtalk nachschaut, sind in einer Konfigurationsdatei gespeichert. Anwender können sich nur vor einer solchen „Beobachtung“ schützen, indem sie den *finger-port* vom Systemadministrator auf ihrer Maschine (und damit für alle Benutzer der Maschine) schließen lassen.

Den gleichen Mechanismus verwendet Greenberg in seinem System *Peepholes* [57]. Anstatt einer einfachen Anzeige werden hier graphische Symbole verwendet. Außerdem kann ein Adreßbuch mit detaillierten Informationen zur Person (Abteilung, Adresse, Telefonnummer, E-Mail) abgerufen werden.

Andere Systeme setzen Videokameras ein, um spezifische Informationen über mögliche Kooperationspartner zu erhalten. Mit *Portholes* [36] stellten Dourish und Bly eine der ersten verteilten videobasierten Macro-Level awareness Umgebung vor. In den beiden Xerox Laboratorien in Palo Alto (PARC) bzw. Cambridge (EuroPARC) wurden in verschiedenen Büro- und Aufenthaltsräumen Kameras aufgestellt. Von einem Server konnten Schnappschüsse, die alle ein bis zwei Minuten aktualisiert wurden, abgefragt werden. Die Wissenschaftler verwendeten Portholes einerseits um herauszufinden, ob eine bestimmte Person im Moment erreichbar ist. Andererseits gab das System den Anwendern ein Gefühl der „Nähe“.

## 2.7 Zusammenfassung

Die Entwicklung einer kooperativen Verwaltungsumgebung bedarf der intensiven Unterstützung durch computerbasierte Systeme. Insbesondere CSCW und Workflow Techniken eignen sich für eine zukünftige Verwaltungsumgebung und bieten die dazu notwendigen Voraussetzungen. In diesem Kapitel ist daher ein Überblick der für diese Arbeit relevanten Techniken und Ansätze aus diesem Bereich gegeben worden. Es ist eine Einführung in CSCW und Groupware im allgemeinen und in die Bereiche Telekooperation, Konferenzumgebungen und computergestützte Vorgangsbearbeitung im speziellen gegeben worden. Für die synchron verteilte Kooperation in der Verwaltung eignen sich insbesondere

- Konferenzumgebungen
- Videokonferenzsysteme
- Whiteboards
- Sharing Komponenten

Die asynchrone Kooperation wird durch

- E-Mail und
- Workflow-Systeme unterstützt.

Konferenzumgebungen verwalten Telekonferenzsysteme und integrieren dazu Audio/Videokomponenten, Whiteboards und/oder Sharing Komponenten. Der Standard T.120 der international Telecommunication Union (ITU) bietet diese Möglichkeit und vereinheitlicht Schnittstellen, um eine einheitliche Kooperationsbasis zu bilden. Audio/Video-Konferenzsystemkomponenten ermöglichen dabei die Durchführung einer Besprechung, ohne die örtliche Präsenz aller Teilnehmer vorauszusetzen. Whiteboards unterstützen eine verteilte Diskussion durch die Möglichkeit, skizzenhafte Anmerkungen auf ein in den Hintergrund des Whiteboards geladenes Rasterbild, etwa eine Seite eines Textdokumentes oder eine CAD-Zeichnung, machen zu können. Sharing Komponenten verteilen die Ausgabe einer Standard-Applikation und erlauben so das gemeinsame Arbeiten. Ein Floor-Control-System stellt sicher, daß Eingaben nur von einem Teilnehmer zu einer Zeit an die Standard-Applikation weitergeleitet werden kann.

Workflow-Systeme automatisieren Büroprozesse. Dies beinhaltet die Festlegung, Überwachung und zeitliche Steuerung einzelner Teilaktivitäten eines Geschäftsprozesses sowie die Bereitstellung und den Austausch dabei nötiger Dokumente. Die Workflow Management Coalition (WfMC) definiert derzeit ein Workflow-Referenzmodell auf dem Weg zu einem Standard. Neben einem generellen Architekturrahmenwerk hat sie fünf wesentliche Schnittstellen an der verarbeitenden Einheit eines Workflow-Systems, der Workflow-Engine spezifiziert.

Problematisch an heutigen CSCW-Systemen ist, daß durch die neuen kooperativen Möglichkeiten auch neue gruppensdynamische Aspekte entstehen, die in den Entwicklungen nur selten betrachtet werden. Den Anwendern muß die Bedienung der komplexen Systeme so einfach wie möglich gemacht werden, da sie von der eigentlichen Aufgabe, der Durchfüh-

zung eines bestimmten Sachverhalts, nicht ablenken dürfen. Für die synchrone Interaktion bedeutet dies weiterhin, daß der Fokus des Interesses (FDI), für alle Partner gleichsam zugänglich sein muß. Neben der Möglichkeit den FDI zu sehen, ist es wichtig, Mechanismen zur Interaktion anzubieten. Hier unterstützen die Technologien Whiteboard und Application Sharing jeweils nur zum Teil.

Weiterer wichtiger Punkt ist die Schaffung von Transparenz im neu entstandenen Gruppenraum, um den Mangel an Erwartungskonformität, der durch die gegenseitige Beeinflussung der verschiedenen Nutzer bedingt ist, zu kompensieren [43]. Ein weiterer Abschnitt dieses Kapitels beschäftigte sich deshalb mit dem Thema nutzungsbezogener Transparenz. Die dort vorgestellten Konzepte lassen sich in Makro-Level- und Mikro-Level-Transparenz unterteilen. Realisiert sind bisher allerdings nur videobasierte Systeme. Diese besitzen den entscheidenden Nachteil, daß sie auch zur Überwachung eingesetzt werden können. Dadurch sind sie im Umfeld von Verwaltungen nicht einsetzbar.

In der Einleitung wurde bereits auf einen Ansatz zur Lösung der dargestellten Probleme, der sich durch die Verwendung von Software Agenten auszeichnet, vorausgegriffen. In Abschnitt 2.5 wurde daher auf diesen Technologiebereich gesondert eingegangen. Software Agenten operieren ohne die direkte Intervention von Menschen oder anderen Entitäten und haben dabei ein gewisses Maß an Kontrolle über ihre Aktivitäten und ihren Status. Dabei interagieren sie mit anderen Agenten und/oder Menschen. Sie nehmen ihre Umwelt wahr und reagieren in einer zielgerichteten Art auf Veränderungen in ihr.



## **Kapitel 3**

# **Transparenz in kooperativen Verwaltungsumgebungen**

## **3.1 Motivation**

### **3.1.1 Software-Ergonomie als treibende Kraft**

Die Bedeutung von Softwareergonomie für die Gestaltung und Qualität von Arbeitsplätzen nimmt zu, da Computerunterstützung in Arbeitsbereichen Einzug hält, die bisher nicht davon betroffen waren. Kooperation und Kommunikation wurde bisher durch Konferenzen und informelle Diskussionen, durch Telefonate, Briefe und Fax ermöglicht. In den Bereichen (Tele-) Kommunikation und Kooperation steigt der Einfluß von Rechnertechnologie in entscheidendem Maße. Gleichmaßen kann sich der Technologieeinsatz auch als Hürde für die Arbeitsgestaltung erweisen. Dies trifft insbesondere dann zu, wenn wirtschaftliche Faktoren den Einsatz von Computertechnologie erzwingen, ohne daß die entsprechende Software anwendergerecht konzipiert wurde. Aspekte anwendergerechter Softwaregestaltung werden vom Markt dann vernachlässigt, wenn sie nicht für den Einzug der Technologie in die Unternehmen erforderlich sind. Diese Gefahr besteht auch für die Technologien zur computer-unterstützten Gruppenarbeit. Sie können zu einer Verschlechterung von Arbeitsbedingungen und zum Hemmnis für die Einführung innovativer Organisationsstrukturen führen, wenn sich ihre Konzeption stärker an den technischen Möglichkeiten als an der Unterstützung des Menschen in seiner Arbeitssituation orientiert.

Jedes Anwendungs-System, gleich welcher Art (Desktop-Publishing-System, Datenbank-System, Textverarbeitungs-System, CAD-System, usw.), repräsentiert sich dem Benutzer durch seine Benutzungsoberfläche. Als Schnittstelle zwischen dem arbeitenden Menschen und dem für die Arbeit benutzten System bestimmt sie nicht nur die Akzeptanz des Systems und die Effizienz seines Einsatzes, sondern auch die menschengerechte Gestaltung der Arbeit beim Umgang mit dem Rechner [47]. Über das tatsächliche Anwendungspotential eines Systems wird insbesondere dessen software-ergonomische Gestaltung entscheiden. Dies gilt im besonderen Maße für komplexe CSCW Umgebungen, deren Handhabung nicht von der eigentlichen Aufgabe, etwa einer kooperativen Vorgangsbearbeitung, ablenken soll.

In einer kooperativen Umgebung ergibt sich die Notwendigkeit, das mentale Modell des einzelnen Benutzers auf den Gruppenraum zu vergrößern. Die Definition von Thimbley

[144] zur Benutzungsschnittstelle: „Eine Benutzungsoberfläche ist ein Informationskanal, der Informationen zwischen dem Benutzer und dem Computer transportiert“ kann erweitert werden zu „Eine Benutzungsoberfläche ist ein Informationskanal, der Informationen zwischen dem Benutzer und dem Computer und zwischen Mitgliedern einer kollaborierenden Gruppe über den Computer transportiert“. Transparenz macht dabei das Handeln einzelner Nutzer für andere Kooperationspartner sichtbar. Dies bezieht sich sowohl auf den Zustand des Systems selbst, als auch auf den Zustand einer kooperativ (synchron wie auch asynchron) zu erledigenden Aufgabe.

### 3.1.2 Einsatzgebiete

Um ein besseres Bild über die potentiellen Einsatzgebiete von Transparenz für das Referenzmodell zu bekommen, soll ein Szenario aus der Studie zum Informationsverbund Berlin – Bonn (IVBB) [28] näher betrachtet werden. Die IVBB gibt in ihrem Kapitel 3 einen Überblick der Aktivitäten im Informationsverbund Berlin – Bonn. Im einzelnen sind dort die folgenden Aktivitäten beschrieben:

- Eingang erhalten
- Beschaffung von Informationen
- Abstimmung in kleinen Gruppen
- Abstimmung in großen Gruppen
- Vortrag/Rücksprache
- Stellungnahme anfordern
- Zeichnung einholen
- Arbeitsergebnisse mitteilen
- Verteilung von Informationen
- Terminabsprache
- Ressourcenreservierung

Die Aktivitäten werden in der Studie unter anderem bezüglich ihrer funktionalen Anforderungen beleuchtet. Für die hier vorgestellte Arbeit sind dabei vor allem die Anforderungen wichtig, die einen Bedarf an nutzungsbezogener Transparenz im Groupware-Bereich haben. Interessant sind demnach Aktivitäten, die Bereiche wie etwa synchrone und asynchrone Telekooperation, kooperative Dokumentenbearbeitung oder Informationsbeschaffung berühren.

Das in der IVBB Studie verwendete Szenario soll hier nun in stark gekürzter Form aufgegriffen werden (siehe [28] S. 3 - 6ff). Die einzelnen extrahierten Aktivitäten werden durch Transparenz betreffende Anmerkungen angereichert. Dabei wird auch auf die funktionalen Anforderungen eingegangen.



<b>Arbeitsaktivität und Bezug zu Transparenz</b>	<b>Kommunikationsrelevante Aktivität</b>
Herr R. betritt sein Büro und sieht als erstes seine Eingänge durch. Anschließend wirft er einen Blick auf seinen elektronischen Terminkalender. Verschiedene Eingänge verlangen neben der normalen Registrierung (elektronisch) eine Quittung. Der Empfänger macht transparent, daß er den Eingang erhalten hat.	Eingang erhalten
Heute vormittag steht eine Besprechung innerhalb des Ministeriums an. Die Abstimmung des Termins erfolgte elektronisch. Die Teilnehmer und der Zeitraum für den Termin wurden von Herrn R. festgelegt. Da es sich um einen dringenden Termin handelt, machte er eine entsprechende Markierung. Der Termin wurde vom System automatisch unter Berücksichtigung der Wünsche der Teilnehmer bestimmt, ohne daß eine Person Einblick in die Terminkalender der anderen Sitzungsteilnehmer bekommt.	Terminabstimmung
Für diese Besprechung benötigt Herr R. noch eine Akte. Im Vorgangsbearbeitungs-System steht ihm eine Funktion zum Suchen nach Akten zur Verfügung.	Beschaffung von Informationen
Die Akte verteilt Herr R. an die Teilnehmer der Besprechung. Die Verteilung erfolgt elektronisch. Herr S. kann heute nicht persönlich an der Besprechung teilnehmen. Er wird sich per Telekonferenz zuschalten. Herr R. benutzt die elektronische Post zur Verteilung der Akte. Da er häufig mit den Teilnehmer der heutigen Besprechung zu tun hat, hat er sich eine entsprechende Verteilerliste angefertigt. Für globale Informationen stehen ebenfalls solche Verteilerlisten in den Sekretariaten zur Verfügung. Herr R. ist in der Lage, sich selbstständig in solche Listen ein- und auch auszutragen.	Verteilen von Informationen
Herr R. reserviert einen Raum mit einem Videokonferenz-System für die Besprechung. Die elektronische Ressourcenreservierung nimmt ihm diese Arbeit ab.	Ressourcenreservierung
Die Besprechung findet statt. Herr S. ist über das Videokonferenz-System angebunden. Herr S. sitzt an seinem persönlichen Arbeitsplatz, welcher mit einem Videokonferenz-System ausgestattet ist. Er wird elektronisch eingeladen, an der Konferenz teilzunehmen. In der Einladung wird ihm das Thema der Sitzung und die Teilnehmer bekannt gegeben. Zur Bearbeitung der Akte wird diese durch ein <i>application sharing</i> System allen Teilnehmer sichtbar gemacht.	Abstimmung in kleinen Gruppen

Tabelle 2: Transparenz im Verwaltungsalltag - Ein Szenario

<b>Arbeitsaktivität und Bezug zu Transparenz</b>	<b>Kommunikationsrelevante Aktivität</b>
Nach der kleinen Besprechung nimmt Herr R. an einer größeren Ressortsitzung teil. Videokonferenzen für große Gruppen werden zusätzlich zu persönlichen Büro-Video-Arbeitsplätzen durch die Liveboard Technologie unterstützt.	Abstimmung in großen Gruppen
Herr R. kehrt zurück zu seinem Arbeitsplatz und findet eine neue Nachricht mit der Bitte um Rücksprache in seinem elektronischen Briefkasten vor. Es geht um eine Anfrage aus dem Deutschen Bundestag. Herr R. legt sich eine neue Akte in der elektronischen Schriftgutverwaltung an. Der Verbleib der Akte wird mit „bei Herrn R.“ gekennzeichnet. Diese Information ist über die elektronische Schriftgutverwaltung allgemein zugänglich.	Eingang erhalten
Herr R. trifft sich zu einem persönlichen Gespräch, nach vorheriger Terminabsprache im Sekretariat, mit seinem Abteilungsleiter.	Terminabstimmung Vortrag/Rücksprache
Zur Beantwortung der Anfrage benötigt Herr R. einige fachliche Stellungnahmen. An der elektronisch übermittelten Unterlage können die Sachbearbeiter in textueller, graphischer oder sprachlicher Form Anmerkungen anbringen. Der Verbleib der Unterlage wird durch den Vorgang „Eingang erhalten“ beim Sachbearbeiter jeweils dem Vorgangsverfolgungs-System mitgeteilt. Herr R. kann sich erkundigen, wer gerade im Besitz der Unterlage ist. Herr R. hat der Unterlage die Priorität „hoch“ zugeteilt. Das Vorgangsverfolgungs-System weist den Sachbearbeiter auf die Dringlichkeit hin.	Stellungnahme anfordern
Herr R. erinnert sich an eine Veröffentlichung in den „elektronische Neuigkeiten“. Er recherchiert in der Datenbank für amtliche Veröffentlichungen. Dem System ist die Benutzerkennung des Herrn R. bekannt, der sich beim Anmeldevorgang verifiziert hat. Das System gibt Herrn R. Zugang zu bestimmten Datenbankressourcen. Benötigt er Zugang zu weiteren Dokumenten, kann er einen entsprechenden Antrag an die Administration stellen. Herr R. stellt weiterhin fest, daß eine bestimmte von ihm noch benötigte Akte bei Herrn K. weilt. Er erkennt, daß die Akte fertig bearbeitet aber noch nicht freigegeben ist. Mittels der elektronischen Post schickt er an Herrn K. eine Anfrage, die Akte für ihn frei zu stellen.	Beschaffung von Informationen
Herr R. erledigt weitere Vorgänge und verteilt die Ergebnisse über die elektronische Schriftgutverwaltung. Die elektronische Schriftgutverwaltung schickt nach einer kurzen Weile Herr R. eine Bestätigung über das Eingehen der Ergebnisunterlagen beim Adressaten.	Arbeitsergebnis mitteilen
Herr R. beendet gegen 17.00 Uhr seinen Arbeitstag.	

Tabelle 2: Transparenz im Verwaltungsalltag - Ein Szenario

### 3.1.3 Transparenzbezug

Betrachtet man den beispielhaften Arbeitstag des Herrn R., so kann man die folgenden nutzungsbezogenen Transparenzdatensätze extrahieren:

- Telekonferenz
  - Kommunikationsbereitschaft (Telefonmetapher: Besetzt, kein Anschluß, usw.)
  - Kommunikationsmöglichkeiten (Partner, Werkzeuge, Netz, usw.)
- Terminplanung
- Verbleib von Akten, Umläufen, Unterlagen
- Zugriffsrechte auf Akten, Umläufe, Unterlagen
- Zustand von Akten, Umläufen, Unterlagen
- Zugriffsrechte auf Ressourcen

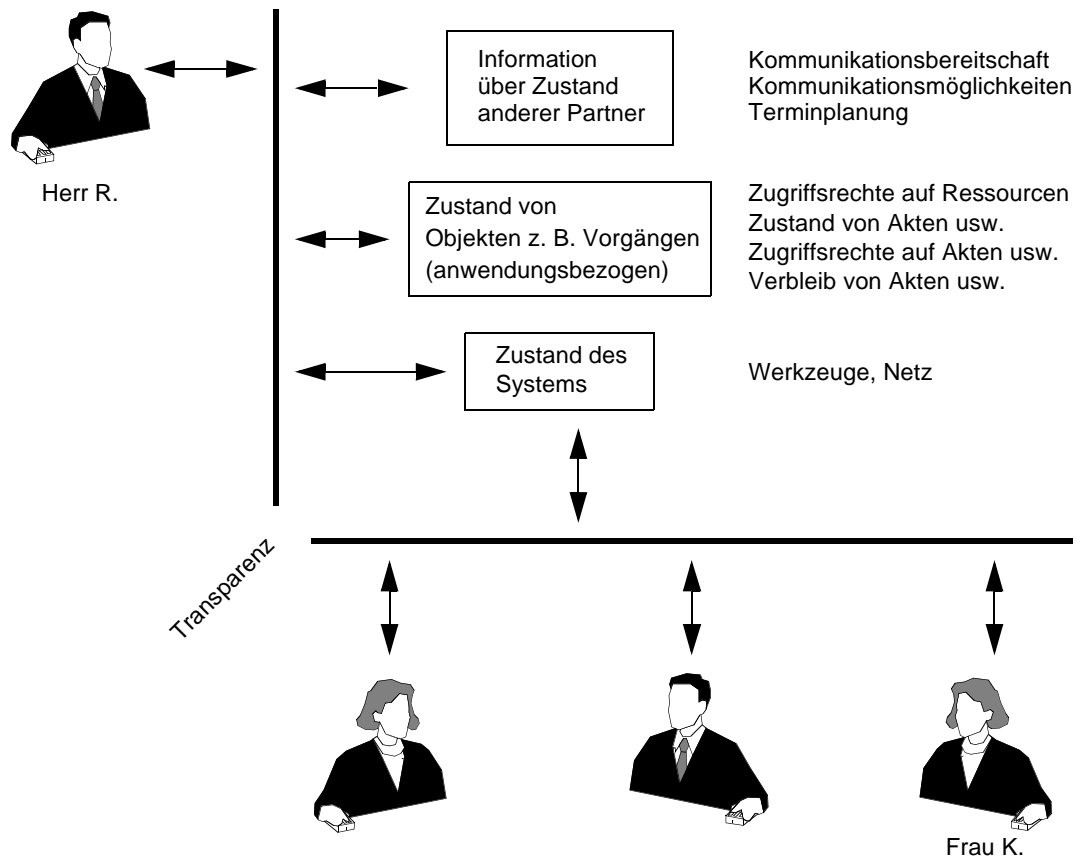


Abbildung 17: Nutzungsbezogene Transparenz

Abbildung 17 verdeutlicht diesen Transparenzbezug graphisch. Diese dargestellten Beziehungen sind grundsätzlich erst einmal wechselseitig zu verstehen. Herr R. kann etwa genauso feststellen, daß Frau K. im Moment nicht durch eine Telekonferenz gestört werden will, wie auch Frau K. Einblick in den „Telekonferenz-Status“ bei Herrn R. hat.

## 3.2 Analyse der Benutzeranforderungen

### 3.2.1 Einstieg

Um sich ein Bild der Anforderungen an eine nutzungsbezogene transparente Büroumgebung zu machen, ist eine Analyse aus Benutzersicht durchgeführt worden. Ausgangsbasis waren neben [28], [63], [65], [66], [130] und [150] die Anwendungsszenarien und Arbeits- und kognitionspsychologischen Kriterien aus dem DeTeBerkom Projekt “Mensch-Maschine-Schnittstellen im Multimedialen Verwaltungsbüro” [10].

Um einen Einstieg in die vielfältigen Möglichkeiten zur Untersuchung bezüglich Transparenz zu finden, soll zu Beginn der Analyse ein Blick in die IVBB Studie [28] geworfen werden. Dort findet sich auf Seite 3-33 die folgende Tabelle 3. Sie beschreibt eine Zuordnung der obengenannten Basisaktivitäten zu den heute gebräuchlichen Kommunikationsvorgängen.

	<b>persönlich besprechen</b>	<b>telefonieren</b>	<b>Dokumenten- austausch</b>
Eingang erhalten			X
Beschaffung von Informationen	X	X	X
Abstimmung in kleinen Gruppen	X	X	X
Abstimmung in großen Gruppen	X		X
Vortrag/ Rücksprache	X	X	
Stellungnahme anfordern			X
Arbeitsergebnisse mitteilen	X	X	X
Verteilung von Informationen		X	X
Terminabsprache	X	X	X
Ressourcen- reservierung	X	X	X
Zeichnung einholen			X

Tabelle 3: Zuordnung von Basisaktivitäten zu Kommunikationsvorgängen (nach [28])

### 3.2.2 Szenario aus dem Büroalltag

Um eine Analyse vornehmen zu können, inwieweit Transparenz für die einzelnen Aktivitäten und damit insgesamt für den Büroalltag relevant ist, ist eine Abschätzung der Häufigkeit der Durchführung einer Aktivität notwendig. Es werden dazu zwei Quellen herangezogen. Zum einen die Studie IVBB und zum anderen die diesbezüglichen Ergebnisse des Projektes „Mensch-Maschine-Schnittstelle im multimedialen Verwaltungsbüro“ (MMI) [10]. Referenzszenario des MMI-Projektes ist die Bearbeitung eines Drittmittelantrages der TU Berlin. Diese wird im Referat IIC durchgeführt. Zu den Basisaktivitäten der IVBB ist hier die Aktivität „Dokumentenbearbeitung“ eingeführt worden. Das Szenario ist im folgenden aufgeführt:

Beteiligte Personen:

Herr A. – Antragsteller eines Forschungsprojektes

Sachbearbeiter und Vorgesetzter der Abteilung IIC der TU-Verwaltung

Beschreibung des Ablaufs	Klassifikation
Herr A. schickt das Formular 1502 für eine Forschungsanzeige ausgefüllt an die Abteilung IIC.	
Ein Sachbearbeiter registriert den Eingang und prüft das Formular 1502 am Bildschirm auf formale und inhaltliche Korrektheit und markiert verbesserungswürdige und unklare Stellen im Dokument. Er holt den Standardbrief für Stellungnahmen zu einem Drittmittelantrag auf den Bildschirm und füllt ihn aus. Der Sachbearbeiter leitet nach Fertigstellung die Akte bestehend aus Antrag und Stellungnahme an den Vorgesetzten weiter.	Eingang erhalten/ Dokumentenbearbeitung  Dokumentenbearbeitung Stellungnahme anfordern
Der Vorgesetzte öffnet die Akte und überprüft Antrag und Stellungnahme und stellt Unklarheiten über die angegebene Kalkulation "Sachmittel" und "Raumbedarf" im Formular 1502 fest.	Dokumentenbearbeitung
Der Vorgesetzte verlangt eine Rücksprache mit dem zuständigen Sachbearbeiter.	Terminabsprache
Weil das System dies derzeit für die beste Kommunikationsmöglichkeit hält, wird eine Videokonferenz zwecks Rücksprache und Klärung der offenen Fragen gestartet. Das Formular 1502 von Herrn A. wird verteilt dargestellt. Der Vorgesetzte markiert die fraglichen Stellen.	Rücksprache  Abstimmung in kleinen Gruppen

Tabelle 4: Bearbeitung eines Drittmittelantrages

Beschreibung des Ablaufs	Klassifikation
Die Videokonferenz wird beendet. Die Akte wird mit kommentiertem Antrag und Stellungnahme zurück an den Sachbearbeiter geleitet.	Arbeitsergebnis mitteilen
Der Sachbearbeiter bearbeitet die Akte weiter.	Dokumentenbearbeitung

**Tabelle 4: Bearbeitung eines Drittmittelantrages**

Anhand einer Befragung der Benutzer zur Häufigkeit der Durchführung einer bestimmten Aktivität ist diesbezüglich eine Abschätzung vorgenommen worden. Tabelle 5 zeigt diese getrennt nach IVBB und Referat III C auf. Die Summe ist größer als 100%, da Aktivitäten teilweise parallel bearbeitet werden.

	IVBB	Referat III C
Eingang erhalten	< 70% <sup>a</sup>	< 40%
Beschaffung von Informationen	< 40%	< 40%
Abstimmung in kleinen Gruppen	< 40%	> 70%
Abstimmung in großen Gruppen	< 40%	< 40%
Vortrag/Rücksprache	< 40%	< 40%
Stellungnahme anfordern	< 70%	< 70%
Arbeitsergebnisse mitteilen	< 70%	> 70%
Verteilung von Informationen	< 40%	> 70%
Terminabsprache	< 70%	< 40%
Ressourcenreservierung	< 40%	< 10%
Dokumentenbearbeitung		> 70%
Zeichnung einholen	< 70%	< 40%

**Tabelle 5: Häufigkeiten der Basisaktivitäten (nach [28] und [10])**

a. Häufigkeit einer Tätigkeit: selten = bis 10%, gelegentlich = bis 40%, häufig = bis 70%, Regelfall = größer 70%

### 3.2.3 Analyse

Die beiden Tabellen und die Aufzählung der Anforderungen aus Abschnitt 2.6 sollen nun in Zusammenhang gesetzt werden. Ziel ist eine Abschätzung der Relevanz von Transparenz für das Verwaltungsbüro.

Im Abschnitt 2.6 wurden anhand des beispielhaften Arbeitstages des Herrn R. folgende Anforderungen an Transparenz gestellt:

- Transparenz über Kommunikationsbereitschaft und Kommunikationsmöglichkeiten
- Transparenz bei der Terminplanung
- Transparenz bei Zugriffsrechten auf Ressourcen
- Transparenz über Verbleib und Status von Objekten (z. B. Akten, Umläufe, Unterlagen)

Betrachtet man die Definition von Transparenz aus Abschnitt 2.6 ist es sinnvoll, eine Trennung nach Makro- bzw. Mikro-Level-Transparenz durchzuführen. Makro-Level-Transparenz unterstützt asynchrone Tätigkeiten, die von einzelnen Gruppenmitgliedern getrennt von einander durchgeführt werden. Mikro-Level-Transparenz unterstützt dagegen gemeinsame synchrone Tätigkeiten innerhalb einer Gruppe. Daraus kann geschlossen werden, dass Mikro-Level-Transparenz nur dann innerhalb einer IVBB-Aktivität von Nöten ist, wenn hier eine synchrone Kommunikation stattfindet. Betrachtet man Tabelle 3, so werden hier die synchronen Kommunikationstätigkeiten „Persönlich besprechen“ und „telefonieren“ verwendet. Außerdem kann allgemein angenommen werden, daß die Einleitung einer gemeinsamen synchronen Tätigkeit durch eine oder mehrere asynchrone Tätigkeiten vorgenommen wird.

Für Tabelle 6, die das Ergebnis der Analyse der Benutzeranforderungen widerspiegelt, können daher die folgenden Annahmen getroffen werden:

- Ausgangspunkt sind die Basisaktivitäten von IVBB [28] und MMI [10].
- Transparenzdatensätze sind jeweils für Kommunikations- und Objektstatus sowie für Termin- und Ressourcenplanung einer Aktivität notwendig.
- Mikro-Level-Transparenz kann nicht ohne Makro-Level-Transparenz auftauchen, da einer synchronen Tätigkeit immer eine asynchrone vorausgeht.
- Mikro-Level-Transparenz ist nur dann notwendig, wenn synchrone Kommunikationstätigkeiten innerhalb einer Aktivität stattfinden.

Aus Tabelle 3 wird abgeleitet, ob ein Eintrag vorgenommen wird. Für Termin- und Ressourcenplanung wird keine Unterscheidung nach Makro- und Mikro-Level-Transparenz vorgenommen, da die Ergebnisse identisch sind. Die Häufigkeiten sind Tabelle 5 entnommen. Die letzte Zeile gibt den Durchschnittswert an.

	Kommunikationsstatus		Objektstatus		Terminplanung (Makro&Mikro)	Ressourcenplanung (Makro&Mikro)
	Makro	Mikro	Makro	Mikro		
Eingang erhalten	a		< 70% < 40%			
Beschaffung v. Informationen	< 40% < 40%	< 40% < 40%	< 40% < 40%	< 40% < 40%		
Abstimmung in kleinen Gruppen	< 40% > 70%	< 40% > 70%	< 40% > 70%	< 40% > 70%	< 40% > 70%	
Abstimmung in großen Gruppen	< 40% < 40%	< 40% < 40%	< 40% < 40%	< 40% < 40%		< 40% < 40%
Vortrag/ Rücksprache	< 40% < 40%	< 40% < 40%		< 40% < 40%	< 40% < 40%	
Stellungnahme anfordern			< 70% < 70%			
Arbeitsergebnisse mitteilen	< 70% > 70%	< 70% > 70%	< 70% > 70%	< 70% > 70%		
Verteilung von Informationen	< 40% > 70%	< 40% > 70%	< 40% > 70%	< 40% > 70%		
Terminabsprache	< 40% < 70%				< 40% < 70%	< 40% < 70%
Ressourcenreservierung	< 40% < 10%					< 40% < 10%
Dokumentenbearbeitung	> 70%	> 70%	> 70%	> 70%		
Zeichnung einholen			< 70% < 40%			
<b>Durchschnitt</b>	bis 56%	bis 61%	bis 63%	bis 61%	bis 55%	bis 40%

**Tabelle 6: Relevanz von Transparenz im Verwaltungsbüro**

a. Häufigkeit einer Tätigkeit: selten = bis 10%, gelegentlich = bis 40%, häufig = bis 70%, Regelfall = größer 70%, leeres Feld: keine Zuordnung  
Bewertung oben: IVBB, Bewertung unten: Referat III C.



### 3.2.4 Schlußfolgerungen

Die Bildung des Gesamtdurchschnittswertes (56%) zeigt auf, daß Transparenz bei über der Hälfte aller kooperativen Aktivitäten im Büroalltag relevant ist. Betrachtet man die Ergebnisse im Einzelnen, so ergeben sich kaum signifikante Unterschiede, weder im Mikro noch im Makro Bereich, zwischen Transparenz über Kommunikationsbereitschaft bzw. Kommunikationsmöglichkeiten, Transparenz über Verbleib und Status von Objekten sowie Transparenz bei der Terminplanung. Lediglich Transparenz bei der Planung von Ressourcen scheint leicht weniger relevant.

Aus den Ergebnissen kann geschlossen werden, daß im Zuge einer Computerisierung des Büros und einer Ausweitung der Systeme auf den Gruppenraum nutzungsbezogene Transparenz eine wesentliche Rolle spielt. Herrmann et al. schreibt dazu „Nutzungsbezogene Transparenz kann bei Groupware dazu beitragen, die Kommunikation und Kooperation zwischen den Nutzern zu fördern. So wurde ... darauf hingewiesen, daß bei verteilter Bearbeitung von Vorgängen Probleme aus mangelhafter Transparenz des Bearbeitungsstandes resultierten“ [65] (S.201).

Problematisch an Transparenz ist allerdings gerade für den personenbezogenen Bereich der Aspekt der Überwachung. Gerade Videobasierte Systeme wie [36] oder direkt überwachende wie [57] sind hier kritisch zu betrachten. Die Umgebungen wurden vor allen Dingen von Wissenschaftlern eingesetzt, die von vorne herein ein aufgeschlosseneres Verhältnis zur Thematik hatten. Im Umfeld öffentlicher Verwaltungen oder größeren Betrieben ist die Akzeptanz einer solchen „Überwachung“ sehr gering. Zitat eines Probanden: „Dann kann meine Vorgesetzte ja sehen, was ich tue“. Die Leiterin des Referats III C befürchtete vor allen Dingen Probleme mit gewerkschaftlichen Einrichtungen, obwohl sie der Thematik Transparenz offen gegenüber war. Eine direkte Überwachungsmöglichkeit der Tätigkeit eines Arbeitnehmers ist, zumindest in Deutschland, gesetzlich verboten.

Dies korrespondiert zum Recht auf informationelle und kommunikative Selbstbestimmung [130]. Um dieses Recht zu wahren, muß eine Regulierung der Transparenz vorgenommen werden. Anwender und Administratoren müssen hier eingreifen können und nach individuellen Wünschen oder benutzerklassenspezifisch die Auswirkungen der Transparenz beeinflussen können. In [65] wird in diesem Zusammenhang auch von der „Steuerbarkeit wechselseitiger Beeinflussung“ und von „Handlungsflexibilität“ gesprochen (siehe dazu auch [150], S.17). Transparenz über Kommunikations- oder Objektstatus, über Termin- und Ressourcenplanung muß demnach regulierbar sein in Bezug auf:

- ihre Sichtbarkeit,
- ihre Erreichbarkeit,
- ihren Zugang,
- ihre Verfügbarkeit und
- ihre Verteilung.

Die Einführung von Transparenz kann nur erfolgreich sein, wenn gleichzeitig gewährleistet wird, daß sie ohne Überwachung stattfindet.

### 3.3 Technisches Transparenzmodell

#### 3.3.1 Modell der Transparenzdatensätze

Im Abschnitt 2.6 wurde eine Definition in umgangssprachlicher Form von Transparenz gegeben. Im folgenden soll der Begriff Transparenz aus technischer Sicht formalisiert werden. Dabei ist es von Interesse, ein Modell zu finden, welches einer späteren Umsetzung in einer realen Groupware Umgebung gerecht wird.

Transparenz kann, nach den bisherigen Erkenntnissen, als eine Vereinigung verschiedener Informationen aus unter Umständen unterschiedlichen Quellen angesehen werden. Visualisiert macht sie dem Anwender einen bestimmten nutzungsbezogenen Sachverhalt deutlich. Im folgenden wird diese Vereinigung von Informationen als Transparenzdatensatz (TDS) bezeichnet.

Zunächst kann vom eigentlichen Inhalt eines TDS abstrahiert werden. Die folgende Abbildung 18 zeigt schematisch diesen Aufbau:

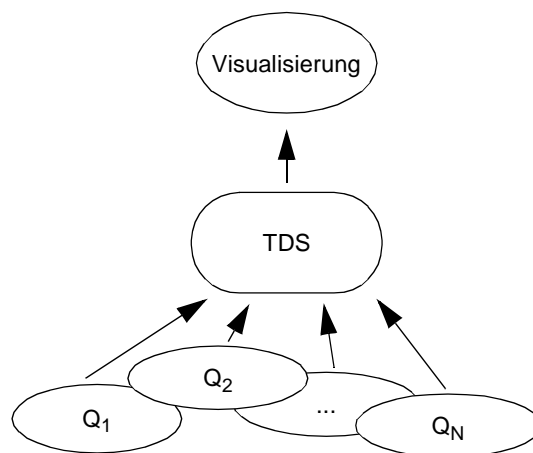


Abbildung 18: Erstellung und Visualisierung eines Transparenzdatensatzes

Zur Verfeinerung dieser Ansicht eignet sich die aus dem Smalltalk-Kontext bekannte *Model-View-Controller* (MVC) Metapher (siehe dazu etwa [53] oder [83]). Unter dem *Model* kann man im MVC-Kontext die Komponenten eines Systems verstehen, die eine (reale) Anwendung simulieren („*components which actually do the work*“). Diese werden getrennt von ihrer visuellen Ausprägung, dem *View*, betrachtet. *Controller* werden verwendet um Nachrichten zum Modell zu versenden und stellen eine Schnittstelle zwischen dem Modell mit seinen assoziierten Ansichten und den interaktiven Eingabegeräten (z.B. Tastatur, Maus) zur Verfügung. Jeder *View* besitzt genau einen *Controller*, der genau einem *Model* zugeordnet ist; ein *Model* hingegen kann beliebig viele *View/Controller* Paare besitzen. Vorteil dieser Metapher ist die Trennung eines Objektes von dessen visueller Ausprägung. Durch die zusätzlich vom Objekt getrennte Kontrolleinheit ergibt sich so die

Möglichkeit unterschiedliche Sichtweisen auf ein Objekt zu implementieren, ohne auf das Objekt selbst einwirken zu müssen.

Um Abbildung 18 nach der MVC-Metapher zu verfeinern, kann ein Transparenzdatensatz als *Model* angesehen werden. Die visuelle Ausprägung, der *View*, ist vom TDS getrennt und erhält nur bei einer Zustandsänderung des TDS von ihm eine Nachricht. Der *View* wird zusätzlich von einem Kontrollmechanismus überwacht. Dieser *Controller* steuert durch eine Ansichtsbeschreibung den *View* auf den TDS. Controller und TDS tauschen Kontrollnachrichten und Meldungen über Zustandsänderungen aus. Abbildung 19 veranschaulicht das *Model-View-Controlling* mit Transparenzdatensätzen.

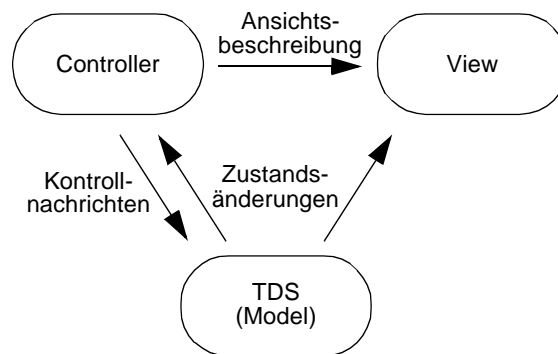


Abbildung 19: Model-View-Controlling mit Transparenzdatensätzen

Dieses Prinzip kann nun erweitert werden. Um die Visualisierung der TDS möglichst flexibel zu gestalten, sollte sie abhängig vom Anwender gestaltet werden. Dazu werden Benutzerprofile benötigt, anhand derer eine entsprechende Entscheidung über die Art der Ansicht eines TDS getroffen werden kann. Der *Controller* ist nach dem MVC-Prinzip derjenige, der die Benutzerprofile auswertet und entsprechende Ansichtsbeschreibungen an den *View* weiterleitet. Dieser verarbeitet die Eingaben zusammen mit dem Zustand des TDS zu einer

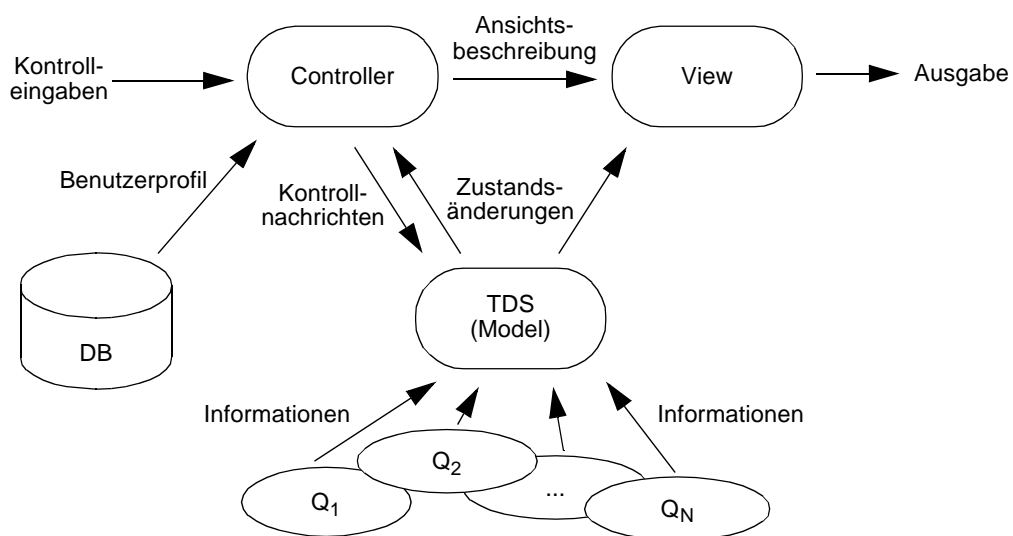


Abbildung 20: MVC -TDS Prinzip

(visuellen) Ausgabe. Der TDS bekommt, wie bereits erwähnt, seine für seinen Zustand nötigen Informationen aus unterschiedlichen Quellen (auf die später noch eingegangen wird). Der Prozeß kann zusätzlich von außen durch Kontrolleingaben an den *Controller* gesteuert werden. Abbildung 20 zeigt diese Erweiterung des MVC-Prinzips graphisch.

### 3.3.2 Geographische Verteilung von Transparenzdatensätzen

Die Einbindung der Möglichkeit einer geographischen Verteilung der TDS ist Voraussetzung für eine Groupware-Umgebung. Eine solche geographische Verteilung von TDS kann man sich wie folgt vorstellen: An einer Lokalität A existieren verschiedene Quellen, die zur Erzeugung von Transparenzdatensätzen geeignet sind. Ebenfalls existiert dort eine Instanz nach dem MVC-TDS-Prinzip. An einer Lokalität B existieren andere Quellen und eine weitere Instanz. Ein einfaches Beispiel dafür sind zwei verschiedene Arbeitsplatzrechner innerhalb einer Verwaltung, an denen jeweils ein Benutzer „eingeloggt“ ist. An der Lokalität A können nun TDS-Ansichten von unterschiedlichen Quellen betrachtet werden. Die Instanzen sind miteinander verbunden und können dadurch auf TDS der jeweils anderen Instanz zugreifen. Dabei kann der selbe TDS an beiden Lokalitäten, u. U. in einer unterschiedlichen Ausprägung, gleichzeitig angezeigt werden. In Abbildung 21 ist dies der TDS der Quelle i aus Lokalität A. Ersichtlich wird in der Abbildung auch, daß ein TDS nicht unbedingt an der Lokalität seiner Quelle angezeigt werden muß.

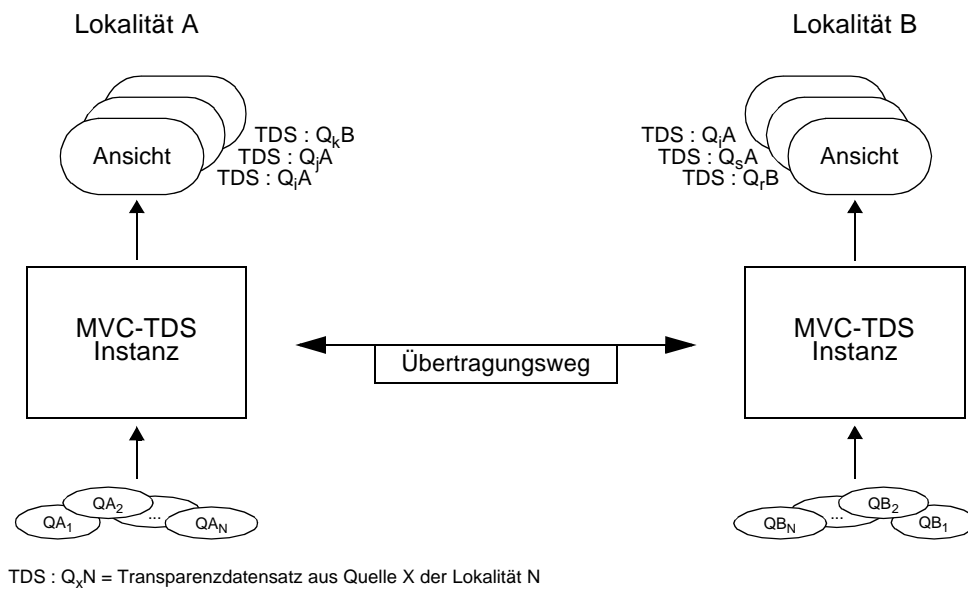


Abbildung 21: MVC-TDS-Prinzip bei geographisch verteilten Lokalitäten

### 3.3.3 Quellen der Transparenzdatensätze

Die bisher nicht näher erläuterten Quellen der Transparenzdatensätze sollen nun betrachtet werden. In Abschnitt 3.1.2 wurden diese bereits indirekt definiert.

Betrachtet man den beispielhaften Arbeitstag des Herrn R. im vorherigen Abschnitt 3.1, so wurden dort die folgenden nutzungsbezogenen Transparenzdatensätze extrahiert:

- Telekonferenz
  - Kommunikationsbereitschaft (Telefonmetapher: Besetzt, kein Anschluß, usw.)
  - Kommunikationsmöglichkeiten (Partner, Werkzeuge, Netz, usw.)
- Terminplanung
- Verbleib von Akten, Umläufen, Unterlagen
- Zugriffsrechte auf Akten, Umläufe, Unterlagen
- Zustand von Akten, Umläufen, Unterlagen
- Zugriffsrechte auf Ressourcen

Die Quellen sind demnach Software für Telekonferenzen, Verwaltung von Akten, Umläufen und Unterlagen, Terminplanung und Ressourcenverwaltung, oder allgemein gesprochen, Groupware-Komponenten.

Prinzipiell ist MVC-TDS-Prinzip unabhängig von der Art der Quelle für einen TDS spezifiziert worden. Auch ist in der Modellbildung offen gelassen worden, in welcher Form der Nachrichtenaustausch zwischen den einzelnen Instanzen zu realisieren ist, ob etwa Client/Server Modelle, Broker- oder Agent-Konzepte verwendet werden sollen. Das Modell, betrachtet man es in seiner Gesamtheit als Vereinigung seiner Instanzen, ist also in diesem Sinne als generisch anzusehen.

### **3.4 Zusammenfassung**

In diesem Kapitel wurden zunächst anhand von Szenarien aus dem Verwaltungsbereich die Anforderungen und die Relevanz bezüglich Transparenz analysiert. Transparenz ist ein wesentlicher Faktor bei der Einführung von neuen Groupware-Technologien. Sie trägt zur Verbesserung der software-ergonomischen Eigenschaften eines Systems bei und verbessert den Bedarf an Erwartungskonformität innerhalb des Gruppenraums. Die Ergebnisse der Analyse lassen sich wie folgt zusammenfassen: Bei über der Hälfte aller kooperativen Aktivitäten im Büroalltag ist Transparenz von wichtiger Bedeutung. Gefordert wird hier

- Transparenz über Kommunikationsbereitschaft und Kommunikationsmöglichkeiten,
- Transparenz bei der Terminplanung,
- Transparenz bei Zugriffsrechten auf Ressourcen und
- Transparenz über Verbleib und Status von Objekten (z. B. Akten, Umläufe, Unterlagen).

Weiterer wichtiger Aspekt ist, daß der Zugriff auf diese Daten so reguliert wird, das keine Überwachung stattfinden kann. Teilweise handelt es sich hier um persönliche Daten. Weiterhin könnten die Daten genutzt werden, um die Tätigkeit eines Arbeitnehmers genau zu überwachen. Transparenz muß demnach regulierbar sein in Bezug auf:

- ihre Sichtbarkeit,
- ihre Erreichbarkeit,
- ihren Zugang,
- ihre Verfügbarkeit und
- ihre Verteilung.

Auf Grundlage dieser Erkenntnisse ist ein technisches Modell zur Transparenz entwickelt worden, welches im nächsten Kapitel als eine Basiskomponente des Referenzmodells dienen wird. Transparenz kann als eine Vereinigung verschiedener Informationen aus unterschiedlichen Quellen angesehen werden. Visualisiert macht sie dem Anwender einen bestimmten nutzungsbezogenen Sachverhalt deutlich. Das Modell der Transparenzdatensätze bedient sich der Modell-View-Controller Metapher und ist in verteilten Szenarien einsetzbar. Als Quellen für die vorgestellten Transparenzdatensätze sind allgemein Groupware-Komponenten identifiziert worden wie etwa Telekonferenz- oder Workflow-Systeme. Dabei ist das Modell prinzipiell unabhängig von der Art der Quelle.

## Kapitel 4

# Konzeption des Referenzmodells

## 4.1 Einleitung

Viele Aufgaben in der öffentlichen Verwaltung werden heute bereits mit Hilfe des Computers und Standard-Applikationen bearbeitet. Wie in der Motivation bereits erläutert, setzt die Verwirklichung des kooperativen computerunterstützten Verwaltungsbüros aber eine wesentlich weitergehende technische Hilfe voraus. Unterschiedliche Technologiebereiche wie Telekooperation oder Workflow haben sich als dazu geeignet erwiesen. Das hier vorgestellte Referenzmodell soll in der Lage sein, einerseits die Nutzung der verschiedenen kooperativen Technologien zu ermöglichen und andererseits diese Nutzung möglichst komfortabel und benutzeradäquat zu gestalten.

Grundlage für das Referenzmodell ist der Einsatz von Software Agenten als integrativer Bestandteil. Wie bereits angedeutet, zwingt die Heterogenität der beiden Basistechniken Workflow und Telekooperation zu einem Umdenken von einer statischen Lösung zu dynamisch flexiblen Ansätzen. Die Definition einer direkten Schnittstelle oder eines Konverters zwischen diesen Basistechniken ist nicht sinnvoll, da die Aufgabengebiete offensichtlich konträr sind. Der Versuch einer solchen Definition wäre auch nur schwer und unbefriedigend zu realisieren, da sich die Informationseinheiten der jeweiligen Komponenten zu sehr von einander unterscheiden. Dennoch ist eine Zusammenführung auf Benutzerebene sinnvoll, da der Nutzen für den Anwender eindeutig vorhanden ist.

Die Berührungspunkte sind über die Aufgabenziele des Benutzers definiert. Der Übergang zwischen den Systemen findet zum Beispiel in dem Augenblick statt, in dem der Nutzer von der alleinigen Bearbeitung einer Akte auf gemeinsame Bearbeitung umstellt, etwa weil er eine Rückfrage hat und per Telekonferenz den zuständigen Sachbearbeiter kontaktiert. Allgemein wird hier der Wechsel von asynchroner auf synchrone Gruppenarbeit beschrieben. Durch ein gefiltertes Weiterreichen der Informationseinheiten und zusätzlich automatisierte Kontrollaktionen kann ein Mehrwert geschaffen werden. Dazu sind autonome und anpassungsfähige Einheiten notwendig, die in der Lage sind, selbständig entsprechend der gerade vorherrschenden Situation Entscheidungen treffen zu können. Software Agenten erfüllen diese Voraussetzungen.

Software-Agenten eignen sich weiterhin für die Erfüllung der ergonomischen Anforderungen. Agenten nehmen ihre Umwelt wahr und reagieren auf Veränderungen in ihr. Sie sind damit in der Lage, sich an den Nutzer anzupassen und ihn individuell zu unterstützen. Durch Interaktion mit anderen Agenten und Menschen können gruppentransparente Aktio-

nen realisiert werden. Die Autonomie und Glaubwürdigkeit der Agenten stellt dabei sicher, daß Überwachung nicht stattfindet. Eine automatisierte Kontrolle und Koordination wird dadurch möglich, daß Agenten auch ohne die direkte Intervention von Menschen in einer zielgerichteten Art die Initiative ergreifen können. Eine vereinfachende Benutzerführung kann somit erreicht werden.

Im folgenden wird zunächst ein technisches Anforderungsprofil des Referenzmodells erstellt. In Abschnitt 4.3 wird dann die Spezifikation des Referenzmodells vorgenommen. Die in Kapitel 3 aufgestellten Benutzeranforderungen und das daraus gewonnene Modell der Transparenzdatensätze werden in das auf Software-Agenten basierende Referenzmodell miteinbezogen. Anschließend werden die einzelnen Agentenklassen dargestellt. Mit einem auf Fuzzy-Logic aufsetzenden Mechanismus zur Entscheidungsfindung wird dann in Abschnitt 4.4 erläutert, wie autonome Handlungsweisen in das Modell aufgenommen werden können.

## **4.2 Anforderungsprofil**

### **4.2.1 Integration unterschiedlicher Groupware-Komponenten**

Der Einsatz von CSCW-Techniken soll sowohl gruppenweite Aktionen als auch die einzelne Person in ihrer Arbeit unterstützen. Workflow-Management-Systeme ermöglichen es, Büroprozesse zu modellieren, zu simulieren und zu realisieren. Telekooperations-Systeme erlauben es innerhalb kürzester Zeit, trotz großer Entfernung zwischen den verschiedenen beteiligten Personen einer kooperierenden Gruppe, Aufgaben effizienter durchzuführen. Sie sind damit entscheidende Komponenten für eine zukunftsweisende Verwaltungsumgebung. Die wesentlichen Fragestellungen, die sich für eine Integration ergeben, sind:

- Wie wird eine Integration koordiniert und kontrolliert?
- Welche Informationen sollen ausgetauscht werden?
- Wie wird eine Übertragung der Informationen bewerkstelligt?

### **4.2.2 Software-ergonomische Integration**

Das tatsächliche Anwendungspotential kooperationsunterstützender Systeme wird im wesentlichen von der software-ergonomischen Gestaltung dieser Systeme sowie der Art ihrer Entwicklung und der Einbettung in die Organisation mitbestimmt. Bisherige Systemgestaltungen sind allerdings vorrangig einzelplatzorientiert und berücksichtigen die Kennzeichen vernetzter, kooperationsunterstützender Systeme nicht in ausreichendem Maße. Die software-ergonomische Integration der verschiedenen Werkzeuge auf der Bedienungsebene soll hier entgegen wirken. Die wesentlichen Sichtweisen zu dieser Problematik sind dabei bereits im Abschnitt 3.2 beschrieben worden. Aus technischer Sicht ergeben sich die folgenden Anforderungen:

- Die Einbindung individueller Benutzerprofile, z. B. von Vorlieben für bestimmte Termine oder für bestimmte Kommunikationsmedien.



- Die automatisierte Informationsbeschaffung<sup>4</sup> wie z. B. zu möglichen Kommunikationspartnern anhand eines bisherigen oder zukünftigen Arbeitsprozeßverlaufs.
- Eine vereinfachende Benutzerführung, z. B. durch automatisiertes Vorladen bzw. Rückführen von Dokumenten oder automatisierten Kommunikationsaufbau.
- Automatisierte Kontrolle des Systems durch linguistische Approximation umgangssprachlich definierter Anforderungen.

### 4.2.3 Transparenz innerhalb des Gruppenraums

In heutigen Verwaltungen ist oft ein Mangel an notwendigen Informationen zum Arbeitsumfeld festzustellen. Nutzungsbezogene oder Makro-Level-Transparenz macht das Handeln einzelner Anwender für andere Kooperationspartner sichtbar und beugt so Problemen bei verteilter Bearbeitung von Vorgängen vor, die aus mangelhafter Transparenz des Bearbeitungsstandes resultierten. Makro-Level-Transparenz innerhalb einer Gruppe muß zum Nutzen jedes Einzelnen in der Gruppe etabliert werden. Notwendig ist daher wie in Abschnitt 3.2 bereits erläutert die Schaffung von

- gruppenweiter Makro-Level-Transparenz
- Transparenz ohne Überwachung.

Dabei soll Makro-Level-Transparenz, unter Wahrung des Rechts auf kommunikative und informelle Selbstbestimmung, vor allen Dingen Klarheit schaffen über:

- Kommunikationsbereitschaft und Kommunikationsmöglichkeiten
- Verbleib und Status von Objekten (z. B. Akten, Umläufe, Unterlagen)
- Termin- und Ressourcenplanung.

Der Schwerpunkt wird dabei auf den personenbezogenen Bereich der Makro-Level-Transparenz gelegt. Dies vor allen Dingen im Hinblick darauf, daß Workflow-Management-Systeme bereits im objektbezogenen Bereich Unterstützung bieten (vergleiche Abschnitt 2.6.3). Ebenso sind in diesem Bereich bereits andere wissenschaftliche Arbeiten plaziert (siehe etwa [50], [62] oder [136]).

## 4.3 Spezifikation des Referenzmodells

### 4.3.1 Grobkonzeption

Zunächst soll ein Grobkonzept des Referenzmodells aufgestellt werden. Basis ist ein Agentenverbund, der, wie im folgenden beschrieben wird, die Anforderung nach software-ergonomischer Integration heterogener Groupware-Komponenten unter Berücksichtigung des Ansatzes zu nutzungsbezogener Transparenz im Gruppenraum erfüllt.

---

4. Nicht im Sinne automatischer Informationssuche, z. B. im Internet, sondern die Bereitstellung von Informationen, die sich aus dem unmittelbaren Arbeitsumfeld ergeben

Für die Agenten können Basisklassen aus den obigen Anforderungen abgeleitet werden. Über die Steuermechanismen dieser Agentenklassen ist der Anwender in der Lage, in einer einheitlichen Umgebung Services und Applikationen zu kontrollieren und ohne Medienbruch zu verwenden. Die automatische Koordination zwischen den Agenten erlaubt eine ergonomische Benutzungsschnittstelle zum Anwender. Die Agenten bilden dazu jeweils eine Schale um ihren technologischen Bereich.

Das hier vorgestellte Referenzmodell zeichnet sich durch einen offenen Agentenverbund aus. Für jede Groupware-Komponente ist eine Klasse von Agenten verantwortlich. Eine Erweiterung des Modells ist auf Basis einer neuen Agentenklasse möglich (siehe dazu auch Abschnitt 8.3 „Ausblick auf weitere Entwicklungen“). Innerhalb einer Agentenklasse werden die zu der entsprechenden Groupware-Komponente gehörenden Informationen benutzerspezifisch aufbereitet und angezeigt oder weitergeleitet. In einer Sitzung wird der Anwender von seinem persönlichen Agenten individuell unterstützt. Sie handeln für ihren „Auftraggeber“ als Stellvertreter in der ihnen aufgetragenen Sache, wie etwa dem Aufbau einer Telekonferenz. Ein persönlicher Agent kennt dabei alle Integritätsbedingungen, die mit einem Bearbeitungsauftrag verbunden sind. Daraus ergeben sich im Zusammenhang mit einem Bearbeitungsauftrag (für den Anwender verborgen) kaskadisch Nachrichten an andere Agenten, die für einen anderen relevanten Abschnitt verantwortlich sind. Anhand der Anforderungen aus Abschnitt 3.2 sind die folgenden Kategorien identifiziert worden:

- Telekooperations-Agenten
- Workflow-Agenten
- Termin-Agenten
- Administrations-Agenten

Das daraus entstehende Referenzmodell einer computergestützten kooperativen Verwaltungsumgebung basierend auf Software-Agenten ist in Abbildung 22 graphisch dargestellt.

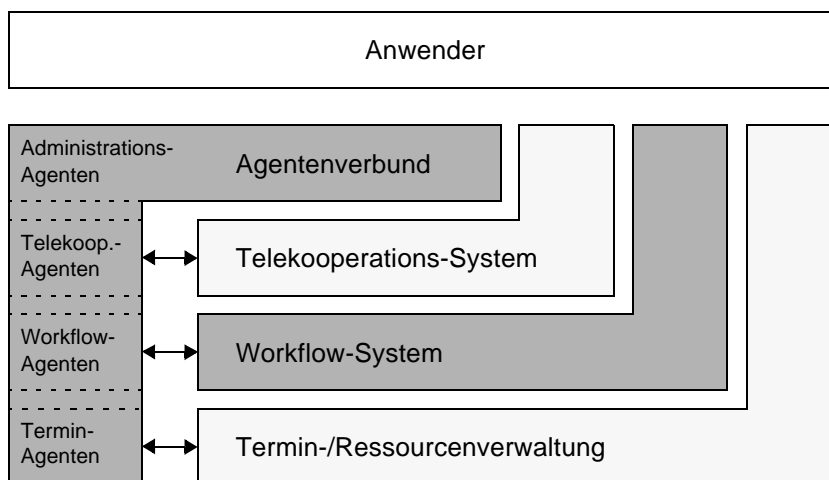


Abbildung 22: Grobkonzept des Referenzmodells

Die ortsunabhängigen Agenten erlauben individuelle Einstellungen, die von einem Administrations-Agenten verwaltet werden und damit jederzeit und überall im Agentenverbund zur Verfügung stehen. Dadurch ist es beispielsweise auch möglich, eine Verbindung zu einem Kooperationspartner aufzubauen, wenn dieser sich nicht an seinem üblichen Arbeitsplatz aufhält.

Der Telekooperations-Agent übernimmt die Integration einer Telekooperationslösung und ermöglicht die Steuerung der Software. Die Aufgaben umfassen beispielsweise das Anzeigen des Kommunikationsstatus potentieller Partner und den Aufbau einer Kommunikation inklusive der Auswahl eines geeigneten Mediums.

Der Anschluß an ein Workflow-System wird durch einen Workflow-Agenten realisiert. Durch Verschmelzung von mehreren Informationsbereichen eines Workflow-Systems kann Metainformation geschaffen und unter Verwendung von neuen Metaphern dem Anwender transparent gemacht werden. Die Weiterleitung der Metainformationen des Workflow-Agenten an einen Telekooperations-Agenten erlaubt weiterhin die optimale Wahl eines Kooperationspartners. Neben der automatischen Bestimmung einer zuständigen Person (z. B. Verfasser einer Akte) wird zusätzlich ein geeignetes Kommunikationsmittel, je nach Präferenzen und Zustand (etwa E-Mail oder Videokonferenz), automatisch durch die persönlichen Agenten bestimmt.

Die Vermittlung von Terminen und die Reservierung von Ressourcen ist dem Termin-Agenten vorbehalten. Die Information zu Terminen kann dazu genutzt werden, den Kommunikationsstatus einer Person zu bestimmen.

Die Agenten bestehen aus einem applikationsabhängigen und einem applikationsunabhängigen Teil. Die interne Kommunikation zwischen Agenten erfolgt auf der Basis einer applikationsunabhängigen Agentensprache, womit grundsätzlich eine Erweiterung um andere Groupware-Lösungen ermöglicht wird. Applikationsabhängige Teile greifen auf die Schnittstellen (APIs) einer konkreten Groupware-Lösung zu.

Die Koordination zwischen den Agenten wird von Administratoren übernommen. Anfragen können über den Administrator an andere Agenten weitergeleitet werden. Diese Agentenklasse hat zusätzlich die Aufgabe, die anderen Agenten mit Benutzerprofilen zu versorgen.

Bevor in den folgenden Abschnitten die einzelnen Agenten näher spezifiziert werden, soll der Zusammenhang Agenten - Transparenzdatensätze aufgezeigt werden. Die Verknüpfung resultiert in einem verfeinerten Referenzmodell.

### **4.3.2 Transparenzdatensätze im Referenzmodell**

Das in Abschnitt 3.3 vorgestellte Modell der Transparenzdatensätze (TDS) ist Basis für die Einführung von Transparenz in einer Groupware-Umgebung. Transparenz wird als eine Vereinigung verschiedener Informationen aus unterschiedlichen Quellen angesehen, die visualisiert dem Anwender einen bestimmten Sachverhalt deutlich macht. Eine Quelle ist dabei eine Groupware-Komponente. Abbildung 23 vereinigt das TDS-Modell mit dem

Referenzmodell. Das Ergebnis verfeinert das Referenzmodell und zeigt die Bezüge zwischen einer Groupware-Komponente und dem korrespondierenden Agenten auf.

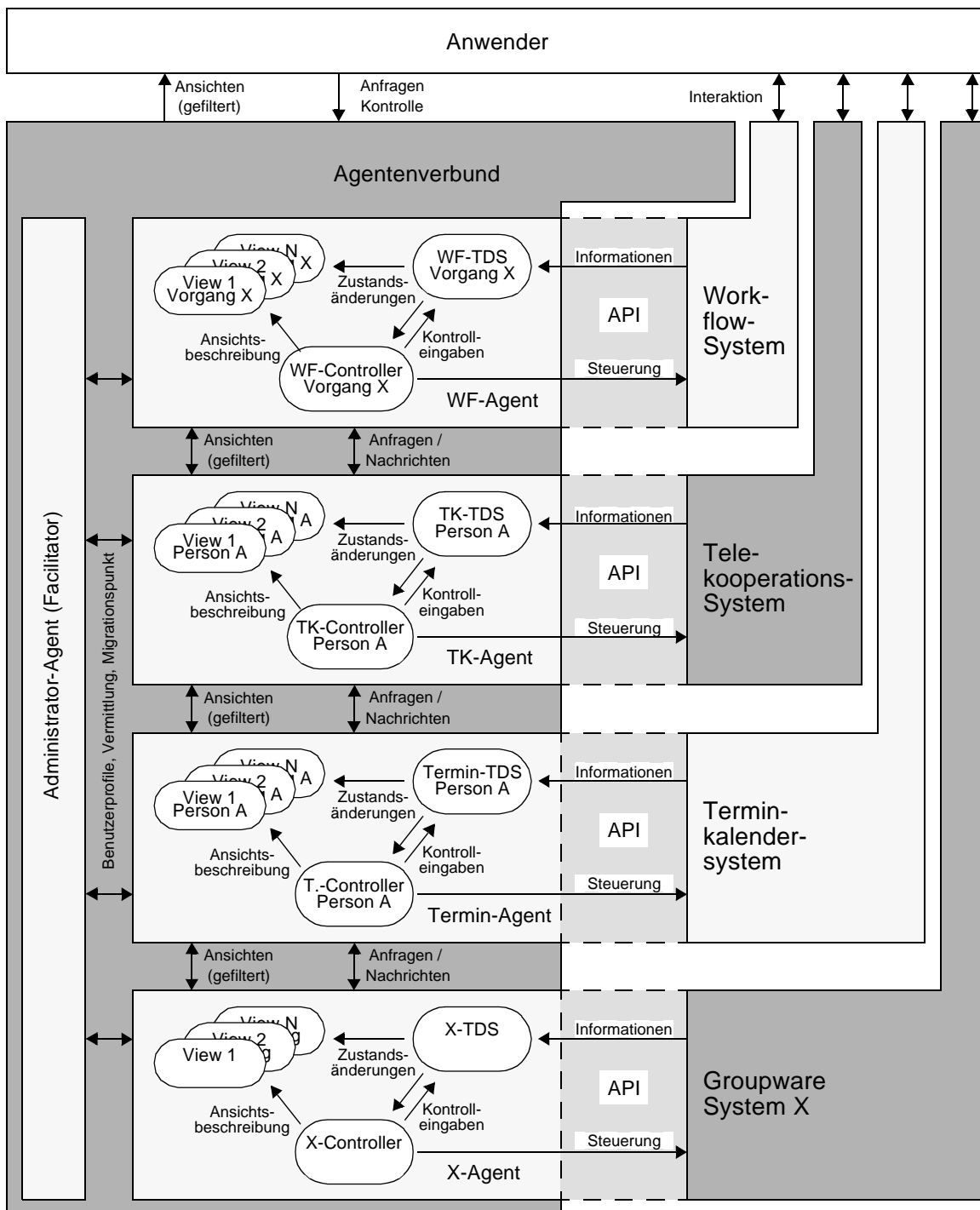


Abbildung 23: Verfeinertes Referenzmodell

Um eine Groupware-Komponente als Informationsquelle anzusprechen, wird die Programmierschnittstelle (*Application Programmers Interface*, API) der Komponente vom Agenten verwendet. Die Informationen werden in einem Transparenzdatensatz abgelegt. Je nach

Anfrage werden neue Transparenzdatensätze erzeugt oder bereits bestehende verwendet. Der Controller bestimmt den Rahmen der Informationen durch Kontroll- und Steuernachrichten an die Groupware-Komponente. Außerdem bestimmt er die Ansicht auf einen Transparenzdatensatz für den jeweiligen Nutzer eines TDS. Zustandsänderungen eines TDS werden dem Controller und der Ansicht automatisch mitgeteilt. Zwei Beispiele sollen den hier dargestellten Sachverhalt verdeutlichen. Zur besseren Erläuterung wird dabei auf erst später näher beschriebene Funktionalitäten vorausgegriffen:

- Der TDS des Termin-Agenten A ist mit Informationen über die Termine von Person A gefüllt. Person A fragt nach dem nächsten Termin. Die Informationen werden gefiltert und der aktuelle Termin im Klartext dargestellt.
- Der TDS des Termin-Agenten A ist mit Informationen über die Termine von Person A gefüllt. Der Termin-Agent einer Person B fragt nach einem möglichen Termin in der nächsten Woche. Die Informationen werden gefiltert, als Fuzzy-Set dargestellt und dem Agenten A übermittelt. Dieser berechnet den Termin und teilt ihn Agent B mit.

Die jeweilige Anfrage wird vom Controller in eine Ansichtsbeschreibung umgewandelt und eine entsprechende Ansicht wird erstellt. Wesentlich ist, daß eine Überwachung seitens des Controllers in Bezug auf die Inhalte und die Verteilung der Informationen stattfindet. Die geforderte Regulierung der Transparenz in Bezug auf ihre

- Sichtbarkeit,
- Erreichbarkeit,
- Zugang,
- Verfügbarkeit und
- Verteilung

ist damit gewährleistet.

Eine der Hauptanforderungen an Transparenz, die Gewährleistung, daß keine Überwachung von Personen stattfinden kann, ist hiermit erfüllbar.

Der Administrations-Agent vermittelt Anfragen zwischen Agenten und leitet sie weiter. Dazu kennt er die Aufgabenbereiche und Standorte der einzelnen Agenten in seinem Zuständigkeitsbereich. Er dient weiterhin als Migrationspunkt für andere Agentenklassen. In einem realen Szenario steht einem Anwenderverbund, z. B. einer Abteilung, je ein Administrator-Agent zur Verfügung. Andere Agentenklassen, beispielsweise Telekooperations-Agenten, instanzieren sich je nach Anwendungsfall einmal pro Anwender. Beendet ein Nutzer eine Sitzung, migriert sein persönlicher Agent zum Administrator und kann dort in eingeschränkter Form seine Aufgaben weiter durchführen. Meldet sich der Anwender wieder beim Agentenverbund an, migriert der Agent wieder zurück und steht in seiner vollen Funktionsfähigkeit dem Anwender zur Verfügung.

Um eine Vermittlung zwischen den einzelnen Agenten zu gewährleisten, muß der Agentenverbund auf einer für Multi-Agentensysteme geeigneten Plattform aufsetzen. Diese Basisarchitektur wird im folgenden Abschnitt beschrieben.

### 4.3.3 Basisarchitektur des Agentenverbundes

Betrachtet man die Ergebnisse des Abschnitt 2.5 über Software Agenten, so sollte eine geeignete Architektur für den hier entwickelten Agentenverbund ausgewählt werden. Für die Anforderungen des Referenzmodells eignet sich besonders das halföderative System Nummer 5 aus Abbildung 14 auf Seite 31. Wie in Abbildung 24 verdeutlicht, bietet dieses System den Vorteil, daß sich bekannte Agenten direkt ansprechen können (rechte Seite). Falls notwendig, kann sich ein Agent über den Facilitator trotzdem Empfehlungen für einen bestimmten Service geben lassen (linke Seite). Ein halföderatives System zeichnet sich durch Schnelligkeit und eine einfache Administration aus. Der Facilitator kann einfach gehalten werden und zusätzlicher Kommunikationsaufwand entfällt.

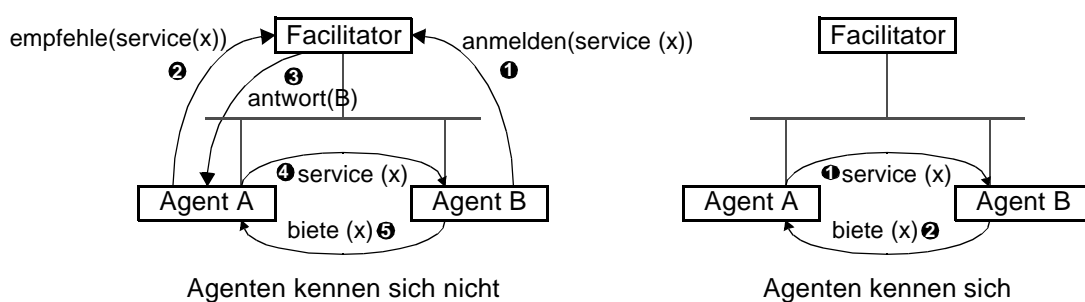


Abbildung 24: Halföderatives Agentensystem

Die Aufgaben des Facilitators können vom Administrator-Agent übernommen werden. Falls notwendig, werden Anfragen über ihn vermittelt. Falls Agenten sich bereits kennen, kann eine Anfrage direkt durchgeführt werden.

### 4.3.4 Workflow-Agent

Aufgabe eines Workflow-Agenten ist die Vermittlung zwischen dem Agentenverbund und einem Workflow-System. Im wesentlichen werden dazu Daten aus dem Workflow-System erfragt, gefiltert und weitergeleitet bzw. umgekehrt eingehende Daten von anderen Agenten aufbereitet und in das Workflow-System geführt. Als standardisiertes Beispiel eines Workflow-Systems wird das Referenzmodell der Workflow Management Coalition (WfMC) verwendet (siehe Abschnitt 2.4.3 ab Seite 23). Abbildung 25 zeigt schematisch die Integration des Workflow-Agenten in das Referenzmodell der WfMC. Die grau umrandeten Komponenten sind dabei aus dem WfMC-Referenzmodell übernommen worden.

Die Interaktion mit dem WfMC-Referenzmodell erfolgt über die Workflow-API und deren Austauschformate. Interessant sind die Informationen, die sich aus den Arbeitslisten ergeben und damit das Interface 2 "Workflow Client Application Interface" [162]. Es können hier Workflow-Prozeßdefinitionen und deren Attribute sowie Workflow- und Applikationsdaten abgefragt und gespeichert werden. Der Workflow-Agent agiert hier in ähnlicher Weise wie der Arbeitslisten-Verwalter. Die Daten werden in einem Transparenzdatensatz abgelegt und stehen dann dem Agentenverbund zur Verfügung. Umgekehrt können Daten

vom Controller des Workflow-Agenten an das Workflow-System weitergeleitet werden. Dadurch ist es beispielsweise möglich, von außen einen Vorgang innerhalb des Workflow-Systems zu initiieren.

Gerade letzterer Punkt kann im Verwaltungsumfeld sehr interessant sein. Eine Organisationseinheit wird im allgemeinen ihr Workflow-System aus Sicherheitsgründen vor äußeren Eingriffen schützen. Dies bedeutet aber auch, daß Bearbeitungsvorgänge immer innerhalb der Organisationseinheit gestartet werden müssen. Ein Kunde löst einen solchen Vorgang durch ein Telefonat, einen Brief oder ein Fax aus. Diese Angaben müssen in elektronische Form gebracht werden, bevor das Workflow-System in Aktion treten kann. In Kombination mit den neuen Möglichkeiten des Internet kann diese Arbeit durch den Workflow-Agenten übernommen werden. Formblätter im Internet werden angeboten und nach einer Plausibilitätsprüfung durch den Workflow-Agenten in das Workflow-System gebracht und an einen zuständigen Sachbearbeiter automatisch weitergeleitet (siehe dazu auch Abschnitt 8.3 „Ausblick auf weitere Entwicklungen“).

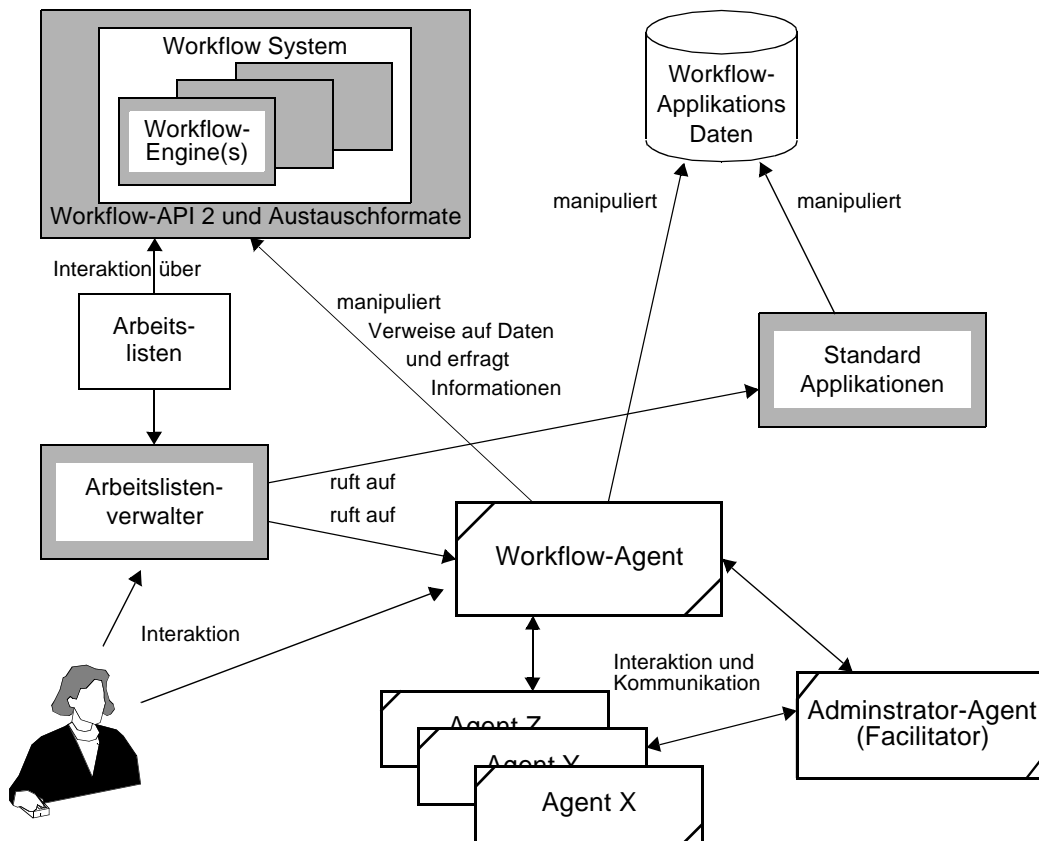


Abbildung 25: Intergration des Workflow-Agenten in das Referenzmodell der WfMC

Über den Zugriff auf die Workflow-API ist es prinzipiell für den Workflow-Agenten möglich, nicht nur Initiierung und Statusabfragen durchzuführen, sondern den vollständigen

Aufgabenbereich eines Arbeitslisten-Verwalters zu übernehmen. Dies sollte aber nur für spezielle Fälle und in eingeschränktem Maße durchgeführt werden.

Der Workflow-Agent wird zum Initialisierungszeitpunkt einer realen Person zugeordnet. Wird der Agent vom Arbeitslisten-Handler, der ihn als externe Applikation ansieht, aufgerufen, wird die Person übernommen, für die der Arbeitslisten-Verwalter derzeit zuständig ist. Wird er von einem anderen Agent aktiviert, wird der Name der Person mit übergeben. Durch die Zuordnung ist der Agent in der Lage, im Interesse "seines" Benutzers zu agieren.

Im folgenden sollen zunächst die zu verarbeitenden Daten und Informationen für den Workflow-Agent näher spezifiziert werden.

Die aktuelle Arbeitsliste enthält alle notwendigen Informationen des gerade bearbeiteten Vorgangs (aktive Prozeßinstanz). Aus dieser Liste können für weitere Aktionen wichtige Informationen extrahiert werden. Die mit dem Vorgang verknüpften Datenobjekte, wie Leitwege und Dokumente, sind als Verweis oder direkt vorhanden. Weiterhin fallen darunter die folgenden für jeden Vorgang ausgezeichneten Personen (vergleiche Abschnitt 2.4.2 ab Seite 22):

- der Erzeuger des Vorgangs
- der Vorgänger der gerade aktuellen Arbeitseinheit
- der Nachfolger, wobei dies eine Rolle sein kann

Im Zusammenhang mit den meisten in der Verwaltung vorkommenden Vorgängen ergeben sich weiter:

- der verantwortliche Sachbearbeiter
- der Entscheidungsträger

Je nach Vorgang können weitere ausgezeichnete Rollen existieren. Für verteilt arbeitende Ämter, etwa zwischen Berlin und Bonn, kann nicht angenommen werden, daß jeder Person jede andere Person, die eine solche Rolle annimmt, bekannt ist. Der Workflow-Agent kann diese Information allerdings über die Workflow-API extrahieren. Bei einem automatisierten Kommunikationsversuch wird diese dann beispielsweise verwendet, um den richtigen Partner anzusprechen. Der Anwender muß an dieser Stelle nur die Wahl der Rolle oder Stelle treffen, mit der er kommunizieren will.

Sind die Informationen nicht vor Ort erfragbar, weil es sich um mehrere (etwa über das Interface 4 der WfMC) miteinander verbundene Workflow-Engins handelt, tritt ein weiterer Workflow-Agent in Aktion. Die Vermittlung übernimmt der Administrator-Agent (Facilitator). Unter Umständen ist es dazu notwendig, andere Facilitatoren anderer Föderationen zu befragen. Ein Beispiel: Die Oberfinanzdirektion Berlin mit ihren Bauämtern sowie das Bundesbauministerium in Bonn haben sich für den Einsatz von automatisierter Vorgangsbearbeitung entschieden und verwenden dazu je eine Workflow-Engine. Um die Kommunikation und Kooperation zwischen den Bauämtern so ergonomisch wie möglich zu gestalten, wird ein System nach dem hier vorgestellten Referenzmodell eingesetzt. Aus technischer Sicht eines Workflow-Agenten in Bonn (WFA-Bonn) ergibt sich dann das in Abbildung 26



dargestellte Bild. Um Informationen über Vorgänge in Berlin zu erhalten, erfragt er einen dafür zuständigen Workflow-Agenten in Berlin über seinen Facilitator (2). Dieser gibt die Anfrage weiter an den Facilitator der Föderation Berlin (3) und erhält den Namen des dort zuständigen Agenten (4), den er an seinen Agenten weitergibt (5). Die Agenten in Bonn und Berlin handeln anschließend die Informationen direkt miteinander aus (6+7). Die Föderationen tauschen Informationen über die jeweiligen Dienste aus (1), um eine Kooperation zu ermöglichen.

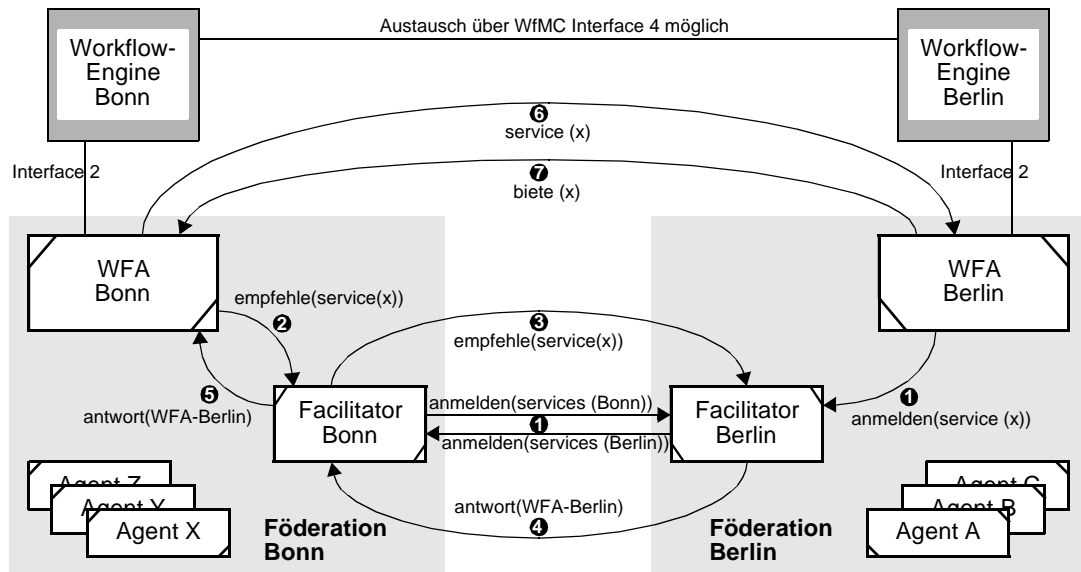


Abbildung 26: Beispiel einer verteilten Agenten-Föderation

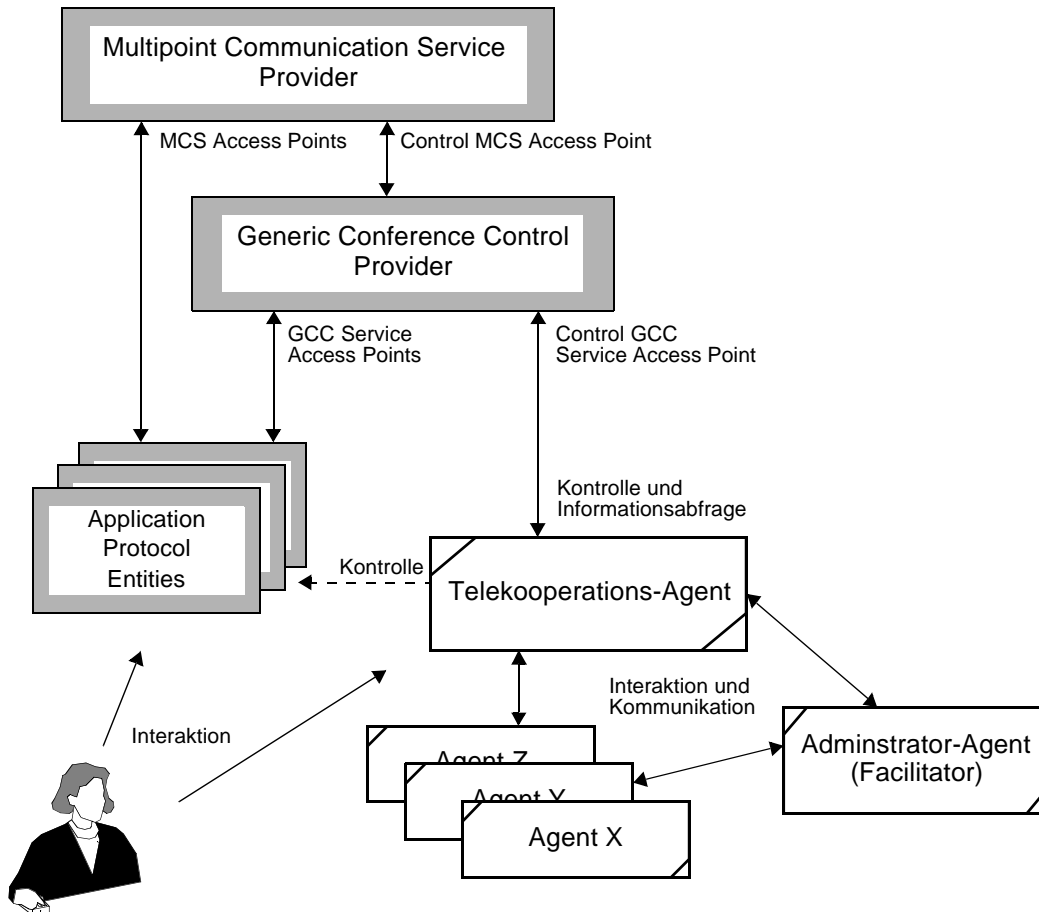
### 4.3.5 Telekooperations-Agent

Der Telekooperations-Agent übernimmt die Integration und Ansteuerung von Telekooperationslösungen. Die Steuerung umfaßt Konfiguration, Auf- und Abbau einer Telekonferenz oder anderer Telekommunikationsmedien inklusive der Auswahl der Partner. Weiterhin können Informationen, etwa zu derzeit laufenden Konferenzen, erfragt werden. In Abhängigkeit von der Komplexität der Telekooperations-Software übernehmen ein oder mehrere Agenten die Steuerung.

Die Auswahl von Kooperationspartner erfolgt mit Hilfe der Adreßverwaltung des Administrator-Agenten und Transparenzdatensätzen anderer Agenten, beispielsweise des Workflow-Agenten. Je nach Leistungsfähigkeit der Telekommunikations-Software können zwei oder mehrere Personen an der Konferenz teilnehmen.

Wie bereits angesprochen, besteht auch der Telekooperations-Agent aus einem applikationsabhängigen und einem applikationsunabhängigen Teil. Die interne Kommunikation zwischen den Agenten folgt auf der Basis eines applikationsunabhängigen Protokolls, womit eine spätere Erweiterung um andere Telekooperationslösungen ermöglicht wird. Die applikationsabhängigen Teile greifen auf die Schnittstellen (APIs) der Telekooperationslösungen zu. Die folgende Abbildung beschreibt die Leistungen des Telekooperations-Agen-

ten graphisch. Als Telekooperationslösung wird hier der Standard T.120 eingesetzt, um die Allgemeingültigkeit der Lösung zu unterstreichen.



**Abbildung 27: Integration des Telekooperations-Agenten in die T.120 Empfehlung der ITU**

Abbildung 27 zeigt die Konfiguration für einen Konferenzknoten innerhalb der Empfehlung der *International Telecommunication Union* (ITU). Der Telekooperations-Agent übernimmt hier T.120-seitig die Funktion des *Node-Controllers*. Über den *Control Services Access Point* des *Generic Conference Control (GCC) Provider* kann der Agent Konferenzen kreieren, beenden, Einladungen versenden und bestehenden Konferenzen beitreten. Informationen zu gerade laufenden Konferenzen und wie einer solchen beigetreten werden kann, können ebenfalls abgefragt werden. Dadurch ist der Agent in der Lage, die volle Kontrolle über einen Konferenzknoten auszuüben und den Kommunikationsstatus bzw. die Erreichbarkeit potentieller Partner transparent zu machen. Die Informationen werden wiederum in einem Transparenzdatensatz abgelegt und stehen somit dem Agentenverbund zur Verfügung. Der *GCC Control Services Access Point* bietet hierzu die Services:

- GCC-Conference-Create,
- GCC-Conference-Query,
- GCC-Conference-Join,

- GCC-Conference-Invite,
- GCC-Conference-Add,
- GCC-Conference-Disconnect und
- GCC-Conference-Terminate an.

Für die Etablierung einer Konferenz werden im T.120 mehrere Prozeduren vorgeschlagen. Je nach Anwendungsfall soll eine der folgenden Strategien angewendet werden (siehe auch Abschnitt 2.3.3):

- *Meet-Me*: Die Konferenz wird in einer Multi-Point-Controll-Unit (MCU) von einem Terminal-Knotenpunkt aus oder der MCU selbst etabliert. Weitere Terminal-Knotenpunkte schließen sich eigenständig der Konferenz an.
- *Call-out*: Die Konferenz wird von einer MCU etabliert. Terminal-Knotenpunkte werden eingeladen und schließen sich der Konferenz an.
- *Call-through*: Diese Strategie ist identisch zur Call-out Strategie mit dem Unterschied, daß die Etablierung von einem Terminal-Knotenpunkt ausgeht.
- *Point-to-Point*: Im Unterschied zu den bisher vorgestellten Strategien wird in diesem Fall keine MCU benötigt. Die Etablierung geht von einem Terminal-Knotenpunkt aus, der einen weiteren Knoten zu einer Konferenz einlädt.
- Etablierung über Multiport-Terminals: Ein Multiport-Terminal faßt die Funktionalität einer MCU und eines Terminal-Knotenpunktes zusammen. Die Etablierung einer Konferenz erfolgt analog der bisher vorgestellten Strategien.

Die Strategien *Meet-Me* von einem Terminal-Knotenpunkt, *Call-through* und *Point-to-Point* eignen sich besonders für das hier vorgestellte Integrationsmodell, da die Initiative vom Telekooperations-Agenten ausgehen kann. Außerdem kann die Tatsache genutzt werden, daß sich die einzelnen Agenten untereinander austauschen können. Vorabinformationen aus Transparenzdatensätzen können so genutzt werden, um die Etablierung möglichst benutzeradäquat zu gestalten. Ein Beispiel: Ein Vorgesetzter verlangt eine Rücksprache mit einem zuständigen Sachbearbeiter. Der Telekooperations-Agent erfragt beim Workflow-Agenten die Daten der aktuellen Arbeitsliste. Die Daten werden vom Workflow-Agenten gefiltert und der entsprechende Sachbearbeiter wird ermittelt. Der Telekooperations-Agent nimmt Kontakt mit dem dem Sachbearbeiter zugeordneten Telekooperations-Agenten auf. Es wird ein Kommunikationsmittel und ein Kommunikationszeitpunkt (unter Umständen durch Rückfrage beim jeweiligen Nutzer) verhandelt. In diesem Beispiel ist das Ergebnis der Verhandlungen eine sofortige Videokonferenz. Diese wird mit Hilfe der Point-to-Point Strategie aufgebaut.

Das Ergebnis der Aushandlung kann auch eine andere Kommunikationsart wie ein elektronischer Anrufbeantworter oder E-Mail, oder ein anderer Termin für eine Videokonferenz (die dann zum gewählten Zeitpunkt automatisch initiiert werden würde) sein. Ein Austausch mit Agenten zur Ressourcen- und Terminverwaltung kann hierzu notwendig werden. Die Wahl des Kommunikationsmediums ist nur durch die technischen Möglichkeiten der Teilnehmer einer Kommunikation beschränkt.

An dieser Stelle wird deutlich, daß eine Integration anderer Kommunikationsmedien wie etwa E-Mail sinnvoll ist. E-Mail Adressen oder Telefonnummern werden vom Administrator-Agent mitverwaltet und können abgefragt werden. Die Integration eines E-Mail Werkzeuges kann somit über Aufruf und Parameterübergabe erfolgen.

### 4.3.6 Ressourcen- und Terminkalender-Agent

#### 4.3.6.1 Terminkalendersysteme

Die Vermittlung von Terminen und die Reservierung von Ressourcen wird vom Termin-Agenten durchgeführt. Unter Ressourcen werden hier allgemein zugängliche Räume oder Geräte verstanden, die zur Benutzung der Reservierung innerhalb einer Verwaltung bedürfen. Die eigentliche Verwaltung der Termine wird von einem Terminkalendersystem durchgeführt. Der Agent steuert diese Komponente über deren Programmierschnittstelle (API) an. Im Gegensatz zu den Bereichen Telekommunikation und Workflow hat sich hier kein Referenzmodell gebildet. Das *Calendar and Scheduling Interface* (CSA), welches von der X/Open Group und der X/400 API Assosiation definiert wurde [101], bietet eine Schnittstelle zum Austausch von Terminkalenderdaten und zum Ansteuern von Terminkalenderdiensten. Wesentliche Funktionalitäten, die dabei zum Tragen kommen, sind das Abfragen von Terminen (innerhalb eines bestimmten Zeitraums) und das Eintragen von Terminen.

Je nach Leistungsfähigkeit des Terminkalendersystems übernimmt der Agent die Verhandlung von neuen Terminen oder läßt sie vom System selbst durchführen. Ein wesentlicher Aspekt ist in jedem Fall die Schaffung von Transparenz und deren Kontrolle. Die Informationen zu Terminen werden dazu genutzt, den Kommunikationsstatus einer Person zu bestimmen. Anhand eines Beispiels soll dieser Sachverhalt verdeutlicht werden.

Der folgende Datensatz basiert auf dem *Calendar and Scheduling Interface* und stellt einen Termin zwischen drei Personen dar:

```
BEGIN:CALENDAR
BEGIN:EVENT
CATEGORIES:MEETING
DTSTART:19990216T160000Z
DTEND:19990216T170000Z
ATTENDEE;ROLE=OWNER;STATUS=CONFIRMED:Ralph Peters
ATTENDEE;ROLE=ATTENDEE;STATUS=TENTATIVE:Stefan Noll
ATTENDEE;ROLE=ATTENDEE;STATUS=CONFIRMED:Martin Rausch
DESCRIPTION:Review des Projektantrags Virtue II
LOCATION:Raum 242
RESOURCES:OVERHEAD PROJECTOR
CLASS:PUBLIC
END:EVENT
END:CALENDAR
```

Aus dem Datensatz wird ersichtlich, daß die drei Personen am 16.02.1999 zwischen 16 und 17 Uhr einen Termin vereinbart haben. Weiterhin wird eine Thematik für das Meeting und

eine Örtlichkeit angeben. Für einen Dritten kann diese Information dazu genutzt werden, warum beispielsweise ein spontan initiiertes Kommunikationsversuch innerhalb dieses Zeitraums scheitert. In dringenden Fällen kann die Information zur Örtlichkeit genutzt werden, trotzdem eine Verbindung aufzubauen. Geht man von einer geographisch verteilten Büroumgebung aus, etwa dem Berlin-Bonn Szenario der Bundesministerien, ist diese Art von Transparenz offensichtlich vorteilhaft.

Der Terminkalender-Agent hat für seinen Benutzer Zugriff auf diese Daten. Sie werden auf Anfrage in einem Transparenzdatensatz abgelegt und können dann weiterverarbeitet werden. Dabei ist die Filterung der Daten anhand der Benutzerpräferenzen ein wichtiger Aspekt. Je nach Einstellung kann die Einsicht auf bestimmte Teile verwehrt sein. Dies Begründet sich durch die Forderung nach Schutz vor Überwachung. Im Extremfall kann jegliche Information verweigert werden. Der Agent schützt hier das Recht auf kommunikative und informelle Selbstbestimmung seines Nutzers.

Um dieses Recht möglichst flexibel zu gestalten, ist es notwendig, auf die Bedürfnisse der Anwender und deren Eigenschaften besonders einzugehen. Die Umsetzung von natürlich-sprachlichen Attributen ist dazu ein wichtiger Bestandteil. Menschliche Formulierungen wie „dringend“, „wichtig“ oder „unter Umständen“ müssen in die Präferenzen einarbeitbar sein. Um auf den folgenden Abschnitt 4.4 „Entscheidungsfindung – Autonomie für Agenten“ vorwegzugreifen, Fuzzy-Logic bietet diese Möglichkeit. Die Umwandlung der Attribute kann in sogenannten linguistischen Variablen erfolgen. Diese Attribute können sowohl bei einem spontanen Kommunikationsversuch verwendet werden, als auch bei der Vereinbarung eines Termins. Diese Methode hat gegenüber anderen Systemen, die einen rein regel-basierten [64] oder einen Prioritäten-Ansatz [18] verwenden, den Vorteil, daß sie mehr Flexibilität verspricht und Benutzerwünsche besser umgesetzt werden können.

Der Termin-Agent muß, um für seinen Nutzer handeln zu können, mit anderen Agenten kommunizieren. Zwei Beispiele sollen das Zusammenspiel der Agenten untereinander verdeutlichen.

#### **4.3.6.2 Terminvereinbarung**

Falls das Terminkalendersystem nicht in der Lage ist, Termine zu vereinbaren, muß das Agentensystem diese Aufgabe übernehmen. In jedem Fall kann es aber sinnvoll sein, das Aushandeln eines Termins dem Agentenverbund zu überlassen, da dieser flexibler auf die Benutzerwünsche eingehen kann. Abbildung 28 zeigt das Aushandeln eines Termins am oben aufgezeigten Beispiel.

Zunächst wird vom Initiator, hier der Termin-Agent von Herrn Peters, ein Terminwunsch an die Agenten der Teilnehmer Noll und Rausch und einen Agenten zur Raumplanung gereicht (1). Die angesprochenen Agenten ermitteln für ihre Anwender die freien Termine und erstellen zusammen mit den Präferenzen ein resultierendes Fuzzy-Set (siehe Abschnitt 4.4). Dies hat den Nebeneffekt einer Art Verschlüsselung. Existierende Termine sind anhand des Fuzzy-Sets nicht mehr zurückrechenbar (2), da das Fuzzy-Set aus einer Kombination von Terminen und Präferenzen besteht. Der Raum-Agent liefert entsprechend

einen Belegungsplan für den angegebenen Zeitraum. Alle Präferenzen und Termine werden dem Fuzzy-Modul übergeben (3), und es wird ein optimaler Termin berechnet (4). Der resultierende Termin wird den Agenten mitgeteilt (5). Diese tragen ihn in den entsprechenden Terminkalender ein und informieren ihre Nutzer.

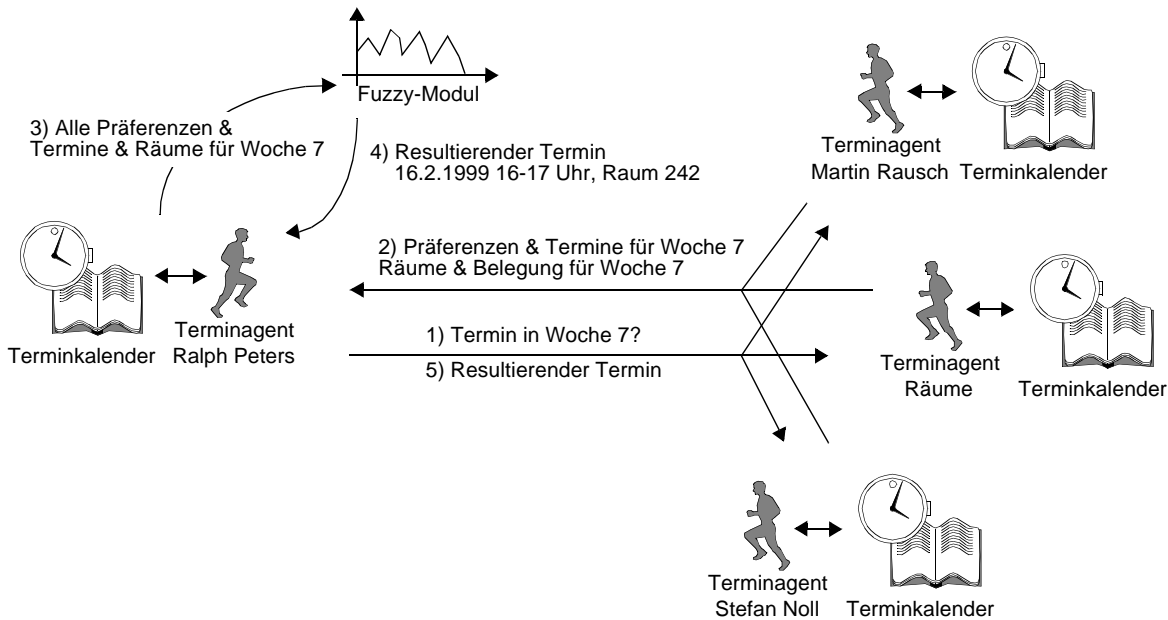


Abbildung 28: Agentenbasierte Terminaushandlung

#### 4.3.6.3 Spontaner Kommunikationsversuch

Um einen spontanen Kommunikationsversuch zu unterstützen, ist eine Kooperation von Termin- und Telekooperations-Agenten notwendig. Der Ablauf eines spontanen Kommunikationsaufbaus ist in Abbildung 29 graphisch dargestellt. Der Kommunikationswunsch wird zunächst vom Telekooperations-Agenten des Initiators an den Agenten des Koopera-

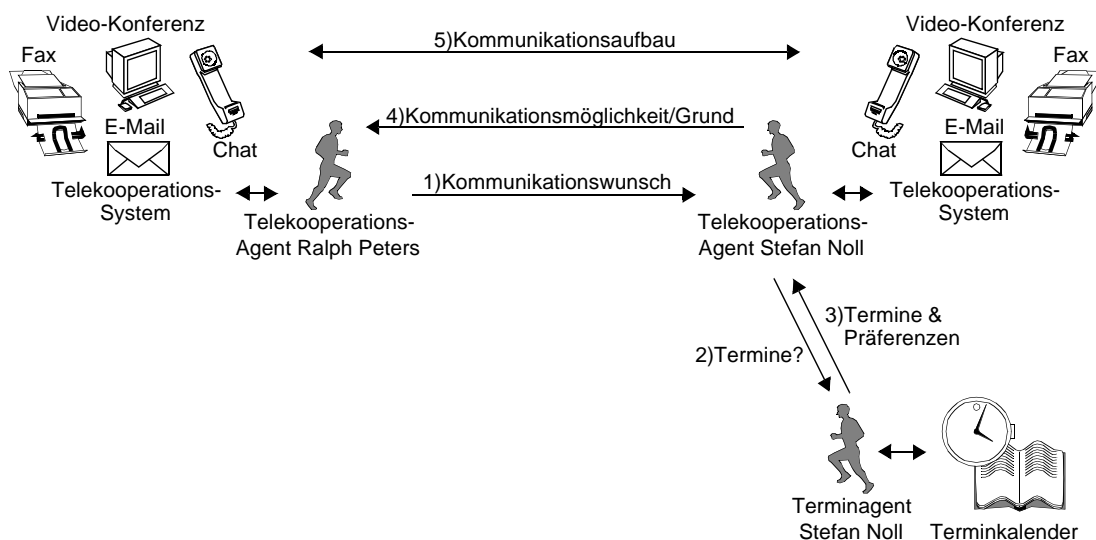


Abbildung 29: Spontane Kommunikation

tionspartners geleitet (1). Dieser überprüft mit Hilfe des Termin-Agenten (2+3), ob derzeit Termine vorliegen und welche Präferenzen sein Nutzer diesbezüglich derzeit hat. Der Telekooperations-Agent meldet dann eine optimale Kommunikationsmöglichkeit und einen Grund für diese zurück (4). Kann keine synchrone Kooperation aufgebaut werden, etwa aufgrund einer Besprechung oder weil der Kooperationspartner auf Dienstreise ist, kann dies dem Initiator transparent gemacht werden. Je nach Möglichkeit wird sonst ein Kommunikationsmedium gewählt und eine Kommunikation aufgebaut (5).

#### **4.3.7 Administrator-Agent**

Hauptaufgabe des Administrator-Agenten ist die Koordination der übrigen Agentenklassen. Er übernimmt damit die in der Basisarchitektur festgelegte Aufgabe des Facilitators. In dem hier verwendeten halb-föderativen Agentenverbund sind Agenten in der Lage, direkt miteinander zu kommunizieren. Dies ist aber nur dann möglich, wenn die Agenten sich bereits kennen. Ist dies nicht der Fall, kontaktieren sie den Administrator-Agenten. Jeder Agent hat sich dazu beim Administrator-Agenten zunächst anzumelden. Der Administrator-Agent kennt dann alle Agenten seiner Domäne mit ihren Fähigkeiten. Bekommt er eine Anfrage zu einem spezifischen Thema, kann er die Vermittlung übernehmen. Administrator-Agenten kommunizieren dazu untereinander. Ein globaler Service erlaubt den initialen Kommunikationsaufbau unter den Administrator-Agenten.

Die Agentenklasse der Administratoren hat zusätzlich die Aufgabe, die anderen Agenten mit Benutzerprofilen zu versorgen. Dazu hat sie Zugriff auf ein Datenrepository. Hiermit werden zwei Anforderungen an den Administrator-Agenten erfüllt. Zum einen können Schablonen von Benutzerprofilen für typische Verwaltungskräfte angefertigt werden. Neu in den Verbund kommende Agenten können diese abfragen und ihren „neuen“ Nutzer von Beginn an möglichst adäquat unterstützen. Der andere Aspekt betrifft die Lebenszeit von Agenten. Im allgemeinen kann nicht davon ausgegangen werden, daß ein Benutzer ständig in einem System arbeitet, respektive angemeldet ist. Mit dem Abmelden vom System sterben seine Agenten, sofern sie nicht zu einem anderen Rechner migrieren. Um dies zu unterstützen, kann ein Agent bei einem Administrator-Agenten das Profil und die darin enthaltenen Präferenzen seines Benutzers ablegen. Der Administrator-Agent kann in diesem Falle sogar als Stellvertreter fungieren. Terminvereinbarungen können so auch in Abwesenheit von Personen durchgeführt werden. Unterstützt wird hierdurch auch die Möglichkeit für einen Benutzer, sich an einem anderen Rechner als bisher anzumelden. Die gestarteten Agenten können beim Administrator das Profil erfragen.

#### **4.3.8 Weitere Groupware Agenten**

Das hier vorgestellte Referenzmodell ist offen für weitere Groupware Agenten. Durch Anwendung des gleichen Prinzips können zusätzliche Groupware-Komponenten angebunden werden. Voraussetzung ist das Vorhandensein einer Schnittstelle (API) zu dem entsprechenden System.

Für die Anbindung neuer Agenten ist dabei der Nutzen für den Anwender in den Vordergrund zu stellen. Wesentlicher Aspekt ist die Erweiterung der Arbeits- und Kooperations-

möglichkeiten in der Gruppe und die Konzentration auf die zu lösenden Aufgabengebiete. Dieser Punkt wird im Ausblick nochmals aufgegriffen.

Eine sinnvolle Ergänzung sind Agenten zur Informationsbeschaffung. Agenten, die an ein Dokumentenmanagement- oder Archivierungs-System angeschlossen sind, können für einen bestimmten Sachverhalt in Kombination mit Termin- oder Workflow-Agent beispielsweise genutzt werden, um ein Meeting vorzubereiten. Mit einem zusätzlichen Agenten, der einen Anschluß an das World-Wide-Web besitzt, ist die Beschaffung von korrelierenden Informationen möglich. Weitere Agenten könnten etwa *Group Decision Support* oder *Meeting-Support*-Systeme anbinden, um synchrone oder asynchrone Entscheidungsprozesse zu unterstützen.

## 4.4 Entscheidungsfindung – Autonomie für Agenten

### 4.4.1 Einführung

Eine der wesentlichen Eigenschaften von Software Agenten ist Autonomie. Agenten müssen ohne die direkte Intervention von Menschen oder anderen Entitäten operieren können und müssen ein gewisses Maß an Kontrolle über ihre Aktivitäten und ihren Status haben. Agenten sollten dabei nicht einfach nur auf eine Veränderung der Umgebung reagieren, sondern fähig sein, dies in einer zielgerichteten Art zu tun, indem sie eigenständig die Initiative ergreifen.

Die Verwendung eines Expertensystems – diese Systeme können durch Definition entsprechender Entscheidungsregeln für die angesprochene Problematik adaptiert werden – scheint zunächst ein gangbarer Weg. Doch sind diese Systeme für die hier gestellte Aufgabe überqualifiziert und verbrauchen zu viele Systemressourcen. Die verwendeten Agenten agieren oft in einer quasi „Echtzeit Situation“ (z. B. Telekommunikations-Agent). Entscheidungen müssen „schnell“ getroffen werden.

Eine bessere Möglichkeit zur Lösung des Problems stammt von *Berger, M.O. und Kubitz O* [20], die einen autonomen Agenten in Form eines autonomen Fahrzeuges entwickelten. Dieses Fahrzeug steuert vorgegebene Zielpunkte an und vermeidet dabei selbständig eine Kollision mit auftretenden Hindernissen. Die Kontrollaktionen des Fahrzeuges auf die durch Sensoren wahrgenommenen Umweltinformationen wurden mittels Fuzzy-Logik gesteuert.

Besonders interessant an diesem autonomen Agenten ist die Form seiner Steuerung durch die „unscharfe“ Fuzzy-Logik. Dieses Beispiel sowie weitere Artikel wie etwa [77], [143] oder [145] zeigen die Möglichkeiten und Vorteile der „unscharfen Logik“ gegenüber anderen regelbasierten Systemen auf. Im Vergleich zu Expertensystemen sind Fuzzy-Regler kleiner, schneller und verbrauchen weniger Systemressourcen.

Der vielleicht größte Vorteil der Fuzzy-Logik ist die Möglichkeit der linguistischen Approximation von menschlichem Wissen und Erfahrungen. Linguistische Variablen können ver-



wendet werden, um umgangssprachliche Attribute in einer mathematisch exakten Form zu erfassen.

Aufgrund dieser Vorteile eignet sich ein Fuzzy-Logik basiertes System am besten, um die Agenten mit Autonomie auszustatten. Die Agentenföderation erhält somit eine zusätzliche Möglichkeit, im positiven Sinne für seinen Benutzerverbund zu agieren.

Im folgenden wird zunächst ein Exkurs zur Fuzzy-Logik einen Einblick in die unscharfe Mathematik geben. Diese dient dem anschließend im Abschnitt 4.4.3 beschriebenen Modul zur Realisierung von Entscheidungsfindungsprozessen für die im Referenzmodell spezifizierten Software-Agenten als Basis.

## 4.4.2 Exkurs Fuzzy-Logik

### 4.4.2.1 Einleitung

Lofti A. Zadeh, Professor für Computerwissenschaften an der Universität Berkeley in Kalifornien, CA, entwickelte in den sechziger Jahren die Theorie der unscharfen Logik (fuzzy logic) [164]. Sie erweitert die klassischen scharfen Zustände um Zwischenstufen, was unserem menschlichen Denken und Empfinden eher entspricht. Die Fuzzy-Logik ermöglicht es vage, toleranzbehaftete Aussagen mathematisch zu beschreiben.

Da man Unschärfe mit Ungenauigkeit gleichsetzte, was mit Computern als unvereinbar galt, wurde erst Mitte der siebziger Jahre die Theorie der Fuzzy-Logik in die Praxis umgesetzt. Die erste kommerzielle Anwendung der Fuzzy-Logik wurde 1974 zur Steuerung eines Zementofens der Firma F.L. Smidth & Co in Kopenhagen benutzt (vergleiche [145]). Der endgültige Durchbruch der Fuzzy-Technologie ist jedoch auf das Engagement der Japanischen Industrie zurückzuführen, die Ende der achtziger Jahre erfolgreich auf Fuzzy-Logik basierende Konsumelektronik auf den Markt brachte. In Europa und den USA kamen solche Produkte erst in den letzten Jahren auf den Markt. Man hat die Möglichkeiten, die die Fuzzy-Logik bietet, erkannt und wendet sie inzwischen in allen Bereichen der Steuer- und Regelungstechnik an.

### 4.4.2.2 Unscharfe Mengen

Wesentlicher Bestandteil der Fuzzy-Logik ist das Konzept der unscharfen Mengen, welche eine echte Verallgemeinerung der gewöhnlichen scharfen Mengen ist (die folgenden Definitionen sind [143], [145] bzw. [91] entnommen):

Sei  $\Xi$  eine scharfe Menge. Eine *unscharfe Menge* (englisch *fuzzy-set*)  $A$  über  $\Xi$  wird charakterisiert durch eine Zugehörigkeitsfunktion  $\mu_A(\Xi)$ , die jedem Element aus  $\Xi$  eine reelle Zahl aus dem Intervall  $[0, 1]$  zuordnet. Der Wert von  $\mu_A$  an der Stelle  $\xi$  wird als *Zugehörigkeitsgrad* von  $\xi$  zur Menge  $A$  bezeichnet.

Je größer der Wert von  $\mu_A(\xi)$  ist, desto höher ist der Zugehörigkeitsgrad von  $\xi$  zu  $A$ . Für unscharfe Mengen ist  $\mu_A$  eine *verallgemeinerte charakteristische* Funktion, die für jedes

Element  $\xi \in \Xi$  stufenlos zwischen Nichtzugehörigkeit und Zugehörigkeit von  $\xi$  zu  $A$  entscheidet (siehe Abbildung 30).

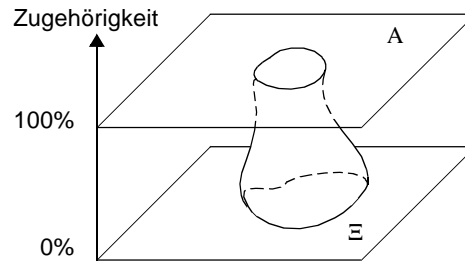


Abbildung 30: Zugehörigkeit der Elemente aus der Menge  $\Xi$  zur unscharfen Menge  $A$

Als Notation unscharfer Mengen hat sich als gebräuchlichste Form die Aufzählung aller Zwei-Tupel, die eine unscharfe Menge  $A$  eindeutig charakterisieren, durchgesetzt:

$$A = \{(\xi_1, \mu_A(\xi_1)), \dots, (\xi_n, \mu_A(\xi_n))\}$$

zum Beispiel:  $A = \{(10, 0.1), (11, 0.2), (30, 0.4)\}$ .

Es entspricht der allgemeinen Konvention, Elemente mit Zugehörigkeitsgrad Null bei dieser Notation wegzulassen.

#### 4.4.2.3 Darstellungsformen unscharfer Mengen

Man unterscheidet zwischen parametrisierter und diskreter Darstellung der Zugehörigkeitsfunktion  $\mu_A(\xi)$  von unscharfen Mengen. Eine parametrisierte Darstellung bietet sich immer an, wenn die Grundmenge sehr viele Elemente besitzt oder ein Kontinuum ist (z. B. Druck, Temperatur, Geschwindigkeit...). Eine diskrete Darstellung ist dann zu wählen, wenn die Grundmenge aus endlich vielen Elementen mit explizitem Zugehörigkeitsgrad besteht.

Beispiele für Zugehörigkeitsfunktionen sind Dreiecks-, Trapez- oder S-Funktion und Z-Funktion. Die entsprechenden Definitionen können [91] entnommen werden. Eine allgemeine Darstellung erlauben generelle Fuzzy-Sets:

Ein *genereller* Fuzzy-Set  $\Gamma$  wird repräsentiert durch eine Menge von Zwei-Tupeln

$$\Gamma := \{(\xi_1, \psi_1), \dots, (\xi_n, \psi_n)\}$$

wobei  $(\xi_i, \psi_i)$ ,  $i = 1, \dots, n$ , jeweils Punkte sind, an denen sich die Steigung der Zugehörigkeitsfunktion ändert. Zwischen jeweils zwei benachbarten Punkten wird die Funktion durch eine Gerade approximiert.

Weitere Bedingungen für ein generelles Fuzzy-Set sind die streng aufsteigende Anordnung der Grundmengenelemente sowie die Beschränkung der  $\psi$ -Werte auf das Einheitsintervall.

Es gilt:  $\xi_i < \xi_j$ ,  $\forall i < j$ .  $\xi_i, \xi_j \in \Gamma$  und  $0 \leq \psi_i \leq \psi_n \forall i \in \{1, \dots, n\}$

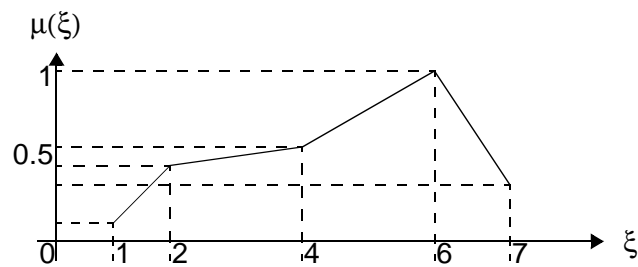


Abbildung 31: Beispiel eines generellen Fuzzy-Set

Abbildung 31 zeigt eine durch  $\{(1,0.1), (2,0.4), (4,0.5), (6,1.0), (7,0.3)\}$  definierte unscharfe Menge.

Mit dem Konzept des generellen Fuzzy-Sets kann in Abhängigkeit der Anzahl der Stützstellen jede stetige Zugehörigkeitsfunktion mit beliebiger Genauigkeit approximiert werden.

#### 4.4.2.4 Linguistische Variablen

Die *linguistischen Variablen* sind eine besondere Form unscharfer Mengen. Die Werte dieser Variablen sind Worte und Ausdrücke einer natürlichen oder künstlichen Sprache. Linguistisch ausgedrücktes Wissen kann so mit all seinen Unschärfen angemessen repräsentiert und mit einem Computer verarbeitet werden. Der formale Ansatz für die Beschreibung von linguistischen Variablen wurde von Zadeh [165] entwickelt:

Eine *linguistische Variable* wird charakterisiert durch ein Fünf-Tupel  $(\xi, T(\xi), Y, \Gamma, M)$ , wobei  $\xi$  der Name der linguistischen Variablen ist.  $T(\xi)$  oder einfach  $T$  bezeichnet den *Term-Set* von  $\xi$ , das heißt: die Menge der Namen der *linguistischen* Werte von  $\xi$ , wobei jeder linguistische Wert (hier als  $\Xi$  bezeichnet) eine unscharfe Menge über dem Grundbereich  $Y$  mit der entsprechenden *Basis-Variablen*  $v$  ist.  $\Gamma$  ist eine *syntaktische Regel* (normalerweise in Form einer kontextfreien Grammatik) und dient zur Generierung der Namen  $X$  der linguistischen Werte von  $\xi$ , das heißt: des Term-Sets  $T(\xi)$ ;  $M$  ist eine *semantische Regel*, die jedem  $\Xi$  eine bestimmte Bedeutung  $M(\xi)$  zuordnet, wobei  $M(\xi)$  eine unscharfe Teilmenge über dem Grundbereich  $Y$  ist. Ein einzelnes  $\Xi$ , also ein durch  $\Gamma$  erzeugter Name einer unscharfen Menge, wird als *Term* bezeichnet. Ein Term, der lediglich ein Wort oder mehrere Wörter, die zusammen als eine Einheit betrachtet werden (z.B. wenn sie immer gemeinsam auftreten), enthält, wird als *atomarer Term* bezeichnet. Ein Term, der aus einem oder mehreren atomaren Termen besteht, ist ein *zusammengesetzter Term*. Die Konkatenation der Komponenten eines zusammengesetzten Terms ist ein *Sub-Term*. Wenn  $\Xi_1, \Xi_2, \dots$  Terme in  $T$  sind, dann kann  $T$  ausgedrückt werden als die Vereinigung

$$T = \Xi_1 + \Xi_2 + \dots$$

Folgendes Beispiel soll die Definition verdeutlichen: Sei  $\Xi$  eine linguistische Variable *Alter*, daß heißt:  $\xi = \text{Alter}$ , mit dem Grundbereich  $Y = [0, 100]$  gegeben. Die *Basis*-Variable  $v$  ist das Alter in Lebensjahren. Ein linguistischer Wert von *Alter* könnte *jung* oder auch *alt* sein, wobei *jung* und *alt* atomare Terme wären. Der Wert *sehr alt* wäre in diesem Fall ein zusammengesetzter Term, der *alt* als atomare Komponente und *sehr* und *alt* als Sub-Terme enthält. Ein weiterer Wert könnte zum Beispiel *mehr oder weniger jung* sein. Das Term-Set von *Alter* könnte also wie folgt notiert werden:

$$T(\text{Alter}) = \text{alt} + \text{sehr alt} + \text{nicht alt} + \text{mehr oder weniger jung} + \text{ziemlich jung} + \dots$$

Dabei ist jeder Term der Name einer unscharfen Menge über der Grundmenge  $Y = [0, 100]$ .

$M(\xi)$  ist eine Vorschrift, die jedem Term der linguistischen Variablen eine bestimmte Bedeutung zuweist. Mathematisch formuliert ist der Wert *alt* zum Beispiel gegeben durch

$$M(\text{alt}) = \{(v, \mu_{\text{alt}}(v)) \mid 0 \leq v \leq 100\}$$

mit

$$\mu_{\text{alt}}(u) = \begin{cases} 0 & 0 \leq u < 50 \\ \left( \left( 1 + \left( \frac{u-50}{5} \right)^2 \right)^{-1} \right) & 50 \leq u \leq 100 \end{cases}$$

#### 4.4.2.5 Mengenoperationen auf linguistischen Variablen

Für gewöhnliche Mengen kennt man als grundlegende Operationen die Bildung des Durchschnittes und der Vereinigung sowie die Bildung des Komplements. Für unscharfe Mengen sind Verknüpfungen dieser Form nicht geeignet, da sie lediglich binäre Operatoren verknüpfen können. Für unscharfe Mengen wird daher eine mehrwertige Logik definiert, in der die logischen Konnektoren *und*, *oder* und *nicht* als Funktionen aufgefaßt werden. Zusätzlich genügen *und*- bzw. *oder*-Funktion *t*-Normen bzw. *t*-Conormen, wodurch Monotonie, Symmetrie und Assoziativität gewährleistet wird (vergleiche [84] und [91]).

Der Durchschnitt zweier unscharfer Mengen  $A$  und  $B$  mit den Zugehörigkeitsfunktionen  $\mu_A(\xi)$  und  $\mu_B(\xi)$  ist die unscharfe Menge  $X$ , in Zeichen  $X = A \cap B$ , die punktweise definiert ist durch die Zugehörigkeitsfunktion:

$$\mu_X(\xi) := \min(\mu_A(\xi), \mu_B(\xi)), \forall \xi \in \Xi$$

Die Vereinigung zweier unscharfer Mengen  $A$  und  $B$  mit den Zugehörigkeitsfunktionen  $\mu_A(\xi)$  und  $\mu_B(\xi)$  ist die unscharfe Menge  $X$ , in Zeichen  $X = A \cup B$ , die punktweise definiert ist durch die Zugehörigkeitsfunktion:

$$\mu_X(\xi) := \max(\mu_A(\xi), \mu_B(\xi)), \forall \xi \in \Xi$$

Das Komplement einer unscharfer Menge  $A$  mit den Zugehörigkeitsfunktion  $\mu_A(\xi)$  ist die unscharfe Menge  $A'$ , in Zeichen  $A' = \neg A$ , die punktweise definiert ist durch die Zugehörigkeitsfunktion:

$$\mu_{A'}(\xi) := 1 - \mu_A(\xi), \forall \xi \in \Xi$$

Bei der Komplementbildung ist besonders die Grundmenge zu beachten, denn zur neuen Menge gehören nur die Zugehörigkeitsgrade der Elemente, die zur Grundmenge gehören.

Die drei Mengenoperationen Durchschnitt, Vereinigung und Komplementbildung sind in Abbildung 32 graphisch dargestellt.

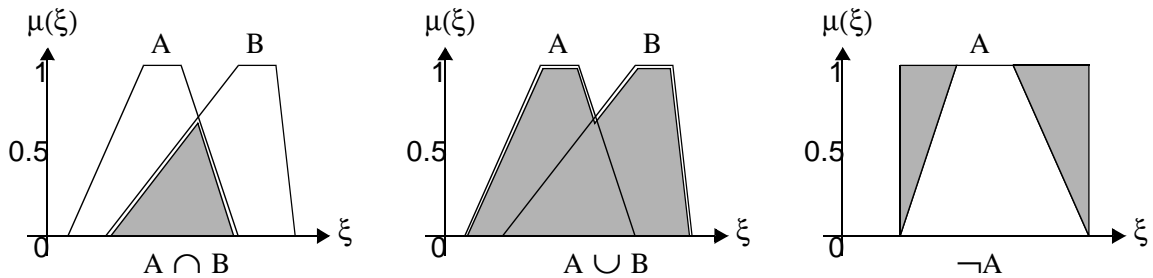


Abbildung 32: Durchschnitt, Vereinigung und Komplementbildung unscharfer Mengen

Neben Minimum- und Maximum-Operator existieren weitere Operatoren zur Aggregation unscharfer Mengen. Einige bekannte sind das Hamacher-, das Algebraische-, das Einstein-, das Schweizer- sowie das Drastic-Produkt bzw. Summe. Die Definitionen dieser und weiterer Operatoren finden sich beispielsweise in [91] oder [143].

Modifikatoren bieten eine weitere Möglichkeit Fuzzy-Sets oder linguistische Variablen mathematisch zu transformieren und das Ergebnis sprachlich zu interpretieren. Ein Modifikator manipuliert als unärer Operator genau eine unscharfe Menge. Wie sprachliche Modifikatoren, etwa *nicht*, *mehr oder weniger* oder *sehr*, mathematisch realisiert werden sollen, ist weitgehend ungeklärt.

Für den Modifikator *nicht* kann die Komplementbildung angenommen werden. Für die Modifikatoren *mehr oder weniger* und *sehr* können folgende Definitionen aus [91] übernommen werden:

$$\begin{aligned} \mu_{\text{mehr-oder-weniger}}(\xi) &= \sqrt{\mu(\xi)} \quad , \quad \forall \xi \in \Xi \\ \mu_{\text{sehr}}(\xi) &= \mu(\xi)^2 \quad , \quad \forall \xi \in \Xi \end{aligned}$$

Die folgende Abbildung soll das Verhalten dieser Operatoren angewendet auf die Ausprägung *alt* verdeutlichen.

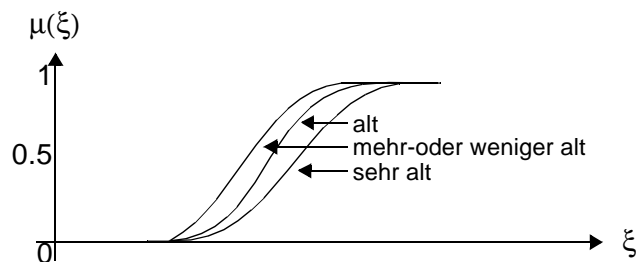


Abbildung 33: Verhalten verschiedener Modifikatoren

### 4.4.3 Fuzzy-Logik-basierte Entscheidungsfindung

Autonomie ist eine der wichtigsten Eigenschaften von Software Agenten. Voraussetzung dafür ist, Mechanismen zur Verfügung zu stellen, die einem Agentenverbund das Aushandeln von Aktionen und eine damit verbundene Entscheidung ermöglichen. Um eine möglichst effektive Methode für das Referenzmodell zu erhalten, ist sich für den Einsatz der Fuzzy-Logik entschieden worden. Die regelbasierte unscharfe Mathematik erlaubt die Implementierung von schnellen und auf die Situation abgestimmten Entscheidungsfindungsprozessen, die dem Anwender eine optimale Unterstützung in unterschiedlichen Konstellationen und Sachverhalten gibt.

Zur Realisierung des Fuzzy-Logik basierten Entscheidungsmoduls wird ein Fuzzy-Controller (FC) spezifiziert. Anhand der Abbildung 34 kann der prinzipielle Aufbau eines regelbasierten Fuzzy-Controllers studiert werden. Der Controller verfügt über ein Fuzzifizierungs-Modul, eine Regelbasis, eine Inferenzmaschine und ein Defuzzifizierungs-Modul.

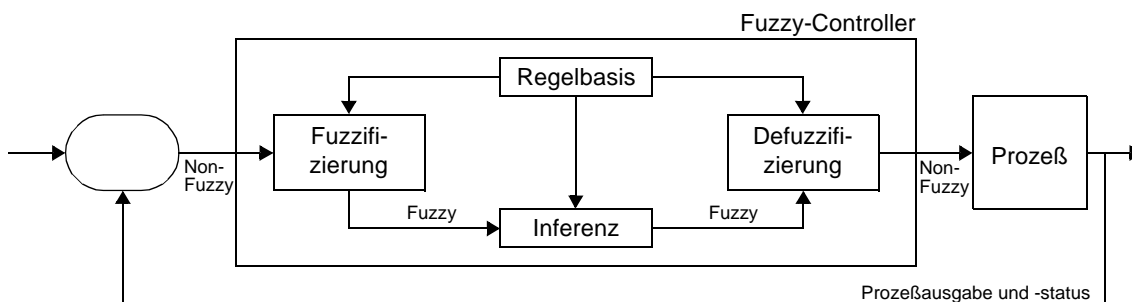


Abbildung 34: Allgemeine Struktur eines Regelkreises mit Fuzzy-Control nach [102]

Der Entwurf eines Reglers kann dabei in folgende Schritte unterteilt werden:

- Festlegen der Eingangs- und Ausgangsvariablen
- Erstellen einer Regelbasis
- Festlegen der Inferenzstrategie
- Berechnung der *scharfen* Ausgangsgrößen

Um den Prozeß der Entscheidungsfindung innerhalb des Agentenverbundes zu spezifizieren werden in den nächsten Abschnitten diese Schritte ausführlich beschrieben und anhand des Beispiels der Terminvereinbarung diskutiert.

#### 4.4.3.1 Fuzzifizierung

Am Beginn des Entwurfsprozesses steht die Systemdefinition der gewünschten Regelfunktionen mit den Ein- und Ausgangsvariablen.

Die Variablen sind nach dem Prinzip der Fuzzy-Logik als linguistische Variablen aufzufassen, deren Wertebereiche und physikalische Einheiten festzulegen sind. Im nächsten Schritt

muß für jede Variable die Anzahl der Ausprägungen (Fuzzy-Sets) und deren verbale Charakterisierung festgelegt werden.

Sehr wichtig bei dieser Modellierung der Variablen ist die quantitative Umsetzung der verbalen Begriffe. Für jedes Fuzzy-Set ist die Form der Zugehörigkeitsfunktion festzulegen. Aufgrund der effizienten Berechenbarkeit werden hierfür häufig trianguläre Fuzzy-Sets verwendet. Eine generelle Empfehlung kann dafür jedoch nicht ausgesprochen werden. Das Festlegen der Zugehörigkeitsfunktion bleibt ein geistig-kreativer Prozeß, der durch „try and error“ solange fortgesetzt werden muß, bis der Fuzzy-Controller optimal arbeitet. Von Vorteil ist es oft, die Fuzzy-Sets einer Variablen leicht überlappen zu lassen, weil dann „Lücken“ im Reglerverhalten vermieden werden [96].

Das Fuzzifizierungs-Modul hat nun die Aufgaben aus den *scharfen* Eingangsgrößen die entsprechenden Zugehörigkeitsgrade zu den Fuzzy-Sets zu ermitteln. Das heißt, es wird eine linguistische Interpretation der technischen Größen durchgeführt. Für die Terminplanung soll dies nun beispielhaft durchgeführt werden.

Zu Beginn einer Terminplanung muß der Zeitraum spezifiziert werden, in dem der Termin liegen soll. Dieser wird durch ein Beginndatum ( $b_t$ ) und ein Enddatum ( $e_t$ ) festgelegt. Weiterhin muß die Termindauer ( $l_t$ ) angegeben werden. Die kleinste zeitliche Einheit beträgt  $m$  Minuten. Für die weiteren Berechnungen wird der Terminzeitraum durch eine bilineare Funktion  $f$  in die Menge der positiven ganzen Zahlen  $Z^+$  abgebildet, wobei gilt:

$$f(\tau) = \text{Minuten } (b_t, \tau) / m \quad b_t \leq \tau \leq e_t$$

$$\text{Minuten } (\tau_1, \tau_2) = \text{Anzahl Minuten im Zeitintervall } [\tau_1, \tau_2]$$

Die Umkehrung dieser Funktion  $f$  ergibt für eine Zahl  $\xi$  aus dem Intervall  $[f(b_t), f(e_t)]$  wieder das korrespondierende Datum  $\tau$ . Diese Funktion wird im folgenden  $T$  genannt und wird definiert durch:

$$T(\xi) = \text{Datum } (b_t, m\xi) \quad f(b_t) \leq \xi \leq f(e_t)$$

$$\text{Datum } (\tau, \xi) = \tau + \xi \text{ Minuten} \quad \tau \in \text{Datum}, \xi \in Z^+$$

Das geschlossene Intervall  $[f(b_t), f(e_t)]$  ist somit der Definitionsbereich der Fuzzy-Variablen.

Als Ausprägung besitzt die Fuzzy-Variable eine Ausprägung *Zeit*, welche die Terminmöglichkeiten repräsentiert. Für jede an der Terminplanung beteiligten Person wird eine solche Variable erstellt. Diese Terminmöglichkeiten ergeben sich zum einen aus den bereits bestehenden Terminen, welche wie in Abschnitt 4.3.6 beschrieben aus dem persönlichen Kalender ausgelesen werden können, und zum anderen aus den persönlichen Präferenzen für bestimmte Tageszeiten. Diese Präferenzen werden vom Benutzer in einem bestimmten Takt, zum Beispiel 30 Minuten pro Einheit, für jeden Wochentag definiert. Dabei kann zwischen unterschiedlichen Werten, etwa „Bitte kein Termin“, „Termin unter Umständen möglich“, „Termin möglich“ und „Bevorzugter Termin“, gewählt werden.

Realisiert wird die Ausprägung *Zeit* durch ein normiertes Fuzzy-Array über dem gesamten Definitionsbereich der Fuzzy-Variablen.

Die Fuzzyifizierung der Terminmöglichkeiten und damit der Aufbau des Fuzzy-Arrays *Zeit* erfolgt in drei Schritten:

1. Übertragen der persönlichen Vorlieben:

Für jeden Punkt  $\xi$  des Fuzzy-Arrays *Zeit* wird mit der Funktion  $T(\xi)$  das korrespondierende Datum errechnet und mit einer Hilfsfunktion der Wochentag und die Uhrzeit bestimmt. Mit diesen beiden Werten kann aus den Präferenzen der entsprechende Wert gelesen und auf das Fuzzy-Array an der Stelle  $\xi$  abgebildet werden. Dabei werden die linguistischen Begriffe wie folgt ersetzt:

<i>Bitte kein Termin</i>	= 0.0
<i>Termin unter Umständen möglich</i>	= 0.5
<i>Termin möglich</i>	= 0.75
<i>Bevorzugter Zeitpunkt für Termin</i>	= 1.0

2. Übertragen der Kalenderdaten

Im 2. Schritt werden nun die schon bestehenden Termine für den spezifizierten Zeitraum aus dem persönliche Kalender ausgelesen. Für diese Termine wird dem korrespondierenden Bereich des Fuzzy-Arrays *Zeit* der Wert 0.0 = *keine Zeit* zugewiesen.

Das bis hierher entstandene Fuzzy-Array zeigt nun alle freien Termine (Wert an der Stelle  $f(\tau) > 0$ ) im spezifizierten Zeitraum an.

3. Eliminieren der zu kleinen Zeiträume

Im letzten Schritt wird nun die gewünschte Terminlänge  $l_t$  berücksichtigt.

Damit ein Termin vereinbart werden kann, muß es mindestens einen Bereich der Größe  $f(l_t)$  im Fuzzy-Array *Zeit* geben für den gilt  $Zeit(\xi) > 0$  für alle  $\xi$  aus diesem Bereich.

Weiterhin muß gelten:

Gibt es einen Bereich  $A = [a, b]$  im Fuzzy-Array *Zeit* für den gilt

$$Zeit(\xi) > 0 \quad \forall \xi \in A \quad \text{und} \quad |A| > f(l_t)$$

so liegt der letzte mögliche Termin im Bereich  $A$  an der Stelle  $b - f(l_t) + 1$ .

Aus diesen beiden Forderungen leitet sich nun folgende Vorgehensweise ab:

Ausgehend von der Stelle  $f(e_t)$  wird das Fuzzy-Array *Zeit* bis zur Stelle  $f(b_t)$  (quasi von rechts) nach allen Bereichen durchsucht, für die gilt  $Zeit(\xi) > 0$ .

Ist die Größe eines gefundenen Bereiches kleiner als  $f(l_t)$  so wird er eliminiert, d.h. die Werte von *Zeit* werden für diesen Bereich auf 0.0 gesetzt.

Ist die Größe eines gefundenen Bereiches  $A = [a, b]$  größer oder gleich  $f(l_t)$  so wird die Größe dieses Bereiches um  $f(l_t) - 1$  von rechts her verkleinert, d.h.

$$Zeit(\xi) = 0.0 \quad \forall \xi \in [b - f(l_t) + 1, b]$$



#### 4.4.3.2 Regelbasis

Die Regelbasis enthält in Form von Regeln das Expertenwissen über den zu modellierenden Prozeß. Am häufigsten werden dafür Produktionsregeln verwendet, da diese natürlichsprachlichen „Wenn-Dann-Regeln“ gegenüber anderen Repräsentationsformen, wie Semantische Netze, Logik oder objektorientierte Wissensrepräsentation einige Vorteile aufweisen:

- *Erweiterbarkeit*: Neue Regeln können unabhängig von anderen Regeln der Wissensbasis hinzugefügt werden.
- *Modularität*: Jede Regel definiert ein kleines, relativ unabhängiges Stück Information.
- *Modifizierbarkeit*: Alte Regeln können relativ unabhängig von anderen Regeln erneuert werden.
- *Verständlichkeit*: Leichte Nachvollziehbarkeit der Wirkungsweise des Fuzzy-Systems.
- *Transparenz*: Das regelbasierende System kann unter Umständen selbsterklärend sein.

Die Regelbasis enthält alle Wenn-Dann-Regeln mit ihren Klauseln und logischen Konnektoren in den Prämissen und Konklusionen. Die Prämissen enthalten entweder eine oder mehrere Klauseln; die Konklusion unterstützt jeweils nur eine Klausel. Die Form einer jeden Klausel ist: „*Variable ist Attribut*“, wobei *Variable* eine linguistische Variable repräsentiert. Jedes Attribut einer Variablen ist ein Fuzzy-Set, welches auch modifiziert sein kann. Als Modifikatoren sind *sehr*, *mehr oder weniger*, *nicht* und andere möglich.

Die Beschreibung einer Regelbasis erfolgt gewöhnlich durch eine kontextfreie Grammatik. Als Beispiel wird nun eine vollständige Regelbasis in Anlehnung an die Backus-Naur-Form dargestellt:

```

Wenn | hat | dann | Sicherheit | Variable | Fuzzy-Set |
Modifizierer | und | oder

Regelbasis      : Regel
                  | Regel oder Regel

Regel           : Wenn Prämissen dann Konklusion
                  | Wenn Prämissen dann Konklusion Sicherheit

Prämisse       : Klausel
                  | Prämisse und Klausel

Konklusion     : Klausel

Klausel        : Variable hat Attribut

Attribut       : Fuzzy-Set
                  | Modifizierer Fuzzy-Set

```

Jeder Regel kann ein Sicherheitsfaktor (s.u.) als reelle Zahl aus dem Intervall  $[0,1]$  als stufenloser Ausdruck für das Vertrauen in die Gültigkeit der Regel zugeordnet werden. *Null* bedeutet dabei kein Vertrauen, *Eins* bedeutet volles Vertrauen. Dieses Vertrauensmaß ist als Zugehörigkeitsgrad einer Regel zur Menge der absolut gültigen Regeln interpretierbar.

Für den Fall der Terminplanung existiert für jede beteiligte Person eine Fuzzy-Variable. Die Variable des Initiators wird mit *Host* und die der anderen Personen mit *User1*, *User2* usw. benannt. Weiterhin gibt es eine Ergebnisvariable *Termine* vom gleichen Typ wie die *User*-Variablen.

Das im vorherigen Abschnitt bestimmte Fuzzy-Set *Zeit* kann auch über eine Regel bestimmt werden. Der Modifikator *nicht* wird durch Komplementbildung berechnet:

```
Wenn User hat Präferenz und
User hat nicht Termin dann
Zeit hat Zeit
```

Diese Regel läßt sich durch die Wichtung der eigenen Termine erweitern. Dazu wird, korrespondierend zu den Präferenzen der freien Termine, die Wichtigkeit eines Termins für eine Person bestimmt. Zunächst wird ein Basiswert für einen Termin festgelegt.

$$\text{Angenommener Termin} = 0.3$$

Dieser kann dann mit Modifikatoren wie etwa *mehr oder weniger wichtig* und *sehr wichtig* durch den Anwender angepaßt werden (der Modifikator *nicht* aus der obigen Regel entfällt). Für die Terminplanung können die in Abschnitt 4.4.2.5 eingeführten Modifikatoren übernommen werden. Die Interpretation ist wie folgt: je wichtiger ein Termin ist desto geringer ist der Wert an der Stelle und desto unwahrscheinlicher wird eine erneute Terminvereinbarung zu diesem Zeitpunkt:

$$\begin{aligned} \mu_{\text{mehr-oder-weniger-wichtig}}(\xi) &= \sqrt{\mu(\xi)} \quad , \forall \xi \in \Xi \\ \mu_{\text{sehr-wichtig}}(\xi) &= \mu(\xi)^2 \quad , \forall \xi \in \Xi \end{aligned}$$

Die Ausprägung *Zeit* stellt jetzt die gewichteten Terminmöglichkeiten der jeweiligen Personen dar. Der Initiator der Terminplanung besitzt eine weitere Möglichkeit die Ausprägung der Fuzzy-Variablen zu beeinflussen: *Dringlichkeit*. Diese linguistische Variable eröffnet innerhalb des gewählten Zeitraums die Möglichkeit den Wunschtermin genauer zu spezifizieren.

Für die Terminplanung wird folgende Regel definiert, die sukzessive auf den gesamten Definitionsbereich der Fuzzy-Variablen angewandt wird:

```
Wenn Host hat Dringlichkeit Zeit und
User1 hat Dringlichkeit Zeit und
User2 hat Dringlichkeit Zeit und
...
UserN hat Dringlichkeit Zeit dann
Termin hat Zeit
```

Zur Berechnung kann der Modifikator *Dringlichkeit* ausgeklammert werden, da zur Verknüpfung der einzelnen Fuzzy-Variablen der Minimum-Operator verwendet wird. Ein Ter-

min ist nur dann möglich, wenn alle beteiligten Personen auch Zeit haben d.h. die Ausprägung aller Fuzzy-Arrays *Zeit* an dieser Stelle einen Wert größer 0.0 haben.

Jeder Punkt  $\xi$  der Ergebnisvariable enthält nach der Fuzzy-Berechnung einen Wert für die Terminmöglichkeit an dieser Stelle  $\xi$  bzw. zum Zeitpunkt  $T(\xi)$ . Dieser Wert ist um so höher, je größer die Übereinstimmung der einzelnen Variablen ist.

Die Dringlichkeit wird durch eine nicht notwendigerweise symmetrische Glockenkurve repräsentiert. Der Definitionsbereich der Glockenkurvenfunktion entspricht dem gewählten Zeitraum und ergibt sich aus dem Intervall  $[0, f(\text{Enddatum})]$ .

Das Maximum der Kurve läßt sich über den gesamten Zeitraum verschieben und spezifiziert so den Wunschzeitpunkt ( $w_t$ ).

Die linke Seite der Glockenkurve wird durch eine S-Funktion repräsentiert, deren Minimum bei  $f(b_t)$ , deren Maximum bei  $f(w_t)$  und deren Wendepunkt auf

$$a = \frac{f(w_t) - f(b_t)}{2}$$

liegt:

$$S(\mathbf{x}, a) = \begin{cases} 2\left(\frac{\mathbf{x}}{2a}\right)^2 & 0 \leq \mathbf{x} \leq a \\ 1 - 2\left(1 - \frac{\mathbf{x}}{2a}\right) & a < \mathbf{x} \leq f(w_t) \end{cases}$$

Die rechte Seite der Glockenkurve wird analog durch eine Z-Funktion repräsentiert. Ihr Wendepunkt liegt auf

$$a = \frac{f(e_t) - f(w_t)}{2}$$

Ihr Maximum liegt entsprechend bei  $f(w_t)$  und ihr Minimum bei  $f(e_t)$ :

$$Z(\mathbf{x}, a) = 1 - S(\mathbf{x}, a)$$

Die Dringlichkeitsfunktion  $\Delta(\xi)$  ist also wie folgt definiert:

$$D(\mathbf{x}) = \begin{cases} S\left(\mathbf{x}, \frac{f(w_t) - f(b_t)}{2}\right) & f(b_t) \leq \mathbf{x} \leq f(w_t) \\ Z\left(\mathbf{x}, \frac{f(e_t) - f(w_t)}{2}\right) & f(w_t) < \mathbf{x} \leq f(e_t) \end{cases}$$

#### 4.4.3.3 Inferenz-Modul

Das Inferenz-Modul hat die Aufgabe aus den Regeln des Regelsystems und den Eingangsfakten Schlüsse zu ziehen. Für die folgenden Erklärungen des Inferenz-Mechanismus sind einige Annahmen zu treffen:

Die Regelmenge des Fuzzy-Systems verfügt über Eingangsvariablen  $\Xi_1, \dots, \Xi_n$  und eine Ausgangsvariable  $\Psi$ . Jeder dieser Variablen wird repräsentiert durch eine linguistische Variable, deren Ausprägungen  $A_{ik}$  bzw.  $B_i$  wiederum Fuzzy-Sets sind. Die Klauseln haben die Form  $\Xi_1 = A_{1k}$  bzw.  $\Psi = B_j$ . Die Regeln haben dann folgende Form:

$$\text{Wenn } \Xi_1 = A_{1j} \dots \text{ und } \Xi_i = A_{ij} \dots \text{ und } \Xi_n = A_{nj} \text{ dann } \Psi = B_j$$

Die Fakten liegen in der Form  $A = A_1, \dots, A_n$  als Ausprägungen der  $n$  Variablen vor. Das Inferenz-Modul muß also aus den vorgegebenen Fakten und Regeln den Wert für  $\Psi$  bestimmen. Die Auswertung der Regelbasis vollzieht sich dabei in mehreren Schritten.

Als erstes erfolgt die Berechnung des Übereinstimmungsmaßes (Kompatibilitätsmaß) der Klauseln einer Regeln mit den aktuellen Fakten. Dieses Maß ist eine reelle Zahl aus dem Intervall  $[0, 1]$ . Es ergibt sich unmittelbar aus dem Wertebereich der Fuzzy-Sets und kann einfach durch die Berechnung der Zugehörigkeitsgrade für jedes Fuzzy-Set an der Stelle  $A_j$  ermittelt werden. Es gilt:

$$\alpha_{1j} = \text{komp} ( A_{1j}, A_1 )$$

$$\alpha_{2j} = \text{komp} ( A_{2j}, A_2 )$$

...

$$\alpha_{nj} = \text{komp} ( A_{nj}, A_n )$$

Nach Ausführung der Operation ist bei der Terminplanung an dieser Stelle das *Zeit* Fuzzy-Set einer jeden Person inklusive der Modifikatoren berechnet.

##### 4.4.3.3.1 Aggregations-Operatoren

Nun werden für die Linke Seite einer Regel  $j$  die einzelnen Kompatibilitätsmaße zu einem Gesamtkompatibilitätsmaß  $\alpha_j$  aggregiert. Dieses Maß drückt aus, in welchem Maß die Vorbedingung einer Regel erfüllt ist.

Im Unterschied zur Regelbearbeitung in einem normalen Expertensystem, bei dem die Vorbedingung einer Regel entweder erfüllt ist oder nicht, „feuert“ in einem Fuzzy-System eine Regel bereits dann, wenn ihre Vorbedingung mit einem Maß größer *Null* erfüllt ist.

Als Operatoren für die Aggregation kommen je nach Anwendung t-Normen, Durchschnitts-Operatoren oder t-Conormen in Frage. Für die Terminplanung wird der Min/Max-Operator verwendet. Mathematisch wird die Gesamtkompatibilität oder Prämisse ausgedrückt als Verknüpfung der einzelnen Klauseln:

$$\alpha_j = \text{aggOp} ( \alpha_{1j}, \alpha_{2j}, \dots, \alpha_{nj} )$$

#### 4.4.3.3.2 Sicherheitsfaktoren

In der Fuzzy-Logik wird davon ausgegangen, daß die Schlußfolgerung einer Regel immer im gleichen Grad erfüllt ist wie ihre Vorbedingung. Dieses Vorgehen entspricht nicht immer dem menschlichen Denken. Dies ist der Fall, wenn verschiedene Regeln unterschiedliche Aussagen über den Zustand einer Variablen treffen. Ein Mensch würde dann das Vertrauen in die Richtigkeit der Regeln und die Grade, zu denen die Vorbedingungen erfüllt sind, ins Verhältnis setzen. In der Fuzzy-Set-Theorie gibt es dafür die Methode des „unscharfen Schließens“.

In einer vereinfachten Form wird der Grad, zu dem die Schlußfolgerung gelten soll, durch eine Verknüpfung des Vertrauensmaßes (Sicherheitsfaktor  $\chi_j$ ) mit dem Grad, zu dem die Vorbedingung der Regel  $j$  gilt, ermittelt:

$$\alpha_j^* = \text{cerOp}(\alpha_j, \chi_j)$$

$\alpha_j^*$  ist dabei der modifizierte Gesamtkompatibilitätsgrad, der für den nächsten Schritt des Inferenz-Prozesses benötigt wird.

Bei der Terminplanung wird ein Sicherheitsfaktor von 1.0 angewendet. Die Schlußfolgerungen gelten somit immer als sicher.

#### 4.4.3.3.3 Inferenz-Operatoren

Die Inferenz-Operation hat nun die Aufgabe, den mit dem Vertrauensmaß modifizierten Kompatibilitätsgrad  $\alpha_j^*$  auf das resultierende Fuzzy-Set  $B_j$  abzubilden:

$$B_j^* = \text{infOp}(\alpha_j^*, B_j)$$

Die Schlußfolgerung einer Regel kann höchstens zu dem Grad erfüllt sein, wie ihre Vorbedingung. Es gilt also folgende Relation:

$$\text{hgt}(B_j^*) \leq \alpha_j^*$$

Für den Einsatz als Inferenz-Operator können im Prinzip alle nicht kompensatorischen t-Normen gewählt werden.

Die folgende Abbildung 35 zeigt das unterschiedliche Modifikationsverhalten dreier Operatoren für ein trapezförmiges Fuzzy-Set  $B_j$  und dem Kompatibilitätsgrad  $\alpha_j^* = 0,7$ . Die resultierenden Fuzzy-Sets  $B_j^*$  sind jeweils grau hervorgehoben.

Der Minimum-Operator (links) „schneidet“ das Fuzzy-Set  $B_j$  an der Achse  $y = 0,7$  ab und wird bei der Terminvereinbarung verwendet. Der Algebraic-Product-Operator (mitte) ist in seiner Wirkungsweise dem Minimum-Operator ähnlich. Der Unterschied liegt in der Größe der produzierten Fläche, was unmittelbar mit den algebraischen Eigenschaften der Operation zusammenhängt: der Operator liefert für die Argumente  $x$  und  $y$  immer Werte zurück, die kleiner oder gleich dem Ergebnis des Minimum-Operators sind. Der Drastic-Produkt-Operator (rechts) beschränkt das Fuzzy-Set  $B_j^*$  am stärksten. Dieser Operator ist nur für

normalisierte Fuzzy-Sets bzw. für absolut sichere Regeln geeignet, da eine Höhe des Fuzzy-Sets  $B_j$  kleiner als *Eins* ( $hgt(B_j) < 1$ ) zu einer Fläche der Größe *Null* und damit zur Eliminierung der Schlußfolgerung führt.

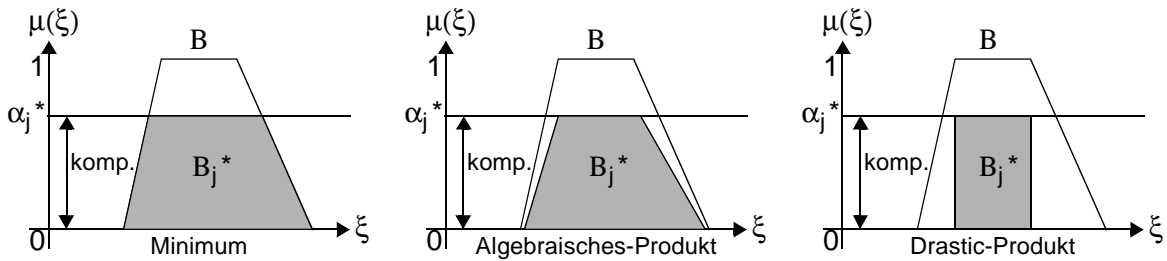


Abbildung 35: Inferenz mit verschiedenen Operatoren

#### 4.4.3.3.4 Akkumulations-Operatoren

In der letzten Operationsphase des Inferenz-Moduls wird das Gesamtergebnis  $B^*$  durch Akkumulation der Teilergebnisse  $B_j^*$  aller  $k$  Regeln ermittelt:

$$B^* = accOp ( B_1^*, B_2^*, \dots, B_k^* )$$

Da alle Regeln alternativen Charakter haben, sollte die Kombination durch eine Oder-Verknüpfung realisiert werden. Mögliche Operatoren sind alle bekannten t-Conormen. In den meisten Fuzzy-Controllern kommt der Maximum-Operator zur Anwendung. Werden gleichzeitig das Maximum als Akkumulations-Operator und das Minimum bzw. das algebraische Produkt als Inferenz-Operator eingesetzt, spricht man von Max-Min-Inferenz respektive von Max-Prod-Inferenz.

#### 4.4.3.4 Defuzzifizierung

Das Ergebnis des Inferenz-Prozesses repräsentiert das oben erwähnte Output-Fuzzy-Set  $B^*$ , das nach den zahlreichen Transformationen und Verknüpfungen ein entsprechend „bizarres“ Aussehen haben kann. Soll dieses Ergebnis nicht nur graphisch angezeigt werden, ist eine entsprechende Weiterverarbeitung notwendig. In Fuzzy-Expertensystemen ist es wegen der hohen Mensch-Maschine Interaktion üblich, das Output-Fuzzy-Set durch ein Verfahren der linguistischen Approximation verbal zu charakterisieren. Die linguistische Approximation beschäftigt sich mit der Problematik einen betrachteten Fuzzy-Set umgangssprachlich zu interpretieren (siehe [165]).

In der Steuer- und Regelungstechnik muß die resultierende unscharfe Menge wieder in eine skalare Steuergröße, also meist eine reelle Zahl, überführt werden. Diese Operation nennt man *Defuzzifizierung*. Das Problem besteht darin eine Zahl zu finden, die das Output-Fuzzy-Set möglichst genau charakterisiert. Hierbei geht zwangsläufig ein Teil der Informationen verloren.

Die bekanntesten Methoden der Defuzzifizierung sind die Maximum- und die Flächen-schwerpunkts-Methode, die nachfolgend erklärt werden.

- Maximum

Dieses Defuzzifizierungsverfahren wählt einfach den Wert, an dem die Zugehörigkeitsfunktion des Output-Fuzzy-Set maximal ist. Dieses Verfahren führt für Fuzzy-Sets basierend auf unscharfen Zahlen zu eindeutigen Ergebnissen. Wird das Inferenz-Ergebnis aber mit dem Minimum-Operator, der Fuzzy-Sets intervallweise „abschneidet“, berechnet, ist das Ergebnis nicht eindeutig.

Zur Lösung dieses Konflikts bietet sich an, den Mittelwert der Ergebnismenge zu nehmen, wobei allerdings nicht garantiert ist, daß der Zugehörigkeitsgrad des Mittelwertes maximal ist. Eine Alternative dazu ist die Wahl des Median bzw. des Zentralwertes der Ergebnismenge.

- Flächenschwerpunkt

Die häufigste Methode der Defuzzifizierung ist die Berechnung des Flächenschwerpunktes der resultierenden Zugehörigkeitsfunktion. Dieses Verfahren hat sich in der Praxis bewährt und ist relativ einfach zu implementieren [91]. Da die Zugehörigkeitsfunktion des Output-Fuzzy-Set im allgemeinen nicht bekannt ist, verwendet man zur Berechnung Verfahren der numerischen Integration. In Frage kommen dafür etwa die *Trapezregel*, die *Simpsonsche Regel* oder das *Rombergsche Verfahren* (siehe dazu z.B. [94]).

Liegt das Output-Fuzzy-Set als generelles Fuzzy-Set vor, ergibt sich folgende Formel für die Berechnung des Flächenschwerpunktes:

$$S_x = \frac{\sum_{i=1}^n \frac{1}{3} m_i (t_i^3 - t_{i-1}^3) + \frac{1}{2} c_i (t_i^2 - t_{i-1}^2)}{\sum_{i=1}^n \frac{1}{2} m_i (t_i^2 - t_{i-1}^2) + c_i (t_i - t_{i-1})}$$

Zur Defuzzifizierung bei der Terminplanung findet die Maximum-Methode Verwendung. Es wird also nach dem Maximum der Ausprägung *Zeit* der Ergebnis-Variable gesucht. Der gefundenen Wert wird dann mit einem zuvor definierten Schwellwert als untere Schranke verglichen. Gibt es einen maximalen Wert an der Stelle  $\xi$ , der größer als der Schwellwert ist, so kann mit das Datum des gefundenen Termins mittels  $T(\xi)$  errechnet werden. Der Schwellwert ist direkt abhängig von einer von Initiator bestimmten *Wichtigkeit* für einen Termin. Diese *Wichtigkeit* kann als Gegenstück zur oben definierten *Wichtung* von bereits bestimmten Terminen angesehen werden. Allerdings sollte der Basiswert niedriger liegen, als der Basiswert eines Termins (vergleiche Abschnitt 4.4.3.2), da im allgemeinen ein bestehender Termin nicht von einem „gleichwichtigem“ Termin überschrieben werden sollte.

$$\text{Schwellwert für Termine} = 0.25$$

Der Anwender kann nun die Modifikatoren *mehr oder weniger wichtig* und *sehr wichtig* je nach Wichtung seines Terminwunsches bestimmen. Allerdings ist die interne Interpretation an dieser Stelle genau umgekehrt: je wichtiger ein Termin ist desto höher ist der Schwellwert an einer Stelle und desto wahrscheinlicher wird eine Terminvereinbarung zu diesem Zeitpunkt:

$$\begin{aligned} \mu_{\text{sehr-wichtig}}(\xi) &= \sqrt{\mu(\xi)} \quad , \quad \forall \xi \in \Xi \\ \mu_{\text{mehr-oder-weniger-wichtig}}(\xi) &= \mu(\xi)^2 \quad , \quad \forall \xi \in \Xi \end{aligned}$$

#### 4.4.3.5 Umsetzung in das Agentenmodell

Die in den letzten Abschnitten gemachten theoretischen Überlegungen sollen nun in Bezug zum Agentmodell dargestellt werden. Um den Sachverhalt deutlicher darstellen zu können, wird der Fuzzy-Logik Prozeß graphisch aufbereitet. Der in Abschnitt 4.3.6 vorgestellte Ablauf der Terminvereinbarung dient dazu als Basis. Die Terminvereinbarung findet, wie in der Erläuterung der Regelbasis bereits dargestellt, zwischen dem *Host-Agent* und den *User-Agenten* statt. Die User-Agenten schicken dazu eine fuzzifizierte Repräsentation des fraglichen Zeitintervalls zum Host-Agenten, so daß dieser eine gewichtete Kombination aller Präferenzen vornehmen kann, um den optimalen Termin zu berechnen. Der erste Teil des Algorithmuses findet bei den Partnern statt:

Der eigene Terminkalender wird ausgelesen und die bereits vergebenen Termine werden zu Zeit/Werte Paaren verknüpft. Das resultierende Fuzzy-Set repräsentiert die für eine Person möglichen Termine (Abbildung 36), die unter Umständen eine unterschiedliche Wichtigkeit haben.

Die möglichen Termine werden anhand der Zeitdauer des Termins modifiziert. Für eine halbstündige Sitzung ist die letzte halbe Stunde eines möglichen Termins kein Zeitpunkt für den Beginn der Sitzung. Diese Zeiträume sind invalide, werden eliminiert und dadurch vom Algorithmus nicht berücksichtigt (Abbildung 37 und 38).

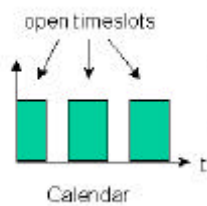


Abbildung 36: Freie Termine

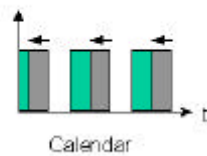


Abbildung 37: Modifikation



Abbildung 38: Mögliche Termine

Im nächsten Schritt werden die resultierenden möglichen Termine mit einer nach den Präferenzen ausgerichteten Gewichtung versehen. Dies wird durch die Bildung der gewichteten Summe der Minima erreicht (Abbildung 39 und 40):

$$\mu_{\text{und}}(x, y) = \gamma \cdot \min(x, y) + \frac{(1 - \gamma) \cdot (x + y)}{2}$$



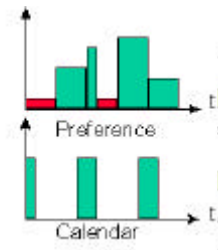


Abbildung 39: Gewichtung

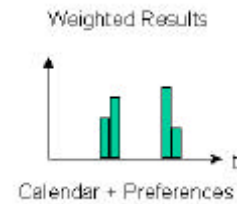


Abbildung 40: Resultierendes Fuzzy-Set

Die resultierenden Fuzzy-Sets werden an den Host-Agenten gesendet, der die Berechnung des Termins vornimmt (Abbildung 41).

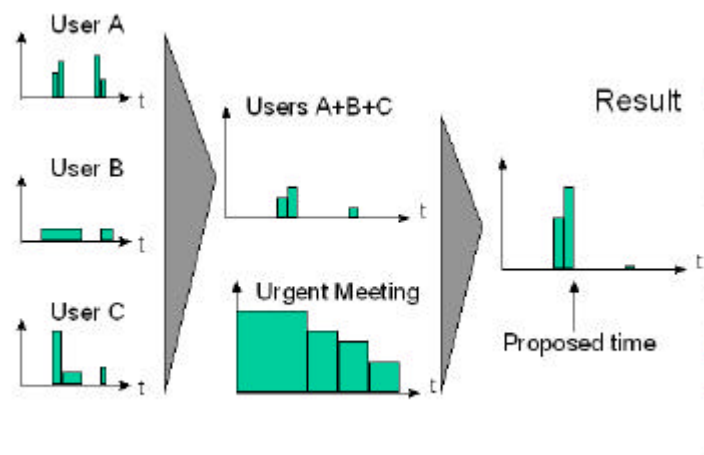


Abbildung 41: Kombination der Ergebnisse, Wichtung nach Dringlichkeit und Berechnung des resultierenden Termins

Wiederum wird die gewichtete Summe gebildet. Das Ergebnis spiegelt das kombinierte Fuzzy-Set aller Benutzer mit ihren Präferenzen wider.

Dieses Fuzzy-Set wird nun durch die Dringlichkeitsfunktion  $\Delta(\xi)$  gefiltert. Falls keine besondere Dringlichkeit spezifiziert wurde, hat diese Operation keinen Einfluß auf das Ergebnis. Falls doch, werden frühere mögliche Zeitpunkte eher berücksichtigt als spätere.

Im letzten Schritt wird anhand der Maximum-Methode der beste Termin ausgewählt.

Ist das Ergebnis größer als der Schwellwert, wird der Termin an die anderen Agenten versandt. Der Schwellwert ist abhängig von der Wichtigkeit des Termins für den Initiator und den Präferenzen der Partner. Ein Sitzungstermin für eine unkritische Diskussion mag dadurch nicht zustande kommen, während eine sehr wichtige Sitzung einen Termin bekommt.

## 4.5 Zusammenfassung

Die Konzeption des Referenzmodells einer computergestützten kooperativen Verwaltungsumgebung ist das Hauptanliegen dieses Kapitels gewesen. In der Verwaltung kommen im Zuge einer Automatisierung neue kooperative Techniken zum Einsatz, die einen flexiblen Ansatz bei der Integration und Einführung erfordern. Die Grundlage des Modells ist ein Agentenverbund, der durch seine autonomen und anpassungsfähigen Einheiten die Voraussetzungen für eine optimale Unterstützung des Anwenders und die Möglichkeit zu einer software-ergonomischen Integration der Techniken bietet.

Der Agentenverbund ist nach den Prinzipien eines halb-föderativen Systems aufgebaut. Aus den in der Motivation und in Kapitel 3 hergeleiteten Anforderungen sind dazu vier Basis-Klassen für den Agentenverbund abgeleitet worden. Je eine Agentenklasse für die Interaktionen mit einem Workflow- oder Telekooperations-System, eine für Termin- und Ressourcenplanung und eine zur Administration. Eine Erweiterung des offenen Modells ist dabei auf Basis von neuen Agentenklassen möglich.

Durch die Einbindung der Transparenzdatensätze aus Kapitel 3 ist das Referenzmodell verfeinert worden. Die Eigenschaften der Agenten, Autonomie, Sozialfähigkeit, Pro-Aktionsfähigkeit und Glaubwürdigkeit, erlauben eine Regulierung der Transparenz. Eine der Hauptanforderungen an Transparenz, die Gewährleistung, daß keine Überwachung von Personen stattfinden kann, ist hiermit erfüllt.

Der Workflow-Agent vermittelt zwischen dem Agentenverbund und dem Workflow-System. Eine Anbindung an das Referenzmodell der Workflow-Management Coalition (WfMC) ist dazu aufgezeigt worden. Hierdurch wird eine Abfrage von Meta-Daten ermöglicht, die in Transparenzdatensätzen des Workflow-Agenten abgelegt und von ihm kontrolliert werden. Je nach Anfrage und Anfrager werden diese Daten gefiltert weitergereicht und können so zur weiteren Verarbeitung genutzt werden. Diese Daten können zum Beispiel Informationen zum verantwortlichen Sachbearbeiter oder zum Entscheidungsträger eines Vorgangs beinhalten und für einen Kommunikationsaufbau genutzt werden. Neben dieser Informationsbeschaffung ist der Workflow-Agent in der Lage, Vorgänge zu initiieren und zu manipulieren.

Die Integration und Ansteuerung einer Telekooperationslösung wird vom Telekooperations-Agenten realisiert. Hier wurde der Standard T.120 als Beispiellösung für ein Telekooperations-System verwendet. Der Agent übernimmt die Aushandlung eines geeigneten Kommunikationsmediums und initiiert verteilte Konferenzen unterschiedlichster Art. Dabei nutzt er die Transparenzdatensätze des Workflow- und des Terminkalender-Agenten.

Der Terminkalender-Agent vermittelt Termine und verwaltet Ressourcen, wie Räume oder Geräte. Dazu wird das Calendar and Scheduling Interface (CSA), welches von der X/Open Group und der X/400 API Assosiation definiert wurde, herangezogen.

Um den Agenten autonome Entscheidungen zu erlauben wird die Mathematik der Fuzzy-Logik verwendet. Ein kurzer Exkurs führte diesen Bereich ein. Mit Hilfe der Fuzzy-Logik

lassen sich natürlichsprachliche Anforderungen von Benutzern durch linguistische Variablen in eine mathematisch exakte Form bringen. Vorteil ist, daß auch unscharfe Aussagen attribuiert werden und in Entscheidungen des Agentenverbund einfließen können. Anhand des Beispiels einer Terminvereinbarung wurden konkrete Definitionen für linguistische Variablen spezifiziert.

Das Referenzmodell integriert heterogene CSCW-Werkzeuge und führt den Benutzer in software-ergonomischer Form durch die komplexen Systeme. Dies wird durch die Verwendung von Meta-Informationen, die in Transparenzdatensätzen repräsentiert werden, und durch die, gegenüber statischen herkömmlichen Lösungen, flexiblen und autonomen Möglichkeiten des Agentenverbundes, erreicht. Im Gegensatz zu anderen Ansätzen ergibt sich weiterhin, daß Transparenz im Gruppenraum ohne Überwachung stattfindet. Dem Anwender wird so die Ausnutzung der durch Transparenz gegebenen Vorteile, ohne das ihm Nachteile entstehen, ermöglicht.



## Kapitel 5

### Mikro-Level-Transparenz oder Der Fokus des Interesses

#### 5.1 Motivation

In [38] wird von Ellis et al. Groupware definiert als Software, die eine Schnittstelle zu einer kooperativ genutzten Umgebung zur Unterstützung einer Gruppe, die eine gemeinsame Aufgabe oder Ziel hat, bietet. Für synchrone Interaktion bedeutet dies, daß der Fokus des Interesses (FDI), (englisch: *point of interest, POI*) für alle Partner gleichsam zugänglich sein muß. Neben der Möglichkeit den FDI zu sehen, ist es wichtig Mechanismen zur Interaktion anzubieten. Diese synchrone kooperative Interaktion mit dem FDI ist als Schlüsselfaktor im Bereich der computerunterstützten Gruppenarbeit anzusehen [112].

Synchrone kooperative Interaktion mit dem FDI kann als wesentliches Teilgebiet der Mikro-Level-Transparenz (siehe Abschnitt 2.6.2) angesehen werden. Sie unterstützt die Mitglieder eines Team gezielt bei der Lösung von Problemen. Gemeinsam kann in einem Arbeitsbereich diskutiert und aktiv am aktuell zu bearbeitenden Objekt manipuliert werden. Autorentams muß dazu die Möglichkeit gegeben werden, Dokumente gleichzeitig bearbeiten und annotieren zu können. Dabei darf eine räumliche Distanz zwischen den einzelnen Mitgliedern kein Hindernis sein. Für die Bearbeitung der Dokumente sollte dem Anwender die Wahl der Applikation überlassen werden. Dies bedeutet, daß synchrone kooperative Interaktion mit dem FDI mit unterschiedlichsten Anwendungen ermöglicht werden muß.

Der Grad der Interaktion ist mit heute üblichen CSCW-Komponenten unterschiedlicher Natur. Im Gegensatz zu Sharing-Komponenten, bei denen eine Standard-Applikation oder eine sogenannte CSCW-unaware Applikation im Fokus des Interesses steht, sind Whiteboards speziell für die Kooperation entworfen und werden als CSCW-aware Applikationen bezeichnet (siehe Abschnitt 2.1.3). Whiteboards dienen damit selbst als FDI in einer Konferenzsituation. Der Unterschied ist tatsächlich, daß Sharing-Komponenten im Gegensatz zu Whiteboards eine *aktive* Interaktion mit dem FDI zulassen. Whiteboards unterstützen hingegen eine *passive* Interaktion. Beide Techniken erlauben allerdings WYSIWIS (*What You See Is What I See*) und erfüllen damit die Grundvoraussetzung für synchrone interaktive CSCW-Werkzeuge.

An dieser Stelle sollte klar gestellt werden, daß Audio/Video-Komponenten sich eindeutig zur Unterstützung von Mikro-Level-Transparenz eignen. Allerdings werden sie hier nur als Basis-Werkzeuge angesehen. Gerade eine gute Audio-Verbindung wird als essentiell für die Diskussion am FDI angesehen. Video-Kanäle haben sich als besonders geeignet gezeigt, Mikro-Level-Transparenz zusätzlich zu unterstützen (siehe etwa [36], [71] oder [136]). Der FDI selbst, wie er hier verstanden wird, wird aber selten durch ein Audio- oder Video-Signal verkörpert. Von daher soll dieser Bereich nur nebenläufig diskutiert werden. Auch der Bereich der Konferenzsysteme soll hier ausgeklammert werden. Konferenzsysteme sind verantwortlich für die Administration einer synchronen Telekonferenz. Teilnehmer werden eingeladen, Konferenzen werden auf- und wieder abgebaut. Zur Durchführung der eigentlichen Konferenz werden andere Applikationen, wie etwa Sharing Komponenten und Whiteboards, aber auch Audio/Video-Komponenten gestartet (siehe auch Abschnitt 2.3). Konferenzsysteme sind daher organisatorische Bestandteile und unterstützen nicht direkt die Interaktion mit dem Fokus des Interesses.

Kooperative Gruppeneditoren sind dagegen in Betracht zu ziehen. Sie erlauben sowohl den gemeinsamen synchronen als auch den asynchronen Zugriff auf ein Dokument. Wie bereits erläutert (siehe Abschnitt 2.2.4) kann der asynchrone Zugriff auf Mechanismen, die ähnlich konkurrierenden Datenbank-Zugriffen sind, zurückgeführt werden. Diese Mechanismen werden auch beim synchronen Zugriff verwendet. Um auf den hier verwendeten Kontext des FDI zurückzukommen, können zwei Szenarien unterschieden werden. Jeder Autor editiert einen unterschiedlichen Teil im Dokument, oder die Autoren diskutieren gemeinsam einen bestimmten Teil des Dokumentes. Im ersten Fall existieren mehrere FDI, da jeder Autor seinen eigenen FDI besitzt. Dieses Szenario kann abgebildet werden auf das asynchrone Szenario und es können die Mechanismen des konkurrierenden Zugriffs in Betracht gezogen werden. Im zweiten Szenario wird eine synchrone kooperative Interaktion mit dem FDI durchgeführt. Die Interaktion mit einer bestimmten Entität des Dokumentes wird ebenfalls von den Mechanismen des konkurrierenden Zugriffs gesteuert. Um den Zugriff zu regulieren wird ein *Floor-Control*-System verwendet (siehe Abschnitt 2.2.5). Von Anwenderseite ist der zweite Fall damit identisch zu einem Shared-Application-Szenario. Der einzige Unterschied ist, daß die Manipulation einer Entität entweder im Datenrepositorium des Gruppeneditors durch Datenbanktransaktionen oder in der verteilten Standard-Applikation direkt stattfindet. Die Sharing-Komponente übernimmt im letzteren Fall nur die Kontrolle und das Demultiplexen der Eingabe des Benutzers. Da Sharing-Komponenten einen allgemeineren Charakter besitzen und ein breiteres Anwendungsfeld bieten, sollen sie im folgenden den Interessenschwerpunkt bilden. Die vorgestellten Konzepte lassen sich in den wesentlichen Zügen aber auch auf Gruppeneditoren übertragen.

Im folgenden wird ein generelles Konzept zur Realisierung von synchroner Interaktion mit dem Fokus des Interesses innerhalb einer verteilten kooperativen Sitzung im Kontext einer verteilten Verwaltung dargelegt. Dazu wird zunächst der Begriff „Interaktion“ mit dem FDI näher beleuchtet werden. Daran anschließend wird das Konzept zur Interaktion spezifiziert. Neben transparenten Annotationen werden dabei die Semantische Verknüpfung und die dynamischen Annotationen vorgestellt. Abschnitt 5.4 beschreibt dann die Integration der in diesem Kapitel erarbeiteten Ergebnisse in das Referenzmodell und zeigt die sich daraus ergebende Referenzarchitektur auf.

## 5.2 Definitionen

### 5.2.1 Aktive und passive Interaktion

An dieser Stelle soll zunächst genauer spezifiziert werden, was aktive und passive Interaktion mit dem FDI bedeutet. Dazu muß zunächst der Fokus des Interesses als solches festgelegt werden. In einem kooperativen Szenario kann er als der Punkt definiert werden, der zu einem bestimmten Zeitpunkt diskutiert wird. Weiterhin existiert eine Entität, die den FDI repräsentiert. Dies kann zum Beispiel ein spezifischer Satz oder Paragraph in einem Dokument, welches von einem verteilten Texteditor bearbeitet wird, ein Detail einer 2D Designstudie repräsentiert durch ein eingelesenes Bild in einem Whiteboard, oder ein graphisches Primitiv eines 3D Modells in einem verteilten CAD-System sein.

Für die aktive Interaktion mit dem FDI kann damit definiert werden:

- Eine aktive Interaktion mit dem Fokus des Interesses (FDI) ist definiert durch die Bearbeitung der den FDI repräsentierenden Entität durch einen Benutzer.

Bevor die passive Interaktion definiert werden kann, muß definiert werden, was eine Annotation im Kontext des FDI bedeutet:

- Eine Annotation ist eine Anmerkung zu einer speziellen FDI-Entität, die durch ein oder mehrere Medien (z.B. Graphikprimitive, Text, Sprache, Video) repräsentiert wird. Es existiert eine klare Korrelation zwischen Annotation und FDI-Entität, welche für jeden Nutzer sichtbar ist.

In dem hier zu Grunde liegenden Kontext der synchronen Kooperation ist es von entscheidender Wichtigkeit, daß eine Annotation nicht nur von allen Gruppenmitgliedern erfaßt werden kann, sondern auch die Korrelation mit dem FDI jedem verständlich ist. Für die passive Interaktion mit dem FDI kann somit folgende Definition festgelegt werden:

- Die passive Interaktion mit dem Fokus des Interesses (FDI) wird definiert als die Annotation des FDI durch einen Benutzer.

Es wird nun deutlich, warum Sharing Komponenten sich mehr für aktive Interaktion und Whiteboards mehr für passive Interaktion mit dem FDI eignen. Dieser Punkt wird in den folgenden Abschnitten noch genauer beleuchtet werden.

### 5.2.2 Granularität

Man kann unterschiedliche Granularitäten bei der Interaktion mit dem Fokus des Interesses unterscheiden. Für eine einzelne Person kann der FDI vom gesamten Dokument bis zu einem einzelnen Wort reichen. Dies korrespondiert zu einer sehr groben bis zu einer sehr detaillierten Sichtweise. Jeder Abstraktheitsgrad paßt zu einer bestimmten Verantwortlichkeit des Nutzers, etwa von der Erzeugung oder der Bestimmung des Inhalts eines Dokumentes bis zu Inhalt, Syntax oder Schreibweise eines Absatzes. Im Falle einer verteilten synchronen Kooperation einer Benutzergruppe kann angenommen werden, daß das Team

eine bestimmte Aufgabe, etwa die Erstellung eines gemeinsamen Dokumentes, zusammen lösen soll. Betrachtet man dies von einem abstrakten Blickwinkel, ist der FDI des Teams das Dokument. Jedes einzelne Mitglied des Teams ist aber verantwortlich für bestimmte Paragraphen und Abschnitte, also für eine mittlere Granularitätsstufe.

Kooperative Arbeit muß als eine kontinuierliche Kette von abwechselnd asynchronen und synchronen Tätigkeiten angesehen werden. Phasen asynchroner Arbeit, in der ein Gruppenmitglied seine individuellen Aufgaben löst, sind gefolgt von synchronen Tätigkeiten, in denen unterschiedliche Nutzer ihre Arbeiten koordinieren und harmonisieren. Dieser Prozeßschritt tritt dann ein, wenn ein Teammitglied etwa eine Inkonsistenz oder eine Widersprüchlichkeit zwischen seiner und der Arbeit eines anderen Mitglieds feststellt. Eine synchrone Interaktion (z.B. eine Telekonferenz) wird initiiert, das Problem - sozusagen der gemeinsame FDI - wird diskutiert und es wird annotiert oder abgeändert, bevor die Individuen wieder ihren eigenen persönlichen Aufgaben nachgehen. Vergleicht man die Zeitdauer synchroner Interaktion mit der von asynchronen Tätigkeiten, so wird deutlich, daß erstere sich zwischen Minuten bis zu wenigen Stunden bewegt, während letztere Tage bis Wochen in Anspruch nimmt. Diese Tatsache verlangt nach einer Langzeitunterstützung für den asynchronen Fall und einer flexiblen, einfach zu handhabenden Unterstützung des synchronen Falls. Mit der Einführung des Referenzmodells ist eine solche für beide Fälle gegeben.

Konzepte für die Administration von asynchroner Arbeit sind durch den Einsatz von Workflow-Systemen gegeben. Diese Werkzeuge koordinieren den Zugriff der Benutzer auf Dokumente. Diese Techniken können für das Management des FDI auf einem groben Granularitätsniveau adaptiert werden. Als ein Ergebnis wird als Basis für die asynchrone Gruppenarbeit ein Workflow- oder ein Dokumentenmanagement-System als Voraussetzung angesehen.

Die Diskussion am FDI auf einem mittleren oder feinen Granularitätsgrad, wie etwa über Paragraphen oder Sätzen muß durch CSCW-Werkzeuge unterstützt werden, die sich zunächst auf Funktionalitäten zum Redigieren oder Annotieren und nicht primär zum simultanen Modifizieren des FDI konzentrieren. Betrachtet man die Definitionen zur aktiven und passiven Interaktion mit dem FDI in vorherigen Abschnitt, können folgende Anforderungen an ein solches System gemacht werden:

- Asynchrone Vorbereitung einer Konferenz durch den Benutzer
- Synchroner WYSIWIS-Unterstützung für den FDI innerhalb einer Konferenz
- Modifikation des FDI während einer Konferenz (Aktive Interaktion)
- Möglichkeit zum Annotieren des FDI (Passive Interaktion)
- Fließender Wechsel zwischen beiden Interaktionsformen
- Abspeichern der Annotationen als Diskussionsergebnis



## 5.3 Konzeption zur Interaktion

### 5.3.1 Transparente Annotationen

Für die kooperative Diskussion am Dokument eignen sich die bereits vorgestellten Technologien *Whiteboards* und *Gruppeneditoren*, sofern letztere synchrone Kooperation unterstützen. *Whiteboards* besitzen allerdings ein starres, nicht dynamisches Rasterbild als Diskussionsgrundlage. Die Möglichkeit, das Dokument gleichzeitig zu bearbeiten, fehlt. Die Forderung nach einem fließenden Wechsel zwischen Annotation und Bearbeitung kann ebenfalls nicht erfüllt werden. *Gruppeneditoren* wiederum erlauben nicht die Integration unterschiedlicher Standard Anwendungen. Der Anwender kann sein bevorzugtes Dokumentenverarbeitungs-System nicht nutzen. Mit der Technologie des *application sharing* ist es möglich eine beliebige Applikation kooperativ zu nutzen. Eine Lösung ist daher die Kombination aus Whiteboard und Sharing-Komponente mit einer beliebigen Applikation als Basis.

Verbindet man die Sharing-Komponente mit den Möglichkeiten eines Whiteboards, erhält man ein System, welches die wesentlichen Anforderungen im synchronen Bereich unterstützt. Aktive und passive Interaktion mit dem FDI wird ermöglicht. Dazu ist es allerdings notwendig, daß die Whiteboard-Technologie erweitert wird:

- Der Arbeitsbereich des Whiteboards überlagert die durch eine Sharing-Komponente verteilte Applikation.
- Der Arbeitsbereich des Whiteboards ist transparent, dadurch sind nur die Annotationen sichtbar.
- Der Arbeitsbereich des Whiteboards ist durchlässig für Aktionen des Benutzers, die der darunterliegenden Applikation gelten.
- Ein fließender Wechsel zwischen Annotationsmodus des Whiteboards und Bearbeitungsmodus der Applikation ist möglich.

Dies kann erreicht werden durch die Einführung eines transparenten Whiteboards. Als Hintergrund wird hier die durch eine Sharing-Komponente verteilte Ausgabe einer beliebigen Applikation verwendet. Die Clients des transparenten Whiteboards überlagern die jeweiligen Ausgaben der Applikation und ermöglichen den Sitzungsteilnehmern das Annotieren der Applikation und somit die passive Interaktion mit dem FDI. Durch den transparenten Arbeitsbereich ist die Applikation und damit der FDI weiterhin sichtbar. Zusätzlich ist der Arbeitsbereich der Whiteboard-Clients durchlässig für Aktionen der Benutzer, die der Applikation gelten. Hierdurch kann eine aktive Interaktion mit dem FDI stattfinden. Mit einem speziellen Mechanismus wird zwischen Annotationsmodus und Aktionsmodus hin und her geschaltet. Abbildung 42 zeigt den Aufbau einer solchen Kombination aus Sicht eines Benutzers.

Die Architektur dieser Umgebung kombiniert die bereits vorgestellten Konzepte von Whiteboard und Sharing-Komponente. Eine Standard-Applikation wird mit Hilfe einer Sharing-Komponente an die in einer Konferenz angeschlossenen Arbeitsplatzrechner verteilt. Das

transparente Whiteboard überlagert die Ausgabe der Applikation und ermöglicht den Sitzungsteilnehmern das Annotieren der Applikation.

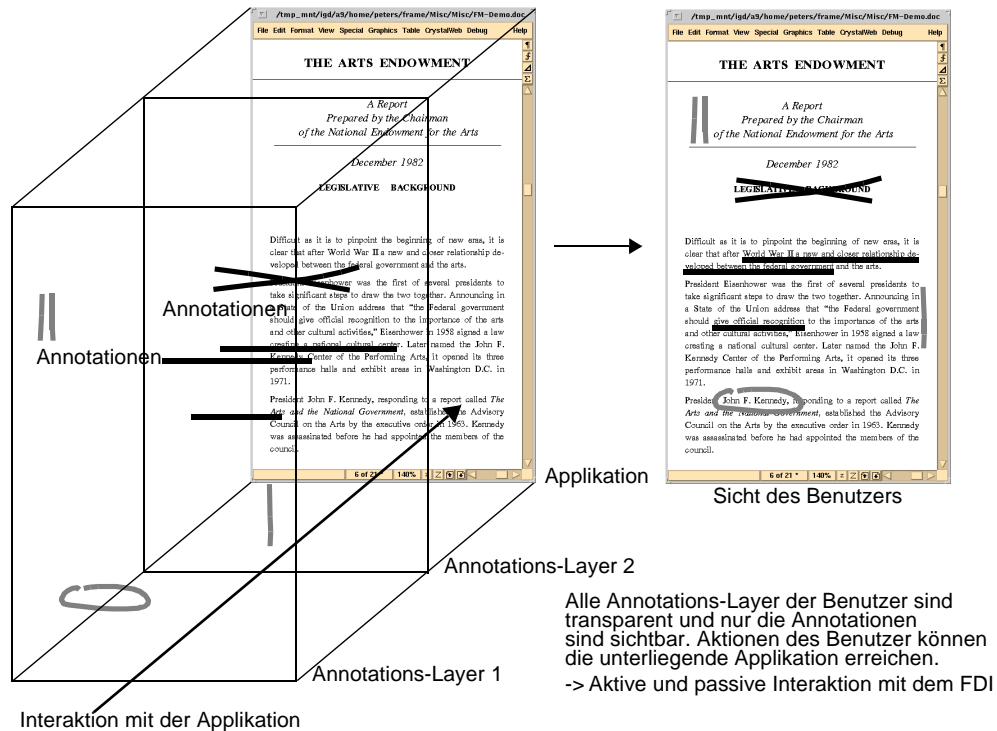


Abbildung 42: Technik eines transparenten Whiteboards

Die Realisierung dieser Technik ist zum Teil, wie etwa die Transparenz des Whiteboards, abhängig vom verwendeten Window-System. Im Kapitel 6 ab Seite 117 werden für zwei Window-Systeme konkrete Lösungen angegeben.

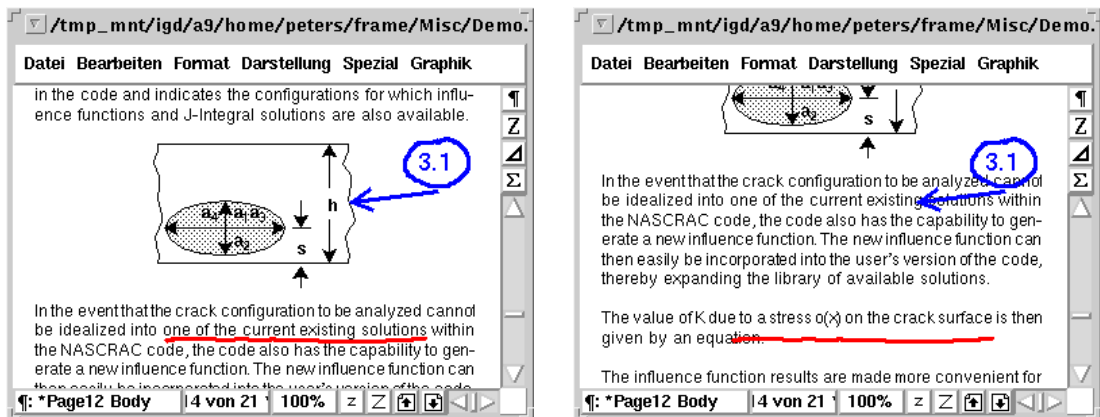
### 5.3.2 Semantische Verknüpfung

Mit der Möglichkeit die Applikation bearbeiten und annotieren zu können, muß gewährleistet bleiben, daß eine Annotation sich nicht unabsichtlich relativ zum annotierten Objekt der Applikation verschiebt. Der Betrachter bringt eine Annotation mittels deren Position in Relation zu einem Text- oder Graphikobjekt. Diese Relation zwischen Annotation und annotiertem Objekt wird mit *Semantischer Verknüpfung* bezeichnet (siehe dazu auch [107] und [109]).

Die Probleme treten auf, weil der Anwender die Applikation weiterhin „normal“ benutzen kann (und auch soll). Dies betrifft einerseits die Ansicht des Dokumentes, da im allgemeinen nicht davon ausgegangen werden kann, daß das vollständige Dokument sichtbar ist. Der angezeigte Ausschnitt kann etwa durch *Scrollen* oder auch durch Suchen nach einem Wort verändert werden (siehe Abbildung 43, Punkt 1). Eine Ausschnittsveränderung tritt ebenfalls durch eine Größenveränderung des Anzeigefensters oder durch *Zoomen* auf.

Andererseits kann der Benutzer den Inhalt des Dokumentes verändern (siehe Abbildung 43, Punkt 2). Texte oder Graphiken können neu eingefügt oder gelöscht werden. Weiterhin können Wörter oder Absätze verschoben werden, Attribute, etwa die Zeichengröße oder das Layout, geändert werden.

## 1. Scrollen



## 2. Text einfügen

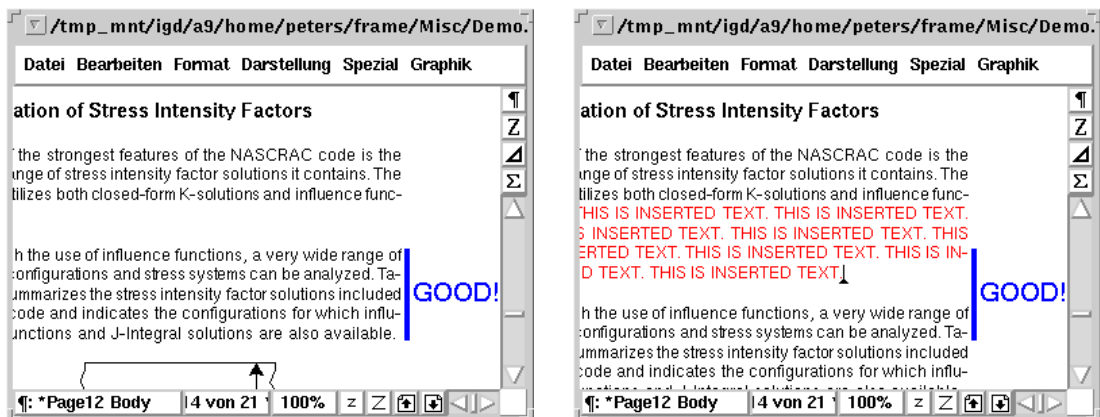


Abbildung 43: Probleme ohne Semantischen Verknüpfung

Es ist hierbei ein prinzipieller Unterschied, ob diese Änderungen „irgendwo“ gemacht werden, oder aber in einem Paragraph, in dem sich eine Annotation befindet, etwa ein unterstrichenes Wort (siehe Abbildung 44). Hier müßte die Annotation nicht nur in vertikaler (oder horizontaler) Richtung verschoben werden, sondern auch in ihrem Aussehen manipuliert werden.

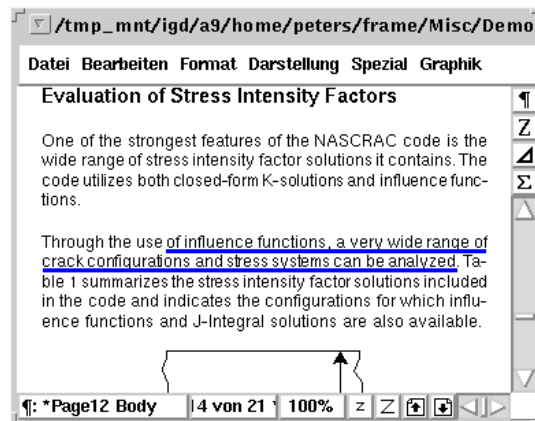
Es hat sich herauskristallisiert, daß zur Realisierung der Semantischen Verknüpfung eine Programmierschnittstelle (API) verwendet werden muß. Die minimalen Anforderungen an eine solche API sollen nun herausgestellt und kurz, anhand der gerade aufgezeigten Probleme, begründet werden.

## 1. Name des Dokumentes

Zur Identifikation von Dokumenten ist diese Information notwendig. Der Wechsel von Dokumenten kann über sie erkannt werden. Auch für ein Abspeichern bzw. Laden von Annotationen in Kombination mit einem Dokument wird sie verwendet.

Vorher: markierter Textabschnitt

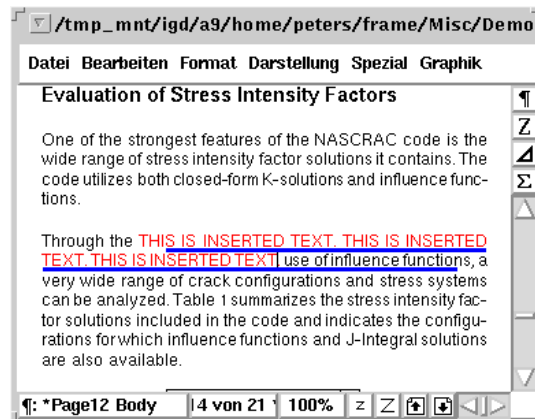
Annotationsgruppe bestehend aus zwei Linien.



Neuer Text ist eingefügt worden:

Ergebnis ohne Semantische Verknüpfung

Annotationsgruppe ist unverändert und hat ihren Kontext verloren.



Wunschergebnis mit Semantischer Verknüpfung

Annotationsgruppe bestehend aus drei (!) Linien, der Kontext bleibt erhalten.

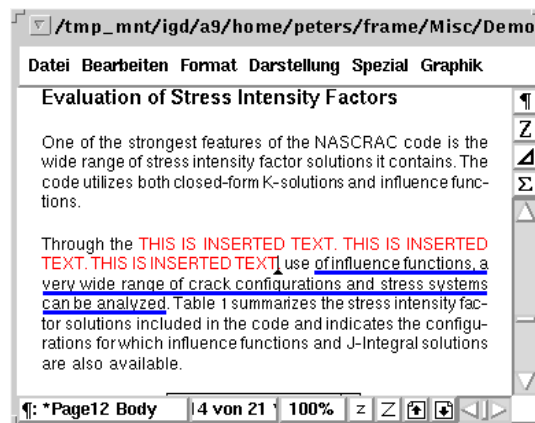


Abbildung 44: Einfügen im Paragraph

## 2. Der sichtbare Ausschnitt des Dokumentes

Um ihn zu berechnen benötigt man das Verhältnis der Größen des sichtbaren Ausschnitts zum Gesamtdokument und die Position des sichtbaren Ausschnitts im Gesamtdokument. Wird der sichtbare Ausschnitt des Dokumentes etwa durch Scrollen verändert, müssen notwendige Veränderungen der Annotationen anhand dieser Informationen berechnet werden. Sie entsprechen den Angaben, die bei einer Transformation zwischen verschiedenen Koordinatensystemen verwendet werden. Alternativ kann statt dem Verhältnis der Größen auch nur die Größe des unvergrößerten oder verkleinerten Dokumentes und ein Zoomfaktor verwendet werden.

## 3. Setzen von eindeutigen fließenden Markierungen und Abfragen von deren Position

Eine Veränderung des Inhalts im Dokument führt im allgemeinen, wie es in Abbildung 43, Punkt 2 oder Abbildung 44 dargestellt wird, zu einer Umformatierung des Textes. Eine Umformatierung betrifft aber u. U. alle bis dahin gemachten, zumindest aber alle weiter „unten“ bzw. „hinten“ liegenden Annotationen. Ein einfaches Beispiel dafür ist das Einfügen eines neuen Paragraphen ganz am Anfang eines Textes. Alle anderen Paragraphen „rutschen“ entsprechend weiter „runter“. Um solche Veränderungen verarbeiten zu können, muß es eine Möglichkeit der Kontrolle über diese Veränderungen geben. Diese kann nur im Dokument selbst erfolgen. Mit eindeutigen fließenden Markierungen, die in das Dokument gesetzt werden, ist eine solche Kontrolle zu einem großen Teil durchführbar.

Beim Erstellen einer Annotation wird eine korrespondierende eindeutige Marke genau an der Position in das Dokument eingesetzt, die der Bezugsposition der Annotation entspricht. Findet eine Umformatierung im Dokument statt, wird die Marke, da sie Teil des Dokumentes ist, mitformatiert und erhält eine neue Position. Diese Position kann dann (nach einer Notifikation s.u.) zusammen mit der aus 2. ermittelten Transformation zur endgültigen neuen Position und Größe der entsprechenden Annotation verarbeitet werden. Diese Form der Semantischen Verknüpfung mit Marken, wird im folgenden mit *semantic tagging*<sup>5</sup> bezeichnet.

Betrachtet man verschiedene Desktop-Publishing Systeme, ist es nicht unbedingt möglich jede beliebige Position im Dokument als Position für eine fließende Marke (die Betonung liegt hier auf „fließende“) zu verwenden. Hier kann durch die Einführung eines *offset* Abhilfe geschaffen werden. Ein weiterer Nachteil kann dadurch entstehen, daß die Marke Bestandteil des Textes ist und somit natürlich auch vom Anwender versehentlich gelöscht werden könnte. Dagegen halten kann man allerdings, daß mit dem Löschen der Marke auch die zur Annotation gehörige Entität gelöscht wird. Die Annotation ist damit bezugslos und kann ebenfalls gelöscht werden (u. U. nach Rückfrage beim Benutzer).

Mit der Einführung des *semantic tagging* ergeben sich zusätzlich neue Annotationsmetaphern, wie etwa *Dynamische Annotationen*, die mehr als nur eine Marke zum *semantic tagging* verwenden (siehe Abschnitt 5.3.3 „Dynamische Annotationen“).

---

5. Statt „Marke“ wird entsprechend synonym das Wort *tag* oder *semantic tag* verwendet

#### 4. Notifikation bei Verschiebung einer Marke oder Änderung des sichtbaren Ausschnitts des Dokumentes

Das transparente Whiteboard ist in Bezug auf Veränderungen in der unterliegenden Applikation ein passives Werkzeug in dem Sinne, daß es die Applikation, die es überlagert, nicht steuert. Eine Notifikation über eine Änderung, unabhängig davon, ob diese Änderung den Inhalt (und damit eine Verschiebung einer Marke verursacht) oder den sichtbaren Ausschnitt betrifft, muß konsequenterweise von der Applikation über die API selbst kommen, damit das transparente Whiteboard entsprechend reagieren und die Annotationen anpassen kann.

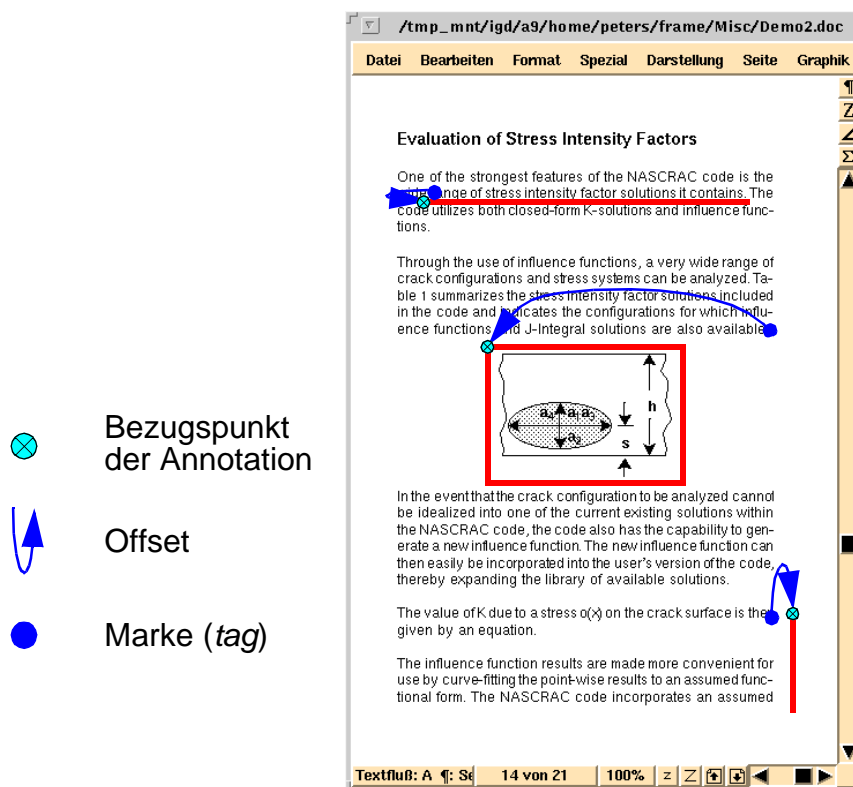


Abbildung 45: Semantic Tagging

Abbildung 45 verdeutlicht das Prinzip des *semantic tagging* mit verankerten Rahmen als *tag*. Wie bereits erwähnt, ist es nicht immer möglich, an der exakten Position, die für die Annotation gilt, einen *tag* anzubringen. Um diese Eigenschaft der unterliegenden Applikation auszugleichen wird ein *offset* zusammen mit dem *tag* abgelegt. Das Verfahren des *semantic tagging* wird zusammen mit der Notifikationsmethode verwendet, um die Semantische Verknüpfung zwischen Annotation und der Entität des FDI zu erreichen. Im Kapitel „Realisierungen“ wird eine konkrete Implementierung der Semantischen Verknüpfung vorgestellt.

### 5.3.3 Dynamische Annotationen

Im Gegensatz zu normalen graphischen Annotationen verwenden dynamische Annotationen mehr als einen *semantic tag*, um eine Semantische Verknüpfung vom Annotationsobjekt zur FDI-Entität zu erzeugen. Dadurch kann ein dynamisches Verhalten der Form der Annotation erreicht werden.

Durch das *semantic tagging* wird erreicht, daß sich Annotationen dem dynamischen Verhalten eines Dokumentes während einer Sitzung in ihrer Position anpassen. Mit dynamischen Annotationen wird nun auch die Form der Annotationen selbst in diesen Prozeß miteinbezogen. Die Form der Annotation paßt sich der Form der korrelierenden FDI-Entität dynamisch an.

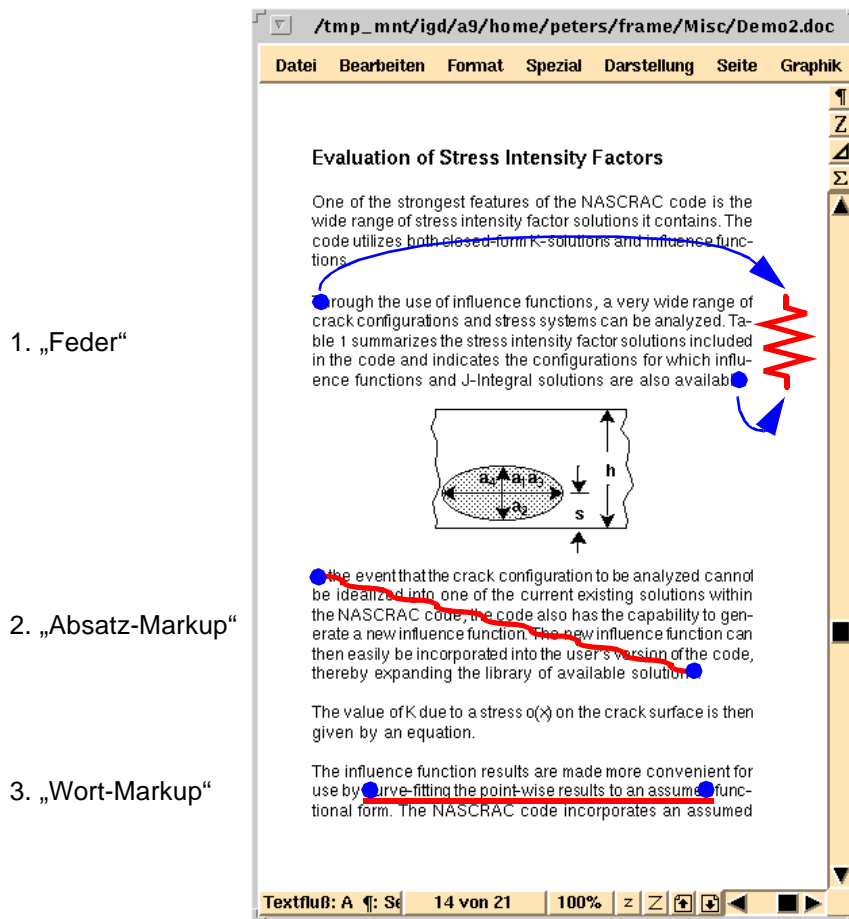


Abbildung 46: Dynamische Annotationen

Die Einführung von dynamischen Annotationen bedeutet auch eine fortschrittliche Erweiterung der bisher verwendeten graphischen Annotationen. Statische „Papierannotationen“ werden durch in mehrfacher Hinsicht (Position und Form) dynamische Annotationen verbessert.

Einige Beispiele sollen helfen, die Funktionalität von dynamischen Annotationen zu verdeutlichen. Abbildung 46 zeigt drei mögliche Arten von dynamischen Annotationen. Die erste stellt eine Art Feder dar, die einen Absatz markiert. Sie besitzt einen *semantic tag* jeweils am Anfang und am Ende des Absatzes. Wird der Absatz editiert, so verändert sich die Position des *semantic tags* am Ende des Absatzes. Die neue Y-Koordinate des *semantic tags* kann nun benutzt werden, um die Feder entsprechend der Höhe des Absatzes neu zu „spannen“.

Die beiden anderen dynamischen Annotationen verwenden das gleiche Prinzip: Zwei *semantic tags* für eine Annotation. Sie verändern sich allerdings dynamisch in X und Y als „Absatz-Markup“ bzw. nur in X-Richtung als „Wort-Markup“. Mit einem dritten *semantic tag* können beispielsweise auch 3-dimensionale Dynamische Annotationen realisiert werden.

## 5.4 Integration in das Referenzmodell

### 5.4.1 Anforderungsprofil

Um ein Aufgabenbild für die Integration von synchroner kooperativer Interaktion mit dem Fokus des Interesses in das Referenzmodell zu erhalten, soll zunächst ein detailliertes Anforderungsprofil durch Betrachtung unterschiedlicher Szenarien erstellt werden. Dabei soll das gesamte kommunikationsrelevante Aufgabenpotential innerhalb einer Verwaltung miteinbezogen werden. Resultierend wird eine systembezogene Referenzarchitektur für eine kooperative Verwaltungsumgebung innerhalb des Referenzmodells erstellt. Die folgenden Szenarien beinhalten kommunikationsrelevante Vorgänge mit dem Schwerpunkt auf Anmerkungen und sind zum einen der IVBB Studie entnommen (siehe [28] S. 3-20ff). Weitere Szenarien sind aus dem Verbundprojekt „Mensch-Maschine-Schnittstellen im multimedialen Verwaltungsbüro“ [10] abgeleitet worden:

- **Stellungnahme:** Zu einer übermittelten Unterlage soll eine Stellungnahme abgegeben werden. Die Bearbeitung erfolgt in der Regel in mehreren Schritten. Es können Anmerkungen gemacht werden, die in erster Linie in Form eines Textes oder von Sprache erfolgen, aber auch andere Formen wie Festbild oder graphische Anmerkungen sind wünschenswert. Es muß sichergestellt werden, daß die Verantwortlichkeit der Anmerkungen eindeutig gekennzeichnet sind und der Gesamtzusammenhang deutlich wird.
- **Abstimmungen:** Es muß sichergestellt sein, daß der zu unterzeichnende Entwurf nicht durch eine beteiligte Stelle verändert werden kann (Integrität), die beteiligte Stelle muß den Entwurf aber durch Ergänzung oder durch einen Änderungsvorschlag kommentieren können. Ändert ein Vorgesetzter einen ihm zur Zeichnung vorgelegten Entwurf förmlich oder sachlich ab, so muß er den Umfang der Änderung an dem Entwurf kenntlich machen.



- Umläufe: Durch Umläufe wird Information verteilt oder eine sachliche Bearbeitung gesteuert. Da die Bearbeitung der Umläufe in der Regel in mehreren Schritten erfolgen kann, muß sichergestellt werden, daß die Verantwortlichkeit der Anmerkungen eindeutig gekennzeichnet ist und der Gesamtzusammenhang deutlich wird.
- Rücksprache: Während der Bearbeitung eines Vorgangs tritt ein Problem auf, das mit Unterstützung von Kollegen, anderen Sachbearbeitern oder Vorgesetzten gelöst werden kann. Eine bilaterale Rücksprache sollte hier spontan ablaufen können. Das benötigte Schriftgut kann von den beteiligten Personen eingesehen und kommentiert werden. Siehe hierzu auch „Diskussion“.
- Persönliche Notizen: Persönliche Notizen werden am Dokument angebracht, um beispielsweise auf wichtige Stellen in einem Dokument aufmerksam zu machen, Kommentare zum Inhalt zu geben oder eigene Gedanken zu einem Thema festzuhalten. Sie dienen meist für den späteren Gebrauch, um sich bestimmte Sachverhalte leichter in Erinnerung zu bringen.
- Redigieren: Beim Redigieren wird ein Dokument formal korrigiert, etwa nach den Korrekturvorschriften des „Duden“. Daneben werden in einem Redigierzyklus oftmals auch Anmerkungen zum Ausdruck oder Stil gemacht.
- Inhaltliche Änderungen: Im Gegensatz zum Redigieren sind bei inhaltlichen Änderungen auch Anmerkungen in Form von längeren Texten, Graphiken oder Festbildern wichtig.
- Diskussion: Bei der Arbeit in Gruppen sollte es möglich sein, daß ein Dokument von allen Beteiligten eingesehen werden kann, und daß auf Teile des Dokumentes gezeigt werden kann. Vorgenommene Anmerkungen und Änderungen müssen direkt bei allen Partnern präsent sein.

Die sich aus den Szenarien ergebenden Anforderungen technischerseits können wie folgt zusammengefaßt werden:

- Asynchrone Arbeit muß unterstützt werden
- Der Übergang von asynchroner zu synchroner Arbeit muß ermöglicht werden
- Synchroner Interaktiver Arbeit muß unterstützt werden
- Dokumente können gleichzeitig bearbeiten und annotiert werden
- Annotationen müssen dem Erzeuger zuordbar sein
- Annotationen können an den unterschiedlichsten Standard-Applikationen angebracht werden
- Die Ergebnisse einer Sitzung müssen zu einem späteren Zeitpunkt wieder zur Verfügung stehen
- Bei der Weiterleitung von Dokumenten muß der Gesamtzusammenhalt gewährleistet bleiben

Im folgenden Abschnitt werden diese Anforderungen in Bezug zum Referenzmodell gesetzt und eine Referenzarchitektur abgeleitet. Die Integration der in Abschnitt 5.3 herausgearbeiteten transparenten Annotationen wird im Abschnitt 5.4.3 erläutert und durch die Einbin-

derung von *compound entities*, zusammenhängende Dateneinheiten, komplettiert. Zum Abschluß des Kapitels wird in Abschnitt 5.4.5 anhand eines typischen Arbeitsablauf der Zusammenhang der Komponenten beschrieben.

#### 5.4.2 Systembezogene Referenzarchitektur im Referenzmodell

Die Referenzarchitektur ist in Abbildung 50 dargestellt. Der Aufbau der Architektur ergibt sich konsequent aus dem bereits vorgestellten Referenzmodell und den dargestellten Anforderungen, Lösungsansätzen bzw. Technologien. Es werden die Techniken Workflow, Telekooperation, Annotation und Standard-Applikationen über autonome Software Agenten integriert.

Den zentralen Punkt für den Anwender bildet ein Pool von Standard-Applikationen. Ihm bleibt es hierdurch freigestellt, mit welcher Applikation ein bestimmtes Dokument bearbeitet werden soll. Ein wesentlicher Vorteil dieser Lösung ist, daß eine Weiterentwicklung der Standard-Applikationen automatisch mitberücksichtigt wird. Der diesbezügliche Nachteil von Insellösungen für spezielle Probleme, etwa bei verteilten Gruppeneditoren, tritt hier nicht auf.

Flankiert wird der *Application-Pool* von einer *Workflow-* und einer *Telecooperation-Engine*. Diese Elemente werden in dieser Arbeit nicht detailliert ausgearbeitet. Es werden an dieser Stelle standardisierte Komponenten, wie die der Workflow Management Coalition und der International Telecommunication Union, angenommen. Durch deren Verwendung kann ein hoher Grad an Allgemeinheit erreicht und gleichzeitig auf bereits entwickelte Technologien zurückgegriffen werden.

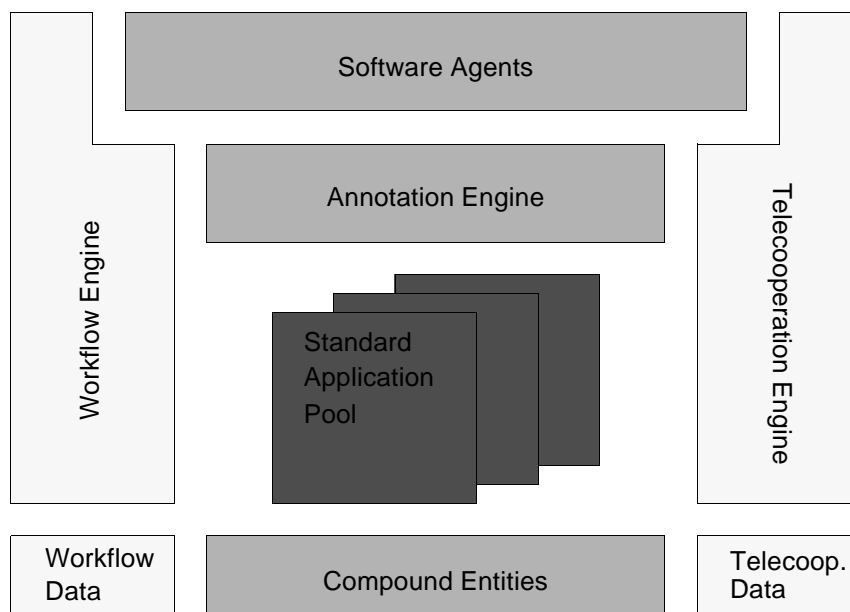


Abbildung 47: Referenzarchitektur einer agentenbasierten, verteilten Verwaltungsumgebung

Oberhalb des *Application-Pools* liegt die *Annotation-Engine*. Sie ist verantwortlich für die Instanziierung und Verwaltung des transparenten Whiteboards. Annotationen können hier im Gegensatz zu herkömmlichen Systemen an beliebigen Anwendungen angebracht werden. Annotationen und Dokument werden zusammen mit einer Legende in *compound entities* zusammengefaßt (siehe Abschnitt 5.4.4). Die *compound entities* können vom Workflow-System wie eine elektronische Akte von einem Anwender zum nächsten weitergeleitet werden.

Die Integration der Telecooperation- und Workflow-Engine erfolgt über die entsprechenden Software Agenten und wurde bereits in Kapitel 4 ausführlich diskutiert. Die Anbindung der Annotation-Engine ist Inhalt des folgenden Abschnitts.

### 5.4.3 Anbindung der Annotation-Engine

Die bisher gemachten Überlegungen führen zu einer Verbindung der Telecooperation-Engine mit der Annotation-Engine, womit indirekt zusätzlich eine Anreicherung von Standard-Applikationen mit CSCW-Funktionalität erreicht wird. Basis sind die Architekturen des transparenten Whiteboards und der Sharing-Komponente.

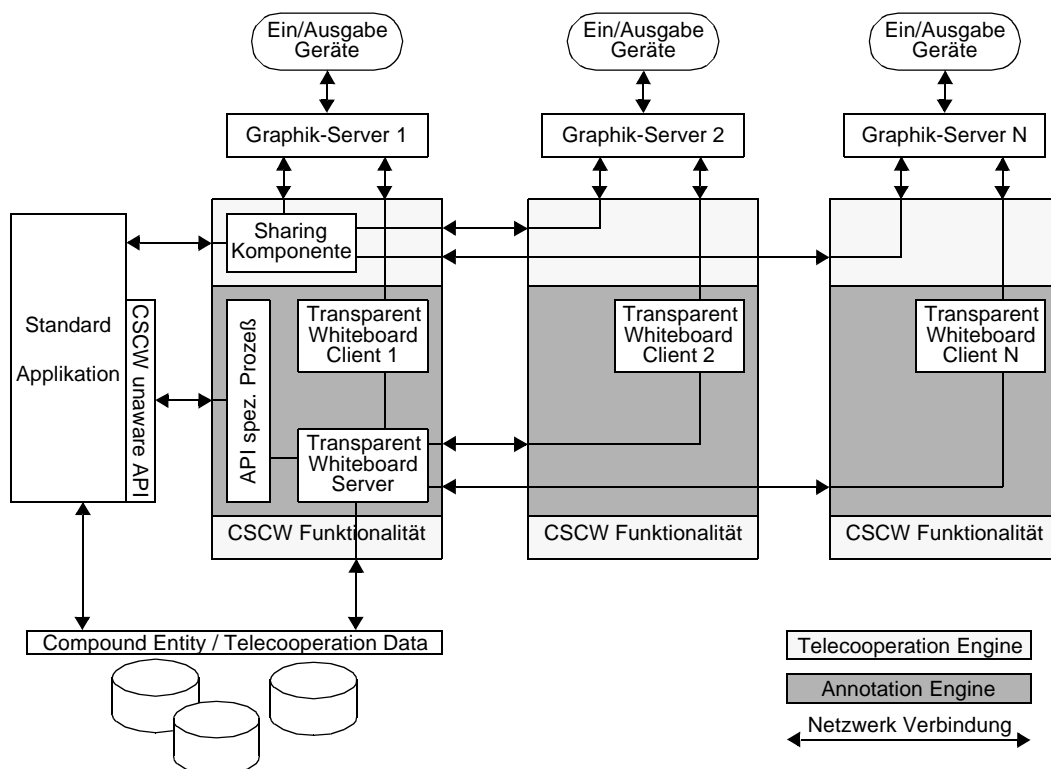


Abbildung 48: Verbindung von Telecooperation-Engine, Annotation-Engine und Standard-Applikationen

Abbildung 48 zeigt den konzeptionellen Aufbau. Die hellgraue Telecooperation-Engine bietet notwendige CSCW-Funktionalitäten an. Im Detail ist hier das *application sharing*

dargestellt, welches auf die Ein- und Ausgabe der Standard Applikation Einfluß nimmt. Die Annotation-Engine, dunkelgrau dargestellt, überlagert die Applikation durch ein transparentes Whiteboard und ermöglicht die Semantische Verknüpfung über die links im Bild dargestellte API der Standard-Applikation und eine eigene applikationsspezifische Komponente. Die Compound-Entity mit Dokument und Annotationen sowie Groupware spezifische Daten werden in einem lokalen Datenrepositorium gehalten.

#### 5.4.4 Compound Entities

Gerade für die Weiterleitung, die durch die *Workflow Engine* realisiert wird, ist es erforderlich, daß Dokument und Annotationen zusammengehalten werden. Eine Linkstruktur ermöglicht die Verbindung als eine einzige komposite Entität. Die Zuordnung kann über die oben eingeführte Semantische Verknüpfung bestimmt werden. Eine solche Sichtweise erlaubt die Weiterleitung von Dokument und Annotationen als Einheit. Der Aufbau einer solchen Entität kann mit einer Aktenmappe verglichen werden. Sie enthält Dokument, Textanmerkungen und alle Annotationen, wie graphische Primitive, Standbilder und Audioannotationen. Zusätzlich werden für die Protokollierung Daten in Form einer Legende gehalten. Dies beinhaltet Zuordnungen von Annotationen zu Personen und Zeitstempel. In Abbildung 47 wurde diese Einheit mit *Compound Entity* bezeichnet. Die folgende Abbildung 49 zeigt die Verbindung von *Workflow*-, *Annotation-Engine* und dem *Standard Applikation Pool* über *Compound Entities*. Als zentrale Einheit werden sie von der *Annotation Engine* verwaltet. Im einzelnen erzeugt und bearbeitet die *Annotation Engine* Annotationen und Legende einer *Compound Entity*. Das dazugehörige Dokument wird von einer Standard Applikation bearbeitet. Die asynchrone Weiterleitung an Teilnehmer eines Arbeitsprozesses übernimmt die *Workflow Engine*.

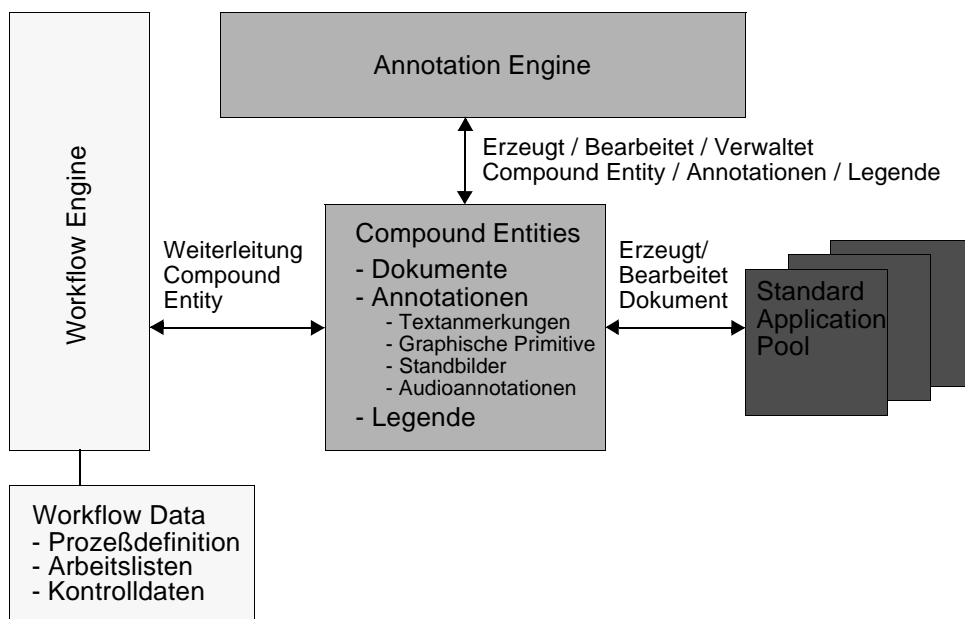


Abbildung 49: Verbindung von Workflow-, Annotation-Engine und Standard Applikation durch Compound Entities

### 5.4.5 Arbeitsablauf innerhalb der Referenzarchitektur

Ein Ausschnitt aus dem im Abschnitt 3.2 gegebenen Szenario soll an dieser Stelle verwendet werden, um anhand eines typischen Arbeitsablaufs im Büro die einzelnen Komponenten der Referenzarchitektur näher zu beschreiben. Das Szenario ist in Tabelle 7 wiedergegeben und wurde leicht verändert, um besser die Möglichkeiten der Referenzarchitektur aufzuzeigen.

Drei technische Vorgänge sind in dem Szenario zu unterscheiden:

1. Die Erstellung eines annotierten Dokumentes (Akte) und dessen Weiterleitung
2. Der Aufbau einer Kommunikation (Videokonferenz); anschließende Diskussion am Dokument; Konferenzabbau und Rückführung des kommentierten Dokumentes
3. Bearbeitung eines bereits annotierten Dokumentes

Beteiligte Personen:

Herr A. – Antragsteller eines Forschungsprojektes

Sachbearbeiter und Vorgesetzter der Abteilung III C der TU-Verwaltung

Beschreibung des Ablaufs	Klassifikation	Komponenten
Herr A. schickt das Formular 1502 für eine Forschungsanzeige ausgefüllt an die Abteilung IIIC.		
Ein Sachbearbeiter registriert den Eingang und prüft das Formular 1502 am Bildschirm auf formale und inhaltliche Korrektheit und markiert verbesserungswürdige und unklare Stellen im Dokument. Er holt den Standardbrief für Stellungnahmen zu einem Drittmittelantrag auf den Bildschirm und füllt ihn aus. Der Sachbearbeiter leitet nach Fertigstellung die Akte bestehend aus Antrag und Stellungnahme an den Vorgesetzten weiter.	Eingang erhalten/ Dokumentenbearbeitung Dokumentenbearbeitung Stellungnahme anfordern	WFE <sup>a</sup> , StA, SA, AE, CE
Der Vorgesetzte öffnet die Akte und überprüft Antrag und Stellungnahme und stellt Unklarheiten über die angegebene Kalkulation "Sachmittel" und "Raumbedarf" im Formular 1502 fest.	Dokumentenbearbeitung	WFE, StA
Der Vorgesetzte verlangt eine Rücksprache mit dem zuständigen Sachbearbeiter.	Terminabsprache	SA

Tabelle 7: Bearbeitung eines Drittmittelantrags

Beschreibung des Ablaufs	Klassifikation	Komponenten
Weil das System dies derzeit für die beste Kommunikationsmöglichkeit hält, wird eine Videokonferenz zwecks Rücksprache und Klärung der offenen Fragen gestartet. Das Formular 1502 von Herrn A. wird verteilt dargestellt. Der Vorgesetzte markiert die fraglichen Stellen.	Rücksprache Abstimmung in kleinen Gruppen	SA, TCE, StA, AE, CE
Die Videokonferenz wird beendet. Die Akte wird mit kommentiertem Antrag und Stellungnahme zurück an den Sachbearbeiter geleitet.	Arbeitsergebnis mitteilen	SA, CE, WFE
Der Sachbearbeiter bearbeitet die Akte weiter.	Dokumentenbearbeitung	WFE, SA, CE, StA, AE

**Tabelle 7: Bearbeitung eines Drittmittelantrags**

a. WFE: Workflow Engine; StA: Standard Application; SA: Software Agents; AE: Annotation Engine; CE: Compound Entity; TCE: Telecooperation Engine.

In der Abbildung 50 wird der Ablauf dieses Szenarios innerhalb der Referenzarchitektur dargestellt. Um die Funktionsweise der Architektur und die Datenbereiche (Workflow-Data; Compound-Entity; Telecooperation-Data) zu erläutern, werden die einzelnen Aktionen innerhalb der drei technischen Vorgänge einzeln beschrieben (vergleiche dazu auch Abschnitt 4.3). Die Startpunkte 1, 9 und 16 der drei Vorgänge sind in der Abbildung zur besseren Übersicht hervorgehoben.

1. Die Workflow-Engine lädt die aktuelle Arbeitsliste des Sachbearbeiters aus ihrem Datenbereich (A) und stellt ihm die entsprechenden Daten dar. Der Sachbearbeiter wählt den Vorgang von Herrn A. und öffnet das dazugehörige Dokument (Formular 1502). Eine Standard-Applikation wird dazu verwendet (B).
2. Der Sachbearbeiter meldet per Knopfdruck den Wunsch an, das Dokument zu annotieren. Der Annotations-Agent wird aktiviert.
3. Der Annotations-Agent erfragt mit Hilfe des Workflow-Agenten über das *Workflow Client Application Interface*, Interface 2 der WfMC, bei der Workflow-Engine Daten der aktuellen Arbeitsliste.
4. Die Informationen (TDS) werden gefiltert und an die Annotation-Engine weitergeleitet. Falls sie noch nicht gestartet war, wird sie vom Annotations-Agenten aktiviert.
5. Die Annotation-Engine ermöglicht dem Sachbearbeiter das Annotieren des Dokumentes, welches mit der Standard-Applikation gleichzeitig bearbeitet werden kann. Dazu verwendet die Annotation-Engine spezielle Mechanismen auf der Standard-Applikation an (C), die in Abschnitt 5.3.1 und 5.3.2 beschrieben wurden.
6. Eine *Compound-Entity* wird bestehend aus Dokument, Annotationen und Legende von der Annotation-Engine erstellt.

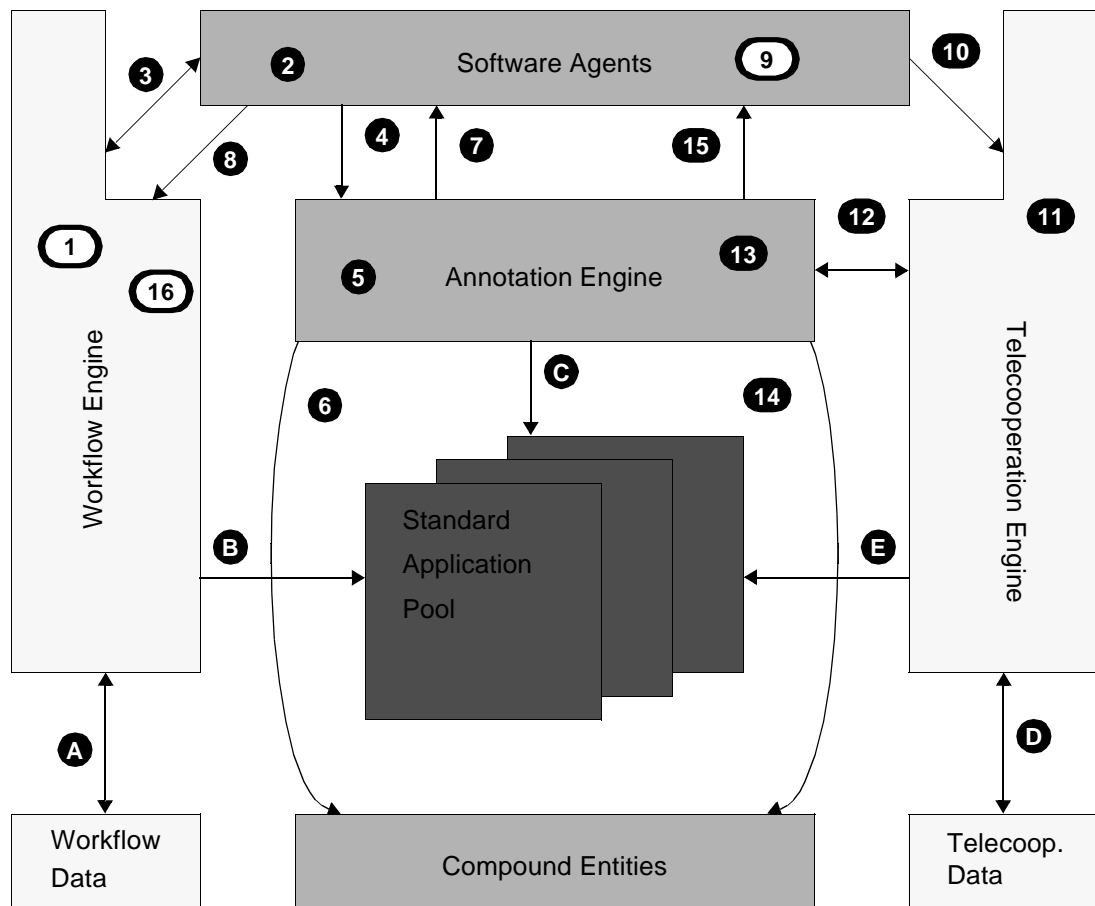


Abbildung 50: Arbeitsablauf innerhalb der Referenzarchitektur

7. Die *Compound-Entity* wird nach beenden der Sitzung von der Annotation-Engine an den Annotations-Agenten zurückgereicht.
8. Der Agent ersetzt mit Hilfe des Workflow-Agenten über das Interface 2 der WfMC das Dokument in der Arbeitsliste des Sachbearbeiters durch die *Compound-Entity* unter demselben Namen.  
Die Akte wird weitergeleitet und vom Vorgesetzten geöffnet und begutachtet (Punkte 1 bzw. 16; s.u.)
9. Der Vorgesetzte verlangt eine Rücksprache mit dem zuständigen Sachbearbeiter. Der Telekooperations-Agent wird aktiviert. Der Agent erfragt, wiederum mit Hilfe des Workflow-Agenten, bei der Workflow-Engine die Daten der aktuellen Arbeitsliste (wie Punkt 3). Die Daten werden gefiltert und der entsprechende Sachbearbeiter wird ermittelt. Der Agent nimmt Kontakt mit dem TK-Agenten des Sachbearbeiters auf, den er über den Administrator-Agenten ermittelt hat. Es wird ein Kommunikationsmittel und mit Hilfe der Termin-Agenten ein Kommunikationszeitpunkt (unter Umständen durch Rückfrage beim jeweiligen Nutzer) verhandelt. In diesem Beispiel ist das Ergebnis der Verhandlungen eine sofortige Videokonferenz. Das Ergebnis kann auch eine andere

Kommunikationsart, wie etwa ein elektronischer Anrufbeantworter, Telefon, E-Mail oder Fax, oder ein anderer Termin für eine Videokonferenz (die dann zum gewählten Zeitpunkt automatisch initiiert werden würde) oder ein physisches Treffen sein. Die Wahl ist nur beschränkt durch die technischen Möglichkeiten der Teilnehmer einer Kommunikation.

10. Die Daten zur Konferenz werden vom TK-Agenten über den *GCC Control Services Access Point* des T.120 an die Telecooperation-Engine weitergeleitet.
11. Die Telecooperation-Engine versorgt sich unter Umständen mit weiteren Daten, etwa Kommunikationswege, aus ihrem Datenbereich (D) und baut die Konferenz auf. Die Standard-Applikation wird verteilt dargestellt (application sharing).
12. Die beidseitigen Annotation-Engines werden mit den notwendigen Daten (welches Dokument soll annotiert werden, welche Standard Applikation wird verwendet) von ihren Annotations-Agenten versorgt.
13. Die Annotation-Engine der initiiierenden Seite startet eine kooperative Sitzung. Die Teilnehmer der Konferenz können Anmerkungen am verteilten Dokument anbringen.
14. Eine Compound-Entity, bestehend aus Dokument, Annotationen und Legende, wird von der Annotation-Engine erstellt. Die Zuordnung von wem eine Annotation wann gemacht wurde ist Teil der Legende und wird mitprotokolliert.
15. Das Ergebnis der Sitzung wird nach Beendigung der Konferenz dem Annotations-Agenten des Initiators übergeben. Dieser übergibt die Compound Entity dem Workflow-Agenten, der sie in der Arbeitsliste der Workflow Engine einträgt (wie in Punkt 8).
16. Die Akte wird an den Sachbearbeiter weitergereicht, der die Bearbeitung fortsetzt. Beim Öffnen der Akte, die eine Compound Entity enthält, wird der Annotations-Agent aktiviert. Dieser übernimmt die Entity (Punkt 2) und reicht sie an die Annotation-Engine (Punkt 4). Die Annotation-Engine startet die zum Dokument passende Standard-Applikation und lädt die Annotationen durch die in Abschnitt 5.4.4 beschriebenen Mechanismen (5 und C). Der weitere Ablauf entspricht dann den Punkten 6 bis 8.

Der dargestellte Ablauf macht das Zusammenspiel der Software Agenten mit den eingesetzten Werkzeugen Telecooperation-, Workflow- und Annotation-Engine deutlich. Ersichtlich wird auch, wie die Agenten zu unterschiedlichen Zwecken eingesetzt werden. Daten werden von Agent zu Agent weitergereicht und die Informationen werden genutzt, um eine optimale Führung des Benutzer durch die komplexen Systemstrukturen zu gewährleisten. Die Umsetzung vom Referenzmodell in eine systembezogene Referenzarchitektur zeigt damit deren Realisierungsmöglichkeit auf.

## 5.5 Zusammenfassung

Die synchrone Interaktion mit dem Fokus des Interesses innerhalb einer kooperativen Telekonferenz gilt als Schlüsselfaktor im Bereich der computerunterstützten Gruppenarbeit.



Dem Benutzer sind Audio/Video-Konferenzsituationen aus dem täglichen Leben in ähnlicher Form (Telefon, informelle Diskussion) bekannt, und es ist anzunehmen, daß diese schnell adaptiert werden können. Die Arbeit mit synchronen CSCW-Werkzeugen, wie etwa *Whiteboards* oder *Sharing* Komponenten, ist dagegen neu. Innerhalb einer Konferenzsituation muß sich die Unterstützung auf den Fokus des Interesses der Anwendergruppe konzentrieren. Dieser Teilbereich der Mikro-Level-Transparenz war das Hauptthema dieses Kapitels.

Zunächst wurde dazu die aktive und die passive Interaktion mit dem Fokus des Interesses (FDI) definiert. Die aktive Interaktion ist dabei durch die Manipulation des FDI durch einen Benutzer festgelegt worden; die passive Interaktion hingegen durch die Annotation des FDI durch einen Benutzer. Eine Annotation steht dabei immer in einer eindeutigen Korrelation zur FDI repräsentierenden Entität.

Basierend auf den beiden Technologien Whiteboard und Application Sharing ist das Konzept der transparenten Whiteboards entwickelt worden. Als Hintergrund des Whiteboards wird hier die durch eine Sharing-Komponente verteilte Ausgabe einer beliebigen Applikation verwendet. Das transparente Whiteboard überlagert die Ausgabe der Applikation und ermöglicht den Sitzungsteilnehmern so das Annotieren der Applikation. Zusätzlich ist das Whiteboard durchlässig für Aktionen des Benutzers, die der Applikation gelten. Dieses Konzept erlaubt sowohl aktive als auch passive Interaktion mit dem FDI.

Problematisch an diesem Lösungsansatz ist allerdings, daß gewährleistet bleiben muß, daß eine Annotation sich nicht unabsichtlich relativ zum annotierten Objekt der Applikation verschiebt. Der Betrachter bringt eine Annotation mittels deren Position in Relation zu einem Text- oder Graphikobjekt. Diese Probleme treten beispielsweise beim Einfügen von neuem Text auf. Um dieses Problem zu lösen, ist die Semantische Verknüpfung hergeleitet und spezifiziert worden. Mit Hilfe der hier entwickelten Technik des *semantic tagging* kann eine feste Relation zwischen Annotation und annotierter Entität erreicht werden.

Aus den Möglichkeiten des *semantic tagging* sind weiterhin dynamische Annotationen, die sowohl ihre Position als auch ihre Form an das annotierte Objekt dynamisch anpassen können, entwickelt worden.

Diese Konzepte wurden anschließend in das Referenzmodell einbezogen. Resultierend wurde eine systembezogene Referenzarchitektur für eine kooperative Verwaltungsumgebung innerhalb des Referenzmodells erstellt. Der Aufbau der Architektur beinhaltet die bereits vorgestellten Techniken zu Workflow, Telekooperation und Software-Agenten. Hinzu kommt das in diesem Kapitel erarbeitete Konzept eines generischen Annotationswerkzeuges und die Zusammenfassung von Objekten und Annotationen in Form einer Compound Entity.

Um die Zusammenhänge innerhalb der Referenzarchitektur aufzuzeigen, sind anhand der Beschreibung des Arbeitsablaufs eines konkreten Beispiels aus der Verwaltung die technische Vorgänge beschrieben worden. Dabei ist zunächst die Erstellung eines annotierten Dokumentes (Akte) und dessen Weiterleitung aufgezeigt worden. In einem zweiten Schritt wurde der Aufbau einer Kommunikation (Videokonferenz) und anschließender Diskussion

am Dokument, einschließlich dem Konferenzabbau und der Rückführung des kommentierten Dokumentes in das Workflow-System beschrieben. Abschließend wurde die Bearbeitung eines bereits annotierten Dokumentes dargestellt.

## Kapitel 6

### Realisierungen

#### 6.1 Einleitung

Zur Überprüfung der hier vorgestellten Konzepte des agentenbasierten Referenzmodells und der synchronen Interaktion mit dem Fokus des Interesses sind zwei Prototypen entwickelt worden. Der erste Prototyp realisiert ein transparentes Whiteboard und erlaubt die Integration von Annotationen in einer applikationsunabhängigen Form und die damit verbundene synchrone Interaktion mit dem FDI. Dieser Prototyp - CrystalPad - wird in Abschnitt 6.3 beschrieben. Der zweite Prototyp implementiert das Referenzmodell anhand der systembezogenen Referenzarchitektur. Dazu sind Software Agenten für die Bereiche Telekooperation, Workflow und Terminkalender entwickelt worden. Zusätzlich ist zu Demonstrationszwecken eine Anbindung an eine *Computer-Telephone-Integration* Lösung durchgeführt worden. Das System mit dem Namen *Virtue* wird in Abschnitt 6.4 erläutert. Zuvor sollen die Inhalte des Projektes „Mensch-Maschine-Interaktion im multimedialen Verwaltungsbüro“ dargelegt werden, da sie als Grundlage für die Arbeiten im Bereich „Transparenz“ dienen.

#### 6.2 Mensch-Maschine-Interaktion im multimedialen Verwaltungsbüro

Das innerhalb des Berkom-Programms angesiedelte und von der Deutsche-Telekom geförderte Projekt [10][111] hat sich zum Ziel gesetzt, durch die Entwicklung, Pilotierung und Evaluierung einer intelligenten Mensch-Maschine Schnittstelle eine deutliche Verbesserung der Oberflächengestaltung für Telekooperationsanwendungen zu schaffen. Dazu werden Dienste aus den Bereichen Multimodalität, Sprachverarbeitung, taktile MMI-Verfahren, wie Stifteingabe und Touchscreen, adaptive MMI-Agents und „Flexibilität und Transparenz“ (diesen Bereich bearbeitete der Autor) konzipiert und implementiert. Diese werden in realen Anwendungsszenarien aus den Bereichen der Vorgangsbearbeitung und der synchronen Telekooperation in der Bürokommunikation öffentlicher Verwaltungen evaluiert. Durch ein Integrationskonzept wird die Vereinigung der einzelnen Dienste zu einem gesamten MMI-System ermöglicht. Arbeits- und kognitionspsychologische Kriterien, die Entwicklung und Nutzung eines Qualitätssicherungs- und Optimierungskonzeptes sowie die Erstellung von Richtlinien und Kriterien sichern einen realitätsnahen Bezug.

Um diese Ziele zu erreichen wurde in den ersten beiden Projektphasen zunächst eine Grob- bzw. eine Feinspezifikation angelegt. Die einzelnen Dienste des MMI-Systems, „Sprachverarbeitung“, „Stifteingabe/Touchscreen“, „Adaptivität von Benutzungsschnittstellen“, „Multimodalität“, „Flexibilität und Transparenz“ und „Liveboard“ sind einzeln vorgestellt und inklusive einer Beschreibung der jeweiligen Anwendungsszenarien spezifiziert worden. Der Dienst „Flexibilität und Transparenz“ umfaßte dabei erste Ansätze zur Realisierung von Makro-Level-Transparenz.

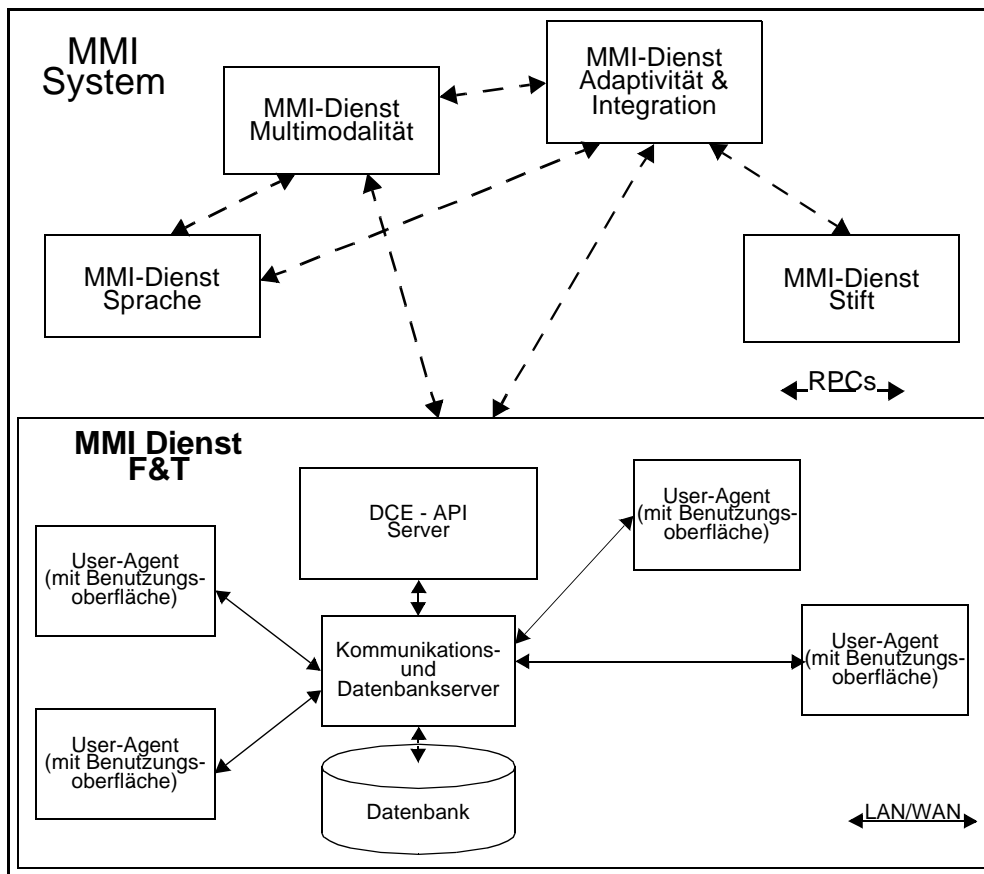


Abbildung 51: Prozessstruktur des F&T Dienstes im MMI-System

Um eine spätere Integration der Dienste zu erleichtern wurde mit einer einheitlichen Dienstarchitektur ein Rahmen geschaffen, der die Grundlage des Integrationskonzeptes bildet. Eine Untersuchung verschiedener Industriestandards für Dienstplattformen wurde durchgeführt, um u. a. eine Basis für das gesamte MMI-System und seine einzelnen Dienste zu finden. Es wurde sich hier für die Dienstplattformen DCE entschieden. Die Anwendungsfelder ODP, IN und TMN wurden ebenso analysiert und auf ihre Relevanz zu MMI hin beleuchtet, wie ein Überblick zum Stand der Technik der einzelnen Dienste Aufschluß darüber gab, inwieweit bereits vorhandene Konzepte oder Produkte direkt im MMI-System verwendbar bzw. einsetzbar sind. Ein Kriterienkatalog zur Gestaltung und Evaluierung der MMI-Dienste wurde aufgestellt, um nach der Gegenüberstellung verschiedener Methoden für eine

Arbeitsplatzanalyse, ein Konzept für eine geeignete Analyse der Arbeitsplätze im Verwaltungsbüro vorzuschlagen. Um die Problematiken der Qualitätssicherung nicht zu vernachlässigen, wurde eine Einführungsstrategie definiert und in einem Qualitäts- und Optimierungskatalog sowohl die qualitativen Probleme der Softwareerstellung als auch die möglichen Lösungen erläutert.

In der dritten Projektphase war die Hauptaufgabe die Definition der Schnittstellen und die Implementierung der ersten Prototypen der MMI-Dienste. Abbildung 51 zeigt den Dienst „Flexibilität und Transparenz“ (F&T Dienst) innerhalb des MMI-Systems. Erkennbar sind hier erste Ansätze zur Anbindung von Systemkomponenten mit Hilfe von Software Agenten.

Weitere Schwerpunkte waren neben der Erweiterung des Integrationskonzeptes die Erstellung des software-ergonomischen Gesamtkatalogs sowie die Analyse der Benutzeranforderungen und Profile im modernen Verwaltungsbüro. Die entwickelten Prototypen wurden dann in der anschließenden vierten Projektphase evaluiert. Es wurden zunächst Tests mit freiwillige Testpersonen (meist aus dem Hochschulumfeld der einzelnen Partner) durchgeführt. Gleichzeitig wurde der jeweilige Anwender mit dem System vertraut gemacht und in die Bedienung eingeführt, damit er anschließend den Dienst testen und seine Beurteilung abgeben konnte. Die Auswertung der Evaluierung führte zu Redesignvorschlägen bei den einzelnen Diensten, die zu einer Verbesserung der software-ergonomischen Akzeptanz beim Benutzer führen sollten. Parallel wurden die Prototypen selbst weiterentwickelt und am Ende der vierten Projektphase in einer integrierten Form als MMI-System vorgestellt. Schwerpunkt des fünften Projektabschnittes war die Realisierung der Redesignvorschläge und die Evaluierungsphase II. Durch eine selbstkritische Analyse der Dienste und des Hauptszenarios konnte das Zusammenspiel der Dienste im gesamten MMI-System verbessert, eine Reduzierung der Plattformen erreicht und das Auftreten der Dienste im Szenario konkretisiert werden. Die dadurch entstehende Vermaschung der MMI-Dienste kann als weiterer Schritt der Integration des MMI-Systems betrachtet werden. Im letzten Abschnitt des Projektes wurden letzte Anpassungen vorgenommen und der Demonstrator des MMI-Systems fertiggestellt. Dieser wurde in einem verteilten Szenario erfolgreich getestet.

## 6.3 CrystalPad

### 6.3.1 Übersicht

CrystalPad [105][107][109] ist eine verteilte computergestützte Annotationsumgebung, die Gruppendiskussionen auf beliebigen Anwendungen und damit die synchrone Interaktion mit dem FDI unterstützt. Wie ein Satz durchsichtiger Overheadfolien überlagert CrystalPad die jeweilige Anwendung; jeder Teilnehmer erhält seine eigene Folie, auf der er seine Anmerkungen in der ihm zugeordneten Farbe skizziert. CrystalPad ist in Telekonferenz-Systeme integriert, so daß mehrere örtlich verteilte Konferenzteilnehmer gleichzeitig an derselben Anwendung arbeiten als auch darauf annotieren können. Abbildung 52 zeigt einen Schnappschuß der Umgebung vom Bildschirm.

CrystalPad bietet die Möglichkeit der aktiven und passiven Interaktion mit dem Fokus des Interesses. Dies wurde durch die Entwicklung spezieller Mechanismen, die für unterschiedliche graphische Benutzungsoberflächensysteme umgesetzt wurden, erreicht (siehe Abschnitt 6.3.3). Um beispielhaft die Semantische Verknüpfung zu realisieren wurde für das Desktop-Publishing System FrameMaker® eine spezielle Anbindung, wie in Abschnitt 5.3.2 spezifiziert, realisiert. Anmerkungen können innerhalb einer Konferenz zusammen mit Konferenzpartnern getätigt werden, ohne daß diese vor Ort sein müssen. Die Ergebnisse einer Diskussion können für spätere Sitzungen oder zur Weiterverarbeitung abgespeichert werden.

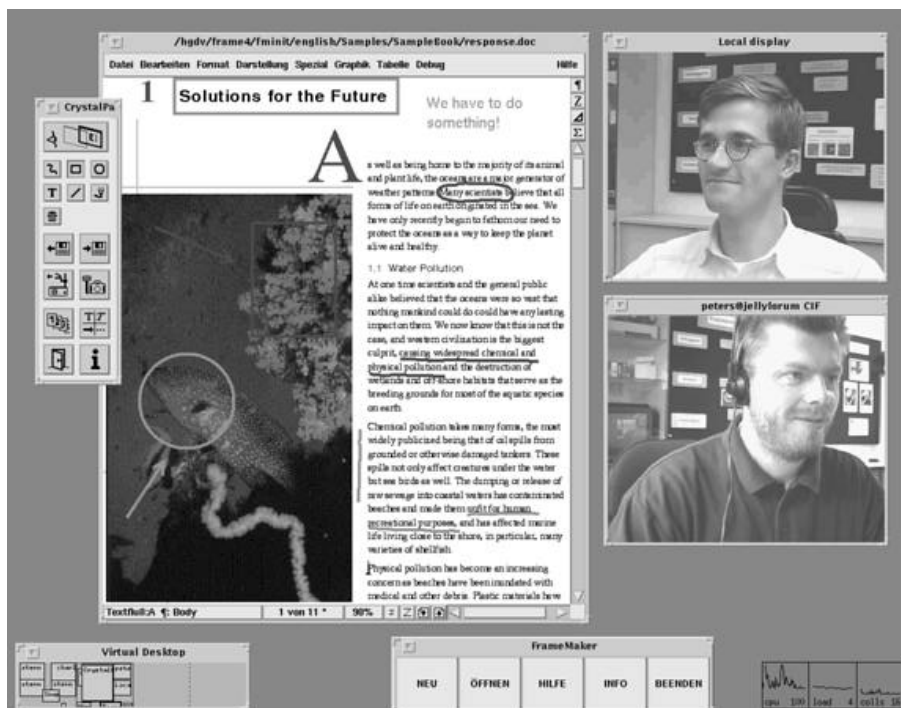


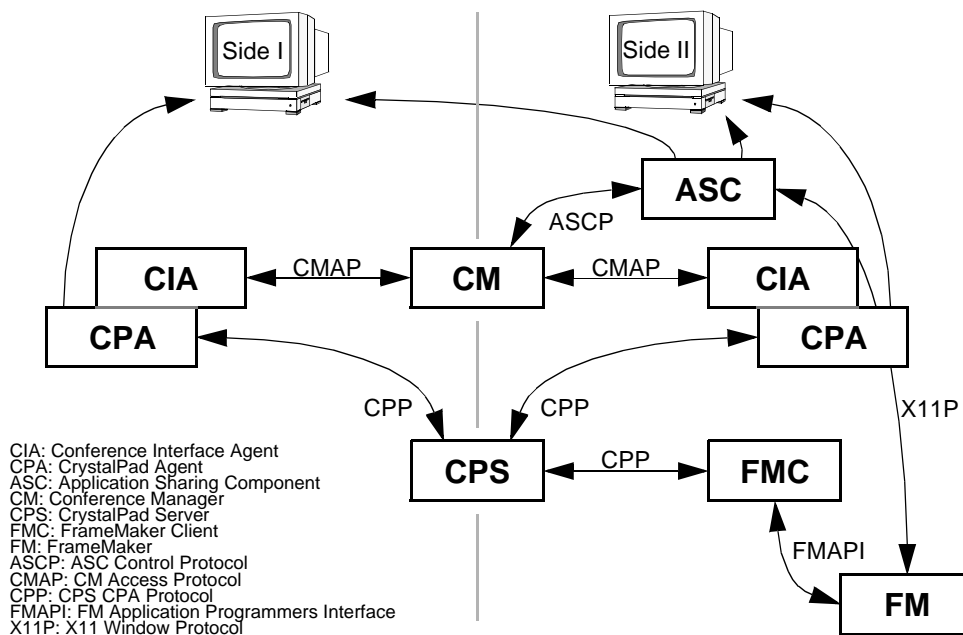
Abbildung 52: CrystalPad-Umgebung

CrystalPad bietet neben graphischen und textuellen Annotationen die Möglichkeit Sprache, längere Texte, Rasterbilder oder auch (in Zukunft) Bewegtbilder als Annotation in das Dokument miteinzubringen. Im einzelnen besitzt CrystalPad die folgende Funktionalität:

- Kooperative Diskussion mit Annotationen
- Annotationen mit unterschiedlichen Medien
- WYSIWIS: What You See Is What I See
- Simultane Interaktion aller Anwender
- Annotationen direkt auf der Desktop-Anwendung (passive Interaktion mit FDI)
- Desktop-Anwendung bleibt normal benutzbar (aktive Interaktion mit FDI)
- Semantische Verknüpfung mit Desktop-Anwendung
- Integration in Konferenzumgebung

CrystalPad ist in GroupX, dem BERKOM kompatiblen Multi Media Collaboration System von SIEMENS, eingebunden. Dies ermöglicht eine einfache Handhabung und einen leichten Zugriff auf die Annotationsmöglichkeiten von CrystalPad innerhalb der MMC Umgebung. SIEMENS setzt dieses System im POLIKOM Projekt POLIVest ein. Dort werden innovative Telekooperationstechniken bei Beratungsverfahren im Bundesrat und Baugenehmigungsverfahren realisiert (siehe dazu auch Abschnitt 7.2.2).

Abbildung 53 zeigt die Prozeßstruktur und die Kommunikationswege der Integration von CrystalPad in die BERKOM MMC Umgebung mit FrameMaker als verteilte Applikation. Zur Integration wird CrystalPad als Bibliothek angeboten, die in den *conference interface agent* (CIA) eingebunden wird. Der CIA kommuniziert dabei über das *conference manager access protocol* (CMAP) mit dem *conference manager* (CM). Mit ihm sind weitere CIAs innerhalb einer Sitzung verbunden. Der CM ist gleichzeitig mit der *application sharing component* (ASC) verbunden, die für die Verteilung der Applikationen zuständig ist. Aus Gründen der Übersichtlichkeit sind die weiteren Komponenten der BERKOM MMC Umgebung, wie *conference directory*, *audio video manager*, usw., in Abbildung 53 nicht dargestellt (vergleiche dazu Abschnitt 2.3.2).



**Abbildung 53: Kommunikation in der BERKOM-MMC Umgebung mit CrystalPad**

Die Applikation FrameMaker (FM) wird nun mittels der ASC über das X11 Protokoll (X11P) verteilt. Der CIA der jeweiligen Seite teilt dem entsprechenden CrystalPad-Agent (CPA) mit, auf welchem Fenster zu annotieren ist. Der CPA, der über den CrystalPad-Server (CPS) mit dem FrameMaker-Client (FMC) kommuniziert, kann jetzt auf das Erzeugen von Annotationen reagieren und den FMC benachrichtigen. Dem gegenüber kann der FMC Änderungen am Dokument, die er über die FrameMaker-API (FMAPI) erkennt, dem CPA

übermitteln. Der gegenseitige Austausch von Informationen findet über das interne CrystalPad Protokoll (CPP) statt.

### 6.3.2 Agieren und Annotieren

Für X11-basierte Plattformen wurde eine Lösung für das Problem des gleichzeitigen Bearbeitens und Annotierens durch Verwendung der *X-Shape*-Extension gefunden und realisiert. Die *X-Shape* Extension ermöglicht es, nicht-rechteckige Fenster (*window*) zu erzeugen. Dazu kann eine Bitmap als Maske verwendet werden. Diese wird, bildlich gesprochen, über das ursprünglich rechteckige Fenster gelegt. Sichtbar bleibt dann vom Fenster nur der Teil, der nicht von der Maske schattiert wird. Damit ist es möglich, dem Fenster jegliche Form, insbesondere auch konkave und mit Ausschnitten behaftete, zu geben. Die Operationen dazu werden im X-Server ausgeführt. Eine Folge davon ist, daß auch der Window-Manager mit diesen nicht-rechteckigen *Windows* bzw. *Widgets* arbeitet. Eingabeereignisse können durch Ausschnitte hindurch zu den dahinterliegenden Event-Anwärttern gereicht werden.

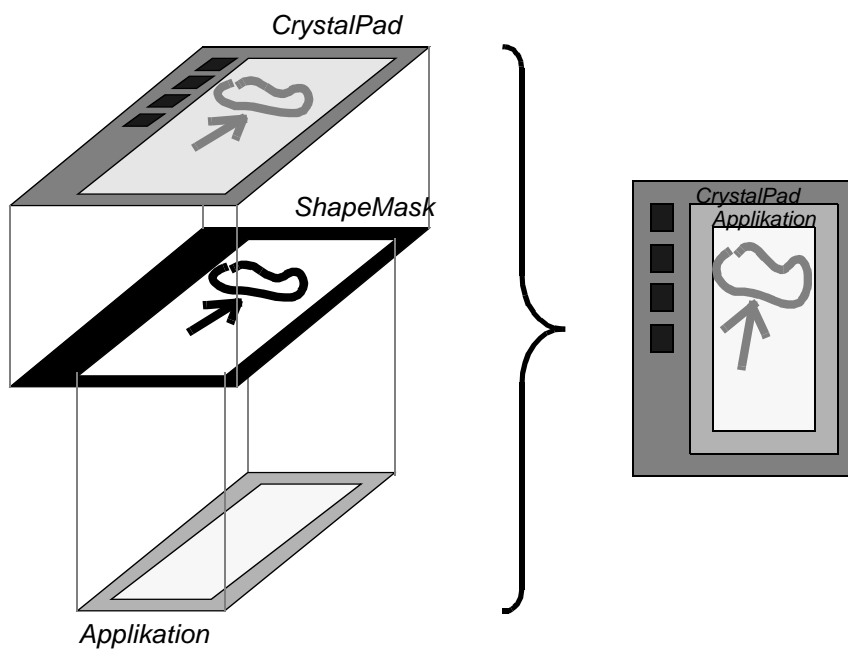


Abbildung 54: Annotation über X-Shape Extension

Diese Eigenschaften wurde zur Lösung des Basisproblems zur passiven und aktiven Interaktion mit dem FDI benutzt. Wird ein Ausschnitt von der Größe der unterliegenden Applikation in CrystalPad verwendet und CrystalPad entsprechend über der Applikation positioniert, besitzt man einen sowohl durchsichtigen als auch durchlässigen Arbeitsbereich. Ist CrystalPad im Zustand des Annotierens, wird die Maske entsprechend der Annotationen verändert, so daß die Annotationen sichtbar werden. Abbildung 54 verdeutlicht dies. In diesem Zustand wird dem X-Server mitgeteilt, daß die Eingabe von Maus und Tastatur zwingend an CrystalPad weitergereicht wird und nicht durch den Ausschnitt hindurch. Um mit der unterliegenden Applikation zu arbeiten, wird dem X-Server mitgeteilt,



wieder „normal“ mit eingehenden Ereignissen umzugehen. Durch die Ausschnitte in CrystalPad kann der Benutzer wie gewohnt mit der unterliegenden Anwendung arbeiten.

Die Prinzipien können auf die MS-Windows Plattformen übertragen werden. Die Mechanismen der X-Shape-Extension stehen hier allerdings nicht zur Verfügung. Durch die Verwendung des Subclassing-Mechanismus konnte hier ein Ersatz gefunden werden. Durch Subclassing ist es möglich, das Verhalten eines Fensters zu modifizieren. Das Verhalten des Fensters wird durch die Message-Behandlungsroutine bestimmt. Die Modifikation des Fensterverhaltens besteht darin, für bestimmte Messages vor, nach oder anstelle dieser Behandlungsroutine einzugreifen. Dazu wird eine neue Behandlungsroutine benutzt, die, abgesehen von den gewünschten Veränderungen, die ursprüngliche Behandlungsroutine aufruft. Die Annotationen werden nun zunächst in eine Pixmap ausgegeben, um eine Maske der Annotationen zu erzeugen. Über die Umkehrung dieser Maske kann die Pixmap ausgegeben werden. Beim Auslesen werden mit dieser Maske die Annotationen entfernt und das Originalbild der überlagerten Applikation ausgegeben.

### 6.3.3 Semantische Verknüpfung

Um die Annotationen mit dem Text dynamisch „mitfließen“ zu lassen, ist das Prinzip des *semantic tagging* entwickelt worden (vergleiche Abschnitt 5.3.2). Es ermöglicht eine Verknüpfung zwischen einer von CrystalPad erzeugten Annotation und einem Applikationsobjekt. Die Semantische Verknüpfung wird am DTP-System FrameMaker® über dessen API und dem Frame Developer's Kit implementiert. Über einen speziellen FrameMaker Client, der in Verbindung zu FrameMaker und dem CrystalPad-Server steht, werden für die Semantische Verknüpfung notwendige Informationen abgefragt und weitergeleitet (siehe Abbildung 53). Zusätzlich wird durch die API die Möglichkeit gegeben, Annotationen und Dokument zu Dokumentationszwecken, zur Weiterverarbeitung oder für spätere Sitzungen abzuspeichern.

Als Anker für eine Annotation wird das FrameMaker-Objekt „Verankerter Rahmen“ verwendet. Der „Verankerte Rahmen“ wird in ein Dokument eingefügt, indem man einen Anker im Textfluß *verankert*. Während der Anker mit dem Text mitfließt, wird der Rahmen an die nächste freie Stelle plaziert. Erzeugt man einen Rahmen mit einer minimalen Größe von etwa einem Mikrometer Seitenlänge, so befindet sich der Rahmen exakt an der Stelle des Ankers und fließt so mit dem Text mit. Wegen seiner geringen Größe ist der Rahmen unsichtbar und beeinflusst das Seitenlayout nicht. Zudem ist es, wie bei allen Graphikobjekten möglich, vom „Verankerten Rahmen“ auf die absolute Position innerhalb der Seite zu gelangen. Über die FrameMaker API werden nun die zur Verankerung einer Annotation notwendigen Daten ausgetauscht bzw. wieder abgefragt. Durch den in FrameMaker vorhandenen Mechanismus zur Notifikation wird der FrameMaker-Client informiert sobald eine Veränderung des Textes eintritt und die Position von Annotationen angepaßt werden muß. Diese Notifikation wird gefiltert und an CrystalPad weitergeleitet. Dort findet dann die eigentliche Verschiebung einer Annotation statt.

## 6.4 Virtue

### 6.4.1 Übersicht

VIRTUE realisiert das Referenzmodell auf Basis der Referenzarchitektur durch eine konkrete Implementierung (siehe Abbildung 55). VIRTUE ist eine agentenbasierte Groupware-Umgebung und bietet Komponenten zur Unterstützung von verteilten Arbeitsprozessen aus dem Bürobereich. Es wurden die in dieser Arbeit vorgestellten Konzepte von nutzungsbezogener Transparenz umgesetzt. Außerdem wurde speziell auf die Verbesserung der software-ergonomischen Gestaltung der kooperativen Umgebung und der Benutzungsoberfläche Wert gelegt.

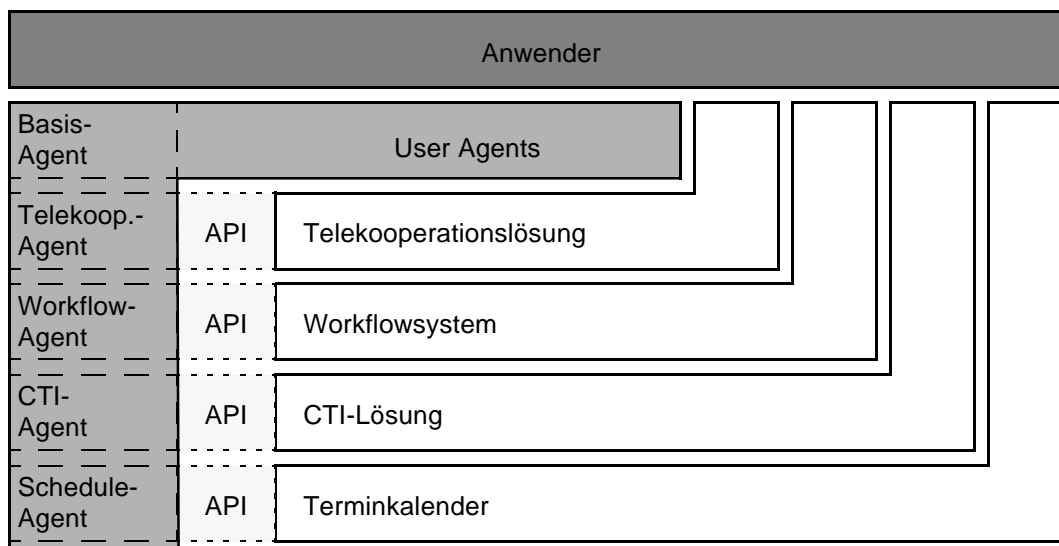


Abbildung 55: Umsetzung des Referenzmodells in VIRTUE

VIRTUE ist in der Lage einerseits die Nutzung der verschiedenen kooperativen Technologien zu ermöglichen und andererseits diese Nutzung möglichst komfortabel und benutzeradäquat zu gestalten. Grundlage ist der Einsatz der im Referenzmodell vorgestellten Software Agenten als integrativer Bestandteil. Abbildung 56 zeigt die Virtue Umgebung wie sie sich dem Anwender darstellt.

Die folgenden Abschnitte beschreiben die realisierten Agenten. Bei den integrierten CSCW- und Workflow-Systemen handelt es sich grundsätzlich um kommerzielle Systeme.

### 6.4.2 Basis-Agent

Der Basis-Agent ist die Kernverwaltungskomponente des Systems und ist für das Management von VIRTUE zuständig und realisiert den Administrator-Agenten des Referenzmodells. Der Agent verwaltet die Nutzer über die Datenbank des Systems und liefert auf Anfrage entsprechende Informationen. Die Datenbank von VIRTUE verfügt über statische Felder, z.B. Name, Telefonnummer usw. sowie über dynamische Felder, wie etwa aktueller

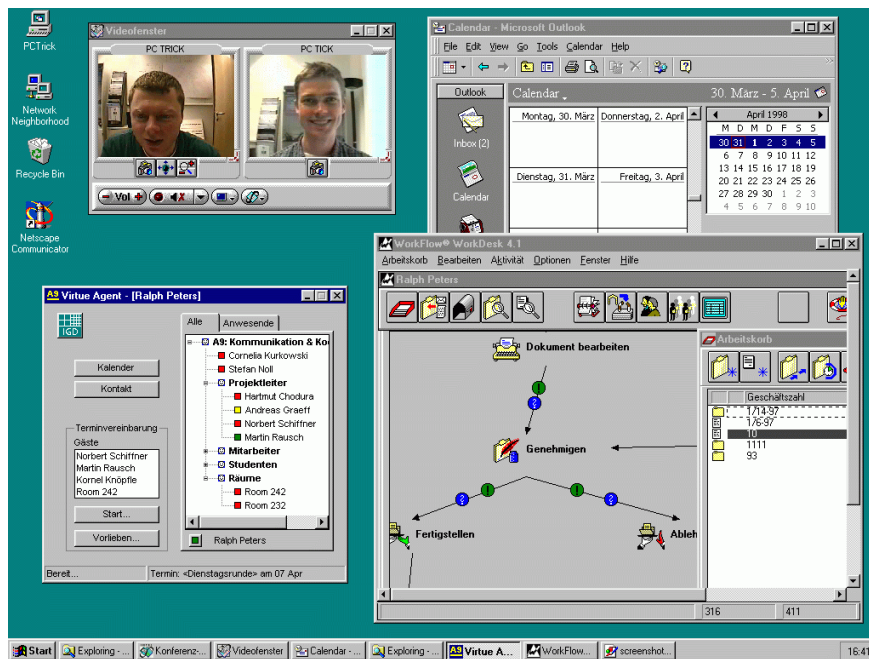


Abbildung 56: Virtue-Umgebung

PC des Anwenders. Weiterhin ist der Agent als Facilitator des Agentenverbundes die Kommunikationszentrale.

Um die Möglichkeit zu besitzen auch von Partnern, die nicht *online* sind, Informationen zu erhalten, erlaubt der Basis-Agent die Migration von Agenten. Verläßt ein Benutzer eine Sitzung migriert der Termin-Agent zum Basis-Agenten und erfüllt dort seine Aufgaben im Namen des Benutzers. Eine Terminvereinbarung ist somit auch mit nicht anwesenden Personen jederzeit möglich. Bei einer erneuten Anmeldung des Benutzers am System, migriert der Agent zurück.

### 6.4.3 Telekooperations-Agent

Der Telekooperations-Agent (TKA) übernimmt die Integration der Telekooperationslösungen. Er besteht aus einem applikationsabhängigen und einem applikationsunabhängigen Teil. Die interne Kommunikation zwischen den Agenten erfolgt auf der Basis eines applikationsunabhängigen Protokolls, womit eine spätere Erweiterung um andere Telekooperationslösungen ermöglicht wird. Die applikationsabhängigen Teile greifen auf die Schnittstellen (APIs) der Telekooperationslösung zu.

Der TKA ermöglicht Auf- und Abbau einer Telekonferenz inklusive Auswahl der Partner. Dabei greift der Agent auf den Basis-, den Termin- und den Workflow-Agenten zurück, um die notwendigen Informationen für einen flexiblen und an die Situation angepaßten Kommunikationsaufbau zu erhalten. Zusätzlich steuert er die Annotationskomponente Crystal-Pad an und versorgt sie mit den notwendigen Informationen. Konkret werden die folgenden Kommunikationsmedien eingesetzt:

- Intel ProShare Conferencing Video System 200 (T.120 konform)
- Microsoft NetMeeting
- CrystalPad (Annotationskomponente)
- Chit Chat (Freely available textbasiertes Chat-Tool)
- Netscape E-Mail oder Microsoft Explorer E-Mail

Der TKA macht weiterhin den Kommunikationsstatus bzw. die Erreichbarkeit potentieller Partner transparent.

#### 6.4.4 Workflow-Agent

Der Anschluß der Workflow-Systeme wird durch den Workflow-Agenten (WFA) realisiert. Er besteht wie der TKA aus einem applikationsabhängigen und einem applikationsunabhängigen Teil. Das kommerzielle Workflow-System

- CSE/WorkFlow®

ist integriert. CSE ist aktives Mitglied der Workflow Management Coalition und paßt seine Schnittstellen den dort definierten Standards an.

Der WFA kooperiert mit den anderen Agenten und stellt Informationen unterschiedlichster Art zu einem Vorgang zur Verfügung. Die Verknüpfung der Meta-Informationen des WFA zusammen mit dem TKA erlauben beispielsweise die optimale Wahl eines Kooperationspartners. Neben der automatischen Bestimmung einer zuständigen Person (z.B. Verfasser einer Akte) wird zusätzlich ein geeignetes Kommunikationsmittel, je nach Präferenzen und Zustand (etwa E-Mail oder Videokonferenz) automatisch durch die Agenten bestimmt. Weiterhin kann das Workflow-System vom WFA angesteuert werden, beispielsweise um einen Vorgang zu initiieren.

#### 6.4.5 CTI-Agent

Der CTI-Agent (*Comuter Telephone Integration*) ermöglicht die Anbindung eines Intelligen Directory an ein Workflow-System. Ziel ist die optimale Unterstützung einer Bürgerschnittstelle, um Anfragen zu Vorgängen schnellstmöglich, unkompliziert und flexibel von einer zentralen Stelle aus beantworten zu können. Diese Schnittstelle kann auch für interne Zwecke innerhalb einer Verwaltung zur Abfrage von Informationen genutzt werden. Zusätzlich besteht die Möglichkeit, Vorgänge zu initiieren oder Notizen an bestehende Vorgänge zu koppeln. Der CTI-Agent bietet hierzu die notwendigen Informationsabfragemechanismen und Kontrollmöglichkeiten über das unterliegende Workflow-System an und verwendet dazu den Workflow-Agenten. Mit dem Intelligenten Directory der Firma

- Bürger Kommunikationsberatung

wurde zu Demonstrationszwecken eine prototyphafte Integration implementiert.

### 6.4.6 Termin-Agent

Der Termin-Agent (TA) integriert ein Terminkalendersystem. In Kombination mit dem Terminkalender wird eine Aushandlung von Terminen basierend auf einem Fuzzy-Modul durchgeführt. Er besteht dazu wie der TKA aus einem applikationsabhängigen und einem applikationsunabhängigen Teil. Die Daten des Terminkalenders werden außerdem zur Bestimmung des Kommunikationsstatus verwendet. Der Termin-Agent integriert derzeit den

- Lotus Organizer® und
- Microsoft Outlook.

Zur Aushandlung von Terminen ist der TA mit einer Benutzungsschnittstelle ausgestattet, die auch die Einstellung von persönlichen Präferenzen erlaubt.

Neue Termine werden dem Benutzer vom Termin-Agent mitgeteilt. Ist der Anwender nicht mit dem Termin einverstanden, hat er die Möglichkeit, den Termin abzulehnen. Diese Information wird, wie auch eine Bestätigung, dem Initiator des Terminwunsches übermittelt. Die Entscheidung, ob ein Termin stattfinden soll, obliegt dann dem Initiator.



# Kapitel 7

## Systemvergleich

### 7.1 Einleitung

In diesem Kapitel soll das realisierte System und die dazu verwendeten Ansätze mit anderen existierenden Systemen verglichen werden. Um aus der Vielzahl der Systeme im CSCW Bereich einige adäquate für einen Vergleich auszusuchen, ist ein Mindestumfang bezüglich der angebotenen Funktionalitäten festgelegt worden. Das für diese Arbeit wichtige Umfeld einer computerunterstützten Verwaltungsumgebung dient dabei als Grundlage zur Auswahl. Die Systeme sollen zumindest automatisierte Vorgangsbearbeitung und synchrone Kooperation in einer integrierten Umgebung unterstützen. Dies ist für den großen Bereich der Büroautomatisationssoftware ein wesentliches Filterkriterium. Wie bereits in Kapitel 2 "Technologischer Überblick" ausgeführt, existieren viele Systeme, die bestimmte einzelne Probleme der Büroautomatisierung lösen. Hingegen existieren nur wenige, die einen ganzheitlichen Ansatz verfolgen.

Vor allen Dingen die Systeme, die in den POLIKOM Projekten entwickelt wurden, erfüllen die geforderten Mindestansprüche. Von daher wird im folgenden genauer auf die Systeme der POLIKOM Projekte POLITEAM und POLIWORK näher eingegangen. Desweiteren werden die Systeme WooRKS-UTUCS und WoTel betrachtet und zu einem Vergleich herangezogen.

### 7.2 Analyse und Vergleich

#### 7.2.1 POLITEAM

POLITEAM [31][123] ist ein vom BMBF gefördertes Verbundprojekt im Rahmen der POLIKOM Projekte. Ziel ist eine Verbesserung der Kooperation innerhalb von geographisch weit verteilten Organisationen. Hintergrund ist, wie bei allen POLIKOM Projekten, die Dezentralisierung der Verfassungsorgane der Bundesrepublik Deutschland auf Berlin und Bonn. POLITEAM basiert auf zwei Metaphern [27]:

- „Die elektronische Laufmappe bildet das System der Hauspost auf dem Computer nach und vereinfacht koordinierte Abläufe. Elektronische Laufmappen können beliebige Dokumente enthalten und unterstützen durch ihren schnellen Transport die Kooperation über große Distanzen.“

- „Der gemeinsame Arbeitsbereich bietet allen Beteiligten einen einfachen Zugriff auf gemeinsam zu bearbeitende Dokumente. Kooperation wird vereinfacht, da diese Dokumente nicht kompliziert ausgetauscht werden müssen, sondern direkt auf dem Arbeitsbereich verfügbar sind und dort gleich bearbeitet werden können.“

Die Entwicklungen innerhalb des POLITEAM Projektes bauen auf dem kommerziellen Workflow-System LinkWorks von Digital auf. Dieses wurde durch verschiedene Werkzeuge erweitert und es wurden anwendungsspezifische Anpassungen vorgenommen. Neben den oben angesprochenen Elementen „elektronische Laufmappe“ und „gemeinsamer Arbeitsbereich“ waren dies vor allen Dingen:

- die Integration einer SmartCard Lösung zur Unterstützung der digitalen Signatur,
- die Einrichtung von signierten Umläufen,
- die Integration eines Bar-Code-Lesers zur Registrierung und Verbindung von Papierdokumenten und ihrem elektronischen Gegenstück,
- die Integration eines Telekonferenzsystems,
- die Möglichkeit zur Annotation von Dokumenten,
- die Integration eines Ereignisdienstes und
- die Modellierung von Organisationseinheiten und Verwaltungsvorgängen.

Als Telekonferenzsystem wird das kommerzielle System ProShare von Intel verwendet. Im Umfang von ProShare ist ein Notizbuch vorhanden, welches auch als Whiteboard verwendet werden kann. Dieses wird zur Annotation von Dokumenten eingesetzt.

Der Ereignisdienst [49][137] erlaubt eine automatische Benachrichtigung von Benutzern beim Eintreten von bestimmten Ereignissen. Ziel ist die Schaffung von mehr Transparenz über die Aktivitäten anderer Nutzer innerhalb der Verwaltungsvorgänge. Ein Benutzer muß sein Interesse an einem Ereignis durch eine Interessensbeschreibung explizit darlegen. Zur Wahrung der Vertraulichkeit von Ereignissen kann ein Benutzer Ereignisse, die er selbst auslöst, mit einer Vertraulichkeitsintensität belegen. Hierdurch wird festgelegt, wie unzugänglich Ereignisse gegenüber anderen sind. Eine Ereignishistorie speichert alle Ereignisse permanent und erlaubt so, bereits vergangene Aktivitäten nachzuvollziehen. Damit wird ein schnellerer Wiedereinstieg in das Geschehen ermöglicht.

Das Modell der Transparenzdatensätze wird in VIRTUE zur Erreichung von Makro-Level-Transparenz auf Benutzerebene eingesetzt. Im Gegensatz dazu konzentriert sich das POLITEAM Projekt mit seinem Ereignisdienst [49] mehr auf die Objektebene. Es wurde eine wichtige Arbeit zur Schaffung von Makro-Level-Transparenz über aktuelle und vergangene Ereignisse innerhalb von Verwaltungsvorgängen erstellt (vergleiche dazu Abschnitt 2.6.3). Im Bereich der Mikro-Level-Transparenz wird auf kommerzielle Werkzeuge (Intel ProShare) zurückgegriffen. Eine erweiterte Unterstützung durch quasi gleichzeitige aktive und passive Interaktion mit dem Fokus des Interesses, wie in dieser Arbeit vorgestellt, existiert nicht. Statt dessen sind Schwerpunkte auf andere Anforderungen (Integration von digitaler Signatur und signierte Umläufe, Integration eines Bar-Code-Lesers) gelegt worden.



Das System POLITEAM wurde in einem zyklisch partizipativen Entwicklungsprozeß (siehe dazu etwa [82] S.25 ff.) aufbauend auf das Workflow-System LinkWorks entwickelt. Das Workflow-System wurde dabei entsprechend der Anforderungen der Anwender angepaßt und erweitert. Die sich daraus ergebende enge Verwobenheit mit einem bestimmten Workflow-System ist ein weiterer wesentlicher Unterschied zwischen dem POLITEAM und dem VIRTUE System. VIRTUE zeichnet sich durch Offenheit und Flexibilität gegenüber den integrierten externen Komponenten aus. Das in dieser Arbeit vorgestellte Referenzmodell legt dazu den Grundstein. Der Einsatz von Software Agenten bietet weiterhin, etwa durch den Einsatz von linguistischen Variablen, bessere Anpassungsmöglichkeiten an den Benutzer.

### 7.2.2 WoTel

Zur flexiblen Einbindung von Telekonferenzen in Workflow-Management wurde WoTel (Workflow und TELEkooperation) [154][155] am Siemens Telekooperationszentrum (STZ) Saarbrücken und an der Universität Ulm im Rahmen des gleichlautenden DTAG/BERKOM Projektes entwickelt. Ziel war die nahtlose Integration der beiden Technologien, um den im Workflow-System modellierten Ablauf durch synchrone, kooperative Aktivitäten ergänzen zu können. Dabei wurde auf eine echte Interoperabilität Wert gelegt.

Es sollte erwähnt werden, daß die Ideen und Konzepte von WoTel in das parallel laufenden Projekt POLIVEST eingebracht wurden (siehe [34]).

Die Durchführung einer kooperativen synchronen Sitzung erfolgt in WoTel in sechs Phasen [154]:

- Bestimmung der Schnittstellen-Parameter
- Sitzungseröffnung, Aufruf des Konferenzsystems
- Sitzungsablaufsteuerung
- Sitzungskontrolle
- Schließen der Sitzung
- Fortschreiten im Workflow

Es existieren weiterhin in WoTel zwei Sitzungsmodelle. Im ersten Modell wird eine Sitzung in einen Vorgang als Aktivität hinein modelliert. Im zweiten Modell können *ad hoc* Sitzungen spontan von einer (beliebigen) Aktivität aus gestartet werden. Während einer Sitzung dient ein Monitoring-Mechanismus der Überwachung der Sitzung und zur Rückführung in das Workflow-System. Eine Checkliste, die innerhalb des Monitoring-Prozesses manuell von den Sitzungsteilnehmern ausgefüllt wird, kann am Ende einer Sitzung zur Weiterleitung des Vorgangs dienen, sofern es sich um eine eigene Sitzungsaktivität handelte. Es wurden die drei möglichen Ergebnisse erfolgreich, teilweise erfolgreich und nicht erfolgreich definiert.

Die notwendigen Informationen werden dabei über einen separaten *conference broker* zwischen dem Workflow-System und dem Telekooperations-System ausgetauscht. Dieser ist

aufgespalten in *workflow* und *conference* Module, die über einen *trader* Informationen austauschen.

WoTel ist nach dem Client/Server-Prinzip aufgebaut und wurde prototypisch mit Hilfe kommerzieller Komponenten (z.B. WorkParty von SNI und GroupX von SIEMENS) realisiert.

Der Schwerpunkt von WoTel liegt in der Modellierung von synchronen Gruppen-Aktivitäten innerhalb eines Vorgangs. Das WoTel-Modell sieht eine Unterscheidung von dynamischen und statischen Konferenzen sowie von geplanten und ad-hoc Konferenzen vor.

Dynamische Konferenzen sind in ihrem Ablauf nicht vorhersehbar und haben einen freien Charakter. Statische Konferenzen haben im Gegensatz dazu einen vorbestimmten Ablauf. Sie werden durch einen Sub-Vorgang modelliert. Die zeitliche Abarbeitung der (synchronen) Konferenz wird dadurch festgelegt. Entsprechend der vorliegenden Informationen können geplante Konferenzen an sich besser unterstützt werden, da die Parameter der Konferenz im wesentlichen schon zum Modellierungszeitpunkt des Vorgangs festgelegt werden.

Geplante Konferenzen sind inklusive des Teilnehmerkreises (u.U. durch Rollen bestimmt) bereits zum Zeitpunkt der Modellierung eines Vorgangs bekannt und werden in den Vorgang hinein modelliert. Der genaue Zeitpunkt der Konferenz wird zum Ausführungszeitpunkt festgelegt. Durch Auflösung von Rollen oder Stellen in existente Personen, sofern sie zum Ausführungszeitpunkt bekannt sind, wird die Teilnehmerliste bestimmt. *Ad hoc* Konferenzen werden nicht in Vorgängen vormodelliert. Die aktuelle Aktivität ist keine Sitzungsaktivität und wird damit auch nicht nach Beendigung einer Sitzung verlassen.

Synchrone Telekooperation wird somit als ein Teil eines Vorgangs angesehen. Vorteil dieses Ansatzes ist die Möglichkeit der späteren Nachvollziehbarkeit von synchronen Abläufen innerhalb eines Vorgangs, da sie als eigene Aktivität innerhalb des Workflow-Systems gespeichert werden. Der in WoTel erarbeitete Ansatz ist prinzipiell unterschiedlich zu dem in dieser Arbeit vorgestellten Modell. Parallelen finden sich bei der Verwendung von Metainformation zur Bestimmung des Kooperationspartners. Dieser wird entweder bereits zum Modellierungszeitpunkt bestimmt (u.U. durch eine Rolle, die dann zum Ausführungszeitpunkt aufgelöst wird) oder ad hoc anhand der zur Verfügung stehenden Informationen. Problematisch an der Systemlösung WoTel ist, daß das System an sich in den Mittelpunkt gestellt wird und nicht der Anwender. VIRTUE erleichtert hier durch den Einsatz von autonomen software Agenten die Bedienung der komplexen heterogenen Systemstruktur. Die Bereiche Makro- und Mikro-Level-Transparenz waren bei der Entwicklung des Systems WoTel kein Gestaltungsaspekt.

An dieser Stelle muß angemerkt werden, daß die Komponente CrystalPad, allerdings ohne sematische Verknüpfung, in GroupX integriert ist. Der Autor arbeitete an dieser Stelle mit den Entwicklern von WoTel zusammen.

### 7.2.3 WooRKS-UTUCS

WooRKS-UTUCS [32][33] vereinigt das Workflow-Management System WooRKS und das Groupware System UTUCS. WooRKS wurde im ESPRIT Projekt ITHACA entwickelt. Das Ziel des Projektes ITHACA ist es, die Langzeitkosten für die Entwicklung neuer Anwendungen sowie für die Wartung von Standardapplikationen in ausgewählten Bereichen durch den Einsatz von objekt-orientierten Methoden zu reduzieren.

WooRKS ist hauptsächlich ein Grundgerüst für ein Workflow-System. Die Implementation von WooRKS baut auf dem objekt-orientierten Entwicklungs-System HooDS (*Highly object oriented Development System*) auf. HooDS stellt fast 400 wiederverwendbare Objektklassen in sechs Modulen zur Verfügung. Die sechs Module sind:

- Organisationsmodul: beschreibt wer an Workflow Prozeduren beteiligt ist sowie die entsprechenden Aufgaben;
- Informationsmodul: beschreibt die semantischen Eigenschaften und Inhalte der innerhalb von Workflow Prozeduren ausgetauschten Daten;
- Zeitmodul: stellt die grundlegenden zeitlichen Randbedingungen (z.B. Abgabetermine, etc.) für Prozeduren und Aktivitäten zur Verfügung;
- Tätigkeitsmodul: beschreibt die atomaren Handlungen jedes Beteiligten;
- Koordinationsmodul: beschreibt, wie eine Gruppe von Personen zusammenarbeitet, um eine Prozedur auszuführen;
- Benutzeroberflächenmodul: enthält die notwendigen Mechanismen zur Erstellung der Benutzeroberfläche.

Das System UTUCS unterstützt eine Gruppe von untereinander vernetzten Personen (ein Büro, ein Team) bei ihrer Kommunikation. Dabei kommen verschiedene synchrone (z.B. Treffen von Angesicht zu Angesicht) bzw. asynchrone (z.B. E-Mail) Kommunikationsmedien zum Einsatz. Die Grundeinheit der Kommunikation ist dabei das Gespräch zwischen zwei oder mehreren Personen. Dabei wird der Initiator des Gesprächs sowie dessen unmittelbarer Gesprächspartner einbezogen. Optional können weitere Personen als Beobachter des Gespräches vom Initiator eine Teilnahmeberechtigung bekommen. Die Gespräche in UTUCS werden je nach Verpflichtung in folgende vier Kategorien eingeteilt:

- „Verpflichtung zum Handeln“: alle Gespräche, bei denen die Beteiligten die Durchführung einer Aktivität besprechen und Verantwortliche dafür bestimmen;
- „Verpflichtung zum Sein“ (im Sinne der Einnahme einer Rolle): Gespräche, in denen eine Abänderung der Organisationsstrukturen abgewägt wird;
- „Informationsverarbeitung“: diese Konversationsart kann dazu dienen, jemandem die Aufgabe der Datenverwaltung zu übertragen;
- „Informationsbereitstellung“: dient der Anforderung oder Bereitstellung von Informationen.

UTUCS wird innerhalb von WooRKS zur Behandlung von Ausnahmesituationen eines Vorgangs eingesetzt. Typische Beispiele solcher Situationen sind die folgenden:

- die Ausführung einer Aktivität setzt eine Information voraus, die nicht vorhanden ist;
- eine Aktivität wurde an jemanden mit der falschen Zuständigkeit weitergeleitet;
- die empfangenen Dokumente beinhalten fehlerhafte Daten und müssen revidiert werden;
- der für eine laufende Aktivität Zuständige ist abwesend.

Wenn bei der Ausführung einer Aktivität eine solche Fehlersituation auftritt, muß sich der Betroffene entweder mit einer an dieser Prozedur beteiligten Person oder mit der für diese Prozedur zuständigen Person in Verbindung setzen. In diesen Fällen unterstützt UTUCS die Kommunikation. Die Aktivität, die die Fehlersituation erzeugt hat, ist zusammen mit dem dazugehörigen Prozedurkontext das Thema der anfallenden Gespräche. Das Ergebnis der Gespräche sollte die Behebung des Fehlers sein und die Weiterführung der Prozedur ermöglichen.

Jeder Akteur kann aus seinem Workflow-Korb ein Gespräch über eine Aktivität starten. Daraufhin wird ein Formular bereitgestellt, in dem bereits die Felder Datum, Empfänger (Verantwortlicher für die Aktivität bzw. den Vorgang), Thema (Aktivität) und Fristen automatisch ausgefüllt sind. Allerdings hat der Absender die Möglichkeit jedes dieser Felder zu ändern. Er trifft damit selbst die Entscheidung ein Gespräch mit jemand anderem aufzunehmen, von dem er meint, er könnte ihm bei der Problemlösung behilflich sein.

Im Gegensatz zu VIRTUE basiert WooRKS-UTUCS auf einer kompletten Eigenentwicklung. Das Workflow-System hat nur einen eingeschränkten Funktionsgrad. Es wird E-Mail für asynchrone „Gespräche“ eingesetzt. Synchrone Kooperation wird dadurch unterstützt, daß die Ergebnisse von Angesicht-zu-Angesicht Gesprächen in ein Formular eingetragen werden können. Vorhandene Standards in den Bereichen Telekommunikation und Workflow werden außer acht gelassen. Der Bereich Terminvereinbarung ist ebenfalls nicht Bestandteil der Arbeiten von WooRKS-UTUCS. Die prinzipiellen Überlegungen im Bereich Macro-Level-Transparenz - auch wenn die Autoren von WooRKS-UTUCS sich nicht direkt dieser Problematik widmen - sind jedoch mit den in dieser Arbeit entwickelten vergleichbar.

Informationen aus der Aktivitäten-Verwaltung WooRKS werden zur Initialisierung der Kommunikationskomponente UTUCS verwendet. Die in Abschnitt 4.3.4 gemachten Überlegungen zu Workflow-Agenten und ausgezeichneten Personen sind hier ebenfalls umgesetzt worden. Weitere interessante Ansatzpunkte finden sich in der Formalisierung des „Gesprächs“ in einen finiten Automaten, der die oben erwähnten Konversationsarten durch Zustände *offered*, *requested*, *taken*, *rejected*, *deleted*, *concluded*, *satisfied* und *not satisfied* modelliert. Hier ergeben sich Ähnlichkeiten zum System WoTel, in dem die synchrone Kooperation ebenfalls über einen formalen Ansatz modelliert wird.

#### **7.2.4 POLIWORK**

Ebenso wie POLITEAM ist POLIWORK [16][22] ein vom BMBF gefördertes Verbundprojekt im Rahmen der POLIKOM Projekte. Technisches Ziel ist eine innovative Anwenderlösung für die orts- und zeitunabhängige Zusammenarbeit in räumlich verteilten Organisationen. Dabei soll vor allen Dingen Telekooperation und Dokumentenverwaltung

am persönlichen Arbeitsplatz unterstützt werden. Die innerhalb des Projektes entwickelte Anwenderlösung beinhaltet [22]:

- eine Schriftgutverwaltung zur Erfassung, Pflege und Verwaltung von Vorgängen und Dokumenten. Weiterhin erlaubt sie Attribut- und Volltextrecherche,
- eine Gruppenablage, die als globaler Dokumentenspeicher für ein Mitarbeiterteam dient. Basis ist ein Objektmodell, welches den Austausch zwischen den Ablagen und der Schriftgutverwaltung erlaubt. Die Gruppenablage ist mit einem Informationsdienst ausgestattet, der etwa bei der Versionsänderung eines Dokumentes automatisch den Anwender informiert,
- eine Cross-Plattform Application Sharing Komponente. Die auf Bitmap-Austausch basierende Komponente ist eine Eigenentwicklung [127] und zeichnet sich vor allem durch die Einsatzmöglichkeit auf unterschiedlichen Betriebs- und Window-Systemen aus,
- ein kooperatives Dokumenten-Annotationswerkzeug auf der Basis von PostScript-Dokumenten. Anders als bei herkömmlichen Whiteboards (siehe Abschnitt 2.2.3), ist der Hintergrund des Whiteboard hier ein PostScript-Dokument. Dies erlaubt die seitenweise Annotation der Druckausgabe eines Dokumentes und
- ein Modul zur Unterstützung von PC-Telefonkonferenzen. Das Modul ermöglicht die Sprachkommunikation während einer Telekonferenz unter Verwendung eines Telefondienstes. Vorteil ist, daß der Anwenderarbeitsplatz nicht mit Multimedia-Hardware ausgestattet sein muß.

Im Gegensatz zu anderen POLIKOM Projekten und zu VIRTUE basiert POLIWORK in den Bereichen Workflow und Telekooperation auf Eigenentwicklungen, wie der Schriftgutverwaltung und der Sharing Komponente, und nicht auf der Integration oder Erweiterung von kommerziellen Komponenten. Andere Aspekte wie Transparenz oder Verwendung von Standards werden weniger berücksichtigt. Von einem Vergleich der Gruppenablage und der Schriftgutverwaltung zu einem Workflow-System soll an dieser Stelle abgesehen werden, da letzteres nicht Bestandteil der hier vorliegenden Arbeit ist.

Im Bereich der Mikro-Level-Transparenz muß das kooperative Dokumenten-Annotationswerkzeug auf der Basis von PostScript-Dokumenten betrachtet werden. Gegenüber herkömmlichen Whiteboards bietet diese Komponente des POLIWORK-Systems den Vorteil, daß sie auf einem seitenbasierten Ansatz beruht. Es ist damit kein erneutes Laden einer weiteren Hintergrundseite notwendig. Dennoch ist das Dokument statisch und kann nicht verändert werden. Der hier vorgestellte und mit CrystalPad umgesetzte Ansatz des transparenten Whiteboards ist damit flexibler, da er sowohl passive als auch aktive Interaktion erlaubt.

In diesem Zusammenhang sei noch das System DIVA [136] erwähnt, welches Einfluß auf die Arbeiten im POLIWORK Projekt genommen hat. DIVA besitzt eine interessante Erweiterung zur Schaffung von Makro-Level-Transparenz innerhalb einer synchronen kooperativen Sitzung: statt wie in den meisten Systemen, wie auch CrystalPad, jedem Teilnehmer eine bestimmte Annotationsfarbe zur Identifizierung zuzuordnen, wird zusätzlich eine optische Verbindung zwischen Annotationsobjekt und Teilnehmer geschaffen. Eine gestrichelte

Linie zwischen dem Videobild des annotierenden Teilnehmers und dem gerade gezeichnetem Annotationsobjekt zeigt den Kooperationspartnern an, wer gerade wo welche Annotation anfügt.

### 7.3 Zusammenfassung und Beurteilung

Alle oben genannten Systeme zielen auf den Büro- und Verwaltungsbereich ab und integrieren automatisierte Vorgangsbearbeitung und synchrone Kooperation. Hierzu zählt insbesondere die Bereitstellung eines - kommerziellen oder in Eigenentwicklung implementierten - Workflow-Systems als Basis der jeweiligen Systemumgebung. Es kann angemerkt werden, daß die Entwicklungen POLITEAM und POLIWORK sich an einem partizipativen Entwicklungsprozeß orientieren und sich damit auf ein bestimmtes Workflow- bzw. Schriftgutverwaltungs-System festlegen. WoTel und WooRKS-UTUCS haben, wie auch VIRTUE, einen allgemeineren Ansatz gewählt. Für WooRKS-UTUCS ist allerdings durch die in diesem Bereich durchgeführte Eigenentwicklung das Funktionsspektrum eingeschränkt.

Für synchrone Kooperation werden von den Systemen POLITEAM und WoTel jeweils kommerzielle Systeme eingesetzt. Eine zusätzliche Unterstützung wird - abgesehen von den Integrationsaspekten - nicht angeboten. WooRKS-UTUCS nimmt auch hier eine Sonderstellung ein, da die synchrone Kooperation nur in Form von „Gesprächen“ durchgeführt werden kann, die aus technischer Sicht durch E-Mail und Diskussionen von Angesicht-zu-Angesicht aufgebaut wird. In POLIWORK wurden durch eine Eigenentwicklung hingegen zusätzliche Mechanismen zur verbesserten Interaktion mit dem Fokus des Interesses geschaffen. Das auf einem PostScript-Dokument als Hintergrund basierende Whiteboard bietet die Möglichkeit seitenweise Dokumente zu annotieren. Trotzdem erlaubt es nur die passive Interaktion mit dem Annotationsobjekt. Die Annotationskomponente von VIRTUE CrystalPad hingegen erlaubt sowohl aktive als auch passive Interaktion. Zusätzlich bietet CrystalPad die Möglichkeit jede beliebige Applikation als Hintergrund zu nutzen und ist damit nicht auf PostScript-Dokumente beschränkt.

Der Bereich Transparenz ist nur von den Entwicklern des Systems POLITEAM direkt aufgegriffen worden. Dies zeigt sich vor allem durch den in POLITEAM entworfenen und implementierten Ereignisdienst. Dieser Dienst bietet eine sehr gute Unterstützung für den Bereich Makro-Level-Transparenz auf Objekt-Ebene. Das System VIRTUE unterstützt diesen Bereich nur rudimentär. Dagegen unterstützt VIRTUE die Bereiche Makro-Level-Transparenz auf Benutzerebene und Mikro-Level-Transparenz stärker. Das alle drei Bereiche wichtig und beachtenswert sind, haben die unterschiedlichen Untersuchungen gezeigt (vergleiche Abschnitt 2.6 und 3.2). Eine Ausweitung des System VIRTUE in Richtung Objekt-Ebene ist sinnvoll.

Ein weiterer Punkt ist die mangelnde Hilfe im Bereich der Terminvereinbarung. Terminvereinbarungen können in den vorgestellten Systemen nur durch die bereits in den kommerziellen Workflow-Systemen bereits vorhandenen Möglichkeiten durchgeführt werden. Durch den Einsatz von Software Agenten und die Integration von Fuzzy-Logic in VIRTUE ist eine

wesentlich bessere Benutzerunterstützung möglich. Linguistische Variablen erlauben dem Benutzer in natürlichsprachlicher Weise seine Terminwünsche genauer zu spezifizieren. Die Technologie der Fuzzy-Logic erlaubt die Berechnung von optimalen Terminvorschlägen für alle Beteiligten.

VIRTUE ist das einzige System, daß sich bei der Integration der Agententechnologie bedient. Die daraus entstehenden Vorteile bezüglich Flexibilität, Unterstützung heterogener Systeme und Benutzerführung können mit den eher statischen Ansätzen der erwähnten Systeme nicht erreicht werden. Auch sollte an dieser Stelle das Entwicklungspotential dieses Technologiebereiches nicht unterschätzt werden. Die in dieser Arbeit vorgestellten Ansätze bieten ein weites Spektrum an Entwicklungsmöglichkeiten (siehe Abschnitt 8.3).





## **Kapitel 8**

### **Zusammenfassung und Ausblick**

#### **8.1 Zusammenfassung der Ergebnisse**

Der Einzug von Informations- und Kommunikationstechniken nicht nur in öffentlichen Verwaltungen ist zwingend notwendig, um den Anforderungen im Hinblick auf das kommende Jahrtausend gerecht zu werden. Flexible, bürgernahe und leistungsorientierte Administrationen sind für eine effiziente und qualitativ hochwertige Durchführung heutiger und zukünftiger Aufgabenbereiche notwendig. Um auch der zunehmenden Verteilung der Organe adäquate Unterstützung zu bieten, werden zunehmend CSCW-Werkzeuge eingesetzt. Die technischen Voraussetzungen sind hierfür bereits gegeben. Moderne Workflow-Systeme erlauben die Automatisierung von Vorgängen und verwalten die Dokumentenflut der täglichen Arbeit. Telekooperationslösungen bieten die Möglichkeit unabhängig von Ort und Zeit miteinander auch über große Entfernungen hinweg zu kommunizieren. Durch den Einsatz von Application Sharing und Whiteboard Komponenten wird zusätzlich gemeinsames Arbeiten ermöglicht.

Der Erfolg der Investitionen in Richtung einer computerunterstützten kooperativen Büro-umgebung ist aber nicht zuletzt abhängig von der software-ergonomischen Gestaltung der Umgebung. Dies bedeutet den flexiblen Einsatz von integrierten Komponenten in einer durchgängigen Gesamtumgebung. Nutzungsbezogene Transparenz ist hier ein wesentlicher Bestandteil und muß für eine erfolgreiche Einführung in Anwendungsgebieten mit potentiell als Computerlaien einzustufenden Nutzern etabliert werden. Dabei darf das Recht auf informative und kommunikative Selbstbestimmung nicht außer acht gelassen werden. Gruppenweite transparente Aktionen müssen ohne die Möglichkeit der Überwachung im System angeboten werden. Zusätzlich muß dem Anwender Unterstützung bei der Durchführung kooperativer verteilter Telekonferenzen gegeben werden. Die Interaktion mit dem ihn interessierenden Diskussionspunkt muß einfach und flexibel gestaltet sein.

Aus diesen Anforderungen ergaben sich die folgenden Problemstellungen, die in dieser Arbeit gelöst wurden:

- Integration heterogener CSCW-Komponenten zu einer kooperativen verteilten Verwaltungsumgebung.
- Schaffung von nutzungsbezogener Transparenz im Mikro-Level- und Makro-Level-Bereich innerhalb einer integrierten Groupware-Umgebung.

- Möglichkeit des Systems, autonome Entscheidungen treffen zu können und Umsetzung natürlichsprachlicher Benutzeranforderungen, um eine software-ergonomische Benutzerführung innerhalb der integrierten Systemumgebung zu schaffen.
- Definition und Umsetzung von passiver und aktiver Interaktion mit dem Fokus des Interesses innerhalb einer synchronen kooperativen Sitzung.
- Integration von graphischen Annotationen in applikationsunabhängiger Form zur optimalen Unterstützung von aktiver und passiver Interaktion mit dem Fokus des Interesses.

Um eine integrierte CSCW-Umgebung zur Unterstützung der vielfältigen Anforderung heutiger und zukünftiger Verwaltungsapparate zu gewinnen, hat sich in dieser Arbeit gezeigt, daß der Einsatz von Software Agenten als integrierter und transparenzschaffender Bestandteil, eine entscheidende Verbesserung bedeutet. Software Agenten erlauben durch Autonomie, Reaktions- und Entscheidungsfähigkeit, Glaubwürdigkeit und Sozialfähigkeit die optimale Unterstützung von Anwendern. Gegenüber herkömmlichen Ansätzen können so flexible und dem Nutzer angepaßte Reaktionen vom System aus autonom herbeigeführt werden, die eine bestmögliche Anpassung an die Anforderungen von Anwendern im allgemeinen und einer Verwaltung im besonderen erlaubt.

Durch den Einsatz von Agenten wird, neben der Einbindung von Workflow- und Telekooperationstechniken, die Ausnutzung von Meta-Information ermöglicht. Dies erlaubt die Unterstützung von gruppenweiter Makro-Level-Transparenz und eine medienbruchfreie Integration. Zusätzlich konnte durch die Etablierung von Mikro-Level-Transparenz eine entscheidende Verbesserung der synchronen passiven wie aktiven Interaktion mit dem Diskussionsschwerpunkt im Telekonferenzfall erreicht werden.

Um diese Ergebnisse zu erreichen ist zunächst ein Blick auf den Stand der Technik geworfen worden. Hier wurden unterschiedliche Technologiebereiche, wie CSCW, Telekooperation und Konferenzumgebungen, computergestützte Vorgangsbearbeitung und Workflow untersucht. Ein weiterer Abschnitt beschäftigte sich mit Software Agenten und den zugrundeliegenden Metaphern und Architekturen. Weiterhin ist der aus dem Bereich der Software-Ergonomie stammende Begriff der nutzungsbezogenen Transparenz und dessen Ausprägungen Mikro-Level- und Makro-Level-Transparenz näher beleuchtet worden.

Mit einer Analyse der Benutzeranforderungen aus dem Verwaltungsbereich wurde die Relevanz von Transparenz nochmals bestätigt und die Ansprüche aus der Verwaltung an Transparenz charakterisiert. Diese Untersuchungen mündeten in einem technischen Modell nutzungsbezogener Transparenz. Transparenz kann danach als Vereinigung verschiedener Informationen zu einem bestimmten Sachverhalt angesehen werden. Mit Hilfe der Model-View-Controller Metapher ist das Modell der Transparenzdatensätze entwickelt worden, welches dem Referenzmodell als Basis zum Informationsaustausch und zur Schaffung von Makro-Level-Transparenz dient.

Das Referenzmodell bietet durch ein offenes Schalenmodell die Grundlage für eine integrierte computergestützte kooperative Verwaltungsumgebung. Ein halb-föderativer Agentenverbund ist die Basis des Modells. Verschiedene Agentenklassen kooperieren im Multi-Agentensystem und nutzen die durch die Transparenzdatensätze gegebenen Meta-Inforna-

tionen zur Steuerung und Kontrolle der integrierten Groupware-Lösungen aus den Bereichen Telekooperation, Terminvereinbarung und Workflow. Durch die Visualisierung der Transparenzdatensätze wird für den Anwender Makro-Level-Transparenz im Gruppenraum geschaffen. Die medienbruchfreie Integration durch die Agenten ermöglicht eine softwareergonomische Benutzerführung durch die komplexen Systemteile. Dabei erlauben die ortsunabhängigen Agenten individuelle Einstellungen, die von einem Administrations-Agenten verwaltet werden und damit jederzeit und überall im Agentenverbund zur Verfügung stehen. Für die Klassen Workflow, Telekooperation und Terminkalender ist das Referenzmodell umgesetzt worden. Die Agenten bestehen aus einem applikationsabhängigen und einem applikationsunabhängigen Teil, womit die Offenheit des Modells gewährleistet ist und grundsätzlich eine Erweiterung um andere Groupware-Lösungen ermöglicht wird. Die applikationsabhängigen Teile greifen auf Schnittstellen (APIs) einer konkreten Groupware-Lösungen zu. Es wurde eine Kopplung des Workflow-Agenten an das Referenzmodell der Workflow-Management Coalition vorgestellt und die Verknüpfung des Telekooperations-Agenten an den Standard T.120 gezeigt. Für den Terminkalender-Agenten wurde anhand des genormten *Calendaring and Scheduling Interface* eine Anbindung geschaffen.

Wie bereits angedeutet, ist Autonomie für Software Agenten ein wesentlicher Bestandteil. Um auf Einflüsse der Umgebung und Anforderungen der Anwender oder anderer Agenten reagieren zu können, ist ein gewisser Grad an „Intelligenz“ notwendig. Um diese zu realisieren ist die Mathematik der Fuzzy-Logik eingesetzt worden. Wesentlicher Vorteil ist, daß auch unscharfe Attribute in exakte Lösungen umgesetzt werden können. Für das hier vorgestellte Referenzmodell sind dazu verschiedene linguistische Variablen definiert worden. Diese erlauben es unter anderem, gerade die durch natürlichsprachliche Anforderungen gegebenen Bedürfnisse der Benutzer in effektive Aktionen der Software Agenten umzusetzen. Das entwickelte Fuzzy-Modul wird weiterhin für die Aushandlung von Aktionen innerhalb des Agentenverbundes verwendet.

Um zu einer Realisierungsbasis zu gelangen ist anhand des Referenzmodells eine Referenzarchitektur gebildet worden. Sie diene als Grundlage für spätere Implementierungen. Anhand dieser Architektur ist zusätzlich die synchrone Interaktion mit dem Fokus des Interesses (FDI) in einer kooperativen Telekonferenz aufgezeigt worden. Dazu sind die passive und die aktive Interaktion mit dem FDI definiert worden und es ist ein Konzept zur Interaktion aufgestellt worden. Durch die Einführung von transparenten Annotationen ist hierzu eine Realisierung vorgestellt worden. Mit Hilfe der in dieser Arbeit spezifizierten Semantischen Verknüpfung ist zusätzlich die Möglichkeit für dynamische Annotationen geschaffen worden.

## 8.2 Stand der Entwicklungen

Um die in dieser Arbeit erarbeiteten Konzepte zu prüfen, wurden zwei Prototypen entwickelt. Das System VIRTUE setzt die Referenzarchitektur um und zeigt damit die praktische Umsetzung der Ansätze des Referenzmodells auf. VIRTUE basiert auf Software Agenten, die über Programmierschnittstellen (APIs) kommerziell verfügbare Groupware-Komponenten kontrollieren und steuern. Zusätzlich werden Informationen von den Komponenten

abgefragt und verwendet, um autonom Entscheidungen treffen zu können. Das System ermöglicht weiterhin Markro-Level-Transparenz und wird derzeit in den Referenzbüros der Deutschen Telekom eingesetzt. Konkret sind das T.120 konforme Telekonferenzsystem Intel ProShare® und das Telekonferenzsystem Microsoft® NetMeeting, das kommerzielle Workflow-System CSE/WorkFlow® und die Terminkalender Lotus Organizer® und Microsoft® Outlook integriert worden. Das System VIRTUE befindet sich derzeit im Alpha-Status einer Produktisierung und sollte in einem Piloten von der Deutschen Telekom in der Kreisverwaltung Bitburg eingesetzt werden. Aufgrund finanzieller Probleme ist dieser Pilot allerdings vorerst gestoppt worden.

Das System CrystalPad ist eine verteilte computergestützte Annotationsumgebung, die Gruppendiskussionen auf beliebigen Anwendungen unterstützt. Jeder Teilnehmer erhält seine eigene Folie, auf der er seine Anmerkungen in der ihm zugeordneten Farbe skizziert. CrystalPad ist in das Telekonferenzsystem GroupX von SIEMENS integriert, so daß mehrere örtlich verteilte Konferenzteilnehmer gleichzeitig an derselben Anwendung arbeiten als auch darauf annotieren können. CrystalPad wurde weiterhin prototypisch in das System VIRTUE eingebunden. CrystalPad erlaubt Annotationen mit unterschiedlichen Medien direkt auf der Desktop-Anwendung (passive Interaktion mit FDI). Dabei bleibt die Desktop-Anwendung normal benutzbar (aktive Interaktion mit FDI). Für das Desktop-Publishing System FrameMaker® wurde die Semantische Verknüpfung realisiert. CrystalPad wird zusammen mit GroupX von SIEMENS kommerziell vertrieben.

### 8.3 Ausblick auf weitere Entwicklungen

Auf dem Referenzmodell aufbauend, lassen sich zwei wesentliche Richtungen von Weiterentwicklungen aufzeigen: die gezielte Weiterentwicklung zur Anbindung von neuen Groupware-Komponenten und die Vergrößerung des Anwendungsgebietes durch Integration weiterer Anforderungen und Lösungen.

Die Anbindung weiterer Groupware-Komponenten ist im Referenzmodell bereits vorgesehen. Im Kontext öffentlicher Verwaltungen ist eine sinnvolle Ergänzung die Schaffung einer Bürgerschnittstelle: Anfragen von Bürgern an eine öffentliche Verwaltung können zu einem Großteil am Telefon beantwortet werden. Um solche Bürgeranfragen zu Vorgängen schnellstmöglich, unkompliziert und flexibel von einer zentralen Stelle aus beantworten zu können, sollte in einer computergestützten kooperativen Verwaltungsumgebung eine Anbindung der Telefonanlage an das Workflow-System durchgeführt werden. Ziel muß dabei die optimale Unterstützung einer Bürgerschnittstelle sein. Durch Integration einer CTI (*Computer-Telephon-Integration*) Lösung in das Referenzmodell kann hier ein echter Mehrwert geschaffen werden. Bei einem Anruf wird die anrufende Telefonnummer weitergeleitet und der Anrufer ermittelt. Alle aktuellen Vorgänge werden angezeigt und es können schnell entsprechende Auskünfte erteilt werden. In schwierigen Fällen kann sofort an den verantwortlichen Sachbearbeiter weitervermittelt werden. Diese Schnittstelle kann auch für interne Zwecke innerhalb einer Verwaltung zur Abfrage von Informationen genutzt werden. Zusätzlich sollte die Möglichkeit, Vorgänge zu initiieren oder Notizen an bestehende Vorgänge zu koppeln, bestehen. Der Workflow-Agent bietet hierzu bereits die notwendigen

Informationsabfragemechanismen und Kontrollmöglichkeiten über das unterliegende Workflow-System an. Innerhalb des Systems VIRTUE wurde eine solche Integration bereits prototypisch durchgeführt. Durch die Erweiterung der Möglichkeiten auf das World-Wide-Web kann ein zusätzlicher Fortschritt erzielt werden. Vor allen Dingen die Initiierung von Vorgängen über das WWW bringt entscheidende Vorteile. Für die Verwaltung liegt das ausgefüllte Formular bereits in computerlesbarer Form vor und der Vorgang wird automatisch an die richtige Stelle geleitet. Der Bürger hat die Möglichkeit ohne Behördengang Anträge zu generieren und sich jederzeit über den aktuellen Status seines Antrag zu informieren.

Die Einbindung von Agenten in virtuelle Umgebungen ist ein interessantes neues Forschungsgebiet innerhalb der Informatik und bietet die Möglichkeit zur Realisierung neuer Darstellungsparadigmen in der Bürowelt. 3D-Metaphern werden verwendet, um dem Anwender ein möglichst natürliches Umfeld seines Arbeitsbereiches zu geben. Ein Beispiel einer Zukunftsvision ist die Verwendung von Großprojektionen, die einer Verlängerung des Schreibtischs gleichen und den Eindruck erwecken, als ob der darauf dargestellte Kooperationspartner sich im gleichen Büro aufhält. In einer solchen virtuellen Umgebung muß das System wissen, welche Objekte in der Szene enthalten sind, wo sie sich befinden, wie sie darzustellen sind und wohin sie sich, sofern sie nicht statisch sind, bewegen. Eine Kommunikation zwischen einzelnen Objekten, seien sie durch den Benutzer gesteuert oder durch das Programm kontrolliert, fehlt allerdings meist. Eine Möglichkeit, diese fehlende Funktionalität in virtuellen Realitäten zu ersetzen, ist die Verwendung von Agenten. Die Agenten können, genauso wie menschliche Assistenten, wiederholt auftretende Aufgaben automatisieren und auf intelligente Art komplexe Abläufe zusammenfassen. Für das hier vorgestellte Referenzmodell bedeutet dies eine Adaptierung der Lösungsansätze von der Anbindung von Komponenten hin zur Verknüpfung mit Objekten. Erste Ansätze lassen sich dazu beispielsweise in [78] oder [113] finden.



## Abbildungsverzeichnis

1. Die Johansen time-place Matrix	10
2. Benutzungsoberfläche des Distributed SketchPad	13
3. Prinzipielle Architektur eines Whiteboards	14
4. Konzeptueller Aufbau einer Sharing Komponente	15
5. Das BERKOM Multimedia Mail Szenario	16
6. Basisarchitektur eines CSCW-Systems am Beispiel des BERKOM Multimedia Collaboration Services MMC	18
7. Multipunkt-Verbindungen im T.120	19
8. Das T.120 Systemmodell	20
9. Service Access Points der T.120 Generic Conference Control Komponente	21
10. Das WfMC Referenzmodell	24
11. Relationen der Basisterminologien im Metamodell der WfMC	25
12. Architektur eines Dokumenten-Management-Systems	27
13. Föderatives Agentensystem	30
14. Facilitatoren in Multi-Agenten-Systemen	31
15. Die Architektur von ASAP	33
16. Benutzungsoberfläche von Xhtalk	37
17. Nutzungsbezogene Transparenz	45
18. Erstellung und Visualisierung eines Transparenzdatensatzes	52
19. Model-View-Controlling mit Transparenzdatensätzen	53
20. MVC -TDS Prinzip	53
21. MVC-TDS-Prinzip bei geographisch verteilten Lokalitäten	54
22. Grobkonzept des Referenzmodells	60
23. Verfeinertes Referenzmodell	62
24. Halbföderatives Agentensystem	64

---

25. Intergration des Workflow-Agenten in das Referenzmodell der WfMC	65
26. Beispiel einer verteilten Agenten-Föderation	67
27. Integration des Telekooperations-Agenten in die T.120 Empfehlung der ITU	68
28. Agentenbasierte Terminaushandlung	72
29. Spontane Kommunikation	72
30. Zugehörigkeit der Elemente aus der Menge X zur unscharfen Menge A	76
31. Beispiel eines generellen Fuzzy-Set	77
32. Durchschnitt, Vereinigung und Komplementbildung unscharfer Mengen	79
33. Verhalten verschiedener Modifikatoren	79
34. Allgemeine Struktur eines Regelkreises mit Fuzzy-Control	80
35. Inferenz mit verschiedenen Operatoren	88
36. Freie Termine	90
37. Modifikation	90
38. Mögliche Termine	90
39. Gewichtung	91
40. Resultierendes Fuzzy-Set	91
41. Kombination der Ergebnisse, Wichtung nach Dringlichkeit und Berechnung des resultierenden Termins	91
42. Technik eines transparenten Whiteboards	100
43. Probleme ohne Semantischen Verknüpfung	101
44. Einfügen im Paragraph	102
45. Semantic Tagging	104
46. Dynamische Annotationen	105
47. Referenzarchitektur einer agentenbasierten, verteilten Verwaltungsumgebung	108
48. Verbindung von Telecooperation-Engine, Annotation-Engine und Standard-Applikationen	109
49. Verbindung von Workflow-, Annotation-Engine und Standard Applikation durch Compound Entities	110
50. Arbeitsablauf innerhalb der Referenzarchitektur	113
51. Prozeßstruktur des F&T Dienstes im MMI-System	118
52. CrystalPad-Umgebung	120
53. Kommunikation in der BERKOM-MMC Umgebung mit CrystalPad	121



---

54. Annotation über X-Shape Extension	122
55. Umsetzung des Referenzmodells in VIRTUE	124
56. Virtue-Umgebung	125



## Literaturverzeichnis

- [9] Abdel-Wahab, H. M.;Feit, Mark A.: XTV: A Framework for Sharing X Window Clients in Remote Synchronous Collaboration. X11R5 contrib distribution, contrib/clients/xtv, August 1991.
- [10] Albayrak, S.; Brettschneider, J.; Gessler, S.; Henke, J.; Lux, W.; Peters, R.; Steinbauer, E.: Mensch-Maschine-Schnittstellen im multimedialen Verwaltungsbüro. Meilensteine M1-M6. Technische Berichte. DeTeBerkom, Berlin. 1994-1996.
- [11] Altenhofen, Michael P.: Shared X. Technical report, NESTOR Project, Digital Equipment Corporation, CEC, Karlsruhe, 1990.
- [12] Altenhofen, M. P.; Dittrich, J.; Hammerschmidt, R.; Käppner, T.; Kruschel, C.; Kükkes, A.; Steinig, T.: The BERKOM Multimedia Collaboration Service. In *Proceedings of the ACM Symposium on MultiMedia*. The Association for Computing Machinery, New York, 1993.
- [13] Appelt, W.; Hinrichs, E.; Woetzel, G.: Effectiveness and Efficiency: The Need for Tailorable User Interfaces on the Web. In *Proceedings of the 7th International World Wide Web Conference, WWW'98*. Brisbane, April 1998.
- [14] Bandemer, H.; Gottwald, S.: Einführung in FUZZY-Methoden - Theorie und Anwendung unscharfer Mengen, Band 73, Frankfurt a.M.: Harri Deutsch, 1990
- [15] Bannon, L.; Schmidt, K.: CSCW: Four Characters in Search of a Context. In: Bowers J.M., Benford, S.D. (eds): *Studies in Computer Supported Cooperative Work*. Elsevier Publishers. pp. 3-16. 1991.
- [16] Bapat, A.: POLIWORK - Telekooperation und Dokumentenverwaltung am persönlichen Arbeitsplatz. Der GMD-Spiegel 1/98. pp 38-39. 1998.
- [17] BargainFinder-Homepage: <http://bf.cstar.ac.com/bf>. 1995.
- [18] .Beard, D.; Palaniappan, M.: A Visual Calendar for Scheduling Group Meetings. In: *Proceedings of the Conference on Computer Supported Cooperative Work (CSCW)* pp. 279-290. October 1990.
- [19] Bentley, R.; Horstmann, T.; Sikkell, K. Trevor, J.: Supporting collaborative information sharing with the World Wide Web: The BSCW Shared Workspace System. In

- Proceedings of the 4th International World Wide Web Conference*. Boston, MA. pp 63-74. O'Reilly & Associates. 1995.
- [20] Berger, M.O.; Kubitz, O.: Automatischer Entwurf von Fuzzy Reglern für autonome Agenten mit Evolutionären Algorithmen, <http://alife.informatik.rwth-aachen.de/node3.html>, 1996.
- [21] Berners-Lee, T.; Cailliau, R.; Groff, J.; Pollermann, B.: World-Wide Web: The Information Universe, in *Electronic Networking: Research, Applications and Policy*. Vol.1 No.2, Meckler, Westport CT, Spring 1992.
- [22] Bever, M., Bär, U., Seibt, D., Schmitt, L., Neuhold, E., Knopik, T., Kaack, H., and Engel, A.: POLIWORK: Telekooperation und Dokumentenverwaltung am persönlichen Arbeitsplatz. In Krcmar, H., Lewe, H., and Schwabe, G. (Hrsg.): *Herausforderung Telekooperation*. Fachtagung Deutsche Computer Supported Cooperative Work 1996 (DCSCW '96), Stuttgart-Hohenheim, 30.09.-2.10.1996, pp. 17-34. Springer Verlag, Berlin. 1996.
- [23] Blum, C.; Liebing, A.; Meier, G.-P.; Moeller, E.; Neidecker-Lutz, B.; Neumann, L.; Pusch, H.; Roshandel, M.; Rückert, J.; Scheller, A.; Schürmann, G.; Thomas, S.; Weber, R.; Wolf, F.: The BERKOM Multimedia Mail Teleservice. Technische Berichte. DeTeBerkom GmbH, Berlin. 1994.
- [24] Bonin, E.G.; Rademacher, F.J. (Red.): Der schlanke Staat. Analyse des GI-Ausschuß "Forschung und Technologie". Gesellschaft für Informatik e.V. (GI), Bonn. 1995.
- [25] Bortolan, G.; Degani, R.: Linguistic Approximation of Fuzzy Certainty Factors in Computerized Electrocardiography. In Gupta, M.M.; Yamakawa, T. (Hrsg.): *Fuzzy Computing*. Elsevier Science Publishers B.V., Amsterdam. 1988.
- [26] Brettschneider, J.; Peters, R.; Freitag, U.: Workflow und Mobiles Arbeiten. In Bullinger, H.-J. (Hrsg.): *Dokumenten- und Workflow-Management V*. Fraunhofer IRB Verlag. pp 145-167. November 1997.
- [27] Bundesministerium für Bildung und Forschung (bmb+f): POLIKOM -- Projekt POLITEAM: Entwicklung von Kooperationswerkzeugen zur Unterstützung der Regierungsfunktionen in Berlin und Bonn. <http://orgwis.gmd.de/projects/POLITeam/POLIKOM/politeam.html>. September 1998.
- [28] Bundesministerium des Innern. IT-Unterstützung im Informationsverbund Berlin-Bonn (IVBB). Schriftenreihe der KBSt, Band 30. August 1993.
- [29] Catlin, T., Bush, P., Yankelovich, N.: InterNote: Extending a Hypermedia Framework to Support Annotative Collaboration. In: *Proceedings of Hypertext '89 Conference*. ACM, New York. pp. 365-378. November 1989.

- [30] Collins, J.E.; Sisley, E.M.: Automated Assignment and Scheduling of Service Personnel, *IEEE Expert* 9, 1994.
- [31] Cremers, A.B.; Kahler, H.; Pfeifer, A.; Stiemerling, O.; Wulf, V.: POLITEAM - Kokonstruktive und evolutionäre Entwicklung einer Groupware. *Informatik Spektrum*. Band 21, Heft 4. Springer, Berlin. pp. 194-202. August 1998.
- [32] De Michelis, G., Grasso, M.A.: Routines and Conversations. In: *Structured Programming*, Springer Verlag, (1993) 14. pp. 110-118. 1993.
- [33] De Michelis, G., Grasso, M.A.: Situating Conversations within the Language/Action Perspective: The Milan Conversation Model. In: *Proceedings of the ACM 1994 Conference on Computer Supported Cooperative Work (CSCW'94)*. pp. 89-100. 1994.
- [34] Dietel, C.; Schneider, G.; Schweitzer, J.: POLIVEST Integrierte Televerwaltung. In: *Workshop-Proceedings: Rechnergestützte Kooperation in Verwaltungen und großen Unternehmen*, im Rahmen der 27. Jahrestagung der Gesellschaft für Informatik (Informatik'97), Aachen. September 1997.
- [35] Dorn, J.: Wissensbasierte Echtzeitplanung, *KI* 5, 1991.
- [36] Dourish, P.; Bly, S.: Portholes: Supporting Awareness in a Distributed Work Group. In *Proceedings of CHI '92 Conference on Human Factors in Computing Systems*. Monterey, CA. pp 541 - 547. 1992.
- [37] Ellis, C.A.; Gibbs, S.J.: Concurrency control in groupware systems. In *Proceedings of the ACM SIGMOD Conference on Management of Data*. ACM, New York. pp. 399-407. May 1989.
- [38] Ellis, C.A.; Gibbs, S.J.; Rein, G.L.: Groupware: Some issues and experiences. *Communications of the ACM*, 34(1): 39-58, January 1992.
- [39] Elrod, S.; Bruce, R.; Goldberg, D.; Halasz, F.; Janssen, W.; Lee, D.; McCall, K.; Pedersen, E.R.; Pier, K.; Tang, J.; Welch, B.: Liveboard: a large interactive display supporting group meetings, presentations and remote collaboration. In *Proceedings of CHI '92 Conference on Human Factors in Computing Systems*. Monterey, CA. 1992.
- [40] Encarnação, J.L.; Lockemann, P.C. (eds.): *Engineering Databases*. Springer New York, Berlin. 1990.
- [41] Encarnação, J.L.; Foley J. (eds.): *Multimedia - System Architectures and Applications*. International Communication and Research Centre (IBFI), 1993.
- [42] Encarnacao, J., Hornung, C., Noll, S.: Computer-Supported Cooperative Work (CSCW): Stand und Perspektiven. In: *it+ti - Informationstechnik und technische Informatik*, Ausgabe 36. R. Oldenbourg Verlag. 1994

- [43] Encarnação, J.L.; Noll, S.; Peters, R.: Technologies of Time- and Location Independent Telecooperation. In: *Innovation and Technology XXI*. Kluwer Publication. 1996.
- [44] Engelbart, D.; Lethman, H.: Working Together. In: *Byte* December '88. pp. 245-252. 1988.
- [45] Finin, T.; Labrou, Y.; Mayfield, J.: KQML as an agent communication language. In: Bradshaw, J. (Ed.): *Software Agents*. MIT Press. 1995.
- [46] Finin, T.; Weber, J.; Wiederhold, G.; Genesereth, M.; Fritzson, R.; McKay, D.; McGuire, J.; Pelavin, R.; Shapiro, S.; Beck, C.: Specification of the KQML Agent-Communication Language. The DARPA Knowledge Sharing Initiative – External Interfaces Working Group. University of Maryland Baltimore County. 1993.
- [47] Fraunhofer-Institut für Graphische Datenverarbeitung. Tätigkeitsbericht 1991. Darmstadt. S. 16ff. 1991.
- [48] Frost, H.R.: Documentation for the JAVA Agent Template. <http://cdr.stanford.edu/ABE/documentation/index.html>. 1995.
- [49] Fuchs, L., Sohlenkamp M., Genau, A., Kahler, H., Pfeifer, A., Wulf, V.: Transparenz in kooperativen Prozessen: Der Ereignisdienst in POLITEAM. In: Krcmar, H.; Lewe, H.; Schwabe, G. (ed.): *Herausforderung Telekooperation: Fachtagung Deutsche Computer Supported Cooperative Work (DCSCW '96)*, Stuttgart-Hohenheim, 30.9.-2.10.96. Springer: Berlin. pp. 3-16. 1996.
- [50] Fuchs, L.: Situationsorientierte Unterstützung von Gruppenwahrnehmung in CSCW-Systemen. Dissertation am Fachbereich Mathematik und Informatik der Universität Gesamthochschule Essen. November 1997.
- [51] Gaver, W.W.: Sound Support For Collaboration. In *Proceedings of 2nd European Conference on Computer Supported Cooperative Work (ECSCW'91)*. Kluwer Academic Publishers. pp. 293-308. 1991.
- [52] Genesereth, M.R.; Ketchpel, S.P.: Software Agents. *Communications of the ACM* . 37 (7). pp. 48-53. 1994.
- [53] Goldberg, A.: *Smalltalk-80: The Interactive Programming Environment*. Addison-Wesley Publishers, Menlo Park. 1983.
- [54] Greenberg, S.; Bohnet, R.: GroupSketch: A multi-user sketchpad for geographically-distributed small groups. In *Proceedings of the Graphics Interface Conference*. 1991.
- [55] Greenberg, S.; Roseman, M.; Webster, D.; Bohnet, R.: Issues and Experiences Designing and Implementing two Group Drawing Tools. In *Proceedings of Hawaii International Conference on System Sciences*. pp. 138-150. 1992.

- [56] Greenberg, S.; Marwood, D.: Real Time Groupware as a Distributed System: Concurrency Control and its Effect on the Interface. Research Report 94/543/03, Department of Computer Science, University of Calgary, Alberta, Canada. 1994.
- [57] Greenberg, S. Peepholes: Low Cost Awareness of One's Community. In *Proceedings of CHI'96 Conference*. Short Papers. ACM Press. 1996.
- [58] Grimm, R.: Sicherheit für offene Kommunikation: verbindliche Telekommunikation. In: Rüdiger Grimm: *Sicherheit in der Informations- und Kommunikationstechnik*; Bd. 4, BI-Wiss.-Verl., 1994.
- [59] Grudin, J.: Groupware and Social Dynamics: Eight Challenges for Developers. *Communications of the ACM*, 37(1): 92-105, January 1994.
- [60] Guilfoyle, C.; Warner, E.: Intelligent Agents: The New Revolution in Software. Cutter Information Corp. RP04CD. <http://www.cutter.com/itgroup/reports/agents.htm>. 1994.
- [61] Gutekunst, T.; Schmidt, T.; Schulze, G.; Schweitzer, J.; Weber, M.: A Distributed Multimedia Joint Viewing and Tele-Operation Service for Heterogenous Workstaion Environments. In: Effelsberg, W.; Rothermel, K. (eds.): *Proceedings of the GI/ITG Workshop on Distributed Multimedia Systems*. pp 145-159. Stuttgart. 1993.
- [62] Gutwin, C.; Greenberg, S.: Workspace Awareness for Groupware. Conference Companion. In *Proceedings of ACM Conference on Human Factors in Computing Systems (CHI'96)*. Vancouver. ACM Press. 1996.
- [63] Hartmann, A.; Herrmann, T.; Rohde, M.; Wulf, V. (Hrsg): *Menschengerechte Groupware – Software-ergonomische Gestaltung und partizipative Umsetzung*. German Chapter of the ACM, Berichte 42. B.G.Teubner Verlag, Stuttgart. 1994.
- [64] Haynes, T.; Sen, S.; Arora, N.; Nadella, R.: An Automated Meeting Scheduling Sytsem that Utilizes User Preferences. In *Proceedings of the ACM International Conference on Autonomous Agents*. 1997.
- [65] Herrmann, T.; Wulf, V.; Hartmann, A.: Software-ergonomische Gestaltungsanforderungen für Groupware. In Müller, W.; Senghaas-Knobloch, E. (Hrsg.): *Arbeitsgerechte Software Gestaltung*, S. 193 - 211. Lit Verlag, Hamburg, 1993.
- [66] Höller, H. (Hrsg.). *Kommunikationssysteme - Normung und Akzeptanz*. Braunschweig, 1993.
- [67] International Telecommunication Union: Recommendation H.320 (03/96) - Narrow-band visual telephone systems and terminal equipment. [http://www.itu.ch/itudoc/itu-t/rec/h/h320\\_23397.html](http://www.itu.ch/itudoc/itu-t/rec/h/h320_23397.html). 1996.

- 
- [68] International Telecommunication Union – Telecommunication Standardisation Sector – Study Group 8. T.120 Series. 1996.
- [69] International Telecommunication Union – Telecommunication Standardisation Sector – Study Group 8. Draft Recommendation T.124 – Generic Conference Control. 1995.
- [70] Ishii, H.: TeamWorkStation: Towards a Seamless Shared Workspace. In: *Proceedings of the Conference on Computer-Supported Cooperative Work CSCW '90*, October 7-10, 1990, Los Angeles, CA, pp. 13-27.
- [71] Ishii, H.; Kobayashi, M.: Integration of Interpersonal space and Shared Workspace: ClearBoard Design and Experiments. *ACM Transaction off Information Systems*. V11, N4, pp. 349-375. 1993.
- [72] Jablonski, S.: *Workflow-Management Systeme: Modellierung und Architektur*. Intern. Thomson Publ., Thomson's Aktuelle Tutorien. Bonn 1995.
- [73] Janca, P: *Intelligent Agents: Technology and Application*, GiGa Information Group. 1996.
- [74] Jennings, N.R.; Sycara, K.P.; Wooldridge, M.: A roadmap of agent research and development. *Journal of Autonomous Agents and Multi-Agent Systems*, 1(1):7-38, July 1998.
- [75] Johansen, R.: *Groupware: Computer Support for Business Teams*. In: The Free Press, New York, 1988.
- [76] Jones, Oliver: *Multidisplay Software in X. The X Resource, Issue Six*. O'Reilly and Associates, Inc., 1993.
- [77] Jung, V.: Knowledge-based visualization design for geographic information systems. In *Proceedings of 3rd ACM International Workshop on Advances in Geographic Information Systems*. Baltimore, MD, Dec.95. pp 101-108. ACM Press, New York, 1995.
- [78] Kaminka, G.A.; Tambe, M.: Social Comparison for Failure Detection and Recovery. In *Proceedings of the 4th International Workshop on Agent Theories, Architectures and Languages (ATAL97)*. Providence, RI. USA. July 1997.
- [79] Karl, R., Deiters, D.: *Workflow Management - Groupware Computing 2*. Studie. dsk Beratungs-GmbH und Fraunhofer-Institut für Software- und Systemtechnik, Pfaffenhofen, 1995.
- [80] Koch, O., Zielke, F.: *Workflow Management - Prozeßorientiertes Arbeiten mit der Unternehmens-DV*, Haar bei München, 1996.



- [81] Krallmann, H.; Derszteler, G.: Workflow Management Cycle - An Integrated Approach to the Modelling, Execution, and Monitoring of Workflow Based Processes. In: Scholz-Reiter, B.; stickel, E. (ed.) *Business Process Modelling*. Springer-Verlag. pp 23-42. 1996.
- [82] Krallmann, H.: Systemanalyse im Unternehmen. 2. Auflage. R. Oldenbourg Verlag. 1996.
- [83] Krasner, G.E.; Pope, S.T.: A Cookbook for using the Model-View-Controller User Interface Paradigm in Smalltalk-80. Technical Report. ParcPlace Systems, Palo Alto, CA. 1988.
- [84] Kruse, R.; Gebhardt, J.; Klawonn, F.: Modellierung von Vagheit und Unsicherheit - Fuzzy Logik und andere Kalküle. In KI 4, 1991.
- [85] Lange, D.B.; Chang, D.T.: IBM Aglets Workbench: Programming Mobile Agents in JAVA. IBM White paper, IBM Corporation, Japan. <http://www.trl.ibm.co.jp/aglets/whitepaper.htm>. 1996.
- [86] Litke, H.-D.: Gute Planung - hoher Nutzen, in: *Business Computing*, 7/95. 1995.
- [87] Little, Arthur D.: Multimedia: Europa am Scheideweg. Marktstudie. Editor: W. Knetsch. Berlin 1994.
- [88] Lu, I.M.; Mantei, M.M.: Idea Management in a Shared Drawing Tool. In *Proceedings of ECSCW '91*. Amsterdam. Kluwer Academic Publishers, London. pp. 97-112. 1991.
- [89] Malm, Pal S.: The unOfficial Yellow Pages of CSCW. Technical report, University of Tromso, January 1994.
- [90] Mattern, F.: Mobile Agenten. *it+ti - Informationstechnik und Technische Informatik*, 4/98. R. Oldenbourg Verlag. pp. 12-17. 1998.
- [91] Mayer, A.; Mechler, B.; Schlindwein, A.; Wolke, R.: Fuzzy Logic, Einführung und Leitfaden zur praktischen Anwendung. Addison-Wesley Publisher, Bonn, 1993.
- [92] McCarthy, J.C.; Bluestein, W.M.: The Computing Strategy Report: Workflow's Progress. Forrester Research Inc., Cambridge, Mass. Oct. 1991.
- [93] McKinsey & Company Inc.: Zusammenhänge: Die Entwicklung der Informationstechnologie - Management des Wandels in einer Zeit des Paradigmenwechsels. Düsseldorf 1993.
- [94] Meinardus, G.; Merz, G.: Praktische Mathematik I: für Ingenieure, Mathematiker und Physiker. Bibliographisches Institut, Mannheim/Wien/Zürich. 1979.

- [95] Minenko, W.; Schweitzer, J.: Transparentes Application-Sharing unter X Window - Synchroner Telekooperation für das Team von morgen. *unix/mail* 12 (94) 4. pp. 348-357. 1994.
- [96] Mizumoto, M.: Fuzzy Controls Under Various Fuzzy Reasoning Methods, in *Information Science* 45, 1988.
- [97] Moeller, E.; Neumann, L.; Schürmann, G.; Thomas, S.; Weber, R.; Wolf, F.: The BERKOM Multimedia Mail Teleservices. *Computer Communications Journal*. 1994.
- [98] Munson, Jonathan P.; Dewan, Prasun: A Flexible Object Merging Framework. In *Proceedings of the CSCW '94 Conference, Chapel Hill*. ACM, New York. pp. 231-242. 1994.
- [99] Noll S.; Schendel, M.G.: Cooperative sketching in a network environment for the automotive industry in europe. In *Proceedings of the Eurographics 1991, Technical Report Series, Vienna*, 1991.
- [100] Noll, S.; Paul, C.; Peters, R.; Schiffner, N.: Autonomous Agents in Collaborative Virtual Environments. In *Proceedings of IEEE WETICE Workshop on Knowledge Media Networking*. Stanford University, June 1999.
- [101] Open Group CAE Specification: Calendaring and Scheduling API (XCS). C321. ISBN 1-85912-076-8 April 1995
- [102] Palm, R.; Hellendorn, H.: Fuzzy-Control - Grundlagen und Entwicklungsmethoden. In *KI* 4, 1991.
- [103] Pankoke-Babatz, U.: Awareness: Spannungsfeld zwischen Beobachter und Beobachtetem. In Pankoke-Babatz, U.; Prinz, W.: *Proceedings des Workshops Von Groupware zu GroupAware der D-CSCW '98*. Dortmund. pp. 5-11. September 1998.
- [104] Paul, C.; Spriestersbach, A.; Peters, R.: Intelligente Agenten für virtuelle Umgebungen. In *Proceedings of the Workshop Agenten, Assistenten, Avatars*. Darmstadt, Germany. 1997.
- [105] Peters, R.: Konzeption eines generischen Annotationswerkzeuges. Technischer Bericht 93i011-FIGD. Fraunhofer-Institut für Graphische Datenverarbeitung, Darmstadt, Juli 1993.
- [106] Peters, R.; Albayrak, S.; Brettschneider, J.; Gessler, S.; Henke, J.; Lux, W.; Steinbauer, E.: Mensch-Maschine-Schnittstellen im multimedialen Verwaltungsbüro. Siehe [10].

- [107] Peters, R.; Neuss, Ch.; Bernhard, M.: CrystalPad: Shared Authoring with Annotations. *Proceedings of poster session, IWACA '94 Workshop*, Heidelberg. September 1994.
- [108] Peters, R., Kress, H.: An approach to integrate time and location-independent cooperation in a distributed product development process. In *Proceedings of the International Workshop on Integrated Broadband Communications and Collaborative Work in the Automotive Industry, Trento*. 1995.
- [109] Peters, R.; Neuss, Ch.: CrystalWeb: A Distributed Authoring Environment for the World Wide Web. In: *Proceedings of the Third International World Wide Web Conference, WWW '95. Darmstadt*. Computer Networks and ISDN Systems, Volume 27, Issue 6, pp 861-870. Elsevier Science B.V., Amsterdam. April 1995.
- [110] Peters, R.; Encarnação, J.L.; Noll, S.: Technologies of Time- and Location Independent Telecooperation. Siehe [43]. 1996.
- [111] Peters, R.: Computerunterstützte kooperative Arbeitsprozesse in Büro und Verwaltung. *it+ti - Informationstechnik und technische Informatik 1/97*. R. Oldenbourg Verlag. pp 21-28. 1997.
- [112] Peters, R.; Kress, H.: Distributed Synchronous Interaction Examined Closely. *Concurrent Engineering Research and Applications*. Volume 5. No.3. pp. 219-232. September 1997.
- [113] Peters, R.; Graeff, A.; Paul, C.: Integrating Agents into Virtual Worlds. In *Proceedings of the ACM SIGGRAPH Workshop on new Paradigms in Information Visualisation and Manipulation (NPV'97)*, Las Vegas, NV, USA. November 1997.
- [114] Peters, R.; Brettschneider, J.; Freitag, U.: Workflow und Mobiles Arbeiten. Siehe [26]. November 1997.
- [115] Peters, R.; Stiemerling, O.; Wulf, V.; Cremers, A.B.; Buelens, C.; Boulliard, G.; Maidhoff, M.; Bancroft, C.; Frickers, F.; Durand, C.: TransIT - Investigating SME cooperation across borders in Europe. Siehe [140]. 1998.
- [116] Peters, R.; Iacucci, G.; Stiemerling, O.; Wulf, V.: Telecooperation Systems in Engineering Companies Supplying the Metallurgy Industry. In: Iacucci, G. et al. (ed.): *Globalization of Manufacturing in the Digital Communications Era of the 21st Century*. Proceedings of PROLAMAT '98 IFIP WG 5.2/5.3 Conference, Trento, Italy. Kluwer Academic Publishers. pp. 107-119. September 9-12, 1998.
- [117] Peters, R.; Rausch, M.: Agent-guided Collaboration in Groupware Environments. In: Lees, B.; Müller, H.J.; Branki, C. (ed.): *Proceedings of the International ACM/GI Workshop on Intelligent Agents in CSCW*. Dortmund. pp. 62-74. September 1998.

- [118] Peters, R.; Maidhof, M.; Rix, J.: 3D CAD and Multimedia Telecooperation within the Metallurgy Industry. In *Proceedings of the International Conference of CAD/CAM*. Neuchâtel, Switzerland. February 22-24, 1999.
- [119] Peters, R.; Noll, S.; Paul, C.; Schiffner, N.: Autonomous Agents in Collaborative Virtual Environments. Siehe [100]. 1999.
- [120] Peters, R.; Wulf, V.; Matthias Krings; Stiemerling, O.; Iacucci, G.; Fuchs-Fronhofen, P.; Hinrichs, J.; Maidhof, M.; Nett, B.: Improving Inter-Organizational Processes with Integrated Organization and Technology Development. Siehe [163]. 1999.
- [121] Peters, R.: Innovative Kooperation Unternehmensübergreifend. Akzeptiert für it+ti - Informationstechnik und technische Informatik. R. Oldenbourg Verlag. 2000. (*to be published*)
- [122] Plattner, B. et.al.: Elektronische Post und Datenkommunikation - X.400: Die Normen und Ihre Anwendung. Addison Wesley, 3. Aufl. 1993.
- [123] Prinz, W.; Kolvenbach, S.: Support for Workflows in a Ministerial Environment. In: Ackermann, M. S. (ed.): *Proceedings of the ACM 1996 Conference on Computer Supported Cooperative Work*, Boston, Massachusetts, USA. ACM Press, New York. pp. 199-208. November 1996.
- [124] Prinz, W.; Mark, G.; Pankoke-Babatz, U.: Designing Groupware for Congruency in Use. In *Proceedings of CSCW'98*, Seattle, USA. pp. 373-382. ACM Press. 1998.
- [125] Rath, H.H.: "Literate Specifying" von Hypermedia-Dokumenten und die Selbstbeschreibung des Konzepts. Darmstädter Dissertation D17. Shaker Verlag, Aachen. 1996.
- [126] Ricke, H.; Kanzow, J.: BERKOM - Breitbandkommunikation im Glasfasernetz. R.v.Decker's Verlag, 1991.
- [127] Riexinger, D.; Fehr, C.: Application Sharing Based on Bitmap Exchange. In: *Proceedings of the International Conference on Multimedia Networking, MmNet '95*, Aizu, Japan. IEEE-CS Press, pp 76-85. September 1995.
- [128] Rodden, Tom: A survey of CSCW systems. *Interacting with Computers*, Vol. 3, No. 3., pp. 319-353. 1991..
- [129] Röscheisen, M., Mogensen C., Winograd, T.: A platform for Third-Party Value-Added Information Providers: Architecture, Protocols, and Usage Examples. Technical Report CSDTR/DLTR. Computer Science Department. Stanford University, CA. 1995.

- [130] Roßnagel, A.: Vom informationellen zum kommunikativen Selbstbestimmungsrecht. In Kubicek (Hrsg.): *Telekommunikation und Gesellschaft. Kritisches Jahrbuch der Telekommunikation*. C.F.Müller. Karlsruhe. 1991.
- [131] Roth, V.; Jalali M.: Access Control and Key Management for Mobile Agents. Special Issue Data Security in Image Communication and Networks. *Computers & Graphics*. Vol. 22, No. 3. Pergamon Press. New York. 1998.
- [132] Santos, A.: Multimedia and groupware for editing. *Computer Graphics: Systems and Applications*. Springer-Verlag, Berlin Heidelberg 1995.
- [133] Schmidt, A.; Specker, A.; Partsch, G.; Weber, M.: Agents, Brokers, Traders, and Services in Cooperative Systems. In *Proceedings of the Third International Conference on the Design of Cooperative Systems, COOP'98*, May 1998.
- [134] Scrivener, S.A.R.; Clark, S.M.; Keen, N.: The LookingGlass Distributed Shared Workspace. *Computer Supported Cooperative Work*, Vol. 2, No. 3, pp. 137-157. Kluwer Academic Publishers. 1994..
- [135] ShrEdit 1.1: a shared editor for the Apple Macintosh. User's guide and technical description. Cognitive Science and Machine Intelligence Laboratory. University of Michigan. 1990.
- [136] Sohlenkamp, M.; Chwelos, G.: Integrating Communication, Cooperation and Awareness: The DIVA Virtual Office Environment. In *Proceedings of the Conference on Computer-Supported Cooperative Work CSCW '94*. pp. 331-343. 1994.
- [137] Sohlenkamp, M.; Prinz, W.; Fuchs, L.: PoliAwaC - Design und Evaluation des Poli-Team Awareness Clients. In: Th. Herrmann; K. Just-Hahn (eds.) *Groupware und organisatorische Innovation. Tagungsband D-CSCW'98*. B.G. pp 181-194. Teubner, Stuttgart, Leipzig. 1998.
- [138] Stallings, W.: *SNMP, SNMPv2, and CMIP: The Practical Guide to Network-Management Standards*. Addison-Wesley, 1993.
- [139] Stefik, M. et al.: Beyond the Chalkboard: Computer Support for Collaboration and Problem Solving in Meetings. *Communications of the ACM*, 30(1): 32-47, January 1987.
- [140] Stiemerling, O.; Wulf, V.; Cremers, A.B.; Buelens, C.; Boulliard, G.; Maidhoff, M.; Peters, R.; Bancroft, C.; Frickers, F.; Durand, C.: TRANSIT - Investigating SME cooperation across borders in Europe, *ACM SIGGROUP-Bulletin Vol 19 No 2*. pp. 35-37. 1998.
- [141] Sugeno, M.: An Introductory Survey of Fuzzy Control. In: *Information Sciences* 36, 1985.

- [142] Tazelaar, J.M.: In Depth: Groupware. In: Byte December '88. pp. 242-282. 1988
- [143] Tilli, T.: Fuzzy-Logik: Grundlagen, Anwendungen, Hard- und Software. Franzis, München. 1992.
- [144] Thimbleby. User Interface. Reading, ACM Press Frontier Series. MA. 1990.
- [145] Traeger, D.H.: Einführung in die Fuzzy-Logik. B.G. Teubner, Stuttgart. 1993.
- [146] Trigg, R.H.; Suchman, L.A.; Halasz, F.G.: Supporting Collaboration in NoteCards. In *Proceedings of the First Conference on Computer Supported Cooperative Work, Austin*. ACM. pp. 153-162. 1986.
- [147] Tripp, H.: Einführung von CSCW-Technologien in kleinen und mittleren Unternehmen, Diplomarbeit TH Darmstadt, 1996.
- [148] Turetti, T.: H.261 Software Codec for Videoconferencing over the Internet. INRIA. Research Report No. 1834. 1993.
- [149] van Hoof, A.: Imaging und Archivierung. In: *Software Management und Corporate Computing*. IAO-Seminare. Fraunhofer-Institut für Arbeitswissenschaft und Organisation, Stuttgart. Juli 1996.
- [150] Verein Deutscher Ingenieure (VDI). Richtlinien: Bürokommunikation. Software Ergonomie in der Bürokommunikation. VDI-Verlag, Düsseldorf, 1988.
- [151] Vertegall, R.: Conversational Awareness in Multiparty VMC. Proceedings of the ACM CHI'97 Conference on Human Factors in Computing Systems. Atlanta, GA. 1997.
- [152] Vertegaal, R.: The GAZE Groupware System: Mediating Joint Attention in Multiparty Communication and Collaboration. In *Proceedings of ACM CHI'99 Conference on Human Factors in Computing Systems*. Pittsburgh, PA USA: ACM, 1999.
- [153] Webhound-homepage: <http://webhound.www.media.mit.edu/projects/webhound/>
- [154] Weber, M.; Partsch, G.; Scheller-Houy, A.; Schweitzer, J.; Schneider, G.: Flexible Einbindung von Telekonferenzen in Workflowmanagement. In: *Tagungsband Workshop Geschäftsprozeßmodellierung und Workflowsysteme*, Informatik '96, Klagenfurt. 1996.
- [155] Weber, M.; Schneider, G.; Partsch, G.; Höck, S.; Scheller-Houy, A.; Schweitzer, J.: Integrating Synchronous Multimedia Collaboration into Workflow Management. In *Proceedings of GROUP 97 International Conference on Supporting Group Work: The Integration Challenge*. Phoenix, Arizona, USA. pp. 281-290. November 1997.

- [156] White, J.: Mobile Agents Whitepapare. General Magic. <http://www.genmagic.com/agents/Whitepapaer/whitepaper.html>. 1996.
- [157] Winograd, Terry: A language/action perspective on the design of cooperative work. In: *Proceedings of the First Conference on Computer Supported Cooperative Work, Austin*. ACM. pp. 203-220. 1986.
- [158] Wooldridge, M.; Jennings N.R.: Intelligent Agents: Theorie and Practice. The Knowledge Engineering Review 10 (2). 1995.
- [159] Wooldridge M.; Mueller J.P., Tambe M. (edt.): Intelligent Agents II. Volume 1037 of Lecture Notes in Artificial Intelligence. Springer-Verlag. January 1996.
- [160] Wooldridge, M.; Jennings, N.R.: Pitfalls of agent-oriented development. In Sycara, K.P.; Wooldridge, M. (eds.) *Proceedings of the Second International Conference on Autonomous Agents*. ACM Press, May 1998.
- [161] Workflow Management Coalition: The Workflow Reference Model. Document Number TC00-1003. Workflow Management Coalition, Avenue Marcel Thirty 204, 1200 Brussels. November 1994.
- [162] Workflow Management Coalition: Workflow Client Appliaction (Interface 2). Document Number WFMC-TC-1009. Workflow Management Coalition, Avenue Marcel Thirty 204, 1200 Brussels. October 1996.
- [163] Wulf, V.; Matthias Krings; Stiemerling, O.; Iacucci, G.; Fuchs-Fronhofen, P.; Hinrichs, J.; Maidhof, M.; Nett, B.; Peters, R.: Improving Inter-Organizational Processes with Integrated Organization and Technology Development. In *Journal of Universal Computer Science*, 1999.
- [164] Zadeh, L.A.: Fuzzy Sets, in: *Information and Control* 8, 1965
- [165] Zadeh, L.A.: The Concept of a Linguistic Variable and Ist Application to Approximate Reasoning - I + II. *Information Sciences* 8, 1975