

**Synchrone Groupware für die Software-Engineering-Ausbildung  
- Ein Beispiel für die Ableitung unterstützender Werkzeuge  
aus problemorientierter Sicht –**

Vom Fachbereich Elektrotechnik der  
Gerhard-Mercator-Universität – Gesamthochschule Duisburg

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

von

Diplom Ingenieur Stefan Werner

aus

Duisburg-Rheinhausen

Referent: Prof. Dr.-Ing. Axel Hunger

Korreferent: Prof. Dr.-Ing. Hans-Dieter Kochs

Tag der mündlichen Prüfung: 06.12.2002



für Dana, Lucy Madeleine und Noah Maria



## Vorwort

Die vorliegende Arbeit entstand neben meiner Tätigkeit als wissenschaftlicher Mitarbeiter im Fachgebiet Datenverarbeitung an der Gerhard-Mercator Universität Duisburg (heute Universität Duisburg-Essen). Unterstützung habe ich dabei auf vielfältige Weise von Kollegen, Freunden, Bekannten und meiner Familie erfahren.

Im Umfeld der Hochschule gilt mein besonderer Dank Prof. Dr.-Ing. Axel Hunger für seinen Beitrag zum Auffinden des „richtigen Weges“ zu Beginn meiner Arbeiten, die fortwährende Betreuung und die wertvollen Hinweise und Anregungen in der Schlussphase. Sein Reformwille und das rechtzeitige Erkennen der „Zeichen der Zeit“ lieferten wichtige Beiträge zur Internationalisierung der Gerhard-Mercator Universität Duisburg und die Neuausrichtung des ehemaligen Fachgebietes Datenverarbeitung hin zur Medientechnik. Erst dadurch wurde die inhaltliche Grundlage für zahlreiche medientechnische Projekte innerhalb unserer Arbeitsgruppe geschaffen, die auch meine eigene Ausrichtung maßgeblich beeinflussten und schließlich diese Arbeit überhaupt erst ermöglicht haben. Ebenfalls möchte ich mich bei Prof. Dr.-Ing. Hans-Dieter Kochs für die Übernahme des Korreferates bedanken.

Darüber hinaus gilt mein Dank den zahlreichen Studierenden die mit ihren Studien- und Diplomarbeiten sowie während ihrer Tätigkeit als studentische Hilfskräfte wesentlich zum Gelingen der Arbeit beigetragen haben. Für ihre Unterstützung bedanke ich mich daher bei Dipl.-Ing. Frank Schwarz, Dipl.-Ing. Udo Walter, Dipl.-Ing. Jacek Januschkiwitz, Dipl.-Ing. Markus Jung und Christoph Töller. Mein besonderer Dank gilt Dipl.-Ing. Stefan Freinatis für die zahlreichen Gespräche während des „Endspurts“.

Für die Korrektur der schriftlichen Arbeit bedanke ich mich bei meinem Freund Lothar Swiderek, der durch seinen unermüdlichen Einsatz auch dazu beigetragen hat, dass ich meinen eigenen Zeitplan einhalten konnte.

Schließlich gebührt ganz besonderer Dank meinem privaten Umfeld, allen voran meiner Familie. Meinen Eltern, Maria und Franz Werner, die mir nicht nur meinen Weg

ermöglichten sondern auch stets daran glaubten das ich das Ziel erreiche. Ein besonders herzliches Danke Schön natürlich auch meiner Frau Dana und meinen Kindern Lucy und Noah. Sie haben die Verschiebung meiner Prioritäten zugunsten meiner Promotion nicht nur toleriert sondern mir mit viel Entgegenkommen, Verständnis und Liebe ein Umfeld geschaffen haben, das sich jeder Doktorand, gerade in der Schlussphase seiner Arbeit, überhaupt nur wünschen kann.

Duisburg, im Januar 2003

Stefan Werner

# Inhaltsverzeichnis

Abkürzungsverzeichnis .....	v
Abbildungsverzeichnis .....	vii
Tabellenverzeichnis .....	ix
<b>1. EINLEITUNG.....</b>	<b>1</b>
1.1 Motivation .....	2
1.2 Problemstellung und Ziele .....	3
1.3 Gliederung der Dissertation .....	5
<b>2. RECHNERGESTÜTZTE GRUPPENARBEIT – EIN ÜBERBLICK.....</b>	<b>7</b>
2.1 Begriffsdefinitionen .....	8
2.2 Klassifikationen .....	10
2.2.1 Gruppen und Gruppenprozesse .....	10
2.2.2 Klassifikation von CSCW-Systemen.....	13
2.3 Humankommunikation.....	17
2.4 Gruppenkommunikation im IPv6-Internet.....	22
2.4.1 Technische Aspekte der Gruppenkommunikation .....	23
2.4.2 Multicast-Kommunikation .....	24
2.4.3 Das Internet Protokoll Version 6 .....	25
2.4.4 Das Internet Control Message Protocol .....	29
2.4.5 Migration von IPv4 zu IPv6 .....	30
2.4.6 Grundlegende Routing-Algorithmen des IPv6-Internet .....	31
2.4.7 Multicast-Routing im Internet .....	33
2.4.8 Application Level Framing.....	35
2.4.9 Dienstgarantien im Internet .....	36
2.5 Computer Supported Cooperative Working .....	38
2.5.1 Grundlegende Eigenschaften von CSCW-Systemen.....	38
2.5.2 Gruppenwahrnehmung.....	40
2.5.3 Kooperationsunterstützung in synchronen CSCW-Systemen .....	42
2.5.4 Koordinationsunterstützung in synchronen CSCW-Systemen .....	46
2.5.5 Kommunikationsunterstützung in synchronen CSCW-Systemen .....	48
<b>3. ENTWURF, IMPLEMENTATION UND EVALUATION VON GROUPWARE.....</b>	<b>50</b>
3.1 Ein Vorgehensmodell für die Entwicklung von Groupware .....	51
3.2 Methoden zum Studium von Gruppen.....	52

3.3	Modellierung von Groupware .....	53
3.3.1	Architekturmodelle für Groupware .....	53
3.3.2	Verteilungsarchitekturen .....	59
3.3.3	Implementation von Groupware.....	61
3.3.4	Evaluation von Groupware.....	62
<b>4.</b>	<b>STAND DER TECHNIK BEI DER EINFÜHRUNG VON GROUPWARE IM RAHMEN DER SOFTWARE-ENGINEERING-AUSBILDUNG.....</b>	<b>66</b>
4.1	Software-Engineering .....	66
4.2	Kooperatives Software-Engineering in der industriellen Praxis.....	68
4.2.1	Software-Entwicklung mit parallelen Teams am gleichen Ort.....	69
4.2.2	Software-Entwicklung mit virtuellen Teams .....	70
4.2.3	Anforderungen an Werkzeuge zum kooperativen Software-Engineering.....	70
4.2.4	Werkzeuge zur Unterstützung des kooperativen Software-Engineering .....	73
4.2.5	Zusammenfassung.....	81
4.3	Aktueller Stand der Einführung verteilter, kooperativer Arbeitsformen in die Software-Engineering-Ausbildung.....	82
4.3.1	Studien im Rahmen des JTAP-Projektes .....	83
4.3.2	Infrastrukturuntersuchungen der University of Arizona.....	90
4.3.3	Gemeinsames Pilotprojekt der TU-Darmstadt und der GMD .....	92
4.3.4	Einsatz von Lotus Notes in einem Projektseminar der Universität Bern .....	93
4.3.5	Diskussion des aktuellen Standes .....	93
4.4	Eigene Anforderungen an Groupware zum Einsatz in der Software-Engineering- Ausbildung.....	97
4.5	Zusammenfassung.....	100
<b>5.</b>	<b>EIGENER ANSATZ FÜR EINE SYNCHRONE GROUPWARE ZUM EINSATZ IN DER SOFTWARE-ENGINEERING-AUSBILDUNG .....</b>	<b>102</b>
5.1	Eigene Voruntersuchungen zum Studium der Gruppen.....	103
5.1.1	Ableitung eines Prozessmodells für das Praktikum Programmwurftechnik .....	105
5.1.2	Ableitung eines Gruppenprozessmodells für das Praktikum Programmwurftechnik .....	107
5.1.3	Szenario für den Einsatz der synchronen Groupware PASSENGER .....	113
5.2	Architektur der synchronen Groupware PASSENGER .....	113
5.2.1	Auswahl eines Architekturmodells.....	117
5.2.2	Auswahl einer Verteilungsarchitektur .....	118
5.2.3	Entwicklung einer Systemarchitektur .....	119
5.2.4	Floor-Kontrolle .....	121
5.2.5	Bewertung der entwickelten Floor-Kontrolle .....	126
5.3	Benutzerschnittstelle der PASSENGER Client Applikation .....	127
5.3.1	Grundlegende Designentscheidungen.....	128



5.3.2	Funktionsumfang .....	130
5.4	Synchrone Kommunikation mit dem PASSENGER-Client .....	131
5.4.1	Designentscheidungen .....	132
5.4.2	Ablauf der Kommunikation .....	134
5.4.3	Gruppenwahrnehmung .....	135
5.4.4	Chat .....	138
5.4.5	Bewertung des Benutzerschnittstellenentwurfs .....	139
5.5	Kooperation mit dem PASSENGER CASE Tool .....	140
5.5.1	Konzeption .....	140
5.5.2	Datenmodell und Dateiformat .....	143
5.5.3	Funktionsweise .....	144
5.5.4	Gruppenwahrnehmung .....	147
5.5.5	Bewertung des Whiteboard-Konzeptes .....	148
5.6	Protokolle und Dienste .....	149
5.6.1	Das PASSENGER Realtime Transport Protocol .....	149
5.6.2	Das PASSENGER Session and Collaboration Protocol .....	160
5.6.3	Das PASSENGER Multimedia Transport System .....	161
5.7	Implementation .....	164
<b>6.</b>	<b>ERSTE ERFAHRUNGEN UND BEWERTUNGEN .....</b>	<b>166</b>
6.1	Untersuchungsdesign .....	167
6.2	Untersuchungsergebnisse .....	172
6.3	Zusammenfassung der Untersuchungsergebnisse .....	178
<b>7.</b>	<b>ZUSAMMENFASSUNG UND AUSBLICK .....</b>	<b>180</b>
7.1	Ergebnisse der Arbeit .....	181
7.2	Weiterführende Arbeiten .....	182
	Literaturverzeichnis .....	184



## Abkürzungsverzeichnis

ACM	Audio-Codec-Management
ADU	Application Data Unit
ALAN	Application Level Active Networks
ALF	Application Level Framing
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
BSCW	Basic Support for Cooperative Work
CACTUS	Computer Assisted Communication Tool for Urgent Support
CASE	Computer Aided Software Engineering
CPU	Central Processing Unit
CSCS	Computer Supported Collaborative Learning
CSCSE	Computer Supported Cooperative Software Engineering
CSCW	Computer Supported Cooperative Work
CSDE	Cooperative Software Development Environment
DAAD	Deutscher Akademischer Austauschdienst
DFN	Deutsches Forschungs Netz
DHCP	Dynamic Host Configuration Protocol
DiffServ	Differentiated Services
DNS	Domain Name Service
DVK	Desktop-Video-Konferenzsystem
DVMRP	Distance-Vector-Multicast-Routing-Protocol
E-Mail	Electronic Mail
FLECSE	Flexible Environment for Collaborative Software-Engineering
FTP	File Transfer Protocol
GMD	Gesellschaft für Mathematik und Datenverarbeitung
GMU	Gerhard-Mercator-Universität Duisburg
GSM	General Standard for Mobile Communication
GU	Global-Unicast-Adressen
HTML	Hypertext Markup Language
IBR	Institut für Betriebssysteme und Rechnerverbund
ICICLE	Intelligent Code Inspection Environment in a C Language Environment
ICMP	Internet Control Message Protocol
IETF	Internet Engineering Task Force
IGMP	Internet Group Management Protocol
IntServ	Integrated Services
IP	Internet-Protokoll
IPSI	Institut für integrierte Publikations- und Informationssysteme
ISO	International Organization for Standardization
IT	Informationstechnik
JISC	Joint Information Systems Committee
JTAP	JISC Technology Applications Programme
KU	Keele University
KD	Kollaborativer Dienst
LAN	Local Area Network

LLU	Link-Lokal-Unicast
MACS	Modular Advanced Collaboration System
Mbone	Multicast Backbone
MME	Multimedia-Extention
MMTng	Multimedia Transport System/ next generation
MTU	Maximum Transfer Unit
MVC	Model-Viewer-Controller
Ngtrans	Next Generation Transition
NNAT	No Network Address Translation
NTU	National Technology Universtity
OSI	Open-Systems Interconnection
OSPF	Open-Shortest-Path-First
PADU	PASSENGER Application Data Unit
PC	Proposed Changes
PFCM	PASSENGER Floor Control Model
PL	Permission List
PMMT	PASSENGER Multimedia Transportsystem
PSCP	PASSENGER Session and Collaboration Protocol
PWDM	PASSENGER Whiteboard Data Model
RFC	Request for Comments
RIP	Routing-Information-Protocol
RPF	Reverse Path Forwarding
RSVP	RessourceReSerVation Protocol
RTP	Realtime Transport Protocol
SEG	Software-Engineering Group
SLU	Site-Lokal-Unicast
SPE	Snart Programming Environment
ST2	Stream Protocol Version 2
SWS	Semesterwochenstunden
TCP	Transmission Control Protocol
TelSEE	Tele-Software-Engineering-Editor
TPM	Team Performance Modell
TRPB	Truncated Reverse Path Broadcasting
TTL	time-to-live
TUD	TU-Darmstadt
UB	Universität Bern
UDP	User Datagram Protocol
UMIST	University of Manchester
UoA	University of Arizona
UoD	University of Durham
VCL	Visual Component Library
WAN	Wide Area Network
WWW	World-Wide-Web
WYSIWIS	What You See Is What I See
ZMAAP	Zeroconf Multicast Address Allocation Protocol

## Abbildungsverzeichnis

Abb. 1: CSCW-Forschung im Überblick [War98] .....	9
Abb. 2: Raum-Zeit-Matrix der Gruppenarbeit .....	14
Abb. 3: Groupware Klassifikation nach Raum und Zeit.....	14
Abb. 4: Stufen Modell für die zunehmende Intensität der Kommunikation [Bor98] .....	16
Abb. 5: Klassifikation nach dem 3K-Modell [Bor98] .....	17
Abb. 6: Kommunikationsmodell einer Online-Kommunikation [Her01] .....	18
Abb. 7: Komponenten eines Kooperationssystems nach [Mis99] .....	21
Abb. 8: Einsatz eines Kooperationssystems [Mis99] .....	22
Abb. 9: Beziehung der Kommunikationsformen nach [Wit99] .....	24
Abb. 10: Aufbau der IPv6-Multicast-Adressen .....	28
Abb. 11: Vorgehensmodell für die Entwicklung von Groupware [Krc91] .....	51
Abb. 12: Die vier Ebenen in Pattersons Architekturmodell.....	54
Abb. 13: Hybride Architektur nach Patterson .....	55
Abb. 14: Das MVC-Konzept [Jan01].....	56
Abb. 15: Das NetMVC-Modell nach [Mah99].....	58
Abb. 16: Ein Bewertungsmodell für Laborstudien [Gap97] .....	64
Abb. 17: Prozessmodell des Praktikums Programmwurftechnik.....	106
Abb. 18: Ablauf während des Praktikums in Anlehnung an die Darstellung in [Bor98] .....	110
Abb. 19: Client/Server-Architektur der synchronen Groupware PASSENGER .....	116
Abb. 20: Das Architekturmodell der synchronen Groupware PASSENGER .....	118
Abb. 21: Verteilungsarchitektur der synchronen Groupware PASSENGER .....	119
Abb. 22: Systemarchitektur der synchronen Groupware PASSENGER .....	120
Abb. 23: Zustandsübergangsdiagramm des PFCM .....	124
Abb. 24: Benutzerschnittstelle des PASSENGER-Client .....	130
Abb. 25: Funktionsklassen.....	131
Abb. 26: Kommunikationsfenster des PASSENGER-Clients .....	133
Abb. 27: Zwischenruffenster .....	135
Abb. 28: Visualisierung der Permission-List-Einträge.....	138
Abb. 29: Visualisierung der verfügbaren Funktionen.....	138
Abb. 30: Integriertes Chat-Werkzeug im PASSENGER Client.....	139
Abb. 31: Whiteboard-Konzept der synchronen Groupware PASSENGER .....	142
Abb. 32: Objektmodell der Whiteboard-Dokumente.....	143
Abb. 33: Grafische Repräsentation und zugehöriges Datenformat .....	144
Abb. 34: Benutzerschnittstelle des PASSENGER-Whiteboard.....	145
Abb. 35: Funktionsschaltflächen des Whiteboard .....	146
Abb. 36: Globale und lokale Dokumentenverwaltung.....	147
Abb. 37: PRTP-Protokollkopf .....	151
Abb. 38: Empfangsautomat des PRTP .....	153
Abb. 39: Vergleich der Code-Effizienzen .....	158
Abb. 40: Vergleich der Signalverzögerungen in Abhängigkeit von der Paketlänge ....	159
Abb. 41: Nachrichtenrahmen des PSCP.....	160
Abb. 42: MMTng-Architektur [Wal99] .....	162
Abb. 43: Architektur des PMMT .....	163

Abb. 44: Bewertungskriterien für die Untersuchungen im Rahmen dieser Arbeit.....	168
Abb. 45: Bewertung der Bedienfreundlichkeit.....	173
Abb. 46: Bewertung der Kriteriengruppen Akzeptanz und Wahrnehmung .....	174
Abb. 47: Bewertung der Einzelkriterien (Akzeptanz und Wahrnehmung) .....	175

## **Tabellenverzeichnis**

Tabelle 1: Funktionelle Klassifizierung .....	15
Tabelle 2: Dimensionen des Bewegungs- und Sprachverhaltens [Fre92] .....	20
Tabelle 3: Ergebnisse der JTAP-Studie 2.....	87
Tabelle 4: Funktion der Schaltflächen .....	134
Tabelle 5: Wahrnehmung der Rollen (Momentaufnahme) .....	136
Tabelle 6: Gruppenwahrnehmung über das eigene Kommunikationsfenster .....	137





# 1. Einleitung

Der positive Einfluss des Internet auf die Informationstechnik (IT) und die IT-nahen Wirtschaftszweige der großen Industrienationen ist seit Jahren in beeindruckendem Maße gestiegen. Mit der zunehmenden Zuverlässigkeit und Verfügbarkeit der Netzwerke und Hardware-Komponenten sowie dem stetig wachsenden Dienstangebot erschließt das Netz-der-Netze ständig neue Anwendungsfelder. Eines dieser Anwendungsfelder steht im Mittelpunkt dieser Arbeit: die rechnergestützte Gruppenarbeit im Internet. Die hierbei verwendeten Werkzeuge zur Unterstützung der Gruppenarbeit werden unter dem Begriff Groupware, das zugehörige Arbeits- und Forschungsgebiet unter dem Begriff Computer Supported Cooperative Work (CSCW) subsumiert.

Groupware grenzt sich in Hinblick auf ihre Komplexität sowohl aus Benutzer- als auch aus Entwicklersicht eindeutig von Einbenutzer-Anwendungen ab. Der Entwurf und die Implementation von Groupware sind in der Regel nicht mehr allein mit den klassischen Methoden der Datenverarbeitung und der Informatik zu beherrschen. Vielmehr hat der interdisziplinäre Charakter des Forschungsgebietes CSCW zu zahlreichen Ansätzen aus unterschiedlichen Disziplinen geführt, deren Ergebnisse bei der Entwicklung von Groupware in angemessener Weise berücksichtigt werden müssen.

Im Laufe der vergangenen Jahre haben zahlreiche sogenannte CSCW-Anwendungen ihren Weg auf den Markt gefunden. Die Mehrzahl von ihnen findet sich als integraler Bestandteil in modernen Office-Paketen für Aufgaben wie z.B. die Terminplanung einer Projektgruppe wieder. Die wesentlichen Kommunikationswerkzeuge in diesen Produkten sind E-Mail-Systeme. Werkzeuge, die zur Unterstützung synchroner Kommunikations- und Kooperationsprozesse auch Komponenten zur Audio- und Videokommunikation integrieren, bilden dabei die Ausnahme im Spektrum der kommerziell angebotenen Groupware-Produkte. Es existieren zwar einige, zum Teil sogar ausgereifte Implementierungen für das MBone (Multicast Backbone) oder ISDN-basierte Systeme wie die

Intel\* Team Station; die Kombination von audiovisuellen Kommunikationswerkzeugen und Groupware-Funktionalität findet jedoch noch vorrangig in den Forschungslabors statt.

## 1.1 Motivation

1986 fand in Austin (Texas) die erste CSCW-Konferenz statt. Als ein Arbeitsgebiet, das sich zur Unterstützung durch Groupware anbietet, wurde bereits dort das Arbeitsgebiet Software-Engineering genannt [Gib89]. Diese These wird im Wesentlichen durch die Tatsache gestützt, dass die gemeinsam zu bearbeitenden Daten in der Regel schon in digitaler Form vorliegen und die Hemmschwelle beim Einsatz von Computern und Computerwerkzeugen bei Software-Entwicklern gering ist [Bor98]. In den Tagungsbänden der einschlägigen Konferenzen finden sich dementsprechend auch eine beträchtliche Anzahl von Beiträgen, deren Anwendungskontext im Software-Engineering-Umfeld anzusiedeln ist. Auch in der Praxis ist die Software-Entwicklung, in zum Teil sogar weltweit verteilten Teams, längst zur Realität geworden, vgl. [Haa94], [Gor96], [Cus97]. In einer vom Bundesministerium für Bildung und Forschung durchgeführten Studie "Analyse und Evaluation der Softwareentwicklung in Deutschland" [Pro00] werden dementsprechend "Prozesse, Methoden und Werkzeuge zur Unterstützung räumlich verteilter Produkt- und Softwareentwicklung" als ein zukünftiger Forschungsschwerpunkt vorgeschlagen.

Die Hochschulen reagieren auf das sich ändernde Arbeitsumfeld der zukünftigen Software-Ingenieure nur zögerlich. Hierfür ist einerseits sicherlich der hohe organisatorische und administrative Aufwand bei der Einführung von CSCW in die Software-Engineering-Ausbildung verantwortlich. Andererseits ist wegen des Fehlens geeigneter Werkzeuge die Integration von Groupware in existierende Veranstaltungen problematisch. Dennoch konzentrieren sich aktuelle Arbeiten in erster Linie auf eine Untersuchung der Möglichkeiten zum Einsatz existierender, kommerzieller oder frei verfügbarer Werkzeuge und so gut wie gar nicht auf die zielgerichtete Entwicklung neuer, problemorientierter Werkzeuge. Auf diesem Umstand gründet die wesentliche Motivation

für die hier beschriebenen Arbeiten zur Entwicklung der Groupware PASSENGER und deren Einsatz in der Software-Engineering-Ausbildung.

Darüber hinaus wurden die Arbeiten durch den 1997 mit Förderung durch den Deutschen Akademischen Austauschdienst (DAAD) eingerichteten auslandsorientierten Studiengang "Informations- und Kommunikationstechnik" [Hun98] beeinflusst. Sowohl die Auslandsorientierung als auch die Anwendung innovativer Lehr- und Lernformen waren u.a. programmatische Ziele dieses Experimentalstudiengangs. Die in dem Zusammenhang entstandene Vision eines Software-Engineering-Praktikums mit weltweit verteilten Studententeams und der daraus resultierende Beitrag zur Auslandsorientierung, aber auch zur Zukunfts- und Praxisorientierung der Studierenden waren eine zusätzliche Motivation.

## **1.2 Problemstellung und Ziele**

Die Organisation und Durchführung eines Praktikums mit räumlich verteilten Teilnehmern erfordert Werkzeuge zur Unterstützung in verschiedenen Phasen:

- Im Vorfeld muss die Organisation des Praktikums unterstützt werden. Hierzu zählt z.B., dass die Studierenden sich fristgerecht anmelden können und dass der Betreuer aus diesen Daten die Gruppeneinteilungen und Zeitpläne generieren kann.
- Im Vorfeld der Sitzungen müssen die relevanten Sitzungsinformationen bekannt gemacht werden, bzw. die Teilnehmer müssen explizit zu einer Sitzung eingeladen werden. Zu den relevanten Sitzungsinformationen gehören z.B. Termine und Zeiten der Sitzungen sowie Verbindungsinformationen.
- Während einer Sitzung sind Werkzeuge zur Unterstützung einer oder mehrerer Formen der Zusammenarbeit erforderlich: die synchrone Gruppenarbeit, die asynchrone Gruppenarbeit und die individuelle Arbeit. Diese Werkzeuge müssen zusätzlich den konfliktfreien Zugriff auf die gemeinsamen Ressourcen gewährleisten.

Für den speziellen Fall des hier betrachteten Software-Engineering-Praktikums sind keine komplexen Probleme bezüglich der Organisation im Vorfeld des Praktikums oder

der Einladung zu einer Sitzung zu lösen. Hierzu kann vielmehr auf Standard-Internet-technologien wie HTML (Hypertext Markup Language) und Web-Datenbanken zurückgegriffen werden. Die in diesem Zusammenhang und im Rahmen dieser Arbeit entstandenen Werkzeuge werden am Rande beschrieben, sofern dies für das Verständnis der Groupware PASSENGER nötig ist.

Der Schwerpunkt der Arbeit liegt auf der Entwicklung einer synchronen Groupware zur Unterstützung der Studierenden während einer Sitzung im Rahmen eines Software-Engineering-Praktikums. Ausgangspunkt der Betrachtungen ist dabei ein an der Gerhard-Mercator-Universität Duisburg (GMU) existierendes Praktikum und die dort beobachteten Arbeitsabläufe und Gruppenprozesse.

Der Gestaltung der Benutzeroberfläche und der Unterstützung dedizierter Kommunikations- und Arbeitsabläufe kommt dabei eine deutlich höhere Bedeutung zu, als dies in anderen Arbeiten der Fall ist. Das vorrangige Ziel dieser Arbeit ist die beispielhafte Realisierung einer synchronen Groupware aus problemorientierter Sicht, die aus Benutzersicht eine breite Unterstützung bei der Diskussion und der gemeinsamen Bearbeitung der Entwurfsdokumente bietet.

Das Erreichen der Ziele dieser Arbeit erfordert einerseits ein Verständnis für die Abläufe während des Praktikums und andererseits eine Berücksichtigung der besonderen Anforderungen durch die Studierenden. Die in dem Zusammenhang zu bearbeitenden Probleme betreffen vorrangig die Gestaltung der Benutzerschnittstelle und insbesondere die Entwicklung spezieller Lösungen zur Steigerung der Gruppenwahrnehmung sowie zur Kooperations-, Koordinations- und Kommunikationsunterstützung.

Die beispielhafte Realisierung der Groupware PASSENGER erfolgt für die Windows-Plattform und ist damit konform zu den Zielen der bisher für das MBone entwickelten Systeme, die nun sukzessive portiert werden. Darüber hinaus werden Internettechnologien der nächsten Generation verwendet, um einerseits schon jetzt die damit verbundenen Möglichkeiten nutzen zu können und andererseits die aus einer späteren Portierung resultierenden Probleme bereits im Vorfeld zu vermeiden.

### 1.3 Gliederung der Dissertation

Nachdem in der Einleitung ein Überblick über die Motivation, die Problemstellung und die Ziele der Arbeit gegeben wurde, werden im zweiten Kapitel die grundlegenden Konzepte und Ansätze zur rechnergestützten Gruppenarbeit vorgestellt. Die Darstellung umfasst Begriffsdefinitionen und Klassifikationen und beschreibt die Aspekte der Gruppenkommunikation sowohl aus Sicht der Humankommunikation als auch vor dem Hintergrund einer technischen Realisierung. Das Forschungsgebiet CSCW wird in einem eigenen Abschnitt beschrieben und liefert einen Überblick über die grundlegenden Eigenschaften von CSCW-Systemen sowie eine Darstellung der Konzepte zur Kooperations-, Koordinations- und Kommunikationsunterstützung. Insgesamt bietet Kapitel zwei einen Überblick über die theoretischen Grundlagen, die zum weiteren Verständnis der Arbeit benötigt werden.

Das dritte Kapitel betrachtet vorrangig Aspekte, die für den Entwurf, die Implementation und die Evaluation von Groupware von Bedeutung sind. Dabei erfolgt zunächst eine Diskussion von Vorgehensmodellen und Methoden zum Gruppenstudium. Der Schwerpunkt des Kapitels liegt auf einer Darstellung der verschiedenen Architekturen und Modelle, die für die Implementation von Groupware von Bedeutung sind und schließt mit Aspekten der Evaluation von Groupware.

Im Kapitel vier werden die Arbeitsgebiete Software-Engineering und rechnergestützte Gruppenarbeit zusammengeführt. Dies geschieht zunächst vor dem Hintergrund des Einsatzes virtueller Teams im professionellen Software-Engineering. Im Zentrum des Kapitels steht jedoch der Anwendungskontext der Arbeit; die Einführung von Groupware im Rahmen eines Software-Engineering-Praktikums. Hierzu werden relevante Arbeiten, die sich mit der Einführung von Groupware in der Software-Engineering-Ausbildung befassen, diskutiert und daraus offene Probleme abgeleitet. Am Ende des Kapitels wird ein Anforderungskatalog für eine Groupware zum Einsatz in der Software-Engineering-Ausbildung vorgestellt, der im Rahmen dieser Arbeit entwickelt wurde.

Kapitel fünf beschreibt die eigenen Arbeiten zur beispielhaften Entwicklung einer synchronen Groupware für die Software-Engineering-Ausbildung. Ausgehend von der Ableitung eines Gruppenprozessmodells werden sukzessive die Auswahl einer geeigneten Architektur und der Entwurf der Benutzerschnittstelle beschrieben. Die dabei entwickelten Konzepte zur Kommunikations- und Kooperationsunterstützung werden diskutiert und deren Implementation beschrieben.

In Kapitel sechs erfolgt eine Zusammenfassung der bisherigen Erfahrungen mit dem Einsatz der synchronen Groupware sowie eine Beschreibung einer ersten Untersuchung und der daraus resultierenden Ergebnisse.

In dem abschließenden Kapitel der Arbeit erfolgt die Zusammenfassung der wichtigsten Ergebnisse sowie der weiterhin offenen Probleme und ein Ausblick auf weiterführende, zukünftige Arbeiten.

## 2. Rechnergestützte Gruppenarbeit – Ein Überblick

Die meisten Lebewesen nutzen die Vorteile einer Bündelung der vorhandenen Ressourcen zur Bewältigung einer Aufgabe. Einige setzen hierzu sogar Werkzeuge ein. Weinberg benutzt die Art des Werkzeugeinsatzes zur Abgrenzung der Menschen gegenüber anderen Lebewesen:

*"I believe, however, that humans are the only animals that we know who invent tools for working together - and they have done that as long as we have considered them Human."<sup>1</sup>*

Im Laufe der Evolution haben sich neben den Aufgaben vorrangig die Formen der Zusammenarbeit und die dabei eingesetzten Werkzeuge verändert. Die seit dem Beginn der industriellen Revolution vorherrschende räumliche Konzentration der Arbeitswelt wurde, durch die Einführung der weltweit flächendeckenden Telekommunikationsnetze, zunehmend um Dezentralisierungsaspekte erweitert. In der jüngsten Vergangenheit werden zunehmend die Möglichkeiten des Rechnereinsatzes zur Unterstützung räumlich verteilter Arbeitsgruppen diskutiert.

Dieses Kapitel beschreibt, ausgehend von elementaren Begriffsdefinitionen und Klassifikationen, in einem Überblick die wesentlichen Grundlagen, die zum weiteren Verständnis der Arbeit nötig sind.

---

<sup>1</sup> Vorwort von Gerald M. Weinberg zu [Mar92] zitiert in [Lub95]

## 2.1 Begriffsdefinitionen

Obwohl die im Folgenden verwendeten Definitionen und Klassifikationen weitgehend akzeptiert sind und die einschlägige Literatur dominieren, sei darauf hingewiesen, dass noch keine Standardisierung der Begrifflichkeiten existiert [Bor98].

### Gruppenarbeit

Gruppenarbeit beschreibt die Situation, dass eine einzelne Person mit mindestens einer weiteren Person zusammenarbeitet [Ste99]. Demnach liegt die Untergrenze für die Gruppengröße bei mindestens zwei Personen, eine Obergrenze ist in der Regel nicht definiert. Borghoff und Schlichter ergänzen zudem:

*"Die Gruppenarbeit hat einen Anfangs- sowie einen Endzustand, wobei der Endzustand das Ergebnis der Gruppenarbeit repräsentiert." [Bor98]*

### Gruppenprozess

Die Definition der Gruppenarbeit abstrahiert von einem konkreten Vorgehen bei der Lösung einer Aufgabe. Dieses Vorgehen kann als Prozess aufgefasst werden. Speziell für elektronisch unterstützte Gruppen definieren Borghoff und Schlichter einen Gruppenprozess als

*"die Spezifikation von Informationen, Aktivitäten und Eigenschaften einer elektronisch unterstützten Gruppe bei Festlegung des Rahmens, in dem die Gruppenarbeit stattfindet." [Bor98]*

### Computer Supported Cooperative Working (CSCW)

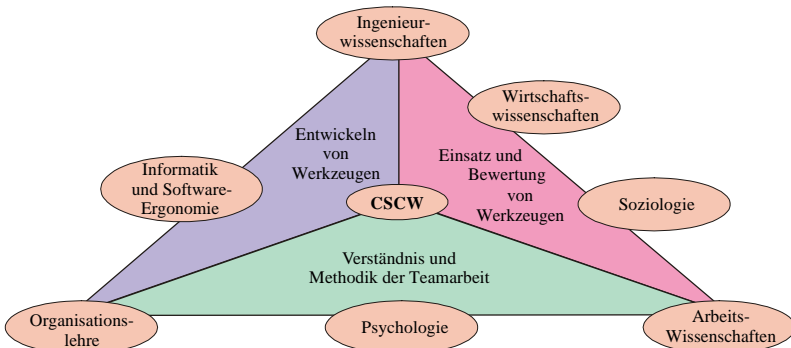
Die Erforschung der zur Realisierung der rechnergestützten Gruppenarbeit benötigten theoretischen Grundlagen ist Gegenstand des Computer Supported Cooperative Working. CSCW ist ein in hohem Maße interdisziplinäres Forschungsgebiet, das nach Ellis et al. bereits 1991 wie folgt definiert wurde:



*"CSCW looks at how groups work and seeks to discover how technology (especially computers) can help them work." [Eli91]*

Obwohl die Definition von Ellis et al. die am häufigsten zitierte Definition von CSCW ist, existieren eine Reihe weiterer Definitionen. Beispiele hierzu finden sich u.a. in [Bor98], [Teu95] und [Whi96].

Aus der Interdisziplinarität des Forschungsgebiets CSCW resultieren unterschiedliche Sichten der verschiedenen Wissenschaftsdisziplinen. Abbildung 1 gibt hierzu einen Überblick.



**Abb. 1:** CSCW-Forschung im Überblick [War98]

Im Rahmen dieser Arbeit sind insbesondere die Sichten der Informatik- und der Ingenieurwissenschaften von Bedeutung. In dem Zusammenhang beschäftigen sich diese Disziplinen mit Methoden, Techniken und Werkzeugen zur Unterstützung der Gruppenarbeit, vgl. [Gey99], [War98].

## Groupware

Der Begriff Groupware hat sich zunehmend für die aus den Forschungsarbeiten resultierenden CSCW-Systeme durchgesetzt. Eine der ältesten Definitionen für Groupware wurde bereits 1988 von Johansen gegeben:

*"Groupware is a generic term for specialized computer aids that are designed for the use of collaborative work groups. Typically, these groups are small project-oriented teams that have important tasks and tight deadlines. Groupware can involve software, hardware, services and/or group process support." [Joh88]*

In neueren Publikationen wird der Begriff Groupware oftmals in Abhängigkeit von den verwendeten Medientypen oder Technologien erweitert. Häufig ist von Real-Time-Groupware die Rede, wenn die Kommunikation Echtzeitbedingungen erfüllen muss [Bat98], [All01]. Dies ist in der Regel der Fall, wenn Audio- und Videokanäle eingesetzt werden. Kovacs spricht in diesem Fall von Multimedia-Groupware [Kov93] und Allison et al. von Real-Time-Multimedia-Groupware [All01].

## 2.2 Klassifikationen

### 2.2.1 Gruppen und Gruppenprozesse

Die einzelnen Mitglieder einer Gruppe werden in der Regel durch eine Beschreibung ihrer Rolle innerhalb der Gruppe hinreichend genau klassifiziert. Dagegen sind zur Klassifikation einer Gruppe deutlich mehr Parameter erforderlich. Diese Parameter und ihre Ausprägungen werden zur Beschreibung des Gruppenprozesses verwendet und sind auch bei der technischen Realisierung einer Gruppenkommunikation von Bedeutung.

Der Gruppenprozess besteht aus statischen und dynamischen Anteilen. In den statischen Anteilen wird die Gruppe und ihre Umgebung in den Parametern Gruppenziele, Gruppenorganisation, Gruppenprotokoll und Gruppenumgebung beschrieben. Der dynami-

sche Anteil charakterisiert den Ablauf der Gruppenarbeit und den aktuellen Zustand in den Parametern Gruppendokumente, Gruppenaktivitäten, Gruppensitzungen und Gruppenzustand. Nachfolgend werden die Parameter im Einzelnen beschrieben:

- **Statische Parameter**

Anhand des Parameters *Gruppenziele* werden die globalen Ziele einer Gruppe beschrieben. Diese können von den individuellen Zielen einzelner Gruppenmitglieder verschieden sein.

Der Parameter *Gruppenorganisation* charakterisiert die Gruppenmitglieder anhand ihrer Stellung innerhalb der Gruppe. Die Definition als statischer Parameter darf in diesem Fall nicht zu eng gefasst werden, da sich die Stellung einzelner Mitglieder durchaus dynamisch ändern kann.

In dem Parameter *Gruppenprotokoll* werden die Art und Weise der Kommunikation innerhalb der Gruppe beschrieben. Face-to-Face-Diskussionen folgen in der Regel sozialen Protokollen. Die systemgestützte Kommunikation mit Hilfe von sozialen Protokollen ist dagegen problembehaftet. Einerseits sichert sie die Kontrollierbarkeit des Kommunikationsablaufs durch die Teilnehmer, andererseits bietet sie im Problemfall keine Systemunterstützung. Technische Protokolle dagegen kontrollieren die Einhaltung definierter Regeln, führen aber unter Umständen zu starken Einschränkungen und Zwängen.

Der Parameter *Gruppenumgebung* beschreibt die räumliche und technische Infrastruktur der einzelnen Gruppenmitglieder.

- **Dynamische Parameter**

Der Parameter *Gruppendokumente* bezeichnet einerseits die Artefakte selbst, andererseits charakterisiert er deren Eigenschaften, die in der Regel dynamisch sind.

Die Eigenschaften des Parameters *Gruppenaktivitäten* beschreiben die Aktivitäten, die in einem Gruppenprozess durchgeführt werden. Innerhalb eines Gruppenprozesses

ses können einzelne Aktivitäten gelöscht, geändert oder neu erzeugt werden. Ebenso können verschiedene Aktivitäten temporal oder kausal voneinander abhängen. Die Gruppenaktivitäten werden im Rahmen von Gruppensitzungen ausgeführt.

Der Parameter *Gruppensitzung* unterscheidet zunächst, ob es sich um eine synchrone oder um eine asynchrone Sitzung handelt. Darüber hinaus definiert er die Operationen, die im Rahmen einer Gruppensitzung möglich sind.

Der Parameter *Gruppenzustand* beschreibt den aktuellen Zustand der Gruppe. Jeder Gruppenprozess hat in der Regel einen Anfangs- und einen Endzustand, wobei der Endzustand im Allgemeinen das Endergebnis der Gruppenarbeit repräsentiert.

Die beschriebenen Parameter zur Klassifikation von Gruppen und zur Beschreibung von Gruppenprozessen sind nicht exklusiv. Zitterbart und Wittmann [Wit99] z.B. beschreiben eine Gruppe durch die Parameter Lebensdauer, Bekanntheit, Offenheit und Heterogenität:

Der Parameter *Lebensdauer* unterscheidet permanente und transiente Gruppen. Permanente Gruppen existieren auch dann noch, wenn aktuell keine Gruppenmitglieder existieren, wohin gegen transiente Gruppen nur existieren, solange es auch aktive Gruppenmitglieder gibt [Wit99].

Der Parameter *Bekanntheit* gibt an, ob es sich um eine anonyme Gruppe oder eine bekannte Gruppe handelt. Im Falle einer bekannten Gruppe sind die Mitglieder der Gruppe zu jeder Zeit bekannt. Im Falle einer anonymen Gruppe ist dies nicht gegeben.

Zur Klassifikation einer Gruppe anhand des Parameters *Offenheit* muss der Informationsfluss in die Betrachtung einbezogen werden. Im Falle einer geschlossenen Gruppe ist der Informationsfluss nur innerhalb einer Gruppe möglich. Die Mitglieder offener Gruppen hingegen können auch Informationen von Nicht-Gruppenmitgliedern erhalten.

Der Parameter *Heterogenität* bezieht die Hard- und Softwareumgebung zur Realisierung des Informationsflusses in die Betrachtung ein. Homogene Gruppen benutzen hinsichtlich der erforderlichen Ressourcen zur Ermöglichung eines effektiven Informationsaustausches äquivalente Hardware- und Software-Systeme. Die Ausstattung heterogener Gruppen hingegen weist in diesem Punkt erhebliche Unterschiede auf.

## Das Team Performance Modell (TPM)

Eine weitere Möglichkeit zur Beschreibung von Gruppenprozessen stellt das Team Performance Modell (TPM) nach Drexler und Sibbet dar, vgl. [Bor98]. Das TPM wird beschrieben durch die Entstehungsphasen der Orientierung, der Vertrauensbildung, der Ziel- und Aufgabendefinition und der Verpflichtung sowie durch die Festigungsphasen der Implementierung, der hohen Leistung und der Erneuerung. Die Phasenübergänge erfolgen, sofern alle Probleme innerhalb einer Phase gelöst sind. Rückschritte, in Folge ungelöster Probleme, erfolgen für die Entstehungsphasen in die vorherige Phase und für Festigungsphasen in eine Entstehungsphase auf gleicher Ebene.

### 2.2.2 Klassifikation von CSCW-Systemen

Für die Klassifikation von CSCW-Systemen existieren verschiedene Schemata nach teilweise unterschiedlichen Kriterien. Diese resultieren aus verschiedenen Sichten auf die Systeme. Im Folgenden werden häufig in der Literatur zitierte Klassifikationsschemata kurz vorgestellt.

#### Klassifikation nach der Raum-Zeit-Matrix

Aus arbeitsorganisatorischer Sicht erweitert rechnergestützte Gruppenarbeit traditionelle Arbeitsformen um Möglichkeiten der räumlichen und zeitlichen Verteilung. Die verschiedenen Optionen bei der Kombination von Raum und Zeit sowie die daraus resultie-

renden Bezeichnungen für die verschiedenen Formen der Gruppenarbeit werden in Abbildung 2 dargestellt.

Raum \ Zeit	gleiche Zeit	verschiedene Zeit	
		vorhersehbar	nicht vorhersehbar
gleicher Raum	synchrone Gruppenarbeit	asynchrone Gruppenarbeit	
verschiedener Raum vorhersehbar	räumlich verteilte synchrone Gruppenarbeit	räumlich verteilte asynchrone Gruppenarbeit	
verschiedener Raum nicht vorhersehbar			

**Abb. 2:** Raum-Zeit-Matrix der Gruppenarbeit

Die dargestellte 3x3-Raum-Zeit-Matrix geht auf die ursprünglich von Johansen [Joh88] vorgestellte 2x2-Raum-Zeit-Matrix zurück. Sie wurde von Grudin [Gru94] um Beispiele für eine technische Unterstützung der jeweiligen Arbeitsform erweitert, vgl. Abbildung 3.

Raum \ Zeit	gleiche Zeit	verschiedene Zeit	
		vorhersehbar	nicht vorhersehbar
gleicher Raum	Face-to-Face-Sitzung	Schichtarbeit	„schwarzes Brett“
verschiedener Raum vorhersehbar	Video-konferenz	E-Mail	kollaboratives Verfassen von Dokumenten
verschiedener Raum nicht vorhersehbar	Mobilfunk-konferenz	Nicht-Echtzeit-Rechner-konferenz	Vorgangsbearbeitung

**Abb. 3:** Groupware Klassifikation nach Raum und Zeit

Die in Abbildung 3 dargestellten Einzelkomponenten können für sich genommen als Groupware aufgefasst werden. Ein umfassendes System sollte jedoch mehrere dieser Einzelkomponenten kombinieren.

## Funktionelle Klassifikation

Ellis et al. [Eli91] stellten 1991 eine funktionale Klassifikation vor, deren Vorteil in der Anwendungsorientierung liegt [Bor98]. Sie unterscheiden Nachrichtensysteme, Gruppendeditoren, elektronische Sitzungsräume, Konferenzsysteme, gemeinsame Informationsräume, Agentensysteme und Koordinationssysteme, siehe Tabelle 1.

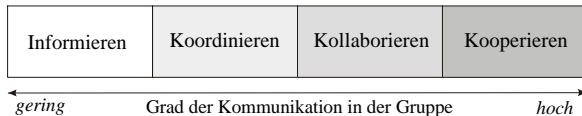
Kategorie	Ausprägung und Beschreibung
<b>Nachrichtensysteme:</b>	Asynchroner Nachrichtenaustausch zwischen Gruppenmitgliedern.
<b>Gruppendeditoren:</b>	Gruppendeditoren sind Mehrbenutzereditoren zur gemeinsamen Erstellung und Bearbeitung von Dokumenten. <u>Realzeiteditoren</u> erlauben das Editieren durch mehrere Benutzer zur selben Zeit. <u>Asynchrone Editoren</u> erlauben nur das zeitversetzte Arbeiten an den Objekten.
<b>Elektronische Sitzungsräume:</b>	Sitzungsräume mit spezieller Recherausstattung für Face-to-Face-Sitzungen, z.B. zur Unterstützung von Gruppenentscheidungen.
<b>Konferenzsysteme:</b>	<u>Nicht Realzeitkonferenz:</u> Asynchrones Konferieren, z.B. per E-Mail, der Gruppenmitglieder über ihre Arbeitsplatzrechner. <u>Realzeitrechnerkonferenz:</u> Synchroner Rechnerkonferenz ohne direkte Audio- oder Videoverbindung. Vorrangig beschränkt auf den Datenaustausch und die Bearbeitung elektronischer Informationen. <u>Telekonferenz:</u> Unterstützung der Interaktionen einer Gruppe durch Telekommunikationsmittel (Audio- und/oder Videoverbindung). Zwischen den Konferenzräumen besteht eine Audio-, möglicherweise auch eine Videoverbindung. Die gemeinsame Bearbeitung elektronischer Informationen ist nicht möglich. <u>Desktopkonferenz:</u> Kombiniert unter Verwendung von Arbeitsplatzrechnern und Multi-Medianetzen die Eigenschaften von Realzeitrechnerkonferenz und Telekonferenz. Die Videobilder werden in ein Bildschirmfenster integriert.
<b>Gemeinsame Informationsräume:</b>	Die systemorganisierte Verwaltung der Gruppendokumente steht im Vordergrund. Erfolgt auch die Kommunikation ausschließlich über die gemeinsamen Gruppendokumente, wird von impliziter Kommunikation gesprochen.
<b>Agentensysteme:</b>	Ein Programm, das als aktiver Sitzungsteilnehmer auftritt, um z.B. ein Sitzungsprotokoll aufzuzeichnen oder zu überwachen.
<b>Koordinationsysteme:</b>	Zur Unterstützung der Koordination der Tätigkeiten einzelner Gruppenmitglieder. <u>Formularorientierte Systeme</u> modellieren den Datenfluss innerhalb einer Organisation, z.B. beim Umlauf eines Dokuments. <u>Prozedurorientierte Systeme</u> modellieren die Funktionen und Abläufe innerhalb einer Organisation, wie z.B. den Software-Entwicklungsprozess. In <u>konversationsorientierten Systemen</u> basiert die Kooperation auf dem Austausch sprachlicher Äußerungen. Modelliert werden dabei die Interaktionen zwischen den Gruppenmitgliedern sowie die daraus resultierenden Aktionen. <u>Kommunikationsorientierte Systeme</u> modellieren komplexe Kommunikationsstrukturen innerhalb einer Organisation. Diese Kommunikationsstrukturen drücken einerseits die Organisationsstruktur aus, andererseits beinhalten sie die verschiedenen Rollen, die innerhalb der Organisation auftreten.

**Tabelle 1:** Funktionelle Klassifizierung

Ein umfassendes CSCW-System sollte Funktionalitäten verschiedener Kategorien nach Tabelle 1 integrieren.

## Klassifikation nach dem 3K-Modell

Der Kommunikationswissenschaftler J.H. Bair hat 1989 ein Stufen-Modell für die zunehmende Intensität der Kommunikation zwischen den Mitgliedern einer Gruppe vorgestellt [Bai89], vgl. Abbildung 4.

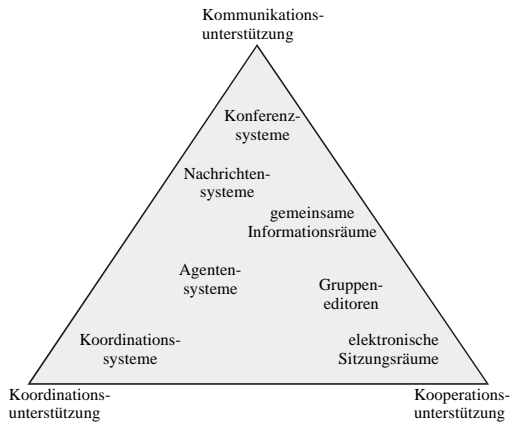


**Abb. 4:** Stufen Modell für die zunehmende Intensität der Kommunikation [Bor98]

Nach Bair ist das Informieren im Wesentlichen geprägt durch den anonymen Austausch von Informationen. Kennzeichnend für den Übergang vom reinen Informieren zum Koordinieren ist dabei im Wesentlichen, dass die Beteiligten nun als loser Verbund auftreten. Eine möglichst effiziente gemeinsame Ressourcennutzung wäre z.B. ein Ziel der Koordination. Ein Gruppenziel im Sinne einer gemeinsam zu lösenden Aufgabe prägt das Kollaborieren, wobei jedoch die persönlichen Ziele höher bewertet werden als das Gruppenziel. Als Konsequenz daraus sind unterschiedliche Bewertungen der erreichten Ergebnisse durch die Beteiligten möglich. Der höchste Grad der Kommunikation ist durch das Kooperieren gegeben. Hier werden die persönlichen Ziele gegenüber den Zielen der Gruppe zweitrangig. Entscheidungen fallen aufgrund eines Gruppenkonsens und das Ergebnis wird von allen Gruppenmitgliedern einheitlich bewertet. Mit zunehmender Komplexität steigen auch die Möglichkeiten einer Rechnerunterstützung der Kommunikation [Bor98].

Das 3K-Modell nach Teufel et al. [Teu95] differenziert, aufbauend auf den Definitionen nach Bair, Kommunikationsunterstützung, Koordinationsunterstützung und Kooperationsunterstützung. Abbildung 5 zeigt eine Anordnung der anwendungsorientierten Funktionsklassen im 3K-Modell. Nach Borghoff und Schlichter [Bor98] sind die meisten CSCW-Systeme in irgendeiner Form "K-orientiert".





**Abb. 5:** Klassifikation nach dem 3K-Modell [Bor98]

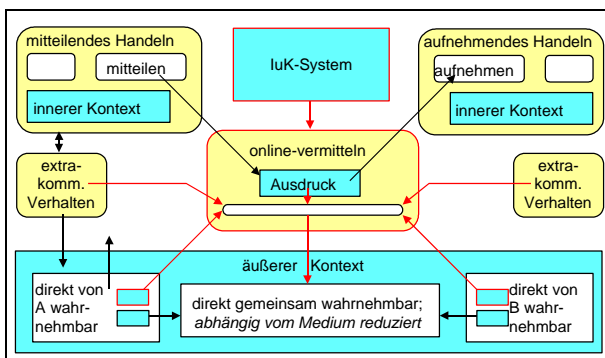
## 2.3 Humankommunikation

Die Entwicklung von Kommunikationsmodellen zur Beschreibung des Austauschs von Informationen zwischen mindestens zwei Gesprächspartnern ist Gegenstand der Kommunikationsforschung. Eines der ersten Kommunikationsmodelle wurde 1949 von Shannon und Weaver vorgestellt [Sha49]. Dieses Modell besteht aus den Grundelementen Sender, Medium und Empfänger und beschreibt die Kommunikation aus nachrichtentechnischer Sicht .

Obwohl später gezeigt werden konnte, dass sich in der Humankommunikation für alle Modellkomponenten des technischen Systems Entsprechungen finden lassen [Fre92], fehlt dem Modell dennoch die entscheidende Möglichkeit, den Kontext der Kommunikation zu erfassen. Gerade fehlende Kontextinformationen haben sich in Untersuchungen jedoch als häufigste Ursache für das Auftreten von Kommunikationsmissverständnissen herausgestellt [Her99]. Frey [Fre96] geht so weit, die Anwendbarkeit des nachrichtentechnischen Kommunikationsmodells auf die Humankommunikation völlig in Frage zu stellen. Er begründet dies damit, dass im

nachrichtentechnischen Kommunikationsmodell der Sender sowohl die Erzeugung eines Signals als auch dessen Interpretation durch den Empfänger kontrolliert. In der Humankommunikation dagegen entscheidet der Empfänger durch seine Interpretation über die Bedeutung der Signale. Das Kommunikationsmodell von Frese und Brodbeck [Fre89] stellt durch die Einbeziehung von Beziehungs-, Handlungs- und Konventionskontext eine deutliche Verbesserung hinsichtlich der Möglichkeiten zur Modellierung der Humankommunikation dar.

Die Modellierung von Online-Kommunikationen stellt zusätzliche Anforderungen an die Kommunikationsmodelle, da die unmittelbare Situation, in die die Kommunikationspartner eingebunden sind, weniger stark wahrnehmbar ist, als dies bei einer Face-to-Face-Kommunikation der Fall ist. Herrmann schlägt in [Her01] ein Kommunikationsmodell zur Beschreibung einer Online-Kommunikation vor, das sowohl den inneren Kontext, den äußeren Kontext und das extra-kommunikative Verhalten der einzelnen Kommunikationspartner sowie den gemeinsam wahrnehmbaren äußeren Kontext berücksichtigt, vgl. Abbildung 6.



**Abb. 6:** Kommunikationsmodell einer Online-Kommunikation [Her01]

Das in Abbildung 6 dargestellte Kommunikationsmodell basiert auf einem Mitteilungskonzept, das berücksichtigt, dass das Entwickeln von Vorstellungen und das

Mitteilen derselben zwar getrennte, aber dennoch eng miteinander verknüpfte und parallel ablaufende Prozesse sind.

Die Mitteilung stellt eine Verbalisierung der Vorstellungen dar, die der Mitteilende aus einer Vielzahl von Möglichkeiten, den sogenannten Paraphrasen, auswählt. Bei dieser Auswahl berücksichtigt der Mitteilende in der Regel das von dem Empfänger gewonnene Partnerbild. Dieses Partnerbild beruht auf einer Einschätzung des möglichen Vorwissens des Empfängers durch den Mitteilenden. Herrmann bezeichnet dies als das Bild des Mitteilenden vom inneren Kontext des Empfängers [Her01]. Das Partnerbild hat wesentlichen Einfluss auf den Inhalt der Mitteilung. Eine besondere Rolle spielt auch die Beziehung, in der die Kommunikationspartner zueinander stehen. Dieser Beziehungsaspekt beeinflusst entscheidend, welche zusätzlichen Informationen der Mitteilende über sich selbst oder seine soziale Stellung in die Mitteilung einbezieht.

Weiterhin modelliert Herrmann in seinem Modell der Online-Kommunikation extra-kommunikatives Verhalten, welches ursächlich nicht kommunikativ gemeint ist und damit auch nicht als Mitteilung konzipiert ist [Mis99]. Extra-kommunikatives Verhalten transportiert zusätzliche Informationen wie z.B. die Situation, dass einer der Kommunikationspartner den Raum verlässt. In einer Groupware wird extra-kommunikatives Verhalten durch so genannte Awareness-Mechanismen bzw. Maßnahmen zur Gruppenwahrnehmung vermittelt. Diese Mechanismen und Maßnahmen werden in Abschnitt 2.5.2 beschrieben.

Eine weitere wichtige Kommunikationseigenschaft wird durch die non-verbale Kommunikation beschrieben. Diese wird in dem Modell nach Herrmann im gemeinsam wahrnehmbaren äußeren Kontext modelliert. Die nonverbale Kommunikation benutzt als Ausdrucksmittel die Gestik, die Mimik und die Körperhaltung. Beides übt nach Sapir [Sap49] eine enorme Suggestivkraft auf den menschlichen Betrachter aus und *„es scheint so, als reagiere der Mensch auf Gestik mit einer enormen Wachsamkeit, in Übereinstimmung mit einem sorgfältig ausgearbeiteten und geheimen Code, der niemals niedergeschrieben wurde, den niemand kennt, aber alle verstehen.“* [Sap49]

Ergebnisse einer Untersuchung von Siegfried Frey [Fre92] zeigen, dass die menschliche Informationsverarbeitung auf dem sachlichen Inhalt einer Nachricht und der affektiven Stellungnahme zur dargebotenen Stimuluskonfiguration aufbaut. Dabei kann sich in der Größenordnung von Millisekunden entscheiden, ob der Gesprächspartner als sympathisch, autoritär, intelligent, gefühlsbetont, hinterhältig oder langweilig angesehen wird. Die durch visuelle Stimuli ausgelösten Persönlichkeitseindrücke sind dabei lang anhaltend und keineswegs nur von flüchtiger Dauer. Eine Korrektur durch kognitive Einflüsse ist sehr schwierig [Fre92].

Siegfried Frey gelang bereits Anfang der 80er Jahre ein Vergleich der Kodierungsprinzipien des Bewegungsverhaltens und der Sprachaussagen. Die in Tabelle 2 dargestellten Ergebnisse belegen eindrucksvoll die Vielzahl der Ausdrucksmöglichkeiten der nonverbalen Kommunikation,

*"die in Bezug auf Detailreichtum und Zuverlässigkeit der Beschreibung der alphabetischen Sprachnotation um nichts nachsteht."* [Fre92]

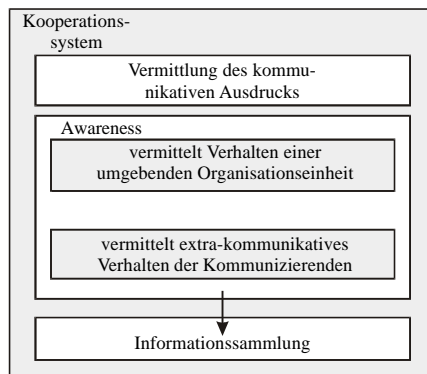
Visueller Kanal		Auditiver Kanal	
Anzahl Dimensionen	Subsystem	Subsystem	Anzahl Dimensionen
3	Kopf	Phoneme	1
49	Gesicht	Lautstärke	1
4	Schultern	Tonhöhe	1
3	Rumpf	Klangfarbe	1
6	Oberarme		
18	Hände		
2	Becken		
5	Beine		
14	Füße		

**Tabelle 2:** Dimensionen des Bewegungs- und Sprachverhaltens [Fre92]

In Hinblick auf eine Online-Kommunikation geben die Untersuchungsergebnisse nach Tabelle 2 außerdem einen klaren Hinweis auf die Bedeutung des visuellen Kanals. Eine Differenzierung des Begriffs der Online-Kommunikation scheint daher angebracht. Im Folgenden wird nun allgemein von Multimedia-Kommunikation gesprochen, wenn zeitkontinuierliche Medien während der Online-Kommunikation eingesetzt werden. Die Erweiterung des Kommunikationskanals durch zeitkontinuierliche Medien hat den

Vorteil, dass die Vermittlung schwer zu fassender Informationen besser möglich ist, da insbesondere das visuelle Aufnahmesystem des Menschen sehr leistungsfähig ist. Zudem entspricht die Informationsaufnahme mit Hilfe mehrerer Sinne eher der natürlichen Kommunikationsweise des Menschen.

Die Unterstützung von Kooperation durch Koordination ist eine der wesentlichen Aufgaben von Kommunikation [Her01]. Wenn es sich um eine Online-Kommunikation handelt, muss das Kommunikationsmodell aus Abbildung 6 zusätzliche Dienste zur Kooperationsunterstützung bereit stellen. Herrmann und Misch schlagen für Kooperationsysteme die in Abbildung 7 dargestellte allgemeine Struktur vor [Mis99].



**Abb. 7:** Komponenten eines Kooperationsystems nach [Mis99]

In Abbildung 8 ist die Kombination des Kommunikationsmodells aus Abbildung 6 und des Kooperationsystems nach Abbildung 7 dargestellt.

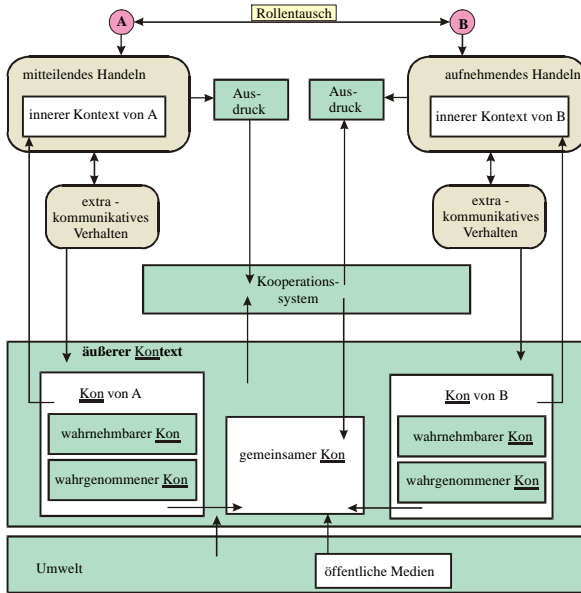


Abb. 8: Einsatz eines Kooperationsystems [Mis99]

## 2.4 Gruppenkommunikation im IPv6-Internet

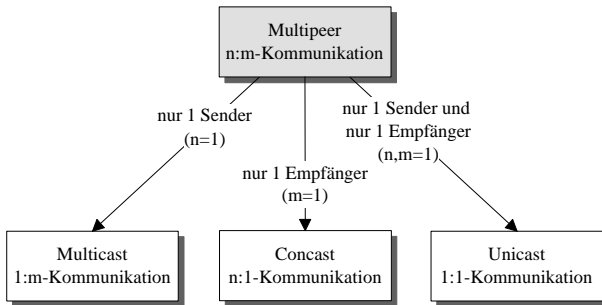
Moderne Multimedia-Applikationen stellen sowohl quantitative als auch qualitative Anforderungen an das verwendete Transportsystem. Die quantitativen Anforderungen, wie z.B. geringe Verzögerung, kleiner Jitter und hoher Datendurchsatz, werden dabei in der Regel durch den multimedialen Anteil der Anwendung bestimmt, wohingegen die qualitativen Anforderungen wie z.B. die technische Realisierung der Gruppenkommunikation sowie Daten- und Zugriffssicherheit durch den Groupware-Charakter der Anwendung definiert werden. In den nächsten Abschnitten werden Konzepte sowie ausgewählte Protokolle und Dienste zur technischen Realisierung einer Gruppenkommunikation im IPv6-Internet vorgestellt.

## 2.4.1 Technische Aspekte der Gruppenkommunikation

Die bisherigen Darstellungen der Gruppenkommunikation abstrahieren weitgehend von technischen Aspekten. Die technischen Aspekte werden in Basistechnologien, die grundsätzlich den Austausch digitaler Daten ermöglichen, sowie spezielle Technologien, welche die Basis für eine Gruppenkommunikation bieten, unterschieden.

Die Basistechnologien sind im Wesentlichen geprägt durch die stetig fortschreitende Digitalisierung aller Kommunikationsbelange in den vergangenen Jahrzehnten [Con96] und die daraus resultierenden Technologien und Systeme zur Datenkommunikation. Die Situation, dass genau ein Sender und genau ein Empfänger am Datenaustausch beteiligt sind, wird als Unicast- oder 1:1-Kommunikation bezeichnet und stellt die klassische Kommunikationsform dar. Unicast-Verbindungen sind unidirektional vom Sender zum Empfänger gerichtet und ermöglichen über den Rückkanal vom Empfänger zum Sender lediglich den Versand von Kontrolldaten. Zur Realisierung eines bidirektionalen Datenaustauschs zwischen den beiden Kommunikationspartnern sind daher zwei Unicast-Verbindungen nötig. Im Hinblick auf eine Gruppenkommunikation repräsentiert die Unicast-Kommunikation gewissermaßen den Sonderfall, dass die Sender- und Empfängergruppen auf jeweils eine Instanz reduziert sind.

Eine Gruppenkommunikation im Vergleich zur Unicast-Kommunikation ist dadurch gekennzeichnet, dass die Dateneinheiten innerhalb einer Gruppe von Sendern und Empfängern ausgetauscht werden. Die dabei insgesamt möglichen Kommunikationsformen und deren Beziehungen zueinander zeigt Abbildung 9.



**Abb. 9:** Beziehung der Kommunikationsformen nach [Wit99]

Die allgemeinste Form der Kommunikation ist die Multipeer-Kommunikation, bei der ein bidirektionaler Datenaustausch zwischen einer Gruppe von Sendern und einer Gruppe von Empfängern realisiert ist. Dabei können Sender und Empfänger während der Kommunikation ihre Rollen wechseln. Existiert genau ein Sender und ist die Gruppe von Empfängern in ihrer Zusammensetzung begrenzt und/oder bekannt, so handelt es sich um eine Multicast-Kommunikation. Der Datenaustausch erfolgt dabei unidirektional vom Sender zu den Empfängern.

Der Fall, bei dem der Datenaustausch unidirektional zwischen mehreren Sendern und genau einem Empfänger stattfindet, wird als Concast- oder m:1-Kommunikation bezeichnet.

## 2.4.2 Multicast-Kommunikation

Eine Multicast-Kommunikation kann grundsätzlich durch den Aufbau entsprechend vieler Unicast-Verbindungen emuliert werden [Wit99]. Ebenso lässt sich eine Multipeer-Kommunikation durch den Aufbau je einer Multicast-Kommunikation von jedem Sender zu den übrigen Empfängern emulieren.

Bei der Emulation einer Multicast-Kommunikation durch Unicast-Verbindungen werden die Daten mehrfach versendet. Im Falle des World-Wide-Web, in dem der Daten-



austausch auf dem IP-Protokoll basiert, bedeutet dies das mehrfache Versenden gleicher IP-Datagramme. Auf ihrem Weg vom Sender zu den Empfängern durchlaufen diese Datagramme dann mehrere Router, in denen mindestens die Adressinformationen des IP-Protokolls ausgewertet werden. Insgesamt werden die zu übertragenden Dateneinheiten:

- vom Sender mehrfach dupliziert und zu dem ersten Zwischensystem gesendet (die Anzahl der Duplikate ist dabei bestimmt durch die Anzahl der Empfänger);
- mehrfach in den Zwischensystemen kopiert und bearbeitet (die Anzahl der Duplikate ist dabei bestimmt durch die Anzahl der über das Subnetz erreichbaren Empfänger);
- mehrfach von den Zwischensystemen weitergeleitet (die Anzahl der Duplikate ist dabei bestimmt durch die Anzahl der über das Subnetz erreichbaren Empfänger).

Bei der Realisierung einer Multicast-Kommunikation durch eine Multicast-Verbindung werden die Dateneinheiten, die auf dem Weg vom Sender zu der Empfängergruppe über gleiche Verbindungsstrecken weitergeleitet werden nur einmal übertragen. Dadurch reduziert sich der Verarbeitungsaufwand beim Sender und in den Zwischensystemen erheblich und die Netzlast wird reduziert. Jedoch erfordert eine Multicast-Verbindung eigene Multicast-Protokolle in den Endsystemen, bei den Sendern und Empfängern sowie in den Zwischensystemen [Wit99].

### 2.4.3 Das Internet Protokoll Version 6

Das Internet ist eine *"Ansammlung von physischen Netzen, die über Zwischensysteme verbunden werden."* [Com98] Zu den bekanntesten Diensten im Internet gehören u.a. das World-Wide-Web (WWW) und die Electronic Mail (E-Mail). Der Datenaustausch zwischen den Endsystemen wird über das Internet-Protokoll (IP) organisiert. IP ist ein Protokoll der Netzwerkschicht (Schicht 3 des ISO/OSI-Referenzmodells), das der darüber liegenden Schicht einen verbindungslosen und unzuverlässigen Dienst anbietet

und eine Betriebssystem- und Netzwerk-unabhängige Schnittstelle für die Rechner-zu-Rechner-Kommunikation darstellt.

Die einzelnen Systeme im Internet werden mit einer Internet-Protocol-Adresse (IP-Adresse) adressiert. Mit Hilfe dieser IP-Adresse werden die Dateneinheiten (Datagramme) im Internet weitergeleitet (Routing). Die aktuelle IP-Version 4 verwendet hierzu 32-Bit breite Adressen. Durch das zunehmende kommerzielle Interesse am Internet gehen Schätzungen davon aus, dass zwischen den Jahren 2005 und 2010 der bisher verfügbare Adressraum von IPv4 nicht mehr ausreicht [Brau99]. Die Internet Engineering Task Force (IETF) hat daher frühzeitig entschieden, eine neue Version des Internet-Protokolls mit einem deutlich größeren Adressraum zu entwickeln. Es entstand die neue Protokollversion IPv6, die neben einem deutlich größeren Adressraum mit nun 128-Bit breiten IP-Adressen eine Reihe weiterer Neuerungen bietet. Diese lassen sich nach Comer [Com98] in die fünf folgenden Hauptkategorien gliedern:

- **Adressgröße**

Im Zuge der Verwendung von 128-Bit breiten IPv6-Adressen schlägt die IETF auch eine neue Syntax zur Reduzierung der Adresskomplexität vor. Nach der Doppelpunkt-Hexadezimal-Notation werden die 128-Bit breiten IPv6-Adressen in acht Blöcke zu je 16-Bit zerlegt. Die Darstellung der einzelnen Blöcke erfolgt hexadezimal und die Trennung der einzelnen Blöcke durch Doppelpunkte.

- **Header-Format**

Das Header-Format wurde grundlegend geändert, so dass eine direkte Abbildung der Felder des IPv4-Protokoll-Headers auf Felder des IPv6-Protokoll-Headers nur bedingt möglich ist<sup>2</sup>. Das Feld Hop Limit ersetzt z.B. das Feld *TTL* (*time-to-live*) beim IPv4-Header. Es legt die maximale Anzahl von Zwischensystemen fest, die das Paket durchlaufen darf.

---

<sup>2</sup> Für eine Beschreibung der einzelnen Felder des IPv6-Headers siehe [Dee98].

- **Mehrere Header**

Das IPv6-Format sieht neben einem Basisheader die Möglichkeit zur Verwendung mehrerer optionaler Header vor. Dadurch sollen die Verarbeitungskosten in den Zwischensystemen bei IP-Rahmen ohne optionale Header reduziert werden.

- **Echtzeitunterstützung**

IPv6 unterstützt Echtzeitdatenübertragungen als Grundlage für effiziente Video- und Audiokommunikation. Hierzu gehört die Möglichkeit zur Definition von Prioritäten und die Zuweisung zu Verkehrsklassen für bestimmte Dateneinheiten. Darüber hinaus besteht die Möglichkeit, Pakete, die zu einem bestimmten Datenfluss gehören, durch eine Flussmarke (flow label) zu kennzeichnen. Den Datenflüssen können dann, zur schnelleren Verarbeitung in den Zwischensystemen, bestimmte Dienstgüten in Verbindung mit einer Priorität zugewiesen werden.

- **Erweiterbares Protokoll**

Bei der Entwicklung des IPv6-Protokolls wurde Spielraum für zukünftige Implementationen und Erweiterungen gelassen.

Anhand des höchstwertigen Adressbyte werden Unicast- und Multicast-Adressen unterschieden. Unicast-Adressen werden durch das Adresspräfix „0“ gekennzeichnet. Das Adresspräfix einer Multicast-Adresse besteht dagegen aus acht Einsen.

## Multicast-Adressen

Multicast-Adressen sind von besonderer Bedeutung für die Gruppenkommunikation. Die Identifikation einer IPv6-Adresse als Multicast-Adresse erfolgt wiederum anhand ihres Präfixes, das in diesem Fall aus acht Einsen besteht. An das Präfix schließen sich die Felder Flags, Scope und Gruppenkennung an, vgl. Abbildung 10.

Präfix	Flags	Scope	Gruppenkennung
8-Bit: 1111 1111	4-Bit	4-Bit	112-Bit
? - - - - - 128-Bit - - - - - ?			

**Abb. 10:** Aufbau der IPv6-Multicast-Adressen

Über das Feld Flags werden transiente und permanente Gruppen unterschieden. Die Reichweite bzw. der Gültigkeitsbereich der Gruppe wird durch das Feld Scope festgelegt. Dabei können die Gültigkeitsbereiche Link, Site oder Global definiert werden. Die letzten bzw. die niederwertigen Bit der Adresse beschreiben die Gruppennummer bzw. -kennung. Grundsätzlich ist die Generierung einer eindeutigen Multicast-Adresse mit Gruppenkennung nicht problemlos.

## Unicast-Präfix-Multicast-Adressen

Einen vielversprechenden alternativen Ansatz zur Generierung einer eindeutigen Multicast-Adresse stellt das Unicast-Präfix-Multicast-Adressformat dar. Der Grundgedanke dieses von Habermann und Thaler [Hab01] vorgeschlagenen Verfahrens basiert auf der Tatsache, dass jedes System eine eindeutige Unicast-Adresse im jeweiligen Gültigkeitsbereich besitzt:

- Link-Lokale-Unicast-Adressen (LLU)

LLU-Adressen sind unstrukturierte Adressen, die keine Identifikation von Subnetzen enthalten und daher nur innerhalb von isolierten IPv6-Netzwerken verwendet werden können. LLU-Adressen werden daher von Routern nicht weitergeleitet.

- Site-Lokale-Unicast-Adressen (SLU)

SLU-Adressen sind strukturierte Adressen, die eine Subnetz-ID enthalten. Daher können sie innerhalb eines Standortes, bzw. einer Gruppe von IPv6-Subnetzen, die durch die gleiche Subnetz-ID gekennzeichnet sind, verwendet werden. Da SLU-Adressen nur innerhalb eines Standortes gültig sind, dürfen sie nicht von Routern nach außen weitergeleitet werden.

- Global-Unicast-Adressen (GU)

GU-Adressen sind öffentliche Adressen, vergleichbar mit den IPv4 Adressen der Klassen A, B und C. In ihrer allgemeinen Form setzen sie sich aus einem 64-Bit breiten Netzwerk-Präfix in den höherwertigen Bit und einer 64-Bit breiten Interface-ID in den niederwertigen Bit zusammen.

Zur Bildung einer Unicast-Präfix-Multicast-Adresse werden zunächst die höchstwertigen 64-Bit der Unicast-Adresse des Senders - der so genannte Netzwerk-Präfix - in die Gruppenkennung integriert. Die niederwertigen Bit der Gruppenkennung können vom Zeroconf Multicast Address Allocation Protocol (ZMAAP) relativ einfach gesetzt werden. Auf diese Weise garantiert die Gruppenkennung in Verbindung mit der Scope-Angabe die Eindeutigkeit im Gültigkeitsbereich. In korrekt konfigurierten Netzen sind dadurch Kollisionen im jeweiligen Gültigkeitsbereich ausgeschlossen.

## 2.4.4 Das Internet Control Message Protocol

Das Internet Control Message Protocol (ICMP) [Pos81] ist ein Kontrollprotokoll zur Unterstützung der Ende-zu-Ende-Kommunikation von Zwischen- und Endsystemen. Dabei werden die ICMP-Dateneinheiten in IP-Datagramme gekapselt. Ab der Version 6 des ICMP (ICMPv6) [Con98] übernimmt dieses auch Aufgaben zur Gruppenverwaltung, die bisher über das Internet Group Management Protocol (IGMP) organisiert wurden.

### Das Internet Group Management Protocol (IGMP)

Das IGMP [Dee89] wurde ursprünglich zur Multicast-Unterstützung im IPv4-Internet implementiert. Zu den Aufgaben von IGMP gehört z.B. die Verwaltung der Gruppenzugehörigkeit (Beitreten oder Verlassen einer Gruppe). Die zur Zeit eingesetzte Version ist die Implementation der Version IGMPv2 [Fen97].

IGMP-Rahmen werden in IP-Rahmen gekapselt und setzen den Wert des TTL-Feldes auf einen Wert von Eins. Dadurch erreicht das IP-Datagramm nur den lokalen IGMP-

Router und verlässt nicht das Subnetz. Der IGMP-Router organisiert die Weiterleitung der Gruppenzugehörigkeit über spezielle Multicast-Routingprotokolle. Zum Beitreten und Verlassen einer Gruppe werden von IGMP zwei Aktionen unterschieden: Anfragen (Queries) und Antworten (Reports).

Anhand der Anfragen werden Informationen über alle Gruppen mit Mitgliedern im angeschlossenen Teilnetz (General Membership Queries) bzw. mit Mitgliedern einer speziellen Gruppe im angeschlossenen Teilnetz (Group specific Membership Queries) ausgetauscht.

Über so genannte Membership-Reports erhalten die Router Informationen über neue Gruppenmitgliedschaften.

Der Austausch von IGMP-Nachrichten erfolgt periodisch sowohl zwischen Endsystemen und Routern als auch zwischen Routern. Beim Beitreten zu einer Gruppe werden von den Multicast-Routern Query-Pakete versendet. Dadurch wird zunächst die Gruppe, der das jeweilige Endgerät angehört, ermittelt. Das Verlassen einer Gruppe kann in der Version IGMPv2 nur durch das Ausbleiben des Report-Pakets erkannt werden. Bei der Weiterentwicklung von IGMP zu IGMPv3 kann das Verlassen einer Gruppe aktiv dem Router mitgeteilt werden und dadurch die Belastung des Subnetzes verringert werden.

## 2.4.5 Migration von IPv4 zu IPv6

Der Wechsel von der Version 4 des Internetprotokolls auf die neue Version 6 erfordert insbesondere für die Übergangszeit geeignete Konzepte und Mechanismen. Die Länge der Übergangszeit ist momentan schwer vorhersehbar. Neu zu installierende Systeme werden daher vollständige Implementationen beider Protokoll-Stacks, so genannte Dualstacks, besitzen. Grundsätzlich lässt sich der Übergangszeitraum nochmals in zwei Abschnitte zerlegen. Der erste Abschnitt erstreckt sich bis zu dem Zeitpunkt, an dem der gesamte IPv4-Adressraum erschöpft ist. Bei geeigneter Wahl der IPv6-Adressen ist es möglich, dass bis zu diesem Zeitpunkt alle IPv6-Adressen relativ einfach auf IPv4-Adressen abgebildet werden können.

Zum Transport von IPv4-Paketen in IPv6-Netzen ergibt sich die IPv4-kompatible IPv6-Adresse durch das Auffüllen der freien Adressbit mit Nullen. Für den Fall, dass IPv6-Pakete über ein IPv4-Netz transportiert werden müssen, sind zwei Fälle zu unterscheiden. Verfügt der Zielknoten über eine IPv4-kompatible Adresse und einen Dualstack, so stellt er gleichzeitig das Ende eines so genannten automatischen Tunnels dar. Die Adressen des Tunnelendes und des Zielknotens sind in diesem Fall identisch. Handelt es sich bei dem Zielknoten jedoch um einen reinen IPv6-Knoten, so stellt ein Router das Tunnelende dar. In diesem Fall müssen zur Ermittlung der IPv4-kompatiblen Adresse des Tunnelendes die am Tunnelanfang zur Verfügung stehenden Routing-Informationen benutzt werden, um einen so genannten konfigurierbaren Tunnel aufzubauen.

Die beschriebenen Tunnelmechanismen, die auf dem Ansatz dualer Protokollstacks basieren, können nach der Vergabe aller IPv4-Adressen in der oben beschriebenen Form nicht mehr verwendet werden. Die dann zunehmende Verwendung nicht kompatibler Adressen erfordert alternative Ansätze zur Adressübersetzung wie NNAT (No Network Address Translation) [Brau99] oder Header-Translation [Brau99]. Während der NNAT-Ansatz weiterhin auf dem Konzept der dualen Protokollstacks basiert, nun allerdings mit temporär zugewiesenen IPv4-Adressen in Verbindung mit speziellen DHCP- (Dynamic Host Configuration Protocol) und DNS-Servern (Domain Name Service), verwendet der Header-Translation-Ansatz spezielle Server, die eine direkte Adressumsetzung auf der IP-Schicht vornehmen [Brau99]. Eine Zusammenfassung dieser und weiterer in der IETF-Arbeitsgruppe "Next Generation Transition (ngtrans)" erarbeiteten Ansätze findet sich unter [Dur01] im World-Wide-Web.

## 2.4.6 Grundlegende Routing-Algorithmen des IPv6-Internet

Die Weiterleitung der Datenpakete im Internet erfolgt im Wesentlichen anhand der Empfängeradresse. Mit Hilfe von Routing-Algorithmen werden dabei in den Zwischensystemen, den Routern, die Pfade zu den Zielsystemen ermittelt. Routing-Algorithmen sind entweder statisch oder dynamisch. Im Gegensatz zu dynamischen Routing-Algorithmen können statische Routing-Algorithmen nicht an aktuelle Netzwerksituationen

angepasst werden und sind daher für den Einsatz im Internet wenig geeignet [Wit99]. Dynamische Routing-Algorithmen werden in der Regel als verteilte Algorithmen implementiert [Wit99], bei denen jeder einzelne Router auf der Basis der ihm vorliegenden Informationen seine Routing-Entscheidung trifft. Die meisten der implementierten Verfahren bauen dabei entweder auf Distanz-Vektor-Algorithmen oder Link-State-Algorithmen auf.

## Distanz-Vektor-Algorithmus

Beim Routing auf der Basis des Distanz-Vektor-Algorithmus tauschen benachbarte Router periodisch Routen-Informationen (Wertepaar: destination, distance) aus [Com98]. Der Algorithmus überprüft die periodisch eintreffenden Nachrichten dahingehend, ob einem benachbarten Router eine günstigere Route zu einem Zielknoten bekannt ist als die aktuell gespeicherte. Die dabei aufgebauten Routing-Tabellen verfügen über einen Eintrag für jedes mögliche Zielsystem [Wit99] und enthalten Werte für die Distanz, den nächsten Router sowie die Angabe der Schnittstelle, über die dieser Router erreicht wird [Com98]. Ein Vorteil des Algorithmus ist seine Einfachheit [Wit99]. Nachteilig wirkt sich dagegen die Konvergenzzeit aus, die mit der Größe des Netzwerkes zunimmt [Wit99]. Ein auf der Basis des Distanz-Vektor-Algorithmus implementiertes, weit verbreitetes Routing-Protokoll ist das Routing-Information-Protocol (RIP) [Brau99].

## Link-State-Algorithmen

Link-State-Algorithmen setzen die Kenntnis der gesamten Netzwerktopologie in allen Routern voraus [Wit99]. Jeder Router berechnet auf der Basis dieser Informationen seine Routing-Tabelle. Durch das periodische Versenden zusätzlicher Dateneinheiten ermittelt jeder Router seine direkten Nachbarn, bzw. erkennt inaktive Übertragungsabschnitte. Diese inaktiven Übertragungsabschnitte werden den übrigen Routern im Netzwerk durch spezielle Statusinformationen mitgeteilt [Com98]. Das Konvergenzverhalten des Link-State-Algorithmus ist deutlich besser als das des Distanz-Vektor-Al-



gorithmus. Dies liegt unter anderem daran, dass das Weiterleiten der Statusinformationen über inaktive Übertragungsabschnitte unabhängig von einer Neuberechnung der Routing-Tabelle erfolgt. Beim Distanz-Vektor-Algorithmus dagegen erfolgt vor dem Weiterleiten der adäquaten Information zunächst eine Neuberechnung der Distanzvektoren. Eine Implementierung des Link-State-Algorithmus stellt das Protokoll Open-Shortest-Path-First (OSPF) dar [Brau99].

## 2.4.7 Multicast-Routing im Internet

Zusätzliche Anforderungen an die Routing-Algorithmen und -Protokolle werden durch den Einsatz von Multicasting gestellt. Hierbei ist im Gegensatz zu Unicast-Verfahren zu berücksichtigen, dass nur die Mitglieder einer Gruppe von Empfängern die für sie bestimmten Datenpakete erhalten. Die hierzu benötigten Informationen zur Gruppenzusammensetzung bzw. zum Aufenthalt der einzelnen Gruppenmitglieder lassen sich nicht aus den IP-Multicast-Adressen herleiten. Ein sehr einfaches Verfahren zum Weiterleiten der Multicast-Datenpakete stellt das Fluten (engl. Flooding) dar, das ohne den Aufbau von Routing-Tabellen auskommt [Wit99].

### Fluten

Das Fluten veranlasst die Router, die eintreffenden Multicast-Datenpakete über alle Schnittstellen ins Netz weiterzuleiten. Davon ausgenommen ist die Schnittstelle, über die das Datenpaket empfangen wurde. Dadurch wird einerseits sichergestellt, dass alle Router und damit auch alle potenziellen Empfänger die Datenpakete erhalten. Andererseits werden die Datenpakete aber auch in Teilnetze weitergeleitet, in denen sich keine Mitglieder der Multicast-Gruppe befinden. Das Verfahren ist im Distance-Vector-Multicast-Routing-Protocol (DVMRP) [Wai88] implementiert, aber grundsätzlich als ineffizient einzustufen. Ein weiterer Nachteil des Verfahrens ist die Gefahr der Schleifenbildung. Außerdem können wegen des hohen Broadcast-Charakters des Verfahrens keine geschlossenen Gruppen unterstützt werden.

## Reverse Path Forwarding (RPF)

Alternative Ansätze sehen den Aufbau von speziellen Verteilbäumen vor. Das Broadcasting der Multicast-Datenpakete erfolgt dann nur entlang der Pfade des Verteilbaumes. Hierdurch wird zwar die Schleifenbildung vermieden [Ste99], aber ein hohes Verkehrsaufkommen nahe der Wurzel. Diese stellt dabei einen Single-Point-of-Failure dar, da ihr Ausfall den Ausfall des Gesamtsystems bewirkt.

Empfängerbasierte Verfahren wie das Reverse Path Forwarding (RPF) erzeugen einen Verteilbaum je Sender [Wit99]. Die Router leiten die Datenpakete nur dann weiter, wenn sie über eine Schnittstelle eintreffen, die auf dem kürzesten Pfad zur Quelle liegt [Ste99]. Hierfür sind die Unicast-Routing-Informationen ausreichend, solange sie den kürzesten Pfad zur Quelle angeben. Da RPF keine Informationen über die Gruppenzusammensetzung auswertet, ist jedoch kein gezieltes Weiterleiten der Datenpakete möglich, so dass diese auch in solche Teile des Netzes weitergeleitet werden, in denen sich keine aktuellen Mitglieder der Gruppe befinden.

## Truncated Reverse Path Broadcasting (TRPB)

Verbesserungen des RPF-Verfahrens bewerten neben der Schnittstelle, über die ein Router die Datenpakete empfängt, auch die übrigen Schnittstellen. Das Reverse Path Broadcasting Verfahren (RPB) leitet die Datenpakete nur an Schnittstellen weiter, die in umgekehrter Richtung Teil des kürzesten Pfades sind [Ste99]. Das TRPB-Verfahren erweitert das RPF-Verfahren also dahingehend, dass die Weiterleitung nur dann erfolgt, wenn sich in dem Teilnetz mindestens ein aktives Gruppenmitglied befindet [Wit99]. Die hierzu benötigten Informationen werden aus dem IGMP gewonnen.

## Verteilbäume mit Rendezvous-Stellen

Einen Ansatz, der einen Verteilbaum je Gruppe aufbaut und damit das gezielte Weiterleiten der Datenpakete ermöglicht, stellen Verteilbäume mit Rendezvous-Stelle dar [Wit99]. Hierbei wird zunächst eine Rendezvous-Stelle gewählt, in der Informationen

über die Gruppenzusammensetzung abgelegt werden. Neue Mitglieder senden hierzu entsprechende Informationen an die Rendezvous-Stelle. Die auf diesem Weg liegenden Router speichern die entsprechenden Informationen, und der Zweig des Verteilbaumes wird als aktiv gekennzeichnet. Das Weiterleiten der Multicast-Datenpakete erfolgt dann nur in aktive Teile des Netzes, es handelt sich also nicht um ein Broadcast-Verfahren. Das Verfahren ist geeignet für weitverteilte Gruppen mit geringer Gruppendichte. Nachteilig wirkt sich die hohe Verkehrskonzentration um die Rendezvous-Stelle aus, die zudem einen Single-Point-of-Failure darstellt. Beide Nachteile können durch die Verwendung mehrerer Rendezvous-Stellen - auf Kosten eines erhöhten Verwaltungsaufwandes - reduziert werden.

## Steiner-Bäume

Die Verwendung von Steiner-Bäumen stellt einen weiteren Ansatz dar. Steiner-Bäume werden global optimiert und sind daher bei jeder Topologieänderung neu zu berechnen. Dieser Ansatz ist jedoch ohne praktische Relevanz für die Datenkommunikation, da der Aufbau eines Steiner-Baumes ein NP-vollständiges Problem darstellt. In der Praxis der Datenkommunikation finden Steiner-Bäume keine Verwendung.

### 2.4.8 Application Level Framing

Das Konzept des Application Level Framing (ALF) wurde 1990 von Clark und Tennenhouse veröffentlicht [Cla90]. Es sieht vor, dass sämtliche Datenmanipulationsfunktionen ausschließlich auf der Anwendungsebene erfolgen und die Protokollschichten unterhalb der Anwendungsebene keine weiteren Manipulationen vornehmen. Die daraus resultierenden Datenblöcke werden als Application Data Unit (ADU) bezeichnet. Eine ADU soll auch dann noch verarbeitet werden können, wenn die Reihenfolge der ADU's verändert wurde oder ADU's fehlen. Außerdem sollen ADU's, die nicht in einem IP-Rahmen übertragen werden können, bereits auf der Anwendungsebene fragmentiert werden. Die hierfür benötigten Zusatzinformationen sind bereits in der ADU enthalten,

wodurch ein eigenes Rahmenformat benötigt wird. Ein solches Rahmenformat wird durch das offene und erweiterbare Realtime Transport Protocol (RTP) [Sch96] definiert.

Neben einer Entlastung der unteren Protokollschichten bietet das ALF-Konzept zusätzlich den Vorteil, dass weitere Dienste, z.B. Fehlererkennungs- und Behandlungsmechanismen auf der Anwendungsebene implementiert werden können. Dies ist insbesondere in Kombination mit unzuverlässigen Transportdiensten wie dem User Datagram Protocol (UDP) [Pos80] von Interesse. UDP stellt im Internet einen verbindungslosen unzuverlässigen Dienst auf der Transportebene zur Verfügung. Im Gegensatz dazu bietet das Transmission Control Protocol (TCP) [Pos81a] einen verbindungsorientierten, zuverlässigen Transportdienst.

#### 2.4.9 Dienstgarantien im Internet

Multimediale Echtzeit-Anwendungen lassen sich in zufriedenstellender Qualität nur bei ausreichender Bandbreite und Einhaltung der Obergrenzen für die Ende-zu-Ende-Verzögerung und den Verzögerungsjitter realisieren. Der zunehmende Einsatz multimedialer Dienste und Anwendungen im Internet zeigt deutlich die Grenzen des bisher im Internet vorherrschenden Best-Effort-Services, der keinerlei Dienstgarantien übernimmt. Zur Reservierung der benötigten Ressourcen existieren zwei grundsätzliche Ansätze. Der Internet Integrated Services Ansatz (IntServ) [Bra94] sieht eine Reservierung der benötigten Ressourcen pro Anwendung bzw. Datenstrom vor [Brau97]. Eine Alternative hierzu stellt der Differentiated Services Ansatz (DiffServ) [Bla98] dar, der Reservierungen für eine Menge von Datenströmen aggregiert [Wit99]. Die beiden Ansätze werden im Folgenden vergleichend diskutiert.

#### Der Integrated Services Ansatz (IntServ-Ansatz)

Die Reservierung der Ressourcen nach dem IntServ-Ansatz erfolgt pro Datenstrom und erfordert entsprechende Reservierungsinformationen in den Zwischensystemen (Router oder Switches). Zur Auswertung dieser Informationen benötigen die Zwischensysteme

zusätzliche Funktionalität. Die Reservierungsinformationen werden mit speziellen Signalisierungsprotokollen weitergeleitet, wobei empfängerorientierte und senderorientierte Ressourcenanforderungen unterschieden werden [Wit99]. In aktuellen Diskussionen um die Entwicklung geeigneter Signalisierungsprotokolle werden hauptsächlich das ResourceReSerVation Protocol (RSVP) [Brad97] und das Stream Protocol Version 2 (ST2) [Del95] genannt. RSVP ist ein nach dem empfängerorientierten Konzept arbeitendes Signalisierungsprotokoll. Der Nutzdatenverkehr wird über das IP-Protokoll organisiert.

RSVP erfordert die Implementierung von RSVP-Dämonen sowohl in den Zwischensystemen als auch in den Endsystemen [Wit99]. Der mit RSVP implementierte Dienst realisiert ein Soft-State-Konzept, bei dem der Empfänger keine Quittung für seine Reservierungsanforderungen erhält und auch keine Informationen über deren Erfolg oder Misserfolg.

ST2 implementiert ein senderorientiertes Konzept, das im Gegensatz zu RSVP ein eigenes Protokoll für den Nutzdatentransport integriert und damit nicht auf IP als Vermittlungsprotokoll aufbaut. Der ST2-Ansatz spielt daher im Internet keine große Rolle.

## Der Differentiated Services Ansatz (DiffServ-Ansatz)

Das Ziel des DiffServ-Ansatzes ist es, die aus dem IntServ-Ansatz resultierenden Nachteile des erhöhten Aufwandes für die Zustandshaltung in den Zwischensystemen zu vermeiden. Hierzu werden die Reservierungen für eine Menge von Datenströmen aggregiert. Die Zustandshaltung in den Zwischensystemen wird durch Mitführen der Dienstgüteinformation in den Dateneinheiten vermieden. Dadurch ist auch kein explizites Signalisierungsprotokoll erforderlich.

Beim DiffServ-Ansatz erfolgen die Ressourcenreservierungen zwischen den Netzsegmenten der Provider. Die daraus entstehenden DiffServ-Domänen haben im Gegensatz zu den Reservierungen des IntServ-Ansatzes über einen längeren Zeitraum Bestand. Die Aufteilung der aggregierten Reservierungen auf die einzelnen Datenströme erfolgt in

den Zwischensystemen und ist von der aktuellen Situation im Netzwerk abhängig. Es ist nicht vorgesehen, dass absolute Reservierungsanforderungen einzelner Datenströme zu erfüllen sind. Die Reservierungsanforderungen sind vielmehr immer relativ zu der aggregierten Reservierung zu sehen. Spezielle Protokoll-Implementationen für den DiffServ-Ansatz stehen noch nicht zur Verfügung.

## **2.5 Computer Supported Cooperative Working**

Die Anfänge des CSCW reichen zurück in die sechziger Jahre. Die Pionierarbeiten von Engelbart, der 1968 die Vision eines Konferenzraumes für zwanzig Personen mit gemeinsamem Darstellungsbereich und Mauszeiger hatte, sind unter anderem in [Lub95], [Eli91] und [Gru94] zitiert. Mitte der 70er-Jahre wurden mit der Office-Automation die Hoffnungen auf eine Gruppenunterstützung im Bürobereich genährt [Gru94]. In den achtziger Jahren war ein zunehmendes Interesse an computerunterstützter Kooperation und die Vision eines weltumfassenden Computernetzes, zu dem jedermann Zugang besitzt, um sowohl im privaten wie auch im professionellen Bereich mit Anderen zu kooperieren, zu verzeichnen [Schw98]. Jedoch haben sich nicht alle Hoffnungen erfüllt, und viele der ersten CSCW-Systeme scheiterten [Lyn90], [Gru88]. Seit Mitte der 80er-Jahre wurden in verschiedenen Wissenschaftsdisziplinen durch intensive Forschungsarbeiten wichtige Beiträge sowohl zu den theoretischen Grundlagen als auch Implementierungskonzepte für Groupware erarbeitet. Dieser Abschnitt soll hierzu einen Überblick geben, wobei der Schwerpunkt auf den grundlegenden Konzepten liegt, die für das weitere Verständnis dieser Arbeit von Bedeutung sind.

### **2.5.1 Grundlegende Eigenschaften von CSCW-Systemen**

#### **Asynchrone CSCW-Systeme**

Asynchrone CSCW-Systeme unterstützen die Kommunikation und die Gruppenarbeit räumlich entfernter und zu unterschiedlichen Zeiten arbeitender Gruppenmitglieder. Die

Unterstützung der Gruppenarbeit bezieht sich im Wesentlichen auf die Zugriffsverwaltung auf gemeinsame Arbeitsmittel. Dabei handelt es sich vorrangig um Dokumente sowie Werkzeuge zu deren Bearbeitung [App01]. Gemeinsame Arbeitsbereiche stellen den zentralen Zugriffspunkt auf die Informationen dar. Borghoff und Schlichter [Bor98] verwenden im Zusammenhang mit der systemorganisierten Verwaltung von Gruppendokumenten den Begriff des gemeinsamen Informationsraumes. In kollaborationsorientierten, asynchronen CSCW-Systemen steht die Unterstützung der Gruppenarbeit im Vordergrund [App01]. Grundlegende Konzepte hierbei sind die Benutzer- und die Zugriffsverwaltung sowie der gemeinsame Arbeitsbereich. Einer der bekanntesten Vertreter dieser Systemgruppe ist das von der GMD (Gesellschaft für Mathematik und Datenverarbeitung) entwickelte Shared Workspace System BSCW (Basic Support for Cooperative Work [Ben95]).

Viele asynchrone CSCW-Systeme enthalten auch eine Kommunikationskomponente. Erfolgt dabei die Kommunikation ausschließlich über die Gruppendokumente, so sprechen Borghoff und Schlichter von impliziter Kommunikation [Bor98]. Pankoke-Babatz [Pan01] verwendet für die Kommunikationsunterstützung räumlich entfernter und zu unterschiedlichen Zeiten arbeitender Personen den Begriff der asynchronen, elektronischen Kommunikationssysteme. Zu den bekanntesten Vertretern zählen hierbei die E-Mail-Systeme. Mit E-Mail-Systemen lassen sich sowohl einzelne Empfänger als auch Gruppen von Empfängern<sup>3</sup> adressieren. Newsgroups und Bulletin-Boards sind weitere Anwendungsbeispiele.

---

<sup>3</sup> Hierfür werden sogenannte Verteiler- oder Mailinglisten verwendet.

## Synchrone CSCW-Systeme

Synchrone CSCW-Systeme unterstützen die Kooperation und die Kommunikation gleichzeitig arbeitender Gruppenmitglieder. Hier ist weiter zu differenzieren, ob sich die Gruppenmitglieder im gleichen<sup>4</sup> oder in verschiedenen Räumen<sup>5</sup> befinden.

Kommunikationsorientierte synchrone CSCW-Systeme bieten vorrangig Möglichkeiten zur technisch vermittelten Kommunikation [Kai01]. Kollaborationsorientierte synchrone CSCW-Systeme fokussieren die zeitgleiche Zusammenarbeit mehrerer Personen in einem gemeinsamen Arbeitsbereich [Hol01]. Die hierzu benötigten Funktionen erfordern spezielle Konzepte hinsichtlich der Kooperation, der Koordination und der Kommunikation bei der Verwendung des gemeinsamen Arbeitsbereichs. Einige dieser Konzepte werden in den nächsten Abschnitten vorgestellt.

### 2.5.2 Gruppenwahrnehmung

Von zentraler Bedeutung ist bei CSCW-Systemen, dass sich alle Teilnehmer, jederzeit, aller aktuellen und vergangenen Aktionen im Arbeitsbereich bewusst sind. Es muss zum Beispiel nicht nur ersichtlich sein, dass ein Objekt verändert wird, sondern es muss auch klar sein, wer es verändert. Hierfür hat sich im Englischen der Begriff *group awareness* oder einfach *awareness* etabliert. Im Deutschen wird von Gruppenbewusstsein, Gewahrsein, sozialer Präsenz oder Gruppenwahrnehmung gesprochen. Prinz [Pri01] argumentiert, dass es nicht das Ziel ist, ein Bewusstsein zu erzielen, sondern vielmehr eine Wahrnehmung des kooperativen Geschehens. Er schlägt daher die Begriffe Gruppenwahrnehmung oder Geschehenswahrnehmung vor. Diesem Vorschlag wird in dieser Arbeit gefolgt.

---

<sup>4</sup> Entsprechend der in Tabelle 1 eingeführten funktionalen Klassifikation handelt es sich hierbei z.B. um Elektronische Sitzungsräume.

<sup>5</sup> Entsprechend der in Tabelle 1 eingeführten funktionalen Klassifikation handelt es sich hierbei z.B. um Desktopkonferenzsysteme.



Ansätze zur Steigerung der Gruppenwahrnehmung lassen sich auf verschiedenen Ebenen finden. Die meisten betreffen die Anwendungs- bzw. die Darstellungsebene, aber auch auf der netzwerktechnischen Ebene kann z.B. die Wahl einer geschlossenen statt einer offenen Multicast-Gruppe als Maßnahme zur Steigerung der Gruppenwahrnehmung angesehen werden. Prinz stellt ein Klassifikationsschema vor, das nach zeitlicher und räumlicher Kooperationsform unterscheidet sowie nach aufgabenorientierter und sozialer Gruppenwahrnehmung [Pri01].

## Zeitliche und räumliche Kooperationsform

Die Gruppenarbeit findet entweder synchron oder asynchron und an gemeinsamen oder getrennten Orten statt. Davon abhängig kommen der Laufzeit und der Granularität der Ereignisvermittlung unterschiedliche Bedeutung zu. Im Falle synchroner Gruppenarbeit sollen den Teilnehmern alle relevanten Ereignisse möglichst zeitgleich übermittelt werden. Darüber hinaus kann der erforderliche Detaillierungsgrad der Ereignisse im Falle synchroner Gruppenarbeit feingranularer sein als im asynchronen Fall.

## Aufgabenorientierte Gruppenwahrnehmung

Bei der aufgabenorientierten Gruppenwahrnehmung bezieht sich die Wahrnehmung auf Ereignisse im direkten Zusammenhang mit der Aufgabe und den gemeinsamen Objekten. Relevante Informationen können durch Benachrichtigungen an die einzelnen Gruppenmitglieder übermittelt werden. Bei der Änderung eines gemeinsamen Dokumentes sind dies z.B. Informationen über das aufgabenbezogene Objekt (z.B. Dokument, Terminplan), die Aktivität (z.B. Erstellen, Löschen, Ändern eines Dokumentes), das Ergebnis der Zustandsänderung (z.B. Dateiname, neue Versionsnummer) und den Bearbeiter (Name).

## Soziale Gruppenwahrnehmung

Im Vordergrund der sozialen Gruppenwahrnehmung steht die Wahrnehmung von Aktivitäten und Handlungen in einer gemeinsam genutzten Umgebung. Relevante Informa-

tionen betreffen hierbei z.B. die An- oder Abwesenheit einzelner Gruppenmitglieder. So genannte Media-Spaces steigern die soziale Gruppenwahrnehmung durch den Aufbau permanenter Audio- und Videoverbindungen.

### 2.5.3 Kooperationsunterstützung in synchronen CSCW-Systemen

Die Kooperationsunterstützung für synchrone CSCW-Systeme stellt Grundfunktionen zur Verfügung, welche die Bearbeitung gemeinsamen elektronischen Materials durch eine Gruppe von Benutzern überhaupt erst ermöglichen [Hol01]. Im Einzelnen gehören hierzu Funktionen zur Kontrolle und Organisation der Sitzung, die Bereitstellung von gemeinsamem Material und die Erzeugung eines gemeinsamen Kontextes.

#### Konferenz-Management und Sitzungs-Kontrolle

Synchrone CSCW-Systeme sind in der Regel Konferenzsysteme und erfordern die Bereitstellung spezieller Konferenzdienste. Hierzu gehören das Konferenz-Management, die Sitzungs-Kontrolle und die Floor-Kontrolle. Die Floor-Kontrolle wird als Konzept zur Koordinationsunterstützung in Abschnitt 2.5.4 vorgestellt.

Das Konferenz-Management stellt Funktionen für die notwendigen Aktionen *vor* Beginn einer Konferenz zur Verfügung [Hol01]. Eine einfache Möglichkeit zur Ankündigung einer Konferenz stellt beispielsweise das Versenden einer E-Mail dar. Das Konferenz-Management bietet aber auch Möglichkeiten, um z.B. die Art der Video- und Audiokodierung zu bestimmen und wichtige Konferenzparameter wie den Beginn und die Zeitdauer festzulegen. Weiterhin erfolgt das Initiieren und Löschen von Konferenzen mit Hilfe des Konferenz-Managements.

Um die organisatorischen Aktionen, die während einer aktiven Konferenz erfolgen, von denen des Konferenz-Managements abzugrenzen, wird der Begriff der Sitzung (engl. Session) eingeführt [Bran97]. Eine Sitzung<sup>6</sup> ist die

*"Online-Aggregation von Teilnehmern, die auf gemeinsam genutzten Ressourcen zusammenarbeiten."* [Gey99]

Zu den Aufgaben der Sitzungs-Kontrolle zählen im Wesentlichen die Teilnehmerverwaltung<sup>7</sup> und das Einleiten, Unterbrechen, Wiederaufnehmen und Beenden einer Sitzung.

## Gemeinsames Material

Das gemeinsame Bearbeiten einer Aufgabe setzt den Zugriff der einzelnen Teilnehmer auf das gemeinsame Material<sup>8</sup> und den zu dessen Bearbeitung erforderlichen Werkzeugen voraus. Gemeinsames Material kann von allen Teilnehmern bearbeitet werden und speichert das Ergebnis der Gruppenarbeit. Es schafft dadurch ein gemeinsames Verständnis, liefert einen Bezugspunkt und dient als Gedächtnis [Schw01].

*"Dadurch können die Teilnehmer feststellen, wie weit sie mit ihrer Arbeit sind, nachvollziehen, wie sie dort hingekommen sind und neue Beiträge in den Kontext des bisher Erreichten einordnen."* [Schw01]

Das gemeinsame Material muss stets für alle Teilnehmer im aktuellen Zustand sichtbar sein [Hol01]. Das Problem der Aktualisierung der Daten wirft aus technischer Sicht die Frage nach der jeweils geeigneten Strategie zur Verteilung der Daten auf [Hol01]. Darüber hinaus kommt im Falle einer Änderung des gemeinsamen Materials der Transparenz zwischen Ursache und Wirkung eine besondere Bedeutung zu. Nicht nur der Einzelne muss die Folgen seines Handelns nachvollziehen können sondern auch die

---

<sup>6</sup> Auch die Aufteilung einer Sitzung in mehrere Untersitzungen und das Zusammenfassen mehrerer Sitzungen zu einer Supersitzung ist möglich.

<sup>7</sup> Teilnehmer einer Sitzung können sowohl Menschen als auch Software-Agenten sein.

<sup>8</sup> Eigenes Material, in Abgrenzung zu gemeinsamem Material, ist nur im lokalen Arbeitsbereich seines Besitzers sichtbar und bearbeitbar.

übrigen Teilnehmer müssen unterscheiden können, durch wessen Handlungen, welche Folgen verursacht wurden. Hierbei kommen vorrangig Konzepte zur Steigerung der Gruppenwahrnehmung zum Einsatz.

## Gemeinsamer Kontext

Nach Geyer [Gey99] ist eine gemeinsame Sicht auf die von der Gruppe zu bearbeitenden Objekte die Grundlage synchroner Kooperation. Der zentrale Aspekt beim Entwurf synchroner CSCW-Systeme ist daher die Entwicklung einer gruppenbewussten Benutzerschnittstelle zur Darstellung der gemeinsamen Sicht. Borghoff und Schlichter sprechen in dem Zusammenhang vom gemeinsamen Kontext [Bor98].

Das grundlegende Konzept zur Realisierung des gemeinsamen Kontextes stellt allen Benutzern - zu jeder Zeit - alle Bildschirminformationen auf die gleiche Art und Weise dar. Dieses Konzept des strikten "What You See Is What I See" (WYSIWIS) gewährleistet den Benutzern eine konsistente Darstellung von gemeinsamen Dokumenten [Bor98]. Solange die Benutzer bei Verwendung gleicher Bildschirme mit gleicher Auflösung das Arbeitsfenster an der gleichen Stelle positionieren, ist die Referenzierung einzelner Objekte einfach zu realisieren, z.B. über einen zusätzlichen Audiokanal. Da die Aktionen der anderen Gruppenmitglieder jederzeit am Bildschirm verfolgt werden können, ist auch die Implementation zusätzlicher Funktionen zur Steigerung der Gruppenwahrnehmung relativ einfach. Ist jedoch eine der genannten Voraussetzungen nicht erfüllt, führt striktes WYSIWIS zu Problemen, da dann nicht mehr von identischen Bildschirminformationen bei allen Gruppenmitgliedern ausgegangen werden kann. In jedem Fall sind Regeln oder Absprachen zur Veränderung der Bildschirminformationen erforderlich, um einen Scroll War oder einen Window War zu vermeiden.

Ein Scroll War liegt z.B. vor, wenn ein Benutzer die aktuelle Bildschirmseite lesen möchte, während ein zweiter Benutzer durch Blättern (scrollen) den Bildschirminhalt ändert. Von einem Window War ist die Rede, wenn ein Benutzer die aktuelle Bildschirmseite durch das Erzeugen eines neuen Fensters überdeckt [Bor98]. Aus Gründen

der Konsistenzhaltung der Bildschirminhalte untersagt das Konzept des strikten WYSIWIS auch die Benutzung privater Arbeitsbereiche.

Das strikte WYSIWIS wirkt oftmals einschränkend. Eine Abschwächung des Konzeptes ermöglicht individuelleres Arbeiten bei gleichzeitiger Verminderung des Gruppenbewusstseins [Gey99]. Dieses so genannte relaxierte WYSIWIS existiert in verschiedenen Ausprägungen:

- **Getrennte Arbeitsbereiche**

Das Prinzip der getrennten Arbeitsbereiche sieht eine Trennung von privaten und öffentlichen Arbeitsbereichen vor. Die Darstellung erfolgt üblicherweise in verschiedenen Fenstern, wobei der öffentliche Arbeitsbereich dem Konzept des strikten "WYSIWIS" folgt und sich über den gesamten Bildschirm erstreckt. Der private Arbeitsbereich ist für andere Teilnehmer unsichtbar und kann benutzt werden, um unabhängig von den anderen Gruppenteilnehmern Ideen vorzubereiten oder auszuarbeiten.

- **Getrennte Sichten**

Bei dem Konzept der getrennten Sichten verwenden die Teilnehmer entweder verschiedene Sichten mit unterschiedlichen Darstellungen des gleichen Datenbestandes oder die gleiche Sicht auf verschiedene Bereiche des gemeinsamen Dokumentes. Der erste Fall erlaubt z.B. die Darstellung der Daten als Graph oder als Tabelle. Im zweiten Fall wird z.B. bei einem Teilnehmer das Inhaltsverzeichnis eines Textdokumentes und bei einem anderen Teilnehmer die Zusammenfassung dargestellt. In beiden Fällen geht die Möglichkeit zur einfachen Referenzierung von Objekten verloren. Ebenso werden die Möglichkeiten des gruppenbewussten Arbeitens teilweise eingeschränkt.

- **Getrennte Synchronisation**

Die Synchronisation der Bildschirminhalte erfolgt bei dieser Abschwächung des strikten WYSIWIS nicht mehr gleichzeitig bei allen Teilnehmern. Die Verzögerung kann systemintern oder explizit von einem Teilnehmer gesteuert werden. Im zweiten Fall erfolgt die Aktualisierung des Bildschirminhalts zunächst lokal bei einem Teilnehmer. Erst nach Freigabe der Änderungen durch diesen Teilnehmer erfolgt die Aktualisierung der Bildschirminhalte bei den übrigen Teilnehmern.

- **Getrennte Untergruppen**

Das Konzept der getrennten Untergruppen sieht die Aufteilung der nach dem strikten WYSIWIS-Konzept arbeitenden Teilnehmer in verschiedene Untergruppen vor.

## 2.5.4 Koordinationsunterstützung in synchronen CSCW-Systemen

Die synchrone, computerunterstützte, verteilte Gruppenarbeit ist geprägt von zeitgleichen, parallelen Aktivitäten an gemeinsamen Objekten. Eine Koordinationsunterstützung für synchrone CSCW-Systeme betrifft daher in erster Linie die Unterstützung dieser Aktivitäten [Hol01]. Dabei sind Lösungen für Konfliktsituationen zu erarbeiten, die in Präsenzsituationen z.B. durch soziale Protokolle gelöst werden, vgl. [Gey99]. Einen ersten Ansatz bietet die Kombination mehrerer CSCW-Systeme. Zusätzliche Audio- und Videokonferenzsysteme erweitern z.B. beim Einsatz eines Whiteboard das Spektrum der möglichen Kommunikationskanäle. Diese steigern einerseits die soziale Gruppenwahrnehmung. Andererseits können sie beschränkt zur Koordination der parallelen Aktivitäten eingesetzt werden [Gey99]. Die prinzipielle Notwendigkeit kollaborativer Dienste bleibt davon aber unberührt. Zu den wichtigsten kollaborativen Diensten gehören die Sitzungskontrolle (Session-Control), die Floor-Kontrolle, der Telepointer und das Voting. Die Sitzungskontrolle als Konzept für die Kooperationsunterstützung wurde bereits im vorigen Abschnitt vorgestellt. Die übrigen Konzepte werden nachfolgend beschrieben.

### Floor-Kontrolle

Die zentrale Frage im Zusammenhang mit einer Koordinationsunterstützung für synchrone CSCW-Systeme ist, "wer darf überhaupt wann und wie lange welche Aktionen im gemeinsamen Arbeitsbereich ausführen?", vgl. [Hol01]. Im Gegensatz zu klassischen verteilten Systemen, bei denen aus Konsistenzgründen das Ziel der Nebenläufigkeitskontrolle die Vermeidung gleichzeitiger Veränderungen an gemeinsamen Objekten ist, ist die Bedeutung des Nebenläufigkeitsbegriffs im Zusammenhang mit CSCW-Systemen ein leicht anderer [Bor98]. Hierfür wird häufig auch der Begriff der Floor-Kon-

trolle verwendet, den Zitterbart und Brand [Bran97] gegenüber der klassischen Nebenläufigkeitskontrolle durch die direkte Involvierung der Teilnehmer und somit der erforderlichen Berücksichtigung sozialer Protokolle abgrenzen. Der Floor steht in dem Zusammenhang als Metapher für ein gemeinsames Rednerpult [Gey99] und ist als abstrakter Begriff mit dem Zugriffsrecht auf bestimmte Ressourcen verknüpft [Bran97]. Diese Berechtigung ist transient und stattet ihren Besitzer mit wohldefinierten Rechten aus [Gey99]<sup>9</sup>.

Bekannte Verfahren zur Floor-Kontrolle sind entweder pessimistisch oder optimistisch. Optimistische Verfahren setzen zur Konfliktlösung "*auf den gesunden Menschenverstand und das soziale Verhalten* [Flu96]". Grundsätzlich kann dabei jeder Benutzer zu jedem Zeitpunkt jedes Objekt ändern. Pessimistische Verfahren versuchen Konflikte a priori zu verhindern. Fluckiger [Flu96] differenziert pessimistische Verfahren weiter:

- Implizites Sperren: Beim Zugriff auf eine Ressource wird diese durch das System exklusiv für den jeweiligen Benutzer gesperrt. Erst wenn der Benutzer die Ressource eine längere Zeit nicht verwendet, wird sie für die übrigen Teilnehmer wieder freigegeben.
- Explizites Sperren: Der Zugriff auf eine Ressource muss durch den Benutzer explizit angefordert werden. Die Anforderungen werden vom System verwaltet und bedient, z.B. First-Come-First-Serve-Prinzip. Zur Steigerung der Gruppenwahrnehmung sollten die aktuellen Anforderungen für alle Teilnehmer sichtbar sein.
- Vorsitzkontrolle: Ein Teilnehmer agiert als Sitzungsleiter und entscheidet über die Vergabe des Floors. Die Rolle des Sitzungsleiters sollte flexibel handhabbar sein.

Brand und Zitterbart haben insgesamt festgestellt:

*"Obwohl bei modernen Konferenz- und Kollaborations-Anwendungen die Existenz einer Floor-Kontrolle notwendig erscheint, ist die Anzahl von Systemen mit realisierter Kontrollstrategien relativ klein". [Bran97]*

---

<sup>9</sup> Die nötigen Informationen über die grundlegenden Objekte einer Sitzung, die Teilnehmer und Ressourcen, werden der Floor-Kontrolle von der Sitzungs-Kontrolle bereitgestellt.

## 2.5.5 Kommunikationsunterstützung in synchronen CSCW-Systemen

Die notwendigen Voraussetzungen zur Kommunikation in synchronen Systemen werden in der Regel durch Video-, Audio- oder Chat-Komponenten geschaffen. Kommunikation im Sinne einer Unterstützung der parallelen Aktivitäten an gemeinsamen Objekten erfordert darüber hinaus gehende Funktionalitäten. Im Folgenden werden hierzu die Konzepte des Telepointers und des Votings vorgestellt.

### Telepointer

Die Möglichkeit, die Aufmerksamkeit der Teilnehmer durch explizites Zeigen auf bestimmte Objekte zu lenken, ist eine wesentliche Eigenschaft von Face-to-Face-Arbeitssitzungen. Diese Möglichkeit wird in synchronen CSCW-Systemen durch das Telepointer-Konzept nachgebildet. Borghoff und Schlichter definieren einen Telepointer als

*"ein ausgezeichnete Cursor, der auf mehr als einem Bildschirm erscheint und von verschiedenen Teilnehmern einer Gruppensitzung bewegt werden kann." [Bor98]*

Aus der Sicht des Systems ist ein gemeinsamer Telepointer als eine zusätzliche Resource zu sehen, die über die Floor-Kontrolle verwaltet werden sollte.

Eine weitere Möglichkeit besteht darin, je einen Telepointer pro Teilnehmer zu implementieren. Die Übersichtlichkeit nimmt dabei in dem Maße ab, wie die Teilnehmerzahl zunimmt. Ebenso ist die Zuordnung der Telepointer zu den Benutzern schwierig.

Der Gültigkeitsbereich eines Telepointers kann sich auf ein oder mehrere Fenster erstrecken. Mit zunehmender Relaxation des WYSIWIS-Prinzips steigt auch die Komplexität der Implementation von Telepointern. Bei der Umsetzung des strikten WYSIWIS gestaltet sich die Implementation von Telepointern am einfachsten.



## Voting

Kooperative Arbeitsprozesse erfordern von Zeit zu Zeit die Bewertung von Alternativen. In Face-to-Face-Arbeitssitzungen sind solche Abstimmungen leicht durchzuführen. Die Implementation in CSCW-Systemen erfolgt als Voting-Werkzeug. Neben der Möglichkeit zur Abstimmung lassen sich über Voting-Werkzeuge auch einfache Meinungsumfragen implementieren. Insbesondere letztere bieten z.B. einem Vortragenden die Möglichkeit, Rückmeldungen aus dem Kreis der übrigen Teilnehmer zu erhalten [Gey99].

Wie auch der Telepointer, werden Voting-Werkzeuge aus Systemsicht als zusätzliche Ressourcen betrachtet, die über die Floor-Kontrolle verwaltet werden müssen.

### 3. Entwurf, Implementation und Evaluation von Groupware

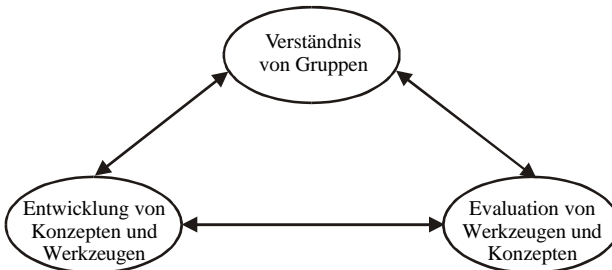
Die Implementation von Groupware ist im Vergleich zur Implementation von Einbenutzeranwendungen deutlich komplexer. Schümmer und Schuckmann [Schü01] fassen für die Implementation von Real-Time-Groupware die folgenden Herausforderungen zusammen.

- Aufwendigere Benutzerschnittstellen durch Umsetzung geeigneter Konzepte zur Gruppenwahrnehmung.
- Aufwendigere Datenhaltung durch erwünschten Mehrbenutzerzugriff. Gleichzeitige Eingaben von verschiedenen Benutzern sind möglich und müssen verarbeitet werden.
- Aufwendigere Konsistenzhaltung, da es sich im allgemeinen um verteilte Anwendungen handelt. Fragestellungen in dem Zusammenhang betreffen u.a. die Interprozesskommunikation, die Kommunikation zwischen Prozessen auf mehreren Rechnern und die Prozesssynchronisation.

Darüber hinaus gestaltet sich auch die Evaluation vielschichtiger und anspruchsvoller. Insbesondere gilt es, positive und negative Aspekte innovativer Groupware-Systeme in der Evaluation offen zu legen [Eng01]. In diesem Kapitel werden zunächst ein Vorgehensmodell für die Entwicklung von Groupware und anschließend Methoden zum Studium von Gruppen vorgestellt. Darauf aufbauend werden Architekturmodelle und Verteilungsarchitekturen zur Modellierung von Groupware diskutiert. Implementationsaspekte sowie ein Bewertungsmodell für Laborstudien und sein Einsatz bei der Evaluation von Groupware schließen das Kapitel ab.

### 3.1 Ein Vorgehensmodell für die Entwicklung von Groupware

Die Entwicklung von Einbenutzersystemen folgt vielfach dem von Boehm vorgeschlagenen Wasserfallmodell [Boe78], bzw. einem seiner vielen Derivate. Im Zentrum dieser technisch-zentrierten Sicht stehen die zu realisierenden Funktionen. Die Benutzer-zentrierte Sicht, die vorrangig die Bedürfnisse eines Anwenders bei der Benutzung einer Groupware betrachtet, ist in diesen Modellen jedoch unterrepräsentiert. Krmar [Krc91] forderte bereits Anfang der neunziger Jahre eine engere wechselseitige Verzahnung von Arbeiten zum Verständnis von Gruppen, der eigentlichen Konzept- und Werkzeugentwicklung sowie der Evaluation dieser Konzepte und Werkzeuge. Aus diesen Forderungen leitete Krmar das in Abbildung 11 dargestellte Vorgehensmodell für die Entwicklung von Groupware ab.



**Abb. 11:** Vorgehensmodell für die Entwicklung von Groupware [Krc91]

Vergleichbare Modelle wie die von Dobson<sup>10</sup> oder auch Tang [Tan91] folgen dem gleichen Ansatz. Während Krmar keine Angaben zur Reihenfolge der Abarbeitung der Arbeitsschritte macht, schlägt Tang einen iterativen Prozess vor. Dabei ist zunächst durch eine Analyse der Gruppenarbeit ein Verständnis für den aktuellen Zustand zu schaffen.

---

<sup>10</sup> Dobson stellte sein Modell 1991 auf einem Workshop in Berlin vor. Eine Beschreibung findet sich in [Lub95].

## 3.2 Methoden zum Studium von Gruppen

Für das Studium von Gruppen existieren eine Reihe unterschiedlicher Methoden die jeweils Stärken und Schwächen aufweisen. Folgende Fragestellungen stehen dabei im Vordergrund [Bor98]:

1. Wer sind die Akteure?
2. Welche Verhaltensweise soll untersucht werden?
3. Was ist der aktuelle und realistische Kontext der Gruppenarbeit?

Methoden zum Studium von Gruppen sind u.a.:

- **Feldstudien:** Reale Arbeitsgruppen werden ohne Eingriff in den Kontext der Gruppe beobachtet. Die Beobachtung kann z.B. mittels Video- und Audioaufnahmen dokumentiert werden. Die Methode zeichnet sich durch eine hohe Genauigkeit hinsichtlich des Realismus der Situation aus.
- **Feldexperimente:** Hierbei findet ein Eingriff in den Kontext der Gruppe mit dem Ziel, Informationen über die Auswirkungen dieses Eingriffs zu gewinnen, statt. Sonst sind Feldexperimente den Feldstudien sehr ähnlich.
- **Laborexperimente:** Die Gruppenumgebung wird im Labor nachgebildet. Dabei sind alle externen Bedingungen kontrollierbar und damit deren Einflüsse und Auswirkungen auf die Gruppenarbeit studierbar.
- **Testsimulationen:** Das real existierende System wird hierbei zu Studienzwecken nachgebildet.
- **Umfragen:** Gezielt ausgewählte Personen bilden die Stichprobe und stellen sich den gleichen Fragen.
- **Beurteilungsstudien:** Im Allgemeinen werden kleinere Stichproben als bei Umfragen gebildet und gezielt im Hinblick auf strittige Punkte befragt.

Die beschriebenen Methoden bieten nicht in gleicher Weise eine Unterstützung für die oben genannten Fragestellungen. Zum Beispiel ist für Feldstudien der Unterstützungs-

grad der Fragestellung 3 hoch, geht aber zu Lasten der Fragestellungen 1 und 2. Labor-experimente hingegen unterstützen das Studium der Verhaltensweise (Fragestellung 2), liefern jedoch kaum aussagekräftige Antworten hinsichtlich der Fragestellungen 1 und 3. Umfragen dagegen zeichnen sich durch eine hohe Generalisierbarkeit aus (Fragestellung 1). Andererseits liefern Umfragen aber auch nur eine geringe Genauigkeit bei der Messung der Verhaltensweisen (Fragestellung 2) und besitzen wenig Aussagekraft bzgl. des Realismus (Fragestellung 3). Zur Abdeckung des gesamten Spektrums der Fragestellungen ist daher der Einsatz mehrere Methoden mit unterschiedlichem Fokus sinnvoll.

### 3.3 Modellierung von Groupware

Groupware-Anwendungen sind im Allgemeinen verteilte Anwendungen, die aus verschiedenen, über mehrere Rechner verteilte Komponenten bestehen. Die Zerlegung der Groupware in Komponenten wird durch so genannte Architekturmodelle, deren Verteilung auf verschiedene Rechner anhand von Verteilungsarchitekturen beschrieben. Dem Metamodell nach Roth [Rot00] folgend, erfolgt das Design einer Groupware-Anwendung gemäß eines Architekturmodells<sup>11</sup>, wohingegen Verteilungsarchitekturen zur Implementation genutzt werden. *"Die Wahl einer bestimmten Verteilungsarchitektur hat enorme Einflüsse auf das entsprechende Groupware-System"*. [Rot00]

#### 3.3.1 Architekturmodelle für Groupware

Software-Architekturmodelle (im Folgenden kurz: Architekturmodelle) sind aus der Welt der Einbenutzeranwendungen bekannt. Als klassischer Vertreter kann hier das Seeheim-Modell [Pfa83] genannt werden. Das erste Architekturmodell für Groupware wurde 1994 von Patterson [Pat95] vorgestellt. Es stellt die Aufgabe der Konsistenzhal-

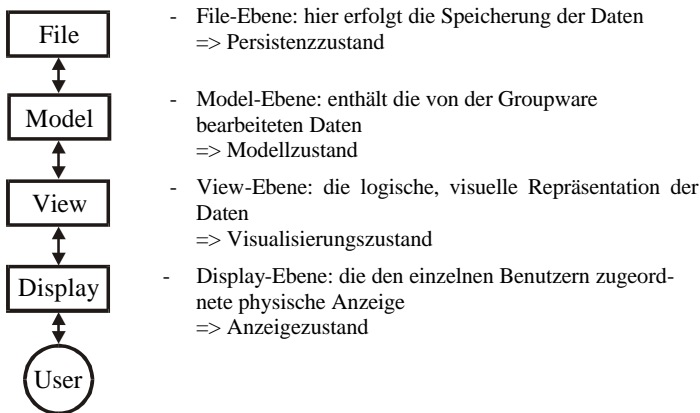
---

<sup>11</sup> Roth fasst in seinem Modell die im Metamodell von Philips [Phi99] vorgeschlagenen Ebenen Referenzmodell und Architekturstil zu einer neuen Ebene Architekturmodell zusammen.

tung der gemeinsamen Daten in den Vordergrund. Den eher klassischen Ansatz der Trennung zwischen Benutzerschnittstelle, Daten und Algorithmen verfolgt das MVC-Modell (Model-Viewer-Controller). Eine Reihe weiterer Modelle greifen das Grundkonzept des MVC-Modells auf und erweitern es um spezielle Aspekte. Im Rahmen dieser Arbeit wird neben den genannten Modellen zusätzlich noch das NetMVC-Modell behandelt.

## Groupware-Architekturen nach Patterson

Das Modell nach Patterson [Pat95] gliedert eine Anwendung in vier Ebenen und ordnet den Ebenen Anwendungszustände zu:



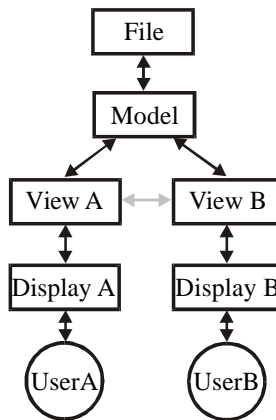
**Abb. 12:** Die vier Ebenen in Pattersons Architekturmodell

Patterson unterscheidet Modellvarianten mit gemeinsamen oder synchronisierten Zuständen. Eine dritte Variante ergibt sich aus der Kombination von gemeinsamen und synchronisierten Zuständen.

Die Variante synchronisierter Zustand geht davon aus, dass alle vier Ebenen dezentral vorliegen. Zur Konsistenzerhaltung müssen die korrespondierenden Ebenen der Benut-

zer miteinander kommunizieren, dazu sind die Zustände der einzelnen Ebenen zu synchronisieren. Die direkte Synchronisation der Displays wird in der Regel nicht weiter betrachtet, da sie nur schwer realisierbar ist. Sie kann jedoch über eine Synchronisation der Views erreicht werden.

Bei der Variante gemeinsamer Zustand werden ab einer bestimmten Ebene oberhalb der Display-Ebene die Zustände zentral verwaltet. Die nach Patterson geeignetste Architektur für synchrone Groupware ist die hybride Architektur: die Kombination aus gemeinsamen und synchronisierten Zuständen, vgl. Abbildung 13.



**Abb. 13:** Hybride Architektur nach Patterson

Die hybride Architektur nach Abbildung 13 kombiniert die Einfachheit einer zentralen Verwaltung des Modells mit der Flexibilität dezentraler Views. Daraus ergibt sich die für Groupware wichtige Möglichkeit, zwischen streng und lose gekoppelten Anzeigezuständen zu wechseln [Rot00].

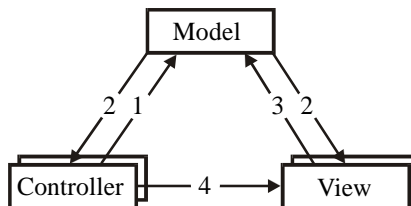
Hinsichtlich der Möglichkeiten zur Modellierung kontinuierlicher Datenströme weist das Architekturmodell nach Patterson jedoch Schwächen auf. Ebenso sind gemeinsame Anzeigezustände (z.B. Telepointer), die nicht auf Modellebene modelliert werden, nicht

abbildbar. Eine Verallgemeinerung des Architekturmodells nach Patterson wurde 1996 von Dewan vorgestellt [Dew96]. In seinem Modell sind weder die Anzahl der Modellebenen, noch deren Bedeutung genau festgelegt.

## Das MVC-Modell

Das MVC-Modell beschreibt interaktive Anwendungen anhand der Modellkomponenten *Model*, *View* und *Controller* (in den folgenden Beschreibungen wird aus Gründen der Einfachheit auf das Substantiv Modellkomponente verzichtet). Das *Model* enthält alle Programmteile, bestehend aus Datenstrukturen und Algorithmen, die nicht mit der Anzeige zusammenhängen. Die entsprechenden Funktionen zur Darstellung der Daten sind in den jeweiligen *Views* enthalten. Der *Controller* nimmt Benutzereingaben entgegen, z.B. über Mausklick oder Tastatur. Das *Controller-View*-Paar ist üblicherweise Bestandteil der Benutzerschnittstelle und greift auf genau ein *Model* zu. Jedoch können mehrere verschiedene *Controller-View*-Paare auf ein und dasselbe *Model* zugreifen und so jeweils unterschiedliche Sichten implementieren.

Der Datenfluss zwischen den einzelnen Modellkomponenten wird anhand der Abbildung 14 und der darin verwendeten Nummerierung erläutert.



**Abb. 14:** Das MVC-Konzept [Jan01]

Der *Controller* nimmt die Eingaben der Benutzer entgegen und leitet diese an das *Model* weiter (1). Das *Model* teilt den *Views* mit, dass es verändert wurde (2). Die *Views* wiederum erfragen die Änderungen im Einzelnen (3) und führen ihre Darstellung nach. Es ist ebenfalls möglich, dass die Benutzereingabe direkt auf die Darstellung wirkt (4),

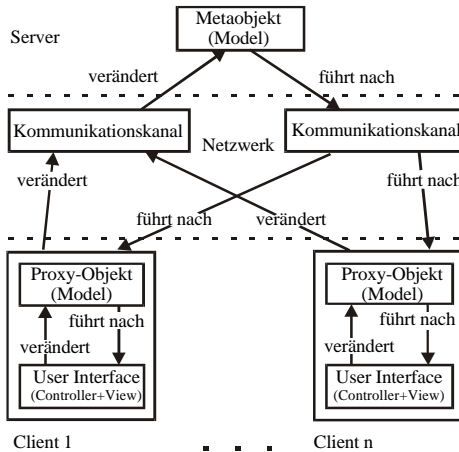


zum Beispiel um den Bildausschnitt zu scrollen. In diesem Fall wird das *Model* nicht verändert.

Das MVC-Modell kann zur Modellierung von Einbenutzer- und Mehrbenutzer-Anwendungen verwendet werden. Die Mehrbenutzer-Varianten basieren dabei auf der Koppelung des *Models*. Spezielle Groupware-Aspekte, wie z.B. die Synchronisation von Zugriffen auf gemeinsame Daten, können jedoch nicht vollständig modelliert werden [Rot00].

## Das NetMVC-Modell

In [Mah99] beschreiben Mahalek und Zitterbart eine Erweiterung des MVC-Konzeptes im Hinblick auf seine Netzwerkfähigkeit, mit der speziellen Zielsetzung es als Grundlage für Werkzeuge zur verteilten Software-Entwicklung einzusetzen. Dazu wurde das ursprüngliche Konzept zunächst dahingehend erweitert, dass das *Model* und die *Controller-View*-Paare in einem Netzwerk verteilt werden. Das *Model* liegt auf einem Server-Rechner und die *Controller-View*-Paare, die nun zur *Benutzerschnittstelle* zusammengefasst sind, befinden sich auf verschiedenen anderen Client-Rechnern. Das *Model* und die *Benutzerschnittstellen* kommunizieren über jeweils eigene Kommunikationskanäle, siehe Abbildung 15.



**Abb. 15:** Das NetMVC-Modell nach [Mah99]

Die Verteilung von *Model* und *Benutzerschnittstelle* im Netzwerk führt im Vergleich zur Anordnung auf einem Rechner zu Geschwindigkeitseinbußen bei der Kommunikation. Aus diesem Grund übermittelt das Model nicht nur die Nachricht, dass es verändert wurde, sondern direkt im ersten Schritt auch die für eine Änderung der Darstellung benötigten Daten. Diese werden von so genannten Proxy-Objekten gesammelt. Die Proxy-Objekte haben einerseits die Funktion eines Zwischenspeichers, andererseits verwalten sie Replikationen des *Models*. Die von den Proxy-Objekten gesammelten Daten werden erst auf das lokale Datenmodell angewendet und dann an die Benutzerschnittstelle weitergeleitet. Ebenso werden Eingaben von der Benutzerschnittstelle erst an die Proxy-Objekte gesendet und erst von dort an das *Model* auf dem Server. Das Proxy-Objekt kann dabei je nach Anwendung zunächst unterschiedlich viele Änderungswünsche sammeln. Im Falle einer Textverarbeitung kann z.B. überlegt werden, ob wirklich jeder geänderte Buchstabe als Änderungswunsch versendet wird oder nur ganze Wörter. Die Einführung von Proxy-Objekten erlaubt darüber hinaus die Verwendung privater Sichten. Ebenso erhöht es die Robustheit der Anwendung gegenüber Netzwerkausfällen.

### 3.3.2 Verteilungsarchitekturen

Verteilungsarchitekturen beschreiben die Verteilung der Groupware-Komponenten auf verschiedene Rechner und lassen sich in die drei folgenden Klassen einteilen:

#### Zentrale Verteilungsarchitekturen

Die gemeinsame Anwendung läuft auf einem zentralen Rechner. Auf den Anwendungsrechnern befinden sich lediglich die physikalischen Ein- und Ausgabeprozesse. Je nachdem ob sich die gemeinsame Anwendung ihres Einsatzes als Groupware bewusst ist oder nicht, wird von einer kollaborationsbewussten oder einer kollaborationstransparenten Anwendung gesprochen [Schü01]:

- Kollaborationstransparente, zentrale Architekturen sind besonders für die Einbindung von Einbenutzeranwendungen geeignet [Bor98]. Ein Vorteil dieser auch als Application-Sharing<sup>12</sup> bekannten Technik ist ihre einfache Implementation. Nachteilig wirkt sich aus, dass zu jedem Zeitpunkt nur ein Benutzer mit der Anwendung interagieren kann und dass keine entkoppelten Sichten möglich sind.
- Kollaborationsbewusste, zentrale Architekturen erlauben nicht den Einsatz beliebiger Anwendungen. Vielmehr sind die Anwendungen in diesem Fall speziell als Groupware konzipiert. Die Eingaben der Benutzer werden individuell behandelt und private Zustände sind möglich. Ein zentraler Prozess muss hierzu die Benutzerschnittstellen der Gruppenmitglieder bedienen.

#### Replizierte Verteilungsarchitekturen

Auf jedem Anwendungsrechner läuft eine vollständige Kopie der Anwendung.

---

<sup>12</sup> Die Application-Sharing-Technik basiert auf dem transparenten Einfügen einer Verteilungskomponente für Grafikanweisungen und einer Bündelungskomponente für Benutzerschnittstelleneignisse zwischen Anwendung und Fenstersystem [Schü01].

- Kollaborationstransparente, replizierte Architekturen verwenden die Schnittstelle zwischen Betriebssystem und Anwendung zur Koordination der physikalischen Eingaben. Darüber lässt sich dann bei gemeinsamen Anfangszuständen die gesamte Anwendung synchronisieren. Hierzu sind spezielle Algorithmen zur Konsistenzerhaltung erforderlich.
- Kollaborationsbewusste, replizierte Architekturen synchronisieren die Anwendungszustände. Das bietet ein höheres Maß an Flexibilität und erlaubt über private Zustände auch individuelle Sichten. Problematisch sind solche Fälle, in denen Benutzer verspätet einer Sitzung beitreten.

Replizierte Architekturen bieten den Vorteil, dass die Anwendungsinstanzen voneinander unabhängig sind und die Benutzerschnittstellen mit kurzen Antwortzeiten reagieren. Diese theoretischen Vorteile sind in der Praxis jedoch schwer umsetzbar. Insbesondere die Realisierung einer verteilten Konsistenzerhaltung ist eine überaus komplexe Aufgabe.

Die zuvor diskutierten zentralen und replizierten Verteilungsarchitekturen sind in ihrer Reinform in der Realität nur bedingt für die Entwicklung von Groupware geeignet. In der Praxis hat sich gezeigt, dass moderne, kollaborationsbewusste Anwendungen eher Mischformen der beiden Ansätze benutzen. Dies führt zu einer dritten Klasse von Verteilungsarchitekturen: den hybriden bzw. semi-replizierten Architekturen.

## Hybride Verteilungsarchitekturen

Die Vorteile der replizierten Architektur können genutzt werden, wenn die Konsistenzerkomponente zentralisiert wird. Dies führt auch zu einer deutlichen Vereinfachung im Hinblick auf die Realisierung.

- Kollaborationstransparente, hybride Architekturen entsprechen weitestgehend der replizierten Architektur und koordinieren ebenfalls die physikalischen Eingaben. Die entsprechende Konsistenzerkomponente ist in diesem Fall jedoch zentralisiert.

- Kollaborationsbewusste, hybride Architekturen verwenden ebenfalls eine zentrale Konsistenzerhaltungskomponente, synchronisieren aber wie im kollaborationsbewussten, replizierten Fall den gemeinsamen Anwendungszustand.
- Kollaborationsbewusste, hybride Architekturen mit Primarkopie halten zusätzlich den gemeinsamen Anwendungszustand zentral und vereinfachen dadurch das Problem des verspäteten Sitzungsbeitritts einzelner Teilnehmer.

### 3.3.3 Implementation von Groupware

Groupware lässt sich - stark vereinfacht dargestellt - in Anwendungsfunktionen und Gruppenfunktionen zerlegen [Rot00]. Die Implementation der Anwendungsfunktionen wird durch die zur Verfügung stehenden Werkzeuge hinreichend unterstützt. Vorrangig sind hier integrierte Entwicklungsumgebungen zu nennen, die verschiedene Werkzeuge zur Entwicklung bereitstellen: z.B. Editoren, Übersetzer und Bibliotheken. Der Einsatz von Datenbanksystemen kann darüber hinaus den Aufwand bei der Realisierung und Verwaltung umfangreicher Datenbestände minimieren. Die Implementation der Gruppenfunktionen kann und wird prinzipiell auch mit Hilfe der vorhandenen Werkzeuge und Bibliotheken realisiert werden, diese bieten allerdings keine problemspezifische Unterstützung. Wünschenswert wären spezielle Entwicklungsumgebungen mit z.B.:

- speziellen Spracherweiterungen bzw. neuen syntaktischen Elementen<sup>13</sup>;
- speziellen Bibliotheken zur effizienten Implementierung von Gruppenfunktionen;
- integrierten Testumgebungen für den Test von Groupware-Anwendungen.

Die Realisierung dieser speziellen Entwicklungsumgebungen erfolgt im Rahmen der Entwicklung spezieller Groupware-Plattformen oder -Frameworks [Rot00]. Diese erlauben es dem Entwickler, sich bei der Implementation wieder stärker auf die Anwendungsfunktionen zu konzentrieren. Die Implementation der Gruppenfunktionen wird

---

<sup>13</sup> Diese würden es dem Entwickler ermöglichen, die ihm bekannten Strukturen auch bei der Implementation von Gruppenfunktionen einzusetzen.

weitgehend durch die Groupware-Plattform unterstützt. Einen Überblick über bekannte Groupware-Plattformen bietet [Rot00]. Ein Vergleich verschiedener Implementationsstrategien bei der Entwicklung von Groupware findet sich bei Dewan und Riedl [Dew93]. Diese beschreiben ihre Erfahrungen mit den drei folgenden Varianten.

- **Manuelles Implementieren:** Hierbei werden Anwendungen von Grund auf neu implementiert. Dewan und Riedl bezeichnen diese Form der Anwendungsentwicklung einerseits als besonders komplex. Andererseits bietet sie dem Programmierer größtmögliche Flexibilität. Manuelles Implementieren bietet Vorteile bei der Entwicklung von Anwendungen, die komplexe grafische Darstellungen erfordern.
- **Generic Wrappers:** Dies sind spezielle Programme, in denen die Konsistenzerkomponente implementiert ist. Generic Wrappers nehmen die Eingaben der verteilten Benutzer entgegen, leiten diese an ein Einbenutzerprogramm weiter und stellen Funktionen zur Nebenläufigkeitskontrolle bereit. Der Verwendung von Generic Wrappers ist auf die Entwicklung kollaborationstransparenter Anwendungen beschränkt.
- **Flexible Frameworks:** Dewan und Riedl haben für die Entwicklung ihrer experimentellen Software-Entwicklungsumgebung FLECSE (Flexible Environment for Collaborative Software-Engineering) ein Framework, bestehend aus den beiden Systemen Multi-User Suite und SuiteSound Systems eingesetzt. Obwohl sie den Einsatz des Frameworks als überaus erfolgreich bezeichnen, ist zu bezweifeln, dass das verwendete Framework den heutigen Anforderungen noch genügen würde. Insbesondere, da es nicht die Entwicklung von Anwendungen mit komplexen Grafikfunktionen unterstützt.

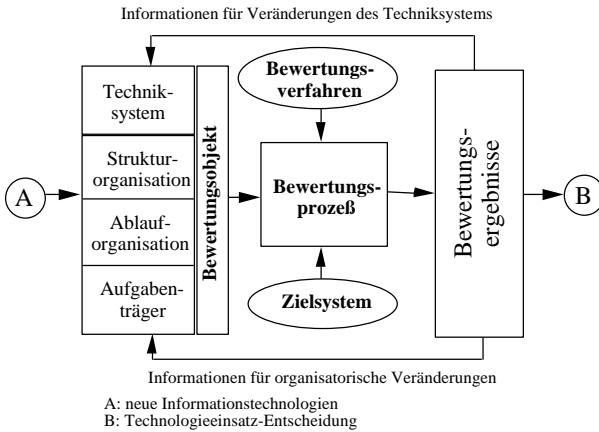
### 3.3.4 Evaluation von Groupware

Im Vorgehensmodell zur Entwicklung von Groupware nach Abbildung 11 ist die Evaluation als eine der Hauptaufgaben bei der Entwicklung von Groupware verankert. Das

Modell lässt jedoch offen, wann die Evaluation stattfinden sollte. Für Englberger [Eng01] ist eine Evaluation vor der Implementation unmöglich, und auch nach der Implementation ist nicht der optimale Zeitpunkt. Als realisierbare Alternative kommt eine Evaluation während eines innovativen Prozesses in Frage, also "*die Evaluation im 'hier und jetzt'*". [Eng01] Gappmaier und Häntschel interpretieren Krcmars Modell [Krc91] im Hinblick auf die Evaluation wie folgt:

*"Theorien beeinflussen die Entwicklung und Evaluierung von Werkzeugen, indem sie neue Entwicklungs- und Evaluierungsgrundlagen schaffen. Erkenntnisse, die bei der Entwicklung und Evaluierung von Werkzeugen gewonnen werden, wirken auf die Theorieentwicklung zurück. Die entwickelten Werkzeuge bestimmen den Gestaltungsrahmen für die Werkzeugevaluierung; die Evaluierungsergebnisse sind Input für die Weiterentwicklung der Werkzeuge".* [Gap97]

Der Evaluation von Groupware kommt sowohl aus technischer Sicht, im Hinblick auf ihre Komplexität und ihre Gruppenfunktionen, als auch aus organisatorischer Sicht eine besondere Bedeutung zu. Aus organisatorischer Sicht werden von der Evaluation Einschätzungen über Chancen und Risiken auf methodisch abgesicherter Basis erwartet [Eng01]. Die kommunikationstheoretische Sicht rückt die Bewertung der Kommunikationskomponente in den Vordergrund [Her99]. Diese wird auf der technischen Ebene durch die implementierte Benutzeroberfläche realisiert. Erfahrungen hinsichtlich der Wirkung von Groupware und dem Gruppenverhalten beim Einsatz von Groupware lassen sich mit Feld- und Laborexperimenten gewinnen [Eng01]. Wirklich brauchbare Untersuchungsmethoden sind jedoch kaum bekannt, bzw. nicht erfolgreich angewendet [Gap97]. Herrmann und Misch [Her99] schlagen speziell für die Evaluation von CSCL-Systemen z.B. die Entwicklung eines Kriterienkataloges zur Ermittlung der Systemunterstützung für definierte Kommunikationsaufgaben vor. Gappmaier und Häntschel dokumentieren eine der ersten Evaluationen, die auf einem anerkannten Bewertungsmodell für Laborstudien basiert, siehe Abbildung 16.



**Abb. 16:** Ein Bewertungsmodell für Laborstudien [Gap97]

Das Untersuchungsobjekt in dem Modell nach Abbildung 16 ist ein Informationssystem im Sinne eines Mensch-Aufgabe-Technik-Systems. Der Mensch ist als Aufgabenträger modelliert. Die Aufgabe selbst wird durch die Struktur- und Ablauforganisation abgebildet. Das Techniksystem, bestehend aus Hard- und Software, komplettiert das Untersuchungsobjekt. Das Bewertungsobjekt repräsentiert den Ausschnitt der Wirklichkeit, der in der Laborumgebung nachgebildet wird.

Die Eingaben des Bewertungsmodells sind neue Informationstechnologien. Der Bewertungsprozess wird durch das Zielsystem und das Bewertungsverfahren bestimmt. Dabei legt das Zielsystem die zu verwendenden Bewertungskriterien und das Bewertungsverfahren die anzuwendenden Methoden, Techniken und Werkzeuge fest.

Für dedizierte Evaluationen ist das angegebene Bewertungsmodell weiter zu konkretisieren. Insbesondere sind die Bewertungskriterien näher zu spezifizieren, sowohl hinsichtlich der zu bewertenden Funktionen bzw. Ziele als auch in Hinblick auf die einzusetzenden Untersuchungsmethoden, -techniken und -werkzeuge. Darüber hinaus ist ein Versuchsplan zu erstellen, und Hypothesen sind zu formulieren.



Gappmaier und Häntschel kommen zu dem Schluss, dass mit dem beschriebenen Modell Workflow-Management-Systeme mit einer Qualität bewertet werden können, die mit keiner anderen Untersuchungsmethode erreicht wird. Die Verwendung des Modells ist dabei keinesfalls auf die Evaluation von Workflow-Management-Systemen beschränkt<sup>14</sup>. Über die Eingabe (A) ist es vielmehr offen für neue Informationstechnologien.

---

<sup>14</sup> Vor den zitierten Laborexperimenten von Gappmaier und Häntschel wurde das Bewertungsmodell z.B. zur Evaluation von Client/Server Architekturen und lokalen Netzwerken benutzt.

## **4. Stand der Technik bei der Einführung von Groupware im Rahmen der Software-Engineering-Ausbildung**

Den Schwerpunkt dieses Kapitels bildet eine Analyse des Standes bei der Einführung von Groupware im Rahmen der Software-Engineering-Ausbildung und die Ableitung von Anforderungen an eine Systemunterstützung. Zunächst werden aktuelle Entwicklungen und Forschungsarbeiten zum Einsatz kooperativer Arbeitsformen und virtueller Teams im industriellen Software-Engineering vorgestellt. Anschließend werden ausschließlich Arbeiten im Anwendungskontext dieser Arbeit, der Software-Engineering-Ausbildung, diskutiert.

### **4.1 Software-Engineering**

Bereits zu Beginn des kommerziellen Computereinsatzes in den fünfziger Jahren wurden vereinzelt Modelle zur Arbeitsorganisation und -teilung in der Software-Entwicklung eingesetzt [The95]. Den aus der stetig steigenden Projektgröße und den fehlenden Möglichkeiten zur Leistungskontrolle resultierenden Problemen wurde in erster Linie mit Ansätzen begegnet, die eine Verbesserung der Programmierung zum Ziel hatten. Erst Ende der sechziger Jahre mehrten sich Ansätze, die eine Verbesserung der Arbeitsweise und eine Anlehnung des Vorgehens bei der Software-Entwicklung an die übrigen Ingenieurdisziplinen propagierten. Ein erster Höhepunkt dieser Entwicklungen wurde 1968 mit der ersten Software-Engineering-Konferenz in Garmisch [The95] erreicht. In der Folge wurde der Software-Entwicklungsprozess zunehmend durch Pha-

senmodelle beschrieben und verschiedene Vorgehensmodelle<sup>15</sup> wurden in den folgenden Jahren entwickelt. Dadurch wurde eine zeitliche und organisatorische Strukturierung des Entwicklungsprozesses mit eindeutig definierbaren Hierarchien und Zuständigkeiten, eine Aufteilung in unabhängig arbeitende Projektgruppen sowie eine am Arbeitsprozess orientierte Arbeitsteilung möglich [The95]. Aus dieser prozessorientierten Sichtweise resultieren aber auch neue Problemkreise, wie das Erkennen von Zusammenhängen zwischen Produktionsfehlern in den einzelnen Phasen und den daraus entstehenden Kosten zur Fehlerbehebung. Eine Zusammenstellung der zehn häufigsten Fehler bei der Software-Entwicklung findet sich bei [Wen95].

Insgesamt kann zusammengefasst werden, dass sich das moderne Software-Engineering den üblichen Schritten industrieller Produktentwicklungsprozesse angenähert hat und heute eine Reihe methodischer Ansätze und eine Vielzahl von Werkzeugen zur Verfügung stehen. Wallmüller definiert das Software-Engineering als eine Disziplin, die mit ingenieurmäßigen Mitteln und ökonomischem Vorgehen dem Entwickler hilft, qualitative hochwertige Software zu erstellen und zu pflegen, vgl. [Kne93]. Dennoch gibt es immer wieder Beispiele für fehlerhafte Software-Systeme, deren Ursprung in nach wie vor fehlerhaften Software-Entwicklungsprozessen begründet sind, vgl. [Gib94], [Nus97] und [Lev93]. Software-Engineering befindet sich immer noch in der Entwicklungs- und Einführungsphase bei einem sich rasant ändernden Umfeld sowie neuen Anforderungen, neuen Klassen von Anwendungen und neuen Arbeitsformen wie CSCW. Vor diesem Szenario spricht Derek Andrews schon von einer neuen Software-Krise [And99].

---

<sup>15</sup> Eine Unterteilung des Software-Entwicklungsprozesses in einzelne Phasen wurde erstmals 1970 von W. Royce [Roy70] vorgestellt und danach unter anderem von B.W. Boehm [Boe78] erweitert. Es existieren eine Vielzahl verschiedener Gliederungsmöglichkeiten, die sich in den Phasenbezeichnungen, -abgrenzungen, den jeweiligen Aktivitäten und dem Detaillierungsgrad unterscheiden. Neben dem Wasserfallmodell existieren eine Reihe weiterer Vorgehensmodelle wie das V-Modell [Boe78] oder das Spiralmodell [Boe86].

## 4.2 Kooperatives Software-Engineering in der industriellen Praxis

Der gesamte Lebenslauf heutiger Software-Systeme ist nur noch in Ausnahmefällen von einzelnen Entwicklern zu beherrschen. In großen Projekten ist sowohl eine Aufteilung der vertikalen Phasen Entwurf, Implementierung und Test auf mehrere Entwicklerteams als auch eine horizontale Aufteilung der Aktivitäten innerhalb der einzelnen Phasen üblich. Der Anteil an kooperativ durchgeführten Arbeiten liegt bei ca. 70% der gesamten Arbeitszeit eines Software-Entwicklers. In dem Zusammenhang wird in der Literatur auch der Begriff des kooperativen Software-Engineering verwendet, vgl. [Tie01]. Für Projektteams, deren Mitglieder darüber hinaus räumlich verteilt organisiert sind, hat sich in der Literatur der Begriff des virtuellen Teams etabliert, vgl. z.B. [Gor96].

Bei Gorton und Motwani [Gor96] findet sich eine Zusammenfassung der potenziellen Vorteile, die sich aus dem Einsatz virtueller Teams ergeben können:

- Ermöglichung einer Rund-um-die-Uhr-Entwicklung bei geeigneter Überlappung der Zeitzonen.
- Reduktion der Time-to-Market-Zeitspanne für Software-Produkte.
- Minimierung doppelten Expertenwissens an verschiedenen Standorten und bessere Nutzung der zur Verfügung stehenden, erfahrenen Entwickler.
- Ausnutzen des geringeren Verwaltungsaufwandes und der gut ausgebildeten Arbeitskräfte der Entwicklungsländer.
- Einbringen globaler Perspektiven in die Produktentwicklung bereits während der Entwicklung.
- Verbesserung der Software-Qualität durch einen verbesserten Entwicklungsprozess.

Wenn im Folgenden von verteiltem kooperativen Software-Engineering die Rede ist, ist damit die Kombination von kooperativem Software-Engineering und virtuellen Teams gemeint.

Die von Gorton und Motwani zusammengestellten Vorteile beim Einsatz virtueller Teams resultieren im Wesentlichen aus einer Analyse weltweit verteilter Teams. Dennoch lassen sich auch mit parallelen Teams am gleichen Standort einige dieser Vorteile nutzen.

#### 4.2.1 Software-Entwicklung mit parallelen Teams am gleichen Ort

Microsoft verfolgt einen Ansatz, bei dem große Projekte von mehreren kleinen, parallel arbeitenden Subteams (drei bis acht Entwickler) entwickelt werden. Aus übergeordneter Sicht agieren diese Subteams als ein großes Projektteam. Der Entwicklungsprozess gliedert sich in eine Planungs-, eine Entwicklungs- und eine Stabilisierungsphase. Speziell in der Entwicklungsphase wird die Aufgabe auf mehrere kleine, parallel arbeitende Subteams verteilt, die für ihr Teilprojekt den gesamten Projektzyklus von der Entwicklung über die Integration und das Testen bis hin zur Fehlerbehebung durchlaufen. Cusumano und Selby bezeichnen den Microsoft-Ansatz als Synch-and-Stabilize-Methode [Cus97], da

- die parallel arbeitenden Subteams die Aufgabe haben sich täglich zu synchronisieren ("synch") und so
- eine periodische, inkrementelle Verbesserung des Produktes ("stabilize") erreicht wird.

Ein Prinzip der Synch-and-Stabilize-Methode ist, zu jedem Zeitpunkt über ein lieferbares Produkt zu verfügen. Cusumano und Selby gehen soweit zu behaupten,

*"without its synch-and-stabilize structured approach, Microsoft would probably have never been able to design, build and ship the products it offers now and plans to offer in the future." [Cus97]*

Die tägliche Synchronisation erfordert sowohl ein hohes Maß an Kommunikation als auch umfangreiche Koordinationen. Zur Koordination des Zugriffs auf den Quellcode eines Produktes unterliegen alle Entwickler den gleichen strengen Regeln. Die in einem gemeinsamen Datenpool abgelegten Quellcode-Module eines Produktes dürfen zwar jederzeit von jedem beteiligten Entwickler geladen werden, allerdings erst nach einge-

hendem Test des Entwicklers und nur zu vorgegebenen Zeiten wieder in dem Datenpool gespeichert werden.

#### 4.2.2 Software-Entwicklung mit virtuellen Teams

Der Entwicklungsprozess der Siemens-ISDN-Kommunikationssoftware für die Systeme Hicom 300 und ROLM CBX 9751 kann als Beispiel für einen speziell auf den Aspekt der globalen Software-Entwicklung zugeschnittenen Workflow genannt werden [Haa94]. Hierzu nutzt Siemens als weltweit operierendes Unternehmen seit Beginn der neunziger Jahre die verschiedenen Zeitzonen seiner sechs Entwicklungsstandorte in Berlin, Boca Raton, München, Santa Clara, Wien und Zürich aus. Die Standorte sind über das globale Siemens-Forschungsnetz miteinander verbunden, wobei den Entwicklungsstandorten München und Boca Raton eine besondere Rolle zukommt. Dort werden jeweils die zentralen Datenbestände für die europäische bzw. die amerikanische Systemausprägung geführt. Wesentliche Voraussetzung für die verteilte Systementwicklung ist eine für beide Kontinente gemeinsame und eng abgestimmte Software-Entwicklungsstrategie. Unter dieser Voraussetzung ist es möglich, Teile der Software z.B. in München zu entwickeln, anschließend in den US-Datenpool nach Boca Raton zu übertragen und von dort ein testfähiges Gesamtsystem zurückzubekommen und dieses dann in München zu testen. Die Zeitdifferenz zwischen München und Boca Raton (6 Std.) bzw. Sta. Clara (9 Std.) bietet dabei den Vorteil, dass sie - weltweit betrachtet - einen 20-Stunden-Betrieb der Systementwicklung gestattet. Gorton und Motwani sprechen in dem Zusammenhang vom Overnight-Gain-Effect [Gor96].

#### 4.2.3 Anforderungen an Werkzeuge zum kooperativen Software-Engineering

In der Forschung wurden in der Vergangenheit eine Vielzahl von Mehrbenutzeranwendungen zur Kooperationsunterstützung sowie neue Möglichkeiten zur Gruppenkommunikation entwickelt. Viele dieser Ansätze eignen sich zwar prinzipiell für einen Einsatz im kooperativen Software-Engineering, aber nur die wenigsten sind speziell vor diesem

Hintergrund entwickelt worden. Ebenso existieren im Bereich CASE (Computer Aided Software-Engineering) schon seit längerem Werkzeuge, die zwar Gruppenfunktionen bereit stellen, aber insgesamt nicht als Groupware konzipiert sind<sup>16</sup>.

Die Anforderungen an Werkzeuge zum kooperativen Software-Engineering ergeben sich in erster Näherung aus einer Kombination der innerhalb der Arbeitsgebiete CSCW und CASE formulierten Anforderungen an eine Systemunterstützung. In dem Zusammenhang wurde auch der Begriff "Computer Supported Cooperative Software-Engineering" (CSCSE) geprägt und als eigenständiges Arbeitsgebiet etabliert [Tie01].

Das kooperative Software-Engineering wird in der Literatur im Wesentlichen aus zwei Perspektiven betrachtet. Im Zentrum der produktorientierten Sicht steht das Software-Produkt, das durch eine Vielzahl von Dokumenten (z.B. Entwurfsspezifikation und Benutzerhandbuch) beschrieben und gebildet (z.B. Quellcode-Dateien) wird. Wesentliche Aspekte der produktorientierten Sicht betreffen u.a. die Verfügbarkeit dieser Artefakte, den gleichzeitigen Zugriff auf verteilte Ressourcen und das Versionsmanagement.

Die prozessorientierte Sicht stellt die Dynamik des Entwurfsprozesses in den Vordergrund und betrachtet unterschiedliche Formen der Teamarbeit, die gemeinsame Arbeit an den Artefakten und den Informationsaustausch innerhalb der Teams [Alt98].

Altmann und Weinreich [Alt98] sehen in der Kombination der beiden Sichten eine Grundvoraussetzung für die erfolgreiche, kooperative Software-Entwicklung. Die von ihnen identifizierten Anforderungen gelten für Arbeitsumgebungen, die sowohl die prozessorientierte Sicht als auch die produktorientierte Sicht berücksichtigen. Im Folgenden werden die in [Tie01], [Obe94], [Sch99] und [Alt98] angegebenen Anforderungen zusammenfassend dargestellt.

---

<sup>16</sup> Hier können exemplarisch die bekannten Werkzeuge RCS/ CVS zur Versionsverwaltung und zum Konfigurationsmanagement genannt werden.

## **1. Gemeinsame Bearbeitung von Artefakten**

- 1.1 Die Umgebung soll sowohl die synchrone als auch die asynchrone Gruppenarbeit unterstützen<sup>17</sup> [Tie01] und gemeinsame Sichten und Bearbeitungsmöglichkeiten für die Entwicklungsdokumente bieten [Tie01].
- 1.2 Die Umgebung soll sich an die gewählte Entwicklungsmethode anpassen lassen [Tie01].
- 1.3 In synchronen Systemen sind zusätzliche Möglichkeiten zur Kooperationsunterstützung und zur Koordinationsunterstützung bereit zu stellen, z.B. Konferenz-Management, Sitzungs-Kontrolle, Floor-Kontrolle usw.
- 1.4 Die Umgebung sollte kooperatives Testen in dem Maße ermöglichen, wie es zur Erreichung der Lernziele erforderlich ist.

## **2. Kommunikation und Kommunikationsunterstützung**

Durch die Umgebung sind Möglichkeiten zur zeit- und ortsunabhängigen Kommunikation zu schaffen [Obe94]. Dabei ist der Kommunikationsaufwand bei gleichzeitiger Maximierung der Kommunikation zu minimieren, vgl. [Sch99]. Die hierbei relevanten Kriterien werden getrennt für die Unterstützung der synchronen und der asynchronen Kommunikation angegeben.

### **2.1 Asynchrone Kommunikation, z.B.:**

- implizit, über die gemeinsam bearbeiteten Artefakte oder
- explizit, z.B. über E-Mail.

### **2.2 Synchrone Kommunikation:**

- Video-, Audio- oder Chat-Komponenten als Grundvoraussetzung;
- z.B. Telepointer oder Voting zur Kommunikationsunterstützung.

## **3. Kooperative und individuelle Arbeitsmodi**

Die Umgebung soll neben kooperativem Arbeiten auch individuelles Arbeiten ermöglichen. Grundvoraussetzungen hierfür sind die Bereitstellung von:

- 3.1 Öffentlichen Arbeitsbereichen.
- 3.2 Privaten Arbeitsbereichen.

---

<sup>17</sup> Hierbei ist besonders auf konsistente Zustände zu achten, vgl. [Tie01]



3.3 Darüber hinaus sollten Umgebungen zum kooperativen Software-Engineering integrierte Umgebungen sein, die eine kontinuierliche Bearbeitung der Artefakte mit jedem der integrierten Werkzeuge erlauben [Tie01].

#### **4. Nachvollziehbarkeit**

Die Anforderung der Nachvollziehbarkeit der Entwicklung resultiert im Wesentlichen aus dem speziellen Anwendungskontext des kooperativen Software-Engineering und fordert:

4.1 Gruppen-, projekt- und informationsspezifische Informationen sind bereit zu stellen [Alt98].

4.2 Informationen über die Projekthistorie sollen bereit gestellt werden [Obe94].

4.3 Änderungen an den Artefakten müssen nachvollziehbar sein.

4.4 Zuständigkeiten der Mitglieder sollen sowohl für die Vergangenheit, die Gegenwart und die Zukunft sichtbar sein.

#### **5. Gruppenwahrnehmung [Alt98]**

In CSCSE-Systemen kommt den Möglichkeiten zur aufgabenorientierten Gruppenwahrnehmung eine besondere Bedeutung zu. Kriterien in dem Zusammenhang sind:

5.1 Die aktuellen Aktivitäten der Gruppenmitglieder müssen erkennbar sein.

5.2 Die Rollen der Mitglieder sollen unmittelbar erkennbar sein.

5.3 Änderungen und Modifikationen an den gemeinsamen Artefakten sind in synchronen Systemen unmittelbar darzustellen. In asynchronen Systemen sind die Mitglieder auf geeignete Weise über Änderungen zu informieren. Die Änderungen müssen den jeweiligen Personen zugeordnet werden können.

### **4.2.4 Werkzeuge zur Unterstützung des kooperativen Software-Engineering**

Im Folgenden werden Werkzeugansätze zur Unterstützung des kooperativen Software-Engineering diskutiert<sup>18</sup>. Dabei beschränkt sich die Darstellung auf Werkzeuge, die spe-

---

<sup>18</sup> Ein Überblick über allgemeine Werkzeugansätze findet sich z.B. bei [Krc96].

ziell für den Einsatz im verteilten Software-Engineering konzipiert wurden. Zunächst werden vorrangig solche Werkzeuge vorgestellt, die eine Unterstützung der kooperativen Arbeit in mehreren Phasen des Software-Lebenszyklus bieten. Mit ICICLE (Intelligent Code Inspection Environment in a C Language Environment), CACTUS (Computer Assisted Communication Tool for Urgent Support) und dem LAZY-Consistenz-Ansatz werden aber auch Konzepte vorgestellt, die eine Unterstützung für einzelne Aspekte innerhalb des kooperativen Arbeitsprozesses bieten. Mit der Auswahl der vorgestellten Arbeiten wird kein Anspruch auf Vollständigkeit erhoben, sondern versucht, die thematische Breite innerhalb des Forschungsgebietes CSCSE abzudecken.

## Die experimentelle Software-Engineering Umgebung FLECSE

Dewan und Riedl veröffentlichten 1993 ihre Arbeiten zur Entwicklung der Experimentalumgebung FLECSE (Flexible Environment for Collaborative Software-Engineering) [Dew93]. Das mit diesen Arbeiten verfolgte Ziel war es, kooperatives Software-Engineering in möglichst allen Phasen des Software-Lebenszyklus zu unterstützen. Bei FLECSE handelt es sich um eine integrierte Umgebung, die einerseits existierende Werkzeuge, bzw. Erweiterungen existierender Werkzeuge und andererseits spezielle Implementierungen kombiniert und aufeinander abstimmt. FLECSE bietet Unterstützung für

- die verteilte, rechnergestützte Versionskontrolle,
- das verteilte, rechnergestützte Debugging,
- die verteilte, rechnergestützte Code-Inspektion und
- das verteilte, rechnergestützte Editieren und Codieren.

Zur synchronen Audiokommunikation in einem Computer Netzwerk kann TeleConf in Verbindung mit den übrigen FLECSE-Werkzeugen eingesetzt werden.

Insgesamt bietet FLECSE in erster Näherung Werkzeuge, die die Phasen Analyse, Design, Implementierung und Test unterstützen. Die Werkzeuge für die konstruktiven Tätigkeiten sind durchgehend textbasiert, weshalb sich die Unterstützung der Analyse und

Designphase im Wesentlichen auf das Verfassen und Verwalten gemeinsamer Textdokumente konzentriert. Spezielle, grafische Modellierungsverfahren werden ebenso wie die Modellierung von Rollen nicht unterstützt.

## C-SPE

Das von Grundy et al. vorgestellte System C-SPE [Gru95] erweitert die integrierte Software-Entwicklungsumgebung SPE (Smart Programming Environment) zu einem Werkzeug zum kooperativen Software-Engineering. SPE unterstützt verschiedene Sichten während der Entwicklung eines Smart<sup>19</sup>-Programms und stellt ein Konsistenzmanagement für diese verschiedenen Sichten zur Verfügung. Für alle Änderungen an den Sichten oder den Software-Komponenten generiert SPE Änderungsbeschreibungen. Diese Änderungsbeschreibungen werden von C-SPE genutzt, um andere Entwickler über Änderungen zu informieren. Hierzu werden die Änderungsbeschreibungen im Broadcast-Verfahren an alle Entwickler übertragen und in Abhängigkeit vom gewählten Arbeitsmodus in deren Entwicklungsumgebungen als geänderte, gemeinsame Version gespeichert oder dem Entwickler angezeigt. C-SPE unterstützt drei verschiedene Modi der Zusammenarbeit:

- **Asynchroner Modus:** dem optimistischen Ansatz zur Nebenläufigkeitskontrolle folgend können verschiedene Benutzer lokale Versionen eines Artefakts erstellen und ändern. Die Änderungen führen zu neuen, lokalen Versionen. Ein Benutzer ist für das Zusammenführen der verschiedenen Änderungen und das Erstellen einer neuen Version verantwortlich. Konflikte werden weitestgehend automatisch erkannt und teilweise auch behoben.
- **Synchroner und semi Synchroner Modus:** Im synchronen Modus verfolgen die beteiligten Entwickler dasselbe Ziel und benutzen dieselbe Sicht. Die Änderungsbeschreibungen werden unmittelbar ausgewertet und gleichzeitig bei allen Entwicklern dargestellt. Im semi synchronen Modus verfolgen die Entwickler unterschiedliche

---

<sup>19</sup> Smart ist eine objektorientierte Variante von Prolog [Gru95].

Ziele und benutzen ihre eigene Sicht. In dem Fall werden die Änderungsbeschreibungen anderer Entwickler in Dialog- oder Textfenstern dargestellt.

Da sowohl Quelltexte als auch Designdiagramme in C-SPE bearbeitet werden können, eignet sich die Umgebung für einen Einsatz in der Analyse-, Design- und Implementierungsphase.

C-SPE erfüllt in hohem Maße die Anforderungen an die gemeinsame Bearbeitung von Artefakten und die Kommunikation. Nachteilig wirkt sich die Beschränkung auf Snart und die fehlenden Möglichkeiten zur Kommunikationsunterstützung aus. C-SPE bietet keinen individuellen Arbeitsmodus. Die Nachvollziehbarkeit wird durch das Konzept der Änderungsbeschreibungen insgesamt sehr gut unterstützt. Die Gruppenwahrnehmung wird nur durch die Behandlung der Änderungen an den Artefakten unterstützt.

## Koordination kooperativer Arbeit mit CSDE [Alt98]

Die kooperative Software-Umgebung CSDE (Cooperative Software Development Environment) wurde an der Universität Linz entwickelt. Dabei handelt es sich nicht um eine integrierte Software-Entwicklungsumgebung, sondern vorrangig um ein System, das den Zugriff auf die im jeweiligen Arbeitsablauf verwendeten Artefakte koordiniert und insgesamt eine Unterstützung der Koordination von kooperativer Arbeit innerhalb des gesamten Software-Entwicklungsprozesses bietet. Während ihrer Arbeit mit CSDE folgen die Entwickler einer festgelegten Agenda und werden dabei von einem Kooperationsassistenten unterstützt.

Ein Artefakt kann immer nur exklusiv durch einen Benutzer bearbeitet werden, weshalb synchrone Kooperation nicht möglich ist. Insgesamt wird ein hohes Maß an Gruppenbewusstsein vermittelt, indem die ausgeführten Aktivitäten der anderen Gruppenmitglieder zum aktuellen Zeitpunkt angezeigt werden. Durch die Bereitstellung von History-Informationen wird darüber hinaus ein relativ hoher Grad an Nachvollziehbarkeit erreicht. Innerhalb eines Benutzerinformationsfensters bietet CSDE einfache Mittel zur direkten Kommunikation.

Damit liegen die Stärken von CSDE in der Koordinationsunterstützung, der Gruppenwahrnehmung und der Nachvollziehbarkeit. Durch die eingeschränkten Möglichkeiten zur synchronen Kooperation wird eines der wesentlichen Kriterien zur gemeinsamen Bearbeitung von Artefakten nicht erfüllt. Ebenso fehlen Basisfunktionen zur Unterstützung kooperativer und individueller Arbeitsmodi. CSDE unterstützt keine Rollen, bietet aber Verlaufsinformationen über das laufende Projekt und die Artefakte.

## Unterstützung synchroner Kooperation mit TUKAN [Sch99], [TUK99]

TUKAN wurde im Rahmen einer Diplomarbeit an der TU Darmstadt in Kooperation mit dem Fraunhofer Institut für integrierte Publikations- und Informationssysteme (IPSI) entwickelt. Das System unterstützt die Software-Entwickler im Wesentlichen während der Implementierungsphase, indem alle Aktionen der einzelnen Software-Entwickler von TUKAN "beobachtet" werden. Darüber hinaus "bemerkt" TUKAN, falls zwei oder mehr Software-Entwickler in dem selben Teil des Artefakts arbeiten. In diesem Fall wird dem Benutzer nicht nur angezeigt, dass andere Entwickler in dem gleichen Bereich arbeiten, sondern mit einer Wettermetapher wird auch die Höhe des Konfliktpotentials quantisiert. Die Informationen über die Aktionen anderer Benutzer im gleichen Bereich hängen von der Beziehung der beteiligten Artefakte zueinander ab. Ein spezieller Klassenbrowser, der Pair Programming Browser, ermöglicht synchrone Kooperation. Zur synchronen Kommunikation ist ein Chat-Werkzeug implementiert. Die asynchrone Kooperation wird durch History-Informationen unterstützt. Entwicklern die ein Projekt für längere Zeit verlassen haben, werden bei ihrer Rückkehr Informationen über potenziell interessante Bereiche angeboten.

TUKAN bietet nahezu umfassende Unterstützung für die Anforderungen an die gemeinsame Bearbeitung der Artefakte und die Kommunikation. Darüber hinaus erfüllt TUKAN alle Kriterien die zur Unterstützung kooperativer und individueller Entwicklungspfade gefordert werden, sowie die Anforderungen an die Nachvollziehbarkeit. Es werden jedoch keine Rollen unterstützt.

## Das MACS-Rahmenwerk

MACS (Modular Advanced Collaboration System) [Bran99] bildet ein Rahmenwerk zum Aufbau von Kollaborationssystemen und wurde am Institut für Betriebssysteme und Rechnerverbund (IBR) der TU Braunschweig entwickelt. Es zielt auf die Implementierung einer applikationsübergreifenden Kontrollinstanz, sowie auf eine einfache, möglichst intuitive Steuerung des Systems durch den Anwender. Schwerpunkte der Entwicklung liegen auf der Modularität und der Portabilität einerseits und der Sitzungs- und Floor-Kontrolle andererseits.

Das MACS-Rahmenwerk trennt die Sitzungs- und die Systemkontrolle, so dass ein Benutzer gleichzeitig in mehreren Sitzungen teilnehmen kann. Die Konferenzsteuerung erfolgt in einem virtuellen Konferenzraum mit Hilfe grafischer Symbole. Teilnehmer werden anhand von Bildern dargestellt: Über zusätzliche grafische Elemente erhalten die übrigen Teilnehmer Auskunft über die Ressourcen (z.B. ob der Teilnehmer Audio-daten senden kann) und den Status jedes Teilnehmers (z.B. ob er momentan anwesend ist). Innerhalb des virtuellen Sitzungsraums sind hierarchische Konferenzen möglich, z.B. die private Kommunikation zwischen zwei Teilnehmern. Zur Floor-Kontrolle sind in [Bran99] sowohl implizite als auch explizite Strategien angekündigt. Die Floor-Kontrolle ist dabei mit Ressourcen und nicht mit Applikationen verknüpft.

Zu den in MACS einsetzbaren Applikationen, gehören das IBR-Whiteboard und der Mehrbenutzereditor TelSEE (Tele-Software-Engineering-Editor). Zusammen bieten diese Werkzeuge eine Unterstützung für verteilte Software-Engineering-Teams. Dabei werden im IBR-Whiteboard die Entwurfsmodelle dargestellt und in TelSEE die gemeinsamen Textdokumente bearbeitet.

Das IBR-Whiteboard [Bran99] bietet vollwertige Zeichenoperationen, ein Telepointer-Konzept und den Zugriff auf die Floor-Kontrolle. Das IBR-Whiteboard-Konzept benutzt relaxiertes WYSIWIS mit der Möglichkeit, bei gleicher Sicht verschiedene Bereiche eines Dokuments darzustellen. Der Vorteil der Individualität geht dabei zu Lasten der Gruppenwahrnehmung.

Der Mehrbenutzereditor TelSEE [Mah99] erlaubt das gleichzeitige Bearbeiten eines Dokuments sowohl in synchroner als auch in asynchroner Gruppenarbeit und erkennt selbständig eventuelle Konsistenzkonflikte. Diese müssen jedoch explizit, z.B. in einer Videokonferenz, behoben werden.

### ICICLE: Eine Groupware zur Code-Inspektion [Bro90]

Die von Brothers et al. [Bro90] 1990 veröffentlichten Arbeiten gehören zu den ersten Publikationen einer speziell zur Unterstützung des kooperativen Software-Engineering konzipierten Groupware. ICICLE wurde bei Bellcore als Forschungsprototyp speziell für den Prozess der Code-Inspektion entwickelt. Den Teilnehmern einer Sitzung werden spezielle Rollen zugewiesen: der Moderator ist für den Verlauf der Sitzung verantwortlich, ein Leser wählt den zu betrachtenden Code-Ausschnitt und liest diesen den Inspektoren vor. Die Inspektoren diskutieren den Code und stimmen über Änderungsvorschläge und Kommentare ab. Der Schreiber fügt die Kommentare als Annotation in einem speziellen Kommentarfenster dem Code hinzu. Der gemeinsame Kontext wird über öffentliche Arbeitsbereiche realisiert; insgesamt wird durch die Möglichkeit privater Arbeitsbereiche jedoch der Ansatz des relaxierten WYSIWIS-Konzept verfolgt.

ICICLE wurde für Gruppen konzipiert, deren Teilnehmer sich im gleichen Raum befinden und bietet Basisfunktionalitäten zur synchronen Zusammenarbeit sowie gemeinsame Sichten auf die Dokumente. Möglichkeiten zur synchronen Kommunikation für räumlich verteilte Gruppen, wie z.B. Audio, Video und Chat oder Kommunikationsunterstützungen durch Telepointer oder Voting-Werkzeuge sind in ICICLE nicht vorgesehen, wurden von den Autoren aber in ihrem Originalaufsatz im Rahmen der Entwicklung einer Tele-ICICLE-Version in Aussicht gestellt.

### CACTUS: ein Entscheidungsunterstützungsmodell für das Software-Engineering [Tak93]

Takeda et al. stellen mit CACTUS ein Werkzeug vor, das den Prozess der Problembearbeitung im Software-Engineering unterstützen soll. Die Arbeiten bauen auf einer Ana-

lyse des Software-Entwicklungsprozesses auf, die insbesondere die Kommunikation im Problemfall untersuchte. Auf der Basis dieser Untersuchungen wurde ein Modell entwickelt, das jedem der beteiligten Entwickler eine oder mehrere von drei verschiedenen Rollen zuweist: Problemfinder, Entscheider oder Problembeheber. Weiter wird jede Rolle durch drei Attribute charakterisiert:

- das Attribut Charge kennzeichnet den Arbeitsbereich des Entwicklers;
- das Attribut Knowledge beschreibt das Systemwissen des Entwicklers unabhängig von der Ausprägung des Attributes Charge;
- das Attribut Competence legt fest welche Entscheidungen der Entwickler treffen darf.

Darüber hinaus sind durch das Modell verschiedene Kommunikationsformen definiert:

- Standard Kommunikation (planmäßige Meetings, z.B. für Reviews);
- non Standard Kommunikation (z.B. im Problemfall oder außerplanmäßige Kommunikation jeder Art).

Aufbauend auf diesem Modell wurde das Simulationssystem CACTUS entwickelt. Mit diesem System können im Problemfall Vorgehensweisen zur Problembhebung generiert werden, indem ein Entscheider ermittelt wird und diesem für das jeweilige Problem zusätzliche Personen mit entsprechenden Ausprägungen der Attribute Knowledge und Chart zugewiesen werden.

## Der LAZY-Consistenz-Ansatz zur Konsistenzerhaltung [Nar92]

Der LAZY-Consistenz-Ansatz [Nar92] nach Narayanaswamy und Goldman beschreibt einen alternativen Ansatz zu den bekannten Verfahren zur Konsistenzerhaltung, indem geplante Änderungen bereits vor ihrer Durchführung bekannt gemacht werden. Alle davon abhängigen Programmierer haben dann explizit die Möglichkeit, sich zu den geplanten Änderungen zu äußern.



Der Änderungsprozess nach dem LAZY-Consistenz-Ansatz sieht vor, dass ein Programmierer zunächst sein Ziel (z.B. die Behebung eines Fehlers) definiert und im nächsten Schritt die zur Erreichung des Zieles nötigen Änderungen (engl. Proposed Changes, im Folgenden kurz PCs) beschreibt. Diese PCs werden anschließend an alle von diesen Änderungen abhängigen Programmierern geschickt (die Kenntnis über diesen Personenkreis ist im so genannten Transaction Model hinterlegt). Die Programmierer können nun auf die geplanten Änderungen reagieren indem sie entweder:

- alle PCs uneingeschränkt akzeptieren, was zu einer neuen Software-Version führt;
- alle PCs uneingeschränkt ablehnen, wodurch die alte Software-Version erhalten bleibt;
- oder eine Überarbeitung der PCs einleiten.

Die Abarbeitung eines definierten Ziels mit dem beschriebenen Verfahren bezeichnen Narayanaswamy und Goldman als Intra Step. Daneben sind auch Änderungen, die verschiedenen Zielen folgen, möglich. In diesem Fall sind in der Regel mehrere Entwickler an der Initiierung eines so genannten Inter Step beteiligt. Inter Steps erfordern üblicherweise eine engere Kommunikation der Programmierer und eine Aufteilung des Intra Steps in eine Folge von Inter Steps. Der Fall, in dem die Programmierer ihre geplanten Änderungen verschmelzen, führt zu einer Rückgewinnung des konsistenten Zustandes in einem einzigen Änderungsschritt und wird von Narayanaswamy und Goldman als LAZY-Consistenz bezeichnet.

#### 4.2.5 Zusammenfassung

Der momentane Stand der Technik bei Werkzeugen zum kooperativen Software-Engineering wird von Werkzeugen zum gemeinsamen Editieren und Debuggen dominiert [Alt98]. Entsprechend ist das gemeinsame Entwickeln, Implementieren und Testen bereits gut gelöst. Von den untersuchten Ansätzen bieten die Werkzeuge FLECSE, C-SPE, ICICLE und TUKAN eine Chat-Funktion als integralen Bestandteil des Werkzeugs. FLECSE und TUKAN stellen Editoren zur synchronen Bearbeitung des Quellcodes zur Verfügung. ICICLE bietet gemeinsame Ansicht durch Terminal-Sessions.

CSDE unterstützt die Gruppenkommunikation nur in der Form, dass die dafür notwendigen Informationen für Telefon und E-Mail erfasst und dargestellt werden. Nur FLECSE unterstützt computergestützte Audiokonferenzen.

### **4.3 Aktueller Stand der Einführung verteilter, kooperativer Arbeitsformen in die Software-Engineering-Ausbildung**

Die Software-Engineering-Ausbildung an den Hochschulen soll das für die ingenieurmäßige Erstellung von Software notwendige Fachwissen vermitteln [Kne93]. Dazu integriert die Software-Engineering Grundausbildung in aller Regel Projektpraktika, die sich in der Mehrzahl am Software-Lebenszyklus orientieren. Die einzelnen Praktikumskonzepte unterscheiden sich zumeist im Hinblick auf die eingesetzten Methoden und Programmiersprachen, den Grad der Werkzeugunterstützung, die Aufgabenstellung und die Verankerung in der Studienordnung [Sch97], folgen aber in der Regel denselben Lernzielen, wie sie z.B. in [Lew01], [Kne93], [Wer98], [Raa97], [Rys99] und [Sch95] formuliert sind:

- **Lernziel 1: Vermittlung technischer Kompetenz** durch Umsetzung des in der Vorlesung vermittelten abstrakten Wissens in nutzbares Wissen.
- **Lernziel 2: Erarbeitung sozialer Kompetenz** durch das Kennenlernen der Arbeit im Team mit selbstbestimmter Einflussnahme auf die Vorgänge der Arbeitsteilung, Präzisierung von Aufgabenstellungen und Übernahme der Verantwortung für bestimmte Teile der Entwicklung.

Die einzelnen Lernziele werden in der Regel individuell weiter präzisiert, je nach Maßgabe des Praktikum-Konzeptes. Die in dem Zusammenhang oftmals implizit formulierte Zielsetzung der Praxisorientierung wird durch das sich ändernde Umfeld des Software-Ingenieurs zunehmend zu einem eigenen Lernziel expliziert:

- **Lernziel 3: Anwendung zukunftsweisender Arbeitsformen** in einem realitätsnahen Szenario durch den Einsatz von Groupware.

Grundvoraussetzungen zur Erfüllung von Lernziel 3 sind eine ausreichend leistungsfähige Infrastruktur, ein geeigneter organisatorischer Rahmen und eine aufgabenangemessene Groupware. Sowohl der Einsatz existierender Werkzeuge als auch die Entwicklung neuer Ansätze muss dabei immer berücksichtigen, dass die Lernziele 1 und 2 von den Aktivitäten zur Erreichung des dritten Lernziels unberührt bleiben.

Im Folgenden wird ein Überblick über den Stand der Technik bei der Entwicklung und Umsetzung von Konzepten und Werkzeugen zum Einsatz in Software-Engineering Praktika mit kooperativen Arbeitsformen und zum Teil virtuellen Teams gegeben. Die Ergebnisse der in dem Zusammenhang untersuchten Arbeiten und Projekte werden anschließend zusammenfassend bewertet. Als eigenes Ergebnis dieser Untersuchungen wird am Ende des Kapitels ein Anforderungskatalog für Groupware zum Einsatz in einem kooperativen Software-Engineering-Praktikum angegeben.

#### 4.3.1 Studien im Rahmen des JTAP-Projektes

In Großbritannien startete 1995 das "JISC Technology Applications Programme (JTAP)" mit Förderung durch die für Bildung und Weiterbildung zuständigen Ministerien [Bre98a]. Die Organisation und Koordination des Programms erfolgte über das "Joint Information Systems Committee" (JISC), das als Projektträger mehr als 100 JTAP-Projekte finanzierte. Eines der Teilprojekte in JTAP war das Projekt "Developing a Virtual Community for Student Groupwork", das gemeinsam von drei Universitäten durchgeführt wurde:

- Keele University (KU)- Department of Computer Science;
- University of Durham (UoD)- Department of Computer Science;
- University of Manchester (UMIST) - Department of Computation.

Im Rahmen des Projektes wurden mehrere Studien im Rahmen von Software-Engineering Praktika durchgeführt, deren Ziel es war,

- eine funktionierende Umgebung für den Einsatz von verteilter Gruppenarbeit in Studentenprojekten aufzubauen,
- einen Beitrag zur Planung für spätere und größer angelegte Studentenprojekte zu liefern und
- den Studierenden erste Einblicke in die verteilte Gruppenarbeit zu geben.

Im Folgenden werden drei kleinere Studien beschrieben die im Rahmen des JTAP-Projektes durchgeführt wurden.

### JTAP-Studie 1: Einsatz gemeinsamer Informationsräume [Mac97]

An der UMIST wurde in einer Veranstaltung "CSCW and Software-Engineering" BSCW als gemeinsamer Informationsraum eingesetzt. Das Ziel der Studie, an der 15 Studierende des letzten Studienjahres teilgenommen haben, war es zu evaluieren,

- ob ein gemeinsamer Informationsraum die richtige Groupware zur Unterstützung der Veranstaltung ist,
- ob die Studierenden mit der Benutzung vertraut waren und
- in welchem Maß die Studierenden von der Groupware Gebrauch gemacht haben.

BSCW wurde für die folgenden Aufgaben eingesetzt:

- **Gemeinsames Repository** für Vorlesungsunterlagen, Zeitpläne und Jobangebote aus der Industrie.
- **Gemeinsame Plattform für Online-Diskussionen.** Die Studierenden sollten als Vorbereitung auf die nächste Vorlesung jeweils eine Veröffentlichung lesen und sich mit vier oder fünf von den Dozenten gestellten Fragen auseinandersetzen. Während der Vorlesung wurden die Studierenden in Kleingruppen eingeteilt und jede der Gruppen war für die Beantwortung einer Frage zuständig. Im Anschluss an die Vorlesung hatte ein Gruppenmitglied die Aufgabe die Gruppenantwort in dem gemeinsamen BSCW-Arbeitsbereich zu veröffentlichen. In der Folge hatte sowohl der Dozent als auch die anderen Studierenden die Möglichkeit, diese Antwort zu kommentieren bzw. erneut auf Kommentare zu antworten.

- **Gemeinsames Kommunikationsmittel.** Den Teilnehmern wurde eine E-Mail-Adresse zugeteilt, der gemeinsame Arbeitsbereich wurde aber durch Benutzerkennung und Passwort geschützt. Somit war E-Mail-Verkehr nur zwischen den Teilnehmern möglich.

Die Evaluation ergab, dass BSCW grundsätzlich zur Unterstützung der Veranstaltung geeignet ist. Es wurde jedoch beobachtet, dass die Bindung der einzelnen Gruppenmitglieder recht lose war. Die Studierenden waren einerseits mit der Vielzahl der von BSCW angebotenen Funktionen überfordert, die Bedienung der letztendlich benutzten Funktionen war jedoch problemlos. Die Analyse der vom BSCW-System generierten Log Files ergab, dass lediglich die Hälfte der Studierenden das BSCW-System regelmäßig benutzte. Sobald die Studierenden eigene Beiträge veröffentlichten, nahm aber auch der Grad der Nutzung zu.

## JTAP-Studie 2: Einsatz von Konferenzsystemen [Bre98]

Im Rahmen dreier Experimente sollte eine aus frei verfügbaren Komponenten zusammengestellte Groupware-Umgebung<sup>20</sup> dahingehend untersucht werden, inwiefern sie sich zur Unterstützung der synchronen Kommunikation im Rahmen der Diskussion von Dokumenten eignet. Die Groupware bestand vorrangig aus CUSeeMe als Videokonferenz- und Chat-System sowie einem gemeinsamem Arbeitsbereich (Whiteboard). Insgesamt nahmen 10 Gruppen zu je drei Studierenden an der Studie teil. Die Studierenden waren räumlich verteilt, wobei insgesamt 12 Studierende über die drei Hochschulstandorte verteilt waren. Die übrigen Studierenden waren an der UMIST über verschiedene Büros verteilt.

Im ersten Experiment wurde vorrangig Systemanalyse und Brainstorming betrieben, indem ein existierendes manuelles Rechnungssystem anhand gegebener Unterlagen untersucht wurde und erste Ideen zur Realisierung einer Online-Version diskutiert wurden.

---

<sup>20</sup> Die Auswahl der einzelnen Komponenten ist in [Gum97] beschrieben.

Das zweite Experiment sah die Diskussion von Entwurfsdokumenten (Kontext-Diagramm und Entity Relationship Modell) vor. Im dritten Experiment wurde anstelle des Whiteboard ein gemeinsamer Editor (MS-Word) integriert um Annotation zu gegebenen Spezifikationen im Rahmen einer Machbarkeitsstudie zu ermöglichen.

Die Studierenden erhielten eine Einweisung von 20 bis 45 Minuten. Die Durchführung des Experiments inklusive der anschließenden Befragung dauerte im Durchschnitt 85 Minuten.

Die Evaluation erfolgte mit Hilfe verschiedener Fragebögen. Neben personenbezogenen Daten und Fragen zur Erfahrungen der Studierenden im Umgang mit Rechnern und dem Einsatz von Gruppenarbeit erhob ein Fragebogen den Grad der Aufgabenangemessenheit und Bedienbarkeit der eingesetzten Werkzeuge. Die Ergebnisse der Studie sind im Detail in [Mac97], [Bre98] und [Bre98a] dokumentiert. Im Folgenden werden die wichtigsten Ergebnisse vorgestellt und diskutiert.

### **Ergebnisse der JTAP-Studie 2**

Im Rahmen der Befragung wurden die Studenten unter anderem um eine Bewertung der benutzten Werkzeuge (Video, Audio, Chat, Whiteboard) im Hinblick auf eine Unterstützung spezieller Aufgaben (Diskussionen, gemeinsame Diagramme, Präsentationen, verteilte Dokumente sowie Codieren, Testen und Debuggen) innerhalb der einzelnen Phasen des Software-Lebenslaufs gebeten. Die Art der Fragestellung impliziert, dass die Studierenden nach den Experimenten über genügend Erfahrung verfügen, um auch das zukünftige Potenzial der einzelnen Werkzeuge unabhängig von der aktuellen Werkzeugqualität einzuschätzen. Ein Ranking der am häufigsten genannten Werkzeuge zur Unterstützung der untersuchten Aufgaben zeigt Tabelle 3.

<b>Aufgabe</b>	<b>Rang 1</b>	<b>Rang 2</b>	<b>Rang 3</b>	<b>Rang 4</b>
Diskussionen:	Audio (91%)	Video (48%)	Chat (40%)	Whiteboard (3%)
Gemeinsame Diagramme:	Whiteboard (74%)	Chat (3%)	Video (0%)	Audio (0%)
Präsentationen:	Video (40%)	Chat (14%)	Audio (8%)	Whiteboard (2%)
Verteilte Dokumente:	Whiteboard (19%)	Chat (19%)	Video (5%)	Audio (0%)
Codieren, Testen, Debuggen:	Chat (19%)	Whiteboard (0%/38%)	Audio (0%/41%)	Video (0%/45%)

**Tabelle 3:** Ergebnisse der JTAP-Studie 2

Eine nähere Betrachtung des ersten Ranges zeigt, dass sich eindeutige Aussagen lediglich für die Aufgaben Diskussion und gemeinsame Erstellung von Diagrammen treffen lassen. Zur Unterstützung von Diskussionen wird Audio- und Video-Werkzeugen, im Falle von Audio mit 91% sogar sehr deutlich, der Vorzug gegenüber Chat-Werkzeugen gegeben. Hier dominieren offensichtlich die Wünsche nach sozialen Kontextinformationen und persönlichen Kontakten.

Im Falle der gemeinsamen Erstellung von Diagrammen hatte erwartungsgemäß das Whiteboard die meisten Nennungen. In Relation zum Audio-Werkzeug, das im Falle der Bewertung für den Nutzen bei Diskussionen mit 91% dominierte, fiel das Ergebnis für das Whiteboard mit 74% jedoch deutlich schlechter aus. Hier kann davon ausgegangen werden, dass die Studierenden im Einzelfall nicht mit allen Konzepten des CSCW vertraut sind und insbesondere nicht das Potential der Arbeiten zur Gruppenwahrnehmung und Nebenläufigkeitskontrolle im Hinblick auf zukünftige Werkzeugentwicklungen in ausreichendem Maße einschätzen konnten. Dies gilt auch für die Aufgaben im Zusammenhang mit dem gemeinsamen Editieren und Codieren. Hier lassen die Bewertungen keine Schlussfolgerungen auf ein oder zwei spezielle Werkzeuge zu. Vielmehr kann auch hier davon ausgegangen werden, dass auf Grund zu geringer CSCW-Kenntnisse nicht von aktuellen Implementierungen auf zukünftige Implementierungen geschlossen werden konnte.

Im Rahmen der Befragung haben die Studierenden vorrangig drei Schwachstellen der eingesetzten Groupware-Umgebung identifiziert:

- Fehlen eines Gruppeditors  
Studentenkommentar: *fehlende Möglichkeit der Textbearbeitung beim Whiteboard;*
- Fehlende Koordinationsunterstützung, bzw. fehlende Floor-Kontrolle  
Studentenkommentar: *alle Teilnehmer sprechen durcheinander (Audio), jeder kann bearbeiten und löschen, ohne die Anderen zu informieren (Whiteboard);*
- Unzureichende Leistung der Endsysteme und der Protokoll-Implementationen  
Studentenkommentar: *schlechte Qualität (Audio), Aussetzer (Audio), Zeitverzögerung (Audio), fehlende Synchronisation zwischen Audio und Video (System), Videofenster zu klein (Video).*

Die Studie belegt einerseits das Potenzial der CSCW-Technologie für einen Einsatz im Software-Engineering-Praktikum. Andererseits zeigt sie deutlich, dass frei verfügbare Komponenten nur bedingt als exklusive Mittel zur Kommunikation, Kooperation und Koordination geeignet sind. Hierbei ist zu beachten, dass alle Studien auf einmaligen Experimenten basieren und nicht auf Feldstudien unter annähernd realistischen Bedingungen.

### JTAP-Studie 3: Einsatz der SEGWorld Umgebung

Aufbauend auf den Erfahrungen der oben beschriebenen JTAP-Studien wurde an der UoD die Infrastruktur SEGWorld für Aufgaben der Projektorganisation und -koordination in einem verteilten, kooperativen Software-Engineering-Praktikum konzipiert und implementiert. Einen ersten Überblick über SEGWorld liefert [Dru97]. In [Dru00] und [Dru99] finden sich Erfahrungsberichte, und [Dru01] beschreibt eine erste Evaluation von SEGWorld.

Ausgangspunkt der Arbeiten ist das Software-Engineering Group (SEG) Projekt an der UoD. Im Rahmen dieser seit 1984 angebotenen Praxisprojekte folgen die Studierenden einem an das Wasserfall-Modell angelegte Phasenmodell. Zur Unterstützung des Prozesses wurde die auf dem BSCW-System basierende Umgebung SEGWorld konzipiert. SEGWorld besteht im Wesentlichen aus einem Repository mit Unterrichtsmaterialien für das SEG-Projekt und einem öffentlichen Arbeitsbereich, in dem sowohl Studierende



als auch das betreuende Personal verschiedene Software-Werkzeuge und Dienste (z.B. Nachrichtendienste) nutzen können. Darüber hinaus stehen private Arbeitsbereiche zur Verfügung, in denen die einzelnen Gruppen projektrelevante Daten ablegen können.

SEGWorld wurde erstmals im akademischen Jahr 1997/1998 in der Veranstaltung Software-Engineering I<sup>21</sup> eingesetzt. Neben dem regelmäßigen Einsatz von SEGWorld wurde an zwei Terminen zusätzlich ein Desktop-Video-Konferenzsystem (DVK) zur synchronen Kommunikation eingesetzt. Die DVK-Konfiguration umfasste CUSeeMe und bestand im Wesentlichen aus Audio-, Video- und Chat-Werkzeugen sowie einem Whiteboard. Nach einer zweistündigen Einführung wurden zwei Sitzungen zu jeweils zwei Stunden durchgeführt.

Die Evaluation der SEGWorld erfolgt mit Hilfe von Fragebögen und den von BSCW generierten Protokolldateien<sup>22</sup>. Im Wesentlichen sollten mit der Evaluation drei Hypothesen überprüft werden:

- **Hypothese 1:** Die Einführung eines asynchronen, gemeinsamen Arbeitsbereichs unterstützt die Studierenden bei der Organisation und Koordination ihrer Arbeit in einem Software-Engineering-Praktikum;
- **Hypothese 2:** Der Grad der Nutzung des gemeinsamen Arbeitsbereichs, nimmt mit dem Projektfortschritt zu;
- **Hypothese 3:** Asynchrone Kommunikation spielt in einem kooperativen Software-Engineering-Praktikum eine wichtigere Rolle als synchrone Kommunikation.

---

<sup>21</sup> Die Veranstaltung adressiert Studierende im zweiten Studienjahr und besteht aus einer Vorlesung und einem parallel dazu angebotenen Praktikum. Das Praktikum läuft über 16 Wochen.

<sup>22</sup> Die erste Evaluation erfolgte 1997/1998 mit 72 Studierenden, von denen 57 Studierende einen Fragebogen ausfüllten. Eine weiterer Durchgang fand 1998/1999 mit 84 Studierenden statt, von denen 66 Studierende einen Fragebogen ausfüllten.

Die Auswertung der Fragebögen ergab, dass sowohl die Studierenden als auch das betreuende Personal die Umgebung als hilfreich und nützlich für die Organisation und Koordination ihrer Arbeit bewerteten. Allerdings kann nicht generell bestätigt werden, dass der Grad der Nutzung mit dem Projektfortschritt zunimmt, die Hypothese 2 wurde also nicht eindeutig bestätigt. Detaillierte Diskussion der Hypothesen 1 und 2 findet sich in [Dru01] und [Dru99].

Bei der Diskussion der Hypothese 3 muss im vorliegenden Fall die unterschiedliche Einsatzdauer berücksichtigt werden. Ein direkter Vergleich ist daher nicht möglich. Dennoch lassen sich einige Tendenzen herausarbeiten. Grundsätzlich wurde die Notwendigkeit beider Formen der Zusammenarbeit im Rahmen des verteilten, kooperativen Software-Engineering von den Studierenden in etwa gleich bewertet. Darüber hinaus war DVK bei den Studierenden zwar beliebter als SEGWorld, allerdings war die Zufriedenheit mit dem erreichten Endergebnis im Falle der DVK-Sitzungen deutlich schlechter. Drummond und Boldyreff [Dru00] führen dies auf die fehlende Funktionalität zur Unterstützung von Software-Engineering Aufgaben und die Unausgereiftheit der Technologie zurück. Ein beobachteter Nebeneffekt des Einsatzes von DVK war die Steigerung der Gruppenzusammengehörigkeit.

#### 4.3.2 Infrastrukturuntersuchungen der University of Arizona

Die University of Arizona (UoA) bietet seit 1972 Vorlesungen via eines Video-Campus-Programms an. 1985 trat sie der National Technology University (NTU) bei, die bereits 1984 als erste akkreditierte "virtuelle" Universität gegründet wurde [Nat02]. Heute bietet die NTU 19 Studiengänge an, deren Veranstaltungen von einem Konsortium aus 52 Universitäten bereit gestellt werden. Die Veranstaltungen werden per Satellit, Internet, Videokassette oder CD-ROM verbreitet. Die Studierenden der NTU sind in der Regel Beschäftigte in Unternehmen, die über die erforderliche Ausstattung (NTU-Satellitenempfang) verfügen und ihren Mitarbeitern die Teilnahme am NTU-Programm von ihrem Arbeitsplatz aus ermöglichen.

Die Veranstaltung Software-Engineering ist eine der Veranstaltungen, die die UoA der NTU anbietet. Die Inhalte der Vorlesung orientieren sich am Software-Lebenszyklus. Ein weiterer Bestandteil der Veranstaltung ist ein Software-Engineering-Praktikum. Die Teilnehmer des Praktikums sind sowohl Studierende der UoA als auch der NTU und unterscheiden sich zum Teil deutlich hinsichtlich ihrer akademischen Voraussetzungen und ihrer Berufserfahrung. Die meisten der teilnehmenden Studierenden der UoA sind aus den Ingenieurwissenschaften. Teilnehmende Studierende, die an der NTU eingeschrieben sind, stammen aus weitaus unterschiedlicheren Disziplinen, diese reichen von den Ingenieurwissenschaften bis hin zum Gesundheitswesen.

Das Ziel des Experiments war es, einen Beitrag zur Konzeption einer Kollaborationsumgebung zu liefern. Hierzu wurde die zur Verfügung stehende Technologie im Hinblick auf einen Einsatz in einem Software-Engineering-Praktikum untersucht. Zunächst wurden existierende Werkzeuge zum Application Sharing, Audio-, Video- und Chat-Konferenzwerkzeuge sowie Werkzeuge zum File Transfer analysiert und schließlich Microsoft NetMeeting [Mic02] in der Version 2.1 ausgewählt<sup>23</sup>.

Das Experiment benutzt zwei verschiedene Szenarien. Im ersten Szenario befinden sich zwei Clients im LAN (Local Area Network) der UoA und sind über das Internet mit einem Server in San Francisco verbunden. Das zweite Szenario unterscheidet sich dahingehend, dass einer der beiden Clients via Modem (28,8 K) über einen Service Provider mit dem Internet verbunden ist.

Ein Teil des Experiments wurde am Tag und bewusst zu Zeiten besonders hohen Netzwerkaufkommens durchgeführt. Der zweite Teil des Experiments wurde nachts zu Zeiten besonders niedrigen Netzwerkaufkommens durchgeführt. Die Ergebnisse der durchgeführten Messungen sind in [Sar99] dokumentiert. Auf Grund dieser Untersuchungsergebnisse kamen Sarjoughian und Zeigler zu dem Schluss, dass sich frei verfügbare Software und die existierenden Hardware- und Netzwerktechnologien nicht für den Ein-

---

<sup>23</sup> Für nähere Erläuterungen zur Auswahl der Hard- und Software und für Details zur Konfiguration siehe [Sar99].

satz in einem Software-Engineering-Praktikum eignen. Dabei wurden im Wesentlichen die untersuchten Anwendungen zum entscheidenden Faktor, sowohl im Hinblick auf technische Kriterien, wie Datenverlust und CPU-Auslastung, als auch im Hinblick auf Aspekte der Gruppenarbeit, wie die Anzahl der gleichzeitig darstellbaren Videofenster. Als Konsequenz wurde der Einsatz eines kommerziellen Videokonferenzsystems erwogen.

#### 4.3.3 Gemeinsames Pilotprojekt der TU-Darmstadt und der GMD

1997 startete die TU-Darmstadt (TUD) gemeinsam mit der GMD ein Pilotprojekt, um den Studierenden im Rahmen eines Software-Engineering-Praktikums Kooperationsdienste zur Verfügung zu stellen [Schr98]. Dabei kamen im Wesentlichen asynchrone Systeme zum Einsatz. Über das WWW wurde das Unterrichtsmaterial bereit gestellt und spezielle Projekt-Homepages eingerichtet. Das vorrangige Kommunikationsmittel war E-Mail, darüber hinaus wurde zur Verbreitung allgemeiner Informationen eine Newsgroup eingerichtet. Auf einem Server der TUD wurde ein gemeinsames Repository zur Verwaltung der Projektdokumente eingerichtet. Der Dateizugriff erfolgte über FTP (File Transfer Protocol).

Eine Evaluation erfolgte 1998 in Form von Interviews und der Auswertung von Sitzungsprotokollen. Eine detaillierte Diskussion der Evaluationsergebnisse findet sich in [Schr98].

Für die Zukunft soll aufbauend auf den bisherigen Erfahrungen eine Umgebung konzipiert und implementiert werden, die neben asynchroner Kooperation auch synchrone Kooperation ermöglicht. Schroeder et al. beschreiben als Hauptanforderungen an synchrone Systeme, dass die Werkzeuge preiswert und für alle Studierenden verfügbar sein müssen. Die Systeme sollen schnell und robust sein. Darüber hinaus wird eine einfache Bedienbarkeit und ein geringer Einarbeitungsaufwand gefordert. Erste Konzepte sehen hierzu den Einsatz eines CASE-Werkzeuges zum kollaborationstransparenten Docu-

ment-Sharing vor sowie Audiokonferenzen über das Internet zur synchronen Kommunikation. Als gemeinsamer Arbeitsbereich soll BSCW eingesetzt werden.

#### 4.3.4 Einsatz von Lotus Notes in einem Projektseminar der Universität Bern

An der Universität Bern (UB) wurde im Sommersemester 2000, innerhalb eines Software-Engineering-Praktikums, die Groupware Lotus Notes zur Unterstützung der Kooperation bei der Dokumentenerstellung im Team eingesetzt [Röt01].

Nach den gewonnenen Erfahrungen der UB verschaffte der Groupware-Einsatz den Studierenden folgende Vorteile:

- der Informationsaustausch wurde erleichtert;
- die Dokumentenerstellung wurde durch die Möglichkeiten der verteilten und asynchronen Bearbeitung vereinfacht;
- das Dokumentenmanagement wurde durch verschiedene Möglichkeiten der Kategorisierung und die Zuweisung von Verantwortlichkeiten bei der Dokumentenerstellung vereinfacht.

Kreative Gruppenarbeiten wie das Sammeln, Weiterentwickeln und Bewerten von Lösungsansätzen wurden nie über die Groupware abgewickelt. In dieser Situation bewährten sich Arbeitstreffen. Insgesamt wurde Lotus Notes sehr positiv beurteilt.

#### 4.3.5 Diskussion des aktuellen Standes

Die vorgestellten Studien und Pilotprojekte zeigen, dass die aktuellen Bemühungen tendenziell vom Einsatz existierender Werkzeuge bis hin zur Integration dieser Werkzeuge in neu konzipierte Umgebungen reichen. Dabei haben sich asynchrone Werkzeuge wie Lotus Notes oder BSCW insgesamt besser bewährt als die untersuchten Konferenzsysteme. Eine speziell für den Einsatz in einem verteilten kooperativen Software-Engineering-Praktikum entwickelte Groupware ist nicht dokumentiert, ebenso wenig

aktuelle Arbeiten zur Konzeption und Implementierung einer solchen Groupware<sup>24</sup>. Insgesamt konnte durch die beschriebene Analyse die Feststellung von Drummond und Boldyreff [Dru00] bestätigt werden, wonach in der Literatur bisher ausgesprochen wenig Forschungsbeiträge zum Einsatz von CSCW in der Hochschulausbildung und speziell in der Software-Engineering-Ausbildung publiziert sind. So wurden z.B. in den Tagungsbänden der Veranstaltungsreihe "Software im Unterricht der Hochschulen" des German Chapter of the ACM seit 1993 insgesamt 72 Beiträge veröffentlicht, aber nur ein Beitrag, der von Michael Röhlin [Röt01], thematisiert den Einsatz von Groupware in einem Software-Engineering-Praktikum.

Die dargestellten JTAP-Studien und die Pilotprojekte der TUD und der UB zeigen, dass sich das zu Beginn von Abschnitt 4.3 formulierte Lernziel 3 nicht ohne Schulungsaufwand für die Studierenden, teilweise auch für die Betreuer, erreichen lässt. Darüber hinaus muss dieses Lernziel für die Studierenden in der Form transparent gemacht werden, dass der erfolgreiche Einsatz von Groupware von den Studierenden auch als Lernfortschritt empfunden wird. Denn nicht das Endprodukt ist das Ziel eines Software-Engineering-Praktikums, sondern der Lernfortschritt der Studierenden [Schm01]. Als weiteren Aspekt zeigen die analysierten Studien einen erhöhten Bedarf an fortlaufender technischer Betreuung auf. Da sowohl die eingesetzten Systeme, als auch die eingerichteten Repositories während der Durchführung des Praktikums nicht sich selbst oder den Studierenden überlassen werden können. Macaulay et al. [Mac97] leiten aus den angeführten Überlegungen eine Erfolgsformel für den Einsatz von Groupware im Hinblick auf einen Einsatz in der Hochschulausbildung ab:

$\text{Groupware-Erfolg} = \text{Technologie} + \text{Kultur} + \text{Wirtschaftlichkeit} + \text{Politik} \\ + \text{Lernen} + \text{Schulung} + \text{Betreuung}$
---

---

<sup>24</sup> Mit dem Digital Lecture Board [Gey99] wurde zumindest eine vielversprechende Groupware für den Einsatz in Vorlesungen und Seminaren entwickelt.

## Kommunikation und Kommunikationsunterstützung

Ein Vergleich des Ausbildungsbereiches mit dem des professionellen Software-Engineering zeigt, dass der Trend zum Einsatz audiovisueller Konferenzsysteme im Ausbildungsbereich stärker ist. Selbst die Hochschulen, die im Rahmen ihrer Studien bisher auf diese Form der Kommunikation verzichtet haben, erwägen für die Zukunft den Einsatz audiovisueller Kommunikationsmittel. Bei der Suche nach den Gründen dafür muss berücksichtigt werden, dass in industriellen Teams in der Regel die Rollenverteilungen und Teamstrukturen stärker ausgeprägt sind als zu Beginn eines Software-Engineering-Praktikums. Der Bedarf an zusätzlichen sozialen Kontextinformationen zum Aufbau von Partnerbildern und Beziehungsaspekten sowie zur Förderung des Zusammengehörigkeitsgefühls ist daher im Bereich der Ausbildung deutlich höher anzunehmen. Damit eng verknüpft sind auch die Anforderungen an die Kommunikationsunterstützung, wie sie in der JTAP-Studie 2 formuliert sind. Diese Forderung adressiert wichtige Problemfelder des kooperativen Lernens, die nach Pfister et al. [Pfi01] dem heutigen Stand der Forschung und Technik entsprechend erst ansatzweise gelöst sind: die Gruppenwahrnehmung und die Koordination nicht trivialer Aktivitäten über soziale Protokolle.

## Koordinations- und Kooperationsunterstützung

Die vorgestellten Studien haben auch gezeigt, dass die Möglichkeiten zur Diskussion und gemeinsamen Ausarbeitung grafischer Entwurfsdokumente durch die verfügbaren Systeme nur unzureichend unterstützt werden. Neben der fehlenden Koordinationsunterstützung werden insbesondere grundlegende Modellierungsfunktionen vermisst. Auch das von der TUD skizzierte Konzept, das den Einsatz eines CASE-Werkzeuges zum kollaborationstransparenten Document-Sharing vorsieht, erscheint nicht Erfolg versprechend. Zwar werden insgesamt die Möglichkeiten zur Kooperation verbessert, das Problem der Koordinationsunterstützung wird auf Grund der gewählten zentralen Verteilungsarchitektur aber nicht gelöst.

Beim Einsatz von CASE-Werkzeugen ist im Einzelfall zu entscheiden, in welchem Umfang welche CASE- Funktionalität bereit gestellt werden soll. Das in der JTAP-Studie 1 beobachtete "Funktionsunbewusstsein" ist auf ein insgesamt zu hohes Funktionsangebot zurückzuführen. Untersuchungen der Ecole Polytechnique de Montreal unterstützen diese These [Bol98]. Dort wurden Untersuchungen zur Erlernbarkeit von CASE-Werkzeugen durchgeführt, die zeigten, dass sich aus der Sicht der Studierenden drei Komplexitätsebenen identifizieren lassen: die Komplexität des Aufgabenfeldes, die Komplexität der verwendeten Methode oder Technik und die Werkzeugkomplexität. Dabei wird gezeigt, dass die Summe der insgesamt implementierten Funktionen die Werkzeugkomplexität erhöht und in der Folge die Erlernbarkeit einzelner Funktionen begrenzt. Der Einsatz von CASE-Werkzeugen kann demnach zu einem steileren Verlauf der Lernkurve führen.

Darüber hinaus wird vielfach gefordert, CASE-Werkzeuge erst einzusetzen, nachdem die entsprechende Methode mit Papier und Bleistift beherrscht wird [Kne93]. Dieses Argument gilt sicher stärker für Veranstaltungen mit methodenorientierten Inhalten als für solche Veranstaltungen, die vorrangig die Projektorganisation oder das Projektmanagement thematisieren. Soll der Papier-und-Bleistift-Ansatz über eine Groupware praktisch erfahrbar gemacht werden, gelten insgesamt andere Anforderungen an das CASE-Werkzeug. Die üblicherweise von kommerziellen CASE-Werkzeugen an den Schnittstellen der Modellübergänge vorgenommenen Überprüfungen der durch die Entwurfsmethode festgelegten Regeln ist bei diesem Ansatz daher nicht gewünscht. Denn gerade diese Arbeit soll von den Studierenden geleistet werden. Das verwendete CASE-Werkzeug soll daher keinerlei Regeln zur Plausibilitäts- und Kausalitätsprüfung implementieren. Eine weitere Reduktion des Funktionsumfangs auf die Darstellung der durch die verwendete Entwurfsmethode definierten Modellierungsebenen sowie die dort verwendeten Symbole minimiert zusätzlich die Werkzeugkomplexität und führt nach [Bol98] zu einem flacheren Verlauf der Lernkurve.



## Benutzerschnittstelle

Der Gestaltung der Benutzeroberfläche und der Benutzerführung kommt in Lehr- und Lernumgebungen eine besondere Bedeutung zu, da sie nicht nur wesentlichen Einfluss auf die allgemeine Akzeptanz des Werkzeuges hat, sondern im Falle von Groupware auch den Zusammenhalt der Gruppe und die Gruppenwahrnehmung beeinflusst. Zum Beispiel behindert die im Rahmen der Infrastrukturuntersuchungen der UoA eingesetzte Version von Microsoft NetMeeting durch die Begrenzung auf maximal zwei gleichzeitig darstellbare Videofenster das Zusammengehörigkeitsgefühl und die Ausprägung von Beziehungs- und Partneraspekten.

## Organisation

Der organisatorische Aspekt ist ebenfalls von großer Bedeutung: zum Einen entsteht für den Betreuer ein deutlich höherer Aufwand durch die geänderten Anforderungen an die räumliche und technische Infrastruktur. Darüber hinaus erfordert die Organisation eines verteilten kooperativen Software-Engineering-Praktikums in letzter Konsequenz die vollständige Organisation über Dienste wie WWW, FTP oder E-Mail. Dies umfasst die Verteilung der Aufgabenstellung, die Bekanntgabe von Gruppeneinteilungen und Terminen sowie die Verwaltung der Gruppeninformationen und -dokumente. Hier sind sowohl die Betreuer als auch die Studierenden in geeigneter Weise durch das System zu unterstützen. Die untersuchten Studien haben gezeigt, dass Werkzeuge wie IBM Lotus Notes und vor allem BSCW hierzu schon vielfältige Möglichkeiten bieten.

### **4.4 Eigene Anforderungen an Groupware zum Einsatz in der Software-Engineering-Ausbildung**

Im vorigen Abschnitt wurden Erfahrungen mit dem Einsatz von Groupware in der Software-Engineering-Ausbildung diskutiert. Als weiteres Ergebnis dieser Diskussion kann festgehalten werden, dass die vorne definierten Anforderungen an Groupware für einen Einsatz im professionellen Software-Engineering nicht ohne Adaptionen auf den Ausbildungsbereich übertragen werden können, da vorrangig

- die unterschiedliche Ausgangslage in den Teamstrukturen und dem Rollenverständnis einen höheren Bedarf an sozialen Kontextinformationen im Ausbildungsbereich bedingt und
- die Verschiedenartigkeit der Endprodukte eine Betrachtung anderer Parameter für die Beurteilung der erreichten Qualität des Endproduktes "Lernfortschritt" im Falle der Software-Engineering-Ausbildung erfordert.

Im Folgenden werden die in Abschnitt 4.2.3 beschriebenen Anforderungen an Groupware für den Einsatz im professionellen Software-Engineering vor dem Hintergrund der Software-Engineering-Ausbildung präzisiert bzw. um eigene Anforderungen ergänzt, die aus den oben beschriebenen Ergebnissen abgeleitet werden.

## **1. Gemeinsame Bearbeitung von Artefakten**

Ergänzung zu 1.1:

Besondere Bedeutung kommt bei einem Einsatz in der Lehre hierbei auch der Aufgabenangemessenheit zu:

- 1.1.1 im Rahmen eines methodenorientierten Praktikums sind Möglichkeiten zur Bearbeitung textorientierter oder grafikorientierter Entwurfsdokumente bereitzustellen.

Die Funktionsumfänge der Werkzeuge zur Bearbeitung grafischer Entwurfsdokumente sollen sich an den Lernzielen des Praktikums orientieren:

- 1.1.2 Thematisiert eines der Lernziele CASE-Werkzeuge, bestehen keinerlei Einschränkungen hinsichtlich des Funktionsumfangs der eingesetzten CASE-Werkzeuge.
- 1.1.3 Thematisiert eines der Lernziele eine Entwurfsmethode, soll der Funktionsumfang der verwendeten Werkzeuge dahingehend eingeschränkt werden, dass keine von den Studierenden zu liefernden Ergebnisse automatisch generiert werden.
- 1.1.4 Thematisiert eines der Lernziele eine Programmiersprache, sollen keine Codegeneratoren integriert werden.

- 1.1.5 Thematisiert eines der Lernziele nicht explizit das Konfigurationsmanagement, sind die entsprechenden Aufgaben weitestgehend vom System und für die Studierenden transparent zu erfüllen.

## **2. Kommunikation und Kommunikationsunterstützung**

Ergänzung zu 2.1:

- 2.1.1 Die im traditionellen Praktikum bisher eingesetzten Informations- und Diskussionsforen sind in geeigneter Weise systemunterstützt anzubieten, z.B. Schwarzes Brett, Bulletin Board.

Ergänzung zu 2.2:

- 2.2.1 Es sind insgesamt mehrere unabhängige Kommunikationskanäle bereitzustellen, die in Abhängigkeit von der jeweiligen Verkehrssituation ausgewählt werden können.
- 2.2.2 Im Falle einer Videokommunikation sind alle Teilnehmer in geeigneter Weise gleichzeitig darzustellen.
- 2.2.3 Diskussionen sind in geeigneter Weise durch die Implementation sozialer Protokolle zu unterstützen.
- 2.2.4 Ein Telepointerkonzept ist zur Kommunikationsunterstützung zu realisieren. (Voting bei Bedarf)
- 2.2.5 Möglichkeiten zur Maximierung des Zusammengehörigkeitsgefühls sind zu schaffen.

## **3. Kooperative und individuelle Arbeitsmodi**

Ergänzung zu 3.1:

- 3.1.1 Zugriffskonflikte sind unbedingt durch die Implementation geeigneter Protokolle aufzulösen.

Ergänzung zu 3.3:

- 3.3.1 Jedes der integrierten Werkzeuge muss die hier definierten Anforderungen erfüllen. Eine Integration beliebiger Werkzeuge, die nicht den Anforderungen entsprechen, ist abzulehnen.

## **4. Nachvollziehbarkeit**

Ergänzung zu 4.2:

- 4.2.1 Den Studierenden sind Informationen über die an sie gestellten Anforderungen sowie über den eigenen wie auch den Projektfortschritt der Gruppe bereitzustellen.

Nachsatz zu 4.4:

*Zuständigkeiten der Mitglieder sollen sowohl für die Vergangenheit, die Gegenwart und die Zukunft sichtbar sein, sofern diese im Rahmen des Praktikums definiert sind.*

## **5. Gruppenwahrnehmung [Alt98]**

Präzisierung zu 5.2:

- 5.2.1 Diese Forderung gilt sowohl für die aktuelle Arbeitssituation als auch für die aktuelle Diskussionssituation.

**Zusätzliche Anforderung:**

## **6. Technische Infrastruktur**

- 6.1 Die zur Verfügung stehende Infrastruktur ist bei der Systemkonzeption und der Auswahl bzw. der Implementation der Werkzeuge in geeigneter Weise zu berücksichtigen.

## **4.5 Zusammenfassung**

Ausgehend von einer Darstellung der sich ändernden Situation im Arbeitsumfeld der Software-Ingenieure wurde zunächst das Potenzial virtueller Teams und die Notwendigkeit spezieller Groupware zum Einsatz im professionellen Software-Engineering diskutiert, und ein Überblick über Forschungsarbeiten im Bereich Computer Supported Cooperative Software-Engineering (CSCSE) wurde gegeben.

Der Schwerpunkt des Kapitels lag in der Analyse des aktuellen Standes bei der Implementation kooperativer Arbeitsformen in der Software-Engineering-Ausbildung an den Hochschulen. Hierzu wurde zunächst die einschlägige Literatur studiert, und Studien, deren Ergebnisse von hoher Relevanz für diese Arbeit sind wurden präsentiert und diskutiert.

Auf Grund der Tatsache, dass keine speziellen Entwicklungen bzw. Implementierungen einer Groupware für den Einsatz in der Software-Engineering-Ausbildung existieren, wurde als ein erstes Ergebnis dieser Arbeit ein Anforderungskatalog für diese Klasse von Groupware entwickelt. Die in diesem Katalog formulierten Anforderungen berücksichtigen die durch den aktuellen Stand der Technik offenbaren Probleme und reichen weit über allgemeine Anforderungen hinaus. Der Anforderungskatalog ist auf Grund der Breite der formulierten Anforderungen allgemein verwendbar. Die vorhandenen Freiheitsgrade sind im Einzelfall, in Abhängigkeit von den individuellen Anteilen der Lernziele und vom Grad der geplanten Unterstützung, weiter zu präzisieren.

## **5. Eigener Ansatz für eine synchrone Groupware zum Einsatz in der Software-Engineering-Ausbildung**

Die im vorigen Kapitel diskutierten Untersuchungsergebnisse haben zwei Tendenzen aufgezeigt; im Bereich der Projektorganisation und des Projektmanagements existieren mit BSCW und Lotus Notes zwei vielversprechende Werkzeugansätze zur Unterstützung der asynchronen Kooperation und Kommunikation. Auch die Unterstützung der synchronen Kooperation und Kommunikation in der Entwurfsphase ist insgesamt von hohem Interesse und wird vielfach angestrebt. Hier fehlt es jedoch an geeigneten Werkzeugen, die im Rahmen eines Praktikums eingesetzt werden können. Im Ergebnis existiert eine große Diskrepanz zwischen den Bemühungen bei der Einführung von CSCW in die Software-Engineering-Ausbildung einerseits und der Entwicklung adäquater Werkzeuge andererseits. Der Schwerpunkt der Arbeit liegt daher auf der beispielhaften Realisierung einer synchronen Groupware für den Einsatz in einem Software-Engineering-Praktikum.

Dieses Kapitel beschreibt die Konzeption, den Entwurf und die beispielhafte Realisierung einer synchronen Groupware. Schwerpunktmäßig werden die Entwicklung eines Prozess- und Gruppenprozessmodells, die Gestaltung der Benutzerschnittstelle sowie Ansätze zur Floor-Kontrolle und zur Gruppenwahrnehmung behandelt. Der Ausgangspunkt ist dabei eine problemorientierte Sicht auf die Gruppenprozesse unter besonderer Berücksichtigung des Benutzers und der sozialen Protokolle. Diese Problemorientierung ist essenziell für die gesamte Arbeit und prägt auch Entscheidungen des technischen Entwurfs im Rahmen der Architekturüberlegungen, des Benutzerschnittstellenentwurfs und der Implementation von Anwendungsprotokollen, die ebenfalls in diesem Kapitel beschrieben werden.

## 5.1 Eigene Voruntersuchungen zum Studium der Gruppen

In den nächsten beiden Abschnitten wird das Praktikum "Programmmentwurfstechnik", wie es an der GMU angeboten wird, aus curricularer Sicht, aus prozessorientierter Sicht und aus Sicht des Gruppenprozesses beschrieben. Darauf aufbauend werden ein Prozessmodell sowie ein Gruppenprozessmodell abgeleitet. Diese Modelle bilden die Grundlage für eine Reihe von Entwurfsentscheidungen die weiter unten beschrieben werden, andererseits beschreiben sie ein Rahmenkonzept, durch das die Anwendbarkeit der im Rahmen dieser Arbeit entwickelten synchronen Groupware in anderen Praktika abgeschätzt werden kann.

Die Veranstaltung "Programmmentwurfstechnik" ist als einsemestrige Veranstaltung im Grundstudium mehrerer ingenieurwissenschaftlicher Studiengänge der GMU verankert. Sie setzt sich aus zwei Semesterwochenstunden (SWS) Vorlesung und sechs SWS Praktikum zusammen und orientiert sich inhaltlich am Software-Lebenszyklus und den dabei verwendeten Methoden und Modellen<sup>25</sup>.

Das primäre Ziel des Praktikums aus didaktischer Sicht ist die Umwandlung des in der Vorlesung vermittelten theoretischen Wissens in nutzbares Wissen. Weitere Ziele stellen der Erwerb von Handlungskompetenz und die Steigerung der Kooperations- und Kommunikationsfähigkeit der Teilnehmer dar. Um diese Ziele zu erreichen, bearbeiten die Studierenden im Praktikum eine realitätsnahe Aufgabenstellung in Gruppen zu je vier bis fünf Studierenden. Die Durchführung des Praktikums sieht Anteile mit Anwesenheitspflicht für alle Gruppenmitglieder im Umfang von zwei SWS, Anteile selbstorganisierten Lernens im Umfang von vier SWS sowie das Angebot von Sprechstunden zur Diskussion von Teilergebnissen vor.

Die im Praktikum zu lösende Aufgabe ist in ihrem Umfang so gewählt, dass sie nicht von einem einzelnen Mitglied einer Gruppe gelöst werden kann. Dadurch sind die Stu-

---

<sup>25</sup> Hierbei handelt es sich um strukturierte Methoden zum Entwurf von Echtzeit-Systemen nach Ward & Mellor [War91].

dierenden gezwungen, das Problem in überschaubare Teilprobleme zu zerlegen und die erarbeiteten Teillösungen später wieder zusammenzufügen. Zur Lösung der Aufgabe folgen die Studierenden dem Wasserfallmodell und durchlaufen dabei mindestens einmal den gesamten Software-Lebenszyklus. Den Schwerpunkt des Praktikums bildet der Software-Entwurf, wobei bisher aus didaktischen Gründen auf den Einsatz von CASE Tools verzichtet wird. Insgesamt stehen den Studierenden 12 Wochen zur Lösung der Aufgabe zur Verfügung.

Im Rahmen eigener Voruntersuchungen wurde das Praktikum Programmwurftechnik seit 1997 mit verschiedenen Zielsetzungen und unterschiedlichen Methoden analysiert:

- **seit 1997 regelmäßige informelle Befragung der Betreuer und Tutoren** [Wer98], [Peu97];
- **1998: Experimenteller Einsatz von Microsoft NetMeeting**, um erste Erfahrungen beim Einsatz von "Groupware" im Praktikum Programmwurftechnik zu gewinnen, vgl. [Wer01]. Das Experiment wurde mit neun Studierenden durchgeführt. Die Studierenden haben anschließend einen Fragebogen ausgefüllt sowie an einer Abschlussdiskussion teilgenommen;
- **2000: Befragung aller Praktikumssteilnehmer** zur Analyse der Gruppenprozesse, vgl. [Wer01]. Die Studierenden wurden anhand von Fragebögen zur Arbeitsweise und Organisation der Gruppe befragt sowie um eine Bewertung ihrer eigenen Rolle und die des Tutors gebeten. Insgesamt haben 95 Studierende an dem Praktikum teilgenommen und jeweils beim Erreichen eines Meilensteins<sup>26</sup> einen Fragebogen ausgefüllt. Im Mittel wurden 80 Fragebögen ausgefüllt.

Im Folgenden werden die Untersuchungsergebnisse zusammenfassend dargestellt und darauf aufbauend ein Prozessmodell sowie ein Gruppenprozessmodell entwickelt.

---

<sup>26</sup> Ein Meilenstein ist definiert durch die Abgabe eines Zwischenergebnisses.



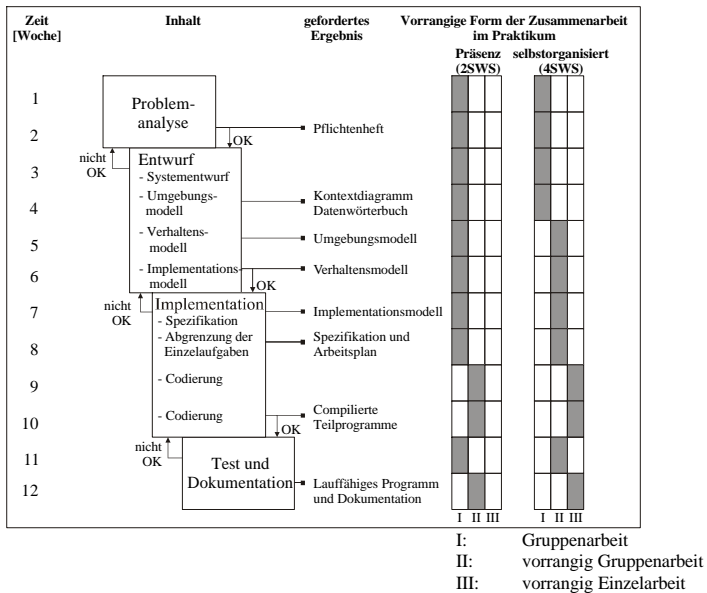
### 5.1.1 Ableitung eines Prozessmodells für das Praktikum Programmwurfstechnik

Die Analyse des Jahres 2000 bestätigte zunächst, dass der Anteil an kooperativer Gruppenarbeit in den frühen Phasen des Software-Lebenszyklus (Problemanalyse und Design) am höchsten ist, vgl. auch [Schr98] und [Mac97]. Insgesamt 85% der Studierenden gaben an, dass kooperative Gruppenarbeit die vorrangige Arbeitsform während der Praktikumszeiten mit Anwesenheitspflicht ist. Die Notwendigkeit von Gruppentreffen auch außerhalb der Praktikumszeiten mit Anwesenheitspflicht wurde mit vier von fünf Punkten sehr hoch bewertet. Insgesamt 67% der Studierenden bezeichneten im Mittel über alle Meilensteine die Gruppenarbeit als "sinnvoll zur Lösung der Teilaufgabe".

Bei den verwendeten Werkzeugen dominierten eindeutig Papier und Bleistift sowie Textverarbeitungsprogramme. Nur ca. 20 % der Studierenden gaben an, Grafikprogramme in der Modellierungsphase eingesetzt zu haben. Generell wurden deutliche Unterschiede in den praktischen Erfahrungen mit dem Einsatz von Computern beobachtet.

Bei der Frage nach der vorrangigen Gruppenstruktur wurde in allen Phasen vor allem das Modell mit gleichberechtigten Mitgliedern genannt (70%), nur 13% nannten das Modell mit aus gleichberechtigten Mitgliedern bestehenden Untergruppen als vorrangig verwendete. Das Diskussionsverhalten wurde durchweg als geordnet beschrieben, 18% der Studierenden gaben an, dass es einen Wortführer innerhalb der Gruppe gab.

Als Ergebnis der Untersuchungen wurde das in Bild 17 dargestellte Prozessmodell für das Praktikum Programmwurfstechnik abgeleitet.



**Abb. 17: Prozessmodell des Praktikums Programmierwurftechnik**

In dem Prozessmodell nach Abbildung 17 ist in der Vertikalen zunächst die Zeitachse aufgetragen. Von links nach rechts sind die am Software-Lebenszyklus orientierten Prozessschritte, die Meilensteine, bzw. die geforderten Zwischenergebnisse und die vorrangige Form der Zusammenarbeit während der Termine mit Anwesenheitspflicht (Präsenz) sowie der selbstorganisierten Zeiten aufgetragen. An den Phasenübergängen hatte der Tutor zu entscheiden, ob die Studierenden mit den Arbeiten der nächsten Phase beginnen können oder ob Nacharbeiten erforderlich sind.

Das Modell abstrahiert von einer konkreten Entwurfsmethode und der Gruppengröße. Bezüglich der Entwurfsmethode bestehen keine Einschränkungen, die Obergrenze für die Gruppengröße in Duisburg beträgt fünf Teilnehmer pro Gruppe<sup>27</sup>.

### 5.1.2 Ableitung eines Gruppenprozessmodells für das Praktikum Programmwurftechnik

Dieser Abschnitt beschreibt die Ableitung eines Gruppenprozessmodells basierend auf den Voruntersuchungen der Jahre 1997, 1998 und 2000. Zunächst werden die Ausprägungen der statischen und dynamischen Parameter des Gruppenprozesses beschrieben. Aspekte, die das Zusammenwachsen der einzelnen Gruppen betreffen und z.B. durch die Herkunft der Gruppenmitglieder aus verschiedenen Lernkulturen beeinflusst werden, lassen sich durch diese Parameter jedoch nicht beschreiben. Aus diesem Grund erfolgt zusätzlich eine Beschreibung des Gruppenprozesses durch das Team-Performance-Modell (TPM), vgl. Kapitel 2.2.1. Zusätzlich werden die möglichen Rollen der Gruppenmitglieder durch ein Rollenmodell beschrieben.

#### **Statische Parameter:**

- Das **Gruppenziel** ist im allgemeinen von dem Wunsch eines erfolgreichen Absolvierens des Praktikums geprägt und durch die geforderten Ergebnisse beschrieben. Dies erfordert zunächst das Erreichen der durch die Aufgabenstellung definierten Ziele. Da die Lieferung der geforderten Ergebnisse entweder nur der gesamten Gruppe oder aber keinem der Teilnehmer bescheinigt wird, handelt es sich hierbei um ein Gruppenziel.
- Die **Gruppenorganisation** unterscheidet zunächst den Tutor und die Studierenden. Dies sind die einzigen statischen Komponenten der Gruppenorganisation. Die Studierenden sind a priori zunächst untereinander gleichgestellt, wobei sich die Stellung einzelner Studierender innerhalb der Gruppe dynamisch ändern kann. Die Dy-

---

<sup>27</sup> Dieser Wert liegt innerhalb des Spektrums der publizierten Werte in vergleichbaren Praktikumsumgebungen, vgl. z.B. [Schr98], [Dru01].

namik wird u.a. beeinflusst von der Ausprägung aktuell benötigter Fähigkeiten und der Persönlichkeitsstruktur der Teilnehmer. Eine nähere Spezifikation findet weiter unten im Rahmen der Beschreibung der verschiedenen Rollen statt, die sich im Verlauf des Praktikums ausprägen.

- **Gruppenprotokoll:** Die Steigerung der Kooperations- und Kommunikationsfähigkeit der Teilnehmer sind implizite Ziele des Praktikums. Vorgaben bezüglich des sozialen Protokolls existieren nicht, bei Bedarf agiert der Tutor als Moderator oder Schlichter.
- Die **Gruppenumgebung** ist im Fall der Praktikumsanteile mit Anwesenheitspflicht statisch. Hier arbeiten jeweils vier Gruppen gleichzeitig in einem großen Computerraum der Hochschule. Jede Gruppe arbeitet in einer, mit jeweils drei identisch konfigurierten Computern ausgestatteten Tischzeile. Die Umgebungen der Sitzungen außerhalb der Hochschule sind unter Umständen deutlich verschieden.

#### **Dynamische Parameter:**

- **Gruppendokumente** sind vorrangig mit "Papier und Bleistift" erstellte Ideenskizzen und Entwurfsunterlagen oder Computerausdrucke von Quelltexten. Von besonderer Bedeutung sind die Aufgabenstellung, der Testabogen und die Dokumente, die Teil- und Endergebnisse enthalten. Die Verwaltung und Versionierung der Dokumente ist Aufgabe der Studierenden.
- **Gruppenaktivitäten:** Ein Teil der Aktivitäten ist durch die gegebene Aufgabenstellung sowohl temporal als auch kausal voneinander abhängig. Zusätzliche Aktivitäten können von den Studierenden frei definiert werden. Eine beispielhafte Beschreibung ausgewählter Aktivitäten erfolgt weiter unten.
- **Gruppensitzungen:** Die einzelnen Aktivitäten werden sowohl in synchronen als auch in asynchronen Sitzungen bearbeitet. Die synchronen Sitzungen mit Anwesenheitspflicht finden zu fest vereinbarten Terminen<sup>28</sup> statt und in der Regel sind alle Gruppenmitglieder anwesend. Diese Voraussetzungen sind bei den selbstorgani-

---

<sup>28</sup> Die Termine werden im Vorfeld per Aushang bekannt gegeben.

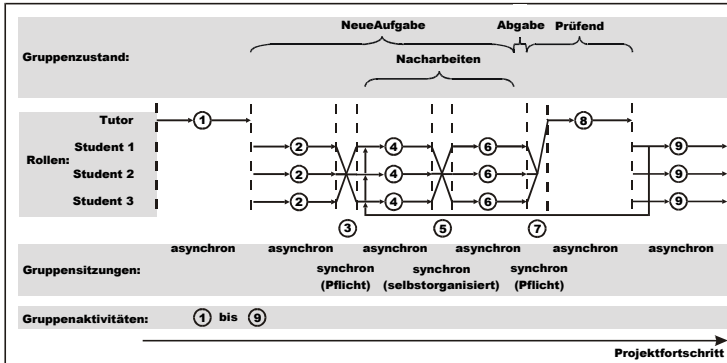
sierten synchronen Sitzungen nicht immer erfüllt. Neben den synchronen Sitzungen sind zusätzlich asynchrone Sitzungen möglich.

- **Gruppenzustand:** Neben einer Reihe von Zuständen, die sich im Laufe des Gruppenprozesses individuell herausbilden, existieren auch eine Reihe von Standardzuständen:

Im Zustand *NeueAufgabe* machen sich die Studierenden mit der nächsten Teilaufgabe vertraut und führen alle Aktivitäten, die zur Lösung der Teilaufgabe erforderlich sind, durch. Dieser Zustand ist nicht zeitlich limitiert und kann synchrone und asynchrone Sitzungen beinhalten. Der Zustand *Abgabe* markiert ein synchrones Treffen mit Anwesenheitspflicht, an dem die Studierenden Teil- oder Endergebnisse abliefern müssen. Im Zustand *Prüfend* erfolgt die Überprüfung der Teil- oder Endergebnisse durch den Tutor. Eine erfolgreiche Überprüfung der Dokumente führt zur Abzeichnung des betreffenden Meilensteins in der Testatliste der Studierenden und führt über in den Zustand *NeueAufgabe*. War die Überprüfung nicht erfolgreich, erfolgt ein Rückschritt innerhalb des Projektes in den Zustand *Nacharbeiten*. Da die Studierenden in der Regel parallel die vom Tutor angemahnten Schwachstellen und die nächste Teilaufgabe bearbeiten, ist eine scharfe Trennung dieser beiden Zustände nur schwer möglich. Ein besonders kritischer Zustand ist *Ausschluss*, der das nicht erfolgreiche Absolvieren des Praktikums markiert. Studierende werden dann vom Praktikum ausgeschlossen, wenn sie mehr als zwei Termine versäumen oder mehr als zweimal endgültig kein akzeptables Teilergebnis geliefert haben. Der zweite Aspekt ist insofern schwer zu beurteilen, als dass in der Regel die Gruppe als Ganzes ein Teilziel erreichen muss. Der Ausschluss einzelner Studierender erfolgt üblicherweise nur dann, wenn sie nachweislich unter Vorsatz entgegen den Gruppenzielen handeln und somit nicht nur das eigene Erreichen der Lernziele gefährden, sondern auch das der übrigen Gruppenmitglieder.

In Abbildung 18 ist exemplarisch für drei Studierende und einen Tutor der Ablauf einer Sequenz innerhalb des Praktikums dargestellt. Außerdem enthält Bild 18 die Gruppenzustände, die Gruppenaktivitäten sowie deren temporale und kausale Abhängigkeiten

und die Art der Gruppensitzungen. Nachfolgend werden die Aktivitäten bzw. die Art der Zusammenarbeit innerhalb des dargestellten Ausschnittes näher beschrieben.



**Abb. 18:** Ablauf während des Praktikums in Anlehnung an die Darstellung in [Bor98]

Im Vorfeld des Praktikums wird die nächste Teilaufgabe vom Tutor bekannt gegeben (Aktion (1)). Die Studierenden machen sich in der Regel getrennt voneinander mit der Teilaufgabe vertraut (Aktion (2)) und diskutieren diese im Rahmen des nächsten Praktikumstermins mit Anwesenheitspflicht (Aktion (3)), um gegebenenfalls den Tutor zu Rate ziehen zu können. Zwischen zwei Praktikumsterminen mit Anwesenheitspflicht findet in der Regel mindestens ein weiteres Gruppentreffen statt (Aktion (5)), und zusätzlich arbeiten die Studierenden mehr oder weniger viel in Einzelarbeit (z.B. Aktionen (4) und (6)). Bei dem nächsten Praktikumstermin mit Anwesenheitspflicht (Aktion (7)) werden die Arbeiten an der aktuellen Teilaufgabe abgeschlossen und das geforderte Teilergebnis abgeliefert. Unabhängig von der Gruppe überprüft der Tutor die gelieferten Ergebnisse (Bestandteil von Aktion (7) oder Aktion (8)) und informiert die Gruppe noch während der laufenden Sitzung oder kurz danach über die Korrektheit der Lösung. In diesem Fall wird der Prozess bei Aktion (9) fortgesetzt, wobei dies einer Iteration beginnend bei Aktion (1) entspricht. Ist die Lösung nicht korrekt, wird die Gruppe nach dem Pflichttermin darüber in Kenntnis gesetzt und fällt zurück in Aktion (4).

## Anwendung des Team-Performance-Modells (TPM)

Den Phasen der Orientierung und Vertrauensbildung kommt im Falle des Praktikums Programmwurfstechnik eine besondere Bedeutung zu, da eine Vielzahl der Studierenden im auslandsorientierten Studiengang erst kurz vor Praktikumsbeginn ihr Studium an der GMU aufnehmen. Daher sind sich nicht nur die meisten Teilnehmer innerhalb einer Gruppe fremd, sondern sie haben in der Regel unterschiedliche Nationalitäten und stammen aus zum Teil verschiedenen Lernkulturen. Eine scharfe zeitliche Abgrenzung der Phasen zur Orientierung und Vertrauensbildung ist daher nur schwer möglich. Dies gilt in ähnlicher Weise für die dritte Phase der Ziel- und Aufgabendefinition. Einerseits herrscht hier frühzeitig Konsens über die Gruppenziele und die dabei zu lösenden Aufgaben, andererseits wird dieser Prozess in jeder Phase des Software-Lebenszyklus mehr oder weniger vollständig durchlaufen. Dies gilt äquivalent für die Phase der Verpflichtung. Die ersten vier Phasen des TPM können daher nahezu zusammengefasst und als Ursache für das hohe Maß an synchroner Gruppenarbeit in den frühen Phasen des Software-Lebenszyklus angesehen werden, vgl. Abb. 17. Im Hinblick auf die Praktikumsanteile mit Anwesenheitspflicht sind hier zwei grundsätzliche Arbeitsweisen beobachtet worden: das gemeinsame synchrone Bearbeiten eines Dokuments und das asynchrone Bearbeiten verschiedener alternativer oder sich gegenseitig ergänzender Entwurfsdokumente durch einzelne Studierende.

Der Anteil an individueller Gruppenarbeit ist in den Phasen Implementierung und Hohe Leistung signifikant höher als in den übrigen Phasen des TPMs. Die dabei in Einzelarbeit erstellten Lösungsansätze werden anschließend meistens noch im Rahmen desselben Treffens im Team diskutiert und gegebenenfalls mit alternativen Lösungsvorschlägen verglichen. Die Erneuerungsphase sollte im Praktikum wegen der vergleichsweise geringen Laufzeit des Praktikums üblicherweise nicht erreicht werden.

## Rollenmodell

Der Arbeitsstil der Studierenden während des Praktikums reicht vom aktiven Stil, der durch eine explorative Grundhaltung gekennzeichnet ist, bis zum passiven Stil, mit stark

rezeptiver Grundhaltung. Der Lerntyp der Studierenden lässt sich als durchweg gruppenorientiert bezeichnen. Den Studierenden werden a priori keine Rollen zugewiesen. Die Rolle des Tutors während des Praktikums ist ambivalent. Einerseits soll er als zusätzliches Gruppenmitglied bei Bedarf den Fortschritt der Gruppe sichern, andererseits aber auch Teilergebnisse einfordern und bewerten. Damit existieren zunächst nur zwei Rollen, die des Tutors und die des Studierenden, wobei sich im Verlauf des Praktikums in der Regel eine oder mehrere der folgenden Rollen innerhalb der Gruppen herausbilden:

- **Projektleiter:** In vielen Fällen bildet sich im Verlauf des Praktikums ein Gruppensprecher heraus, der einerseits die Interessen der Gruppe gegenüber dem Tutor vertritt und andererseits auch die Gruppe leitet und vorrangig über die nächsten Schritte entscheidet. Hat sich die Rolle des Projektleiters einmal in der Gruppe etabliert, ist sie in der Regel dauerhaft von einer Person besetzt.
- **Passive Teilnehmer:** Eine Reihe von Studierenden sind während der gemeinsamen Gruppensitzungen aus verschiedenen Gründen stark passiv und beteiligen sich im Mittel deutlich weniger an Diskussionen und der Bearbeitung von Dokumenten. Oftmals liegt die Stärke der passiven Teilnehmer in der Einzelarbeit.
- **Bearbeiter:** Während der synchronen Sitzungen übernimmt in der Regel jeweils nur ein Gruppenmitglied die Bearbeitung des aktuellen Gruppendokuments. Die anderen Gruppenmitglieder haben jedoch jederzeit die Möglichkeit, Ideen oder Änderungsvorschläge in den Gruppenprozess einzubringen. Es ist ebenfalls jederzeit jedem Studierenden möglich, die Bearbeitung des aktuellen Dokumentes in Absprache mit den anderen Teilnehmern zu übernehmen. Die Rolle des Bearbeiters wird in der Regel dynamisch in Abhängigkeit vom aktuellen Lernfortschritt, der Kreativität oder gerade benötigter Fähigkeiten im Umgang mit Werkzeugen dynamisch zugewiesen.
- **Aktive Teilnehmer:** Der aktive Teilnehmer zeichnet sich im Gegensatz zur Rolle des Bearbeiters dadurch aus, dass er während der synchronen Gruppensitzungen einerseits regelmäßige Diskussionsbeiträge liefert, aber andererseits im Mittel erheblich seltener die Bearbeitung der Gruppendokumente übernimmt.



### 5.1.3 Szenario für den Einsatz der synchronen Groupware PASSENGER

Der Einsatz der synchronen Groupware ist an der GMU sowohl in den traditionellen Diplomstudiengängen vorgesehen als auch in den neu eingeführten Bachelor-/Master-Studiengängen mit starker Auslandsorientierung. Ein Einsatz in weltweit verteilten Teams soll daher direkt berücksichtigt werden. Vor diesem Hintergrund wurde PASSENGER als Name für die synchrone Groupware gewählt.

Im Sinne der Klassifikation nach Abbildung 2 handelt es sich bei der synchronen Groupware PASSENGER um eine Groupware für die räumlich verteilte, synchrone Gruppenarbeit an nicht vorhersehbaren Orten. Eine Kommunikationsunterstützung über Telefon oder der Einsatz von ISDN-basierten Konferenzsystemen scheidet aus Kostengründen aus. Als Transportplattform wird daher das Internet gewählt. Der Zugang zum Internet kann nicht weiter eingeschränkt werden und ist vielmehr von den jeweils individuellen Möglichkeiten abhängig. Es soll ebenfalls nicht davon ausgegangen werden, dass die Studierenden immer denselben Arbeitsplatz verwenden. Eine vergleichbare Ausstattung aller Arbeitsplätze hingegen wird vorausgesetzt. Aus dem beschriebenen Szenario kann nun eine erste Grobspezifikation abgeleitet werden:

- Endsysteme: PC's;
- Transportmedium: Internet;
- Zugang: Modem, ISDN, LAN oder ähnliches.

## 5.2 Architektur der synchronen Groupware PASSENGER

Die synchrone Groupware PASSENGER ist eine verteilte Anwendung für Teilnehmer an zum Teil nicht vorhersehbaren Orten. Dem Standort der GMU kommt eine besondere Bedeutung zu, da hier die Verantwortung für die Sicherung eines reibungslosen Ablaufs des Praktikums liegt. Verallgemeinert kann daher der Heimatstandort als zentrale Organisations- und Koordinationsstelle angesehen werden. Ebenfalls kann angenommen werden, dass hier der Tutor seinen in Hinblick auf Telefonnummer und IP-Ad-

resse zeitinvarianten Arbeitsplatz hat. Darüber hinaus dürfte auch bei der Berücksichtigung weltweit verteilter Teams die Mehrzahl der Studierenden am Heimatstandort und sogar innerhalb der von der GMU bereit gestellten Räume arbeiten. Für die Verwaltung des gemeinsamen Repositories an einem zentralen Ort, wie es Gorton und Motwani [Gor96] vorschlagen, ist daher der Heimatstandort aus verschiedenen Gründen prädestiniert. Die Fakultäten der Hochschulen verfügen zum einen in der Regel über ausreichend leistungsfähige Serversysteme zur Verwaltung der Repositories und gewährleisten einen zeitlich unbegrenzten Zugriff darauf. Darüber hinaus sind die deutschen Hochschulen üblicherweise mit hoher Bandbreite an das DFN angeschlossen und auch innerhalb der einzelnen Institute mit Ethernet oder Fast-Ethernet ausgestattet. Der Bottleneck-Effekt beim gleichzeitigen Zugriff auf die gemeinsamen Ressourcen dürfte im Falle der Hochschulen damit deutlich geringer sein, als z.B. bei der Wahl eines Standortes der Studierenden oder einer ausländischen Hochschule. Auch der Nachteil, dass der Server einen Single-Point-of-Failure darstellt, wird durch die Wahl des Heimatstandortes relativ zu den alternativen Möglichkeiten abgemildert, da hier ausreichend technischer Support im Fehlerfall angenommen werden kann. Damit wird aus Sicht der Datenorganisation eine Client/Server-Architektur mit Server-Standort an der GMU gewählt.

Die Durchführung einer synchronen Sitzung erfordert zum einen Vorbereitungen im Vorfeld der Sitzung, hierfür ist das Konferenz-Management verantwortlich. Für die Organisation während der Sitzung ist das Sitzungs-Management verantwortlich. Aufgaben des Konferenz-Managements sind die Ankündigung einer Sitzung und die Bereitstellung der benötigten Informationen und Parameter für die Initiierung der Sitzung. Die Ankündigung der Sitzungen im Rahmen der Pflichtveranstaltungen erfolgt üblicherweise per Aushang und kann für die virtuellen Teams über eine Internet-Seite bekannt gemacht werden. Die Pflege der Internet-Seite obliegt dem Tutor und sie wird auf einem Web-Server der GMU veröffentlicht. Terminänderungen können per E-Mail bekannt gemacht werden, ebenso Vereinbarungen für Sitzungen außerhalb der Pflichttermine.

Da es sich bei dem Praktikum um geschlossene Gruppen mit Teilnehmern an zum Teil unbekanntem Orten handelt, kommt der Problematik der Gruppenmitgliedschaft sowie des aktuellen Aufenthaltsortes, bzw. der aktuellen IP-Adresse eine besondere Bedeutung zu. Die Gruppenmitgliedschaft ist in der Regel bekannt und liegt für die Dauer des Praktikums fest. Hierfür können also Benutzer- und Gruppenkennungen sowie Passwörter für den Beitritt zu einer Session vergeben werden. Die Speicherung der Zugangsdaten und die Kontrolle im Falle einer Anmeldung findet auf dem Server an der GMU statt. Die IP-Adresse des Servers an der GMU wird im Vorfeld der ersten Sitzung bekannt gegeben. Sie ist üblicherweise für die Dauer des Praktikums konstant. Für den Beitritt zu einer Sitzung wurde folgendes Vorgehen entwickelt: zusammen mit der Benutzerkennung, der Gruppenkennung und dem Passwort werden beim Verbindungsaufbau mit dem Server auch die IP-Adressen<sup>29</sup> des lokalen Systems übertragen und im gemeinsamen Repository der Gruppen abgelegt. Diese Informationen sind damit unmittelbar nach dem Beitritt eines Mitglieds zu einer Sitzung auf dem Server bekannt und können auch den übrigen Gruppenmitgliedern bekannt gemacht werden.

Der Datenaustausch zwischen Client und Server nach dem oben bezeichneten Schema betrifft vorrangig Daten, die einen zuverlässigen Dienst erfordern. Im Internet wird ein solcher Dienst durch das TCP bereitgestellt, das den Erhalt einzelner Datenpakete überprüft und bei Bedarf eine Neuübertragung initiiert.

Ein weiterer Schwerpunkt der Arbeit liegt in der Konzeption und Entwicklung einer geeigneten Kommunikationsumgebung.

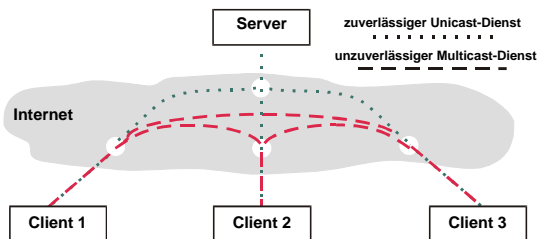
Die Konzeption muss dabei insbesondere auch das im vorigen Abschnitt beschriebene Szenario und die durch die Phasen der Orientierung und Vertrauensbildung definierten Implikationen des Team-Performance-Modells berücksichtigen. Aus diesen Gründen kann auf den Einsatz von Videoverbindungen nicht verzichtet werden. Da nach Frey [Fre92] vor allem der Gesichtsbereich und die Hände für die Vielzahl der Ausdrucks-

---

<sup>29</sup> Bei IPv6 basierten Weitverkehrssystemen ist in der Regel jedes Netzwerk-Interface mit mehr als einer IP-Adresse ausgestattet.

möglichkeiten im Zusammenhang mit der nonverbalen Kommunikation verantwortlich ist, ist eine Darstellung des Oberkörpers durchaus ausreichend. Die Kommunikation während einer Sitzung soll daher standardmäßig über Audio- und Video-Verbindungen erfolgen und erst im Falle schlechter Übertragungsverhältnisse mittels Chat abgewickelt werden. Sowohl im Hinblick auf das Datenvolumen und die Toleranz gegenüber Paketverlusten als auch im Bezug auf die Anforderungen an Delay und Jitter unterscheiden sich Audio- und Videodatenströme deutlich von den oben beschriebenen Daten beim Austausch zwischen Client und Server. So benötigt diese Klasse von Daten nicht zwingend einen zuverlässigen Dienst und kann grundsätzlich über UDP-basierte Dienste abgewickelt werden. Hierfür wird ein direkter Datenaustausch zwischen den beteiligten Clients im Multicast-Verfahren gewählt. Aus Gründen der Zukunftsorientierung und der Flexibilität gegenüber Erweiterungen und der Integration neuer Dienste wird außerdem das IP-Protokoll in der Version 6 als das Basisprotokoll gewählt. Das entwickelte Verfahren zum Generieren der Gruppenadresse wird in Kapitel 5.6 diskutiert.

Insgesamt ergibt sich damit die in Abbildung 19 dargestellte Client/Server-Architektur, mit unzuverlässigen Unicast-Diensten zum Datenaustausch zwischen Client und Server. Gleichzeitig integriert diese Architektur eine Interclient-Kommunikation zum Austausch multimedialer Daten zwischen den Clients über einen unzuverlässigen Multicast-Dienst. Dabei sind zur besseren Übersicht nur drei Clients dargestellt.



**Abb. 19:** Client/Server-Architektur der synchronen Groupware PASSENGER

## 5.2.1 Auswahl eines Architekturmodells

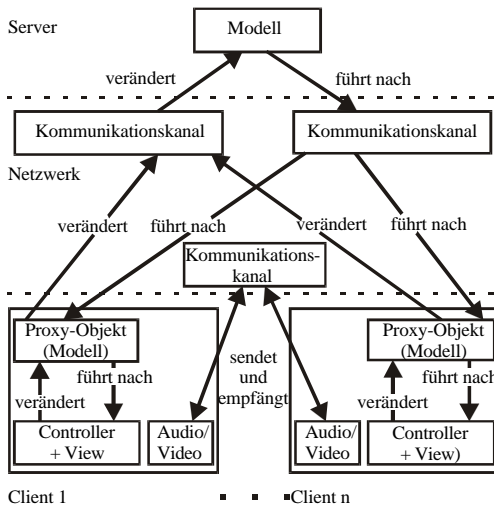
Die in Kapitel drei vorgestellten Modelle zur Beschreibung der Architektur von Groupware sehen in erster Näherung eine zentrale Verwaltung des Datenbestandes vor. Die nach Patterson [Pat95] am besten geeignete Architektur für synchrone Systeme, die hybride Architektur, hat den Nachteil, dass

- kontinuierliche Datenströme nur schwer zu modellieren sind und
- gemeinsame Anzeigezustände, wie z.B. Telepointer, die nicht auf der Modellebene modelliert werden, nicht darstellbar sind.

Darüber hinaus berücksichtigt das Modell nicht explizit die Anforderungen nach kurzen Antwort- und Benachrichtigungszeiten. Das NetMVC-Modell dagegen:

- bietet kurze Antwortzeiten durch lokale Modellkopien in den Proxy-Objekten;
- bietet die Möglichkeit kurzer Benachrichtigungszeiten durch Änderungsmitteilungen und die Übertragung der geänderten Daten in einem Schritt;
- erhöht die Robustheit gegenüber Netzwerkausfällen durch die Kombination von Benutzerschnittstelle und lokaler Modellkopie auf den einzelnen Benutzerrechnern;
- erlaubt die Verwendung privater Sichten.

Das aus diesen Überlegungen resultierende Architekturmodell der synchronen Groupware PASSENGER ist an das NetMVC-Modell angelehnt und in Abbildung 20 dargestellt. Die Darstellung enthält neben den Kommunikationskanälen zum Austausch der Modelldaten auch den Kommunikationskanal zum Transport der audiovisuellen Daten.

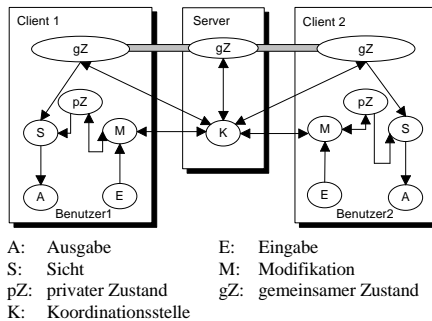


**Abb. 20:** Das Architekturmodell der synchronen Groupware PASSENGER

### 5.2.2 Auswahl einer Verteilungsarchitektur

Bei der Auswahl der optimalen Verteilungsarchitektur müssen auch die bisherigen Festlegungen auf eine Client/Server-Architektur und die Anwendung des NetMVC-Konzeptes in die Diskussion einbezogen werden. Die im Architekturmodell verwendeten lokalen Modellkopien sprechen dabei gegen eine zentrale Verteilungsarchitektur und die zentrale Verwaltung der Modelldaten auf dem Server gegen eine vollständig replizierte Verteilungsarchitektur. Die Mischform der beiden Ansätze, die hybride bzw. die semi-replizierte Architektur, kombiniert die Vorteile der beiden Ansätze. Grundsätzlich wird daher für die synchrone Groupware PASSENGER ein hybrider Ansatz gewählt. Der hybride kollaborationstransparente Ansatz scheidet aus, da er keine privaten Sichten ermöglicht. Der kollaborationsbewusste hybride Ansatz bietet durch die Zentralisierung der Koordinationskomponente diese Möglichkeit, erfordert aber eine Synchronisation der gemeinsamen Anwendungszustände. Andererseits ist durch die

Wahl des NetMVC-Konzeptes die zentrale Verwaltung der Modelldaten bzw. des gemeinsamen Zustands vorgegeben. Diese Primärkopie des gemeinsamen Zustands bietet auch den Vorteil, dass Mitglieder, die verspätet einer Konferenz beitreten, unmittelbar nach ihrem Beitritt Zugriff auf den gemeinsamen Anwendungszustand haben. Außerdem bietet dieser Ansatz aus implementationstechnischer Sicht eine deutliche Vereinfachung gegenüber einer Synchronisation der Anwendungszustände. Damit wird für die synchrone Groupware PASSENGER eine hybride kollaborationsbewusste Verteilungsarchitektur mit Primärkopie gewählt, wie sie in Abbildung 21 dargestellt ist, vgl. auch [Jun01].

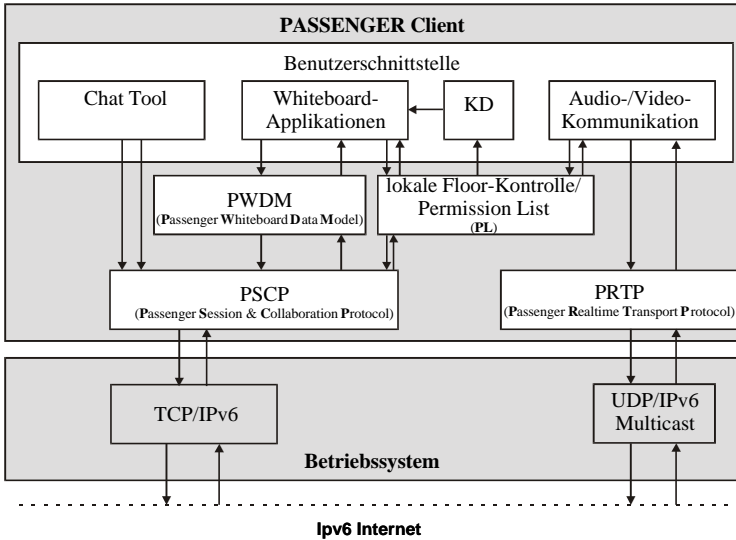


**Abb. 21:** Verteilungsarchitektur der synchronen Groupware PASSENGER

### 5.2.3 Entwicklung einer Systemarchitektur

Der synchronen Groupware PASSENGER kommen zwei wesentliche Aufgaben zu: zum einen die Unterstützung der Studierenden bei der Kommunikation als auch bei der Kooperation. Die Systemarchitektur besteht daher zunächst auf der Anwendungsebene aus einem Whiteboard, Anwendungen zur Audio- und Video-Kommunikation und einem Chat Tool. Die geforderte Kommunikationsunterstützung wird auf der Anwendungsebene als eigenständige Ressource modelliert und z.B. als kollaborativer Dienst (KD) in Form eines Telepointers implementiert. Der Benutzer interagiert mit diesen Anwendungen über die Benutzerschnittstelle, wobei jede der Anwendungen Benutzer-

eingaben entgegen nimmt. Die folgende Abbildung zeigt die gesamte Systemarchitektur der synchronen Groupware PASSENGER.



**Abb. 22:** Systemarchitektur der synchronen Groupware PASSENGER

Die Kooperationswerkzeuge werden als eigenständige Ressourcen über die zentrale Koordinationskomponente auf dem Server verwaltet. Die hierzu entwickelte Floor-Kontrolle - das PASSENGER Floor Control Model (PCFM) - wird im nächsten Abschnitt beschrieben.

Die lokale Floor-Kontrolle nimmt Benutzereingaben entgegen und gibt die lokalen, unter Floor-Kontrolle stehenden Ressourcen frei oder sperrt sie. Hierzu arbeitet die lokale Floor-Kontrolle auf einer Beschreibung des aktuellen Zustands des PCFM, der vom Server in Form der Permission List (PL) an alle Clients repliziert wird. Neben den Kooperationswerkzeugen werden gemäß den Anforderungen aus Kapitel 4.4 zusätzlich sowohl der gemeinsame Arbeitsbereich als auch die auditiven Sendekanäle von der Floor-Kontrolle verwaltet. Der gemeinsame Arbeitsbereich greift entsprechend dem Ar-



chitekturmodell aus Abbildung 20 auf die lokale Modellkopie zu und kann diese auch verändern. In Abbildung 22 ist dieses Modell als PASSENGER Whiteboard Data Model (PWDM) bezeichnet. Das PWDM wird in Abschnitt 5.5.2 beschrieben.

Die Kommunikation mit der Netzwerkschicht erfolgt mit zwei Protokollen der Anwendungsebene, die entsprechend dem ALF-Ansatz die Daten bereits auf der Anwendungsebene aufbereiten. Das PASSENGER Session and Collaboration Protocol (PSCP) stellt die Schnittstelle zu dem zuverlässigen Unicast-Dienst TCP bereit und liegt im Kommunikationspfad zwischen Client und Server. Entsprechend werden die audiovisuellen Daten über das PASSENGER Realtime Transport Protocol (PRTP) an die Transportschicht übergeben und über den unzuverlässigen UDP-Dienst im Multicast-Verfahren versendet. Die beiden Protokolle werden im Abschnitt 5.6 beschrieben.

#### 5.2.4 Floor-Kontrolle

Die Untersuchungen im Rahmen dieser Arbeit haben gezeigt, dass die bisher in Software-Engineering-Praktika eingesetzten Werkzeuge zur synchronen Kommunikation und Kooperation keine ausreichende Kooperations-, Kommunikations- und Koordinationsunterstützung bieten. In Kapitel 4.3.5 wurden als Ursache hierfür zwei wesentliche Problemfelder genannt: die Gruppenwahrnehmung und die Koordination nicht trivialer Aktivitäten über soziale Protokolle. Aufbauend auf dem in Kapitel 5.1.2 beschriebenen Gruppenprozess wird im Folgenden die Entwicklung einer Floor-Kontrolle diskutiert, die den Zugriff auf die gemeinsamen Ressourcen kontrolliert.

Die Aufgaben der Floor-Kontrolle können wie folgt zusammengefasst werden [Dom97]:

- Zugriffskontrolle für gemeinsam genutzte Ressourcen und Zustandsüberwachung von verteilten Datenobjekten (bzgl. Konsistenz).
- Gewährleistung der Fairness bei der Floor-Vergabe und Überwachung des Prinzips des “gegenseitigen Ausschlusses”.
- Bereitstellung verschiedener Kontrollstrategien.

- Sicherungsmechanismen bei Verlust oder Duplizierung des Floors.

Die ersten beiden Forderungen stellen im Kontext dieser Arbeit und aus Sicht der Kommunikation und Kooperation die wesentlichen Punkte dar. Die Zugriffskontrolle und Zustandsüberwachung sichert einerseits die Konsistenz, andererseits hat die Granularität von Sperrmechanismen einen wesentlichen Einfluss auf die Benutzerakzeptanz. Systeme mit feingranularen Sperrmechanismen bieten ein hohes Maß an Individualität, bergen aber auch die Gefahr des Orientierungsverlustes. Hier sind insgesamt aufwändigere Mechanismen zur Gruppenwahrnehmung und zur Konsistenzerhaltung zu entwickeln als im Falle grobgranularer Systeme. Diese werden oftmals allerdings als einschränkend empfunden.

Die im Gruppenprozessmodell definierten Rollen unterscheiden im Wesentlichen aktive und passive Teilnehmer, wobei zu einem Zeitpunkt immer genau eine Person - in der Rolle des Bearbeiters - den aktuellen Entwurf bearbeitet oder kommentiert. Das gleichzeitige Arbeiten in demselben Entwurfsdokument ist nicht die Regel. Diese Arbeitsweise wird am besten durch grobgranulares Sperren auf Dokumentenebene modelliert. Die Möglichkeit, ein Dokument nur zu kommentieren, führte zu der Entscheidung, den Audiokanal an die Bearbeitungsrechte des gemeinsamen Dokumentes zu binden und zusammen mit einem Telepointer unter die gemeinsame Floor-Kontrolle zu stellen.

Eine Floor-Kontrolle, die den Anforderungen der Fairness und des gegenseitigen Ausschlusses genügt, ist um so leichter zu erfüllen, je mehr Kenntnisse über die den Gruppenprozess beschreibenden Parameter vorliegen. Geschlossene Gruppen bieten den Vorteil, dass ihre Teilnehmerzahl konstant ist und sich somit hinsichtlich der Fairness einfache, überprüfbare Maßzahlen ermitteln lassen. Im Folgenden wird, zunächst beispielhaft, von einer Gruppe, bestehend aus drei Studierenden ausgegangen.

Die Fairness eines Systems kann unterschiedlich realisiert werden. Eine technikorientierte Lösung würde allen Studierenden in einem Diskussions- und Arbeitszyklus die gleiche Redezeit zuteilen, vgl. [Bor98]. Diese Lösung kann als unfair aufgefasst werden, nämlich dann, wenn der aktuelle Floor-Besitzer seine Arbeiten und Ausführungen

vor Ablauf seines Zeitfensters noch nicht abgeschlossen hat. Ebenso ist es möglich, dass ein Floor-Besitzer keine Beiträge zu liefern hat. In dem Fall würde die Bearbeitungszeit ungenutzt verstreichen. Aus diesem Grund wurde für die synchrone Groupware PASSENGER die Verwendung von Zeitfenstern nicht in Betracht gezogen und eine stärker problemorientierte Lösung erarbeitet. Diese sieht vor, dass jedem Studierenden innerhalb eines Diskussions- und Arbeitszyklus' die Möglichkeit gegeben wird, den Floor zu beantragen und auch zu erhalten. Dies entspricht auch dem beobachteten Rollenverhalten; ein passiver Teilnehmer wird wahrscheinlich nicht in jedem Zyklus von seinem Recht Gebrauch machen. Andererseits sollen passive Teilnehmer auch nicht von der Diskussion ausgeschlossen werden können. Hierfür wurde ein einfaches, aber effizientes Verfahren implementiert. Bei diesem Verfahren ist ein Arbeitszyklus dadurch definiert, dass jeder der  $n$  Teilnehmer einer Sitzung einmal die Möglichkeit hat, den Floor-Besitz zu übernehmen. Um Blockierungen zu verhindern, wird die Übernahme des Floors durch jeden der  $n$  Teilnehmer jedoch nicht explizit gefordert. Haben  $n-1$  Teilnehmer einmal den Floor-Besitz übernommen und bis zum Zeitpunkt  $t_{n-1}$  den Floor wieder freigegeben, verfällt das Anrecht des letzten Teilnehmers, wenn er bis zum Zeitpunkt  $t_{n-1}$  nicht den Floor beantragt hat. In diesem Fall beginnt zum Zeitpunkt  $t_{n-1}$  ein neuer Arbeitszyklus.

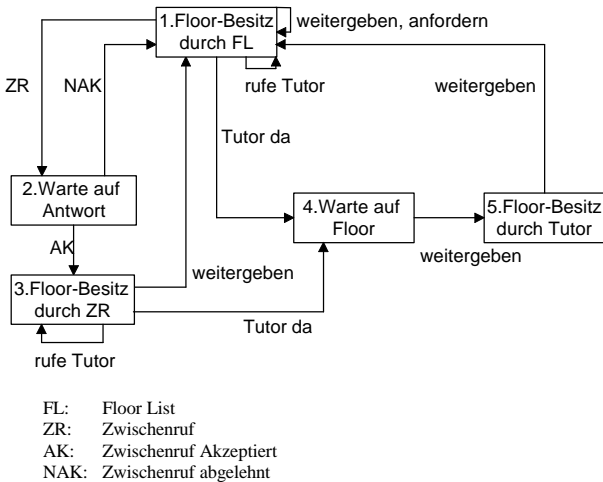
Das beschriebene Verfahren ist als so genannte Floor-List (FL) implementiert und die Größe der Liste ist entsprechend der Gruppengröße bemessen. An erster Stelle der FL steht der aktuelle Floor-Besitzer, gefolgt von den nächsten Bewerbern um den Floor. Dabei gilt die Regel, dass die Liste zu keinem Zeitpunkt zwei identische Einträge aufweisen darf. Dadurch wird sicher gestellt, dass sich jeder Teilnehmer pro Arbeitszyklus einmal um den Floor bewerben kann und diesen dann auch erhält.

Weiter soll - entsprechend dem Rollenkonzept - die Rolle des Tutors berücksichtigt werden. Der Tutor soll jederzeit in die laufende Diskussion eingreifen können. Dazu muss es ihm möglich sein, jederzeit den Floor zu beantragen und auch zu erhalten.

Neben den bisher diskutierten Maßnahmen zur Erfüllung der Anforderungen an eine Floor-Kontrolle soll ein zusätzliches Verhalten im sozialen Protokoll modelliert werden:

die Zwischenbemerkung, bzw. der Zwischenruf. Durch einen Zwischenruf gibt der Studierende zu erkennen dass er den aktuellen Floor-Besitzer, kurzfristig für eigene Anmerkungen –die Zwischenbemerkung-, unterbrechen möchte. Danach kann der ursprüngliche Floor-Besitzer mit seinen Ausführungen fortfahren.

Das aus den beschriebenen Überlegungen abgeleitete technische Protokoll ist in Abbildung 23 als Zustandsmodell gezeigt und ist als PASSENGER Floor Control Model (PFCM) implementiert. Das PFCM bildet die Basis der serverseitig implementierten Floor-Kontrolle.



**Abb. 23:** Zustandsübergangsdiagramm des PFCM

Der aktuelle Zustand des PFCM wird in der Permission List (PL) gespeichert und an alle Teilnehmer repliziert. Die PL verwaltet in der Floor List (FL) den aktuellen Floor-Besitzer sowie die weiteren Bewerber um den Floor und enthält darüber hinaus Einträge für

- den Teilnehmer, der auf Grund eines Zwischenrufs der aktuelle Floor-Besitzer ist,
- den Teilnehmer, der einen Zwischenruf beantragt hat und
- den Teilnehmer, der den Tutor gerufen hat.

Jede Anforderung des Floors, bzw. der Bearbeitungsrechte (*-anfordern-*) und jede Weitergabe mit *-weitergeben-* führt zu einem Update der FL. Dabei gilt, dass die FL niemals zwei gleiche Einträge aufweisen darf. Im Folgenden werden die durch das PFCM definierten Abläufe anhand des Zustandsübergangsdiagramms aus Abbildung 23 beschrieben.

Im Zustand 1 (Floor-Besitz durch FL) sind die Sitzung und alle parallelen Prozesse zur Annahme der Client-Benutzereingaben aktiviert. Ein Benutzer kann jederzeit den Floor *anfordern* oder, sofern er der Floor-Besitzer ist, *weitergeben*. Dies führt zur Änderung der Floor List, nicht aber zu einem neuen Zustand. Ein neuer Zustand kann nur erreicht werden durch das Beenden der Kommunikation, das Beantragen eines Zwischenrufs *-ZR-* oder durch das Rufen des Tutors *-rufe Tutor-*.

Ein Zwischenruf *-ZR-* modelliert die Anfrage für eine Zwischenbemerkung. Während einer normalen Diskussion, Zustand 1 (Floor-Besitz durch FL), können die übrigen Teilnehmer eine Zwischenbemerkung beantragen. Dem Floor-Besitzer wird dann angezeigt, dass Client x eine Zwischenbemerkung machen möchte. Gleichzeitig geht das System ohne Änderung der FL in den Zustand 2 (Warte auf Antwort) über. Der Floor-Besitzer kann die Zwischenbemerkung mit *-NAK-* ablehnen, wodurch das System wieder in den Zustand 1 zurückkehrt. Er kann die Zwischenbemerkung aber auch mit *-AK-* zulassen. In diesem Fall geht das System in den Zustand 3 (Floor-Besitz durch ZR) über. Auch dies führt zu keiner Veränderung der FL. Der Floor-Besitz auf der Grundlage einer Zwischenbemerkung schließt nicht das Recht ein, weitere Zwischenrufe zu zulassen. Im Entwurfsmodell wird dies auf der Prozessebene, durch Deaktivieren des entsprechenden Prozesses zur Entgegennahme von ZR-Anforderungen, modelliert. Dadurch ist gewährleistet, dass der Verlauf der Diskussion nicht außer Kontrolle gerät. Wenn die Zwischenbemerkung abgeschlossen ist und der Floor-Besitz mit *-weitergeben-* weitergegeben wird, erfolgt die Fortsetzung der Diskussion entsprechend der ursprünglichen FL.

Jeder Benutzer kann zu jeder Zeit mit *-rufe Tutor-* den Tutor rufen. Dies bewirkt zunächst keine Änderung des Zustandes. Erst wenn der Tutor der Konferenz beitrifft und

den Floor beantragt, geht das System in den Zustand 4 (Warte auf Floor) über. So wird vermieden, dass das System in einen Deadlock gerät, falls der Tutor nicht an seinem Platz ist.

Wenn der Tutor eingeloggt ist, wird dies den übrigen Teilnehmern in der Statusbar angezeigt und die FL wird aktualisiert. Dazu wird der Tutor an die erste Position in der FL und der Diskussionsteilnehmer, der den Tutor rief, an die zweite Position der FL gesetzt. Der dritte Eintrag bleibt frei und kann von einem der anderen Teilnehmer beantragt werden. Mit der nächsten Weitergabe des Floors geht das System dann in den Zustand 5 (Floor-Besitz durch Tutor) über und der Tutor erhält den Floor. Loggt sich der Tutor ein und beantragt den Floor, ohne explizit gerufen worden zu sein, geht das System ebenfalls in den Zustand (4) über, d.h. mit der nächsten Weitergabe des Floors erhält der Tutor diesen auch. Die übrigen Einträge der FL bleiben in diesem Fall zunächst leer.

### 5.2.5 Bewertung der entwickelten Floor-Kontrolle

Die Stärken des entwickelten Verfahrens zur Floor-Kontrolle liegen zunächst in der Garantie einer definierten Fairness und der Verhinderung des gegenseitigen Ausschlusses und der Blockierung. Die Definition der Fairness basiert dabei auf einer theoretischen Gleichverteilung des Floor-Besitzes hinsichtlich der Häufigkeit. Die Einhaltung dieser Art der Gleichverteilung wird jedoch nicht erzwungen. Auf die Entwicklung von Maßnahmen zur Garantie der zeitlichen Gleichverteilung des Floors wurde verzichtet, da die im Gruppenprozessmodell definierten Rollen jeder Form der Gleichverteilung entgegenstehen. Insbesondere enthält das PFCM keine Begrenzung für die Dauer des Floor-Besitzes. Durch die Möglichkeiten der Zwischenbemerkung und des Rufens des Tutors sind jedoch zwei Mechanismen vorgesehen, die im Falle eines übermäßig langen Floor-Besitzes eingesetzt werden können.

Vergleichbare Verfahren zur Floor-Kontrolle, wie das im Rahmen des USMInt-Projekt entwickelte Protokoll IFloor [Sis98] arbeiten ebenfalls listenorientiert, implementieren

aber keine direkten Maßnahmen zur Gewährleistung der Fairness und der Vermeidung des gegenseitigen Ausschlusses. Indirekt ist dies jedoch durch einen Wechsel des Kooperationsmodus möglich. Im kontrollierten Modus kann der Moderator einzelne Teilnehmer aus der Rednerliste entfernen und somit Einfluss, sowohl auf die Fairness als auch den gegenseitigen Ausschluss nehmen. IFloor skaliert im Gegensatz zum PFCM auch für große Teilnehmerzahlen. Bei der Entwicklung des PFCM stand vorrangig die Unterstützung des entwickelten Gruppenprozessmodells im Vordergrund. Dies sieht maximal fünf Teilnehmer in einer Sitzung und die Verwaltung genau eines Floors vor. Eine Übertragung des im Rahmen dieser Arbeit entwickelten PFCMs auf andere Systemkonzepte ist generell möglich, sofern das zugrunde liegende Gruppenprozessmodell ähnliche Prozesse, Rollen und Gruppengrößen vorsieht. Andererseits kann das PFCM als spezieller Kooperationsmodus alle Systeme mit mehreren Kooperationsmodi sinnvoll ergänzen.

### **5.3 Benutzerschnittstelle der PASSENGER Client Applikation**

Die Gestaltung der Benutzerschnittstelle hat großen Einfluss auf die Akzeptanz eines Systems. Die wesentlichen Aspekte beim Benutzerschnittstellendesign betreffen die Auswahl und Gestaltung der Dialogform, des Dialogmodus' und die Dialogtechnik sowie die Funktionalität und die Technologie. Vor dem Hintergrund des Entwurfs einer Mehrbenutzeranwendung erhöht sich nicht nur die Komplexität bei der Gestaltung dieser Aspekte, sondern es kommen noch zusätzliche Aspekte im Zusammenhang mit der Gruppenwahrnehmung und der Realisierung des gemeinsamen Kontext' hinzu. Die Berücksichtigung der in Kapitel 5.1.2 beschriebenen Ausprägungen der Phasen der Orientierungs- und Vertrauensbildung sichern zudem eine Entwicklungskonformität hinsichtlich des geplanten Einsatzszenarios wie es in den Zielen dieser Arbeit gefordert ist.

### 5.3.1 Grundlegende Designentscheidungen

Die in Kapitel 4.4 definierten Anforderungen an eine Groupware zum Einsatz im Software-Engineering-Praktikum legen bereits grob die Funktions- und Anwendungsklassen des PASSENGER Clients fest. Demnach müssen beim Entwurf im Wesentlichen konzipiert werden:

- die Kommunikationskomponente zur synchronen, audiovisuellen Kommunikation sowie
- die Kooperationskomponente, bzw. der gemeinsame Arbeitsbereich.

Die erste Designentscheidung bezüglich des Layouts betrifft das Fenstersystem. Hier wird festgelegt, dass die beiden Komponenten zur Kommunikation und zur Kooperation in getrennten Fenstern realisiert werden. Eine Besonderheit dabei ist die Festlegung, dass diese sich nicht gegenseitig überdecken können. Diese Entscheidung resultiert zum einen aus der Anforderung, dass alle Videofenster gleichzeitig dargestellt werden sollen und zum anderen soll sie dazu beitragen, das Zusammengehörigkeitsgefühl zu maximieren.

Eine weitere grundlegende Entscheidung betrifft die maximale Anzahl der Teilnehmer. Für die Mindestgruppengröße ergibt nach der Definition der Gruppenarbeit in Kapitel 2.1 der Wert zwei. Für die Obergrenze wird zunächst die Gruppengröße des Standardpraktikums angesetzt, diese beträgt fünf Teilnehmer. In beiden Fällen muss der Tutor hinzu gerechnet werden, so dass für die Teilnehmerzahl  $T$  gilt:  $3 \leq T \leq 6$ . Aus technischer Sicht wird die Teilnehmerzahl durch die Größe der Videofenster und des Monitors sowie durch die Auflösung des Monitors bestimmt. Übliche Grafikprogramme reservieren - bei einer Auflösung von  $1024 \times 768$  Pixel - ca. 560 Pixel für die Ausdehnung des Arbeitsbereichs in der Vertikalen. Unter diesen Voraussetzungen<sup>30</sup> ergibt sich, bei Anordnung der Videofenster in der Horizontalen, eine maximale Teilnehmerzahl

---

<sup>30</sup> Die üblicherweise ebenfalls vorhandenen Komponenten Menüleiste und Statuszeile werden bei den stark idealisierten Betrachtungen nicht berücksichtigt.

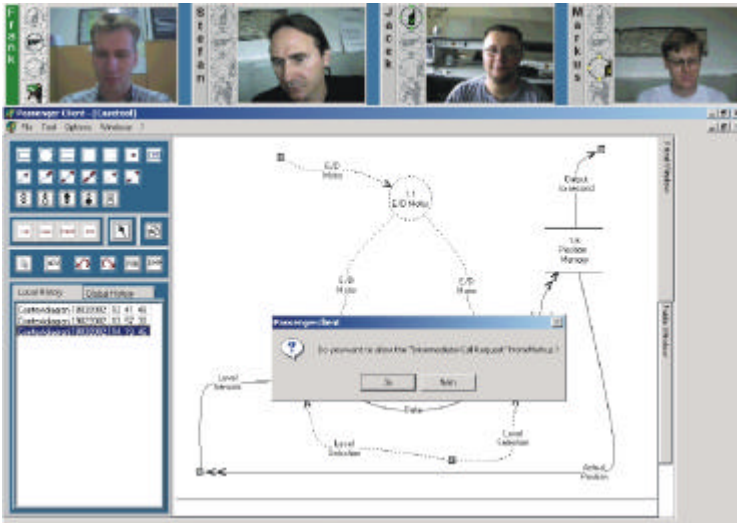


von vier, bei einer Größe der Kommunikationsfenster von  $277 \times 208$  Pixel<sup>31</sup>. In dem gemeinsamen Arbeitsbereich kann dabei ein DIN A4 Blatt im Querformat mit maximal  $792 \times 560$  Pixeln dargestellt werden. Auf die gleiche Weise ergibt sich bei Anordnung von vier Kommunikationsfenstern in der Vertikalen für ein einzelnes Kommunikationsfenster eine Größe von  $256 \times 192$  Pixeln und für die Darstellung eines DIN A4 Blattes im Querformat eine maximale Größe von  $760 \times 537$  Pixeln.

Nach den oben durchgeführten Kalkulationen ergibt sich im Hinblick auf eine Maximierung nach der Größe ein Vorteil für die Anordnung der Kommunikationsfenster in der Horizontalen. Dabei würde eine weitere Vergrößerung der Teilnehmerzahl durch die daraus resultierende Verkleinerung der Kommunikationsfenster einzelne Aspekte der Kommunikation negativ beeinflussen. Demgegenüber ist die Ausdehnung des gemeinsamen Arbeitsbereichs in der Vertikalen zunächst ausreichend und erfordert in erster Näherung keine Vergrößerung. Der bestimmende Faktor bleibt demnach die maximale Teilnehmerzahl  $T_{\max}$ , für die nun  $4 \leq T_{\max} \leq 6$  gilt. Bei der Analyse des Praktikums wurde beobachtet, dass einzelne Gruppen sowohl bei Ausfall eines, als auch bei Ausfall zweier Mitglieder durchaus brauchbare Ergebnisse lieferten. Allerdings war in diesem Fall die Arbeitslast der einzelnen Mitglieder deutlich höher, aber durchaus vertretbar. Im Weiteren wird nun von einer Anordnung der Videofenster in der Horizontalen und einer maximalen Teilnehmerzahl von vier ausgegangen. Die aus den oben angeführten Überlegungen resultierende Implementation der Benutzerschnittstelle ist in Abbildung 24 gezeigt.

---

<sup>31</sup> Hier wird ebenfalls ein Seitenverhältnis von 4:3 angenommen.

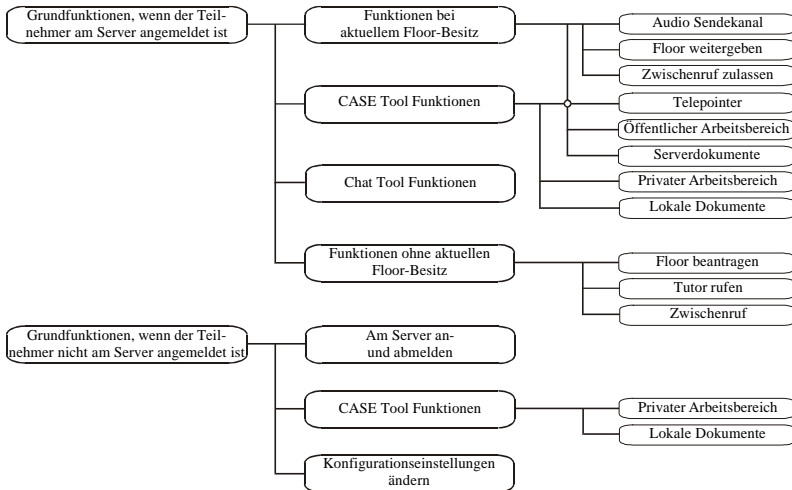


**Abb. 24:** Benutzerschnittstelle des PASSENGER-Client

Die Ermittlung der maximalen Teilnehmerzahl durch eine Betrachtung anderer Kriterien und Einflussfaktoren, wie z.B. der Bemessung der Arbeitslast oder eine Untersuchung des Einflusses der Fenstergröße auf das Kommunikations- und Kooperationsverhalten bleibt Analysen im Rahmen weiterführender Arbeiten überlassen.

### 5.3.2 Funktionsumfang

Der Funktionsumfang der oben dargestellten Benutzerschnittstelle lässt sich in erster Näherung in zwei Funktionsklassen einteilen, abhängig davon, ob der Benutzer an einer Sitzung teilnimmt oder nicht. In Abhängigkeit davon ergeben sich weitere Funktionsklassen die in Abbildung 25 dargestellt sind.



**Abb. 25:** Funktionsklassen

Der Zugriff auf das Whiteboard bzw. das CASE Tool ist jederzeit, allerdings mit unterschiedlichen Funktionsumfängen möglich. Eine Manipulation der Dokumente im öffentlichen Arbeitsbereich und deren Verwaltung auf dem Server sowie der Zugriff auf den Telepointer erfordert nicht nur, dass der Benutzer am Server angemeldet ist, sondern darüber hinaus, dass er der aktuelle Floor-Besitzer ist. In allen anderen Fällen hat der Benutzer nur Vollzugriff auf den privaten Arbeitsbereich und die lokalen Dokumente, bzw. im Falle einer gültigen Sitzung ohne Floor-Besitz kann er sich jederzeit den Inhalt des öffentlichen Arbeitsbereiches ansehen. Der Zugriff auf das Chat Tool ist möglich, sobald der Benutzer am Server angemeldet ist.

## 5.4 Synchrone Kommunikation mit dem PASSENGER-Client

Die synchrone Kommunikation mit Hilfe des PASSENGER-Client kann entsprechend den in Kapitel 4.4 formulierten Anforderungen über verschiedene Kommunikationskanäle erfolgen. Die Verwendung des Chat-Werkzeuges ist dabei unabhängig von der

Floor-Kontrolle und soll in erster Linie helfen, verschiedene Problemsituationen zu entschärfen, z.B. ordentliche Beendigung einer Sitzung in Folge schlechter Bandbreitenverhältnisse oder Hinweis zur Abgabe des Floors im Falle eines übermäßig langen Floor-Besitzes. Darüber hinaus besteht die Möglichkeit den Videokanal zu deaktivieren und im Falle schlechter Übertragungsverhältnisse eine Audiokonferenz durchzuführen.

### 5.4.1 Designentscheidungen

Die oben beschriebenen grundlegenden Designentscheidungen zur Anordnung der Kommunikationsfenster resultieren im Wesentlichen aus technischen Überlegungen zur Dimensionierung der Fenstergröße. Zusätzliche Designentscheidungen berücksichtigen in stärkerem Maße die problemorientierte Sicht. Im Sinne einer Steigerung des Zusammengehörigkeitsgefühls sind insgesamt drei weitere Designentscheidungen getroffen worden.

- **Keines der Videofenster kann in Größe und Position verändert werden.**  
Durch diese Entscheidung soll verhindert werden, dass einzelnen Teilnehmern, bewusst oder unbewusst, unterschiedliche visuelle Prioritäten zugewiesen werden können.
- **Keines der Videofenster kann durch andere Elemente der Bildschirmoberfläche überdeckt werden.**  
Hierdurch soll verhindert werden, dass einzelne Teilnehmer, bewusst oder unbewusst, in ihrer Stellung innerhalb der Gruppe gegenüber anderen Teilnehmern zurückgesetzt werden.
- **Die Anordnung der Teilnehmer in den Videofenstern ist je Client immer gleich und folgt für alle Clients dem gleichen Muster.**  
Hierdurch wird erwartet, dass sich schneller ein Gruppengefühl entwickeln kann. Jeder Teilnehmer erscheint auf seinem Rechner immer im linken Fenster, der Tutor immer im rechten Fenster. Die übrigen Teilnehmer werden anhand einer vom Server






vergebenen ID-Nummer in den übrigen beiden Fenstern, immer an der gleichen Position dargestellt. Fehlt ein Teilnehmer, so bleibt sein Fensterbereich leer.

Den Kommunikationsfenstern kommt insgesamt eine große Bedeutung zu, da sie neben der Darstellung des Videobildes auch den Zugriff auf die oben beschriebenen Funktionen zur Vergabe des Floors und damit auch zur Steuerung der Kommunikation ermöglichen. Darüber hinaus werden die wesentlichen Informationen zur Gruppenwahrnehmung im Kommunikationsfenster dargestellt. In dem Zusammenhang wurden eine Reihe von Piktogrammen und Schaltflächen entworfen und im Kommunikationsfenster angeordnet. Der Entwurf dieser Piktogramme ist in der Diplomarbeit von Frank Schwarz beschrieben [Schw98]. Die folgende Abbildung 26 zeigt die Anordnung der Schaltflächen im Kommunikationsfenster.



**Abb. 26:** Kommunikationsfenster des PASSENGER-Clients










Über die Schaltflächen erfolgt die Manipulation des PFCM. Einen Überblick über die prinzipiell über die Schaltflächen zugreifbaren Funktionen gibt Tabelle 4.

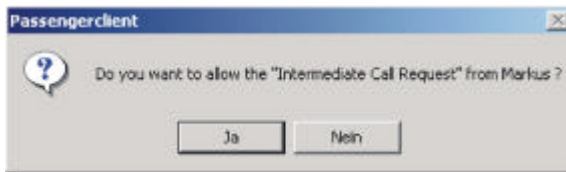
	Funktion der Schaltflächen
	Darstellung des Benutzernamens
	Schaltfläche zur Beantragung des Floors
	Schaltfläche zur Weitergabe des Floors
	Schaltfläche zur Beantragung eines Zwischenrufs
	Schaltfläche zum Ruf des Tutors

**Tabelle 4:** Funktion der Schaltflächen



Die in Tabelle 4 angegebenen Funktionen sind grundsätzlich nur aktiv, wenn der Teilnehmer Mitglied einer aktiven Sitzung ist, bzw. wenn er am Server angemeldet ist. Darüber hinaus wird, wie in Abbildung 25 dargestellt, weiter unterschieden in Funktionen, die immer zugreifbar sind, also auch wenn der Teilnehmer nicht der aktuelle Floor-Besitzer ist, und solche, die nur zugreifbar sind wenn der Teilnehmer der aktuelle Floor-Besitzer ist. Die Piktogramme aktiver Schaltflächen sind farbig dargestellt, wohingegen eine graue Darstellung der Piktogramme kennzeichnend für eine inaktive Schaltfläche ist.

#### 5.4.2 Ablauf der Kommunikation

Die Kommunikation erfolgt entsprechend dem in Abbildung 23 dargestellten PFCM. Sobald die Teilnehmer Mitglied einer gültigen Sitzung sind, empfangen sie alle gesendeten Audio- und Videodatenströme. Ebenso haben sie über die Schaltfläche  und  Zugriff auf alle Funktionen, die keinen Floor-Besitz voraussetzen, vgl. auch Abbildung 25. Sobald der erste Teilnehmer über die Schaltfläche  den Floor beantragt hat, wird er ihn auch erhalten. Danach wird bei ihm die Schaltfläche  zur Weitergabe des Floors aktiv und die Schaltfläche  wird inaktiv. Bei allen anderen Teilnehmern ist nun zusätzlich Schaltfläche  aktiv. Wenn nun ein weiterer Teilnehmer den Floor über die Schaltfläche  beantragt, bekommt er diesen auch zugeteilt, sobald der aktuelle Floor-Besitzer diesen durch Drücken der Schaltfläche  weitergibt. Sobald ein Teilnehmer über die Schaltfläche  eine Zwischenbemerkung beantragt, erhält der aktuelle Floor-Besitzer eine entsprechende Meldung, vgl. Abbildung 27.



**Abb. 27:** Zwischenruffenster


Der aktuelle Floor-Besitzer kann nun die Zwischenbemerkung zulassen oder ablehnen. Lässt er die Zwischenbemerkung zu, so wird die Schaltfläche  zur Weitergabe des Floors bei ihm inaktiv und bei dem neuen Floor-Besitzer aktiv. Gleichzeitig wird bei den übrigen Teilnehmern Schaltfläche  inaktiv.



An den Floor-Besitz ist neben dem Zugriff auf den Audio-Sendekanal auch die exklusive Berechtigung zum Editieren der aktuellen Gruppendokumente geknüpft sowie die Berechtigung zur Benutzung des Telepointers. Der Zugriff auf diese Werkzeuge erfolgt im gemeinsamen Arbeitsbereich und wird in Abschnitt 5.5 beschrieben.

### 5.4.3 Gruppenwahrnehmung

Ein weiterer Schwerpunkt der vorliegenden Arbeit lag in der Definition und Umsetzung geeigneter Maßnahmen zur Gruppenwahrnehmung. Deren Umsetzung erfolgt sowohl über Darstellungen im Kommunikationsfenster als auch über den gemeinsamen Kontext. Im Folgenden werden in dem Zusammenhang die Kommunikationsfenster beschrieben. Eine Beschreibung des gemeinsamen Kontexts erfolgt weiter unten.

Die Gruppenwahrnehmung innerhalb der Kommunikationsfenster erfolgt über unterschiedliche farbliche Darstellungen der Schaltflächen und des Namensfeldes. Dabei wird unterschieden zwischen dem eigenen Kommunikationsfenster auf der lokalen Maschine und denen der übrigen Teilnehmer.

Die Benutzernamen werden in den einzelnen Kommunikationsfenstern in einem Namensfeld am linken Rand dargestellt, vgl.  in Abbildung 26. Über die stets gleiche Anordnung der Kommunikationsfenster auf jeder lokalen Maschine soll das Zusammengehörigkeitsgefühl der Gruppe sowie die Ausbildung von Beziehungsaspekten und Partnerbildern gefördert werden. Gleichzeitig sollen durch eine farbliche Hinterlegung des Namensfeldes Informationen über die aktuelle Rolle des Teilnehmers dargestellt werden. Die folgende Tabelle gibt hierzu einen Überblick.

Rolle	Farbe des Namensfeldes	Farbe der Schaltflächen zur Beantragung des Floors
Bearbeiter= aktueller Floor-Besitzer	Grün oder Gelb	-----
Aktiver Teilnehmer	Grau	und  = Grün oder Grau
Passiver Teilnehmer	Grau	und  = Gelb Blau

**Tabelle 5:** Wahrnehmung der Rollen (Momentaufnahme)

Die Rolle des aktiven Teilnehmers können mehrere Teilnehmer gleichzeitig einnehmen: zum einen Teilnehmer, die den Floor beantragt haben, als auch Teilnehmer, die eine Zwischenbemerkung beantragt haben.

Die übrigen Zustände der PL werden über farbliche Kennzeichnungen der Schaltflächen visualisiert. Hierzu gibt die folgende Tabelle einen Überblick über die Darstellung im Kommunikationsfenster des lokalen Benutzers, vgl. Abbildung26:

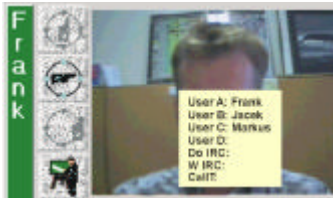


Gruppenwahrnehmung über Farbkodierung				
	Grau	Grün	Gelb	Blau
1	Passiver Teilnehmer ohne Floor-Besitz	Aktueller Floor-Besitzer	Aktueller Floor-Besitzer aufgrund eines Zwischenrufs	-----
2	Inaktiv, da Floor-Besitzer	Floor-Besitz beantragt	-----	Floor kann beantragt werden
3	Inaktiv, da nicht aktueller Floor-Besitzer	-----	-----	Aktiv, da aktueller Floor-Besitzer.
4	Inaktiv, da aktueller Floor-Besitzer	-----	Zwischenruf beantragt	Zwischenbemerkung kann beantragt werden
5	Inaktiv, da Tutor nicht eingeloggt	Aktiv, Tutor kann gerufen werden.	-----	-----

**Tabelle 6:** Gruppenwahrnehmung über das eigene Kommunikationsfenster

Die Darstellung der Schaltflächen in den Kommunikationsfenstern der übrigen Teilnehmer ist leicht unterschiedlich. Dies betrifft im Wesentlichen die Bedeutung der grau hinterlegten Schaltflächen. Der lokale Benutzer bekommt über eine graue Hinterlegung der Schaltflächen angezeigt, dass die korrespondierende Funktion inaktiv ist. In den Kommunikationsfenstern der entfernten Teilnehmer wird dem lokalen Benutzer durch eine graue Hinterlegung der Schaltflächen angezeigt, dass der korrespondierende Zustand nicht belegt ist.

In den Kommunikationsfenstern wird maximal der aktuelle Floor-Besitzer und der potenzielle nächste Floor-Besitzer dargestellt. Auf eine vollständige Visualisierung der Permission List wurde verzichtet, um nicht weitere Farben darstellen und deren Bedeutung interpretieren zu müssen. Den Teilnehmern ist jedoch jederzeit der Zugriff auf die vollständige Permission List möglich. Hierzu muss der Benutzer die Maus im Kommunikationsfenster platzieren. Daraufhin wird ihm die Belegung der gesamten Permission List angezeigt, vgl. Abbildung 28.



**Abb. 28:** Visualisierung der Permission-List-Einträge

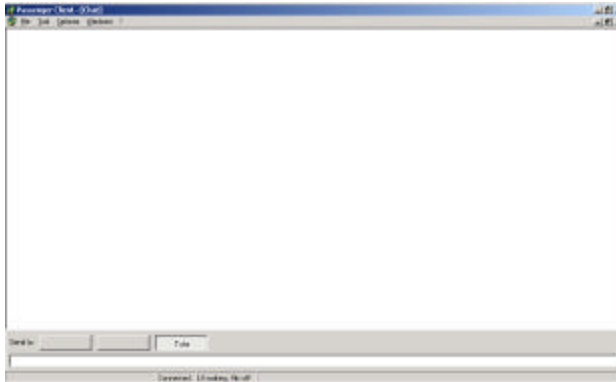
In ähnlicher Weise kann jeder Teilnehmer über einen rechten Mausklick in sein eigenes Kommunikationsfenster die Liste der ihm aktuell zur Verfügung stehenden Funktionen einsehen und alternativ zur Aktivierung über die Schaltflächen auch aufrufen.



**Abb. 29:** Visualisierung der verfügbaren Funktionen

#### 5.4.4 Chat

Über die Werkzeugleiste ist unabhängig vom Protokoll zur Floor-Kontrolle jederzeit der Zugriff auf ein Chat-Werkzeug möglich. Dieses ist als Kommunikationsmittel in Ausnahmesituationen implementiert und soll nicht zur Standardkommunikation verwendet werden. Die Funktionen sowie die Bedienung orientieren sich an üblichen Chat-Werkzeugen und werden deshalb nicht weiter beschrieben.



**Abb. 30:** Integriertes Chat-Werkzeug im PASSENGER Client

#### 5.4.5 Bewertung des Benutzerschnittstellenentwurfs

In diesem Abschnitt wurde die Benutzerschnittstelle der synchronen Groupware PASSENGER beschrieben. Der Entwurf erfolgte auf der Basis der allgemeinen Anforderungen, wie sie in Kapitel 4.4 beschrieben sind, und speziellen Anforderungen, wie sie sich aus dem analysierten Gruppenverhalten ergeben. Insbesondere wurde bei allen Entwurfsentscheidungen die Forderung nach Maßnahmen zur Steigerung des Zusammengehörigkeitsgefühls berücksichtigt. Hierzu wurde für die Anordnung der Videofenster ein Konzept zur Positionierung und zur Größenvarianz erarbeitet und implementiert.

Insbesondere wurden bei der Gestaltung der Benutzerschnittstelle Lösungen zur Floor-Kontrolle und zur Gruppenwahrnehmung erarbeitet. Die entwickelte Floor-Kontrolle ist speziell auf kleine, geschlossene Gruppen zugeschnitten. Sie behebt die in Kapitel 4.3.5 beschriebenen Mängel existierender Systeme im Hinblick auf die Fairness bei der Floor-Vergabe und die implementierten Maßnahmen zur Vermeidung des gegenseitigen Ausschlusses.

Darüber hinaus bietet das in dieser Arbeit implementierte Verfahren zur Visualisierung der Permission List vielfältige Informationen zur Gruppenwahrnehmung im Bezug auf die aktuelle Sitzungssituation.

Schließlich bietet die Entkopplung der Kommunikations- und Kooperationsfenster - innerhalb des PASSENGER Clients - nicht nur einen wichtigen Beitrag zur Steigerung des Zusammengehörigkeitsgefühls der Teilnehmer, sondern ermöglicht auch, die Kommunikationskomponente als eigenständiges Werkzeug zur Videokonferenz einzusetzen. Darüber hinaus können die erarbeiteten Konzepte auch bei der Entwicklung von Desktop-Videokonferenzsystemen ohne besondere Betonung der Groupware-Funktionalität eingesetzt werden.

## **5.5 Kooperation mit dem PASSENGER CASE Tool**

Im Folgenden wird die Konzeption des gemeinsamen Arbeitsbereichs, bzw. die Konzeption der Kooperationskomponente beschrieben, sowie die implementierten Maßnahmen zur Gruppenwahrnehmung und die Funktionsweise erläutert. Insgesamt handelt es sich um die Konzeption eines Whiteboard, das insbesondere den kooperativen Software-Entwurf im Rahmen eines Praktikums unterstützen soll. Einige der im Folgenden diskutierten Konzepte und Lösungen können verallgemeinert und grundsätzlich bei der Entwicklung eines Whiteboard berücksichtigt werden. In Fällen, in denen es sich um Konzepte und Lösungen handelt, die sich aus dem speziellen Anwendungskontext des Software-Engineering ergeben, wird das daraus resultierende Werkzeug als PASSENGER CASE Tool bezeichnet.

### **5.5.1 Konzeption**

Die kreativen Aufgaben innerhalb des Gruppenprozesses, bzw. die gemeinsame Bearbeitung der Artefakte erfolgt im gemeinsamen Arbeitsbereich. Die wesentlichen Anforderungen an die Kooperationskomponente wurden bereits in Abschnitt 4.4 vorgestellt,

die Anwendung des NetMVC-Konzeptes ist in Abschnitt 5.2.2 beschrieben und das entwickelte Modell zur Floor-Kontrolle in Abschnitt 5.2.5.

Die Verteilungsarchitektur der synchronen Groupware PASSENGER (vgl. 5.2.3) wurde unter anderem gewählt um private Sichten zu ermöglichen. Diese Möglichkeit soll im Rahmen dieser Arbeit umgesetzt werden, um den Studierenden die zeitgleiche Erarbeitung individueller Lösungen zu ermöglichen. Die private Sicht basiert nicht a priori auf den lokalen Modelldaten.

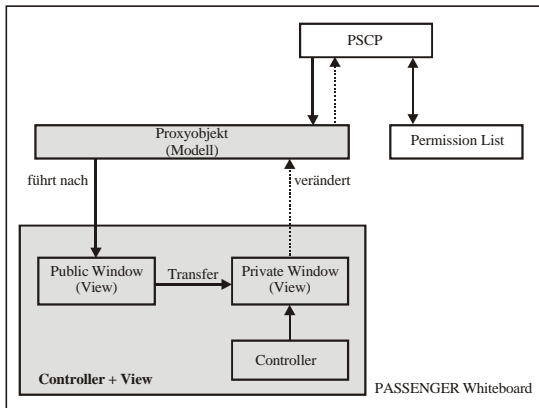
Das NetMVC-Konzept wurde gewählt, um aufbauend auf lokalen Modelldaten individuelle Sichten zu ermöglichen. Im Rahmen der Entwicklung eines Prototypen der synchronen Groupware PASSENGER wurde die grafische Sicht auf die Modelldaten als die einzige individuelle Sicht implementiert.

Der öffentliche Fensterbereich des Whiteboard soll von der Floor-Kontrolle verwaltet werden. Für die Realisierung der privaten und öffentlichen Sichten existieren verschiedene Möglichkeiten, vgl. [Krc96] mit mehr oder weniger komplexen Mechanismen zur Sicherung der Konfliktfreiheit. In dem Zusammenhang wurde bereits bei der Entwicklung der Floor-Kontrolle festgelegt, dass der schreibende Zugriff auf den gemeinsamen Arbeitsbereich zu jedem Zeitpunkt immer nur für einen Teilnehmer, den Floor-Besitzer, möglich ist. Um den Teilnehmern dennoch individuelles Arbeiten zu ermöglichen, ist die Realisierung einer privaten Sicht in einem eigenen Fensterbereich erforderlich.

Die Realisierung der privaten und der öffentlichen Sicht in getrennten Fenstern sieht üblicherweise vor, dass bei einem Wechsel des Floor-Besitzes das zuletzt bearbeitete Artefakt als Gruppendokument im öffentlichen Fenster verbleibt. Die Übernahme eines Artefaktes aus dem privaten Fenster muss explizit erfolgen.

Im Rahmen dieser Arbeit wurde eine Alternative entwickelt, die die Nutzung der Fenster stärker an das Kollaborations-Modell, hier das PFCM, koppelt. Die Floor-Kontrolle bedingt, dass nur der Floor-Besitzer das Artefakt bearbeiten kann und die übrigen Teilnehmer diese Änderungen verfolgen können. Dementsprechend wird das öffentliche

Fenster im Rahmen dieser Arbeit als das darstellende Fenster interpretiert und konsequenterweise das private Fenster als das Arbeitsfenster. Dementsprechend ist das private Fenster des Floor-Besitzers für Änderungen des lokalen und in der Folge auch des globalen Modells verantwortlich. Die daraus resultierenden Veränderungen der Modell-daten werden im öffentlichen Fenster nachgeführt. Für den Floor-Besitzer ist durch diese Trennung das öffentliche Fenster zunächst obsolet, und ein Nachführen der Darstellung ist für den Bearbeitungsprozess nicht nötig. Das Nachführen kann über die Floor-Kontrolle leicht unterbunden werden. Daraus ergibt sich der Vorteil, dass bei einem Wechsel des Floor-Besitzes das zuletzt bearbeitete Artefakt weiterhin im Darstellungsfenster und das aktuelle Artefakt im Arbeitsfenster des Floor-Besitzers dargestellt werden. Ein Wechsel zwischen beiden Artefakten ist ohne Netzwerkzugriff möglich. Abbildung 31 zeigt schematisch dieses Whiteboard-Konzept.

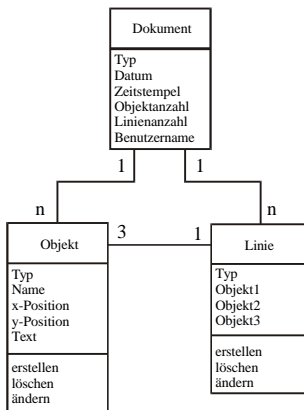


**Abb. 31:** Whiteboard-Konzept der synchronen Groupware PASSENGER

Die gestrichelten Datenflüsse in Abbildung 31 markieren solche, die in Abhängigkeit vom Floor-Besitz ermöglicht oder unterbunden werden. Die Übernahme des Inhalts des öffentlichen Fensters in das private Fenster ist jederzeit für jeden Teilnehmer möglich.

## 5.5.2 Datenmodell und Dateiformat

Das implementierte Whiteboard zeichnet sich durch seine spezielle Unterstützung für den Software-Entwurf nach Ward & Mellor [War91] aus. Ein Dokument ist entsprechend dieser Notation entweder vom Typ Kontextdiagramm, Datentransformation, Zustandsdiagramm, Modulmodell, Prozessmodell oder Taskmodell. Neben dieser Information wird jedes Dokument durch einen Zeitstempel (Datum und Uhrzeit), die Anzahl der in ihm dargestellten Objekte und Verbindungen sowie den Namen des Benutzers, der es auf dem Server abgespeichert hat, beschrieben. Abbildung 32 zeigt das vollständige Objektmodell der Whiteboard-Dokumente.

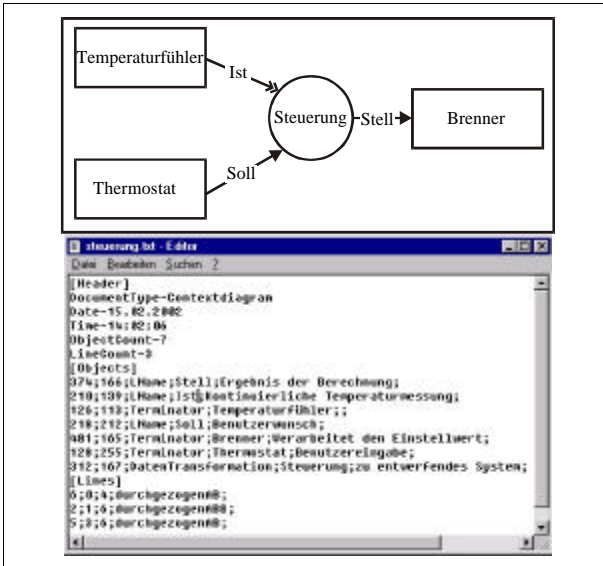


**Abb. 32:** Objektmodell der Whiteboard-Dokumente

Jedes Dokument kann mehrere grafische Objekte enthalten, die über Linien miteinander verbunden sind. Dabei ist die Bezeichnung der Linie als eigenes Objekt vom Typ LName definiert. Daraus resultiert, dass die Verbindungslinie zweier Objekte immer über ein drittes Objekt läuft und bei beliebigen Positionsänderungen der Objekte der Verlauf der Linie auf einfache Weise neu berechnet werden kann.

Das Modell wird auf dem Server in einer ASCII-Datei (American Standard Code for Information Interchange) abgelegt und gleichzeitig an alle Clients repliziert. Die folgende

Abbildung zeigt beispielhaft die grafische Darstellung im öffentlichen Fenster als spezielle Sicht auf dieses Modell sowie das zugehörige Datenformat des Modells.



**Abb. 33:** Grafische Repräsentation und zugehöriges Datenformat

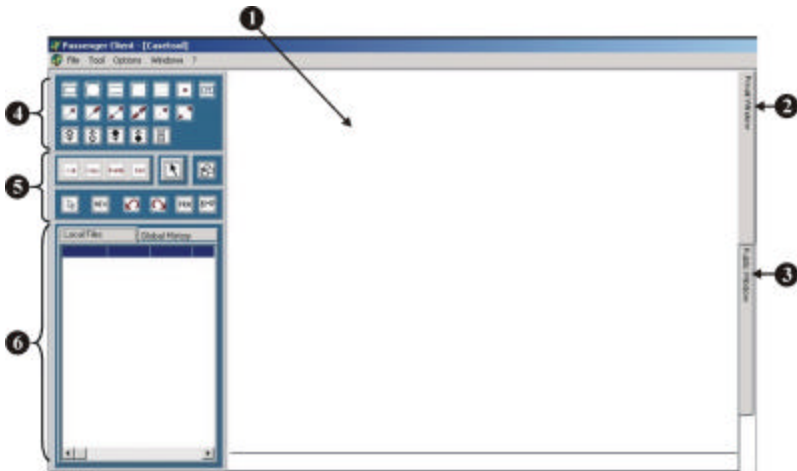
Die Summe der modellierten Daten ist ausreichend zur grafischen Darstellung aber auch zur Darstellung anderer Sichten, z.B. textuelle Darstellung als Datenwörterbuch. Die Implementation eines Codegenerators würde darüber hinaus Beziehungen zwischen verschiedenen Dokumententypen eines Entwurfs erfordern. Das in Abbildung 33 gezeigte Objektmodell bildet auf Grund seiner Erweiterungsfähigkeit auch hierfür die Basis. Der implementierte Prototyp macht von diesen Möglichkeiten jedoch keinen Gebrauch und beschränkt sich auf die grafische Repräsentation der Modelldaten.

### 5.5.3 Funktionsweise

Der Start der Kooperationskomponente erfolgt aus der Programmleiste der synchronen Groupware PASSENGER heraus und ist sowohl im Online- als auch im Offline-Betrieb



möglich. Abbildung 34 zeigt die Benutzerschnittstelle des PASSENGER CASE Tools unmittelbar nach dem Programmstart.




**Abb. 34:** Benutzerschnittstelle des PASSENGER-Whiteboard

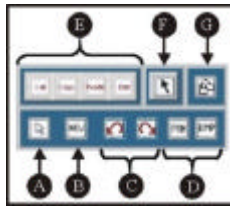
Im Arbeitsbereich ① werden die Dokumente angezeigt bzw. bearbeitet. Dabei gilt grundsätzlich, dass die Erstellung und Bearbeitung im Private Window ② erfolgt. Dies gilt sowohl für Online- als auch für Offline-Sitzungen. Wenn der Teilnehmer in einer Online-Sitzung ist und er darüber hinaus auch der aktuelle Floor-Besitzer ist, wird der Inhalt seines Private Window im Public Window ③ der übrigen Teilnehmer dargestellt. Diese haben weiterhin die Möglichkeit ihr Privat Window für eigene Ideen zu nutzen. Die Umschaltung zwischen den beiden Fenstern erfolgt durch einen Mausklick auf die jeweilige Registerkarte.

In der Oberfläche sind mehrere Schaltflächen implementiert, vgl. die Bereiche ④ bis ⑥ über die eine Reihe von Standardfunktionen aufgerufen werden können.

Im Bereich ④ stehen die durch die Software-Entwurfsmethode nach Ward & Mellor definierten grafischen Symbole zur Verfügung. Diese werden durch einen einfachen





Mausklick ausgewählt und durch einen weiteren Mausklick im Arbeitsbereich dort platziert. Über einen rechten Mausklick auf ein Objekt können dessen Eigenschaften verändert werden.




Im Bereich  stehen einerseits Standardfunktionen, wie sie aus üblichen Grafikprogrammen bekannt sind, zur Verfügung und andererseits Funktionen die sich aus dem Groupware-Charakter der Anwendung ergeben. Bild 35 zeigt eine Ausschnittsvergrößerung dieses Bereichs.



**Abb. 35:** Funktionsschaltflächen des Whiteboard

Den Schaltflächen in der unteren Reihe der Abbildung 35 sind Standardfunktionen wie folgt hinterlegt:


- : zur Objektauswahl;
- : zur Erzeugung eines neuen Dokuments im Private Window;
- : die Funktionen Undo und Redo;
- : Ausgabefunktionen für den Druck und die Ausgabe im Bitmap-Format;

Die Standardfunktionen Cut, Copy, Paste und Delete sind einerseits über die Schaltflächen  sowie über die Standard-Shortcuts implementiert. Über die Schaltfläche  kann, vorausgesetzt der Teilnehmer ist der aktuelle Floor-Besitzer, der Telepointer aktiviert bzw. deaktiviert werden und über die Schaltfläche  kann der Inhalt des Public Window in das Private Window kopiert werden.

Die erstellten Dokumente können jederzeit im lokalen Arbeitsbereich eines Teilnehmers abgelegt werden. In einer Online-Session werden die erzeugten Dokumente bei jedem Wechsel des Floor-Besitzers auf dem Server gespeichert.

### 5.5.4 Gruppenwahrnehmung

Innerhalb des Whiteboard erfolgt die Gruppenwahrnehmung über den gemeinsamen Kontext. Dieser wird im Wesentlichen durch das Public Window, bzw. durch das gemeinsame Material - das Gruppendokument - erzeugt. Das aktuelle Gruppendokument wird im öffentlichen Fenster dargestellt und in der Regel vom aktuellen Floor-Besitzer bearbeitet. Dieser wird in der Kommunikationskomponente explizit ausgewiesen, so dass die aktuellen Aktionen, also die Wirkung, unmittelbar ihrer Ursache zugeordnet werden können.

Auch die übrigen Dokumente der Gruppe sind in der Dokumentenverwaltung, Bereich  in Abbildung 34, jederzeit zugreifbar. Die globale Dokumentenverwaltung aller auf dem Server abgelegten Gruppendokumente erfüllt außerdem die Anforderungen an eine Dokumentenhistorie, vgl. Abbildung 36.



Local History	Global History
Stelethronitondisagiam	30012002 10. 2
Contealdisagiam	30012002 10. 4
Contealdisagiam	30012002 10. 4
Contealdisagiam	30012002 12. 0
Contealdisagiam	30012002 12. 0
Contealdisagiam	30012002 12. 0
Contealdisagiam	30012002 12. 1
Contealdisagiam	30012002 12. 3
Contealdisagiam	30012002 13. 0
Stelethronitondisagiam	14022002 13. 1
Stelethronitondisagiam	15022002 13. 1
Contealdisagiam	30012002 13. 4
Stelethronitondisagiam	16012002 17. 2

**Abb. 36:** Globale und lokale Dokumentenverwaltung

In dieser Historie sind zu jedem Diagramm der Dokumententyp, das Speicherdatum (im Format TTMMJJJJ) und die Uhrzeit der letzten Speicherung angegeben. Diese Einträge werden automatisch auf der Basis der Modelldaten erzeugt und erfüllen damit die Forderung nach einer Entlastung der Studierenden bei der Verwaltung der Dokumente. Darüber hinaus besteht für verspätet zu einer Sitzung hinzu gekommene Teilnehmer oder den Tutor die Möglichkeit, den Gruppenfortschritt nachzuvollziehen. Durch einen Doppelklick auf ein Dokument in der globalen Dokumentenverwaltung kann dieses vom Server in das Privat Window geladen werden. Einträge aus der lokalen Dokumentenverwaltung können durch Auswahl der Option "Upload to Server" in Folge eines rechten Mausklicks auf den Server geladen werden. Diese Funktionen stehen auch außerhalb der gemeinsamen Sitzungen zur Verfügung und unterstützen damit zusätzlich die Forderung nach einem asynchronen Arbeitsmodus.

Als kollaborativer Dienst ist ein Telepointer implementiert, den der aktuelle Floor-Besitzer benutzen kann. Er kann damit im Verlauf einer Diskussion die Aufmerksamkeit der übrigen Teilnehmer auf bestimmte Objekte oder Bildausschnitte lenken.

### 5.5.5 Bewertung des Whiteboard-Konzeptes

Das im Rahmen dieser Arbeit konzipierte und implementierte PASSENGER CASE Tool für den Software-Entwurf zeichnet sich durch sein Konzept zur Realisierung des privaten und öffentlichen Arbeitsbereichs und die prozessspezifische Unterstützung für das Software-Engineering aus. Die getrennte Realisierung des Arbeits- und Darstellungsbereichs ermöglicht dem Floor-Besitzer zunächst grundsätzlich den gleichzeitigen Zugriff auf die beiden letzten Entwurfsdokumente. Er kann so auf einfache Weise die Entwürfe durch einen Wechsel zwischen Arbeits- und Darstellungsfenster vergleichen. Die übrigen Teilnehmer haben im Grunde dieselbe Möglichkeit, vorausgesetzt, sie übernehmen rechtzeitig den Inhalt des Darstellungsfensters in das Arbeitsfenster. Die Übernahme beliebiger Dokumente in das Arbeitsfenster des Floor-Besitzers - und damit in das Darstellungsfenster der übrigen Teilnehmer - sollte grundsätzlich nicht unkommentiert erfolgen, weshalb die Bearbeitungsrechte und der Zugriff auf den Sprachkanal

unter eine gemeinsame Floor-Kontrolle gestellt werden. Die konsequente Umsetzung einer Client/Server Architektur auf der Basis des NetMVC-Konzepts erfüllt damit einerseits die Anforderungen nach Ausfallsicherheit und nach schnellen Reaktionszeiten. Andererseits besteht darüber hinaus die Möglichkeit zur zentralen Verwaltung der Gruppendokumente und damit auf einfache Weise die Implementation einer globalen Dokumenten-History. Die spätere Implementation individueller Sichten ist auf der Basis der lokalen Modellkomponente und des entwickelten PWDm grundsätzlich möglich. Die Implementation der grafischen Entwurfssymbole bietet zudem eine komfortable Systemunterstützung während des Software-Entwurfs, ohne die im Rahmen eines Praktikums zu erbringende Transferleistung zu übernehmen. Damit ist das PASSENGER CASE Tool vergleichbaren Ansätzen im Hinblick auf die domänenspezifischen Anforderungen deutlich überlegen.

## **5.6 Protokolle und Dienste**

Die Architektur der synchronen Groupware PASSENGER, vgl. auch die Abbildungen 19 und 22, sieht zwei verschiedene Kommunikationspfade vor: zum einen TCP-basierte Client/Server-Verbindungen und zum anderen UDP-basierte Interclient-Verbindungen. Entsprechend dem ALF-Konzept sollen die Daten vor der Übertragung auf der Anwendungsebene vorbereitet werden. Hierfür wurden im Rahmen dieser Arbeit zwei Protokolle auf der Anwendungsebene definiert. Im Folgenden werden diese Protokolle beschrieben.

### **5.6.1 Das PASSENGER Realtime Transport Protocol**

Das PASSENGER Realtime Transport Protocol (PRTP) ist auf der Anwendungsebene implementiert und nimmt die Audio- und Video-Daten der Kommunikationskomponente entgegen, bereitet sie für die Übertragung auf und liefert sie an die darunter liegende Netzwerkschicht. Dort erfolgt die Übertragung mit dem unzuverlässigen Dienst UDP. Beim Empfänger erfolgt die Verarbeitung in umgekehrter Reihenfolge.

UDP bietet keinerlei Garantien für die Auslieferung der Daten. Fehlerhaft empfangene Pakete werden bereits auf der Transportschicht verworfen, eine Neuübertragung findet nicht statt. Datenpakete, die auf der Empfängerseite die Anwendungsschicht erreichen, können jedoch als fehlerfrei angesehen werden<sup>32</sup>. Diese Eigenschaften des UDP sind in der Regel für Multimediadaten tolerierbar. Kritischer ist in dem Zusammenhang die fehlende Ordnungserhaltung in Kombination mit zeitlichen Verzögerungen zu sehen.

Hierzu bietet das ALF-Konzept die Möglichkeit, zusätzliche Dienste auf der Anwendungsebene zu implementieren. Dadurch können Datenpakete auf der Anwendungsebene identifiziert werden und zusätzlich

- Paketverluste erkannt und
- eine Ordnungserhaltung auf Anwendungsebene realisiert werden.

### 5.6.1.1 Konzeption

Das PRTP ist als Erweiterung des RTP implementiert und soll speziell den Transport von Echtzeitdaten unterstützen. RTP ist als erweiterbares Protokoll der Anwendungsebene definiert und folgt grundsätzlich dem ALF-Konzept. Es kennzeichnet jedes Datenpaket mit einem Zeitstempel und einer 16-Bit breiten Sequenznummer. Insgesamt können so in einem Zyklus 65.535 eindeutige Sequenznummern vergeben werden<sup>33</sup>. Bei der Fragmentierung großer Dateneinheiten kann zusätzlich das letzte Datenpaket einer zusammengehörenden Folge von Datenpaketen durch ein Markierungsbit gekennzeichnet werden. Insgesamt ermöglicht das RTP

- Die Identifizierung jedes Datenpaketes über eine Auswertung der Sequenznummer, wobei durch die Möglichkeit des Wertebereichüberlaufs - zur Sicherung der Eindeutigkeit - zusätzlich der Zeitstempel ausgewertet werden muss.

---

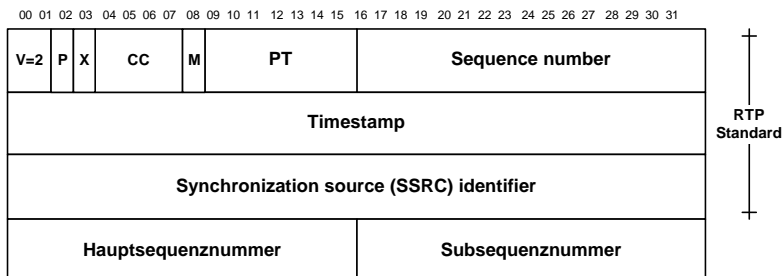
<sup>32</sup> Zusätzlich zur Datensicherungsschicht ist ab IPv6 eine UDP-Prüfsumme zwingend vorgeschrieben.

<sup>33</sup> In Sitzungen mit einer Dauer von ca. 90 Minuten und guten Übertragungsverhältnissen, z.B. 20 Frames pro Sekunde, muss mit einem Überlauf des Wertebereichs der Sequenznummer gerechnet werden. Der Algorithmus des Empfangsautomaten wird hierdurch unnötig komplex.

- Die Erkennung einer Folge zusammengehörender Datenpakete über die Auswertung des Zeitstempels, der Sequenznummer und des Markierungsbit, wobei nicht festgestellt werden kann, aus wie vielen Datenpaketen eine Dateneinheit besteht.

Damit können fragmentierte Dateneinheiten auf der Anwendungsebene des Empfängers rekonstruiert werden, sofern nicht das Paket, welches das Markierungsbit enthält, verloren geht. In diesem Fall müssen zusätzliche Mechanismen implementiert werden, um die richtige Zuordnung der Pakete zu sichern.

In dieser Arbeit wird ein Ansatz gewählt, der durch eine Erweiterung des RTP den Überlauf des Wertebereichs der Sequenznummer toleriert und die Zuordnung einzelner Datenpakete zu einer Dateneinheit vereinfacht. Hierzu wird zunächst der RTP-Protokollkopf um zwei Datenfelder erweitert und jedes Datenpaket zusätzlich mit einer Subsequenznummer und einer Hauptsequenznummer gekennzeichnet. Aus der Hauptsequenznummer lässt sich einerseits das letzte Datenpaket einer fragmentierten Dateneinheit ermitteln und andererseits zu jedem Zeitpunkt feststellen, aus wie vielen Datenpaketen diese Dateneinheit besteht. Die Subsequenznummern fragmentierter Dateneinheiten beginnen immer bei Null. Bei einer Zerlegung in  $n$  Datenpakete ist die höchste Subsequenznummer  $n-1$ . Dieser Wert gilt ebenfalls für die Hauptsequenznummer. Zusätzlich erhalten Datenpakete einer Dateneinheit den gleichen Zeitstempel. Die folgende Abbildung zeigt den Aufbau des PRTP-Protokollkopfes.



**Abb. 37:** PRTP-Protokollkopf

Die Audio- und Videodaten der synchronen Groupware PASSENGER werden als Nutzlast des PRTP versendet und das PRTP als Nutzlast innerhalb eines UDP-Datagramms.

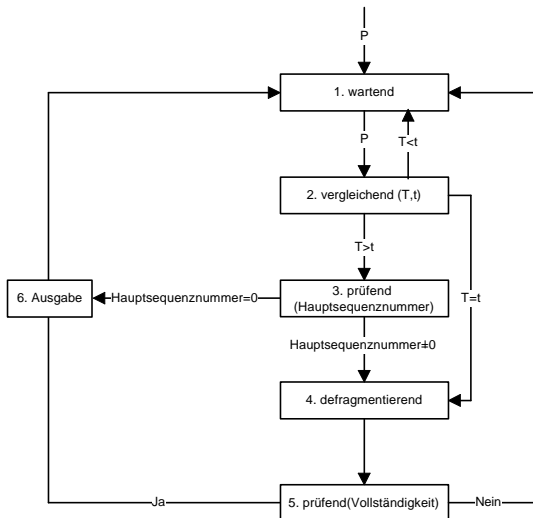
### 5.6.1.2 Vergleich PRTP versus RTP

Das RTP-Konzept sieht vor, dass die Daten unmittelbar nach ihrem Empfang verarbeitet werden. Im Falle eines fragmentierten Videobildes bedeutet dies, dass die einzelnen Fragmente in der Reihenfolge ihres Empfangs dargestellt werden. Durch Paketverluste kann es dazu kommen, dass Folgen unvollständiger Bilder dargestellt werden. Erste Versuche mit der synchronen Groupware PASSENGER im WAN haben gezeigt, dass dieses Verhalten sich sehr störend auswirkt, da den unvollständigen Bildern eine unangemessen hohe Beachtung geschenkt wird.

In dieser Arbeit wird daher der Ansatz gewählt, in der Kommunikationskomponente des PASSENGER Client nur vollständige Videobilder darzustellen. Unvollständig empfangene Videobilder werden verworfen und bis zum Empfang des nächsten vollständigen Videobildes wird das zuletzt vollständig empfangene Videobild dargestellt.

Der beschriebene Ansatz wird durch das PRTP unterstützt und führt zu einer deutlichen Vereinfachung gegenüber dem RTP. Der Empfangsautomat des PRTP ist in Abbildung 38 am Beispiel des Empfangs von Videodaten dargestellt.





P: Datenpaket  
 T: Zeitstempel des empfangenen Datenpaketes  
 t: gespeicherter Zeitstempel

**Abb. 38:** Empfangsautomat des PRTP

Im Zustand 1 (wartend) sind alle Prozesse zum Empfang von Datenpaketen aktiv. Beim Empfang eines Datenpaketes  $P$  geht das System über in den Zustand 2 (vergleichend( $T,t$ )). Gleichzeitig wird in einem parallelen Prozess P1 ein Timer zurückgesetzt und überprüft, ob innerhalb von maximal 3000 ms ein weiteres Datenpaket empfangen wird. Hierdurch werden Blockierungen verhindert, z.B. wenn der sendende Client die Übertragung unterbrochen hat.

Im Zustand 2 (vergleichend( $T,t$ )) werden Prozesse gestartet, die den Zeitstempel  $T$  des empfangenen Paketes  $P$  mit dem aktuell gespeicherten Zeitstempel  $t$  vergleichen.

- Gilt  $T < t$ , so handelt es sich entweder um ein zu spät eingetroffenes Datenpaket oder um ein Duplikat eines bereits früher empfangenen Datenpaketes. In diesem Fall wird das empfangene Datenpaket  $P$  verworfen, und der Automat kehrt zurück in den Zustand 1 (wartend).

- Gilt  $T > t$ , so handelt es sich um ein Datenpaket einer neuen Dateneinheit. In diesem Fall wurde die vorherige Dateneinheit nicht vollständig empfangen, und die zugehörigen Datenpakete werden verworfen und die Speicher neu initialisiert. In diesem Fall geht der Automat über in den Zustand 3 (prüfend(Hauptsequenznummer)).
- Gilt  $T = t$ , so handelt es sich um ein weiteres Datenpaket der aktuellen Dateneinheit. In diesem Fall geht der Automat über in den Zustand 4 (defragmentierend).

In den, dem Zustand 3 (prüfend(Hauptsequenznummer)) zugeordneten Prozessen, wird überprüft, ob es sich um eine Dateneinheit handelt, die aus einem einzigen Datenpaket besteht. Dies ist der Fall für *Hauptsequenznummer*=0 und führt dazu, dass die Dateneinheit im Zustand 6 (Ausgabe) an die nächst höhere Protokollschicht übergeben wird. Der Automat geht anschließend bedingungslos in den Zustand 1 (wartend) über. In allen anderen Fällen geht der Automat in den Zustand 4 (defragmentierend) über.

Im Zustand 4 (defragmentierend) wird die Ordnung der Datenpakete anhand der Subsequenznummer überprüft und bei Bedarf hergestellt. Außerdem erfolgt an dieser Stelle eine weitere Filterung von Duplikaten. Nach Abarbeitung der dem Zustand 4 (defragmentierend) zugeordneten Prozesse geht der Automat bedingungslos in den Zustand 6 (prüfend(Vollständigkeit)). In diesem Zustand erfolgt eine Auswertung der Subsequenznummer und der Hauptsequenznummer. Ist die Dateneinheit vollständig, so erfolgt der Übergang in Zustand 6 (Ausgabe). Anschließend kehrt der Automat bedingungslos zurück in den Startzustand 1 (wartend). Ist die Dateneinheit weiterhin unvollständig, kehrt der Automat ebenfalls zurück in den Startzustand 1 (wartend).

Zusammenfassend bietet das PRTP gegenüber dem RTP eine bessere Unterstützung für die Darstellung vollständiger Videobilder. Diese resultiert im Wesentlichen aus einer Vereinfachung des Empfangsautomaten, da das PRTP

- Überläufe der Sequenznummer toleriert,
- die eindeutige Identifizierung jedes Datenpaketes deutlich vereinfacht
- und zu jedem Zeitpunkt insgesamt mehr Informationen über fragmentierte Dateneinheiten bereit stellt als das RTP.

### 5.6.1.3 Optimierung der Audiodatenströme

Ein weiterer Vorteil des ALF-Konzeptes besteht darin, Audio- und Videodaten unterschiedlich behandeln zu können. Dadurch kann berücksichtigt werden, dass in einer Online-Kommunikation Qualitätsverluste bei der Videoübertragung eher toleriert werden als bei der Audioübertragung. Im Folgenden wird das hierzu entwickelte Verfahren zur Verarbeitung der Audio- und Videodaten auf der Sendeseite diskutiert. Das Verfahren basiert im Wesentlichen auf einer Bestimmung der Nutzlastlänge des PRTP vor dem Hintergrund einer Priorisierung der Audiodaten gegenüber der Videodaten.

Die Bestimmung der Nutzlastlänge des PRTP setzt Kenntnisse der Anwendungsebene über Eigenschaften und Parameter der Netzwerkebene voraus. In dieser Forderung wird ein wesentlicher Unterschied des ALF-Konzeptes zu üblichen Ebenenmodellen, wie dem ISO/OSI-Referenzmodell sichtbar. Während Ebenenmodelle eine strikte Trennung der verschiedenen Ebenen vorsehen, hebt das ALF-Konzept gerade diese Trennung auf, indem es explizite Kenntnisse über Netzwerkparameter auf der Anwendungsebene fordert.

Die entscheidende Größe in dem Zusammenhang ist die maximale Größe eines Netzwerkpaketes auf dem Weg von der Quelle zum Ziel, die PathMTU. Der genaue Wert der PathMTU wird auf der Vermittlungsschicht ermittelt (PathMTU Discovery) und an die Anwendungsschicht übergeben. In IP-basierten Netzwerken kann für den unteren Grenzwert der PathMTU eine Größe von 567 Byte angenommen werden. Übliche Werte für die Obergrenzen liegen bei 1500 Byte, vgl. [Dee98], wobei sich der Wert auch während der Übertragung ändern kann.

Grundsätzlich kann aus der PathMTU die maximale Größe der Nutzdaten pro Paket ermittelt werden. Diese Größe wird im Folgenden, in Anlehnung an die Definition der Application Data Unit (ADU) des ALF-Konzeptes, als PASSENGER Application Data Unit (PADU) bezeichnet. Entsprechend dem ALF-Konzept gilt, dass die in einer PADU enthaltenen Daten auch dann noch von der Anwendung verarbeitet werden können,

wenn die Reihenfolge der PADUs verändert wurde, einzelne PADUs fehlen oder fehlerhaft sind. Die maximale Größe einer PADU berechnet sich aus der PathMTU zu:

$$\begin{aligned} \text{PADU}_{\max} &= \text{PathMTU} - \text{IPv6Header} - \text{UDPHeader} - \text{RTPHeader} \\ &= \text{PathMTU} - 40 - 8 - 16 \\ &= \text{PathMTU} - 64 \end{aligned}$$

Für die statische Code-Effizienz Ceff eines Netzwerkpaketes gilt allgemein:

$$\text{Ceff} = \frac{\text{PADU}}{\text{PathMTU}}$$

Die maximale statische Code-Effizienz beträgt

$$\text{Ceff}_{\max} = \frac{\text{PADU}_{\max}}{\text{PathMTU}} = \frac{\text{PathMTU} - 64}{\text{PathMTU}}$$

Im Sinne der Aufgabenstellung ist die optimale Code-Effizienz definiert durch die Maximierung des Audiodatenstroms. Dabei kann zunächst davon ausgegangen werden, dass eine Audiodateneinheit nach der Kompression eine maximale Größe von ca. 1.500 Byte hat und somit in einem Datenpaket versendet werden kann. Videodateneinheiten dagegen haben in Abhängigkeit von der gewählten Komprimierung und Qualität eine maximale Größe von 5.000 Byte und müssen in der Regel fragmentiert werden. Für diese Fragmentierung können im Hinblick auf eine optimale Code-Effizienz bei der Aufteilung der Dateneinheit auf mehrere PADUs zwei Fälle untersucht werden:

Fall 1: Es werden so lange Pakete mit der maximalen Code-Effizienz generiert, wie auf Grund der Größe der Dateneinheit und der PathMTU, PADUs der maximal möglichen Größe erzeugt werden können. Das letzte Datenpaket enthält dann im Extremfall nur ein Byte und besitzt damit die minimale Code-Effizienz:

$$\text{Ceff}_{\min} = \frac{\text{PADU}_{\min}}{\text{PathMTU}} = \frac{1}{\text{PathMTU}}$$

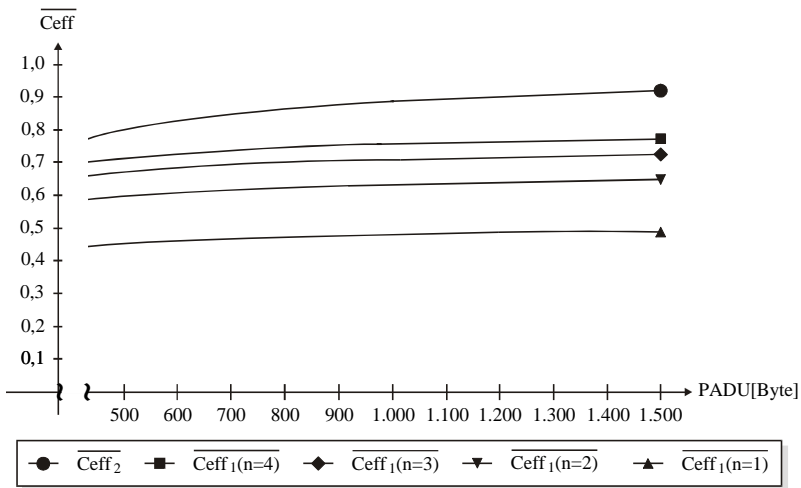
Bei einer Aufteilung auf  $m$  Netzwerkpakete ergeben sich damit  $n=m-1$  Netzwerkpakete mit der maximalen Code-Effizienz und damit eine mittlere Code-Effizienz von:

$$\begin{aligned} \overline{\text{Ceff}}_1 &= \frac{n \cdot \frac{\text{PADU}_{\max}}{\text{PathMTU}} + \frac{1}{\text{PathMTU}}}{n+1} \quad \text{mit PathMTU} = \text{PADU}_{\max} + 64 \\ &= \frac{n \cdot \text{PADU}_{\max} + 1}{(n+1) \cdot (\text{PADU}_{\max} + 64)} \end{aligned}$$

Fall 2: Eine zweite Möglichkeit besteht darin, die Dateneinheiten so aufzuteilen, dass alle daraus resultierenden Pakete die gleiche Code-Effizienz besitzen. Markus Jung hat in [Jun01] gezeigt, dass in diesem Fall die Größe der Nutzdaten niemals kleiner als  $\text{PADU}/2$  wird. Damit ergibt sich für diesen Fall eine mittlere Code-Effizienz von:

$$\overline{\text{Ceff}}_2 = \frac{\text{PADU}}{\text{PADU} + 128}$$

Einen Vergleich der beiden Verfahren, für den praktisch relevanten Bereich, zeigt Abbildung 39. Dabei wurde bereits berücksichtigt, dass übliche Audio-/Video-Dateneinheiten in der synchronen Groupware PASSENGER eine maximale Größe von 5.000 Byte besitzen. Dieser Wert ist abhängig von der gewählten Qualität der Audio- und Videodaten sowie den implementierten Codecs. Damit ergibt sich für den Wertebereich des Parameters  $n$ ,  $n \in \{1,2,3,4\}$ .



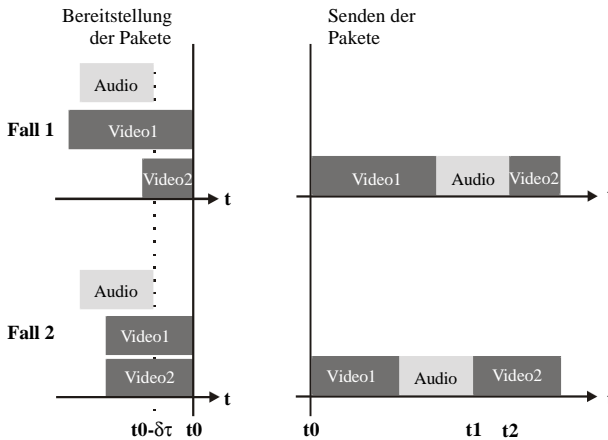
**Abb. 39:** Vergleich der Code-Effizienzen

Die Abbildung 39 zeigt, dass innerhalb des praktisch relevanten Bereichs die mittlere Code-Effizienz für Fall 2 durchgehend höher liegt. Insbesondere für kleinere bis mittlere PathMTU-Werte werden schon mittlere Code-Effizienzen oberhalb von 85% erreicht.

Bei der Implementation der synchronen Groupware PASSENGER wurde daher eine Aufteilung der Dateneinheiten entsprechend Fall 2 gewählt. Weiter wurde bei der Implementation berücksichtigt, dass der Verlust von Videopaketen beim Benutzer stärker toleriert wird als der Verlust von Audiopaketen. Daher werden die Audiopakete auf der Applikationsebene durch einen parallelen Prozess mit Echtzeit-Priorität verarbeitet. Die Transformation der Videodaten erfolgt dagegen im Hauptprozess. Insgesamt ergibt sich dadurch eine höhere Priorität von Audiopaketen gegenüber Videopaketen. Diese höhere Priorität bei der Verarbeitung der Audiodatenpakete wird zusätzlich durch den zweiten Fall unterstützt.

Im ersten Fall werden solange Videopakete maximaler Größe erzeugt, wie dies möglich ist. Ein Audiopaket mit höherer Priorität muss dabei im Mittel länger auf seine Übertragung warten als im zweiten Fall.

Die Abbildung 40 zeigt hierzu einen Vergleich, bei dem die Länge der Pakete gleichzeitig deren Übertragungsdauer symbolisiert. In beiden Fällen wird insgesamt der gleiche Datendurchsatz erreicht.



**Abb. 40:** Vergleich der Signalverzögerungen in Abhängigkeit von der Paketlänge

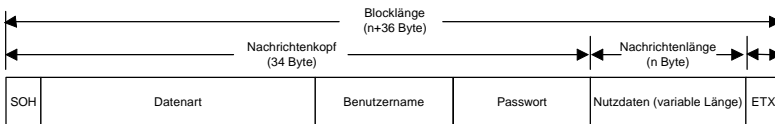
Im linken Teil der Abbildung 40 sind die Zeitpunkte dargestellt, zu denen die Datenpakete zur Übertragung bereitstehen. In beiden Fällen wird erst das Datenpaket Video1 übertragen und dann auf Grund der höheren Priorität das Datenpaket Audio. Im rechten Teil der Abbildung ist dargestellt, wann die Pakete vollständig gesendet wurden. Im Fall 2 ist dies auf Grund der kleineren Paketlänge für Video1 früher als im ersten Fall.

## 5.6.2 Das PASSENGER Session and Collaboration Protocol

Entsprechend der in Abbildung 22 dargestellten Systemarchitektur werden die Daten zur Client-Server-Kommunikation ebenfalls auf der Anwendungsebene aufbereitet und in einem eigenen Anwendungsprotokoll, dem PASSENGER Session and Collaboration Protocol (PSCP), an die Netzwerkschicht übergeben. Der Transport der PSCP-Pakete im Internet erfolgt als Nutzlast des zuverlässigen TCP.

### 5.6.2.1 Aspekte der Konzeption

Über das PSCP werden Daten verschiedener Anwendungsprozesse übertragen. Zum einen sind dies sitzungsspezifische Daten zur Teilnehmerverwaltung und Konferenzsteuerung und zum anderen werkzeugspezifische Daten, wie CASE Tool Daten oder Chat Tool Daten. Das PSCP ist ein textbasiertes Protokoll im ASCII-Format. Die folgende Abbildung zeigt den prinzipiellen Aufbau des Protokolls.



- SOH (2 Byte): Start of Header, leitet ein neues Nachrichtenpaket und einen festen Protokollkopf von 34 Byte ein
- Datenart (16 Byte): Kennzeichnend für den zugehörigen Anwendungsprozess
- Benutzername (8 Byte): Benutzername des Senders
- Passwort (8 Byte): Passwort des Senders
- Nutzdaten (variabel): Nutzdaten des durch Datenart beschriebenen Typs.
- ETX (2 Byte): Ende des Nachrichtenpaketes

**Abb. 41:** Nachrichtenrahmen des PSCP

Das Feld Benutzername und Passwort dient der Authentifizierung des Senders. Es wird benötigt bei Login- und Logout-Prozessen sowie dem Dokumenten-Upload.



### 5.6.2.2 Aspekte der Implementation

Alle an der Client-Server-Kommunikation beteiligten Daten werden bereits auf der Anwendungsebene vollständig formatiert. Dazu werden die Daten als Objekt der Klasse „TDaten“<sup>34</sup> an den Prozess zur Generierung der PSCP-Rahmen übergeben.

Die Größe eines PSCP-Datenpaketes wird durch den jeweiligen Anwendungsprozess und den aktuellen Anwendungszustand bestimmt. Die Größe der von den Anwendungsprozessen gelieferten Daten liegt in der Regel deutlich unter 8 KByte<sup>35</sup>, dem Wert, der standardmäßig auf der Betriebssystemebene für die Sende und Empfangspuffer eingestellt wird. Damit können grundsätzlich mehrere PSCP-Rahmen als Nutzlast eines TCP-Paketes versendet werden.

### 5.6.3 Das PASSENGER Multimedia Transport System

Eine weitere Zielsetzung im Rahmen dieser Arbeit war die Entwicklung einer IPv6-fähigen Anwendung. Naheliegende Zielplattformen waren damit einerseits das 6Bone, aber auch das MBone. Diese Overlay-Netze bieten einerseits eine Plattform zur Erprobung innovativer Applikationen, andererseits wurde die Entwicklung einiger Applikationen grundsätzlich erst durch diese Netze motiviert, z.B. die MBone-Tools. Allerdings sind die MBone-Tools zunächst primär für IP-Multicasting auf der Basis von IPv4 implementiert. Darüber hinaus sind zu den MBone-Tools vergleichbare 6Bone-Tools noch in einem sehr frühen Entwicklungsstadium [Fin02]. Im Rahmen dieser Arbeit werden die aus der Entwicklung und dem Einsatz der MBone-Tools resultierenden und veröffentlichten Erfahrungen zur Implementation vergleichbarer Anwendungen für den Einsatz im 6Bone genutzt. Die daraus resultierenden Ergebnisse können zwar als Beitrag auf dem Weg zur Entwicklung von 6Bone-Tools angesehen werden, allerdings war

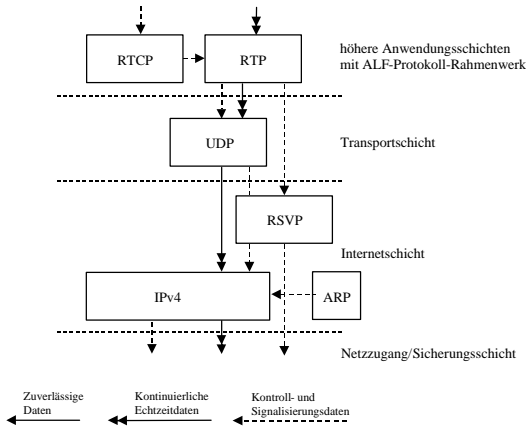
---

<sup>34</sup> TDaten ist von der abstrakten Klasse TStringList abgeleitet.

<sup>35</sup> Das CASE Tool liefert die größten Datenmengen, wobei der vollständige Inhalt in der Regel durch maximal 4 kByte beschrieben werden kann.

diese Art der Verallgemeinerung der Ergebnisse kein Ziel, das im Rahmen dieser Arbeit verfolgt wurde.

In diesem Abschnitt wird das in dieser Arbeit entwickelte PASSENGER Multimedia Transportsystem (PMMT) beschrieben. Das PMMT basiert im Wesentlichen auf dem von der GMD vorgestellten "Multimedia Transport System/ next generation" (MMTng) [San97], siehe Abbildung 42.

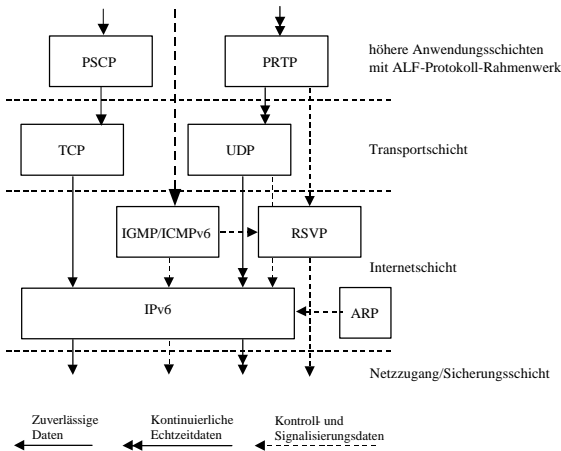


**Abb. 42:** MMTng-Architektur [Wal99]

Die MMTng-Architektur sieht vor, spezielle Funktionen auf der Anwendungsebene zu implementieren, um z.B. Audio- und Videodatenströme unterschiedlich behandeln zu können. Es benutzt hierzu das RTP. Der äquivalente Dienst wird im PMMT durch das oben beschriebene PRTP bereit gestellt. Beide Transportsysteme vermeiden entsprechend dem ALF-Konzept das Fragmentieren und Multiplexen der Datenpakete auf der IP-Ebene, um sie unabhängig voneinander verarbeiten zu können.

Zur Ressourcen-Reservierung integriert das MMTng das RSVP. Das Konzept des PMMT enthält ebenfalls das RSVP zur Ressourcen-Reservierung, obwohl die aktuelle Implementation der synchronen Groupware PASSENGER kein Quality of Service un-

terstützt. Ein bedeutender Unterschied der beiden Transportsysteme betrifft die Unterstützung für das Internet Protokoll. Während MMTng auf IPv4 aufsetzt und Multicasting als IP-over-ATM-Dienst implementiert, setzt das PMMT auf IPv6 auf und benutzt IP-Multicasting. Hierzu enthält PMMT das IGMP . Darüber hinaus benutzt die synchrone Groupware PASSENGER zusätzlich das TCP zur Client-Server-Kommunikation. Die resultierende Architektur des PMMT ist in Abbildung 43 dargestellt.



**Abb. 43:** Architektur des PMMT

Das implementierte IPv6-Multicasting benutzt zur dynamischen Multicast-Adressreservierung das Zeroconf Multicast Address Allocation Protocol (ZMAAP), in Kombination mit Unicast-Präfix-Multicast-Adressen. Die Verwendung der Unicast-Präfixe garantiert die Eindeutigkeit der Multicast-Adressen zwischen verschiedenen Links. Die Gültigkeitsprüfung der Adresse muss daher nur innerhalb eines Links erfolgen. Die Entwicklung und Implementierung des IPv6-Multicasting ist in [Jun01] beschrieben und war nicht Bestandteil dieser Arbeit. Die Funktion konnte im institutseigenen IPv6-Testnetzwerk für Link-Lokale und Site-Lokale Multicast-Adressen nachgewiesen werden. Eine Beschreibung des institutseigenen IPv6-Testnetzwerkes findet sich in [Jun01] und [Krz00].

## 5.7 Implementation

Die synchrone Groupware PASSENGER besteht aus Architektursicht im Wesentlichen aus zwei Anwendungen: zum einen dem PASSENGER-Server und zum anderen dem PASSENGER-Client. Der PASSENGER-Server bietet zentrale Dienste, wie z.B. Anmelden und Abmelden, und verwaltet das gemeinsame Repository. Im Rahmen dieser Arbeit wurde ein "minimales" Server-System implementiert, das die benötigte Basisfunktionalität bereit stellt um die Funktion der synchronen Groupware PASSENGER zu demonstrieren. Das bisher implementierte Server-System ist in [Jun01] und [Wal99] beschrieben und wird hier nicht weiter betrachtet.

Da bei Auswahl der Zielplattform für die synchrone Groupware PASSENGER, anders als bei vergleichbaren Anwendungen, wie z.B. [Gey99], [Sis98], [Bran99], nicht auf das MBone aufgesetzt wurde, bestand a priori auch keine Affinität gegenüber der UNIX-Welt.

Die synchrone Groupware PASSENGER ist in der Entwicklungsumgebung Delphi für Windows-basierte Systeme implementiert worden. Dabei konnten auf vielfältige Weise die Delphi-Standardbibliotheken und die Windows Application Programming Interfaces (API's) genutzt werden. Zur Verarbeitung der Multimedia Daten wurde z.B. eine zum Delphi-Standardumfang gehörende JPEG-Komponente genutzt. Für die Aufnahme und Wiedergabe von Audiodaten wurde die Windows-Multimedia-Extension-API (MME) in Kombination mit der Audio-Codec-Management-API (ACM) zur Kompression und Dekompression der Daten eingesetzt. Der verwendete GSM v6.10-Codec gehört zum Standardumfang von Windows.

Bei der Implementation des PMMT erfolgt der Zugriff auf die Protokollsoftware über das WinSock-2-API. Dieses API ist eine Microsoft Adaption der BSD-Socket-API und in der Programmiersprache C implementiert. Um die WinSock-Funktionen auch in der Programmierumgebung Delphi zur Verfügung zu stellen, muss die Visual Component Library (VCL) verwendet werden. Jede Verbindung zur WinSock-API wird in der Programmierumgebung Delphi als Objekt repräsentiert. Die zur Netzwerkkommunikation

bereit gestellten Delphi-Standardkomponenten enthalten keine Objekte zum Zugriff auf den IPv6-Protokoll-Stack. Die Übersetzung bzw. Erweiterung der entsprechenden Objekte war jedoch problemlos.

Insgesamt erwies sich die Programmierumgebung Delphi, bzw. die darin verwendete Programmiersprache Object Pascal, als ausreichend leistungsfähig, um komplexe Mehrbenutzeranwendungen zu implementieren. Über eine Kombination von Kylix und Delphi™ ist grundsätzlich auch die Möglichkeit einer Portierung auf das Betriebssystem Linux gegeben.

## 6. Erste Erfahrungen und Bewertungen

In diesem Kapitel werden erste Erfahrungen und Bewertungen beschrieben, die beim Einsatz der synchronen Groupware PASSENGER im Rahmen dieser Arbeit entstanden sind. Die Funktionsfähigkeit des Werkzeuges konnte im institutseigenen IPv6-Testnetzwerk sowohl in einer link-lokalen Umgebung als auch in einer site-lokalen Umgebung getestet und nachgewiesen werden. Die einwandfreie Funktion des Routings der Multicast-Pakete konnte durch Überwachungsfunktionen auf dem PASSENGER-Server belegt werden. Die Auswahl einer gültigen Multicast-Adresse durch das ZMAAP funktionierte ebenfalls problemlos. Das System hat sich auch im Dauerbetrieb stabil verhalten und verursachte keine „Systemabstürze“. Der Einsatz in einer globalen Umgebung konnte bisher nicht untersucht werden. Dies liegt zum einen am fehlenden Anschluss der GMU an das 6Bone [JOI01]. Einen alternativen Zugang stellen sogenannte Tunnelbroker bereit. In Deutschland wird ein entsprechender Dienst von der Universität Leipzig angeboten [Uni02]. Allerdings konnte auch diese Alternative im Rahmen dieser Arbeit nicht weiter verfolgt werden, da die Universität Leipzig den entsprechenden Tunnel trotz frühzeitiger Anmeldung nicht eingerichtet hat. Für die im Folgenden beschriebenen Untersuchungen stellte dies jedoch kein Problem dar, da hierfür bewusst eine link-lokale-Versuchsumgebung gewählt wurde. Diese bietet hervorragende technische Randbedingungen für die Untersuchung der grundsätzlichen Funktionen und vermeidet zusätzliche Einflüsse einer globalen Umgebung. Der Einfluss dieser Parameter wird Gegenstand zukünftiger Untersuchungen nach Abschluss dieser Arbeit sein.

Von besonderer Bedeutung für diese Arbeit war die Untersuchung des entwickelten Werkzeuges aus problemorientierter Sicht. Hierzu wurden im Rahmen dieser Arbeit im Wintersemester 2001/2002 Laborexperimente in den Büro- und Laborräumen des Instituts für Medientechnik und Software Engineering durchgeführt. Als Netzwerk wurde das institutseigene Fast-Ethernet (100Mbit/sec) genutzt und mit dem ebenfalls institutseigenen IPv6-Testnetzwerk verbunden. Die link-lokale Konfiguration und die hohe ver-

fügbare Bandbreite garantierten dabei eine stabile und hoch performante Netzwerkumgebung.

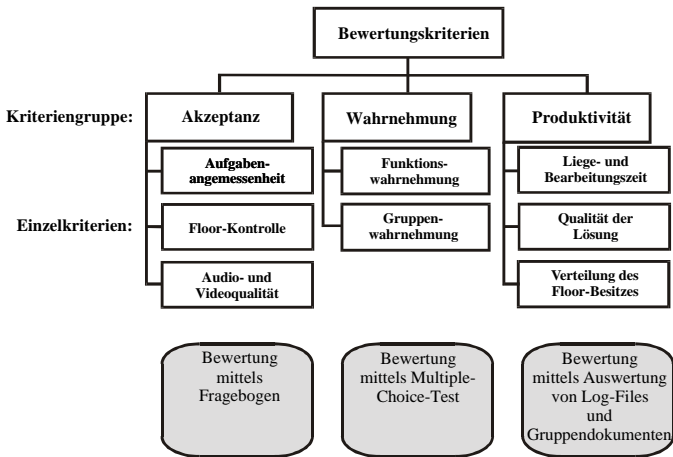
Insgesamt nahmen 21 Studierende aus 12 verschiedenen Nationen an den Experimenten teil, weshalb die verwendete Sprache ausschließlich Englisch war. Die Vorerfahrungen mit dem Einsatz von Videokonferenz-Technologie und Groupware sowie den Methoden des Software-Engineering waren stark heterogen. Aus diesem Grund wurden im Vorfeld des Experiments Basisinformationen zur synchronen Groupware PASSENGER sowie zur verwendeten Software-Engineering-Methode per E-Mail verschickt bzw. zum Download zur Verfügung gestellt. Vor dem eigentlichen Experiment erhielt jede Gruppe eine Präsentation, in der ein Überblick über die Bedienung der synchronen Groupware PASSENGER und den Software-Entwurf nach Ward & Mellor [War91] gegeben wurde. Die Dauer der Einführung betrug im Durchschnitt 25 Minuten und wurde von 43% der Teilnehmer als sehr gut und von 52% der Teilnehmer als gut bezeichnet. Nur 5% der Teilnehmer bewerteten die Angemessenheit der Einführung als mittelmäßig. Im Anschluss an die Präsentation erfolgte die Durchführung des Experiments. Dabei sollte jede der sieben Gruppen in einer Sitzung eine gegebene Aufgabenstellung bearbeiten. Bei der Aufgabe handelte es sich um eine einfach strukturierte Aufgabe von geringer Komplexität zur Entwicklung eines Kontextdiagramms für eine Heizungssteuerung. Die Dauer des Experiments betrug 25 Minuten. Abschließend bewerteten die Teilnehmer die synchrone Groupware PASSENGER, indem sie einen Fragebogen ausfüllten.

## **6.1 Untersuchungsdesign**

Im Rahmen des Untersuchungsdesigns wurden Bewertungskriterien und ein Fragebogen entwickelt. Durch die geringe Teilnehmerzahl, den begrenzten zeitlichen Umfang, die einmalige Durchführung des Experiments und die geringe Komplexität der Aufgabe scheiden komplexe Bewertungskriterien mit abhängigen Variablen wie z.B. der Zeit, dem Vorwissen der Teilnehmer oder der Aufgabenkomplexität a priori aus. Insbeson-

dere Lerneffekte beim Einsatz der synchronen Groupware PASSENGER konnten daher nicht untersucht werden.

Als wesentliche Schwächen existierender Systeme wurden in Abschnitt 4.3.5 die unzureichende Unterstützung bei der Diskussion und der gemeinsamen Ausarbeitung grafischer Entwurfsdokumente sowie der erhöhte Bedarf an sozialen Kontextinformationen identifiziert. Die daraus resultierenden Anforderungen an Groupware zum Einsatz in der Software-Engineering-Ausbildung sind in Abschnitt 4.4 beschrieben. Darauf aufbauend wurden zur Bewertung der synchronen Groupware PASSENGER die *Kriteriengruppen Akzeptanz, Wahrnehmung und Produktivität* abgeleitet. Innerhalb der einzelnen Kriteriengruppen erfolgt eine weitere Differenzierung in verschiedenen Einzelkriterien. Pro Einzelkriterium wurden ein oder mehrere Messwerte definiert. Im Folgenden werden die Kriteriengruppen näher beschrieben. Eine Übersicht der verwendeten Bewertungskriterien zeigt Abbildung 44.



**Abb. 44:** Bewertungskriterien für die Untersuchungen im Rahmen dieser Arbeit



## Kriteriengruppe Akzeptanz

Der Erfolg eines Werkzeuges ist im Wesentlichen geprägt von seiner Akzeptanz durch den Benutzer. Ein Schwerpunkt liegt daher auf der Untersuchung der Akzeptanz der Dialogschnittstelle und der Akzeptanz des Werkzeuges zur Erfüllung einer Sachaufgabe. Bei der Sachaufgabe handelt es sich vorrangig um Aktivitäten innerhalb des Software-Entwurfs. Dies schließt auch die Diskussion einzelner Entwurfsdokumente ein. Diese Aspekte werden im *Einzelkriterium Aufgabenangemessenheit* untersucht.

Der Dialogschnittstelle der synchronen Groupware PASSENGER kommt auch beim Aufbau von Beziehungs- und Partnerbildern sowie bei der Ausprägung eines Gruppengefühls eine übergeordnete Bedeutung zu. Da diese Aspekte jedoch Abhängigkeiten von der Zeit und der implementierten Floor-Kontrolle aufweisen, die beide aus den oben genannten Gründen im Rahmen dieser Untersuchung nicht berücksichtigt werden, scheiden sie dementsprechend als Kriterien für diese Untersuchung aus, so dass die Akzeptanz der Floor-Kontrolle als Einzelkriterium definiert wird. Dabei sollen die Studierenden die prinzipielle Eignung der implementierten Floor-Kontrolle zur Koordinationsunterstützung bewerten.

Eine weitere entscheidende Rolle bei der Akzeptanz spielt die Qualität der Audio- und Videokommunikation. Obwohl die Experimente in einer nahezu idealen Netzwerkkumgebung hinsichtlich der zur Verfügung stehenden Bandbreite durchgeführt wurden, soll dennoch eine Bewertung dieser Qualität in die Untersuchung einbezogen werden.

Die Einzelkriterien *Aufgabenangemessenheit*, *Floor-Kontrolle* sowie *Audio- und Videoqualität* wurden mit Hilfe eines Fragenkataloges untersucht. Alle Fragen und Antworten folgen dabei dem gleichen Muster. Eine typische Frage zum Einzelkriterium *Aufgabenangemessenheit* lautete:

*„How would you rate the discussion by means of the applied system in comparison to a natural Face-to-Face discussion?“*

Zu jeder Frage wurden den Studierenden fünf Antworten angeboten. Die vorgegebenen Antworten zur oben dargestellten Frage orientieren sich am Schulnotensystem und lau-

ten: „*very good (sehr gut)*“, „*good (gut)*“, „*medium (befriedigend)*“, „*poor (schlecht)*“ und „*very poor (sehr schlecht)*“.

Zur weiteren Auswertung wurden die Antworten der einzelnen Fragen mit Faktoren von eins (sehr gut) bis fünf (sehr schlecht) gewichtet und aus den Fragebögen die Klassenhäufigkeiten ermittelt. Anschließend wurde die Gesamtnote einer Frage als gewichteter Mittelwert über alle Antworten berechnet.

## Kriteriengruppe Wahrnehmung

Innerhalb der synchronen Groupware PASSENGER erfolgt die Gruppenwahrnehmung im Wesentlichen über die Kommunikationskomponente. Die aktuelle Rolle der einzelnen Teilnehmer wird dabei über das Namensfeld für alle Teilnehmer transparent gemacht. Die aktuelle Diskussionssituation bzw. der weitere Diskussionsverlauf werden über die Permission List bzw. die Schaltflächen innerhalb der einzelnen Kommunikationsfenster visualisiert. Inwieweit die Implementation dieser Maßnahmen tatsächlich einen Beitrag zur Gruppenwahrnehmung leistet, wurde in einem Multiple-Choice-Test ermittelt. Hierzu wurden den Teilnehmern innerhalb des Fragebogens Aufgaben vorgelegt, die jeweils auf einer bestimmten Konferenzsituation aufbauten, die durch einen Bildschirmausdruck beschrieben wurde.

Entscheidend für die Kriteriengruppe *Wahrnehmung* ist neben dem Einzelkriterium *Gruppenwahrnehmung* auch die *Funktionswahrnehmung*. In der Funktionswahrnehmung zeigt sich zum einen, inwieweit die in Abbildung 25 dargestellten Funktionsklassen insgesamt wahrgenommen werden. Andererseits ist die Verfügbarkeit einzelner Funktionen an die Floor-Kontrolle gebunden und nur über die Schaltflächen innerhalb der Kommunikationsfenster wahrnehmbar. Aufgrund dieser Abhängigkeit und der Vielzahl der implementierten Funktionen ist die Komplexität der Funktionswahrnehmung deutlich höher als die der Gruppenwahrnehmung einzuschätzen. Zusätzlich wird auf Grund dieser Komplexität auch eine zeitliche Abhängigkeit der Funktionswahrnehmung erwartet, die sich als Steigung einer entsprechenden Lernkurve ausprägt. Wegen der

oben genannten Einschränkungen werden diese Effekte jedoch zunächst nicht berücksichtigt, so dass sich die Bewertung des Einzelkriteriums *Funktionswahrnehmung* aus den Messwerten „Wahrnehmung der Zwischenruffunktion“, „Funktionen bei Floor-Besitz“ und „Telepointerfunktion“ berechnet. Die entsprechenden Informationen wurden wiederum mit Hilfe eines Multiple-Choice-Tests erhoben.

Die Auswertung der Multiple-Choice-Tests lieferte den Quotienten der erreichten Punkte und der erreichbaren Punkte als Prozentwert. Dieser Prozentwert wurde zum besseren Vergleich mit den Ergebnissen der Kriteriengruppe *Akzeptanz* in eine an das Schulnotensystem angelehnte Note umgerechnet. Dabei entspricht der Prozentwert 100 einer sehr guten Bewertung und der Prozentwert Null einer sehr schlechten Bewertung mit der Note fünf.

## Kriteriengruppe Produktivität

Als dritte Kriteriengruppe wurde die Produktivität in die Untersuchung einbezogen. Das Einzelkriterium *Liege- und Bearbeitungszeit* untersucht dabei die Anteile des Floor-Besitzes als Messwert für die Bearbeitungszeit und die Anteile des freien Floors als Messwert für die Liegezeit. Hohe Werte für den Quotienten aus Bearbeitungszeit und Liegezeit werden dabei als produktivitätssteigernd im Sinne der Gruppenarbeit interpretiert. Zusätzlich kann über das Einzelkriterium *Verteilung des Floor-Besitzes* die Qualität des implementierten Verfahrens zur Sicherung der Fairness abgeschätzt werden. Die entsprechenden Messwerte wurden in den laufenden Sitzungen auf dem Server protokolliert. Der Einfluss der Aufgabenkomplexität und der Zeit wurden in dieser Untersuchung auf Grund der geringen Teilnehmerzahl und dem begrenzten zeitlichen Umfang nicht berücksichtigt.

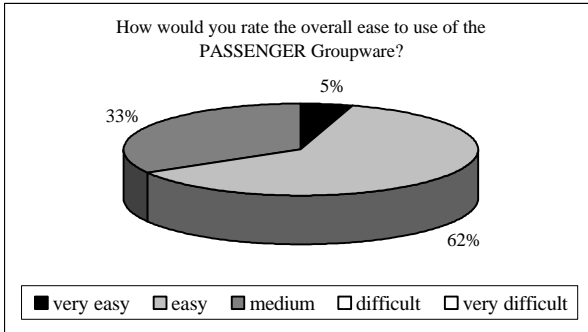
Als weiteres Einzelkriterium wird die *Qualität der Lösung* in die Untersuchung einbezogen. Diese wurde von einem wissenschaftlichen Mitarbeiter des Instituts aus den Gruppendokumenten bestimmt, wobei über die inhaltliche Qualität der Lösung in erster Näherung Rückschlüsse auf die Eignung des Systems als Medium zur Kommunikation

gezogen werden können. Die formale Darstellung innerhalb der Lösung kann als Indiz für die Eignung des Systems im Rahmen des Software-Entwurfs angenommen werden.

## 6.2 Untersuchungsergebnisse

Im Folgenden werden die Untersuchungsergebnisse präsentiert, die sich aus der Auswertung der Fragebögen, der Multiple-Choice-Tests, der Log-Files und der Gruppendokumente ergeben haben. Der verwendete Fragebogen ist papiergebunden und umfasst insgesamt 32 Fragen. Zusätzlich zu den Fragen der Kriteriengruppen *Akzeptanz* und *Wahrnehmung* wurden Fragen zu den bisherigen Erfahrungen der Teilnehmer mit Videokonferenz-Technologie, Groupware und Software-Engineering gestellt. Außerdem wurden über weitere Fragen Informationen zur Organisation und Struktur der Gruppen gesammelt.

Die Auswertung der Fragebögen ergab, dass ca. ? der Teilnehmer über keinerlei Erfahrung mit Desktop-Videokonferenz-Systemen verfügten. Ebenfalls ? der Teilnehmer bezeichneten die Bedienung der synchronen Groupware PASSENGER als einfach bis sehr einfach, vgl. Abbildung 45.



**Abb. 45:** Bewertung der Bedienfreundlichkeit

Diese gute Bewertung der Bedienfreundlichkeit der synchronen Groupware PASSENGER gewinnt vor dem Hintergrund der Unerfahrenheit der Teilnehmer an zusätzlicher Bedeutung. Außerdem haben 81% der Teilnehmer angegeben, in Zukunft Desktop-Videokonferenz-Systeme gerne häufiger einzusetzen.

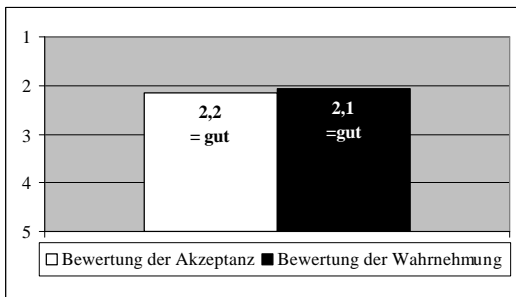
In nur zwei der insgesamt sieben Gruppen waren sich alle Teilnehmer vor dem Experiment bekannt. Dies hatte offensichtlich keinen Einfluss auf das Diskussionsverhalten der Gruppe; denn dieses wurde von 81% der Teilnehmer als gut bezeichnet. Die Bewertung des eigenen Diskussionsverhaltens ergab, dass 76% der Teilnehmer sich regelmäßig an der Diskussion beteiligt haben und 62% regelmäßig die Bearbeitung der Gruppendokumente übernommen haben. Diese Zahlen liefern erste Anhaltspunkte auf eine gute Diskussionsunterstützung durch die synchrone Groupware PASSENGER. Außerdem gaben 91% der Teilnehmer an, dass die Rolle des Bearbeiters während der Sitzung wechselte und liefern damit erste Hinweise zur Bestätigung des in Abschnitt 5.1.2 beschriebenen Rollenmodells.

Diese ersten Ergebnisse zeigen, dass wesentliche Ziele der Arbeit durch die entwickelten Konzepte bzw. das implementierte Werkzeug unterstützt werden. Hierbei sind vor allem die gute Bewertung der Bedienfreundlichkeit und die aus den bisherigen Untersuchungsergebnissen abgeleitete gute Diskussionsunterstützung zu nennen. Im Folgenden

werden die Untersuchungsergebnisse der einzelnen Kriteriengruppen dargestellt und interpretiert.

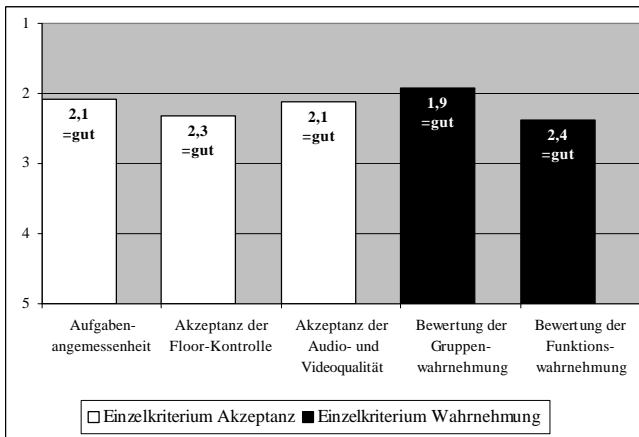
## Untersuchungsergebnisse der Kriteriengruppen Akzeptanz und Wahrnehmung

Die synchrone Groupware PASSENGER wurde in den Kriteriengruppen *Akzeptanz* und *Wahrnehmung* jeweils mit gut bewertet. Dabei fiel die Bewertung der Wahrnehmung gegenüber der Akzeptanz leicht besser aus, vgl. Abbildung 46.



**Abb. 46:** Bewertung der Kriteriengruppen Akzeptanz und Wahrnehmung

Die Interpretation des in Abbildung 46 dargestellten Untersuchungsergebnisses erfordert eine nähere Betrachtung der Einzelkriterien, siehe Abbildung 47.



**Abb. 47:** Bewertung der Einzelkriterien (Akzeptanz und Wahrnehmung)

Abbildung 47 zeigt, dass jedes der insgesamt fünf Einzelkriterien mit gut bewertet wurde. Eine differenziertere Betrachtung der einzelnen Bewertungen macht jedoch Unterschiede deutlich. Diese werden im Folgenden diskutiert.

#### **Diskussion der Bewertung des Einzelkriteriums *Aufgabenangemessenheit***

Die Aufgabenangemessenheit erreichte insgesamt die zweitbeste Bewertung. Im Wesentlichen resultiert dieses Ergebnis aus der guten Bewertung des Werkzeuges als Unterstützungsmedium zur Diskussion und bei der Lösung der gestellten Aufgabe. Dabei muss jedoch berücksichtigt werden, dass die Aufgabe einfach strukturiert und von geringer Komplexität war.

#### **Diskussion der Bewertung des Einzelkriteriums *Akzeptanz der Floor-Kontrolle***

Die insgesamt niedrigere Bewertung der Kriteriengruppe Akzeptanz ist auf den Akzeptanzwert der Floor-Kontrolle zurückzuführen. Dabei muss berücksichtigt werden, dass auf Grund der eher groben Granularität der Floor-Kontrolle diese - insbesondere bei erstmaliger Benutzung - durchaus als einschränkend empfunden werden kann. Es kann jedoch angenommen werden, dass sich die Einzelnote für die Akzeptanz der Floor-

Kontrolle mit der Anzahl der durchgeführten Sitzungen verbessert. Inwieweit diese Einzelnote dabei konvergiert oder sich sogar nach einer bestimmten Anzahl durchgeführter Sitzungen wieder verschlechtert, muss im Rahmen einer umfassenderen Evaluation geklärt werden. In diesem Zusammenhang stellt die ermittelte Einzelnote also einen Anfangswert dar, der insgesamt zufriedenstellend ausgefallen ist.

### **Diskussion der Bewertung des Einzelkriteriums *Audio- und Videoqualität***

Die Bewertung der Audio- und Videoqualität wurde innerhalb eines nahezu idealen Netzwerks ermittelt. Zukünftige Untersuchungen müssen klären, inwieweit diese gute Bewertung auch in realen Umgebungen erreicht werden kann. Insgesamt kann der ermittelte Wert der Einzelnote als Konvergenzwert aufgefasst werden. Ob eine weitere Verschiebung dieses Konvergenzwerts zu besseren Noten hin erreicht werden kann, scheint auf Grund der geringen Abmessungen der Videofenster fraglich. In dem Zusammenhang sollte untersucht werden, inwieweit sich die Verwendung von zwei Monitoren auf die Bewertung auswirkt. Allerdings zählen Entwurfsumgebungen mit zwei Monitoren im Moment noch nicht zur Standardausstattung einer Praktikums Umgebung und stehen auch der Mobilität bzw. der Verwendung von Laptops entgegen.

### **Diskussion der Bewertung des Einzelkriteriums *Gruppenwahrnehmung***

Das Einzelkriterium *Gruppenwahrnehmung* wurde von allen Einzelkriterien am besten bewertet. Die Einzelnote errechnet sich dabei aus den Messwerten zur „Wahrnehmung des Rollenkonzeptes“ und zur „Wahrnehmung des Diskussionsverlaufs“. Aus den guten Bewertungen kann dabei auf eine gute Balance zwischen der Komplexität des Rollenkonzeptes und seiner Darstellung über die Schaltflächen einerseits und der Komplexität bei der Visualisierung des Diskussionszustandes andererseits geschlossen werden.

### **Diskussion der Bewertung des Einzelkriteriums *Funktionswahrnehmung***

Innerhalb der Kriteriengruppe *Wahrnehmung* wurde das Einzelkriterium *Funktionswahrnehmung* erwartungsgemäß weniger gut bewertet als das Einzelkriterium *Gruppenwahrnehmung*. Dies ist in erster Linie auf die oben beschriebene höhere Komplexität innerhalb dieser Kriteriengruppe zurückzuführen, die sich aus dem insgesamt hohen



Angebot an Funktionen und deren teilweise Abhängigkeit von der Floor-Kontrolle erklären lässt. Deshalb wurde nach einer einmaligen Sitzung auch kein besseres Ergebnis erwartet.

Eine Untersuchung der protokollierten Log-Files, in der die Verwendung der einzelnen CASE-Tool-Funktionen erfasst sind, zeigte u.a., dass 90% aller Funktionsaufrufe auf nur ? der Gesamtmenge aller implementierten CASE-Tool-Funktionen entfällt. Dabei handelt es sich vorrangig um Funktionen zur Manipulation der grafischen Objekte. In die Multiple-Choice-Tests wurden aber deutlich mehr Funktionen einbezogen, von denen einige nachweislich nur ein einziges Mal verwendet wurden. Zur tatsächlichen Bewertung der Funktionswahrnehmung müssen daher die Ergebnisse einer langfristig angelegten Evaluation abgewartet werden. Als positives Ergebnis kann auf jeden Fall festgehalten werden, dass die Art der Implementation der Zwischenruffunktion mit einer Bewertung von 2,1 auch schon nach nur einer Sitzung gerechtfertigt wurde.

## Untersuchungsergebnisse der Kriteriengruppe Produktivität

Die Bewertung der Kriteriengruppe *Produktivität* erfolgte im Wesentlichen anhand einer Auswertung der Log-Files und der Gruppendokumente. In beiden Fällen wurde auf Grund fehlender objektiver Maßzahlen auf die Berechnung einer Bewertungszahl verzichtet. Dennoch lassen sich aus den Einzelkriterien wichtige Erkenntnisse ableiten.

Die Untersuchung der Liegezeit ergibt, dass der Floor im Durchschnitt zu 7% der Sitzungsdauer frei war. Dies entspricht einer Dauer von zwei Minuten und dreißig Sekunden bei einer durchschnittlichen Sitzungsdauer von 35 Minuten und 10 Sekunden. Dabei ist eine deutliche Zurückhaltung zu Beginn der Sitzungen zu berücksichtigen, die sich durch eine Gewöhnung an die Situation und die Umgebung sowie der individuellen, vertiefenden Auseinandersetzung mit der Aufgabenstellung erklärt. Eine einmal angelaufene Diskussion kam dagegen anschließend nicht mehr ins Stocken. Dies liefert zunächst zusätzliche Hinweise auf eine prinzipielle Eignung des Systems zur Diskussionsunterstützung. Erste Aussagen zur Produktivität erfordern eine zusätzliche Betrachtung der Gruppendokumente. Deren inhaltliche Qualität wurde im Schulnotensystem

bewertet und ergab im Mittel die Note gut. Damit kann auf Grund des hohen Anteils der Bearbeitungszeit und der guten Qualität der Lösungen auf eine hohe Produktivität beim Einsatz der synchronen Groupware PASSENGER geschlossen werden.

Eine Untersuchung der Dauer und der Häufigkeit des Floor-Besitzes ergab erwartungsgemäß, dass die Dauer des Floor-Besitzes innerhalb der Gruppen nicht gleich verteilt war. Eine Analyse der zeitlichen Verteilung des Floors hat ergeben, dass in fünf Gruppen jeweils ein Teilnehmer zu ca. 50% der Sitzungsdauer den Floor besessen hat. In einer zusätzlichen Häufigkeitsanalyse des Floor-Besitzes haben diese Teilnehmer jedoch in drei Fällen nicht den ersten Rang belegt. Diese Messwerte liefern deutliche Hinweise, dass das Ziel einer ausgewogenen Floor-Verteilung erreicht wurde und bestätigt die Effizienz der in Abschnitt 5.2.4 beschriebenen Maßnahmen zur Realisierung der Fairness bei der Floor-Vergabe und deren Implementation.

### **6.3 Zusammenfassung der Untersuchungsergebnisse**

Die beschriebene, problemorientierte Untersuchung lieferte wichtige Hinweise auf zwei Ebenen. Zum einen können aus den Ergebnissen durchweg positive Aussagen zur Qualität und Effizienz der entwickelten Ansätze und Konzepte sowie deren Implementation gewonnen werden. Andererseits lassen sich aus der Untersuchung Erkenntnisse für eine zukünftige, umfassende Evaluation der synchronen Groupware PASSENGER gewinnen.

Die insgesamt guten Bewertungen der Bedienfreundlichkeit und der Eignung zur Diskussionsunterstützung können in erster Näherung als Bestätigung für die Designentscheidungen im Rahmen des in Abschnitt 5.3 beschriebenen Benutzerschnittstellentwurfs angenommen werden. Die gute Akzeptanz der Aufgabenangemessenheit bestätigt einerseits die Effizienz des in Abschnitt 5.5 beschriebenen Whiteboard-Konzeptes beim Software-Entwurf, andererseits bestätigt sie zusätzlich die Eignung zur Diskussionsunterstützung. Das Konzept zur Visualisierung der Wahrnehmungsinformationen und dessen Umsetzung im Rahmen der Implementation der Kommunikationsfenster wurde

durch die Teilnehmer des Laborexperiments hervorragend bewertet. Das vergleichsweise schlechte Abschneiden der Funktionswahrnehmung in Zusammenhang mit der Floor-Kontrolle wurde oben ausführlich begründet. Dennoch kann auch die Implementation der Floor-Kontrolle als erfolgreich angesehen werden, da nur so die gute Bewertung bei der Eignung zur Diskussionsunterstützung zu erklären ist. Zusätzlich bestätigt die gute Bewertung der Zwischenruffunktion und die Einhaltung der Fairness bei der Floor-Vergabe die Effizienz der Floor-Kontrolle insgesamt. In der Summe wurde damit in erster Näherung der Nachweis erbracht, dass die synchrone Groupware PASSENGER die in Abschnitt 4.3.5 beschriebenen Schwächen anderer Werkzeuge beim Einsatz in einem Software-Engineering-Praktikum behebt.

Die beschriebene Untersuchung kann in keinem Fall eine umfassende Evaluation ersetzen. Diese sollte im Rahmen von Feldstudien und Feldexperimenten durchgeführt werden und insbesondere die bereits diskutierten Abhängigkeiten einzelner Bewertungskriterien berücksichtigen. Die Effizienz der in Abschnitt 5.6.1 beschriebenen Applikationsprotokolle und das in Abschnitt 5.6.3 dargestellte PASSENGER-Multimedia-Transport-System sind zusätzlich im Rahmen einer Untersuchung aus technikoienterter Sicht zu evaluieren.

## 7. Zusammenfassung und Ausblick

Die vorliegende Arbeit beschreibt den Entwurf und die Implementation einer synchronen Groupware für die Software-Engineering-Ausbildung. Wesentlichen Einfluss auf die Arbeit hatten der an der GMU zum Wintersemester 1997 eingeführte auslandsorientierte Studiengang „Informations- und Kommunikationstechnik“ und das veränderte Arbeitsumfeld der Software-Ingenieure. Hieraus entstand die Idee einer Portierung des industriellen Arbeitsumfeldes in die Software-Engineering-Ausbildung und daraus die Motivation zur Entwicklung von Beiträgen zur Zukunfts- und Praxisorientierung der Studierenden. Die Analyse des aktuellen Standes bei der Einführung von Groupware in die Software-Engineering-Ausbildung zeigte einerseits, dass bisher keine adäquaten Werkzeuge aus problemorientierter Sicht entwickelt wurden. Andererseits weisen die alternativ eingesetzten Systeme Defizite auf, die durch unterschiedliche Maßnahmen zwar abgemildert, nicht aber ganz aufgehoben werden können. Die im Rahmen dieser Arbeit entwickelte synchrone Groupware PASSENGER ist damit eines der ersten Systeme, das problemorientiert für den Einsatz in der Software-Engineering-Ausbildung entwickelt wurde; die vorliegende Arbeit ist damit auf Grund mehrerer Aspekte grundlegend:

- Sie stellt die notwendigen Grundlagen für die Entwicklung synchroner Groupware zum Einsatz in der Software-Engineering-Ausbildung bereit.
- Sie stellt den Stand der Technik bei der Einführung von Groupware in die Software-Engineering-Ausbildung dar, beschreibt eigene Voruntersuchungen und enthält die abgeleiteten Anforderungen an synchrone Groupware zum Einsatz in der Software-Engineering-Ausbildung.
- Sie beschreibt einen Rahmen für den Einsatz der synchroner Groupware PASSENGER in der Software-Engineering-Ausbildung durch die Angabe von Prozess- und Gruppenprozessmodellen.

- Sie erweitert das Spektrum der für Evaluationen zur Verfügung stehenden synchronen Groupware.

## 7.1 Ergebnisse der Arbeit

In dieser Arbeit wurden zunächst die Schwächen existierender Systeme beim Einsatz in der Software-Engineering-Ausbildung aufgezeigt und daraus Anforderungen an zukünftige Systeme abgeleitet. Für die daraus resultierenden offenen Probleme der mangelhaften Diskussionsunterstützung, der fehlenden sozialen Kontextinformationen und der unzureichenden Bearbeitungsmöglichkeiten für gemeinsame Gruppendokumente wurden vor dem Hintergrund der beschriebenen Voruntersuchungen aus problemorientierter Sicht Lösungsvorschläge entwickelt.

Zur Diskussionsunterstützung wurde im Rahmen dieser Arbeit eine Floor-Kontrolle entwickelt, die die wesentlichen Anforderungen der Verhinderung des gegenseitigen Ausschlusses und der Einhaltung der Fairness erfüllt. Dies wird dadurch erreicht, dass pro Zyklus jeder Teilnehmer den Floor einmal beantragen kann und dann auch garantiert erhält. Andererseits lässt die Floor-Kontrolle auch ohne die üblichen Nachteile starrer Zeitfensterkonzepte den Floor-Verzicht eines Teilnehmers pro Zyklus zu. Als weitere Option bietet die Floor-Kontrolle die Möglichkeit eines Zwischenrufs. Dieser Ansatz ist ein zusätzlicher Schritt zur Annäherung an natürliche Face-to-Face-Diskussionen und hat nachweislich einen hohen Beitrag zur guten Bewertung der synchronen Groupware PASSENGER geleistet, vgl. Abschnitt 6.2.

Als weiteres Ergebnis wurden im Rahmen dieser Arbeit Konzepte zur Steigerung der Gruppenwahrnehmung erarbeitet. Die daraus resultierende Implementation der Kommunikationsfenster und deren Anordnung innerhalb der Kommunikationskomponente haben im Rahmen der ersten Untersuchung das beste Resultat erzielt. Hierfür sind die einfache Wahrnehmung des Rollenkonzeptes und des aktuellen sowie des weiteren Diskussionsverlaufs verantwortlich. Zusätzlich unterstützt der Ansatz die Ausprägung von Partner- und Beziehungsaspekten und fördert den Aufbau eines Gruppengefühls.

Ein weiterer Schwerpunkt der Arbeit war die Unterstützung der gemeinsamen Bearbeitung der Entwurfsdokumente. Hierzu wurde ein Whiteboard entwickelt, das eine aufgabenspezifische Unterstützung für den Software-Entwurf bietet und kooperative Arbeitsprozesse durch die Trennung in private und öffentliche Arbeitsbereiche unterstützt. Der öffentliche Arbeitsbereich des Whiteboard wird gemeinsam mit einem Telepointer von der gemeinsamen Floor-Kontrolle verwaltet. Dadurch werden wesentliche Schwächen der bisher verwendeten Systeme mit optimistischen Floor-Kontroll-Verfahren behoben und auf einfache Weise für alle Teilnehmer die Zusammenhänge zwischen Ursache und Wirkung im öffentlichen Arbeitsbereich wahrnehmbar.

Schließlich entstand durch die konsequente Anwendung der IPv6-Technologie eine der ersten Groupware-Anwendungen für den Einsatz im zukünftigen IPv6-Internet, die zudem die Möglichkeiten des IPv6-Multicasting ausnutzt.

## **7.2 Weiterführende Arbeiten**

Die Entwicklung der synchronen Groupware PASSENGER orientierte sich an dem analysierten Gruppenverhalten in einem Software-Engineering-Praktikum der GMU und erfolgte aus problemorientierter Sicht. Für die in diesem Zusammenhang zu bearbeitenden Probleme wurden Lösungsansätze erarbeitet, implementiert und im Rahmen einer ersten Untersuchung bewertet.

Einen Ansatz für eine weiterführende Arbeit stellt die umfassende Evaluation des Werkzeuges im Rahmen von Feldstudien und Feldexperimenten dar. Dabei sollten insbesondere Lerneffekte beim Umgang mit der synchronen Groupware PASSENGER untersucht werden und Einflussfaktoren, wie z.B. die Aufgabenkomplexität, in die Untersuchungen einbezogen werden. Der auslandsorientierte Studiengang „Informations- und Kommunikationstechnik“ bietet zudem die interessante Möglichkeit, die Herkunft der Teilnehmer aus verschiedenen Lernkulturen als Variable in die Evaluation einzubeziehen.

Aus technikorientierter Sicht sind Quality-of-Service-Aspekte als ein Schwerpunkt für zukünftige Arbeiten zu nennen. Durch den konsequenten Einsatz der IPv6-Technologie bei der Entwicklung der synchronen Groupware PASSENGER wurde die Basis für die zukünftige Integration leistungsfähiger Protokolle und Dienste geschaffen. Hierdurch besteht auch die Möglichkeit, aktuelle Forschungsergebnisse aus dem Gebiet Active Networks in die weiteren Arbeiten einzubeziehen. Insbesondere die Variante „Application Level Active Networks (ALAN)“ [Fry99] bietet Perspektiven bei der Entwicklung maßgeschneiderter Routing-Protokolle und neuartiger Mechanismen zur Ressourcen-Reservierung.

Die synchrone Groupware PASSENGER unterstützt die Problemanalyse und die Modellierung innerhalb der Entwurfsphase; die Studierenden werden insbesondere von komplexen Aufgaben im Zusammenhang mit der Dokumentenverwaltung und dem Versionsmanagement entlastet. Zunehmend finden aber auch diese Tätigkeit an sich bzw. die dazu eingesetzten Werkzeuge Einzug in die Lehrpläne und die praktische Ausbildung. Eine Kombination der synchronen Groupware PASSENGER und dem speziellen Einsatzgebiet Software-Engineering mit den Aspekten Dokumentenverwaltung und Versionsmanagement wirft stärker prozessorientierte Probleme auf und erfordert problemorientierte Lösungen, z.B. bei der Verwaltung, der Organisation und der Optimierung des Dokumentenflusses. Diese Art von Problemen wurde im Rahmen dieser Arbeit nicht behandelt und wird auch insgesamt bisher vorrangig isoliert betrachtet, vgl. z.B. [Dav96] und [Dew93].

Weiterhin finden sich offene Probleme im Bereich der Kommunikation im Sinne einer multimedialen Systemsicht, die Mensch und Maschine umfasst. Die im Rahmen dieser Arbeit entwickelte Floor-Kontrolle und das Konzept zur Gruppenwahrnehmung erfüllen insgesamt die im Kontext dieser Arbeit gestellten Anforderungen. Ebenso stellen sie einen weiteren Schritt in der Annäherung der rechnergestützten Kommunikation an natürliche Kommunikationen dar. Dennoch besteht weiterhin Forschungsbedarf auf dem Gebiet der Nachbildung typischer Verhaltensweisen in Face-to-Face-Diskussionen, durch entsprechende Substitute in einer rechnergestützten Umgebung.

## Literaturverzeichnis

- [All01] Allison, A., Bateman, M., Ruddle, A.: Realising Real Time Multimedia Groupware on the Web. In *Proceedings of the 6th Eurographics Workshop in Multimedia*, Manchester, 8.-9. September 2001, verfügbar unter <http://virtual.inesc.pt/egmm2001/proceedings/allison.pdf>.
- [Alt98] Altmann, J., Weinreich, R.: An Environment for Cooperative Software Development: Realization and Implications. In *Proc. of the 31th Hawaii International Conference On System Sciences*, Volume 1: Collaboration Systems and Technology, 1998, verfügbar unter URL: <http://www.swe.uni-linz.ac.at/publications/pubs1998.shtml>.
- [And99] Andrews, D: Software-Engineering education in the 21st century. In *Information And Software Technology*, Vol. 41, Issue 6, 1999, Seite 933-936.
- [App01] Appelt, W., Busbach, U., Koch, T.: Kollaborationsorientierte asynchrone Werkzeuge. In: Schwabe, G., Streitz, N., Unland, R. (Hrsg.): *CSCW-Kompendium – Lehr- und Handbuch zum computerunterstützten kooperativen Arbeiten* – Springer Verlag, Berlin, 2001, Seite 194-203.
- [Bai89] Bair, J.H.: Supporting Cooperative Work with Computers: Addressing Meeting Mania. In *Proceedings of COMPCON '89*, San Francisco, CA, 27.2-3.3.1989, Seite 208-217, 1989
- [Bat98] Bates, J.: *The State of the Art in Distributed and Dependable Computing*. A CaberNet-Sponsored Report by the Laboratory for Communications Engineering University of Cambridge, UK, 1998, verfügbar unter URL: <http://www.newcastle.research.ec.org/cabernet/sota/report>.
- [Ben95] Bentley, R., Horstmann, T., Sikkil, K., Trevor, J.: Supporting Collaborative Information Sharing with the WWW: The BSCW Shared Workspace System. In *Proc. Of the 4<sup>th</sup> Int. World Wide Web Conference*, O'Reilly & Associates, Cambridge, Mass., 1995, Seite 63-73.
- [Bla98] Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., Weiss, W.: *An Architecture for Differentiated Services*. Internet RFC 2475, IETF, 1998.
- [Boe78] Boehm, B.W.: *Characteristics of software quality*. North-Holland Publ., New York, 1978.
- [Boe86] Boehm, B.W.: A spiral model of software development and enhancement. In *ACM Software Engineering Notes*, Vol. 11, No. 4, August 1986, Seite 22-42.
- [Bol98] Boloix, G., Robillard, P.N.: Case tool learnability in a Software-Engineering course. In *IEEE Transactions on Education*, Vol. 41, No.3, August 1998, Seite 185-193.
- [Bor98] Borghoff, U.M., Schlichter, J.H.: *Rechnergestützte Gruppenarbeit -Eine Einführung in verteilte Anwendungen*. Springer Verlag, Berlin, 1998.
- [Bra94] Braden, R., Clark, D., Shenker, S.: *Integrated Services in the Internet Architecture: an Overview*. Internet RFC 1633, IETF, 1994.



- [Brad97] Braden, R., Zhang, L., Berson, S., Herzog, S., Jamin, S.: *Resource ReSerVation Protocol (RSVP) - Version 1 Functional Specification*. Internet RFC 2205, IETF, 1997.
- [Bran97] Brand, O., Zitterbart, M.: Steuerung von Konferenz- und Kollaborationsanwendungen. In *Praxis der Informationsverarbeitung und Kommunikation (PIK)*, Heft 4, 1997, Seite 209–216.
- [Brau97] Braun, T., Stüttgen, H.J.: Die Internet-Architektur für integrierte Multimedia-kommunikation. In *it+ti- Informationstechnik und Informatik, Heft 4*, 1997, Seite 12-18.
- [Bran99] Brand, O., Mahalek, W., Sturzebecher, D., Zitterbart, M.: MACS - Eine modulare Kollaborationsumgebung. In *Praxis der Informationsverarbeitung und Kommunikation (PIK)*, Vol. 22, No. 4, 1999, Seite 213-220.
- [Brau99] Braun, T.: *IPnG: neue Internet-Dienste und virtuelle Netze*. dpunkt Verlag, Heidelberg, 1999.
- [Bre98] Brereton, P., Lees, S., Gumbley, M., Boldyreff, C.: Distributed group working in Software-Engineering education. In *Information And Software Technology*, Vol. 40, Issue 4, 1998, Seite 221-227.
- [Bre98a] Brereton, P., Gumbley, M., Lees, S.: Distributed Student Projects in Software-Engineering. In *Proc. of the 11th Conference on Software Engineering Education and Training (CSEE&T '98)*, IEEE Computer Society, 1998, verfügbar unter URL: <http://cssec.co.umist.ac.uk/publications.html>.
- [Bro90] Brothers, L., Sembugamoorthy, V., Muller, M.: ICICLE: Groupware for Code Inspection. In *Proceedings of CSCW 90*, ACM Press, New York, 1990, Seite 169-182.
- [Cla90] Clark, D.D., Tennenhouse, D.: Architectural Considerations for a New Generation of Protocols. In *Proc. ACM SIGCOMM 1990*, Philadelphia, Pennsylvania, September 1990, Seite 200-208.
- [Com98] Comer, Douglas: *Computer Netzwerke und Internets*. Prentice Hall, München 1998.
- [Con96] Conrads, D.: *Datenkommunikation - Verfahren, Netze, Dienste*. Vieweg Verlag, Braunschweig, 1996.
- [Con98] Conta, A., Deering, S.: *Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6)*. Internet RFC 2463, IETF, 1998.
- [Cus97] Cusumano, M.A., Selby, R.W.: How Microsoft builds Software In *Communications of the ACM*, Vol. 40, No.6, 1997, Seite 53-61.
- [Dav96] David, E.: Zur Relevanz der Dokumentation und Kommunikation in der betrieblichen Softwareentwicklung und Wartung. Dissertation. Fachbereich Wirtschaftswissenschaften, Universität – Gesamthochschule – Paderborn, 1996.
- [Dee89] Deering, S.: *Host Extensions for IP Multicasting*. Internet RFC 1112, IETF, 1989.
- [Dee98] Deering, S., Hinden, R.: *Internet Protocol, Version 6 (IPv6)*. Internet RFC 2460, IETF, 1998.

- [Del95] Delgrossi, L., Berger, L.: *Internet Stream Protocol Version 2 (ST2) Protocol Specification - Version ST2+*. Internet RFC 1819, IETF, 1995.
- [Dew93] Dewan, P., Riedl, J.: Toward Computer Supported Concurrent Software-Engineering. In *IEEE Computer* (26), January 1993, Seite 17-27.
- [Dew96] Dewan P.: Multiuser Architectures. In *Proceedings of EHCI'95*, IFIP Working Conference on Engineering for Human-Computer Communication, Chapman and Hall, London, 1996, Seite 247-270.
- [Dom97] Dommel, H.P., Garcia-Luna-Aceves, J.J.: Floor Control for Multimedia Conferencing and Collaboration. In *ACM Multimedia Systems*, Vol. 5, No. 1, January 1997, Seite.23-38.
- [Dru97] Drummond, S., Boldyreff, C.: *Adoption and Diffusion of Groupware in Software-Engineering Projects*. Presented at European Consortium for Informatics and Mathematics (ERCIM97), 1997, verfügbar unter: URL: <http://www.dur.ac.uk/sarah.drummond/papers/>.
- [Dru99] Drummond, S., Boldyreff, C.: SEGWorld: A WWW-based Infrastructure to Support the Development of Shared Software-Engineering Artifacts. In *Proc. IEEE 8th Intl. Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*, IEEE Press, 1999, Seite 120-125.
- [Dru00] Drummond, S., Boldyreff, C.: The Development and Trial of SEGWorld: A Virtual Environment for Software-Engineering Student Group Work. In *Proc. IEEE 13th Conference on Software-Engineering Education and Training (CSEE&T 2000)*, Austin, Texas, USA 6 - 8 March, verfügbar unter <http://www.dur.ac.uk/sarah.drummond/papers/>, 2000, Seite 87 -97.
- [Dru01] Drummond, S., Boldyreff, C., Ramage, M.: Evaluating Groupware Support for Software-Engineering Students. In *Computer Science Education*, Vol. 11, No. 1, March 2001, Seite 33-55.
- [Dur01] Durand, A., Fink, B., Hain, T.: *ngtrans Home Page*. The IPng Transition (ngtrans) working group of the IETF, 2001, verfügbar unter Url: <http://www.6bone.net/ngtrans/>.
- [Eli91] Ellis, C.A., Gibbs, S.J., Rein, G.L.: Groupware: Some Issues and experiences. In *Communications of the ACM*, Vol. 34, No I, 1991, Seite 339-407.
- [Eng01] Englberger, H.: Evaluierung. In: Schwabe, G., Streitz, N., Unland, R. (Hrsg.): *CSCW-Kompendium –Lehr- und Handbuch zum computerunterstützten kooperativen Arbeiten –*. Springer Verlag, Berlin, 2001, Seite 413-423.
- [Fen97] Fenner, W: *Internet Group Management Protocol, Version 2*. Internet RFC 2236, IETF, 1997.
- [Fin02] Fink, B.: 6Bone Tools Sites, 2002, verfügbar unter: URL: [http://www.6bone.de/6bone\\_tools.html](http://www.6bone.de/6bone_tools.html).
- [Flu96] Fluckiger, F.: *Multimedia im Netz*. Prentice Hall, München, 1996.
- [Fre89] Frese, M., Brodbeck, F. C.: *Computer in Büro und Verwaltung*. Springer Verlag, Berlin, 1989.

- [Fre92] Frey, S.: Funktionsprinzipien der Humankommunikation und der technischen Kommunikation. In *ITG Fachbericht Nutzung und Technik von Kommunikationsendgeräten*, ITG-Fachbericht Nr. 121, 1992, Seite 25-38.
- [Fre96] Frey, S., Kempter, G., Frenz, H.G.: Theoretische Grundlagen der multimedialen Kommunikation. In *Spektrum der Wissenschaft*, August 1996, Seite 32-38.
- [Fry99] Fry, M., Ghosh, A.: Application level active networking. In *Computer Networks*, Vol. 31, No. 7, 1999, Seite 655-667.
- [Gap97] Gappmaier, M., Häntschel, I.: Die Evaluierung von Workflow-Management-Systemen in Laborstudien. In: Grün, O., Heinrich, L. J. (Hrsg.): *Wirtschaftsinformatik - Ergebnisse empirischer Forschung*. Springer Verlag, Wien/New York, 1997, Seite 63-77.
- [Gey99] Geyer, W.: *Das digital lecture board: Konzeption, Design und Entwicklung eines Whiteboards für synchrones Teleteaching*. Dissertation, Praktische Informatik IV, Universität Mannheim, 1999.
- [Gib89] Gibbs, S.J.: CSCW and Software- Engineering. In: Tschritzis, D. (Hrsg.): *Object-Oriented Development*, University of Geneva, Switzerland, 1989, Chapter 4.
- [Gib94] Gibbs, W.W.: Software: chronisch mangelhaft. In *Spektrum der Wissenschaft*, Dezember 1994, Seite 56-63.
- [Gor96] Gorton, I., Motwani, S.: Issues in co-operative Software-Engineering using globally distributed teams. In *Information And Software Technology*, Vol. 38, Issue 106, 1996, Seite 647-655.
- [Gru88] Grudin, J.: Why CSCW Applications fail: Problems in the Design and Evaluation of Organizational Interfaces. In *Proc. of CSCW'88*, Portland, Orlando, 1988, Seite 85-93.
- [Gru94] Grudin, J.: CSCW: History and Focus. In *IEEE Computer*, Vol. 27, No. 5, 1994, Seite 19-26.
- [Gru95] Grundy, J.C., Mugridge, W.B., Hosking, J.G., Amor, R.W.: Support for Collaborative, Integrated Software Development. In *Proc. of the 7th Conference on Software-Engineering Environments*, IEEE Computer Society Press, 1995, Seite 84-94.
- [Gum97] Gumbley, M.: Procurement and set-up of low cost desktop video conferencing in a student environment, Internal Report, Keele University, 1997, verfügbar unter URL: <http://cssec.co.umist.ac.uk/publications.html>.
- [Haa94] Haake, G., Wüsteney, J.: Software-Entwicklung im 20-Stunden-Takt. In *Telekom Report 17*, Heft 2, 1994, Seite 85-87.
- [Hab01] Haberman, B., Thaler, D.: *Unicast-Prefix-based IPv6 Multicast Addresses*. Internet Draft, IETF, 2001.
- [Her99] Herrmann, Th., Misch, A.: CSCL-Evaluation aus kommunikationstheoretischer Sicht. In *Tagungsband des Workshops Evaluierung von Computer Supported Cooperative (Tele)-Learning (CSCL)-Systemen*, Universität Hohenheim, 29.-30.1.1999

- [Her01] Herrmann, Thomas: Kommunikation und Kooperation. In: Schwabe, G., Streitz, N., Unland, R. (Hrsg.): *CSCW-Kompendium –Lehr- und Handbuch zum computerunterstützten kooperativen Arbeiten –*. Springer Verlag, Berlin, 2001, Seite 15-25.
- [Hol01] Holmer, T., Haake, J., Streitz, N.: Kollaborationsorientierte synchrone Werkzeuge In: Schwabe, G., Streitz, N., Unland, R. (Hrsg.): *CSCW-Kompendium – Lehr- und Handbuch zum computerunterstützten kooperativen Arbeiten –*. Springer Verlag, Berlin, 2001, Seite 180-193.
- [Hun98] Hunger, A.: The Concepts of the Internationally Orientated Degree Course “Computer Science and Communications Engineering”. In *Proc. of the International Conference on Engineering Education 1998*, erschienen auf CD-ROM, Rio de Janeiro, 1998.
- [Jan01] Januschkiwitz, J.: *Entwurf und Implementierung eines CASE-Tools für kooperatives Software-Engineering*. Diplomarbeit, Fachbereich Elektrotechnik, Fachgebiet Datenverarbeitung, Gerhard-Mercator-Universität Duisburg, 2001.
- [Joh88] Johansen, R.: *Groupware: Computer Support for Business Teams*. The Free Press- Macmillan, New York, 1988.
- [JOI01] JOIN-Projekt-Team. Homepage des Join Open InterNetworks Referenz-zentrums an der Universität Münster, 2001, verfügbar unter: URL: <http://www.join.uni-muenster.de/>.
- [Jun01] Jung, M.: *Entwurf, Implementierung und Integration eines Multicast-Systems für eine Groupware zum verteilten Software-Engineering*. Diplomarbeit, Fakultät für Ingenieurwissenschaften, Institut für Medientechnik und Software Engineering, Gerhard-Mercator-Universität Duisburg, 2001.
- [Kai01] Kaiser, S.: Kommunikationsorientierte synchrone Werkzeuge. In: Schwabe, G., Streitz, N., Unland, R. (Hrsg.): *CSCW-Kompendium –Lehr- und Handbuch zum computerunterstützten kooperativen Arbeiten –*. Springer Verlag, Berlin, 2001, Seite 159-166.
- [Kne93] Kneuper, R.: Anforderungen an den Unterricht der Hochschulen im Fach Software-Engineering aus Sicht eines Software Hauses. In *Tagungsband des Workshops Software-Engineering im Unterricht der Hochschulen (SEUH'93)*, Teubner Verlag, Stuttgart, 1993, Seite 28- 37.
- [Kov93] Kovács, L.: Multimedia Groupware Systems. In *Proc. of the 8th Austrian-Hungarian Conference in Informatics The Challenge of Networking, CON'93*, 1993, verfügbar unter URL: <http://www.sztki.hu/dsd/kovacs.html>.
- [Krc91] Krcmar, H.: Computer Supported Cooperative Work -State of the Art. In *Proc. of the Fourth International Conference on Human-Computer Interaction*, Elsevier, New York, 1991, Seite 1113-1117.
- [Krc96] Krcmar, H., Schwabe, G.: CSCW-Werkzeuge. In *Wirtschaftsinformatik*, Vol. 38, Heft 2, 1996, Seite 209-224.
- [Krz00] Krzoska, K.: *Konzeptionierung und Aufbau eines IPv6 Versuchsnetzwerkes, Diplomarbeit*. Fachbereich Elektrotechnik, Fachgebiet Datenverarbeitung, Gerhard-Mercator-Universität Duisburg, 2000.

- [Lev93] Leveson, N.G., Turner, C.: An investigation of the Therac-25 accidents. In *IEEE Computer*, Vol. 26, No. 7, July 1993, Seite 18-41.
- [Lew01] Lewerentz, C., Rust, H.: Die Rolle der Reflexion in Software Praktika. In *Tagungsband des Workshops Software-Engineering im Unterricht der Hochschulen (SEUH'01)*, dpunkt Verlag, Heidelberg, 2001, Seite 73- 86.
- [Lub95] Lubich, H.P.: *Towards a CSCW-Framework for Scientific Cooperation in Europe*. Springer Verlag, Berlin, 1995.
- [Lyn90] Lynne, M.M., Conolly, T.: Why CSCW Applications Fail: Problems in the Adoption of Interdependent Work Tools. In *Proceedings of CSCW 90*, ACM Press, New York, 1990, Seite 371-380.
- [Mac97] Macaulay, L., Shaikh, A.N., Young R.: Groupware: A Criteria for Success. *Presented at the Multimedia Workshop*, Manchester Computing, November 1997, verfügbar unter <http://www.dur.ac.uk/sarah.drummond/papers/>.
- [Mah99] Mahalek, W., Zitterbart, M.: NetMVC als Basis zur Tele Kollaboration. In *Praxis der Informationsverarbeitung und Kommunikation (PIK)*, Heft 4, 1999.
- [Mar92] Marca, D., Bock, G.: *Groupware: Software for Computer-Supported Cooperative Work*. IEEE Computer Society Press, Los Alamitos, CA, 1992.
- [Mic02] Microsoft Corporation: *Microsoft NetMeeting*, 2002, verfügbar unter: URL: <http://www.microsoft.com/windows/netmeeting/>.
- [Mis99] Misch, A.: Anforderungen an lehrunterstützende Systeme aus kommunikationstheoretischer Sicht. In *Tagungsband zur 8. GI-Fachtagung: Informatik und Schule - Fachspezifische und fachübergreifende didaktische Konzepte.*, Springer-Verlag, 1999, Seite. 58-71.
- [Nar92] Narayanaswamy, K., Goldman, N.: Lazy Consistency: A Basis for Cooperative Software Development. In *Proc. of CSCW 1992*, ACM Press, Toronto, 1992, Seite 257-264.
- [Nat02] National Technology University. Homepage., 2002, verfügbar unter: URL: <http://www.ntu.edu>.
- [Nus97] Nuseibeh, B.: Ariane 5: Who dunnit. In *IEEE Software*, Vol. 14, No. 3, 1997, Seite 15-16.
- [Obe94] Oberweis, A., Stucky, W., Wendel, T.: Rechnergestützte Kommunikation in Software-Entwicklungsprojekten - Unterstützung einer kooperativen Systementwicklung. In *Congressband Online '94*, Hamburg, Feb. 1994, verfügbar unter URL: <http://wi2.wiwi.uni-frankfurt.de/publikationen/aobpub.htm>.
- [Pan01] Pankoke-Babatz, U.: Kommunikationsorientierte asynchrone Werkzeuge. In: Schwabe, G., Streitz, N., Unland, R. (Hrsg.): *CSCW-Kompendium –Lehr- und Handbuch zum computerunterstützten kooperativen Arbeiten –*. Springer Verlag, Berlin, 2001, Seite 167-173.
- [Pat95] Patterson J. F.: A Taxonomy of Architectures for Synchronous Groupware Applications. In *ACM SIGOIS Bulletin –Special Issue on Workshop Write-Ups and Position Papers from CSCW'94*, April 1995, Seite 27-29.

- [Peu97] Peuckert, J.O.: *Möglichkeiten und Grenzen telekooperationsfähiger Software beim Einsatz in einem räumlich verteilten Praktikum*. Schriftliche Hausarbeit im Rahmen der Ersten Staatsprüfung, Fachbereich Elektrotechnik, Fachgebiet Datenverarbeitung, Gerhard-Mercator-Universität Duisburg, 1997.
- [Pfa83] Pfaff, G.E. (Hrsg.): *Proceedings of the Workshop on User Interface Management Systems, Seeheim, FRG, November 1-3*, Springer-Verlag, Berlin, 1983.
- [Pfi01] Pfister, H.-R., Wessner, M.: Kooperatives Lehren und Lernen. In: Schwabe, G., Streitz, N., Unland, R. (Hrsg.): *CSCW-Kompodium – Lehr- und Handbuch zum computerunterstützten kooperativen Arbeiten* –. Springer Verlag, Berlin, 2001, Seite 252-263.
- [Phi99] Phillips W.G.: *Architectures for Synchronous Groupware*. Technical Report 1999-425, Department for Computing and Information Science, Queen's University, 1999, verfügbar unter URL: <http://phillips.rmc.ca/greg/pub/a4sg/1999-425.pdf>.
- [Pos80] Postel, J.: *User Datagram Protocol*. Internet RFC 768, IETF, 1980.
- [Pos81] Postel, J.: *Internet Control Message Protocol*. Internet RFC 792, IETF, 1981.
- [Pos81a] Postel, J.: *Transmission Control Protocol*. Internet RFC 793, IETF, 1981.
- [Pri01] Prinz, W.: Awareness. In: Schwabe, G., Streitz, N., Unland, R. (Hrsg.): *CSCW-Kompodium – Lehr- und Handbuch zum computerunterstützten kooperativen Arbeiten* –. Springer Verlag, Berlin, 2001, Seite 335-350.
- [Pro00] Projektgemeinschaft GfK, IESE, ISI: Analyse und Evaluation der Softwareentwicklung in Deutschland. Eine Studie für das Bundesministerium für Bildung und Forschung, 2000, verfügbar unter URL: [http://www.bmwi-info2000.de/aktuelles/literatur\\_aktuell.htm](http://www.bmwi-info2000.de/aktuelles/literatur_aktuell.htm).
- [Raa97] Raasch, J., Sack-Hauchwitz, A.: Kooperation, Kommunikation, Präsentation: Lernziele im Software-Engineering-Projekt. In *Tagungsband des Workshops Software-Engineering im Unterricht der Hochschulen (SEUH'97)*, Teubner Verlag, Stuttgart, 1997, Seite 34-44.
- [Rot00] Roth, J.: *Entwicklungs- und Laufzeitunterstützung für synchrone Groupware*. Dissertation. Fachbereich Informatik, Fernuniversität-Gesamthochschule Hagen, 2000.
- [Röt01] Röthlin, M.: Der Beitrag von Groupware-Applikationen zur methodengestützten Abwicklung von studentischen Projektseminaren. In *Tagungsband des Workshops Software-Engineering im Unterricht der Hochschulen (SEUH'01)*, dpunkt Verlag, Heidelberg, 2001, Seite 55-64.
- [Roy70] Royce W.W.: Managing the development of large software systems: concepts and techniques. In *Proc. IEEE WESTCON'70*, Aug. 25-28, Los Angeles, 1970, Seite A/1-1-A/1-9.
- [Rys99] Ryser, J., Glinz, M.: Konzipierung und Durchführung eines Software-Praktikums - Ein Erfahrungsbericht -. In *Tagungsband des Workshops Software-Engineering im Unterricht der Hochschulen (SEUH'99)*, Teubner Verlag, Stuttgart, 1999, Seite 55-68.

- [San97] Sanneck, H.: MMTng-Projektseite, 1997, verfügbar unter URL: <http://www.fokus.gmd.de/research/cc/glone/projects/mmtng->
- [Sap49] Sapir, E.: The unconscious patterning of behavior in society In: Mandelbaum D.G. (Hrsg.): *Selected writings of Edward Sapir in language, culture and personality*, University of California Press, Berkley, 1949, Seite 544-559.
- [Sar99] Sarjoughian, H.S., Zeigler, B.P., Ham, M. Parris, J.: Conducting Distributed Group Software-Engineering Projects: Challenges to State-of-the-Art Collaboration Technologies. In *Proc. of WMC 99*, SCS Publishing, 1999.
- [Schw01] Schwabe, G.: Gemeinsames Material und Gruppendächtnis. In: Schwabe, G., Streitz, N., Unland, R. (Hrsg.): *CSCW-Kompendium –Lehr- und Handbuch zum computerunterstützten kooperativen Arbeiten –*. Springer Verlag, Berlin, 2001, Seite 447-453.
- [Schü01] Schümmer, J., Schuckmann, C.: Synchroner Softwarearchitekturen. In: Schwabe, G., Streitz, N., Unland, R. (Hrsg.): *CSCW-Kompendium –Lehr- und Handbuch zum computerunterstützten kooperativen Arbeiten –*. Springer Verlag, Berlin, 2001, Seite 297-309.
- [Schm01] Schmedding, D.: Ein Prozessmodell für das Software-Praktikum. In *Tagungsband des Workshops Software-Engineering im Unterricht der Hochschulen (SEUH'01)*, dpunkt Verlag, Heidelberg, 2001, Seite 87-98.
- [Sch95] Schmedding, D.: Teamarbeit im Software-Praktikum -Ein Erfahrungsbericht-. In *Tagungsband des Workshops Software-Engineering im Unterricht der Hochschulen (SEUH'95)*, Teubner Verlag, Stuttgart, 1995, Seite 90-99.
- [Sch96] Schulzrinne, H., Casner, S., Frederick, R., Jacobson, V.: *RTP: A Transport Protocol for Real-Time Applications*. Internet RFC 1889, IETF, 1996.
- [Sch97] Schmedding, D.: *Konferenzbericht zum 5. Workshop SEUH Software-Engineering im Unterricht der Hochschulen Rostock (27. - 28. Februar 1997)*, 1997, verfügbar unter: URL: [http://pi.informatik.uni-siegen.de/stt/17\\_2/](http://pi.informatik.uni-siegen.de/stt/17_2/).
- [Schr98] Schroeder, U., Brunner, M., Deneke, M.: *Constructionist Learning in Software-Engineering Projects*. In Proc. International Software-Engineering Education Symposium SEES '98, 1998, verfügbar unter URL: [http://www.pi.informatik.tu-darmstadt.de/paper/TR\\_1998.5/SEeposen-abstract.html](http://www.pi.informatik.tu-darmstadt.de/paper/TR_1998.5/SEeposen-abstract.html).
- [Schw98] Schwarz, F.: *Entwurf und Implementierung einer Benutzeroberfläche für eine multimediale Lehr-/Lernumgebung zur räumlich verteilten Software Entwicklung*. Diplomarbeit, Fachbereich Elektrotechnik, Fachgebiet Datenverarbeitung, Gerhard-Mercator-Universität Duisburg, 1998.
- [Sch99] Schümmer, T.: *TUKAN - Entwicklung einer kooperativen Software-Konfigurations-Umgebung*. Diplomarbeit, Fachbereich Informatik, Fachgebiet Industrielle Prozess- und Systemkommunikation, TU-Darmstadt, 1999.
- [Sha49] Shannon, C., Weaver, W.: *The mathematical theory of communication*. University of Illinois Press, Urbana, 1949.

- [Sis98] Sisalem, D., Sanneck, H., Schubert, I., Fellbaum, K.-R., Ullmann, H.-J., Wolisz, A, Köpsel, A.: *USMInt: Abschlußbericht*, 1998, verfügbar unter URL: <http://www.fokus.gmd.de/research/cc/glone/projects/usmint>.
- [Ste99] Steinmetz, R.: *Multimedia Technologie – Grundlagen, Komponenten und Systeme* -. Springer Verlag, Berlin, 1999.
- [Tak93] Takeda, S., Nakatani, M., Nishida, S.: Group communication support system for software development project based on trouble communication model. In *Proc. of the 5th International Conference on Human Computer Interaction*, Elsevier, London, 1993, Seite 961-966.
- [Tan91] Tang, J.C.: Findings from Observational Studies of Collaborative Work. In *International Journal Man-Machine Studies*, Vol 34, No.2, 1991, Seite 143-160.
- [Teu95] Teufel, S., Sauter, C., Mühlherr, T., Bauknecht, K.: *Computerunterstützung für die Gruppenarbeit*. Addison Wesley, Bonn, 1995.
- [The95] Theißing, Florian: *Auf dem Weg in die Softwarekrise ?* Forschungsbericht des Fachbereichs Informatik, Technische Universität Berlin, Bericht 95-14, 1995.
- [Tie01] Tietze, D.A., Schümmer, T.: Kooperative Softwareentwicklung. In: Schwabe, G., Streitz, N., Unland, R. (Hrsg.): *CSCW-Kompodium –Lehr- und Handbuch zum computerunterstützten kooperativen Arbeiten –*. Springer Verlag, Berlin, 2001, Seite 264-275.
- [TUK99] *TUKAN - A Team Environment for Software Implementation*. Forschungsbericht der GMD, 1999, verfügbar unter URL: <http://www.darmstadt.gmd.de/concert/activities/internal/tukan.html>.
- [Uni02] Universität Leipzig: Website des Tunnelbrokers, 2002, verfügbar unter: URL: <http://joshua.informatik.uni-leipzig.de/>
- [Wai88] Waitzman, D., Partridge, C., Deering, S.: *Distance Vector Multicast Routing Protocol*. Internet RFC 1075, IETF, 1988.
- [Wal99] Walter, U.: *Konzeption und Implementierung eines Internet-Transportsystems für ein Telekooperations-Werkzeug*. Diplomarbeit, Fachbereich Elektrotechnik, Fachgebiet Datenverarbeitung, Gerhard-Mercator-Universität Duisburg, 1999.
- [War91] Ward, P.T., Mellor, S.J.: *Strukturierte Systemanalyse von Echtzeit-Systemen*. Hanser Verlag, München, 1991.
- [War98] Warnecke, G., Stammwitz, G., Hallfell, F.: Intranets als Plattform für Groupware-Anwendungen. In *Industrie Management*, Vol. 14, No. 1, 1998, Seite 24-28.
- [Wen95] Wendt, D. (Editor): Klassische Fehler in der Software-Entwicklung. In *Software-technik-Trends*, Band 15, Heft 4, November 1995, Seite 55-64.
- [Wer98] Werner, S., Hunger, A.: Possibilities and Limitations of a Computer Supported Cooperative Learning Environment within a Spatially Distributed Practical Training. In *Proc. of the Euromedia '98*, SCS Publishing, 1998, Seite 235-237.



- [Wer99] Werner, S., Hunger, A., Schwarz, F.: The PASSENGER Approach to conduct a software-Engineering Lab via Telecooperation over the Internet. In Proceedings of ICCE 99, Chiba, Japan, IOS Press, 1999, Seite 356-363.
- [Wer01] Werner, S., Hunger, A., Schwarz, F.: New Concepts for the Usage of Groupware in Software Engineering Education. In *Proceedings ED-Media 2001*, erschienen auf CD-ROM, Finnland, Tampere, 2001.
- [Whi96] Whitaker, R.D.: *Computer Supported Cooperative Work (CSCW) and Groupware*. Overview, Definitions, and Distinctions, 1996, verfügbar unter URL: <http://www.informatik.umu.se/~rwhit/CSCW.html>.
- [Wit99] Wittmann, R., Zitterbart, M.: *Multicasting – Protokolle und Anwendungen* – dpunkt Verlag, Heidelberg, 1999.