

Virtuelle Wissensräume

Ein Ansatz für die kooperative Wissensorganisation

Dissertation

Schriftliche Arbeit zur Verleihung
des akademischen Grades
DOKTOR DER NATURWISSENSCHAFTEN
im Fachbereich Mathematik/Informatik
der Universität Paderborn

von

Thorsten Hampel

Paderborn
Dezember 2001

Einleitung	1
1. Computergestütztes kooperatives Lernen	7
1.1 Bewertunggrundlage kooperativer Systeme	8
1.2 Begriffsbestimmung – Ansätze der Klassifikation	9
1.3 Lernerfolg und computergestütztes kooperatives Lernen	13
1.4 Lernförderliche Infrastrukturen.....	15
1.5 Medienbrüche – durchgängige Verfügbarkeit	16
1.6 Entwicklungslinie: CSCW/CSCL	18
1.6.1 CSCW	18
1.6.2 Systeme mit synchronen und asynchronen Elementen.....	19
1.6.3 Kooperative Lehr-/Lernumgebungen	20
1.6.4 WWW-gestützte Ansätze.....	22
1.7 Entwicklungslinie: Virtuelle Gemeinschaften	24
1.7.1 MUDs und MOOs	27
1.7.2 Klassifizierungsansatz des Nutzungsumfelds.....	31
1.7.3 Ansätze der Kooperationsunterstützung durch MUDs und MOOs	32
2. Kooperative Medienfunktionen – Szenarien	37
2.1 Medienfunktionen als Basis kooperativer Lernprozesse.....	38
2.2 Primäre individuelle Medienfunktionen	41
2.3 Primäre kooperative Medienfunktionen.....	43
2.3.1 Übertragen	45
2.3.2 Zugreifen	46
2.3.3 Synchronisieren	47
2.4 Selbstorganisation in den Medienfunktionen.....	48
2.5 Szenarien: Lernen im kooperativen Wissensraum	50
2.5.1 Seminar	52
2.5.2 Vorlesung.....	60
2.5.3 Übung	67
2.5.4 Kooperative Projektarbeit – Projektgruppe	77
2.5.5 Kooperatives Selbststudium – Prüfungsvorbereitung.....	84
2.6 Wissensräume – funktionale Anforderungen.....	85
2.6.1 Metaphern virtueller Wissensräume	86
2.6.2 Konzeptuelle Einbettung der primären Medienfunktionen.....	87
2.7 Analyse einer Reihe von Systemen.....	94
2.7.1 TeamWave.....	95
2.7.2 Habanero.....	96
2.7.3 Collaborative Virtual Workspace	97
2.7.4 Ergebnisse aus der Umsetzung	99

3.	Konzeptionelle Aspekte kooperativer Wissensräume.....	107
3.1	Der kooperative virtuelle Raum.....	108
3.1.1	Raummetapher.....	109
3.1.2	Persistenzeigenschaft.....	111
3.1.3	Grafische Repräsentationsformen.....	112
3.1.4	Awareness und Koordination	116
3.1.5	Zugriffskontrolle und Zugriffsschutz – Selbstadministration.....	123
3.2	Kernkonzept kooperativer Medienfunktionen	141
3.2.1	Zugrundegelegtes Objektmodell.....	142
3.2.2	Kreieren und Löschen.....	145
3.2.3	Arrangieren.....	147
3.2.4	Verknüpfen	148
3.2.5	Annotieren	150
3.2.6	Zugreifen, Übertragen und Synchronisieren.....	152
3.3	Kommunikation	156
3.3.1	Synchron – asynchron, kanalorientiert – objektbezogen	157
3.3.2	Problembereiche der Strukturierung von Kommunikation.....	158
3.3.3	Implikationen für die gewonnene Konzeption – synchron.....	159
3.3.4	Implikationen für die gewonnene Konzeption – asynchron	160
4.	Architekturkonzepte kooperativer Wissensräume	163
4.1	Technologische Grundkonzepte.....	164
4.1.1	Zentralisiert und repliziert	164
4.1.2	Collaboration Unaware/Aware und Collaboration Transparent	166
4.1.3	Anpassbarkeit – Customisation	167
4.2	Architekturansätze	170
4.2.1	Zentralisierte Architekturen – kooperatives MVC-Konzept.....	170
4.2.2	Replizierte Architekturen.....	172
4.2.3	Objektorientierte Frameworks kooperativer Applikationen	176
4.2.4	Webserver-basierte Architekturkonzepte	182
4.2.5	MUD- und MOO-basierte Architekturkonzepte.....	183
4.2.6	Experimentelle Ansätze, Prototypen der Forschung.....	188
5.	Technische Umsetzung.....	189
5.1	Überblick.....	190
5.2	Server	192
5.2.1	Klassenstruktur des Servers.....	193
5.2.2	Factories.....	198
5.2.3	Attribute.....	201
5.2.4	Annotationen	204
5.2.5	Ereignisbehandlung	204
5.2.6	Persistenzebene – Datenbankschnittstellen	210
5.2.7	Objekt-Strukturen – Object Request Broker.....	213
5.2.8	Schnittstellen des Servers – sTeam-API – COAL	215

5.3	Client.....	217
5.3.1	Client-Realisierungen	217
5.3.2	Verknüpfung: Browser – synchroner Java-Client – Proxy/Annotea	219
5.3.3	Integration verschiedener Clients	224
5.3.4	Architektur synchroner Java-Clients	229
5.4	Zusammenfassung: nutzerzentriert und selbstadministriert.....	234
5.5	Alternative Implementierungen	237
	Zusammenfassung und Ausblick.....	241

Einleitung

Als Gottfried Wilhelm Leibniz (1646 – 1716) seine „Dyadik“¹, das binäre Zahlensystem entwickelte und als „maxime naturae convenit“ (als von der Natur gegeben) beschrieb, war die Wirkungskraft dieser Erkenntnis und Darstellungsform für unser Zeitalter nicht absehbar. Digitale Medien spielen heute eine entscheidende Rolle für alle Bereiche unserer Gesellschaft. Ihre Tragweite reicht von soziologischen bis ökonomischen Faktoren, sie sind Lernmittel und Lerngegenstand zugleich. Die Auswirkungen und Möglichkeiten Neuer Medien im Prozess menschlichen Lernens sind seit einigen Jahren ein wichtiges Thema der Forschung.

Trotz der Einsicht, menschliches Lernen als einen sozialen Prozess zu begreifen, in dem Lernende Erkenntnisse über Medien gegenseitig austauschen und diskutieren, erfolgt eine vorrangige Konzentration auf Medienprodukte für die individuelle Nutzung. Diese „Selbstlernsysteme“ wie multimediale CD-ROMs oder durch Autorensysteme bereitgestellte, elektronische Dokumente führen Lernende entlang zumeist fest vorgegebener Pfade durch Materialien mit dem Anspruch, diese auf höchstem didaktischen Niveau aufzubereiten. Auch die Nutzung des World Wide Web (WWW) als Emissionsmedium digitaler Materialien hat diese Entwicklung nur wenig positiv beeinflusst: Lehrmaterialien werden auf Bildungsservern bereitgestellt, dem Lernenden bleibt lediglich die Möglichkeit sie herunterzuladen, also den Zeitpunkt des Lernens zu bestimmen und allenfalls durch das Verfolgen von Verweisen zu weiterführenden Materialien zu gelangen. Das WWW ermöglicht entsprechend den Zugang zu digitalen Materialien und entwickelt seine Stärken in der durchgängigen, aber nur lesenden Verfügbarkeit von Lehrmaterialien. Mit den Medien tätig umzugehen, sie in Kontexte einzufügen, durch eigene Anmerkungen und Notizen anzureichern oder nach eigenen Vorstellungen und Denkprozessen zu verknüpfen und zu strukturieren, ist hierbei nicht vorgesehen.

¹ Gothofredi Guillelmi Leibnitii: Opera Omnia. Hrsg. von L. Duntens, Band 3: Opera Mathematica, Genf, 1768: „Denn wenn man die einfachste Progression anwendet, nämlich die dyadische, anstelle der dekadischen oder quaternarischen, dann lassen sich alle Zahlen durch 0 und 1 ausdrücken [...]. Dieser Aufbau der Zahlen entspricht am meisten der Natur und enthält erstaunliche Anregungen zum Nachdenken auch für die Praxis, wenn auch nicht für den allgemeinen Gebrauch.“ Nach einer Übersetzung aus dem Lateinischen von Pater Franz X. Wernz in Hochstetter, E., Greve, H.J., Gumin, H.: *Herrn von Leibniz' Rechnung mit Null und Eins*, Siemens AG, 1979, S. 54ff.

Autorensysteme und Webserver können in dieser Form, wie Keil-Slawik treffend bemerkt, als „Einbahnstrasse des Lernens“² charakterisiert werden. Sie stellen Materialien zum ausschließlichen Lesen bereit und bieten keinerlei Mechanismen, die es gestatten, Medien aus verschiedenen Sichten und Quellen miteinander zu verknüpfen oder gemäß persönlicher Vorstellungen zu arrangieren. Aus der Bereitstellung von Medien zum ausschließlichen Lesen zeigt sich zugleich ein defizitäres Verständnis des Begriffes von Hypertext. Hypertext wird als das Verfolgen von Verweisen, also das *nicht-sequentielle Lesen* verstanden, demgegenüber entfaltet sich der wirkliche, mediale Mehrwert erst, wenn Hypertext als *nicht-sequentielles Schreiben*³ aufgefasst wird. Erst wenn Lernende aktiv auf Materialien einwirken können und ihnen gestattet ist, eigene Materialien zu erzeugen und diese mit vorhandenen zu verknüpfen, verliert das Medium seinen Charakter der Einbahnstraße des Lernens.

Neben technischen Möglichkeiten im Umgang mit digitalen Medien und der Qualität bereitgestellter Lehrmaterialien rückt insbesondere ihre Einbettung in Lernprozesse in den Vordergrund. Diese Einbettung sieht Lernprozesse verknüpft mit ihrer infrastrukturellen Umgebung und begreift menschliches Lernen vor allem als einen *sozialen Prozess*. Menschliches Lernen ist also entscheidend durch die Möglichkeiten des Austauschs und der Kommunikation mit anderen Lernenden beeinflusst. Hier kann Technik eine zentrale Rolle einnehmen.

Die Arbeitsgruppe Informatik und Gesellschaft, Paderborn, entwickelt seit einigen Jahren das Konzept der lernförderlichen Infrastrukturen⁴. Hierbei gilt neben der durchgängigen Verfügbarkeit die Alltagstauglichkeit von Werkzeugen und Methoden als maßgebliches Kriterium ihrer Eignung für den Einsatz in verschiedenen Lehr- und Lernkonstellationen. Alltagstauglichkeit untersucht neben dem technisch Machbaren insbesondere die Dimensionen der täglichen Praxis der Einbindung in Lehr- und Lernprozesse sowie ihre Einflussfaktoren auf diese. Alltagstauglichkeit ist damit ein wichtiges Kriterium, um verwendungsfähige Lösungen von experimentellen Technologiestudien unterscheiden zu können, die in der lebenswirklichen Praxis zumeist versagen.

Durchgängige Verfügbarkeit ist ein wichtiges Merkmal von Alltagstauglichkeit. Dahinter stehen zwei primäre Ziele: Zum einen gilt es digitale Medien an allen Orten des Lernens verfügbar zu machen – Menschen lernen an unterschiedlichsten Orten und in verschiedenen Konstellationen. Bezogen auf das universitäre Umfeld ergeben sich resultierend auch in der Situation der Präsenzlehre räumliche und zeitliche Distanzen zwischen Studierenden und Lernenden, die Kommunikations- und Koordinationsprozesse erschweren. Neue Medien können ihre Wirkung dementsprechend nur entfalten, wenn ihre Verfügbarkeit an allen Orten des Lernens sichergestellt ist.

Zum anderen weist die Nutzung digitaler Medien zumeist Brüche in ihrer durchgängigen Verfügbarkeit in Bezug auf das verwendete Medium auf. In einer WWW-gestützten Lehre beispielsweise können Lernende durch eine geeignete, technische Infrastruktur Materialien und Netzseiten begleitend zur Vorlesung betrachten, müssen jedoch zunächst einen Wechsel des Mediums in Kauf nehmen (z.B. diese ausdrucken), um sie mit persönlichen Notizen und Anmerkungen versehen zu können. (Das WWW ist in dieser Form kein aktives Medium, das Nutzern schreibenden Zugriff ermöglicht.) Das

² vgl. [Keil-Slawik 2001, S. 113].

³ vgl. [ebd., S. 113].

⁴ vgl. [Keil-Slawik 1999].

Konzept der Medienbrüche beschreibt dies als durch technische Unzulänglichkeiten erzwungene Wechsel zwischen Medien.⁵

Lernförderliche Infrastrukturen verfolgen das Ziel, die technische Unterstützung für Lernprozesse zu entwickeln, jedoch die didaktische Einbettung und spezifische Nutzung nicht festzulegen.

Im Gegensatz zu Autorensystemen oder Systemen des CBT (Computer Based Training) sind Lerninfrastrukturen auf die Unterstützung der Präsenzlehre und damit auf die Förderung der *Kooperation* zwischen Lernenden und Lehrenden angelegt. Sie bieten Lernenden verschiedene Mechanismen der netzgestützten Ablage (Speicherung) von Materialien oder Fähigkeiten, Kommunikationsmechanismen bzw. Kommunikationskanäle begleitend einem Lernprozess nutzen zu können. In ähnlicher Weise erlauben es Groupware-Umgebungen zwar, verschiedene Formen der Kooperation zwischen Nutzern bereitzustellen, allerdings zeigen sich auch hier Defizite:

Bürowirtschaftlich motivierte Systeme des CSCW (Computer Supported Cooperative Work)⁶ fokussieren einzelne Unterstützungsfunktionen in der Zusammenarbeit einer Gruppe von Nutzern, wie beispielsweise das kooperative Editieren eines digital vorliegenden Dokumentes. Sie sind zumeist aus der Idee der Unterstützung des Büroalltags entstanden und modellieren Kooperationsstrukturen zudem entlang vorgegebener Verarbeitungsprozesse und Rollen im Umgang mit Dokumenten.

Kooperationsstrukturen können sich damit aufgrund technischer und konzeptueller Unzulänglichkeiten nicht frei entfalten, sie sind an feste Regeln und Vorgehensweisen gebunden. Auch ist die Struktur einer kooperationsunterstützenden Umgebung zumeist vorgegeben und nicht von Nutzern selbsttätig an das Einsatzumfeld anpassbar – Eigenschaften, die sich in dem Attribut einer Serverzentriertheit zusammenfassen lassen.

Weiterhin defizitär erweist sich die mangelnde Berücksichtigung spezifischer Techniken des CSCL (Computer Supported Cooperative Learning). So haben Mechanismen der Kooperationsunterstützung wie gekoppelte Sichten auf gemeinsame Objekte, kooperative Zeichenflächen (Shared Whiteboards) oder Systeme, die Annotationen und Umgestaltung von Lehrmaterialien in der Gruppe unterstützen, keine nennenswerte Verbreitung in computergestützten Lernprozessen finden können. Dies liegt zum einen an der technisch anspruchsvollen und aufwändigen Umsetzung derartiger Systeme, zum anderen an ihrer architektonischen Abgeschlossenheit und Eigenständigkeit, die in die Arbeitsumgebung von Lernenden und Lehrenden nur schwer einzubetten ist.

Die vorliegende Arbeit entwickelt ein Konzept zur Aufarbeitung und Beseitigung dieser Defizite und zeigt eine technische Umsetzung: Zu diesem Zweck werden zunächst kooperationsunterstützende Ansätze in Kapitel 1 dargestellt. Berücksichtigt wird sowohl das bürowirtschaftlich motivierte Feld des CSCW, aber auch der aus einem privaten Umfeld entstandene Bereich der MUDs und MOOs. Letztere sind eng verbunden mit dem Phänomen virtueller Gemeinschaften und schaffen *nutzerzentrierte*, virtuelle Kommunikations- und Interaktionswelten.

Als problematisch erweist sich, dass es an einem theoretischen Rahmen fehlt, der eine systematische Abgrenzung und Bewertung vorhandener Ansätze bzgl. ihres medialen Mehrwertes zulässt. Das zu untersuchende Gebiet erscheint zu weitläufig und hete-

⁵ „Mit diesem Konzept bezeichnen wir Situationen, in denen es erforderlich ist, Wissensbestände in Bezug auf ihre medialen Trägerstrukturen zu transformieren, ohne dass damit ein Informationsgewinn auf kognitiver oder sensumotorischer Ebene verbunden ist.“ (vgl. [Keil-Slawik 1998a], S. 88).

⁶ Im Folgenden sei unter CSCW zunächst das Forschungsfeld berücksichtigt, der Begriff der Groupware konzentriert sich auf die entsprechenden Systemlösungen.

rogen, um ohne ein derartiges Modell eine Bewertung vornehmen zu können. Auch klaffen oft die Erwartungen an einen Ansatz und dessen Alltagstauglichkeit weit auseinander. Der Bereich der Kooperationsunterstützung beinhaltet eine Vielzahl interessanter Ideen und Visionen, aber nur wenig für die tägliche Praxis geeignete Lösungen. Die vorliegende Arbeit wählt und erweitert das Konzept der Medienfunktionen, um einen derartigen theoretischen Rahmen zu schaffen (Kapitel 2).

Medienfunktionen eignen sich, um technische Aspekte der Gestaltung lernförderlicher Infrastrukturen bestimmen zu können, also technische von nicht-technischen wie pädagogischen oder didaktischen Problemen differenzieren zu können. Medienfunktionen definieren grundsätzliche Funktionen tätigen Handelns im Umgang mit Artefakten (Materialien)⁷. Sie dienen dazu einen Wahrnehmungsraum zu schaffen, in welchem Vorstellungen und Wirklichkeit durch aktive Handlungen miteinander in Beziehung gesetzt werden können. Es zeigt sich schnell, dass das bestehende Modell der Medienfunktionen im kooperativen Umfeld einer Erweiterung und neuerlichen Systematisierung bedarf. Kooperative Aspekte eines Lernprozesses wie das Schaffen gemeinsamer Sichten auf Lehrmaterialien, ein durch Benutzerrechte kontrolliertes Zugreifen auf gemeinsam genutzte Dokumente und speziell das Übertragen von Materialien zwischen Lernenden während eines Lernprozesses bedürfen einer methodischen Untersuchung aus medientechnischer Perspektive.

Im Rahmen der theoretischen Fundierung der Arbeit wird zunächst der Begriff der primären Medienfunktionen in primäre individuelle und primäre kooperative Medienfunktionen differenziert. In einem zweiten Schritt werden individuelle und kooperative Funktionen von Medien untersucht. Primäre individuelle Medienfunktionen erfassen das Erzeugen, Löschen, Verknüpfen und Arrangieren von Materialien. Primäre kooperative Medienfunktionen betrachten Funktionen des Übertragens von Medien zwischen Lernenden sowie das Zugreifen und Synchronisieren von Ereignissen und Sichten. Letzteres beinhaltet zudem Funktionen der gegenseitigen Wahrnehmung.

Medienfunktionen beschreiben zunächst abstrakte Grundfunktionen, ihre konkrete Realisierung erfordert entsprechende Einsatzszenarios. Als weiterer Schwerpunkt von Kapitel 2 werden basierend auf den Erfahrungen zur computergestützten kooperativen Lehre, in deren Umfeld die vorliegende Arbeit entstanden ist, eine Reihe von derartigen Lernszenarien entwickelt. Diese orientieren sich an der universitären Präsenzlehre und formulieren kooperative Formen gemeinsamen Lernens am Beispiel von Übung, Vorlesung, Seminar und studentischer Projektgruppenarbeit.

Zentrale Metapher und infrastruktureller Bezugspunkt ist dabei der *kooperative Wissensraum*. Lernszenarien dienen in dieser Weise sowohl der Präzisierung als auch der Systematisierung von Anforderungen an technische Infrastrukturen für kooperative Wissensräume, welche die entscheidende Qualität des vorliegenden Ansatzes motivieren.

Die Aussage lautet, dass der kooperative Wissensraum ein entscheidendes Konzept ist, um bestehende Medienbrüche zwischen den primären Medienfunktionen zu verringern und die Integration von Kommunikations-, Wahrnehmungs- und Handlungsfunktionen dadurch entscheidend zu verbessern. In den dargestellten Szenarien schlägt sich dies insbesondere in Mechanismen der Selbstorganisation und Selbstadministration für die Lernenden nieder. Kooperative Wissensräume erheben den Anspruch einer nutzerzentrierten und selbstorganisierten, lernförderlichen Infrastruktur.

⁷ Der Begriff des Artefakts soll im Folgenden ohne weitere Systematisierung für Medien und Materialien als Teil eines Lernprozesses stehen.

Mit dem Ziel, existierende Systeme und Ansätze bzgl. ihrer Eignung und Umsetzbarkeit hinsichtlich der dargestellten Szenarien bewerten zu können, werden abschließend in Kapitel 2 drei fortschrittliche Systemansätze stellvertretend für eine Realisierungsform und Systemklasse untersucht. Dabei zeigt sich, dass die Umsetzungen insbesondere im Bereich selbstorganisierter Kooperationsformen sowie im flexiblen Umgang mit netzgestützten Dokumenten (Annotation, Möglichkeiten der Strukturierung durch die Lernenden) misslingen bzw. sich nur Teilaspekte umsetzen lassen. Es fehlt auch bei diesen fortschrittlichen Systemen ein integratives Gestaltungselement, weshalb sich auch die beschriebenen Lernszenarien mit ihnen nicht oder nur sehr beschränkt verwirklichen lassen.

Das Konzept des kooperativen Wissensraums (Kapitel 3) bildet den zentralen Angelpunkt der Arbeit. Es wird zunächst entlang der Metapher des virtuellen, kooperativen Raums (Abschnitt 3.1) weiter präzisiert. Hierzu werden Eigenschaften der Persistenz, verschiedene grafische Repräsentationsformen, Mechanismen der Awareness und Modelle der Zugriffskontrolle und des Zugriffsschutzes untersucht. Der nächste Schritt ist die Einbettung kooperativer Medienfunktionen in das Modell kooperativer Wissensräume basierend auf einem geeigneten Objektmodell (Abschnitt 3.2). Primäre Medienfunktionen werden in Funktionen kooperativer Objekte einer computergestützten Lernumgebung überführt, es ergibt sich eine erste konzeptuelle Vorstellung möglicher Realisierungsformen. Begleitend werden zudem Formen computermediierter Kommunikation in kooperative Wissensräume eingebettet. Entlang der Unterscheidung synchron, asynchron, kanalarientiert und objektbezogen werden insbesondere Strukturierungsformen von Kommunikationsprozessen innerhalb virtueller Räume dargestellt und konkrete Implikationen für synchrone und asynchrone Kommunikationsmechanismen in der angestrebten Konzeption untersucht.

Da aber ein Konzept allein noch keine praktische Umsetzung gewährleistet, wird in Kapitel 4 und Kapitel 5 die Umsetzung des sTeam-Systems erläutert. Das sTeam-System („Strukturieren von Informationen im Team“) wird mit Hilfe einer Reihe von auf den Ergebnissen dieser Arbeit beantragten Forschungsprojekten entwickelt.⁸ Es orientiert sich an der Metapher virtueller Wissensräume, ist konsequent WWW-basiert und verbindet ereignisorientierte Interaktionsformen zwischen Lernenden mit Eigenschaften netzgestützten Dokumentenmanagements. Kapitel 4 untersucht zunächst verschiedene Architekturkonzepte für kooperative Wissensräume. Hierbei gilt es insbesondere, zentralisierte und replizierte Architekturen zu unterscheiden. Auch die schon konzeptuell untersuchten Entwicklungslinien des CSCW/CSCL sowie der MUDs und MOOs gilt es nun aus technisch-architektonischer Sicht gegeneinander abzugrenzen.

Die technische Umsetzung kooperativer Wissensräume (Kapitel 5) verbindet beide Entwicklungslinien und orientiert sich entlang der Darstellung von Server- und Client-Architektur. Aufseiten des Servers stehen eine spezifische Klassen- und Objektstruktur, die Ereignisbehandlung sowie geeignete Schnittstellen und Protokolle im Vordergrund.

Kooperative Lernprozesse zwischen Nutzern werden als Interaktionen zwischen kooperierenden Objekten verstanden. So lassen sich innerhalb der Szenarien entwickelte Kooperationsformen, wie die elektronische Abgabe eines Übungsblatts oder die Weitergabe eines Dokuments in den Rucksack eines anderen Nutzers, als ereignisgesteuerte Interaktionen zwischen aktiven Objekten innerhalb der Lernumgebung umsetzen. Ein ausgefeiltes Attribut- und Rechtemodell erlaubt verschiedene Mechanismen der Verer-

⁸ Vgl. [Hampel & Keil-Slawik 2001a] und [Hampel & Keil-Slawik 2002].

bung und Weitergabe von Berechtigungen. Sie sind Voraussetzungen für den verfolgten nutzerzentrierten Ansatz.

Clientseitig wird die gewonnene Architektur in einem geeigneten Objektmodell fortgesetzt. Weiterhin werden integrative Aspekte verschiedener Clients, wie Verfahren der Verknüpfung spezifischer (Java-) Clients, mit existierenden WWW-Browsern untersucht. Bei sämtlichen architektonischen Betrachtungen steht die benutzerseitige Anpassbarkeit und Erweiterbarkeit im Vordergrund.

Das letzte Kapitel führt die bislang durchgeführten Arbeiten (theoretisches Modell und Konzepte der Implementierung) zusammen und gibt einen Ausblick auf die nächsten Entwicklungsschritte. Diese liegen zum einen in Bereichen eines besseren Verständnisses kooperativer Lernprozesse aus medientechnischer Sichtweise, beziehen sich aber auch auf die weitere Ausgestaltung und Entwicklung der technischen Rahmenarchitektur kooperativer Wissensräume.

1. Computergestütztes kooperatives Lernen

Auf dem Weg zu einem Konzept von virtuellen Wissensräumen im Prozess menschlichen Lernens gilt es, eine Vielzahl von Strömungen und Ansätzen zu berücksichtigen und zu bewerten. Dies ist zum einen notwendig und unumgänglich, da sich eine durchgängige lernförderliche Infrastruktur an den technischen Rahmenbedingungen existierender Infrastrukturen und Standards orientieren muss. In sich geschlossene, proprietäre Lösungen helfen nur wenig und bieten allenfalls ein abgeschlossenes Erprobungsfeld, aber keine weiter gefassten Einsatzmöglichkeiten. Zum anderen ist eine Gestaltung und Entwicklung eines vollständig neuen Systems ohne Rückgriff auf bewährte Standards und technologische Konzepte kaum denkbar.

Zudem scheidet eine vollständige Erfassung oder Analyse vorhandener kooperationsunterstützender Umgebungen aufgrund fehlender Evaluationsergebnisse oder Kriterien zu einer Klassifikation aus.⁹ Auf dem Weg zur kooperativen Wissensorganisation in virtuellen Wissensräumen werden zunächst zwei Strömungen und Entwicklungslinien berücksichtigt:

Dies ist zunächst der Bereich der computergestützten Gruppenarbeit und des computergestützten Lernens (CSCW/CSCL) (Abschnitt 1.6). Motiviert durch den Anspruch der Rationalisierung von Arbeitsprozessen entstanden eine ganze Reihe von Systemen kooperativer Dokumentenverwaltung und Steuerung des Flusses von Dokumenten (Workflow) in Verwaltungs- und Büroprozessen. Systeme des computergestützten kooperativen Lernens adaptieren zumeist derartige Systeme und erweitern sie um spezifische Funktionalitäten des Lernens in Gruppen von Nutzern. Das WWW schafft in neuester Zeit eine Plattform zur durchgängigen Verfügbarkeit derartiger Systeme.

Als zweite Entwicklungslinie finden sich ereignisorientierte Kommunikations- und Interaktionsformen in der Metapher virtueller Welten (Abschnitt 1.7). Ihre Entwicklung (technisch und konzeptuell) ist zumeist privat motiviert und wird dementsprechend von den Nutzern selbstorganisiert vorangetrieben.

Die vorliegende Arbeit verbindet die beiden oben genannten Entwicklungsstränge sowohl konzeptuell als auch von Seiten der technischen Realisierung einer Architektur kooperativer Wissensräume. Kapitel 2 schafft zunächst das theoretische Modell und damit eine Entscheidungsbasis für den konzeptuellen Anspruch der Gestaltung.

⁹ Zudem handelt es sich bei einer Vielzahl existierender Systeme um reine Prototypen zum technischen Erkenntnisgewinn ohne konkretes Einsatzszenarium.

1.1 Bewertungsgrundlage kooperativer Systeme

Eine Klassifizierung von Ansätzen aus den Bereichen der computergestützten Gruppenarbeit und des computergestützten kooperativen Lernens ist durch die Vielzahl der existierenden Prototypen und Applikationen schwierig und nicht ohne größere Evaluationsstudien möglich.

Die vorliegende Arbeit beschreitet den Weg der Identifikation physischer Elemente innerhalb von kooperativen Lernprozessen, die helfen den technischen Mehrwert verschiedener Lösungen zu begreifen (Abschnitt 2.1). Auf diese Weise gelingt es, ganze Klassen von Systemen durch ihre technisch-konzeptuellen Grundlagen zu begreifen. Ein Beispiel sind aktuelle Autorensysteme. Diese bieten eine Autorensicht, d.h. Funktionen des Erzeugens, Arrangierens und Verknüpfens von Materialien werden lediglich den Erstellern (Autoren) von Lehrmaterialien zuteil. – Lernenden wird hingegen eine passive, konsumierende Rolle zugewiesen. Der Anspruch der Bereitstellung von derartigen Handlungen im Umgang mit Medien (Medienfunktionen) für *alle* Lernenden (Selbstorganisation einer Lernumgebung) dokumentiert einen notwendigen Sichtwandel. Dieser Sichtwandel findet seinen vollständigen Ausdruck in den in Abschnitt 2.3 dargestellten kooperativen Medienfunktionen.

Begleitend zeigt das Konzept der Medienbrüche (vgl. Abschnitt 1.5) Defizite in der durchgängigen Verfügbarkeit bereitgestellter (elektronischer) Materialien auf. Es dient als maßgebliche Bewertungsgrundlage für die Qualität einer infrastrukturellen Durchdringung computergestützter Medien in der Lehre.

Im Folgenden werden für die Realisierung der Rahmenarchitektur virtueller Wissensräume relevante Ansätze vorgestellt. Der Fokus der vorgestellten Systeme wird in zweierlei Hinsicht eingeschränkt: Zum einen werden Systeme aus dem Bereich der computergestützten Gruppenarbeit untersucht, die in ihrer Grundarchitektur Elemente kooperativer Medienfunktionen implementieren und dahingehend eine Vorreiterstellung in der konzeptionellen Innovation wesentlicher Elemente kooperativer Arbeit beinhalten (Abschnitt 1.6).

Zum Zweiten finden sich wichtige Elemente aus den Bereichen der MUDs und MOOs (vgl. [Bartle 1990a] und [Cox & Campbell 1994]), die wesentliche Impulse zur technischen und konzeptionellen Architektur der entwickelten Rahmenarchitektur kooperativer Wissensräume darstellen. MUDs und MOOs sind nicht kommerziell und bieten eine recht hohe Universalität in der Anwendbarkeit auf verschiedene Anwendungskontexte. Insofern ist ein Überblick über Aktivitäten von MUDs und MOOs im Bereich der Unterstützung kooperativer Lehr- und Lernprozesse sinnvoll (Abschnitt 1.7).

Nur wenige der vorgestellten Systeme konzentrieren sich explizit auf die Unterstützung kooperativen Lernens, insofern ist eine Übertragung von Konzepten der kooperativen Arbeit auf den Lernkontext notwendig.

Nicht berücksichtigt werden weiterhin Systeme, die in ihrer Konzeption einen sehr stark asynchronen Charakter aufweisen und somit nicht in die Klasse wirklicher kooperativer Umgebungen zu rechnen sind. Denn sie beinhalten beispielsweise keinerlei Mechanismen der gegenseitigen Wahrnehmung oder sind den Bereichen des reinen Doku-

mentenmanagements oder der Workflow-Unterstützung zuzuordnen.¹⁰ Zudem erfolgt eine Konzentration auf netzgestützte Systeme, die eine enge konzeptionelle Nähe zum WWW und seinen entsprechenden Dokumentenformaten aufweisen, geschlossene kooperationsunterstützende Umgebungen oder reine Hypertextsysteme werden nicht berücksichtigt.

In der folgenden Analyse vorhandener Ansätze werden weniger konkrete Eigenschaften der einzelnen Systeme beschrieben, die Betrachtung erfolgt vielmehr unter dem Blickwinkel der Vermeidung von Medienbrüchen und Bereitstellung elementarer Handlungsmöglichkeiten (kooperativer Medienfunktionen) für die Lernenden.¹¹

Das vorliegende Kapitel nennt zunächst einige grundlegende Begriffsbestimmungen und macht deutlich, dass existierende Klassifikationsansätze kein hinreichendes Instrument zu einer medientechnischen Bewertung existierender Ansätze darstellen (Abschnitt 1.2). In einem zweiten Schritt wird in die grundsätzlichen Ideen computergestützten Lernens eingeführt (Abschnitt 1.3) und das Paderborner Modell lernförderlicher Infrastrukturen dargestellt (Abschnitt 1.4). Hieraus ergibt sich ein Modell der Vermeidung von Medienbrüchen (Abschnitt 1.5). Es dient als ein erster Ansatzpunkt der Bewertung der durchgängigen Verfügbarkeit lernförderlicher Infrastrukturen und hilft existierende Ansätze diesbezüglich beurteilen zu können.

1.2 Begriffsbestimmung – Ansätze der Klassifikation

Die Idee, Computer für die Unterstützung von Kommunikations- und Kooperationsprozessen zu nutzen, ist nicht neu. Schon Mitte der 60er Jahre demonstriert Douglas Engelbart sein NLS-System (vgl. [Engelbart & English 1968]) auf der Fall Joint Computer Conference. Zwanzig Jahre später formiert sich ein neues interdisziplinäres Fachgebiet aus Psychologen, Soziologen und Informatikern um den Begriff Computer Supported Cooperative Work (CSCW).

Die Schöpfer des Grundgedankens CSCW, Greif und Cashman, definieren ihn zunächst recht unspezifisch, für sie untersucht das Forschungsgebiet CSCW die Rolle von Computertechnik in Gruppenprozessen.¹² Ähnlich konzentrieren sich Forscher in den darauf folgenden Jahren sowohl auf grundsätzliche Gruppenprozesse, aber auch auf den Einfluss von Technik auf diese Prozesse.¹³

Während CSCW das universelle Arbeitsgebiet und Forschungsfeld bezeichnet, werden unter Groupware landläufig die entsprechenden Systemlösungen und Applikationen verstanden (vgl. [Borghoff & Schlichter 1995]). In der Definition von Ellis et al. wird interessanterweise ausdrücklich betont, dass Menschen ein gemeinsames Ziel und eine gemeinsame Aufgabe besitzen müssen, was Groupware sehr stark von klassischen Multiuser-Systemen unterscheidet. Ein weiteres Standbein dieser Definition ist das „shared

¹⁰ Workflow-Systeme oder Lernumgebungen, die stark didaktische Modelle enthalten, sind typische Beispiele für die Umsetzung der sekundären Medienfunktionen und werden aus diesem Grunde an dieser Stelle nicht vorrangig untersucht (vgl. Abschnitt 2.1).

¹¹ Analog zu den Forderungen von Michelis werden spezifische Systeme untersucht, die den Arbeits- und Lernplatz der Lernenden flexibel erweitern und Interaktionen zwischen Lernenden unterstützen [Michelis 2000, S. 18].

¹² „Over the last half-dozen years Computer Supported Cooperative Work has emerged as an identifiable research field focused on the role of computers in group work“ [Greif 1988, S. 5].

¹³ „[...] CSCW looks at how groups work and seeks to discover how technology (especially computers) can help them work“ [Ellis et al. 1991, S. 39].

environment“, also eine gemeinsame Umgebung, d.h. eine gemeinsame Lokalität und Arbeitsumgebung.¹⁴

Auch wenn einige Visionäre das Potenzial von Computerunterstützung in der Gruppenarbeit und im menschlichen Lernen erkannt haben, existiert bis heute die Vorstellung, Gruppenprozesse alleinig durch den Austausch von Daten zwischen isoliert verwendeten Werkzeugen realisieren zu können (vgl. [Bullen & Bennett 1990]).

Erst schrittweise werden Probleme der Kommunikationsunterstützung, der Koordination von Gruppenprozessen, der gegenseitigen Wahrnehmung und des kooperativen Umgangs mit Materialien erkannt und systematisch untersucht. So betonen beispielsweise Spellman et al., dass Zugriffs- bzw. Benutzerrechte ein maßgebliches Kriterium in Bezug auf den Datenaustausch in einer Gruppe darstellen, aber beispielsweise auch das Finden und Ansprechen anderer Nutzer in einer Gruppenumgebung ernstzunehmende Anforderungen darstellen können ([Spellman et al. 1997]).

CSCW umfasst damit ein Verständnis der Zusammenarbeit und Koordination, Entwicklung von Werkzeugen und Konzepten sowie ihre Bewertung.

Mit der fortschreitenden Entwicklung der Computertechnik und Softwareentwicklung entstehen für das Gebiet der computergestützten Zusammenarbeit eine ganze Reihe von Begrifflichkeiten und Terminologien, die nur schwer voneinander abzugrenzen und zu differenzieren sind.

Auf das in dieser Arbeit untersuchte Feld des kooperativen Lernens bezogen wird die Rolle von Computertechnik in eine ganz ähnliche Terminologie gefasst. Hier beschreibt der Terminus „Computer Supported Cooperative Learning (CSCL)“ das Fachgebiet der Unterstützung von kooperativen Lernprozessen durch Computertechnik und pädagogisch-didaktische Methoden (vgl. [Wessner & Pfister 2001] und [Mühlhäuser 1995]).

Trotz oder gerade aufgrund dieser recht universellen Definition existiert kein einheitliches Verständnis der Eingrenzung des Bereichs CSCL. Dieses weist eine enge verwandtschaftliche Nähe zum Fachgebiet CSCW auf. Es definiert in dieser Form jedoch keine Systematik oder Konfiguration menschlichen Lernens, vielmehr fasst es verschiedene Forschungsansätze zusammen. Hierzu zählen Fragestellungen der Unterstützung von Gruppen in Problemlöseprozessen, des kooperativen Strukturierens von Materialien oder der Integration synchroner oder asynchroner Kommunikationsmechanismen in Lernprozessen. Die vorliegende Arbeit stützt sich weniger auf pädagogisch-didaktische Methoden, sie beleuchtet basierend auf einer medientechnischen Perspektive technische Rahmenbedingungen von CSCL-Prozessen. Dies geschieht unter dem Blickwinkel der lebensweltlichen Praxis, wie die von Keil-Slawik geforderte Alltagstauglichkeit von Systemen:

„[...] sowohl unter Bezug auf die Prozesse der Herstellung als auch der Nutzung von Informatiksystemen erweist sich die Alltagstauglichkeit als ein entscheidendes Qualitätsmerkmal, um innovative Ideen in nützliche Technik, grundlegende Prinzipien in taugliche Produkte und formale Beschreibungen in verständliche Dokumentationen umzusetzen“ [Keil-Slawik 2000a, S. 199].

Eine ganze Reihe von Forschern sprechen im Zusammenhang mit den Akronymen CSCW und CSCL von „Computer Supported Collaborative Work/Learning“, was den

¹⁴ „[...] computer-based systems that support groups of people engaged in a common task (or goal) and that provide an interface to a shared environment“ [Ellis et al. 1991, S. 40].

Aspekt der gegenseitigen Zusammenarbeit noch stärker betonen soll. Speziell kooperationsunterstützende Umgebungen werden in analoger Weise als „Collaborative Environments“ bezeichnet (vgl. [Dommel & Aceves 1997, S. 24]).

Durch die negative Besetzung des Begriffs „Kollaboration“ wirkt eine direkte Übersetzung ins Deutsche recht unglücklich. Im Folgenden wird aus diesem Grunde nicht explizit zwischen den Begriffen Kooperation und Zusammenarbeit unterschieden. Es wird vom computergestützten, kooperativen Problemlösen und Lernen bzw. von computergestützten, kooperativen Umgebungen gesprochen.

In dem Bereich der synchronen Kooperationsunterstützung werden landläufig drei mögliche Typen von kooperativen Sitzungen unterschieden: Hierzu zählen „face-to-face sessions“ als reale Treffen innerhalb elektronischer Besprechungsräume oder mit speziellen Infrastrukturen ausgestatteter Räume, „distributed sessions“ als virtuell verankerte Treffen, in denen Teilnehmer über einen Computer und klassische Kommunikationsmedien wie Telefon miteinander kommunizieren, sowie „mixed-mode sessions“ als Mischformen, in denen ein Teil der Teilnehmer real und die übrigen virtuell kommunizieren [Ellis et al. 1991, S. 46]. In der vorliegenden Arbeit wird speziell letztere Form des kooperativen Lernens berücksichtigt.

Eine enge verwandtschaftliche Nähe weisen die Fachgebiete CSCW und CSCL zum Begriff des kooperativen Wissensmanagements, dem so genannten Knowledge Management, auf. Auch wenn die Grundintention des kooperativen Wissenserwerbs (CSCL) und der kooperativen Arbeit (CSCW) mit der kooperativen Wissenskonstruktion gleichzusetzen ist, wird in der Wirtschaftsinformatik meist unter Wissensmanagement die Wissenskonstruktion und Verwaltung innerhalb von Organisationen verstanden. Dies bezieht sich auf Wissen, das Mitarbeiter in Bezug auf Geschäftsprozesse erwerben, z.B. Wissen über Prozesse innerhalb der Organisation, Hintergrundinformationen bzgl. herangezogener Entscheidungsgrundlagen oder Hintergrundwissen über die Kunden (vgl. [Probst et al. 1999]).

Ein eigenes Schwerpunktprogramm der Deutschen Forschungsgemeinschaft (DFG) löst sich von diesem auf Wirtschaftsprozesse bezogenen Verständnis und spricht allgemein von „Wissenskommunikation in Gruppen“¹⁵. Die vorliegende Arbeit lehnt sich an die Idee kooperativer Wissenskonstruktion an und untersucht Möglichkeiten virtueller Wissensräume im Kontext kooperativen Lernens.

Mit der Entwicklung kooperationsunterstützender Systeme sind verschiedene Klassifikationsansätze mit dem Ziel einer systematischen Einordnung und Abgrenzung entstanden. Die vielleicht bekannteste Unterscheidung ist die Raum-Zeit-Matrix von [DeSanctis & Gallupe 1987] und [Johansen 1988]. Kooperative Prozesse und Systeme werden nach den Dimensionen der räumlichen Nähe der Gruppenmitglieder und ihrer zeitlichen Zusammenarbeit unterschieden. Auf diese Weise wird differenziert, inwieweit Gruppenmitglieder in einem Raum bzw. örtlich verteilt arbeiten und diese Zusammenarbeit zu einer Zeit (synchron) oder zu verschiedenen Zeiten (asynchron) stattfindet.

Eine zweite Gruppe von Forschern versucht eine Klassifikation nach der Art und Funktion der zugrunde liegenden Applikationen. Sie unterscheiden die Gruppen „message systems“, „multiuser editors“, „group decision support systems and electronic

¹⁵ Vgl. das Schwerpunktprogramm der Deutschen Forschungsgesellschaft:
<http://www.wissenskommunikation.de>, Stand 1.12.2001.

meeting rooms“, „computer conferencing“, „intelligent agents“ und „coordination systems“ [Ellis et al. 1991, S. 42]. Gleichzeitig wird betont, dass sich die gewählten Kategorien sehr stark überlappen. Damit erscheint der Ansatz in gewisser Weise willkürlich gewählt. Zudem ist die Kategorisierung nicht auf wirklichen elementaren Unterstützungsfunktionen für Benutzer zugeschnitten, sie spiegeln mehr die verschiedenen Forschungsfacetten zum Zeitpunkt der Klassifizierung wider.

Reid wählt eine um einiges einfachere Klassifikation. Sie unterscheidet lediglich die Dienste computermediiertes Kommunikation: E-Mail, News und Chat. Diese werden nach der zeitlichen Komponente einer Zusammenarbeit nach asynchron und synchron sowie nach der Anzahl der Rezipienten differenziert [Reid 1994, S. 11ff]. Sie unterscheidet weiterhin Kommunikation, die nach dem Sender-Empfänger-Prinzip stattfindet, und rezeptive Formen der Kommunikation, wie sie beispielsweise für Massenmedien typisch sind. Damit ist eine E-Mail eine Mitteilung von einem Nutzer an einen anderen, News hingegen sind Nachrichten von einem Nutzer an eine Anzahl von Rezipienten, wobei die Mitteilungen themenorientiert auf einem Server abgelegt werden. Chat ist schließlich eine Form von synchroner Kommunikation, die sich an genau einen Gesprächspartner, aber auch eine Gruppe von Teilnehmern richten kann.

Aufbauend auf den Ansatz von [Ellis et al. 1991] erfolgt die Klassifikation anhand der Unterstützungsfunktionen nach [Teufel et al. 1995, S. 27]. Teufel unterscheidet verschiedene Systemklassen entlang der Unterstützungsfunktionen Kommunikationsunterstützung, Kooperationsunterstützung, Koordinationsunterstützung und ordnet sie den Systemklassen Kommunikation, Gemeinsame Informationsräume und Workflow Management zu. Für eine ausführliche Abhandlung funktionaler Klassifikationsschemata vergleiche [ter Hofte 1998, S. 30].

Für den Bereich computergestützter kooperativer Lehrsysteme finden sich nur wenige Klassifikationsansätze. Lediglich Wessner und Pfister unterscheiden CSCL-Systeme nach sieben Dimensionen: Diese umfassen *Ort* (lokales versus verteiltes CSCL), *Zeit* (synchrones, asynchrones Lernen), *Symmetrie* (symmetrisches versus asymmetrisches CSCL – „Flussrichtung“ von Wissen), *Direktivität* (angeleitetes versus selbstgesteuertes CSCL), *Dauer* (persistent versus transient), *Wissen* (individuelles versus verteiltes Wissen) und *Gruppengröße* (kleine versus große Lerngruppen). Der Faktor Direktivität beschreibt, inwieweit der Lernprozess durch Personen oder Programme (Agenten) angeleitet wird oder die Lerngruppe als selbstorganisierte Einheit auftritt (vgl. [Wessner & Pfister 2001, S. 251ff]).

Einen Überblick über verschiedene Klassifizierungsansätze sowie ihre Auswirkungen auf die Gestaltung von Systemen untersucht [Fischer 2001, S. 21ff].

Insgesamt ist anzumerken, dass sowohl die recht triviale Unterscheidung nach der Raum-Zeit-Matrix, welche lediglich zeitlich und räumlich verteiltes Arbeiten und face-to-face Situationen erfasst, als auch Ansätze funktionaler Klassifikation nur wenig konkrete Kriterien zu einer Bewertung von CSCL-/CSCW-Systemen in Bezug auf den Grad und die Qualität ihrer kooperationsunterstützenden Wirkung zulassen. Die meisten Systeme bieten eine hybride Mischung aus asynchronen und synchronen Elementen sowie enthalten verschiedene funktionale Komponenten, die eine Reihe von Unterstützungsfunktionen simultan anbieten.

Die Klassifikation von [Wessner & Pfister 2001] fasst eine ganze Reihe von Faktoren in eine Art Kriterienkatalog. Jedes der vorgestellten Kriterien besitzt sicherlich einen großen Einfluss auf die Art und Qualität einer zu untersuchenden Lernumgebung, sie

bieten jedoch kein generisches theoretisches Fundament zur Klassifizierung beliebiger Funktionen im Hinblick auf den Umgang mit Medien.

1.3 Lernerfolg und computergestütztes kooperatives Lernen

Ähnlich schwierig wie eine theoretische Aufarbeitung des Phänomens des kooperativen Lernens ist es, Aussagen über den Einfluss von Computerunterstützung auf den Lernprozess und den Lernfortschritt zu treffen.

Behavioristische Lernansätze und Theorien prägten lange Jahre die Entwicklung von computergestützten Lernsystemen. Sie finden ihren Ausdruck in einer Vielzahl von Autorenumgebungen zur Entwicklung von computergestützten Lehrmaterialien. Hierbei werden Lernende als passive Konsumenten verstanden, denen ein Lerninhalt in aufbereiteter Form präsentiert wird. Das Programm übernimmt die Steuerung des Lernprozesses. Durch geeignete Stimuli wird der Lernende angeregt Lernleistungen zu erbringen. Ein typisches Beispiel sind die weit verbreiteten Multiple-Choice-Fragenkataloge. Dem Lernenden wird meist lediglich die Möglichkeit eröffnet, eine Anzahl von Wiederholungen oder erneuten Versuchen in der Beantwortung der Fragen durchzuführen. Der Anspruch, durch eine derartige Vorgehensweise netzgestützt Wissen zu vermitteln, ist als wenig realistisch einzuschätzen.

Auch wenn einige Forschergruppen nachweisen, dass kooperatives Lernen dem Lernerfolg dienlich zu sein scheint (vgl. [Johnson & Johnson 1990]), ist es gefährlich, diese Aussage auf kooperative Lernumgebungen zu übertragen.

Innerhalb einer kooperativen Lernumgebung gehen die Teilnehmer nicht in gleicher Form sozial und kommunikativ miteinander um, wie es in der Präsenzsituation eines realen, kooperativen Lernprozesses der Fall ist. In der reinen Zusammenarbeit über das Medium Computer fehlen beispielsweise Gesten, Mimik und eine ungezwungene nicht-technologisch unterstützte Form der Kommunikation. Die im Rahmen der vorliegenden Arbeit vorgestellte Konzeption einer kooperationsunterstützenden Umgebung konzentriert sich primär auf Situationen der Präsenzlehre und asynchrone Formen der Zusammenarbeit. Aus diesem Grunde werden obige Probleme im Rahmen dieser Arbeit nicht weiter untersucht.

Zunächst kann die Aussage getroffen werden, dass zu einer erfolgreichen Einführung von kooperationsunterstützenden Systemen in der Lehre sicherlich die Überwindung einer kritischen Masse notwendig ist, um Frustration bei den Benutzern zu vermeiden. Anwender fühlen sich frustriert, wenn sie in einer virtuellen Umgebung und deren elektronischen Lehrmaterialien auf ihre Fragen keine Antworten erhalten oder in den virtuellen Gruppenarbeitsräumen nur wenige Teilnehmer vorfinden.¹⁶

Gleichzeitig ist die Schaffung einer Akzeptanz aufseiten der Lernenden für eine kooperationsunterstützende Umgebung ein kritisches Moment. Akzeptanz ist zu einem großen Anteil von der Alltagstauglichkeit der geschaffenen lernförderlichen Infrastruktur abhängig (vgl. [Brennecke & Keil-Slawik 1995a]), ist aber ebenfalls von den Fähigkeiten der Lernenden zur kooperativen Zusammenarbeit beeinflusst. So bemerken Miao und Haake im Zusammenhang mit ihren Thesen zum sozialen Konstruktivismus, dass die psychologischen Fähigkeiten des Menschen mit der Teilnahme an kooperativen Pro-

¹⁶ Ähnliche Erfahrung haben z.B. Schenk und Schwabe in ihrem Projekt „Cuparla“ (Computergestützte Parlamentsarbeit) aus dem Bereich des CSCW gemacht: „Gemeinderäte waren z.B. sehr schnell demotiviert, wenn ihre E-Mails durch Kollegen nicht beantwortet wurden, und stellten dann selbst die eigene E-mailnutzung ein“ [Schenk und Schwabe 2000, S. 67].

zessen wachsen (vgl. [Miao & Haake 2001, S. 2ff.]). Also auch der kooperative Prozess selbst ist als Lernprozess zu bezeichnen. Ein derartiges Phänomen macht speziell die Evaluation von kooperativen Lernprozessen schwierig.

Eine gründliche Evaluation von CSCL-Systemen ist unumstritten notwendig, sie ist jedoch nur möglich, wenn ein präziser Evaluationsrahmen definiert ist. Dies setzt die Betrachtung von pädagogischen, technischen, organisatorischen und kulturellen Bedingungen voraus. Pfister und Wessner betonen in diesem Zusammenhang, dass insbesondere die Evaluation selbst als Lernprozess begriffen werden muss, d.h. die Kriterien einer Evaluation müssen selbst formativ evaluiert und damit schrittweise fortentwickelt werden (vgl. [Pfister & Wessner 2000a, S. 139ff.])

Aussagen über die Auswirkungen von computergestützten kooperativen Prozessen auf menschliches Lernen sind dementsprechend nur schwer zu belegen. Zudem sind sie für die im Rahmen dieser Arbeit verfolgten Konzeptionen aus medientechnischer Sicht nur von zweitrangiger Bedeutung.

Aus den Forschungen der Psychologie kann zunächst die Aussage getroffen werden, dass computergestützte Kooperation zumindest keine schlechtere Effektivität wie reale Kooperationsformen aufweist. (Diese Aussage bezieht sich auf den konkreten Vergleich elementarer Formen der Zusammenarbeit mit und ohne Computerunterstützung.) Plötzner et al. beschreiben beispielsweise eine erste interessante Untersuchung zur Effektivität von Problemlöseprozessen im Vergleich von Angesicht zu Angesicht und rechnergestützter Kooperation (bezogen auf das Beispiel Videokonferenz). Es zeigte sich bei der Aufgabe Begriffsnetze kooperativ erstellen zu müssen, dass keine bedeutsamen Unterschiede bzgl. der Problemlöseeffektivität bestanden. So wurden die Aufgaben in einer computervermittelten Kooperation ähnlich gelöst wie in einer von Angesicht zu Angesicht stattfindenden Kooperation (vgl. [Plötzner et al. 2000]).

Gleichzeitig ist jedoch unumstritten, dass computergestützte Kooperation in Abhängigkeit der zeitlichen und räumlichen Distanz der Lernenden zu erheblichen Problemen der Abstimmung, Orientierung und gegenseitigen Wahrnehmung führt.

Zwangsläufig führt der Einsatz von reinen Tele-Learning-Szenarien in der Lehre zu einer Verringerung des Kontextes, in dem menschliches Lernen stattfindet. Es fällt Lernenden schwer zu erkennen, welche anderen Lernenden an einem Lernprozess teilnehmen, an welcher Stelle innerhalb von Materialien sie sich befinden oder wie das weitere Vorgehen von Lernpartnern aussieht. Generell verursacht virtuelles Lernen eine Reduktion des Kontextes gegenüber realen face-to-face Situationen.

Eine Reihe von Ansätzen versuchen insbesondere die Koordination und das Vorgehen der Lernenden innerhalb der Lernumgebung zu koordinieren. Hier sind insbesondere die Lernprotokolle als Systematisierung von Kooperationsprozessen zu nennen [Hesse et al 1997], aber auch die Lernnetze [Pfister et al. 1998] als Repräsentation des Wissens von Lernenden als Netzstruktur.

In der Phase des Entwurfs eines technischen Systems entstehen notwendige Implikationen aus dem sozialen Umfeld des Einsatzes und getroffener Designentscheidungen – umgekehrt beeinflusst die gewählte Technik auch die soziale Struktur des Einsatzfeldes. Einen derartigen Effekt charakterisieren O’Day et al. als „social-technical design circle“, der es notwendig macht, sowohl soziologische Faktoren als auch technische Rahmenbedingungen eines kooperationsunterstützenden Systems zu berücksichtigen (vgl. [O’Day et al. 1996, S. 160ff.]). Ziel ist also eine gemeinsame Entwicklung und sorgfältige Abstimmung von sozialen und technischen Belangen, denn Technik kann nur technische Probleme lösen, aber Technik beeinflusst soziale Systeme.

Resümierend ist festzustellen, dass der Lernerfolg von computergestützter Kooperation nur schwer messbar und qualitativ zu bewerten ist. Es fehlt oftmals ein realistischer Rahmen der alltagstauglichen Einbettung in Lernprozesse. Lernerfolge treten zumeist erst nach längerem Einsatz entsprechender lernförderlicher Infrastrukturen ein. Eine Vergleichsevaluation verbietet sich wegen ihrer Komplexität und der gegenseitigen Beeinflussung des Versuchsfeldes, es bleibt lediglich die formative Evaluation. Gleichzeitig sind technologische Rahmenbedingungen und Beeinflussungsfaktoren eines Lernprozesses nur schwer zu isolieren und nicht zuletzt existieren nur wenige alltagstaugliche Systeme und Infrastrukturen, die einen praktischen Einsatz und eine längerfristige Evaluation zulassen.

Für den Entwurf einer kooperativen Lernumgebung ergeben sich eine große Anzahl technischer und sozialer Faktoren, die wesentliche Bedingungen für einen erfolgreichen Einsatz darstellen: In der Literatur (vgl. [Mynatt et al. 1997]) finden sich z.B. auf der technischen Seite Faktoren einer Persistenz der Umgebung und ihrer Materialien, das Vorhandensein verschiedener Interaktionstechniken sowie die Möglichkeit von Echtzeitinteraktion und Mehrbenutzerfähigkeit. Auf der Seite der sozialen Faktoren werden Voraussetzungen genannt wie das Vorhandensein eines gemeinsamen Ortes und einer Art von Zugehörigkeitsgefühl zu diesem Ort: „articulation of a persistent sense of location“ [ebd., S. 211]. Schließlich wird recht allgemein eine Kopplung von technischen Mechanismen und sozialen Handlungen gefordert, also eine feinfühligte Abstimmung von Technik und sozialen Interaktionen.

Die vorliegende Konzeption konzentriert sich zunächst auf die alltagstaugliche Einbettung lernförderlicher Infrastrukturen in den Lehrbetrieb und entwickelt parallel ein Modell zu ihrer Bewertung aus medientechnischer Perspektive.

1.4 Lernförderliche Infrastrukturen

Auf dem Weg zur theoretischen Fundierung der vorliegenden Arbeit, dem Konzept der Medienfunktionen, sind die Erfahrungen der Arbeitsgruppe Informatik und Gesellschaft in Paderborn von großer Bedeutung.

Hier wird schon seit einigen Jahren das Ziel verfolgt, lernförderliche Infrastrukturen aufzubauen und unter dem Blickwinkel der Alltagstauglichkeit zu evaluieren. Alltagstauglichkeit bedeutet, technische Infrastrukturen möglichst unauffällig und wenig störend in den Wahrnehmungs- und Handlungsraum des Menschen zu integrieren sowie Infrastrukturen und Systeme von der prototypischen Erprobung in den alltäglichen Gebrauch zu überführen.¹⁷

Eine Thematisierung der Rolle von Technik in Bezug auf Prozesse der Wissensverarbeitung (vgl. [Keil-Slawik 1990], [Keil-Slawik 1991]) hat neben der Betrachtung existierender Systeme und Lösungen (vgl. [Keil-Slawik et al. 1997]) zum Aufbau der Paderborner DISCO (Digitale Infrastruktur für computerunterstütztes kooperatives Lernen, (vgl. [Brennecke et al. 1997a], [Keil-Slawik 1998b] [Keil-Slawik 1999], [Hampel et al. 2001]) geführt. Die Paderborner DISCO beinhaltet innovative technologische Infrastrukturen wie digitale Seminarräume und Hörsäle. Sie umfasst jedoch insbesondere neue organisatorische Formen der Lehrveranstaltungen und Softwaresysteme zur

¹⁷ Beaudouin-Lafon und Karsenty wählen in diesem Zusammenhang den Begriff „transparent groupware systems“ [Beaudouin-Lafon & Karsenty 1992, S. 171]. Sie legen die Betonung auf Systeme, die ohne spezielle Hardware und spezifische Voraussetzungen auskommen, sich also reibungslos in den täglichen Umgang mit Computersystemen einfügen.

Unterstützung der Lehre. Sie wird seit einigen Jahren sowohl in kleinen Seminarveranstaltungen (25-50 Teilnehmer) als auch in Grundstudiumsveranstaltungen (z.B. Grundlagen der Systemgestaltung mit ca. 500 Teilnehmern) eingesetzt und erprobt.

Um eine große Anzahl von Dokumenten unterschiedlichster Lehrveranstaltungen verwalten und den Studierenden zugänglich machen zu können, wird seit Mitte der 90er Jahre ein Dokumentenmanagementsystem (Hyperwave, vgl. [Andrews et al. 1995], [Keil-Slawik 2000b]) eingesetzt. Es bietet asynchrone Formen der Dokumentenverwaltung und ist in seiner Grundarchitektur wegen z.B. fehlender Mechanismen der gegenseitigen Wahrnehmung zunächst kein kooperatives System. Mit Hilfe spezifischer, architektonischer Erweiterungen werden Formen der kooperativen Erstellung sowie langfristigen Pflege und Wartung von digitalen Lehrmaterialien systematisch ausgebaut (vgl. [Brennecke & Selke 2000] und [Brennecke & Selke 2001]).

Parallel werden neuartige multimediale Formen der Wissenspräsentation und Interaktion entworfen und teils in interdisziplinärer Zusammenarbeit erprobt (vgl. [Hampel et al. 1998a], [Hampel et al. 1998b], [Hampel et al. 1998c] und [Hampel & Nowaczyk 1999]).

Ziel des Aufbaus einer derartigen lernförderlichen Infrastruktur ist es insbesondere, die Voraussetzungen zu schaffen, Aspekte des kooperativen Lernens unter Alltags- und nicht nur unter Laborbedingungen studieren zu können. Dies ist eine entscheidende Voraussetzung, um verlässliche Ergebnisse in Bezug auf situiertes Alltagshandeln erheben zu können, die zu einer produktiven Weiterentwicklung der Lernumgebung führen.

Aus den Erfahrungen im Bereich der lernförderlichen Infrastrukturen ergeben sich drei Dimensionen ihrer qualitativen Bewertung. Dies ist zum einen die Alltagstauglichkeit verwendeter Werkzeuge und Systeme. Nicht der zu evaluierende Versuchsrahmen ist entscheidend, sondern die Einbettung in alltägliche Prozesse menschlichen Lernens. Als zweite Dimension ist die durchgängige Verfügbarkeit der Lernumgebung mit den in ihr verwalteten Materialien entscheidendes Kriterium. Unter genauerer Betrachtung der durchgängigen Verfügbarkeit wird deutlich, dass seine qualitative Bewertung an technologisch erzwungenen Wechsel der Medien, also Brüchen im Umgang mit Medien zu messen ist. Schließlich erfolgt die qualitative Bewertung der Handlungsmöglichkeiten der Lehrmaterialien (Medien) durch das Konzept der Medienfunktionen (vgl. Abschnitt 2.1). Der folgende Abschnitt führt zunächst in das Konzept der Medienbrüche ein.

1.5 Medienbrüche – durchgängige Verfügbarkeit

Die durchgängige Verfügbarkeit eines Mediums lässt sich durch das Konzept der Medienbrüche (vgl. [Keil-Slawik & Selke 1998b, S. 14ff.] erfassen. Dabei sind Medienbrüche technologisch erzwungene Wechsel in dem Medium bei der Ausübung von Handlungen an Materialien (Medienfunktionen). Sowohl das Konzept der lernförderlichen Infrastrukturen (vgl. Abschnitt 1.4) als auch das im Rahmen dieser Arbeit entwickelte Konzept kooperativer Wissensräume versucht Medienbrüche weitgehend zu minimieren. Resultierend ist die Alltagstauglichkeit einer lernförderlichen Infrastruktur maßgeblich durch die Reduzierung vorhandener Medienbrüche gekennzeichnet.

In der Praxis gelingt eine vollständige Auflösung von Medienbrüchen nur schwerlich. Eine inhärente Mischung von traditionellen Medien und digitalen Neuen Medien führt an einer Vielzahl von Stellen zu einem technologisch erzwungenen Wechsel des verwendeten Mediums. Beispielsweise entsteht in jeder Situation ein Medienbruch, wo elektronische Lehrmaterialien ausgedruckt und mit persönlichen Notizen versehen wer-

den. Diese können in späteren Lernsituationen nicht mit neuen Materialien verknüpft werden und liegen nicht an allen Lernorten vor.

Von der konzeptionell-theoretischen Seite können Medienbrüche nur durch eine kontinuierliche Bereitstellung und gegenseitige Anwendbarkeit von elementaren Handlungsmöglichkeiten an Materialien (primäre Medienfunktionen, vgl. Abschnitt 2.1) vermieden werden: In einem derartigen Ansatz existiert jede Medienfunktion unabhängig von der Gestalt und Form der Materialien und speziell entkoppelt von der Reihenfolge der genutzten Medienfunktionen.

Lediglich organisatorische Rahmenbedingungen wie die Zugreifbarkeit (Benutzerrechte) steuern im kooperativen Umfeld die Medienfunktionen von Materialien. Ohne das Konzept der Medienbrüche weiter systematisieren zu wollen, zeigen sich typische Medienbrüche im Umfeld kooperationsunterstützender Systeme und lernförderlicher Infrastrukturen in der Praxis an vielen Stellen. Im Vorgriff auf die in Abschnitt 2.5 vorgestellten Szenarien zum Lernen in virtuellen Wissensräumen und einer Bewertung verschiedener kooperationsunterstützender Systeme werden elementare Anforderungen in den Handlungsmöglichkeiten und der durchgängigen Verfügbarkeit im Umgang mit Materialien deutlich:

- Materialien müssen unabhängig technischer Rahmenbedingungen für jeden Lernenden erzeugbar sein. Sind Lernende gezwungen, eigene Materialien durch eingeschränkte Fähigkeiten innerhalb eines kooperationsunterstützenden Systems außerhalb der Lernumgebung ablegen und verwalten zu müssen, liegt ein typischer Medienbruch vor.
- Materialien müssen unabhängig technischer Rahmenbedingungen wie Größe und Form an alle Lernorte übertragen und zwischen Lernenden ausgetauscht werden können (durchgängige Verfügbarkeit).
- Materialien müssen beliebig unabhängig ihrer Form verknüpfbar sein. Ein Beispiel ist das Anfügen einer Annotation (Referenz auf ein anderweitiges Dokument) an einen Film. In diesem Sinne ist eine fehlende Möglichkeit zur Annotation als ein Medienbruch zu verstehen.
- Die Arrangierbarkeit und Strukturierbarkeit von Materialien darf nicht durch technische Rahmenbedingungen oder ihren Ablageort beeinflusst werden. Beispielsweise müssen Lernende Materialien für sich persönlich arrangieren können, ohne den Inhalt des jeweiligen Mediums verändern zu müssen.

Medienbrüche sind in dieser Form ein auch von Technikern untersuchbares Kriterium für die durchgängige Verfügbarkeit von Materialien. Sie helfen Unterstützungsfunktionen auf technische Gegebenheiten zurückzuführen und Defizite in vorhandenen lernförderlichen Infrastrukturen zu erkennen. Kooperationsunterstützende Systeme lassen sich in gleicher Weise durch die gezielte Identifikation von Medienbrüchen gegeneinander abgrenzen.

Die in Abschnitt 2.5 aufgezeigten Szenarien sind in vielfältiger Weise geeignet, Medienbrüche in der Unterstützung kooperationsunterstützender Systeme zu erkennen und dienen demgemäß als Bewertungsgrundlage für die untersuchten existierenden Systeme. In den folgenden Abschnitten werden verschiedene Systemklassen und Ansätze durch ihre Eigenschaften der Reduzierung von Medienbrüchen bewertet. Es zeigt sich, dass alle autorenzentrierten Umgebungen, wie klassische hypermediale Autorensysteme, aber auch das WWW wenig geeignet scheinen Medienbrüche zu reduzieren. In Bezug auf die Selbstorganisation und Selbststrukturierung der Umgebung durch die Lernenden erweist sich der Bereich der MUDs und MOOs als richtungweisend. Die im Rahmen der vorlie-

genden Arbeit entwickelte Konzeption kooperativer Wissensräume orientiert sich an dem Anspruch der weitgehenden Reduzierung auftretender Medienbrüche durch die Bereitstellung kooperativer Medienfunktionen bei gleichzeitiger Selbstorganisation der Lernenden.

1.6 Entwicklungslinie: CSCW/CSCL

Wie schon in der Einleitung und Begriffsbestimmung in Abschnitt 1.2 deutlich gemacht, orientiert sich der Bereich des Computer Supported Cooperative Work (CSCW) und Computer Supported Cooperative Learning (CSCL) zumeist an aus dem Arbeits- und Büroumfeld motivierten Fragestellungen. Dementsprechend ist die Anzahl der existierenden Systeme zu groß und ihr Einsatzfeld zu weitläufig, um einen vollständigen Überblick zu ermöglichen. Auch existieren wenig generische Systeme und Ansätze, die nicht einen konkreten Anwendungsbereich computergestützter Arbeit oder einen spezifischen Aspekt menschlichen Lernens berücksichtigen. Zugleich demonstrieren Systeme des CSCL und CSCW speziell in einzelnen Aspekten der Dokumentenverwaltung und der netzgestützten Bereitstellung von Materialien technologisch umsetzbare Realisierungen und in der Praxis erprobte Systemlösungen. Wie bereits dargestellt, sind Medienbrüche ein Kriterium der Bewertung solcher Systeme, weitere kommen mit dem Konzept der Medienfunktionen (vgl. Abschnitt 2.1) sowie der Selbstverwaltung und Selbstorganisation der Lernenden hinzu (vgl. Kapitel 3). Als alternative Herangehensweise zu einer vollständigen Erfassung sämtlicher Aspekte von Systemen des CSCL und CSCW werden in Abschnitt 2.7 exemplarisch drei viel versprechende Vertreter unterschiedlicher Systemklassen auf ihre Umsetzbarkeit der zuvor formulierten Szenarien der Nutzung kooperativer, virtueller Wissensräume hin untersucht.

Eine erste Einschränkung des Untersuchungsfeldes gelingt durch eine Konzentration auf Systeme, die wesentliche Anteile kooperativer Elemente enthalten. Reine Dokumentenmanagementsysteme oder netzgestützte Datenbanksysteme werden durch meist fehlende Fähigkeiten des Austausches von Materialien zwischen Nutzern (Lernenden) oder Fähigkeiten der gegenseitigen Wahrnehmung (Awareness) nicht berücksichtigt. In ähnlicher Weise werden rein asynchrone Systeme, die auf einer Nachrichten-Metapher beruhen, also lediglich sehr eingeschränkte Möglichkeiten des Arrangierens und Strukturierens von Materialien bieten, nur am Rande betrachtet. Analog besitzen sitzungsorientierte (synchrone) Systeme wie Konferenzsysteme oder Shared-Application-Ansätze aufgrund fehlender Persistenzeigenschaften der bereitgestellten Sitzungen mit den enthaltenen Materialien nur wenig Relevanz.

Im Folgenden werden zunächst Systeme aus dem Bereich des CSCW vorgestellt. Bewusst wird keine (wie sonst in der Literatur üblich) strikte Unterscheidung nach synchronen und asynchronen Systemen vorgenommen (Abschnitt 1.6.2). Die meisten modernen Ansätze verbinden Elemente aus beiden Bereichen. Anschließend werden eine Anzahl kooperativer Lehr- und Lernumgebungen vorgestellt (Abschnitt 1.6.3), um schließlich die Eignung WWW-gestützter Ansätze im Umfeld kooperativer Wissensräume zu untersuchen (Abschnitt 1.6.4).

1.6.1 CSCW

Die älteste Idee der computergestützten Gruppenarbeit stammt sicherlich von Engelbart und English in ihren Arbeiten zum „Augmentation System“, also der Unterstützung menschlicher Fähigkeiten durch Computer (vgl. [Engelbart & English 1968]). Auch wenn es sich in diesen frühen Versuchen lediglich um eine Art „application sharing“- oder „shared windows“-Ansatz handelt, d.h. alle Teilnehmer sehen das Bild einer An-

wendung eines ausgewählten Teilnehmers, ist doch der frühe Zeitpunkt und die Verknüpfung unterschiedlicher Ideen bemerkenswert. Erst zu einem erheblich späteren Zeitpunkt umgesetzte Konzepte der Kopplung, wie das eines gemeinsamen, vergrößerten Mauszeigers (Telepointer), werden von den Autoren berücksichtigt.

Thematische Ausrichtung der Forschungen ist die Unterstützung von Konferenzen, also das Präsentieren und Begutachten von Beiträgen. Es wird bewusst Wert auf das kooperative Arbeiten mit Dokumenten gelegt. Wenn die Visionäre der computergestützten kooperativen Arbeit folglich von ihrem „On-Line conference arrangement“ sprechen, ist hiermit ein spezielles kooperatives Arrangement zur Unterstützung kooperativer Gruppensitzungen gemeint.¹⁸ Ohne die Ergebnisse der Untersuchungen von Engelbart und English weiter vertiefen zu wollen, fallen eine ganze Reihe wesentlicher Konzepte und Ergebnisse auf, die für die im Rahmen der vorliegenden Arbeit konzipierte kooperative Lernumgebung durchaus eine Relevanz besitzen. So wird neben der Gestaltung von Software-Lösungen auch die gesamte Infrastruktur (speziell gestaltete Tischarrangements, geneigte Monitore) berücksichtigt. Des Weiteren konzentrieren sich die Forschungen auf die Idee, elektronisch Dokumente in den Wahrnehmungs- und Handlungsraum des Menschen zu bringen, d.h. Kooperation findet durch gezielte Computerunterstützung im Umfeld von Präsenzsituationen statt. Auch wenn die Unterstützung von Korrektur- bzw. Begutachtungsprozessen (Review-Prozessen) im Vordergrund steht, handelt es sich doch um ein Vorgehen, das dem Konzept der Umsetzung von technischen Unterstützungsfunktionen im Umgang mit Materialien sehr nahe kommt. Einen umfassenden Überblick über die geschichtliche Entwicklung des Application Sharing bis hin zu frühen Ansätzen des CSCW gibt Greenberg in seinen Arbeiten (vgl. [Greenberg 1990]).

1.6.2 Systeme mit synchronen und asynchronen Elementen

Einen interessanten Prototyp für die Erprobung der Verschmelzung von synchronen und asynchronen Arbeitsformen stellen Sohlenkamp und Chwelos in ihrer DIVA [Sohlenkamp & Chwelos 1994] Arbeitsumgebung vor. Metaphorisch angelehnt an ein virtuelles Büro werden sowohl asynchrone und synchrone Arbeitstechniken sowie „Communication“, „Cooperation“ und „Awareness“ integriert [ebd., S. 332]. DIVA erlaubt angelehnt an eine Metapher von virtuellen Tischen synchrone Arbeitsformen, indem zwei Personen Dokumente gemeinsam, zeitgleich auf einem Tisch bearbeiten, aber auch asynchrone Arbeitstechniken, indem ein Dokument auf einem Tisch belassen wird und in Abwesenheit eines Kooperationspartners bearbeitet werden kann. Ähnlich findet eine Integration verschiedener Kommunikationsmechanismen statt, indem neben einer Audio-Kommunikation innerhalb von Räumen und der direkten privaten Kommunikation zu einzelnen Teilnehmern (synchrone Kommunikation) ebenfalls die Möglichkeit des Hinterlassens von Mitteilungen an Kooperationspartner besteht (asynchrone Kommunikation).

DIVA ist sicherlich ein interessanter und von den umgesetzten Metaphern gut gestalteter Prototyp. Er demonstriert die Integration verschiedener Arbeitstechniken zusammen mit synchronen und asynchronen Kommunikationsmechanismen in einer Raummetapher. DIVA ist nicht WWW-basiert und wie die Autoren betonen als reiner Forschungsprototyp angelegt. Er ist damit nicht für die tägliche Praxis bestimmt.¹⁹

¹⁸ „We are also experimenting with it in project meetings, using it not only to keep track of agenda items and changes but also to log progress notes, action notes, etc. The review aid is of course highly useful here also“ [Engelbart & English 1968, Abs. 3i3].

¹⁹ Dies liegt zum großen Teil an seiner technischen Umsetzung in Common Lisp.

Aus dem Umfeld der spezifisch für den Bereich der kooperativen Lernunterstützung entwickelten Systeme gilt es insbesondere die Vital- und CROCODILE-Systeme zu berücksichtigen.

1.6.3 Kooperative Lehr-/Lernumgebungen

Die Umgebung VITAL (Virtual Teaching and Learning) [Pfister & Wessner 2000b] aus dem Projektbereich CLear [Pfister & Wessner 2000a] wurde basierend auf der an der Gesellschaft für Medien und Datenverarbeitung (GMD) entwickelten COAST-Technologie [Schuckmann et al. 1996] zur Erprobung von raumbasierten Lernszenarien entwickelt. Sie bilden die Basis einer Reihe von Projekten zum kooperativen Lernen in Gruppen. Durch eine konsequente Adaption der Metapher von virtuellen Räumen erhoffen sich die Entwickler insbesondere eine Verschmelzung von synchronen und asynchronen, kooperativen und individuellen Formen des Lernens. Natürlich wird neben diesen Eigenschaften die Möglichkeit des Lernens an verschiedenen Orten betont. VITAL ist eine proprietäre Umgebung, bietet verschiedene Awareness-Mechanismen und erlaubt den Umgang mit hypermedialen Dokumenten.

Ausgehend von der Metapher des virtuellen Raumes besteht die virtuelle Lernwelt aus einer Reihe von Räumen, die realweltliche Bezeichnungen tragen. Innerhalb der Lernumgebung nehmen die Akteure (Teilnehmer eines Lernprozesses) verschiedene Rollen, wie z.B. Lernender, Trainer, Tutor und Experte ein. Benutzerrechte und Zugriffsstrukturen sind an diese Rollen geknüpft.

Räume können in VITAL während des Lernprozesses erzeugt werden, sind persistent und werden in die Klassen „Privater Raum“, „Gruppen-Raum“ und „Auditorium“ unterschieden. Private Räume dienen der persönlichen Ablage und Organisation von Dokumenten und Lernnetzwerken, Gruppenräume dienen der Diskussion in der Lerngruppe und Auditorien stellen Orte des Umgangs mit Lehrmaterialien dar. Hier präsentieren und begleiten Tutoren oder Experten andere Lernende durch Materialien. Innerhalb des CROCODILE-Systems [Pfister et al. 1998] wurden Raumstrukturen nach der Metapher eines „Virtuellen Instituts“ [Miao & Haake 2001, S. 5ff] abgebildet. Auf das Beispiel des universitären Lernens bezogen existieren Räume wie der „campus“, das „administrational building“, eine „library“, der „public room“ und ein privater Raum, das „home“. Benutzer bewegen sich innerhalb der virtuellen Raumstruktur und sind für andere Lernende durch ein Piktogramm erkennbar. Hierbei folgt das Piktogramm den Bewegungen der Maus innerhalb eines Raumes – es wird eine Awareness-Information über die Position des Lernenden bereitgestellt.

VITAL ist als synchrones System mit einer relativ engen Kopplung auch von der Benutzerschnittstelle, also der Sicht auf einen Raum angelegt. Ähnlich sitzungsbasierten Shared Whiteboard- Systemen existieren z.B. Telepointer zur gegenseitigen Präsentation von Materialien oder zur Fokussierung von Aufmerksamkeit. Unter den computervermittelten Kommunikationsmechanismen unterstützt VITAL sowohl synchrone Kommunikationskanäle als auch asynchrone Kommunikationsformen. So stehen beispielsweise textueller Chat und Audio-Konferenzen und auch Kommunikationsbretter („message boards“) und E-Mail innerhalb von Räumen zur Verfügung.²⁰

Interaktionstechniken innerhalb der virtuellen Umgebung sind sehr stark durch Techniken des Drag&Drop geprägt. So kann beispielsweise ein Nutzer ein Dokument durch einfaches Ziehen mit der Maus auf eine „message box“ in einen anderen Raum

²⁰ Bewegen sich zwei Personen z.B. zu einem Kommunikationswerkzeug, werden ihre Piktogramme nebeneinandergestellt, eine Verbindung zwischen ihnen signalisiert eine private Kommunikation.

versenden. Ähnlich ist die Anbahnung eines kooperativen Editierens an Dokumenten gelöst. Hierzu wird ein Dokument mit der Maus auf ein Whiteboard-Objekt gezogen. Anschließend bewegen sich Benutzer, die das Dokument kooperativ editieren möchten, ebenfalls mit ihren Piktogrammen auf das Whiteboard. Des Weiteren besitzt jeder Raum eine Art Bibliothek, in welcher Hypermedia-Dokumente abgelegt werden können.

VITAL zeigt sich im Umgang mit Lehrmaterialien als hochgradig integrierte Umgebung. Text, Grafiken und Dokumente können in der Umgebung selbst individuell und kooperativ erzeugt und bearbeitet werden. Für sämtliche Medientypen stehen individuelle und kooperative Annotations-Mechanismen bereit. Als Teil dieser Funktionalität können verschiedene Dokumente über Verweise verknüpft werden.

Der Nachfolger von VITAL ist die CROCODILE-Umgebung (Creative Open Cooperative Distributed Learning Environment) [Pfister et al. 1998]. In ihr werden spezielle grafische Editoren integriert, welche auch Lernprotokolle [Wessner et al. 1999a] und Lernnetzwerke als Strukturierungsmechanismen eines Lernprozesses bereitstellen. Sie erlauben eine Art Übersichtsnetzwerk zu einem Wissensgebiet anzufertigen. Knoten des Netzwerkes sind mit entsprechenden erklärenden Hypermedia-Elementen verknüpft.²¹

Von der Einordnung in existierende Ansätze handelt es sich bei den VITAL- und CROCODILE-Lernumgebungen um geschlossene, netzgestützte Applikationen. Sie bieten insofern keine Verknüpfung und Verschmelzung mit Web-Technologie, die über das Einfügen von Verweisen auf Websites hinausgeht. Zielgruppe der Systeme sind kleinere bis mittlere Gruppen von Lernenden. Diesen stehen sowohl synchrone als auch asynchrone Kommunikationsmechanismen zur Verfügung, wobei auf die Entwicklung synchroner Arbeitsformen ein Schwerpunkt gelegt worden ist. Dies zeigt sich beispielsweise in Möglichkeiten der engen Kopplung von Oberflächenelementen und dem Einsatz von Merkmalen aus dem Bereich der Shared Whiteboard-Systeme wie Telepointer.

Eine Integration von Diensten und Werkzeugen in beiden Systemen ist als sehr weit fortgeschritten zu bezeichnen, dies geschieht jedoch auf Kosten einer Verwendung offener Standards und der Möglichkeit beliebige Werkzeuge in die Umgebung zu integrieren. So sind wie in vielen anderen Systemen Lehrmaterialien sehr eng an und in die Lernumgebung gebunden.

Eine enge Verwandtschaft weisen VITAL und CROCODILE zu einer ganzen Familie von kooperativen Lehr-/Lernumgebungen der Universität Toronto, Kanada, auf, welche jedoch eine weniger offene Gestaltung des Lehrprozesses bzw. freie Ausübung von Handlungen an Materialien zulassen (vgl. CSILE und das WebCSILE-System, [Burtis 1997], [Davie 1998]).

Anteile kooperativer Handlungsmöglichkeiten weist das CLARE-System [Wan 1993] auf. Es wurde speziell unter dem Gesichtspunkt der Strukturierung von wissenschaftlichem Arbeiten in Seminaren an der Universität Hawaii basierend auf der ENGRET-Technologie [Johnson 1992] entworfen. CLARE erlaubt es, kooperativ Netzwerke aus logischen Einheiten (Hypothese, Frage, Beobachtung, Folgerung, Argumentation etc.) aufzubauen. Hierzu stellt Wan eine spezielle Repräsentationstechnik RESRA vor, die eine Wiedergabe der thematischen Struktur von wissenschaftlichen Artikeln erlaubt. Lernende werden in mehreren Stufen vom individuellen Durcharbeiten und Zusammenfassen über die kooperative Bewertung bis hin zu einer thematischen Repräsen-

²¹ Lernnetzwerke sind vergleichbar mit dem Ansatz der semantischen Karten (vgl. [Hampel & Selke 1999] und [Klemme et al. 1998]).

tation eines wissenschaftlichen Artikels geführt. CLARE ist weniger von der Seite der technischen Architektur und Umsetzung her interessant – es handelt sich um ein geschlossenes Datenbanksystem mit recht rudimentären Clients – umso mehr besticht die Idee, kooperatives Lernen gezielt durch ein System zu unterstützen, was Lernenden in eine aktive Rolle verhilft. Wan beschreibt den Lernenden als eine mündige Person, die den Lernprozess kooperativ mit anderen Lernenden gestaltet. Auf das Arbeiten mit wissenschaftlichen Artikeln bezogen identifiziert er fünf grundlegende Ebenen kooperativen Arbeitens: „Summerization“ (Extrahieren, Zusammenfassen und Zusammenhänge herstellen), „Evaluation“ (Bewerten und Einschätzen), „Integration“ (In Beziehung Setzen von Sachverhalten), „Argumentation“ (Argumentation, Diskussion), „Construction“ (Erzeugen neuer Elemente, Formulierungen und Interpretationen eines Sachverhaltes). Natürlich lassen sich diese Tätigkeiten nicht klar voneinander abgrenzen, es erscheint jedoch unerlässlich, Systeme bereitzustellen, die diese grundlegenden Funktionen im Umgang und der Analyse von Texten unterstützen.

Insofern dokumentiert der CLARE-Ansatz einen der ersten Versuche, menschliches Lernen als Unterstützung und Bereitstellung verschiedener Funktionen in Handlungen an Medien zu begreifen und innerhalb eines Computersystems zu unterstützen.

Eine Bewertung des Ansatzes fällt wegen des eingestellten Projektstatus recht schwer. Wan untersucht sehr speziell den Umgang mit Texten und konzentriert sich in seinen Repräsentationen stark auf Formalisierungen des Prozesses menschlichen Lernens. Aus diesem Grunde kann nicht von einer offenen Lernumgebung gesprochen werden. – Der Ansatz demonstriert jedoch auf einen spezifischen Bereich bezogen eine Verschmelzung von individuellem und kooperativem Arbeiten bzw. Lernen und löst sich damit von der Vorstellung eines nur konsumierenden, passiven Lernenden.²²

Resümierend ist festzuhalten, dass sich aus dem Bereich der kooperativen Lehr- und Lernumgebungen nur wenige Ansätze finden, die eine unmittelbare Bewertung bzgl. ihrer Unterstützungsfunktion kooperativer Lernprozesse zulassen. Es handelt sich zumeist um recht spezielle, für einzelne Aspekte menschlichen Lernens zugeschnittene Systeme, die sich auf die Kopplung von Sichten auf einzelne Lehrmaterialien konzentrieren. Demgemäß fällt eine Bewertung aus Sicht der Reduzierung von Medienbrüchen schwer. Unsicherheit herrscht bei den untersuchten Ansätzen zudem in Fragen der Kopplungsgrade von Sichten auf Lehrmaterialien, in unterschiedlichen Vorgehensweisen der Vermittlung von gegenseitigen Wahrnehmungsinformationen oder Strukturierungsbemühungen des Lernprozesses selbst. Letzteres geschieht z.B. durch vorgegebene Wege durch die Lehrmaterialien oder feste Regeln der Interaktion innerhalb der Lernumgebung.

Die Frage nach dem Einsatz in sich geschlossener CSCL-Systeme stellt sich mit dem Hintergrund der Entwicklung des WWW nur begrenzt. Das Kriterium der durchgängigen Verfügbarkeit einer Lernumgebung und damit der alltagspraktischen Einbettung in den Lehrbetrieb verschiebt sich mehr und mehr in Richtung WWW-basierter Lösungen.

1.6.4 WWW-gestützte Ansätze

Unumstritten hat sich das WWW als die Basistechnologie zum Informationsaustausch im Internet entwickelt. Es ist das System, mit dem Menschen in unterschiedlichsten Anwendungskontexten netzgestützt arbeiten. Aufgrund seines hohen Verbreitungsgra-

²² Vgl. hierzu ebenfalls das Concept Indexing System von [Voss et al. 1999].

des und seiner breiten Verfügbarkeit stellt es damit zugleich die einzig sinnvolle Basis für ein kooperationsunterstützendes System dar. Proprietäre Lösungen, die spezifischer Zugangswerkzeuge und Infrastrukturen bedürfen, besitzen nur wenig Akzeptanz bei den Nutzern und sind in ihren Möglichkeiten durch fehlende Integration von Standards und Werkzeugen zumeist eingeschränkt.²³

Dem Anspruch, das WWW als Basistechnologie lernförderlicher Infrastrukturen einzusetzen, stehen jedoch erhebliche konzeptuelle Mängel gegenüber. Diese resultieren aus mangelnder architektonischer Fundierung des Anspruchs nutzerorientierter Fähigkeiten. Möglichkeiten der Bereitstellung und Manipulation von Materialien im WWW stehen nur der sehr kleinen Gruppe der Autoren von Netzseiten zur Verfügung. Ursprünglich als Publikationsmedium für Wissenschaftler entwickelt, fokussiert das WWW eine Autorensicht. Materialien werden von einem Autor für viele Lesende bereitgestellt. Lesende besitzen keinerlei Möglichkeiten der Manipulation von Netzseiten, die über das Anzeigen und Verfolgen von Verweisen hinausgehen.

Unumstritten sind die Probleme des WWW im Kontext der Unterstützung von Lern- und Lehrprozessen hinreichend untersucht (vgl. [Brennecke & Keil-Slawik 1995a] und [Brennecke & Keil-Slawik 1995b]). Das WWW stellt aus medienfunktionaler Sicht eine Einbahnstraße des Lernens dar: Materialien werden von Lehrenden für Lernende aufbereitet und abgelegt. Lernenden ist eine lesende und konsumierende, nicht aber eine aktive Rolle vorbehalten.

Damit scheint das allgemeine Defizit erkannt. Auch die Visionäre des WWW sehen die Zukunft des Web in der Unterstützung der Kooperation durch Konstruktion gemeinsamen Wissens (vgl. [Berners-Lee & Fischetti 1999]). Auch wenn die aktuell etablierte Struktur des WWW wenig kooperative Elemente in Bezug auf gemeinsame Wissenskonstruktion erkennen lässt, betont Berners-Lee seine Ursprungsidee einer interaktiven Vorstellung des Web, geprägt von der Metapher eines gemeinsamen Wissensraums, in Form eines kontinuierlichen gegenseitigen Gebens und Nehmens.²⁴ Als Vision eines WWW, in der sich Interaktivität nicht allein auf das Aktivieren von Verweisen zwischen Hypertextdokumenten bezieht, sondern auf wechselseitiges Bereitstellen von Informationen, wird die Begrifflichkeit interaktiv durch das Attribut „intercreativity“ ersetzt (vgl. [Berners-Lee 1999]). In der grundsätzlichen Idee des WWW, beliebige Wissensquellen zu verknüpfen, liegt sicherlich die Mächtigkeit und Qualität des Mediums und sein weltumspannender Erfolg.²⁵

Gleichzeitig liegt jedoch in seiner starren Struktur derartiger Verknüpfungen das primäre Problem und Ausschlusskriterium im Umfeld des kooperativen Lernens. Verknüpfungen zwischen Lehrmaterialien, Anmerkungen, Notizen und semantischen Strukturen müssen zu jedem Zeitpunkt die kooperativen Wissensstrukturen der Lernenden repräsentieren, dieser Anspruch ist mit Hilfe des WWW nur schwer umzusetzen.

²³ Aus diesen Gründen besitzt die in Kapitel 3 bis Kapitel 5 vorgestellte Konzeption und Umsetzung kooperativer Wissensräume eine enge technische Nähe zum WWW. Sie setzt auf existierenden und in Entwicklung befindlichen Technologien im Umfeld des WWW auf, erweitert diese jedoch um wesentliche Elemente kooperativer Medienfunktionen.

²⁴ „The basic idea of the Web is that an information space through which people can communicate, but communicate in a special way: communicate by sharing their knowledge in a pool. The idea was not just that it should be a big browsing medium. The idea was that everybody would be putting their ideas in, as well as taking them out“ [Berners-Lee 1999].

²⁵ „The fundamental thing about the space—about this Web, as I said, is that anything can refer to anything. Otherwise it's no fun. You've got to be able to make the link to anything“ [Berners-Lee 1999].

Einzig eine spezifische, von den Lernenden selbstadministrierte und in seiner Struktur offene Lernumgebung kann dies leisten. Zugleich ist festzustellen, dass eine Vielzahl von Forschern das aktuelle Defizit der fehlenden Organisation und Unstrukturiertheit des Web erkannt haben.

Unabhängig von dem Problem, Informationen in allen Lernkontexten zur richtigen Zeit am richtigen Ort bereitzustellen (vgl. [Gross & Prinz 2000, S. 125]), weist das WWW zudem keinerlei Mechanismen der gegenseitigen Wahrnehmung (Awareness) auf und bietet in dieser Form wenig Ansatzpunkte zu einer Koordination gemeinsamen Lernens. Schon im Jahre 1997 zum Zeitpunkt der Entwicklung erster WWW-basierter, synchroner kooperationsunterstützender Systeme machen Beca et al. deutlich, dass ein Verständnis von Interaktivität im WWW sich derzeit lediglich auf die Generierung von Netzseiten in Abhängigkeit von Eingaben der Nutzer bezieht (vgl. [Beca et al. 1997]). Die Generierung von dynamischen HTML-Seiten ist insofern das Maximum der Interaktivität, was das Web zurzeit erlaubt. Seit den Anfängen des WWW hat sich in der konzeptionellen Fundierung des Problems nur wenig geändert.

Technisch betrachtet liegt eine erste Begründung für die Autorenzentrierung des WWW in der Zustands- und Verbindungslosigkeit des HTTP-Protokolls begründet. Jeder Zugriff eines Nutzers auf das WWW ist unabhängig von vorangegangenen Aktionen und Zuständen aufseiten des Nutzers und des Servers. Zugriffe besitzen keine definierte Reihenfolge und erlauben daher keine komplexen Interaktionsformen zwischen einzelnen Nutzern.

Resümierend ist festzuhalten, dass sich das WWW in seiner jetzigen Form nur bedingt für die kooperative Wissenskonstruktion im Rahmen von Lernprozessen eignet. Die gängige Praxis der Bereitstellung von Lehrmaterialien in Form von WWW-Seiten sieht die Lernenden in einer passiven Rolle des Konsumierens von Medien.

Eine weiterführende Untersuchung auftretender Medienbrüche im Einsatz des WWW in der Lehre erscheint aufgrund der oben genannten konzeptionellen und technischen Defizite wenig sinnvoll. Handlungen an netzgestützten Materialien wie das Erzeugen, Arrangieren und Verknüpfen finden sich lediglich für die Autoren eines Wissensbereiches. Ähnlich hypermedialer Autorenumgebungen sind in dieser Form aus medientechnischer Perspektive die Möglichkeiten für die Nutzer (Lernenden) eingeschränkt. Beispielsweise vermag ein Nutzer typischer WWW-Seiten und Materialien nicht, eigene Informationen hinzuzufügen oder bestehende Elemente in einen persönlichen Kontext zu setzen. In Berücksichtigung der technischen Ansätze zu einer Kooperationsunterstützung auf Basis des Internet genügt das WWW in seiner heutigen Form den Anforderungen der kooperativen Wissensorganisation nicht. Hier erscheint der Bereich der ereignisorientierten Systeme (MUDs und MOOs) als ebenfalls internetbasierte Kooperationstechnologie um einiges viel versprechender.

1.7 Entwicklungslinie: Virtuelle Gemeinschaften

Neben der Entwicklungslinie der bürowirtschaftlich orientierten CSCL- und CSCW-Systeme lässt sich im Umfeld privat motivierter, virtueller Gemeinschaften eine zweite Entwicklungslinie der MUDs und MOOs ausmachen.

Aus konzeptueller Sicht ist der Begriff der virtuellen Gemeinschaft eng mit der Metapher des virtuellen Raums verbunden, entsprechend ergibt sich eine erste Motivation

für die Berücksichtigung in der vorliegenden Arbeit. Zum Zweiten liegt im Bereich der MUDs und MOOs ein konzeptuell und architektonisch viel versprechender Ansatz.²⁶

Eng verbunden mit dem Begriff der virtuellen Gemeinschaft, der Virtual Community, den Rheingold Anfang der 90er Jahre durch seine Erfahrungen mit dem Mailboxsystem „The WELL“ prägte (vgl. [Rheingold 1993]), ist die Idee der virtuellen Lerngemeinschaft. Bis heute fällt es schwer die These zu belegen, dass virtuelle Gemeinschaften eine neue Form von Sozialität herausbilden (vgl. [Becker 2001, S. 193]). Schon Rheingold berichtete, dass die sich innerhalb der Netzkommunikation ausbildenden sozialen Beziehungen ihre Entsprechungen in realweltlichen sozialen Gemeinschaftsformen fanden, indem sich primär Mitglieder in der virtuellen Gemeinschaft etablierten, die nicht mehr als eine Autostunde von der San Francisco Bay Area entfernt wohnten.

Damit erscheint klar, dass auch virtuelle Lerngemeinschaften kein Substitut für realweltliche Gemeinschaften darstellen. Kooperationsunterstützende Systeme können lediglich einen infrastrukturellen Kern bereitstellen, um den sich ein Gemeinwesen entwickeln kann (vgl. [Keil-Slawik & Hampel 2001, S. 246]).

In analoger Weise wird durch die vorliegende Arbeit die These verfolgt, dass kooperative virtuelle Wissensräume die fruchtbare Basis eines kooperativen Lernprozesses und Dialogs zwischen Lehrenden und Lernenden auszubilden vermögen. Gleichzeitig werden jedoch ihre Möglichkeiten und Einschränkungen konsequent aus einer medientechnischen Perspektive hinterfragt und bewertet.

Konstruktivistische Lerntheorien beschreiben das menschliche Lernen als Prozess der schrittweisen Konstruktion und des In-Beziehung-Setzens von Erkenntnissen, auf der Basis individueller Wahrnehmung und Verarbeitung von Erfahrungen und Erlebnissen. Darüber hinaus ist Lernen nicht nur als rein individueller Prozess zu verstehen, obwohl herausragende Leistungen oder Forschungsergebnisse meist an Einzelpersonen festgemacht werden - eine Tatsache, die scheinbar dagegen spricht.

Aber der Lernprozess jedes einzelnen kann maßgeblich durch gemeinsames Lernen im Team, d.h. durch Kommunikation mit anderen, intensiviert und gefördert werden. Lernen wird deshalb im Sinne dieser Arbeit nicht nur als ein individueller, sondern eher als ein kooperativer Prozess angesehen. Er ist geprägt durch gemeinschaftliches Explorieren von Lehrmaterialien, aber vor allem durch Diskussionen, gegenseitiges Befragen und Beraten sowie den Austausch von Materialien.

Menschen lernen nicht, wie früher vermutet, nach dem Prinzip des Nürnberger Trichters, indem sich Informationen in die Köpfe der Lernenden schlicht übertragen lassen, sie verlangen eine soziale Umwelt und situiertes Handeln. Lernen findet also nicht in den Köpfen der Lernenden, sondern mit den Köpfen der Lernenden statt und der Plural ist es, auf den in diesem Zusammenhang Wert gelegt werden muss.

Konstruktivistisches Lernen definiert zwar eine gewisse Eigenverantwortung der Lernenden, indem sie in gewissem Maße selbst für ihren Lernfortgang verantwortlich sind (es kann eigenständig über die Materialien und die einzelnen Lernschritte verfügt werden), dies bedeutet aber nicht, dass einzeln Lernende für sich allein, ohne Kommunikation und kooperativen Wissensaustausch lernen können.

²⁶ Nur am Rande berücksichtigt wird der Bereich der so genannten Avatar-Umgebungen bzw. Collaborative Virtual Environments (CVEs) (vgl. [Churchill et al. 2001]). Trotz seiner Nähe zu den MUDs und MOOs eignet er sich aufgrund mangelnder Fähigkeiten zum kooperativen Umgang mit Medien nur sehr eingeschränkt im Umfeld kooperativer Wissensräume.

Situiertes Lernen bedeutet, Lernen in einer alltäglichen Umwelt, in einer Gemeinschaft einzubetten. Wenger prägte für diese Gemeinschaft unter Lernenden, die sich durch alle Strukturen unserer Gesellschaft zieht, den Begriff der „Community of Practice“ [Wenger 1998]. Jede Lerngemeinschaft wird als soziales System verstanden. Eine „Community of Practice“ ist keine funktionale oder organisatorische Einheit, sondern es ist ein informeller Zusammenschluss von Lernenden, von Gleichgesinnten. Insofern erscheint gerade dieser informelle Weg, Wissen auszutauschen und zu organisieren, das vorrangige Ziel der Bereitstellung von kooperationsunterstützenden Werkzeugen.²⁷

In einem derartigen Verständnis liegt Erlerntes nicht in den Köpfen der Individuen allein, das Gelernte verteilt sich vielmehr über eine gesamte Gruppe von Lernenden. So wird die Gruppe als Ganzes zum Träger von Wissen. Auch wenn die Terminologie von „shared knowledge“ in seiner deutschen Übersetzung als *verteilt* oder *geteiltes* Wissen missverständlich klingt, hat sich doch dieser Terminus für den gemeinsamen Aufbau von Wissensstrukturen etabliert. Im Folgenden wird von kooperativen oder *gemeinsamen* Wissensräumen gesprochen. Denn die Seite der Unterstützungsfunktionen eines Lernprozesses verlangt nach einem kooperativen, gemeinsamen Gedächtnis.²⁸

Analog zur realen Gemeinschaft wird in dieser Weise auch die virtuelle Gemeinschaft oder speziell die virtuelle Lerngemeinschaft durch das soziale Phänomen des Bereitstellens von bedeutungsvollen Verbindungen zwischen Lernenden verstanden (vgl. [Mynatt et al. 1997, S. 210]). Dementsprechend definiert sich die virtuelle Lerngemeinschaft als eine Mischung aus unterschiedlichsten Faktoren; diese reichen von realen, räumlichen Beziehungen der Mitglieder der virtuellen Gemeinschaft bis hin zu sozialen Interaktionen.²⁹

Als ein Zwischenresümee ist damit festzuhalten, dass der Anspruch, virtuelle Lerngemeinschaften alleinig durch infrastrukturelle Maßnahmen aufbauen zu können, nicht der Realität entspricht. Technik kann technische Probleme der Unterstützung von Lernprozessen lösen, didaktische, organisatorische oder soziologische Prozesse nicht. Damit liegt die primäre Aufgabe einer kooperativen Lernumgebung darin begründet, exakt die technologischen Probleme und Funktionen kooperativen Lernens mit Materialien bereitzustellen.

Johnson stellt in seinen Aufzeichnungen zum kooperativen Arbeiten fest, dass Gruppenarbeit einen kaum vorhersagbaren Charakter besitzt [Johnson 1992]. Es erscheint schwierig, wenn nicht gar unmöglich, die Aktivitäten einer Gruppe im Vorfeld zu umreißen, festzulegen, welche Wege eine Gruppenarbeit einschlägt und wie der Prozess der Gruppenarbeit im Detail vonstatten geht. Es lässt sich in dieser Form keine Aussage treffen, wie und wann eine Gruppe Entscheidungen trifft und wie sie ihre Aufgaben löst. Johnson charakterisiert dieses Phänomen und diese Eigenschaft von Gruppenarbeit treffend mit dem Attribut „exploratory“ [ebd., S. 298]. Diese dynamische, unvorhersagbare und freie Struktur von Gruppen lässt sich auf Lerngruppen und insbeson-

²⁷ „Communities of Practice preserve the tacit aspects of knowledge that formal systems cannot capture“ [Wenger 1998].

²⁸ Einige Autoren wie beispielsweise Wan bezeichnen dieses Gedächtnis als „long-term memory“, welches sowohl den Inhalt von Aktivitäten als auch von Prozessen (Lernfortschritten) über eine längere Periode des gemeinsamen Arbeitens erhält [vgl. Wan 1993].

²⁹ Mynatt et al. charakterisieren den Begriff der Community mit sehr viel Sorgfalt: „[...] we would like to point to the loose consensus around community as referring to a multi-dimensional, cohesive social grouping that includes in varying degrees: shared spatial relations, social conventions, a sense of membership and boundaries, and an ongoing rhythm of social interaction“ [Mynatt et al. 1997, S. 211].

dere auf virtuelle Lerngruppen übertragen. Hieraus resultierend erscheint der Anspruch zum Scheitern verurteilt, Gruppendynamik und Problemlöseprozesse vollständig verstehen zu können und deshalb Systeme und Mechanismen entwerfen zu können, die den Prozess des kooperativen Lernens durch feste Strukturen und Vorgehensweisen unterstützen. Kooperatives Lernen in feste Regeln und Modelle gießen zu wollen, ist aus diesem Grunde derselbe Irrweg wie den Anspruch erheben zu wollen, individuelles Lernen vollständig begreifen und strukturieren zu können.

Damit wird deutlich, dass Wissenskommunikation und Wissensorganisation in virtuellen Gemeinschaften in dieser Form nur wenig ausgeprägte Strukturen besitzt [Mambrey et al. 2000]. Als ein dynamischer und verteilt ablaufender Prozess sind nur in geringem Maße feste Strukturen und Zielsetzungen auszumachen. Damit sind Konstrukteure von kooperationsunterstützenden Umgebungen nicht von der Last befreit, möglichst universelle Strukturierungsmöglichkeiten einer Lernumgebung durch die Nutzer selbst bereitzustellen.

Die im Rahmen dieser Arbeit entwickelte Architektur mit der ihr zugrunde liegenden, medientechnischen Perspektive erweitert die Funktion eines gemeinsamen Gedächtnisses einer Lerngemeinschaft bewusst um Fähigkeiten der Selbstorganisation und Anpassbarkeit der Lernumgebung durch die Lernenden. In dem formulierten Anspruch der Anpassbarkeit einer Lernumgebung an die organisatorischen, sozialen und thematischen Rahmenbedingungen eines Lernprozesses in Selbstorganisation durch die Lernenden liegt der Schlüssel zur Lösung des von Johnson geschilderten Dilemmas. Eine kooperationsunterstützende Umgebung kann weder die Struktur von Lernprozessen noch ihre organisatorischen Rahmenbedingungen universell definieren, sie kann und darf lediglich möglichst universelle Funktionen im Umgang mit Medien bereitstellen.

1.7.1 MUDs und MOOs

Parallel zu der Entwicklung kooperationsunterstützender Systeme im Umfeld der CSCW- und CSCL-Forschung verläuft der Entwicklungsstrang zumeist privat motivierter raum- und internetbasierter Kommunikations- und Erlebniswelten. Eines der sicherlich interessantesten Kapitel in der Entwicklung von computergestützten Kooperationsformen stellt damit das Feld der MUDs und MOOs dar. Wenn auch in weiten Kreisen wenig bekannt, entwickelten sich diese ursprünglich als netzgestützte Abenteuer- und Phantasiewelten konzipierten, virtuellen Umgebungen hin zu eigenständigen Interaktions- und Kommunikationsformen. Gleichzeitig besitzen MUDs und speziell die objektorientierte Variante der MOOs ein sehr eigenständiges, architektonisches Design, welches Teilen der in Kapitel 5 vorgestellten Architektur als Vorbild dient.

Als Trubshaw und Bartle das erste MUD an der Universität Essex, England, (in Zukunft als Essex-MUD bezeichnet) entwickelten, stand zunächst der Spielcharakter des MUD im Vordergrund (vgl. [Bartle 1983], [Bartle 1985]). Doch schon zu diesem Zeitpunkt, noch bevor sich externe Nutzer über Modem oder Internet mit dem Spiel verbinden konnten, verneint Bartle den Charakter des MUD als den eines reinen Spieles (vgl. [Bartle 1990a]). Er betont vielmehr, dass das MUD im Umfeld der Entwicklung von Definitionssprachen für Datenbanken entstanden sei und als ein erstes Testfeld für diese fungieren sollte und nicht als reine Umsetzung des Zork-Abenteuerspiels [Lebling et al. 1979] zu sehen ist. Schon 1980 wurde der Nutzerkreis, der zunächst aus Studierenden der Universität bestand, auf externe, über Modemleitungen verbundene Spieler ausgeweitet. Hierzu diente eine erste versuchsweise Anbindung der Essex-University an das amerikanische ArpaNet, dem Vorläufer des Internet.

Interessanterweise sind damit MUDs neben E-Mail oder FTP als die ersten Dienste oder Anwender des Internet zu sehen. Sie gehören ferner zu den Pionieren neuer Interaktionsformen zwischen Menschen über das neue Medium Internet und lassen sich durch ihre Vielfalt an verschiedenen Anwendungen nur schwer eindeutig charakterisieren oder in ihren Eigenschaften vollständig beschreiben.

In den Jahren seit der Entwicklung erster Formen des MUD sind eine Vielzahl von unterschiedlichen MUDs und spezieller Untervarianten entstanden. Hierzu zählen z.B. AberMUD, LPMUD, MOO, MUSH, TinyMUD und DikuMUD. Neben der historischen Entwicklung verschiedener technischer Konzeptionen und Umsetzungen des ursprünglichen Essex-MUDs besitzen die einzelnen Formen verschiedene, thematische Ausrichtungen: AberMUD, benannt nach der University Aberystwyth in England, wurde schon Anfang der 80er Jahre von Cox, einem Spieler des ersten Essex-MUD, entwickelt. Es kann als das erste erfolgreiche MUD mit einer größeren Anzahl von Spielern bezeichnet werden. LPMUD³⁰ wurde von Pensjö Mitte der 80er Jahre ins Leben gerufen und zeichnet sich vor allem durch die leistungsfähige interne Programmiersprache LPC aus, die mit relativ geringem Aufwand erlaubt, neue Objekte der virtuellen Welt hinzuzufügen. In der Implementierung von LPC lag damit der Schlüssel für eine Vielzahl von speziellen MUD-Umgebungen. Anwendungsgebiete des LPMUD sind meist reine Rollenspiele, d.h. Spieler bewegen sich und interagieren in einer in Textform beschriebenen, virtuellen Welt und kommunizieren über Chat.

Ähnlich kann das Ende der 80er Jahre entwickelte MOO (vgl. [Curtis & Nichols 1994] und [Curtis 1997]) als konsequent objektorientierte Implementierung des MUD charakterisiert werden. Im Gegensatz zum MUD werden MOOs primär als virtueller Treffpunkt von Nutzern im Netz verstanden. Ihre Architektur ist datenbankorientiert und basiert auf interagierenden Objekten, welche über spezielle Ereignisse kommunizieren.

Curtis sieht schon früh in der Entwicklung des MOO ein Potenzial in der Unterstützung der Interaktion zwischen Wissenschaftlern, beispielsweise zur gegenseitigen Präsentation von Folien und Illustrationen.³¹ Leider hat sich seine Prognose, dass derartige Interaktionsformen zum Standard wissenschaftlicher Kommunikation gehören, auch nach fast zehn Jahren nicht bewahrheitet.

Weitere Implementierungen des MUD wie das von Aspnes 1989 entwickelte TinyMUD oder MUSH (Multi-User Shared Hallucination) entstanden ebenfalls Ende der 80er Jahre und sind in den meisten Fällen der Klasse der Rollenspiele zuzuordnen. Spätere Implementierungen wie DikuMUD³² wurden Anfang der 90er Jahre entwickelt und besitzen eine interne Programmiersprache und spezielle Mechanismen, um die virtuelle Welt möglichst einfach abbilden zu können.

Neben den technischen Eigenarten von MUDs sind eine ganze Reihe von konzeptionellen und metaphorischen Eigenschaften zu nennen, die zum breiten Erfolg der Idee des MUD beigetragen haben. Hierzu zählen die Versinnbildlichung der Spieler bzw. Nutzer des MUD in Form von Avataren oder beispielsweise die konsequente Anwendung einer Raummetapher (vgl. [Henderson & Card 1985]).

³⁰ Vgl. <http://www.neosoft.com/genesis/>, Stand 1.12.2001.

³¹ „Our plans include allowing scientists to give on-line presentations to their colleagues around the world, complete with ‘slides’ and illustrations automatically displayed on the participants’ workstations. The same approach could be used to create on-line meeting places for workers in other disciplines, as well as for other non-scientific communities. [...] I expect such specialized virtual realities to be commonplace, an accepted part of at least the academic community.“ [Curtis 1992, S. 17]

³² Vgl. <http://dikumud.com/>, Stand 1.12.2001.

Eine strikte Trennung dieser technischen und konzeptionell-sozialen Faktoren gelingt nur schwer. Im Folgenden werden verschiedene Eigenschaften der Idee des MUD anhand einer Auswahl von Definitionen und Beschreibungen aufgeführt.

So betont Curtis in seiner Definition (vgl. [Curtis 1992, S. 1ff.]) zunächst die technischen Grundeigenschaften eines MUD als Softwaresystem, welches Netzwerkverbindungen zwischen verschiedenen Nutzern herstellt und einen Zugang auf eine gemeinsame Datenbank bietet, legt aber gleichzeitig die Betonung auf einen „place“, einen gemeinsamen, elektronisch repräsentierten Ort. Benutzer bewegen sich also innerhalb einer virtuellen Welt, sie „sehen“ und interagieren mit Objekten, die innerhalb von virtuellen Räumen platziert sind. Räume sind über Verbindungen (exits) miteinander verbunden, diese ermöglichen die Navigation zwischen Räumen. Neben einer ungezwungenen synchronen Kommunikation unter den Nutzern sind es diese Interaktionen zwischen Personen und Objekten und Personen untereinander, die eine wirklich neue Qualität des MUD gegenüber klassischen digitalen Medien wie einem herkömmlichen Chat ausmachen.

Derartige Interaktionsformen bezieht Carlstrom zwar primär auf den Anwendungsfall des Rollenspiels, identifiziert sie jedoch als zentrale Eigenschaft eines MUD.³³ Ähnliche Eigenschaften hatte auch schon Bartle in seiner Definition des MUD betont, für ihn ist das MUD ein System der Interaktion zwischen Spielern und Gegenständen (Objekten) sowie Spielern untereinander (vgl. [Bartle 1990b, S. 5ff.]).

Parallel benennt Curtis drei wichtige Kriterien eines MUD. Er betont nochmals, dass es sich bei der Grundidee des MUD weniger um ein Spiel, sondern vielmehr um eine neuartige netzgestützte Interaktionsform zwischen einer Anzahl von Nutzern handelt. Nutzer halten sich zu ein und derselben Zeit innerhalb der virtuellen Welt auf und kommunizieren über synchrone Mechanismen. Sie können die virtuelle Welt aktiv beeinflussen und verändern, also z.B. eigenständig neue Objekte hinzufügen, und nehmen in dieser Form eine aktive Rolle ein.³⁴

Rheingold, der Begründer des Begriffs der Virtual Community [Rheingold 1993], führt den Gedanken eines gemeinsamen Treffpunkts von Nutzern im Netz weiter fort. Rheingold und Kelly untersuchen in ihrem Artikel „The Dragon Ate My Homework“ [Kelly & Rheingold 1993] ‚ernsthafte‘ Anwendungen eines MUD und existierende virtuelle Gemeinschaften. Sie geben einen kurzen Überblick über verschiedene MUDs und MOOs und betonen ebenfalls den gemeinsamen Treffpunkt, den „place“, als die Voraussetzung jeglicher virtueller Gemeinschaften.³⁵ Spätere Autoren beziehen sich in ihrem Verständnis eines MUD meist auf die ursprüngliche Definition von [Curtis 1992], berücksichtigen aber speziell den Einfluss des Internet. Des Weiteren rückt mit der Ent-

³³ „(MUDs) are environments which one can log into from a terminal connected to Internet, and then interact in text with objects, places, and other players [...]“ [Carlstrom 1992].

³⁴ „A MUD is not goal-oriented; it has no beginning or end, no 'score', and no notion of 'winning' or 'success'. In short, even though users of MUDs are commonly called players, a MUD isn't really a game at all. [...] A MUD generally has more than one user connected at a time. All of the connected users are browsing and manipulating the same database and can encounter the new objects created by others. The multiple users on a MUD can communicate with each other in real time“ [Curtis 1992, S. 2].

³⁵ „Well, if 100 other students were to show up in the same virtual 'place', you could have a party, devise pranks, do some role-playing, scheme, even build a better world. All the same time. The only thing you'd need is a place to meet“ [Kelly & Rheingold 1993, S. 68].

stehung von integrierten Programmiersprachen innerhalb der MUDs und MOOs die Erweiterbarkeit durch die Nutzer selbst in den Vordergrund.³⁶

Vielleicht sind es die starren Strukturen des WWW, die die Idee der aktiven Einbeziehung der Nutzer in die Gestaltung einer virtuellen Umgebung mehr und mehr akzentuieren. MUDs kennen zwar eine Hierarchie aus normalen Nutzern und so genannten Wizards, welche nicht am Spielgeschehen teilnehmen und sich um die Erweiterung der virtuellen Welt kümmern, ambitionierte Nutzer entwickeln sich jedoch meist hin zum Wizard-Status und helfen das MUD oder MOO kontinuierlich zu erweitern und zu verbessern. Demgemäß sehen einige Autoren, wie z.B. O'Day et al., das Potenzial eines MUD in seiner Fähigkeit die Grenze zwischen Benutzern und Entwicklern zu verwischen.³⁷

Andere Autoren wie Cooper gehen basierend auf der Idee des kooperativen Schaffens einer virtuellen Gemeinschaft und Umgebung so weit, MUDs als eine neue Form von Kunst, und zwar als eine kooperativ geschaffene Kunstform zu bezeichnen: „communally constructed works of art“, vgl. [Cooper 2000].

Recht aktuelle Arbeiten aus dem Bereich der MUDs und MOOs, wie sie Mynatt et al. liefern, lösen sich von der Idee des MUD als Rollenspielumgebung und betonen die Kerneigenschaften einer persistenten virtuellen Welt, in welcher themenorientiert Interaktionen zwischen Nutzern stattfinden, die auch aus dem Aus- und Weiterbildungsbe- reich stammen können (vgl. [Mynatt et al. 1997, S. 210]). Das MUD wird salonfähig und wird von einer ganzen Anzahl von Autoren als ernsthafte Anwendung zur Kooperationsunterstützung charakterisiert.

Zusammenfassend lassen sich die wesentlichen Eigenschaften der Idee des MUD in technische und konzeptionell-soziale Faktoren differenzieren, beide beeinflussen sich gegenseitig in hohem Maße. Vielleicht ist es insbesondere diese Mischung aus Architektur und gewählter Metapher, die den Erfolg der MUDs und MOOs im Anwendungsbereich der Unterstützung von Kommunikations- und Interaktionsprozessen ausmacht.

Auf der technischen Seite zeichnet sich die Architektur eines MUD zunächst durch einen einfachen Mechanismus des Aufbaus von synchronen Verbindungen zwischen verschiedenen Nutzern aus. Ein MUD-Server kommuniziert über das Internet mit verschiedenen Clients, bei denen es sich im einfachsten Fall um eine einfache Terminal-(telnet-) Verbindung handelt. Interaktionen zwischen den Nutzern und der Umgebung werden über wenige Textkommandos realisiert. Es existieren zudem eine ganze Anzahl von grafischen MUD-Clients. Der MUD-Server stellt eine raumbasierte virtuelle Welt bereit, die grundsätzliche Persistenzeigenschaften besitzt. In einigen MUDs werden regelmäßig Objekte in ihren Ursprungszustand zurückversetzt. – Eine Eigenschaft, die aus der Konzeption des umgesetzten Rollenspiels resultiert.

Die eigentliche MUD-Umgebung (virtuelle Welt) zeichnet sich in den meisten Fällen durch seine leistungsfähige benutzerseitige Erweiterbarkeit aus. Benutzer, die einen entsprechenden Status innerhalb eines MUD besitzen, können ihre Umgebung einfach und unkompliziert erweitern.

Auf der metaphorisch konzeptionellen Seite besticht die MUD-Idee zunächst durch ihre konsequente Konzentration auf der Bereitstellung einer raumbasierten Umgebung, einzelne Räume, welche durch Türen („exits“) miteinander verbunden sind: Gleichzeitig

³⁶ „MUDs are networked, multi-participant, user-extensible systems which are most commonly found on the Internet [...]“ [Reid 1994, S. 2].

³⁷ „[...] the boundaries between users and designers are blurred. Users can change the landscape, program new tools and offer new services“ [O'Day et al. 1996, S. 161].

bilden Avatare die Grundlage der Interaktion zwischen den Nutzern. Die Räume eines MUD strukturieren in natürlicher Weise die Kommunikation innerhalb der virtuellen Umgebung, Nutzer „hören“ Personen, die sich in demselben Raum aufhalten, und bieten zugleich einen persistenten Ort der Ablage von Objekten.

1.7.2 Klassifizierungsansatz des Nutzungsumfelds

Ein vollständiger Überblick zu den Nutzungsfeldern von MUDs und MOOs würde den Umfang dieser Arbeit übersteigen. Schon Anfang der 90er Jahre ergibt der von Bartle angefertigte „MUD Report“ (vgl. [Bartle 1990b, S. 71ff.]) eine erstaunlich große Zahl verschiedener MUDs.³⁸

Im Folgenden seien die wesentlichen Entwicklungsschritte und einige Anwendungsbeispiele speziell aus dem Bereich der Nutzung von MUDs und MOOs zur Kooperationsunterstützung und zum kooperativen Lernen gekennzeichnet.

MUDs und MOOs finden sich in einer ganzen Anzahl von Anwendungsfeldern. Frühe Beispiele sind sicherlich der Versuch, ein MUD zur Koordination von Systemadministratoren einzusetzen [Evard 1993] oder die Bemühungen am Xerox PARC, MOO-Technologie für die Arbeit von Medienwissenschaftlern [Bruckman & Resnick 1995] zu nutzen.³⁹ Ein richtungsweisender Ansatz verknüpfte ein MOO und das Gopher System miteinander (vgl. [Masinter & Ostrom 1993]). Hierdurch ergab sich eine Synthese des hypertextbasierten Gopher als Vorläufer des WWW und den raumbasierten Kommunikationsfähigkeiten des MOO. Die im Rahmen dieser Arbeit vorgestellte Konzeption greift in seiner Architektur diese grundsätzliche Idee auf.

Unter dem Blickwinkel der Nutzung von virtuellen Umgebungen zur Unterstützung von kooperativen Lernsituationen existieren eine ganze Anzahl relevanter Ansätze:

Zunächst liefert Fanderclai unter dem Fokus menschlichen Lernens treffende Definitionen des MUD. Für sie ist das MUD ein Platz des selbstorganisierten Lernens. Es besitzt in wenigen Fällen den Charakter von Unorganisiertheit, ist zugleich aber unheimlich effektiv.⁴⁰

Aus einer Analyse der existierenden Ansätze ergeben sich in der grundsätzlichen Art der Nutzung drei Leitlinien als eine grobe Klassifikation entlang den Dimensionen Unterstützung von Kommunikation, Vermittlung von didaktisch aufbereitetem Wissen und Bereitstellung einer virtuellen Lern- und Arbeitsumgebung.

Gruppe I: MUDs und MOOs als Kooperationsplattform

Hier wird die virtuelle Umgebung nicht zum direkten Transport von Lehrinhalten genutzt. Repräsentiert durch Avatare treffen sich Lernende und Lehrende an festen Orten in der raumbasierten Struktur des MUD um themenorientiert zu kommunizieren. Betont wird in diesem Zusammenhang insbesondere die vergleichsweise zwanglose Kommunikation, an der Studierende teilnehmen. Diese Kommunikation muss nicht zwingend auf einen textorientierten Chat beschränkt sein, in einer ganzen Reihe von Fällen werden Audio- und Videokommunikationsformen integriert, das MUD oder MOO wird zum so genannten Media Space.

³⁸ Vgl. hierzu auch [Curtis & Nichols 1994].

³⁹ Ein aktuell interessantes Beispiel bietet die elektronische Stadt MOOsburg, welche ein MUD als Bürgerinformationssystem nutzt (vgl. [Caroll et al. 2000]).

⁴⁰ „MUDs are places for self-directed learning, learning that blends work and play, that often looks chaotic but that is uniquely effective“ [Fanderclai 1995, S. 8].

Auf der organisatorisch-technischen Seite lassen sich MUDs und MOOs zudem problemlos in eine existierende Lernsituation einflechten. Sie benötigen neben einer Internetverbindung keine speziellen Zugangswerkzeuge und sind einfach zu bedienen. MUDs und MOOs sind durch ihre geringe notwendige Bandbreite der Datenübertragung auch über Modemverbindungen nutzbar und aus diesem Grunde auch ohne leistungsfähige Netzanbindung an allen Lernorten einsetzbar. Als technische Plattform wird an den meisten der untersuchten Einsatzorten das LambdaMOO-System genutzt. Es besitzt gute Persistenzeigenschaften und lässt sich durch seine Datenbankarchitektur mit geringem Aufwand administrieren.

Gruppe II: MUDs und MOOs zur didaktisch aufbereiteten Vermittlung von Lehrmaterialien

Die zweite relativ kleine Gruppe der untersuchten Einsatzmöglichkeiten setzen MUDs und MOOs didaktisch, zur direkten Vermittlung von Lehrinhalten ein. Hierzu werden innerhalb des MUD Kursmaterialien abgelegt oder in Anlehnung an ihre Wurzeln des Rollenspiels Wissen in Form von zu lösenden Aufgaben vermittelt. Ähnlich dem Abenteuerspiel erhalten Lernende für die Lösung von Teilaufgaben einen Anreiz in Form von Erfahrungspunkten oder sozialem Status des Avatars. Gestellte Aufgaben besitzen zu meist den Charakter nicht von einem Lernenden allein gelöst werden zu können, es wird die Kooperation zwischen den Lernenden untereinander und den Lehrenden forciert. Eine Reihe von Umgebungen besitzen spezielle didaktisch aufbereitete Werkzeuge und Anschauungsobjekte, um Lerninhalte anschaulich zu vermitteln bzw. eine Interaktion der Lernenden mit diesen zu ermöglichen.

Gruppe III: MUDs und MOOs als Plattform virtueller Lern- und Arbeitsumgebungen

Die dritte Gruppe der Anwender des MUD und MOO in der Lehre sehen diese schließlich als virtuelle Lern- und Arbeitsumgebung, in der aktiv mit Materialien gearbeitet werden soll. Ähnlich der im Rahmen dieser Arbeit verfolgten Konzeption werden sie als technische Plattformen zum Umgang mit Materialien verstanden und sind Ort der Ablage für gemeinsame Materialien. Leider finden sich nur wenige bis gar keine Beispiele MUDs und MOOs unter diesem Gesichtspunkt in die Lehre erfolgreich einzubinden.

In den folgenden Abschnitten wird ein kurzer Überblick über existierende Ansätze der Nutzung von MUD- und MOO-Technologie in der Lehre und in der Wissenschaft gegeben. Wie schon im Bereich des CSCW und CSCL angemerkt, kann ein vollständiger Überblick aufgrund kaum vorhandener Systematisierungen und nur wenig stichhaltiger Evaluationen nur schwer gelingen. Gleichzeitig zeigt sich, dass weitere Anwendungen der MUD-Technologie nur wenig medialen Mehrwert gegenüber den genannten Grundklassen entwickeln. MUDs zeigen ihre Qualität weniger in der Arbeit mit textuellen oder multimedialen Ausdrucksmitteln, vielmehr in der Möglichkeit zur Selbstorganisation der virtuellen Umgebung. Derartige Eigenschaften prägen die Idee kooperativer virtueller Wissensräume zu einem erheblichen Anteil und sind entsprechend wichtiger Bestandteil der vorliegenden Arbeit.

1.7.3 Ansätze der Kooperationsunterstützung durch MUDs und MOOs

Nicht direkt als Lehr-/Lernmedium aber für die Koordination und Kommunikation der täglichen Arbeit berichten Churchill und Bly von ihren positiven Erfahrungen zum Einsatz ihres LambdaMOO-basierten MUD „Waterfall Glen“ (vgl. [Churchill & Bly 1999]). Insbesondere eine leichte und ungezwungene Kommunikation sowie ein ständig

präsen­ter Treffpunkt sind für sie die wichtig­sten Vor­teile des Ein­setzes eines textba­sierten MUD. Hier­bei wer­den be­wusst ein­fache Dar­stel­lungs­for­men ge­wählt, um das MUD mög­lichst unauf­fäll­ig in die täg­liche Ar­beit zu in­te­grieren. Durch ih­ren drei­jäh­ri­gen Ein­satz die­ser Um­ge­bung hat sich ge­zeigt, dass auf­wändi­ger Me­dienein­satz („me­dia rich­ness“) kei­ne not­wen­di­ge Be­ding­ung für den er­fol­grei­chen Ein­satz ei­ner ko­operati­onsun­ter­stützen­den Um­ge­bung sein muss.⁴¹

Als Haupt­vor­teile der Nut­zung von MUDs ge­gen­über klas­si­schen Chat-Räu­men oder ka­na­lorien­tierten Kom­mu­nikations­me­cha­nis­men (IRC) stel­len Church­ill und Bly die Mög­lich­kei­ten dar, durch we­ni­ge Schrit­te neue Räu­me in­ner­halb ei­nes MUD zu er­zeu­gen, zwi­schen die­sen durch ein­fache Kom­man­dos zu wech­seln und mit Ob­jek­ten in­ner­halb die­ser Räu­me zu in­te­ra­gieren. Vi­elleicht de­finiert sich durch die­se Ei­gen­schaft der Selbst­organi­sa­tion von MUDs ih­re brei­te Popu­larität in der Nut­zung als raumbasierte Kom­mu­nikations­plat­tf­orm.

Ver­tre­ter die­ser Grup­pe (I) der Ein­bet­te­ung des MUD in die Leh­re ohne den didakti­schen An­spruch der Ver­mit­tlung von Lehrin­hal­ten gren­zen sich be­wusst von den Ver­tre­tern der zwei­ten Grup­pe (II) ab. Spezi­ell die stringen­te Steu­erung der In­te­rak­ti­ons­fol­gen von ko­operati­ven Leh­r­situa­ti­onen wird als falscher Weg der Nut­zung der frei­en und unge­zwun­ge­nen MUD- und MOO-Idee be­wertet. So wirft bei­spiels­wei­se Fan­der­clai ei­nen kri­ti­schen Blick auf die ih­rer Mei­nung nach falsche Nut­zung von MUDs und MOOs in der Leh­re (vgl. [Fan­der­clai 1995]). Fan­der­clai be­sitzt weit­rei­chen­de Er­fah­run­gen in der Nut­zung ei­nes MUD zur Un­ter­stüt­zung von Kur­sen zum frei­en Schrei­ben (writing classes) und nutzt das MUD als Me­di­um für die Dis­kus­si­on der Lernenden in der Klasse un­te­rein­an­der bzw. mit an­de­ren, ex­ter­nen Lernenden. Sie ver­urteilt be­wusst den Ver­such, mit Hil­fe ei­nes Sys­tems (wie ei­nes MUD) real existieren­de Lern­struk­tu­ren (z.B. ei­ne ge­samte Uni­ver­si­tät) mit all ih­ren Kon­tro­llin­stan­zen und sozia­len Bar­rie­ren vir­tu­ell nach­bil­den zu wol­len.

Der­arti­ge starre Struk­tu­ren und die mit ihr verbun­de­nen Me­cha­nis­men der In­te­rak­ti­on (z.B. Turn-Taking-Mechanismen) wi­derspre­chen der Idee des MUD als kom­mu­nikationsun­ter­stützen­des flexibles und offenes Me­di­um. Treffend fasst Fan­der­clai den Ef­fekt mit der Fest­stel­lung zu­sam­men: „[...] while com­mands for con­troll­ing stu­dents’ be­havior may seem ef­fi­cient [...]. The po­ten­tial for em­pow­er­ing stu­dents is lost“ [ebd., S. 8]. MUDs sind Orte des selbst­organi­sierten Lernens, die von der Kreativität und der Schaffens­kraft der Lernenden selbst leben.

Als Ver­tre­ter der er­sten Grup­pe (I) fin­den sich ei­ne Viel­zahl von er­fol­grei­chen Ver­su­chen MUDs und MOOs in der Leh­re ein­zu­set­zen. Hier­bei fal­len ei­ne ganze An­zahl von Pro­jek­ten aus den Be­rei­chen des Erlernens von Fremdsprachen auf.⁴²

An­de­re MUDs und MOOs die­nen der Ko­operationsun­ter­stüt­zung von Wis­sen­schaf­tlern aus un­ter­schied­lich­sten Be­rei­chen. Den er­sten, sehr be­kannten Ver­such der Nut­zung von MOO-Techno­logie stellt das MediaMOO des MIT [Bruckman & Resnick

⁴¹ Hier­zu ist je­doch zu be­mer­ken, dass in ei­ner tech­ni­scher ge­präg­ter Um­ge­bung, aus der die Au­to­ren stam­men, die Ak­zeptanz für rein textbasierte Werkzeuge sicher­lich höher als in ei­nem heterogenen uni­ver­si­tä­ren Um­feld ist. – So setzt laut Aus­sa­ge von Church­ill und Bly ein Groß­teil der Nutzer ih­rer Um­ge­bung den Editor EMACS als Zu­gangs­werk­zeug für ihr MUD ein – ei­ne für Nicht-Infor­ma­tiker sicher­lich un­typi­sche Kon­figu­ra­tion.

⁴² Ohne den An­spruch auf Voll­stän­dig­keit zu er­he­ben, seien ge­nannt: MOOFrançais, MOOlano (Französi­scher); MundoHispano (Spani­scher); MOOsaico (Portugiesi­scher).

1995] dar, bei dem es sich um ein Forum für Medienwissenschaftler handelt. Es finden sich jedoch weitere Beispiele aus allen Bereichen der Wissenschaft.⁴³

MUDs und MOOs, welche von ihren Nutzern ausdrücklich dem Ausbildungssektor (Educational MOOs) zugeordnet werden, beschränken ihre Ausrichtung größtenteils nicht auf einen speziellen Nutzerkreis. Sie dienen zumeist der Unterstützung von Kursen oder Gruppen von Studierenden an Schulen und Universitäten, d.h. sie beinhalten Kursmaterialien, die von den Lernenden individuell und kooperativ bearbeitet werden können. Die Anzahl derartiger MUDs und MOOs ist relativ hoch. Eine umfangreiche Liste von MOOs, die zu Lernzwecken eingesetzt werden, wird an der University of Berkeley, USA, gepflegt.⁴⁴

In diesem Zusammenhang stellt sicherlich das Pueblo-System⁴⁵ ein wichtiges Beispiel dar (vgl. [O'Day et al. 1996] und [Mynatt et al. 1997]). Pueblo ist eine MOO-basierte virtuelle Lerngemeinschaft für Schüler und Lehrer und wird in Kooperation zwischen der Longview Elementary School Phoenix, Arizona, und dem Xerox PARC getragen. Pueblo ist nicht nur wegen seiner relativ großen Zahl von Nutzern interessant⁴⁶, das System wurde zudem um einige kooperative Werkzeuge (Shared Applets wie z.B. einen Shared Editor) zur gemeinsamen Bearbeitung von innerhalb der Lernumgebung abgelegten Objekten erweitert und ist deshalb schon der dritten Gruppe (III) der MUDs und MOOs, welche den aktiven Umgang mit Materialien beinhalten, zuzuordnen. Lehrenden werden spezielle Werkzeuge zur Administration und zur Kommunikation mit anderen Lernenden bereitgestellt. Ziel von Pueblo ist es, einen Ort für eine Gemeinschaft aus Lernenden zu schaffen. In diesem Zusammenhang wird speziell die Idee von Mentoren, also gegenseitiger Hilfestellung von Lernenden untereinander bzw. zwischen Lernenden und Dozenten, betont. Ähnlich zählt das LinguaMOO⁴⁷ der University of Texas, Dallas, USA zu den sicherlich am besten dokumentierten MOO-Universitäten (vgl. [Haynes & Holmevik 1998], [Holmevik & Haynes 1999]).

Einer der ersten Versuche, Grundschulkindern Programmierfähigkeiten mit Hilfe eines MOO zu vermitteln, wurde von Bruckmann im Projekt MOOSE [Bruckman 1997a] beschritten. Sie erweiterte das LambdaMOO-System um eine vereinfachte interne Programmiersprache, in der Kinder eigene Objekte entwerfen können und so spielerisch den Umgang mit einer Programmiersprache erlernen. Es handelt sich in dieser Form um eine sehr aktive Umgebung (Gruppe III), die zugeschnitten auf ein spezielles Gebiet den Umgang mit digitalen Medien (Lehrmaterialien) unterstützt.

Slator und Hill schließlich nutzten sämtliche Eigenschaften der MUD- und MOO-Technologie in verschiedenen Projekten (Gruppe I-III). Sie verwenden das MUD nicht nur als Kommunikations- und Diskussionsforum von Studierenden und Lehrenden (Gruppe I). Ihr Geology Explorer [Slator & Hill 1999] unterstützt Studierende der Geologie zudem inhaltlich bei der Analyse von Erdformationen und Gesteinsarten. Hierbei sind Aufgaben und Analysen nicht isoliert, sondern in Gruppen und thematisch stark verzahnt zu lösen (Gruppe II). Das ProgrammingLand MOO [Hill & Slator 2000] bein-

⁴³ Vgl. Jupiter [Curtis et al. 1995], Xerox PARC; LitMUSE der University of South Florida; SCU MOO, Southern Cross University, Lismore, Australia; TWUMOO der Texas Woman's University, welches die Archivierung und den Austausch elektronischer von Frauen verfasster Texte zum Ziel hat; PANGAEA, welches eine von Geologen gepflegte virtuelle Welt bereitstellt; BioMOO, Weizmann Institute of Science, Israel, ein Treffpunkt von Biologen aus aller Welt.

⁴⁴ Vgl. <http://cinemaspace.berkeley.edu/~rachel/moolist/edu.html>, Stand 1.12.2001.

⁴⁵ Vgl. http://pcacad.pc.maricopa.edu/Pueblo/index_frame.html, Stand 1.12.2001.

⁴⁶ Pueblo umfasst ca. 350 aktive Nutzer, vgl. [O'Day et al. 1996, S. 162].

⁴⁷ Vgl. telnet://lingua.utdallas.edu:8888/ und <http://lingua.utdallas.edu>, Stand 1.12.2001.

haltet Kursinhalte zur Programmierung und ist der dritten Gruppe (III) zuzuordnen. Studierende der Informatik bewegen sich u.a. zwischen Vorlesungs- (lecture rooms), Arbeits- (work rooms) und Quiz-Räumen (quiz rooms) [ebd., S. 409ff.] und erlernen elementares Grundwissen zu Programmier-techniken. Das ProgrammingLand MOO enthält eine ganze Anzahl interaktiver Demonstrationsobjekte, welche Slator und Hill als „demonstrations, toys, robots and interactive exhibits“ [ebd., S. 410] beschreiben. Ein Beispiel ist die „code machine“, welche eine schrittweise Ausführung von Programm-codes erlaubt.

In seiner didaktisch-thematischen Ausrichtung ähnelt ProgrammingLand MOO an vielen Stellen der Idee des klassischen Rollenspiels eines MUD – Studierende bekommen Aufgaben gestellt (z.B. im Quiz-Raum), welche es zu lösen gilt. Demonstrationsobjekte innerhalb von Arbeitsräumen werden genau wie in den meisten MUDs nach der Nutzung durch einen Studierenden automatisch in den Ausgangszustand zurückversetzt. Innerhalb der Vorlesungsräume wird jeweils ein Abschnitt der Vorlesung vermittelt.

Zu den langjährigen Nutzern von MUD- und MOO-Technologien im Ausbildungssektor gehören weiterhin die Diversity University⁴⁸, das CheshireMOOn⁴⁹, welches an der University of Austin, Texas, betrieben wird, KCCMO'O⁵⁰ des Kapi'olani Community College in Hawaii und eine Vielzahl weiterer Institutionen, wie z.B. Grassroots MOO⁵¹, AussieMOO⁵², MOOVille⁵³ oder ROO MOO⁵⁴.

Resümierend ist festzuhalten, dass sämtliche der oben genannten Ansätze die Idee der kooperativen Räume berücksichtigen. Kooperative, virtuelle Räume sind thematisch orientiert und bilden einen virtuellen Treffpunkt von Nutzern und Lernenden im Netz. Anteilig finden sich zudem einzelne Ansätze, die unabhängig der Kommunikationsunterstützung des virtuellen Raums auch qualitative Vorteile des kooperativen *Wissensraums* identifiziert und ansatzweise entwickelt haben. Keiner der untersuchten Ansätze entwickelt hierbei ähnlich dem Vorgehen dieser Arbeit gezielt eine Sicht auf medientechnische Unterstützungsfunktionen im Umgang mit Lehrmaterialien. Das Potenzial der kooperativen Gestaltung und Nutzung von digitalen Medien (Lehrmaterialien) erscheint jedoch erkannt.

Im Gegensatz zum Bereich der CSCW- und CSCL-Systeme wird die Gestaltung des virtuellen Raums durch die Nutzer der Umgebung selbst in allen untersuchten Ansätzen als selbstverständliche Qualität der MUD- und MOO-Technologie verstanden und seit den ersten Gehversuchen der zugrunde liegenden Technik praktiziert. Dies macht speziell ihre technische und architektonische Fundierung für die vorliegende Arbeit interessant (vgl. Abschnitt 4.2.5).

Der verfolgte Ansatz versucht nicht, existierende Ansätze vollständig in ihrer Funktionalität zu erfassen und zu beschreiben. Ziel ist es, einzelne Vertreter typischer Systemklassen auf ihre Umsetzbarkeit hinsichtlich praxisorientierter Szenarien zu untersuchen. Eine Bewertung kooperationsunterstützender und lernförderlicher Infrastrukturen und Systeme kann nur auf der Basis eines zugrunde liegenden theoretischen Fundaments

⁴⁸ Vgl. <http://www.du.org/>, Stand 1.12.2001.

⁴⁹ Vgl. <http://www.cwrl.utexas.edu:9000/>, Stand 1.12.2001.

⁵⁰ Vgl. <http://moo.kcc.hawaii.edu/~moo/kccmoo/web/www/index.html>, Stand 1.12.2001.

⁵¹ Vgl. <http://www.enabling.org/grassroots/>, Stand 1.12.2001.

⁵² Vgl. <http://www.aussiemoo.org>, Stand 1.12.2001.

⁵³ Vgl. <http://web.nwe.ufl.edu/writing/help/moo/>, Stand 1.12.2001.

⁵⁴ Vgl. <http://iml.umkc.edu/roo/>, Stand 1.12.2001.

erfolgen.⁵⁵ Benötigt wird ein theoretisches Konzept zur Abgrenzung und Systematisierung von Befunden und abhängigen Variablen, das es auch Technikern erlaubt das Rationalisierungspotenzial von digitalen Medien qualitativ bewerten zu können. Eine derartige Entscheidungsgrundlage ist das Konzept der Medienfunktionen. Die vorliegende Arbeit erweitert ein derartiges Modell zudem um die Dimensionen kooperativen Lernens.

Im Folgenden werden zunächst primäre Medienfunktionen in Abgrenzung zu sekundären und tertiären Medienfunktionen dargestellt (Abschnitt 2.1). Als wesentlicher Teil der theoretischen Grundlage dieser Arbeit wird das Konzept der primären individuellen und primären kooperativen Medienfunktionen unterschieden (Abschnitt 2.2 und Abschnitt 2.3). Die Möglichkeit zur Selbstorganisation und Selbstadministration einer Lernumgebung ist bei der Entwicklung eines Konzepts virtueller Wissensräume wesentlicher Anspruch (Abschnitt 2.4). Anschließend werden eine Reihe von Szenarien entwickelt, die den Umgang mit primären Medienfunktionen im Kontext kooperativen Lernens beschreiben (Abschnitt 2.5). Anhand dieser sehr praxisorientierten Szenarien werden funktionale Anforderungen kooperativer Wissensräume formuliert (Abschnitt 2.6) und eine Reihe von Systemen auf ihre Umsetzbarkeit hin untersucht (Abschnitt 2.7). Schließlich dienen die dargestellten Szenarien der konzeptionellen Entwicklung der im Rahmen dieser Arbeit gestalteten Rahmenarchitektur (Kapitel 3).

⁵⁵ Im Folgenden sei keine detaillierte Analyse kooperativer Formen der Wissensorganisation in der Hochschullehre vorgenommen. Für eine Analyse des Einsatzes Neuer Medien und kooperativer Lehr- und Lernprogramme im Bereich der Erwachsenenbildung vgl. [Keil-Slawik & Selke 1998] und [Schulmeister 2001].

2. Kooperative Medienfunktionen – Szenarien

Menschliches Lernen findet in einer Vielzahl von Konstellationen und Formen statt. In den Theorien zum computergestützten, kooperativen Lernen (CSCL) werden beispielsweise Unterscheidungen getroffen nach synchronem und asynchronem Lernen, kooperativem oder individuellem Lernen, Lernen an einem oder an verschiedenen Orten, vorstrukturiertem oder selbstorganisiertem Lernen sowie symmetrischem (Gruppendiskussion) oder asymmetrischem Lernen (Präsentation).

Eine systematische Aufarbeitung sämtlicher Dimensionen computergestützten, kooperativen Lernens gelingt aufgrund mangelnder Systematisierung und Zusammenführung verschiedener Konzepte nur schwerlich.⁵⁶ Schon ein Teilbereich kooperativen Lernens, die computermedierte Kommunikation, ist aufgrund ihrer Verflechtung von organisatorischen, soziologischen und technologisch bedingten Faktoren nur schwer in ein systematisches Modell abzubilden. Ein derartiges Unterfangen kann und will diese Arbeit nicht leisten.

Trotzdem ist es erforderlich, technische von nicht-technischen Problemen zu differenzieren und ein Instrument für die Bewertung existierender Systeme sowie für die Identifikation, Systematisierung und Gestaltung von Technik in Prozessen menschlichen Lernens zu entwickeln. Die vorliegende Arbeit greift die Idee der Medienfunktionen nach [Keil-Slawik & Selke 1998a, S. 172ff.] auf und erweitert sie um wesentliche Elemente. Hierzu zählt insbesondere eine weitere Systematisierung der so genannten primären Medienfunktionen sowie die Schaffung eines differenzierten Verständnisses der Rolle von medientechnischen Funktionen innerhalb kooperativer Prozesse des Lernens.

Neu und wesentlicher Anteil des theoretischen Ansatzes der vorliegenden Arbeit ist in dieser Form eine Systematisierung (Benennung) und Unterteilung der primären Medienfunktionen in primäre individuelle und primäre kooperative Medienfunktionen sowie die Identifikation dieser Medienfunktionen.

Zu einer ersten Überprüfung der These der Identifikation kooperativer Medienfunktionen als wesentlicher Funktion im Umgang mit Materialien innerhalb von kooperativen Lernprozessen⁵⁷ ist es erforderlich, eine Reihe von Szenarien des Lernens mit Hilfe

⁵⁶ Eine systematische Aufbereitung und Zusammenführung von Modellen und Theorien computergestützten, kooperativen Lernens hat aus diesem Grunde bis jetzt nicht stattgefunden.

⁵⁷ Ziel der vorliegenden Arbeit ist die Bereitstellung primärer Medienfunktionen in virtuellen Wissensräumen. Beeinflussungsfaktoren sind virtuelle Lerngemeinschaften, aber auch der Anspruch der Entwicklung vom rein autorenzentrierten Medium hin zu lebendigen produzentenzentrierten

kooperativer Wissensräume abzubilden. Hierzu werden verschiedene Szenarien aus der alltäglichen Lehre gewählt. Weiterhin rechtfertigt sich die Konzeption und Umsetzung einer technischen Lösung kooperativer Wissensräume nur in dem Defizit nicht vorhandener Systeme und Ansätze. Um die Fragestellung der Umsetzbarkeit der gewonnenen Szenarien mit Hilfe verfügbarer Systeme beantworten zu können, ist ihre detaillierte Untersuchung notwendig.

Das vorliegende Kapitel führt zunächst in das Konzept der Medienfunktionen ein und systematisiert die Aufgaben und Möglichkeiten technischer Systeme im kooperativen Umfeld entlang der primären Medienfunktionen. Auf diese Weise ist die Basis geschaffen, um Anwendungsszenarien der Einbettung primärer Medienfunktionen in die Idee virtueller Wissensräume vornehmen zu können. Am Beispiel von Szenarien der Vorlesung, der Übung, dem Seminar und einer studentischen Projektgruppe werden konkrete Handlungen an Lehrmaterialien dargestellt. Abschließend werden drei viel versprechende Vertreter von Systemen auf die Umsetzbarkeit der dargestellten Szenarien hin untersucht.

2.1 Medienfunktionen als Basis kooperativer Lernprozesse

Die Idee der Medienfunktionen beruht auf theoretischen Systematisierungen zur Rolle von Technik in Prozessen des menschlichen Denkens und Lernens. Keil-Slawik und Selke konzentrieren sich zunächst auf zwei Grundüberlegungen zum Einfluss von Artefakten und Ausdrucksmitteln auf menschliche Denk- und Verarbeitungsprozesse (vgl. [Keil-Slawik 1990]):

Zum einen machen sie, basierend auf den Überlegungen von Gibson (vgl. [Gibson 1982]), deutlich, dass menschliches Denken an physisches Handeln und sinnliche Wahrnehmung gebunden ist. Damit ist auch menschliches Lernen an Medien gebunden, Informationsgewinnung ist nur im tätigen Umgang mit Medien möglich.

Als zweite wesentliche Erkenntnis ist zu beachten, dass die Evolution des menschlichen Geistes im Wesentlichen eine Evolution der Ausdrucksmittel ist. Diese von Leroi-Gourhan angeführte These sieht den heutigen Menschen genetisch nicht besser ausgestattet als seine Vorfahren, lediglich Fähigkeiten des Umgangs mit Zeichensystemen, Schrift und Zahl haben sich entscheidend weiter entwickelt (vgl. [Leroi-Gourhan 1988]). Ohne die Fähigkeiten, mit technischen Mitteln Zeichen und Zeichensysteme in den Handlungsraum des Menschen zu bringen, ist menschliches Lernen dementsprechend kaum denkbar [Keil-Slawik 2001, S. 110].

Eine Analyse kooperativen Lernens beruht auf zwei grundlegenden Argumentationssträngen. Zunächst kann menschliches Lernen basierend auf den wesentlichen Thesen des Denkens als individueller Prozess der Informationsverarbeitung, des tätigen Umgangs mit Artefakten, als Evolution der Ausdrucksmittel verstanden werden. Zum anderen ist menschliches Lernen ein sozialer Prozess zwischen verschiedenen Personen. Lernen ist in dieser Form ein Prozess der kooperativen Erschließung der menschlichen Umwelt. Es ist durch den individuellen sowie kooperativen Umgang mit Medien gekennzeichnet und ohne die Unterstützung von Technik nicht denkbar. Zeichensysteme sind in dieser Form Medien des produktiven Denkens, Erkenntnis- und Ausdrucksmittel. Kernidee der Medienfunktionen ist es, physische (technische) Elemente zu betrachten, die sich innerhalb eines Umgangs mit Medien ausmachen lassen. Derartige physische

Umgebungen. Selbstorganisation und Selbstadministration einer Lernumgebung durch die Nutzer sind wesentlicher Teil dieses Ansatzes.

Elemente erzeugen eine neue Qualität im Umgang mit dem jeweiligen Medium. Entscheidungsgrundlage für die Identifikation einer Medienfunktion ist in dieser Form das Isolieren eines physischen Elementes in ihrer Ausübung. So ist beispielsweise die Adresse eines Empfängers (meist eines Menschen) physisches Element einer eigenen Medienfunktion (des Übertragens). Anhand der Möglichkeit des Adressierens eines Kooperationspartners können demgemäß gezielt mediale Mehrwerte einer untersuchten Umgebung festgemacht werden.

Keil-Slawik und Selke unterscheiden zunächst drei Klassen von Medienfunktionen, die primären, sekundären und tertiären (vgl. [Keil-Slawik & Selke 1998a, S. 173]). Hierbei wird systematisch die Rolle von Technik innerhalb menschlicher Lernprozesse untersucht. Die drei gewählten Klassen repräsentieren den qualitativen Mehrerwerb medientechnischer Funktionen. So beinhalten primäre Medienfunktionen zunächst elementare Funktionen, um Artefakte in den Wahrnehmungsraum des Menschen (einer Gruppe von Menschen) zu bringen, sekundäre Medienfunktionen erweitern die primären Medienfunktionen um Wissen über den Lernprozess an sich, also Konstellationen des Gebrauchs von Artefakten, wie didaktische Modelle. Tertiäre Medienfunktionen berücksichtigen Methoden der Selbstanpassung und künstlichen Intelligenz, d.h. das Medium selbst wird zum Verarbeitungsgegenstand:

- I. *Primäre Medienfunktionen* dienen dazu, Zeichen bzw. Zeichensysteme in das Wahrnehmungsfeld des Menschen zu bringen.
- II. *Sekundäre Medienfunktionen* beinhalten verschiedene Konstellationen des Gebrauchs. Sie verkörpern das Wissen über Lernprozesse, welches sich in didaktischen Methoden und Zusammenhängen widerspiegelt.
- III. *Tertiäre Medienfunktionen* machen das Wissen über den Gebrauch von Zeichen und Zeichensystemen selbst zum Verarbeitungsgegenstand, dabei werden Methoden der Selbstanpassung oder künstlichen Intelligenz berücksichtigt.

Primäre Medienfunktionen beschreiben zunächst das Rationalisierungspotenzial, in das Denk- und Lernprozesse auf den Gebrauch physischer Artefakte angewiesen sind. Sie identifizieren konkrete technische Mehrwerte elementarer medientechnischer Funktionen. Die qualitative Bewertung primärer Medienfunktionen orientiert sich demgemäß an der Art und Weise wie es gelingt, durch technische Hilfsmittel Zeichen und Zeichensysteme, also Lehrmaterialien, in den Wahrnehmungs- und Handlungsraum des Menschen zu bringen. Die vorliegende Arbeit wird im Folgenden das theoretische Modell der primären Medienfunktionen weiter systematisieren und im Hinblick auf kooperative Tätigkeiten entwickeln.

Sekundäre Medienfunktionen sind nicht-technischer Natur und machen den Lernprozess selbst zum Untersuchungsgegenstand. Hier ist die Gestaltung von Lehrmaterialien und der Zugang zu diesen Materialien wesentlich.⁵⁸ Sekundäre Medienfunktionen

⁵⁸ Lehrbücher, Aufgabensammlungen und kommentierte Nachschlagewerke sind Beispiele von didaktisch motivierten Lehrmaterialien. Gleichzeitig werden verschiedene Modelle entwickelt, die die Abfolge von den Lernschritten der Lernenden vorgeben bzw. die Lernenden durch die Lehrmaterialien führen. Systeme des Computer-Based Trainings (CBT), multimediale Lehrmaterialien (CD-ROMs) oder interaktive tutorielle Lernsysteme im WWW sind Ausdrucksformen des Instruktionsdesigns und damit den sekundären Medienfunktionen zuzuordnen. Schließlich fällt auch die Gestaltung von Kooperationsstrukturen in Gruppenprozessen in den Bereich der sekundären Medienfunktionen. Werden innerhalb einer kooperationsunterstützenden Umgebung spezifische Mechanismen bereitgestellt, welche die Struktur des Lernprozesses beeinflussen bzw. ein Kooperationsmodell oder didaktisches Modell zugrunde legen, sind diese ebenfalls dem Bereich der sekundären Medienfunktionen zuzuordnen.

können ohne Frage einen hohen Wirkungsgrad entwickeln, müssen jedoch als eigenständiges Forschungsfeld verstanden werden. In der konkreten Aufgabe der Entwicklung eines Rahmenkonzepts für kooperatives Lernen fallen damit eine ganze Reihe von Forschungsfragen in den Bereich der sekundären Medienfunktionen und werden bewusst nicht in das technische Rahmenkonzept einbezogen.⁵⁹ Zu Forschungsfragen, welche in der konzeptionellen und technischen Realisierung aufgrund ihrer Zugehörigkeit zu der Klasse der sekundären oder teilweise schon tertiären Medienfunktionen nicht berücksichtigt werden, zählen:

- *Kooperationsstrukturen in Gruppenprozessen* – Verfahren, die die Struktur von Zusammenarbeit beeinflussen, wie beispielsweise die Reihenfolge von Redebeiträgen in einem kooperativen System (aufwändige Strategien des Floor Control), oder die Struktur eines kooperativen Schreibprozesses kontrollieren (Autorenzyklus).
- *Feste Rollen und Rechte* – Vorgaben der Rollen innerhalb eines Lernprozesses, wie beispielsweise Lehrer und Schüler, Tutor und Student, die eine Struktur von Lernprozessen in Bezug auf Berechtigungen vorgeben und kooperative Strukturen entlang dieser Rollen organisieren.
- *Abstimmungssysteme/Voting* – Feste Mechanismen der Entscheidungsfindung innerhalb von Lernprozessen und Kooperationsprozessen; Systeme, die eine Systemantik in der Konsensfindung herbeiführen.
- *Brainstorming, Diskussionsstrukturierung* – Systeme, die die Struktur von Denkprozessen und Kommunikation beeinflussen.

Tertiäre Medienfunktionen charakterisieren das Medium selbst als anpassungsfähig und lernfähig. Auch wenn das Problem der Entwicklung von Systemen der Klasse der tertiären Medienfunktionen nicht gelöst ist und die Lösbarkeit allgemein fragwürdig erscheint, existieren doch eine ganze Reihe von Versuchen „intelligente“ und „lernfähige“ Systeme zu schaffen. Beispiele sind adaptive Systeme, die sich automatisch auf ein Benutzerverhalten einstellen und sich basierend auf verschiedenen Benutzermodellen anpassen. Lernsysteme können sich auf diese Weise fortwährend an das Wissen und den Lernfortschritt eines Lernenden einstellen. Derartige intelligente tutorielle Systeme speichern das Wissen von Experten und das Wissen des Lernenden und versuchen über spezielle Strategien auf die Fragen und Probleme des Lernenden zu reagieren.

Bezogen auf den Bereich der kooperativen Wissenskonstruktion existieren eine Reihe von Ansätzen mit Elementen tertiärer Medienfunktionen.⁶⁰ Tertiäre Medienfunktionen besitzen wie schon die sekundären Medienfunktionen für den verfolgten Ansatz nur wenig Relevanz für die Entwicklung von kooperationsunterstützender Technik für Lehr- und Lernprozesse. Nicht das intelligente und lernfähige Medium

⁵⁹ In der aktuellen Forschung zum CSCL/CSCW findet keine derartige Abgrenzung statt, was sich negativ auf die Bewertbarkeit und Klassifizierung vorhandener technischer Systeme auswirkt.

⁶⁰ Davies et al. forschen schon seit einigen Jahren an Methoden und Konzepten, durch einen agentenbasierten Ansatz Wissen in Gruppen verfügbar zu machen. Basierend auf ihren früheren Arbeiten zu dem Jasper-System [Davies et al. 1995] erhebt ihre Umgebung KSE (Knowledge Sharing Environment) den Anspruch, durch gezielte Aufbereitung und Filterung von Wissen (Netzseiten, E-mails, persönliche Notizen etc.) so genannte „communities of interest“ aufzubauen. Persönliche Agenten speichern in dieser Form Meta-Informationen zu Webseiten und stellen diese Informationen für die Gemeinschaft bereit – ein Konzept, das als „social filtering“ oder, in dieser Form geprägt durch automatisches Vorschlagen, als „automated collaborative filtering“ bezeichnet wird [Herlocker et. al. 2000]. Ähnliche Ansätze sind das Concept Indexing System von [Voss et al. 1999] oder GroupLens [Konstan et al. 1997].

ist für den Lernfortschritt von Lernenden verantwortlich, es sind die Lernenden selbst, die durch ihre kooperative Zusammenarbeit und Intelligenz den Wissenszuwachs erzielen.

Wesentlicher Bestandteil jedes Lernprozesses sind damit in erster Linie die primären Medienfunktionen. Sie weisen ein großes Potenzial in der Intensivierung und Rationalisierung von Lernprozessen auf. Mit der Möglichkeit, Materialien in den Handlungsraum des Menschen zu bringen und mit diesen möglichst fließend ohne störende Medienbrüche zu lernen, wächst das geistige Leistungsvermögen des Lernenden und seine Fähigkeit bedeutungsvolle Sinnzusammenhänge zu schaffen. Eine erste detaillierte Auseinandersetzung mit den primären Medienfunktionen findet sich in [Keil-Slawik & Selke 1998]. Hier beschränkt sich ein Verständnis primärer Medienfunktionen zunächst auf die „Erzeugung“, „Verknüpfung“ und „Speicherung“⁶¹, also auf den Bereich *individuellen* Lernens und Arbeitens. Aus dem Defizit der Beschreibung und eines Verständnisses von Prozessen kooperativen Lernens mit Hilfe der primären Medienfunktionen wurden in einem zweiten Schritt die primären Medienfunktionen um das Übertragen von Materialien zwischen Lernenden erweitert (vgl. [Hampel & Keil-Slawik 2001a]). Auch hier ist das Übertragen noch sehr ungenau gefasst. Es wird nicht zwischen dem Übertragen von Materialien an verschiedenen Orten des Lernens und dem Austausch von Materialien zwischen Lernenden unterschieden. Defizitär erscheint weiterhin, dass kooperative Tätigkeiten wie das Synchronisieren von Sichten oder Ereignissen oder auch das Zugreifen auf Materialien innerhalb kooperativer Prozesse sich nicht in den beschriebenen Medienfunktionen erklären lassen.

Die vorliegende Arbeit nimmt eine Differenzierung der primären Medienfunktionen in primäre *individuelle* und primäre *kooperative* Medienfunktionen vor. Den primären individuellen Medienfunktionen wird die Funktion des Löschens hinzugefügt. Auf Seite der primären kooperativen Medienfunktionen wird der zugrunde gelegte Ansatz gezielt um Merkmale kooperativen Arbeitens und Lernens erweitert. Primäre individuelle Medienfunktionen wie Erzeugen, Löschen, Arrangieren und Verknüpfen werden im Kontext kooperativer Lernprozesse gesehen. Hinzu kommen Medienfunktionen des Übertragens, Zugreifens und Synchronisierens, die zusätzliche, grundlegende Bereiche kooperativen Lernens und Arbeitens erfassen.

2.2 Primäre individuelle Medienfunktionen

In einem ersten Schritt werden Medienfunktionen untersucht, die einem individuellen Umgang mit Medien dienen. Dieser individuelle Umgang mit Medien findet auch im kooperativen Lernen statt, insofern ist beispielsweise die Medienfunktion des Erzeugens auch im Umfeld gemeinsam genutzter Materialien auszumachen. Primäre individuelle Medienfunktionen lassen sich gemäß der im Rahmen dieser Arbeit entwickelten Systematisierung auf das Erzeugen, Löschen, Arrangieren und Verknüpfen reduzieren. Primäre kooperative Medienfunktionen stellen den individuellen primären Medienfunktionen im Wesentlichen das Übertragen, Zugreifen und Synchronisieren zur Seite (vgl. Abschnitt 2.3).

⁶¹ Die genannten Medienfunktionen entsprechen weitgehend den in der vorliegenden Arbeit entwickelten Medienfunktionen des Erzeugens, Arrangierens und Verknüpfens.

Kreieren und Erzeugen

Medien schaffen einen Wahrnehmungsraum des Menschen. Dieser erlaubt es, Vorstellungen und Realität durch Handlungen in Beziehung zu setzen und daraus entsprechende Schlussfolgerungen abzuleiten. Die Medienfunktion des Kreierens, also mit technischen Mitteln Zeichen zu entwerfen, ist die erste wichtige Funktion im tätigen Handeln mit Medien. Konkrete Tätigkeiten des Kreierens sind das Schreiben, Anfertigen von Skizzen, aber auch das Erschaffen von Modellen. Im Umfeld des Lernens ist die Medienfunktion des Kreierens eng verbunden mit der durchgängigen Verfügbarkeit von Lehrmaterialien.⁶²

Neben der genannten durchgängigen Verfügbarkeit ist die Persistenz der erzeugten Materialien wichtiges Kriterium der erzeugten Materialien. Kreieren bezieht sich demgemäß auf das dauerhafte, persistente Schaffen von Handlungsräumen für den Lernenden.

Löschen

Analog zum Kreieren von Materialien müssen diese gezielt wieder aus dem Wahrnehmungsraum des Menschen entfernt werden können. Löschen ist grundlegende Basis jedes Lern- und Arbeitsprozesses und wird im Rahmen dieser Arbeit als eigenständige Medienfunktion identifiziert. Im Prozess des Lernens gilt es, an vielen Stellen Materialien zu überarbeiten, sie zusammenzuführen und entsprechend Teile oder Bereiche zu löschen. Löschen kann sich auf das gesamte Material oder auf dessen Teile erstrecken.

Die Medienfunktion des Löschens beeinflusst den Wahrnehmungsraum des Menschen (der Lernenden) in vielfältiger Weise. So gilt es im Zusammenhang mit dem Löschen, speziell die Auswirkungen auf andere Materialien der Lernumgebung zu untersuchen.

Das Löschen einer Referenz beispielsweise beeinflusst die Zugreifbarkeit des referenzierten Materials. Das Löschen einer Übersichtsdarstellung oder Verzeichnisses kann den Zugriff auf die einzelnen Materialien gänzlich unmöglich machen.

In der Gestaltung kooperativer Wissensräume sind folglich Strategien für das Löschen von Materialien und seine Auswirkungen auf andere Materialien zu entwickeln. So sind beispielsweise Festlegungen bei der Löschung eines einzelnen Dokumentes oder ganzer Gruppen von Dokumenten (beispielsweise Löschen eines gesamten virtuellen Raums) zu treffen.

Arrangieren

Menschliches Lernen ist maßgeblich durch In-Beziehung-Setzen von Erkenntnissen gekennzeichnet. Um zu neuen Einsichten zu gelangen, ist es notwendig, verschiedene Materialien (Dokumente, Sichten auf einen Lerngegenstand) miteinander in Beziehung zu setzen. Hierbei werden von dem Lernenden Differenzen und Übereinstimmungen festgestellt und Beziehungen zwischen unterschiedlichen Darstellungen hergestellt.

Aus der Kombination von verschiedenen Beschreibungen, Darstellungsformen und Aussagen ergeben sich neue Einsichten und logische Zusammenhänge. Wichtig für die Unterstützung des Prozesses des In-Beziehung-Setzens ist es, mehrere Materialien mög-

⁶² Lehrmaterialien müssen von verschiedenen Lernorten erreichbar sein, was nicht explizit eine Anwendung der Medienfunktion des Übertragens voraussetzt (diese bezieht sich auf den Austausch von Medien zwischen Lernenden), sondern der Medienfunktion des Zugreifens entspricht (vgl. Abschnitt 2.3.2).

lichst gleichzeitig in das Wahrnehmungsfeld des Lernenden bringen zu können und deren freie Anordbarkeit zu gewährleisten.

Traditionelle Lernprozesse sind dementsprechend durch die räumliche Anordnung von Beschreibungen, Texten, grafischen Darstellungen, Zeichnungen etc. in dem Wahrnehmungsfeld des Lernenden (beispielsweise dem Schreibtisch) gekennzeichnet. Begleitend zum Prozess des Erkenntniszuwachses werden verschiedene Materialien räumlich zugeordnet, verglichen und gegeneinander abgegrenzt.

Die Medienfunktion des Arrangierens charakterisiert diesen Prozess des In-Beziehung-Setzens von Materialien. Dabei sollen logische Zusammenhänge möglichst auch räumlich zusammenhängend verkörpert werden, damit sie schnell erkannt und bearbeitet werden können.

Verknüpfen

Ähnlich wie die Medienfunktion des Arrangierens bezieht sich die Medienfunktion des Verknüpfens auf das Herstellen von Beziehungen zwischen einzelnen Materialien. Arrangieren ist die elementare Vorbedingung zur Schaffung eines Wahrnehmungs- und Handlungsraums für den Lernenden. Um derartige Arrangements nicht für jeden Lernprozess erneut herstellen zu müssen, helfen Verknüpfungen dauerhafte Beziehungen zwischen Lehrmaterialien herzustellen. Diese können in späteren Phasen des Lernprozesses erneut genutzt, aber auch mit anderen Lernenden besprochen werden (vgl. Abschnitt 2.3.1).

Verknüpfungen zwischen Lehrmaterialien erlauben es dem Lernenden, jegliche, für einen erschlossenen Bedeutungszusammenhang notwendige Materialien in den Wahrnehmungsraum zu bringen. Hierbei kann sich der Bedeutungszusammenhang nur auf einen Teil der Unterlagen erstrecken. Wichtiges Prinzip ist in diesem Zusammenhang die Reduktion und Konzentration von Wissen. Beispielsweise bezieht sich ein Autor auf Werke anderer Autoren und verknüpft bestimmte Aussagen dieser Autoren zu einem neuen Argumentationsstrang. Aussagen, Zitate und Kopien verschmelzen zusammen mit den Aussagen des Autors zu einem physisch zusammenhängenden neuen Textgebilde. Verknüpfen bedeutet demgemäß, Referenzen zwischen beliebigen Materialien herzustellen und gezielt Teile von Materialien in einen logischen Zusammenhang zu setzen.

Verknüpfungen sind unabhängig von der Gestalt und Position der verbundenen Materialien (Texte, Grafiken etc.). Sie spiegeln von den Lernenden oder Lehrenden manifestierte mentale Strukturierungen wider. Beispiele der Medienfunktion des Verknüpfens finden sich in allen Bereichen des computergestützten Umgangs mit Materialien.

Physisch nicht verbundene Materialien werden durch Verweise verknüpft. Analog zur Terminologie des Hypertextes werden verschiedene Dokumente (Texte, Grafiken etc.) durch Verweise verknüpft, ohne ihre Position selbst zu verändern. Das Erzeugen, Einfügen eines Verweises zwischen zwei Dokumenten entspricht ihrer dauerhaften Verknüpfung, sie besteht unabhängig von der Position der verknüpften Materialien.

2.3 Primäre kooperative Medienfunktionen

Individuelle primäre Medienfunktionen beschreiben zunächst Tätigkeiten und Handlungen an Materialien eines einzelnen Nutzers bzw. Lernenden. Natürlich sind diese auch im kooperativen Umfeld Basis jeden Lernprozesses. Löst man sich von der Betrachtungsweise isoliert voneinander agierender Lernender, lassen sich wesentliche physische Handlungen an Materialien mittels primärer individueller Medienfunktionen nicht beschreiben. Hierzu zählt beispielsweise das Synchronisieren einer Anzahl von Sichten auf ein Dokument oder das kooperative Betrachten von Netzseiten. Ein wesentlicher Beitrag

des theoretischen Anteils der vorliegenden Arbeit ist es, als Teil der primären Medienfunktionen kooperative Handlungen innerhalb computergestützter Lernprozesse zu identifizieren und systematisch zu untersuchen.

Offensichtliches Beispiel für eine kooperative primäre Medienfunktion ist der Austausch von Lehrmaterialien zwischen Lernenden. Die Übergabe eines Übungsblattes, einer Grafik oder Notizen von einem Lernenden zu einem Mitlernenden besitzt eine neue Qualität gegenüber dem individuellen Arrangieren oder Verknüpfen von Materialien. Eine auf diese Weise identifizierte zusätzliche Medienfunktion des Übertragens lässt sich demgemäß an dem Adressieren eines Menschen (Mitlernenden) festmachen.

In ähnlicher Weise ist das Lesen oder Bearbeiten von Materialien, die Lernende innerhalb eines virtuellen Raums abzulegen vermögen, nicht mit den primären individuellen Medienfunktionen beschreibbar. Berechtigungen kontrollieren Kooperationspartner in ihren Handlungen an gemeinsam genutzten Materialien: Sie finden ihren Ausdruck in der Medienfunktion des Zugreifens.

Schließlich charakterisiert die Medienfunktion des Synchronisierens sämtliche Aspekte gemeinsamer Sichten, Repräsentationen und der Übermittlung von Ereignissen. Kooperationsunterstützende Systeme bieten in dieser Form Sichten auf gemeinsame Objekte und übermitteln Ereignisse und Informationen der gegenseitigen Wahrnehmung.

Basierend auf der im Rahmen dieser Arbeit gewonnenen Systematisierung werden drei primäre kooperative Medienfunktionen unterschieden (vgl. Abbildung 2-1).⁶³

- a) *Übertragen*: Transport (aktives Übertragen) von Medien zwischen Lernenden
- b) *Zugreifen*: Zusprechen von Berechtigungen an gemeinsamen Materialien, kooperative Nutzung individueller primärer Medienfunktionen (Zugriff auf die Materialien einer anderen Person), Zugriff auf Materialien von verschiedenen Lernorten (durchgängige Verfügbarkeit)
- c) *Synchronisieren*: Übermitteln von Sichten auf gemeinsame Objekte, Kopplung, gegenseitige Wahrnehmung

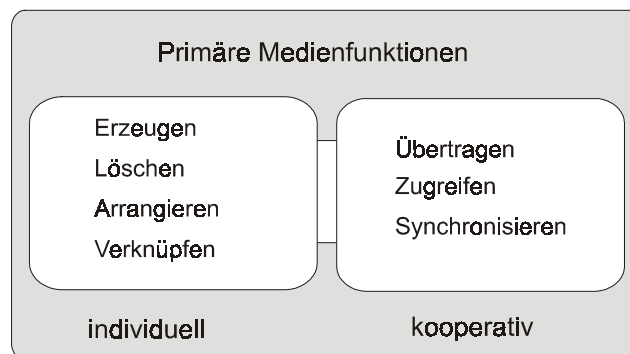


Abbildung 2-1: Primäre individuelle und kooperative Medienfunktionen

Da, wie schon gezeigt, für das technische Rahmenkonzept dieser Arbeit die individuellen und kooperativen primären Medienfunktionen von vorrangiger Bedeutung sind, wird in Zukunft bei Nennung der primären Medienfunktionen schlicht von Medienfunktionen gesprochen.

⁶³ Die Identifikation der primären kooperativen Medienfunktionen des Übertragens, Zugreifens und Synchronisierens geschieht im Vorgriff auf die in Abschnitt 2.5 dargestellten Szenarien des Lernens in virtuellen Wissensräumen.

Im Zusammenhang mit der Bereitstellung kooperativer Medienfunktionen im Umfeld menschlichen Lernens sind zwei Schlüsselfragestellungen zu berücksichtigen: Zum einen gilt es entsprechend dem Anspruch der durchgängigen Verfügbarkeit von Lehrmaterialien, Medienbrüche bei der Ausübung der Medienfunktionen zu reduzieren. Zum anderen verlangen echte, kooperative Medienfunktionen, wie die Notwendigkeit kooperativ arrangieren und verknüpfen zu können, eine Selbstorganisation bzw. Selbstadministration von Teilen der Lernumgebung durch die Lernenden. Im Folgenden werden zunächst kooperative Medienfunktionen des Übertragens (Abschnitt 2.3.1), Zugreifens (Abschnitt 2.3.2) und Synchronisierens (Abschnitt 2.3.3) systematisch untersucht.

2.3.1 Übertragen

Lernen ist ein sozialer Prozess, der an verschiedenen Orten, zu verschiedenen Zeitpunkten, aber insbesondere zwischen einer Anzahl von Personen stattfindet. In modernen Lernsituationen (z.B. der Universität) werden Lehrmaterialien in Gruppen erstellt, zwischen Gruppenteilnehmern ausgetauscht und bewertet. Hierbei wird dem Austausch von Materialien eine ähnlich wichtige Funktion zuteil wie der Möglichkeit auf Lehrmaterialien eines Gruppenteilnehmers zugreifen zu können.

Die kooperative, primäre Medienfunktion des Übertragens bezieht sich auf den Austausch von Medien zwischen Personen bzw. zwischen einer Person und einem Medium (Objekt), welches einen Austausch zu einer Person vermittelt.⁶⁴

Die Abgrenzung des Übertragens gegenüber den Medienfunktionen des Zugreifens oder Synchronisierens gelingt durch die Identifikation eines Adressaten in der Ausübung der Medienfunktion. Insofern ist die Präsentation eines Dokuments oder einer Vorlesungsfolie der Medienfunktion des Übertragens bzw. Synchronisierens zuzuordnen. Das Ablegen bzw. Zugänglichmachen eines Dokuments für eine Gruppe von Lernenden sind Beispiele für die Medienfunktion des Zugreifens. Letzteres kann jedoch ebenfalls der Medienfunktion des Übertragens zugeordnet werden, wenn das Dokument auf diesem Wege anderen Nutzern explizit übereignet wird.

Sämtliche primäre kooperative Medienfunktionen lassen sich in synchrone und asynchrone Formen differenzieren. Am offensichtlichsten wird diese Vorgehensweise bei der Medienfunktion des Synchronisierens (vgl. Abschnitt 2.3.3), jedoch auch bei der Medienfunktion des Übertragens finden sich synchrone und asynchrone Anteile. Synchrone Formen der Medienfunktion des Übertragens stellen eine enge zeitliche Kopplung von Sender und Empfänger her. Asynchrone Formen sind durch eine zeitliche Differenz der Sender-Empfänger-Beziehung gekennzeichnet.

Parallel zur Unterscheidung nach der zeitlichen Differenz zwischen Sender und Empfänger ist die Anzahl und Richtung der beteiligten Personen (Lernenden) von Interesse:

- I. Eins-zu-eins Beziehung: Werden Lehrmaterialien zwischen genau zwei Personen ausgetauscht, entspricht dies der Medienfunktion des Übertragens. Beispiele sind die Weitergabe eines Dokuments von einem Lernenden zu genau einem anderen Lernenden oder Lehrenden. Auch die elektronische Abgabe eines Übungsblatts in einen virtuellen „Briefkasten“ (Abgabefach) sind Beispiele für die Übertragung zwischen genau zwei Instanzen (vgl. Abschnitt 2.5.3). Dies ist der Medienfunktion des Übertragens zuzurechnen, da das aktive Element (vgl. Abschnitt 3.2.6)

⁶⁴ Durch Medien unterstützte Kommunikation zwischen Lernenden ist ebenfalls eine Form der Medienfunktion des Übertragens. Hier werden Informationen zwischen Kommunikationspartnern transportiert. Im Folgenden wird der Austausch von Informationen, der sich nicht an konkreten Dokumenten orientiert, als Kommunikation gesondert untersucht bzw. nicht explizit den Medienfunktionen zugerechnet.

nur eine Vermittlerrolle zwischen einer zweiten Person des Wissensraums einnimmt.

- II. Eins-zu-N Beziehung: Die Medienfunktion des Übertragens charakterisiert aber auch den Prozess der aktiven Weitergabe und der Verbreitung von Materialien an eine Gruppe von Personen. Beispiel ist hier die Weitergabe eines Dokuments an eine ganze Gruppe von Nutzern. Hierbei erhält jede Person eine individuelle Kopie.

Eng verbunden mit dem expliziten Übertragen von Materialien ist die Medienfunktion des Zugreifens. Zugriffsrechte als Strukturierungselemente kooperativen Arbeitens und Lernens spielen in diesem Zusammenhang eine besondere Bedeutung.

2.3.2 Zugreifen

Während die primäre Medienfunktion des Übertragens einen expliziten Akt der Adressierung eines Kooperationspartners (Mitlernenden) beinhaltet, beschreibt das Zugreifen Prozesse, in denen ohne Einwirkung des Erzeugers oder Bereitstellers auf Medien zugegriffen wird.

Offensichtliches Beispiel ist die Ablage eines Dokuments innerhalb eines virtuellen Raums. In Abhängigkeit der Zugriffsrechte auf das Dokument kann es gelesen und geschrieben werden, also ein Zugriff erfolgen. Ein weiteres Beispiel ist die Nutzung eines Shared Whiteboards. Nutzer können sich mit einer Sitzung verbinden und auf die Informationen des Whiteboards zugreifen.

Die Medienfunktion des Zugreifens besitzt folglich zunächst einen asynchronen Charakter. Ein Dokument wird innerhalb eines kooperativen Prozesses bereitgestellt, in Abhängigkeit von zugesprochenen Berechtigungen greifen andere Personen auf dieses zu, können es lesen oder modifizieren. Hierbei ist der Zeitpunkt des Zugriffs zunächst unbestimmt (asynchron), kann aber auch zeitgleich synchron erfolgen (z.B. im Falle eines Shared Whiteboard).

Die Anzahl der Rezipienten bei der Übertragung (Publikation) von Materialien wird über Nutzerrechte bestimmt. Nutzerrechte steuern die Sichtbarkeit und Berechtigung zu einer Ausübung der primären individuellen Medienfunktionen und sind damit wichtiger Bestandteil der Medienfunktion des Zugreifens.

Die Zuordnung von Materialien einer Arbeits- und Lernumgebung zu den Lernenden wird über Rechte vorgenommen, die für jede individuelle Medienfunktion differenziert zu spezifizieren und von der Gestalt der zugeordneten Materialien abhängig sind. Die Leseberechtigung beispielsweise kann die Sichtbarkeit von Unterlagen für Mitlernende beeinflussen. Entsprechend lässt sich die Medienfunktion des Zugreifens entlang verschiedener Berechtigungen bezogen auf die primären, individuellen Medienfunktionen differenzieren. Im kooperativen Wissensraum ist für jede der primären individuellen Medienfunktionen die Zugreifbarkeit für einen Nutzer oder eine Nutzergruppe bestimmbar. Damit liegt eine der Hauptaufgaben eines kooperationsunterstützenden Systems darin, für jede primäre individuelle Medienfunktion basierend auf Nutzerrollen und Nutzergruppen die Zugreifbarkeit zu bestimmen.

Primäre individuelle Medienfunktionen werden im kooperativen Umfeld auf gemeinsamen Dokumenten nutzbar: Erzeugen ist durch die Zugreifbarkeit auf einen gemeinsamen Ort (z.B. einen virtuellen Raum) gekennzeichnet. Dabei wird in Abhängigkeit der Schreibberechtigung für einen Ort (z.B. einen Raum) die Möglichkeit eingeräumt Objekte kreieren zu dürfen.

Das Löschen gemeinsam genutzter Materialien ist in gleicher Weise von einem Zugriffsrecht (z.B. der Schreibberechtigung oder einem expliziten Recht) abhängig.

Auch die primären, individuellen Medienfunktionen des Arrangierens und Verknüpfens lassen sich durch gegenseitigen Zugriff in einen kooperativen Handlungsraum abbilden. Der Schritt von der individuellen zur kooperativen Ausübung der Medienfunktion des Arrangierens liegt in dem *gemeinsamen* Herstellen von logischen Beziehungen durch eine Gruppe von Lernenden. Hierbei werden von einzelnen Nutzern Arrangements und Verknüpfungen zwischen Medien geschaffen, auf welche andere Lernende lesend und schreibend zugreifen. Verknüpfungen und semantische Darstellungen können kooperativ bearbeitet und erweitert werden. Hierbei besitzt eine Gruppe von Lernenden einen gemeinsamen Wahrnehmungsraum. Rollen und Benutzerrechte steuern die Ausübung der Medienfunktion des Arrangierens und Verknüpfens im kooperativen Umfeld. Die Auswahl spezieller Strategien und Vorgehensweisen zur Strukturierung gemeinsamer Prozesse des Arrangierens sind jedoch den sekundären Medienfunktionen zuzurechnen.

Insbesondere im Bereich des gegenseitigen Zugriffs auf gemeinsame Dokumente wird durch die Notwendigkeit eines gemeinsamen Wahrnehmungsraums eine weitere wesentliche Medienfunktion deutlich. Sichten auf gemeinsame Materialien, aber auch Informationen über die Kooperationspartner müssen technisch vermittelt werden. In diesem Bereich der Vermittlung verschiedener Informationen zwischen Kooperationspartnern liegt die Aufgabe der Medienfunktion des Synchronisierens.

2.3.3 Synchronisieren

Die Medienfunktion des Synchronisierens lässt sich an einer Reihe physischer Handlungen des Abgleichens bzw. Aktualisierens von Darstellungen identifizieren. Hierzu zählt zum einen der Bereich des Schaffens gemeinsamer Sichten auf Dokumente und Rückmeldung über Handlungen an diesen, zum anderen die gegenseitige Wahrnehmung zwischen Kooperationspartnern und Lernenden.

Gemeinsame Sichten auf Materialien sind eine Voraussetzung für primäre individuelle Medienfunktionen im kooperativen Umfeld. Erzeugen, Arrangieren und Verknüpfen setzen gemeinsame Sichten der Kooperationspartner (der Lernenden) auf die Materialien voraus. Eng verbunden ist eine derartige gemeinsame Sicht durch die Medienfunktion des Zugreifens. Die Sichtbarkeit, d.h. der Wahrnehmungsraum kann für einzelne Nutzer eingeschränkt bzw. gegenüber der Gruppe erweitert sein. Gemeinsame Sichten meint demgemäß nicht zwingend identische Sichten.

Die Medienfunktion des Synchronisierens schafft gemeinsame Sichten auf einzelne Dokumente oder auf Arrangements von Materialien. Je nach Anwendungsbereich sind hierzu Informationen über die Handlungen von Kooperationspartnern an den Materialien erforderlich.⁶⁵

Synchronisieren bezieht sich auf ein einzelnes gemeinsames Objekt, z.B. einen Text, der synchron von zwei Nutzern editiert wird, oder einen gemeinsam genutzten Darstellungsbereich, wie etwa einen virtuellen Wissensraum, in welchem Objekte räumlich angeordnet werden können.

⁶⁵ Die Literatur bezeichnet derartige Aktivitäts-Informationen als Activity Awareness, vgl. Abschnitt 3.1.4.

Eine Synchronisation erfolgt hierbei zeitnah, d.h. die Darstellungen, Sichten von Kooperationspartnern geben den zeitaktuellen Zustand der Materialien wieder. Ändert sich der Zustand eines Dokumentes, sind die relevanten Sichten auf dieses zu aktualisieren (synchronisieren). Die Medienfunktion des Synchronisierens erfasst in dieser Weise verschiedene Formen der Kopplung von Darstellungen. Diese können sehr eng sein, z.B. auf einer gemeinsamen Zeichenfläche, wo sogar einzelne Bewegungen von Mauszeigern synchronisiert werden, können aber auch einen losen Charakter besitzen, z.B. in Bezug auf das Shared Browsing, bei dem das „Blättern“ der einzelnen Teilnehmer einer Sitzung nicht synchronisiert werden muss.

Neben den Handlungen von Nutzern im kooperativen Wissensraum ist ihre Anwesenheit eine wichtige Information. Damit ist die Bereitstellung von Informationen der gegenseitigen Wahrnehmung eine der wesentlichen Eigenschaften der Medienfunktion des Synchronisierens.

Gegenseitige Wahrnehmung ermöglicht die Koordination bei der Ausübung von Medienfunktionen. Kooperationspartner (Lernende) müssen speziell in synchronen, computermedierten Situationen von der gegenseitigen Anwesenheit wissen. Von der gegenseitigen Wahrnehmung ist maßgeblich die Fähigkeit zur Kommunikation beeinflusst. Diese erstreckt sich speziell auf synchrone Formen der Kommunikation.

Neben einer detaillierten Analyse des theoretischen Modells individueller und kooperativer primärer Medienfunktionen ist die Ausübbarkeit der Medienfunktionen für die Nutzer ein entscheidendes Kriterium. Auch wenn eine Vielzahl von Systemen ansatzweise kooperative Medienfunktionen bereitstellen, bedeutet dies nicht, dass sämtliche Medienfunktionen für alle Nutzer prinzipiell zur Verfügung stehen.

So erlauben verschiedene Autorenumgebungen das Erstellen, Verknüpfen und Arrangieren von Unterlagen. Diese Fähigkeiten werden jedoch zumeist nur wenigen Autoren, meist den Erstellern von Lehrmaterialien, zugänglich gemacht, nicht aber den späteren Nutzern. Unter Selbstorganisation und Selbstadministration einer Lernumgebung wird in dieser Arbeit damit zum einen die prinzipielle Möglichkeit zur Ausübung primärer Medienfunktionen für *sämtliche* Nutzer verstanden, aber auch die Selbstadministration eines kooperationsunterstützenden Systems.

2.4 Selbstorganisation in den Medienfunktionen

Wesentliches Attribut kooperativer Medienfunktionen ist es, ihre Anwendbarkeit nicht auf einige wenige Lernende einzuschränken.

Gängige Praxis ist dagegen, zumeist aufgrund technischer Restriktionen, die Verwaltung einer Lernumgebung (Erzeugen neuer Lehrmaterialien, Veränderung der Struktur der Materialien) nur in die Hand von wenigen Administratoren bzw. Betreuern der Lernumgebung zu legen. Im extremen Fall des WWW beispielsweise werden Materialien von Autoren für Lernende bereitgestellt. Lernende nehmen eine passive, konsumierende Rolle ein. Es ist ihnen weder praktikabel möglich, neue Lehrmaterialien in das WWW einzubringen, noch zusätzliche Verknüpfungen (Verweise) zwischen Materialien zu erzeugen. Lediglich die Medienfunktion des Zugreifens (Laden von digitalen Medien aus dem WWW) ist eingeschränkt gegeben.⁶⁶

⁶⁶ Aus den existierenden Einschränkungen des WWW in Bezug auf passives Konsumieren der Lernenden von durch Autoren (Lehrende) bereitgestellte Materialien fordert Keil-Slawik den Schritt des vom *nicht-sequentiellen Lesen* zum *nicht-sequentiellen Schreiben*: „Digitale Medien können ihren Nutzen für neue Lernkulturen erst dann entfalten, wenn der Computer nicht mehr vorrangig als interaktives System begriffen, sondern als kooperatives Medium eingesetzt wird. Entscheidend für

Die grundlegende Idee von Hypertext [Nelson 1974] basiert dagegen auf dem Hinzufügen von neuen Texten und deren Verknüpfung mit dem bestehenden Textmaterial. Das WWW als quasi Standard der netzgestützten Bereitstellung von elektronischen Medien erfüllt diese Anforderung nur bedingt. Es besteht zwar prinzipiell die Möglichkeit, beliebige elektronische Medien einzufügen und freie Verknüpfungen zwischen abgelegten Medien zu erzeugen, dieser Prozess ist jedoch nur den Verwaltern (Administratoren) des jeweiligen WWW-Servers möglich. Herkömmlichen Nutzern (Lernenden) ist lediglich das passive Verfolgen von Verweisen gestattet. Primäre Medienfunktionen stehen zwar wie im Beispiel des Verknüpfens prinzipiell zur Verfügung, sind aber auf einen engen Nutzerkreis beschränkt.

Zudem ist der Nutzerkreis nicht konzeptionell, strukturell gewählt, sondern aufgrund technischer Restriktionen eingeschränkt vorgegeben. So ist es Administratoren eines Servers möglich, aufgrund ihres Benutzerstatus unabhängig von der Gestalt der verwalteten Objekte oder den sozialen und organisatorischen Rahmenbedingungen beliebige Medienfunktionen anzuwenden. Basierend auf den obigen Einschränkungen ergibt sich die Forderung in Bezug auf zukünftige Lernsysteme, Medienfunktionen unabhängig von den Randbedingungen ihrer Realisierung bereitzustellen.

Im Folgenden wird das mit dieser Forderung verbundene Konzept als *Selbstadministration* und *Selbstorganisation* einer Lernumgebung bezeichnet.

Selbstorganisation einer Lernumgebung umfasst damit zum einen die prinzipielle Möglichkeit der Bereitstellung von Medienfunktionen für sämtliche Teilnehmer einer Lernumgebung. Medienfunktionen wie das Erzeugen müssen allen Teilnehmern eines kooperativen Prozesses zur Verfügung stehen. Verknüpfungen und Arrangements von Lehrmaterialien sollen die mentalen Strukturierungen der einzelnen Lernenden widerspiegeln, müssen also von ihnen selbst erzeugt, angepasst und umgestaltet werden können. Zum anderen werden Fähigkeiten der Einschränkung der Ausübung der Medienfunktionen, wie das Vergeben von Zugriffsrechten, von der zugrunde liegenden Benutzer- und Gruppenstruktur, also von den sozialen Rahmenbedingungen des Lernprozesses, abhängig gemacht. Eine sinnvolle Strategie für die Einschränkung der Ausübung von Medienfunktionen (Vergabe von Zugriffsrechten) ist es beispielsweise, dem Erschaffer eines Objekts die Möglichkeit einzuräumen, Handlungen daran gezielt einzuschränken oder zu gewähren.

Die Medienfunktion des Übertragens ist entsprechend mit einer Strategie zum Umgang mit Zugriffsberechtigungen zu verknüpfen. So kann die Weitergabe eines Dokumentes als Verlust seines Besitzes interpretiert oder lediglich als temporäres Zusprechen von Zugriffsmöglichkeiten für andere Lernende verstanden werden.

Zusammenfassend lässt sich Selbstorganisation in Bezug auf die Bereitstellung von kooperativen Medienfunktionen anhand einer Reihe von Kriterien festmachen:

- Prinzipielle Nutzbarkeit der Medienfunktionen für *alle* Teilnehmer (Lernenden) eines kooperativen Prozesses.
- Flexibles Einschränken und Zusprechen von Medienfunktionen im kooperativen Umfeld.
- Unabhängigkeit der Zugriffsberechtigungen von technischen Rahmenbedingungen.

einen solchen Wandel ist, dass nicht mehr das nicht-sequentielle Lesen im Vordergrund steht, sondern die Techniken des nicht-sequentuellen Schreibens ausgenutzt werden.“ [Keil-Slawik 2001, S. 111]

- Entkopplung der Struktur aus Benutzern und Gruppen und den prinzipiellen Möglichkeiten der Nutzung primärer Medienfunktionen.

Im Vorgriff auf das in den folgenden Kapiteln vorgestellte technische Rahmenkonzept wird ein geringer Grad an Direktivität⁶⁷ der Lernumgebung angestrebt. Lernende werden nicht durch Werkzeuge oder Agenten durch die Materialien geführt. Der Lernprozess gestaltet sich in seiner Grundform als selbstorganisierter Prozess durch Gruppen von Lernenden. Natürlich existieren durchaus Komponenten einer Vorstrukturierung von Wissen und organisatorische Strukturierungen des Lernprozesses, die sich in unterschiedlichen Rollen der Lernenden und Lehrenden niederschlagen. – Ziel ist jedoch die Selbstorganisation des Lernprozesses durch die Lernenden selbst. Das technische Rahmenkonzept implementiert kooperative Medienfunktionen im Umfeld des WWW und stellt Mechanismen der Einschränkung der Medienfunktionen bereit. Die Struktur von Nutzerberechtigungen und Struktur der Lehrmaterialien wird als Teil der sekundären Medienfunktionen so wenig wie möglich von den technischen Rahmenbedingungen beeinflusst.

Im Folgenden werden eine Reihe von Szenarien des kooperativen Lernens mit Hilfe virtueller Wissensräume entwickelt. Dies geschieht aus zweierlei Gründen: Zum einen erlauben detaillierte Szenarien die begründete Bewertung einer Anzahl existierender Systeme, zum anderen dienen sie als konzeptuelle Grundlage der Ableitung von konkreten Anforderungen für das technische Rahmenkonzept kooperativer Wissensräume.

2.5 Szenarien: Lernen im kooperativen Wissensraum

Die Entwicklung umfangreicher Szenarien ist nicht nur aus den oben genannten Gründen des Verständnisses von Technik in kooperativen Lernprozessen wichtig und notwendig. Vielleicht verliert ein Entwicklerteam durch Faszination an der technischen Lösung und Machbarkeit des Neuen das eigentliche Entwicklungsziel aus den Augen. Denn nicht nur neue Systemfunktionen markieren die Verbesserung einer zukunftsweisenden Lehr- und Lerninfrastruktur, es ist das Erreichen eines Grades der Benutzbarkeit und der erfolgreichen Umsetzung von Szenarien, die die Qualität eines jeden Softwaresystems ausmachen.⁶⁸

Umso wichtiger erscheint es bei der Konzeption des vorliegenden Ansatzes, nicht einem ähnlichen Fehler zu erliegen, sondern ausgehend von einer Anzahl von Lernszenarien geeignete Funktionalitäten und Gestaltungsansätze für das System zu entwickeln. Ein derartiges Vorgehen entspricht dem „Use Cases“-Ansatz (vgl. [Booch 1994]), welcher insbesondere in modernen Methoden der Unified Modelling Language (UML) [Booch et al. 1998] angewandt wird.⁶⁹ Anhand von Anwendungsfällen der Praxis werden schrittweise Funktionen eines Systems ermittelt und in eine Objekt- und Klassen-

⁶⁷ Direktivität beschreibt das Spannungsfeld zwischen angeleitetem und selbstgesteuertem Lernen, vgl. [Wessner & Pfister 2001, S. 251ff.].

⁶⁸ Die Erfahrung, dass Entwicklerteams gerne der Faszination des technisch Machbarem erliegen und nur zögerlich innovative, an den Benutzer angepasste Lösungen entwerfen, beschreiben Nodder et al. in Bezug zur Entwicklung und der Evaluation des Microsoft Netmeeting Produktes, vgl. [Nodder et al. 1999, S. 159ff.]. Einen ähnlichen Effekt berichtet u.a. Holst in ihren Arbeiten zur Evaluation von kooperativen Lernumgebungen, vgl. [Holst 2000, S. 201ff.].

⁶⁹ Zu einem Ansatz der didaktischen Einbettung des „Use Cases“-Ansatzes in der Lehre vgl. auch [Hampel et al. 1999c].

struktur überführt. Insofern entspricht der primär inhaltlich motivierte Ansatz auch einem aus technischer Sichtweise sinnvollen Vorgehen.

In einem zweiten Schritt werden die entwickelten Szenarien und Anforderungsprofile in Abschnitt 2.7 zu einer ersten Bewertung einer Reihe von kooperationsunterstützenden Systemen genutzt. Es zeigt sich, dass existierende Systeme nur in geringem Umfang Elemente persönlicher kooperativer Strukturierbarkeit und die Selbstorganisation eines Lernprozesses unterstützen.

Insofern dienen die Szenarien zum einen einer Ableitung elementarer Anforderungen an kooperative Lernprozesse aus medientechnischer Perspektive, zum anderen einer ersten Bewertungsgrundlage vorhandener Systeme.

Das grundsätzliche Ziel der vorliegenden Arbeit, die Entwicklung eines Rahmenkonzepts und einer Architektur für die kooperative Wissensorganisation in der Lehre, lässt sich durch ein abstraktes Aufstellen und Formulieren von Anforderungen nicht lösen. Vielmehr gilt es zum einen anhand konkreter und realistischer Szenarien, grundsätzliche Tätigkeiten im Umgang mit Medien zu ermitteln und darauf basierend Möglichkeiten ihrer Unterstützung abzuleiten. Zum anderen lassen sich die in allen Formen des Lernens in der Schnittstelle zwischen Computer und klassischen Medien auftretenden Medienbrüchen erheblich reduzieren.

Damit dienen verschiedene, im Folgenden entwickelte Szenarien zur Vorlesung und Übung, Seminardurchführung und zur Projektgruppe dazu, wesentliche kooperative, aber auch individuelle Formen von computergestützten Lernprozessen zu ermitteln. Basierend auf den Erfahrungen der Arbeitsgruppe Informatik und Gesellschaft (vgl. Abschnitt 1.4) werden Unterstützungsfunktionen einer kooperativen Lernumgebung in einen derartigen Prozess eingebunden.

Im Folgenden wird zunächst das Szenario einer Seminarveranstaltung untersucht. Es bietet gute Ansatzpunkte der Selbstorganisation von Lernprozessen. Gleichzeitig enthält es Elemente aus Vorlesung und Übung und erfordert verschiedene Tätigkeiten im Umgang mit Dokumenten. In einem nächsten Schritt werden Szenarien zur Vorlesung und Übung entwickelt. Sowohl Übung als auch Vorlesung enthalten Elemente der gegenseitigen Präsentation und kooperativen Strukturierung von Lehr-/Lernunterlagen. Die Übung ist durch die Selbstorganisation des Lernprozesses und der Lehrmaterialien durch die Lernenden geprägt. Es finden sich verschiedene Formen synchroner und asynchroner Kommunikation. Das Szenario zur studentischen Projektgruppe verdeutlicht insbesondere organisatorische Anteile und die Funktion eines gemeinsamen Gedächtnisses kooperativer Wissensräume. Weiterhin werden multimediale Möglichkeiten einer kooperativen Lernumgebung untersucht. Schließlich rundet ein Szenario zur Prüfungsvorbereitung, also ein weitgehend selbstorganisierter Lernprozess ohne strukturelle Einbindung in den Vorlesungs- oder Übungsbetrieb, die Betrachtung möglicher Unterstützungsfunktionen ab.

Die Beschreibung der folgenden Szenarien zeigt eine Vielzahl von Ansatzpunkten zur Vermeidung von Medienbrüchen durch den Einsatz kooperativer Wissensräume auf. Thematischer Schwerpunkt sind verschiedene Tätigkeiten aus dem Bereich der kooperativen Wissenskonstruktion, d.h. das kooperative Zusammentragen von Referenzen, die Verwaltung von Materialien aller Art, das gegenseitige Rezensieren von Texten und die Diskussion.

Insgesamt werden durch den organisatorischen Ablauf von Seminar, Vorlesung und Übung als Präsenzveranstaltungen nur wenige wirklich synchrone Mechanismen wie synchrone Kommunikation (Chat) benötigt.

2.5.1 Seminar

Kooperative Wissensräume können insbesondere im Einsatzgebiet eines Seminars wichtige Unterstützungsfunktionen leisten.

Im Folgenden sei der klassische Ablauf eines Seminars im universitären Umfeld dargestellt, Medienfunktionen analysiert und auftretende Medienbrüche aufgezeigt. Parallel hierzu werden mögliche wichtige Unterstützungsfunktionen eines kooperationsunterstützenden Systems diskutiert. Augenmerk liegt in dem vorgestellten Ansatz auf der Betrachtung von notwendigen Medienfunktionen. Hierbei lassen sich Defizite der praktizierten Vorgehensweise isolieren und Merkmale kooperativer Wissensräume ableiten.

Es wird zur Illustration des vorgestellten Szenarios der typische Ablauf eines Seminars aus dem Themenfeld Informatik und Gesellschaft dargestellt. Hierbei sei speziell das Beispiel eines Seminars zum Thema „Open Source“, also die rechtlichen, historischen und gesellschaftlichen Fragestellungen der Entwicklung im Quelltext frei verfügbarer Software gewählt.⁷⁰

Wesentliche Elemente eines Seminars aus dieser Fachrichtung, wie die studentische Aktivität, die organisatorischen Abläufe und die Betreuungsfunktionen lassen sich ohne Einschränkung auf Seminare anderer Fachrichtungen und thematischer Ausrichtungen übertragen.

Organisatorisch lässt sich der Ablauf eines universitären Seminars entlang einer Anzahl von Phasen festmachen. Dies sind eine Anbahnungsphase, in der Studierende zur Mitarbeit in dem Seminar gewonnen werden, eine Phase der Themenfindung, eine Betreuungs- und Vorbereitungsphase, die eigentliche Präsentation der Ergebnisse, eine Nachbereitungsphase, die Aufbereitung eines Seminarbandes und eine mögliche spätere Prüfung von Seminarinhalten etwa im Rahmen einer Diplomprüfung.

Abhängig von der gewählten Form als Block oder semesterbegleitendes Seminar sind Phasen der Präsentation der Ergebnisse (Vortrag) und der Vorbereitung zeitlich aufeinanderfolgend oder monolithisch innerhalb einer Veranstaltung angeordnet.

Anbahnungs- und Anmeldungsphase

Grundsätzlich lassen sich innerhalb eines Seminars zwei grundlegende Rollen bzw. Personengruppen unterscheiden. Zum einen existieren ein oder mehrere Betreuer, wie ein Hochschullehrer oder Assistent, sowie aktive und passive Teilnehmer, also z.B. Studierende, die im Rahmen des Seminars einen Schein erwerben, oder auch nur fachlich interessierte Teilnehmer. Seminare werden in einigen Fällen interdisziplinär ausgerichtet, hierbei sind Betreuungspersonen oder Berater/Experten aus anderweitigen Fachrichtungen partiell oder permanent anwesend. Im vorliegenden Fall sei das Seminar primär am Hochschulstandort ausgerichtet und betreut, Vortragsphase und Co-Betreuung werden von einem Industriepartner begleitet.

Organisatorisch startet das Seminar mit einer Ankündigung des Oberthemas „Open Source“. Dies geschieht zum einen durch Bekanntgabe im gedruckten Vorlesungsverzeichnis, aber auch durch Ankündigungen in Form von Aushängen und eine Ankündigung im WWW unter den Seiten der Lehre der Fachgruppe.

Auf den vorgestellten Ansatz bezogen wird die Ankündigung des Seminars elektronisch innerhalb des kooperativen Wissensraums vorgenommen. Einer der Betreuer legt einen neuen, virtuellen Raum mit dem Titel des Seminarthemas „Open Source WS

⁷⁰ Das betrachtete Seminar wurde im WS00/01 ohne Unterstützung eines kooperationsunterstützenden Systems durchgeführt. Hierbei wurden insbesondere Schwachstellen und auftretende Medienbrüche sowie die Kommunikation der Teilnehmer untereinander untersucht.

2001“ an, setzt notwendige Berechtigungen für die Betreuer und Industriepartner und stellt den Ankündigungstext als Hypertextdokument in den virtuellen Seminarraum ein.⁷¹

In dem klassischen Ablauf melden sich interessierte Studierende per E-Mail oder persönlich bis zu einem festgelegten Fristtermin bei den Betreuern zur Teilnahme an dem Seminar an. Durch ein kooperationsunterstützendes System lässt sich dieser Prozess organisatorisch vereinfachen: Die Teilnehmer gelangen durch einen publizierten Verweis (aus dem Vorlesungsverzeichnis, über die Informationsseiten der Fachgruppe oder über einen Informationsraum innerhalb des kooperativen Systems) direkt in den mit dem Seminar assoziierten Raum.⁷²

Innerhalb des Raumes befindet sich ein so genanntes Seminaranmeldeskript, welches die Anmeldung zu einem Seminar automatisiert. Hier werden die für eine spätere Scheinausstellung notwendigen Daten wie Name, Vorname, Matrikelnummer, Semesteranzahl und E-Mail-Adresse erhoben und als nur für den Betreuer lesbares Dokument innerhalb des Seminarraumes abgelegt und per E-Mail an die Betreuer gesandt. Ein Betreuer trägt den Studierenden in eine spezielle Nutzergruppe „Seminarer Teilnehmer Open Source“ ein (vgl. Abbildung 5-13). Außerdem richtet er ihm einen Zugang zum kooperationsunterstützenden System ein, über den er ihn per E-Mail informiert.⁷³ Sind genügend Teilnehmer vorhanden, ist damit die Anbahnungsphase des Seminars abgeschlossen.

Vorbereitung der Themenfindung

Die zweite wichtige Phase des Seminars ist die Vorbereitungsphase der einzelnen Seminarthemen. Die Bezeichnung Vergabe der Seminarthemen ist irreführend, da der Prozess der Themenfindung ein kooperativer Prozess aus Betreuenden und Studierenden darstellt. Hierbei stehen die Neigungen und Interessen der Teilnehmer im Vordergrund. Je nach möglicher Teilnehmerzahl und Komplexität des zu bearbeitenden Themas sind Kleingruppen von bis zu drei Teilnehmern denkbar. Dies bietet sich insbesondere an, wenn Themen einen hohen experimentellen oder empirischen Praxisanteil aufweisen, z.B. die Implementierung eines Prototypen oder aufwändige Untersuchungen vorgenommen werden sollen.

Die Themenfindung beginnt mit einem ersten persönlichen Treffen von Betreuern und Seminarer Teilnehmern in der Gruppe zwecks gemeinsamen Kennenlernens und kurzer Einführung in das Themenfeld Open Source. Hierzu wird ein Termin über den virtuellen Seminarraum in einer E-Mail an die Teilnehmer und auf einem elektronischen Nachrichtenbrett bekannt gegeben.

Idealerweise findet das erste Treffen in einem elektronischen Seminarraum statt. Eine derartige lernförderliche Infrastruktur erlaubt den Referenten und Betreuern in einer Vorstellung des Themenfeldes Open Source, fließend Referenzen und Materialien im Netz mit einzubeziehen. Dies bedeutet zum einen das Präsentieren und gemeinsame

⁷¹ Innerhalb des Ankündigungsdokumentes befinden sich Verweise auf weiterführende Netzadressen, z.B. auf bekannte Projekte zu Open Source. Innerhalb des Ankündigungstextes befindet sich des Weiteren ein Verweis auf die E-Mail-Adresse der Betreuer mit einem Hinweis auf Bereitschaft zu weiteren Auskünften.

⁷² Hierzu genügt ein herkömmlicher Browser, es muss kein spezieller Client geladen werden.

⁷³ Der erste Teil des Kennwortes wird unmittelbar über das Anmeldeformular mitgeteilt, die zweite Hälfte des Kennwortes wird per E-Mail zugesandt. In einem ersten Schritt befindet sich der neuangemeldete Benutzer lediglich in der Nutzergruppe der Studierenden des Fachbereichs.

Diskutieren von Netzseiten (Shared Browsing), aber auch eine Präsentation von Dokumenten (wissenschaftliche Artikel, Whitepaper etc.). Auf das Beispiel eines Open Source-Seminars bezogen präsentieren die Betreuer Folien zur Einstimmung in den Themenbereich und zu den Rahmenbedingungen des Seminars. Es wird ein kurzer Zeitablauf gegeben, Kriterien zur Scheinvergabe erläutert und die Einbettung der Veranstaltung in den Studienverlauf dargestellt. Hierzu dient eine kurze Präsentation, die in dem virtuellen Raum zur Veranstaltung abgelegt ist.

Die Präsentation kann mittels Beamer oder über eine Kopplung von Browsern auf den einzelnen Plätzen der Teilnehmer (innerhalb des kooperativen Wissensraums) erfolgen. Bei einer idealen Vorgehensweise werden die Folien über Beamer präsentiert, während es den Teilnehmern möglich ist eigene Notizen zu den Folien anzufertigen. (Dies geschieht durch Nutzung von Annotationen innerhalb des kooperationsunterstützenden Systems oder durch Anfertigen von eigenen Protokolldateien und anschließendem Ablegen innerhalb eines virtuellen Raums.)

Ziel einer derartigen Vorgehensweise ist es, möglichst schon in dieser frühen Phase die Teilnehmer in die virtuelle Lernumgebung einzuarbeiten und ihnen den konkreten Bezug zu bereits innerhalb des virtuellen Open Source-Raumes abgelegten Materialien zu verdeutlichen. So werden während der Einführung in den Themenbereich Open-Source wichtige Abschnitte und Ideen aus dem Artikel „The Cathedral and the Bazar“ [Raymond 2001] zitiert. Innerhalb des virtuellen Open Source-Raumes befindet sich ein Verweis zu dem entsprechenden Artikel im Netz. Anschließend werden von den Betreuern die wesentlichen Themenbereiche, anhand derer sich das Seminar ausrichten soll, erläutert. Dies können z.B. sein: „Open Source – geschichtliche Einordnung“, „Open Source Philosophien“, „Rechtliche Aspekte, Lizenzen und Patente, Nutzung im kommerziellen und privaten Bereich“ und „Open Source-Entwicklung von Softwareprojekten“. Zu jedem dieser Themengebiete sind im Vorfeld im virtuellen Open Source-Raum entsprechende Ordner mit jeweils einer kurzen Beschreibung des Themenfeldes abgelegt. Zudem wurden Verweise auf im Netz verfügbare Artikel oder Webseiten hinzugefügt.

Themenfindung

Ziel des nächsten Schrittes ist es, eine sinnvolle Zahl von Seminarthemen in Abhängigkeit der Anzahl der Seminarteilnehmer und ihrer Neigungen zu finden. Hierzu werden Themenvorschläge von den einzelnen Seminarteilnehmern in der Gruppe diskutiert. Innerhalb des virtuellen Open Source-Raumes werden Vorschläge für mögliche Themen zunächst ohne feste Zuordnung zu einem Themenfeld stichpunktartig notiert. Denkbar wäre u.a. die Anfertigung einer semantischen Karte zum Themengebiet Open Source, auf welcher die wesentlichen Verweise und Artikel zu den Themenbereichen geordnet abgelegt sind. Diese semantische Orientierungshilfe dient den Seminarteilnehmern auch zur Navigation.

Nachfolgend werden die einzelnen Themen den zuvor festgelegten Themengebieten des Seminars zugeordnet. Basierend auf dieser Zuordnung von Themenstellungen zu Themengebieten lässt sich eine erste Verteilung der Seminarthemen vornehmen.

Innerhalb des virtuellen Open Source-Raumes schreiben die Teilnehmer ihren Namen hinter maximal drei Themen, die ihren Wünschen und Vorstellungen entsprechen. Auf diese Weise werden die Themen Einzelpersonen oder kleineren Gruppen von Teilnehmern (bis maximal drei Studierende) zugeordnet. Bei der Zuordnung an eine Gruppe ist es evtl. notwendig, das Thema in eine Anzahl von Unterthemenstellungen aufzuteilen.

Zu den einzelnen Seminarthemen wird eine Übersicht erstellt, in der die Aufgabenstellung und zu bearbeitende Punkte des Seminarthemas schrittweise verfeinert dargestellt werden.

Bezogen auf die anzufertigende Ausarbeitung legen die Betreuer eine Musterausarbeitung zu einem beliebigen Seminarthema ab sowie eine Dokumentenvorlage für den zu erstellenden Seminarbeitrag, die z.B. eine Spezifikation des Zitierstils und Formatierungsrichtlinien enthält. Der spätere Seminarband wird als Papierversion jedem Seminarteilnehmer bereitgestellt und als elektronische Version im Open Source-Raum öffentlich zugänglich gemacht.

Vorbereitungsphase der Seminarthemen

Die Phase der Themenfindung geht unmittelbar in die Vorbereitungsphase der Seminarthemen durch die Seminarteilnehmer über. Diese Phase ist typischerweise nur durch wenige Treffen geprägt, in denen alle Seminarteilnehmer und Betreuer präsent sind. Organisatorische Absprachen werden per E-Mail oder durch Notizen im virtuellen Raum vorgenommen. Bei einer Präzisierung der Seminarthemen erscheint die Kleingruppenkonstellation sinnvoll, da zumeist recht komplexe Fragestellungen diskutiert werden müssen, die sich speziell auf ein Unterthema beziehen. Hierzu ist die Präsenz anderer Seminarteilnehmer nicht bei jedem Treffen erforderlich.

Aus der Sicht der Betreuenden erlaubt die Fixierung und Strukturierung von für das Seminar relevanten Informationen auf den virtuellen Seminarraum den Vorteil einer leichten Koordinierbarkeit verwandter Seminarthemen, die insbesondere dem Gesamtüberblick zu Gute kommt. Probleme der Teilnehmer lassen sich leicht durch Bereitstellen von weiterführender Literatur oder ergänzenden Verweisen lösen. Dies kann asynchron geschehen, indem die Teilnehmer eine kurze Nachricht über ein Problem mit Bitte um Hilfestellung innerhalb des virtuellen Raums hinterlassen.

Als wesentlicher Schritt zur Seminaerausarbeitung werden die schon im Rahmen der Themenfindung vorhandenen Dokumente zu den einzelnen Seminarthemen schrittweise ausgearbeitet. Dies kann durch Hinzufügen von Stichpunkten und kurzen Textpassagen erfolgen, aber auch durch Ergänzung von Zitaten und Referenzen auf weiterführende Artikel. Im Falle von praktischen Themen werden erste Entwürfe und Prototypen abgelegt. So sammelt z.B. eine Teilnehmerin zu ihrem Thema „Open Source – Der Gedanke, und wie er entstand“ eine Reihe von Definitionen zum Verständnis von Open Source und legt sie mit der jeweiligen Referenz auf den Ursprungsartikel ab. Eine erste Gliederung ihrer Ausarbeitung wird in einem eigenen Dokument schrittweise verfeinert und durch Stichpunkte ergänzt. Auf diese Weise wird die Ausrichtung der Ausarbeitung gegenüber anderen Seminarteilnehmern zugespitzt und die eigentliche Botschaft ihrer Arbeit konkretisiert. Bei mehreren Treffen mit ihrem Betreuer werden recherchierte Quellen und Gedanken diskutiert und ihre Relevanz für die Aussage des Seminarbeitrages bewertet.

Zwei weitere Studenten, die zusammen das Thema der Anfertigung eines exemplarischen Open Source-Auftrittes des Softwareprojektes „Fahrplanauskunft“ bearbeiten, stellen in einer Anzahl von kurzen Treffen ihre ersten prototypischen Realisierungen vor, anhand derer wichtige Designentscheidungen und Möglichkeiten der Realisierung diskutiert werden können.

Bewusst werden auch Lösungen, die im Rahmen des Seminars nicht weiter verfolgt werden oder sich als nicht sinnvoll oder gangbar herausgestellt haben, im virtuellen Raum zum Seminar mit abgelegt. Diese können im Rahmen des späteren Vortrages oder

speziell der anschließenden Diskussion wichtige Impulse liefern oder getroffene Designentscheidungen verdeutlichen.

In der Vorbereitung der eigentlichen Seminarphase, also der Vorträge zu den bearbeiteten Seminarthemen, sind von den Teilnehmern Entwürfe der Vorträge (als Foliensätze) und der Seminararbeit mit den Betreuern abzustimmen. Die Folien und der Entwurf der Ausarbeitung werden im virtuellen Seminarraum abgelegt. Die Betreuer kommentieren diese mit angefügten Annotationen, zusätzlichen Texten oder aber in einem gemeinsamen Gespräch.

Insbesondere die Phase der konkreten Vorbereitung des Seminarthemas, welche einen hohen Betreuungs- und Abstimmungsaufwand erfordert, lässt sich durch den Ansatz kooperativer Wissensräume ausgezeichnet unterstützen. In der klassischen Vorgehensweise werden meist Entwürfe von Präsentationen und verschiedene Dokumente zwischen den Betreuern und Seminarteilnehmern in Papierform oder evtl. per E-Mail ausgetauscht. Hierzu ist eine explizite Absprache notwendig.

Als gravierender Nachteil klassischer Abläufe eines Seminars findet zudem meist lediglich eine Kommunikation zwischen den Betreuern und den Teilnehmern des Seminars statt, weniger zwischen den einzelnen Seminarteilnehmern. Der Ansatz eines gemeinsamen Treffpunktes und Datenraumes für das Seminar erleichtert es, sich unkompliziert einen Überblick über die Themenstellungen und den Stand der Aktivitäten der anderen Seminarteilnehmer zu verschaffen und ggf. Anregungen zu geben oder Fragen zu beantworten. Eine derartige Form des gemeinsamen Problemlösens ist in rein auf Papier basierten Szenarien, ohne Medium der gemeinsamen Wissenskonstruktion lediglich in Präsenzsituationen möglich. Hierzu müssen sich die Teilnehmer persönlich zu Gesprächen treffen und gegenseitig über ihre Aktivitäten informieren.

Natürlich kann ein kooperationsunterstützendes System keine realen Treffen ersetzen, es lassen sich jedoch Prozesse des kooperativen Wissenserwerbs unterstützen und speziell die Kommunikation der Seminarteilnehmer untereinander anregen. Da es sich bei den Seminarteilnehmern erfahrungsgemäß meist um eine Gruppe von Studierenden handelt, die sich zuvor gar nicht oder nur flüchtig kannten, ist es zudem umso wichtiger, einen gemeinsamen Treffpunkt zu schaffen, der als Keimzelle von Zusammenarbeit, gegenseitigem Problemlösen und Kommunikation in der Gruppe dienen kann. Hierdurch lassen sich die Beiträge der Teilnehmer insbesondere schon in der Vorbereitungsphase gut aufeinander abstimmen, d.h. Redundanz vermeiden und gegenseitige Bezüge erkennen. Diese gegenseitigen Bezüge können sich z.B. schon in ersten Verweisen innerhalb der stichpunktartigen Seminararbeiten aufeinander ausdrücken.

Hilfreich zum gegenseitigen Austausch ist sicherlich die Angabe der E-Mail-Adresse, der Telefonnummer und die Übersicht über die an den einzelnen Themen arbeitenden Personen. Auf diese Weise fällt die Kontaktaufnahme sowohl über den virtuellen Seminarraum als auch über andere Kommunikationsformen entsprechend leicht.

Resultierend aus der Zielrichtung eines Seminars, möglichst unkompliziert kooperativ arbeiten zu können, sind innerhalb des virtuellen Seminarraumes nur wenige Berechtigungen zu setzen (Einschränkung der Medienfunktion des Zugreifens).

Prinzipiell besitzen alle Seminarteilnehmer identische Berechtigungen: Gegenseitige Leseberechtigung und die Möglichkeit der gegenseitigen Annotation von Dokumenten ist obligatorisch. Die Schreibberechtigung aller Teilnehmer erstreckt sich auf den gesamten kooperativen Seminarraum. Gegenseitiges Löschen von Dokumenten ist nicht möglich. Betreuer besitzen vollständige Berechtigungen für die gesamte Umgebung.

Begleitend zu der eigentlichen Seminararbeit, welche ein gewöhnliches Textverarbeitungsdokument darstellt, sind innerhalb eines Ordners Verweisobjekte auf externe, im WWW zu findende Quellen anzulegen sowie mögliche Beispiele etc. abzulegen.

Synchrones kooperatives Editieren einer Seminararbeit über einen kooperativen Editor (simultanes Schreiben an einem Dokument) macht aus Sicht der Bedienbarkeit derartiger Werkzeuge wenig Sinn. Realistischer erscheint es, im Falle der Bearbeitung eines Seminarthemas durch eine Kleingruppe, die Seminararbeit in Unterabschnitte aufzuteilen, wobei einzelne Abschnitte asynchron bearbeitet und jeweils als eigenständige Dokumente in das kooperationsunterstützende System abgelegt werden. Diese Teildokumente werden erst in einer letzten Version zu einem gemeinsamen Dokument zusammengeführt.

Ein derartiges Verfahren birgt zwar Nachteile in sich, wie z.B. das umständliche gegenseitige aufeinander Beziehen, ist jedoch unabhängig von den verwendeten Werkzeugen zur Textverarbeitung und Zugangsstrukturen der einzelnen Teilnehmer. Weiterhin anzumerken ist, dass die Notwendigkeit, eine klassische Seminararbeit in Form einer wissenschaftlichen Arbeit anzufertigen, als elementarer Bestandteil des Anspruches der Vermittlung von Basiswissen eines Seminars zu werten ist und daher nicht durch z.B. Anfertigung einer Liste von Verweisen oder Stichpunkten abgelöst werden kann.

Präsentation der Seminarthemen

Die eigentliche Präsentation der Seminarbeiträge ist als Präsenzphase konzipiert. Hier sind innerhalb kurzer Vorträge die wesentlichen Ergebnisse der geleisteten Arbeit darzubieten und zur Diskussion zu stellen. In der klassischen Blockveranstaltung finden sich die Teilnehmer des Seminars über einen festen Zeitraum zusammen. Jede Gruppe hat ihr Seminarthema als Teil eines zuvor bekannt gegebenen Programms zu präsentieren. Diskussionen werden meist von den Betreuern moderiert.

Um eine gewisse Sicherheit der Verständlichkeit und Qualität der vorgestellten Beiträge zu gewährleisten, ist es allgemein üblich, dass ein erster Entwurf der Seminarfolien mit den Betreuern abzustimmen ist. Ein derartiger Prozess ist mittels kooperativer Wissensräume effizient zu leisten.

Hierzu sind die Folien einige Tage vor der eigentlichen Seminarphase innerhalb eines virtuellen Raums abzulegen. Betreuer können nun den Foliensatz kommentieren und gegebenenfalls modifizieren. Insbesondere zur Vorbereitung eines realen Treffens lassen sich Betreuer und Vortragende auf diese Weise leicht auf einen einheitlichen Gesprächsstand bringen. In Zusammenhang mit ebenfalls in Stichpunkten oder schon abschnittsweise formulierten Seminararbeiten ist es den Betreuern zudem möglich festzustellen, inwieweit der Vortrag die Kernpunkte der Seminararbeit vorstellt und geeignet in die Thematik einführt bzw. keine Ergebnisse aus anderen Seminarvorträgen vorwegnimmt oder zu sehr auf diesen aufbaut.⁷⁴

In einem nächsten Schritt legen die Betreuer in Absprache mit den Seminarteilnehmern den Ablauf der eigentlichen Vortragsphase als Blockveranstaltung fest. Hierzu

⁷⁴ Derartige Überschneidungen lassen sich gut in der späteren Ablaufplanung des Vortragsprogramms berücksichtigen.

werden zunächst innerhalb von dem virtuellen Seminarraum Referenzen auf die jeweiligen Vortragsfolien erzeugt und anschließend thematisch sortiert.

Die Phase der Seminarvorträge findet wiederum in einem elektronischen Seminarraum (vgl. Abschnitt 1.4) statt. Vorträge werden durch die innerhalb des kooperativen Wissensraums abgelegten Folien gehalten. Es sind verschiedene Szenarien der Unterstützung dieser Vortragsphase denkbar:

Die Folien werden durch die Vortragenden direkt auf die Bildschirme aller Seminarteilnehmer projiziert. Dies kann durch ein didaktisches Netzwerk, d.h. durch eine elektronische Kopplung aller Bildschirminhalte geschehen oder durch eine softwareseitige Synchronisation der Projektionswerkzeuge innerhalb einer Session. Letzteres ließe sich durch einen Shared Browsing-Ansatz realisieren. Hierzu werden die Folien als HTML- oder XML-Version innerhalb der kooperationsunterstützenden Umgebung abgelegt. Ein Shared Browser erlaubt das gemeinsame Betrachten von Internetseiten (in diesem Fall der Folien), wobei die Steuerung bei einer festgelegten Person liegt. Hierzu begeben sich sämtliche Zuhörer und der Vortragende in einen gemeinsamen virtuellen Vortragsraum.

Der Vortragsraum dient der Anbahnung der gemeinsamen Sitzung. Innerhalb des virtuellen Raumes lässt sich eine beliebige Anzahl von Shared Browsern starten. Je nach Grad der Kopplung der Bildschirminhalte sind verschiedene Formen der Partizipation der Zuhörerschaft denkbar. So ließe sich z.B. nur der Folienübergang synchronisieren, die Navigation innerhalb der Folien bzw. Möglichkeiten der Annotation bleiben bestehen. Zudem ist bei einem Shared Browsing-Ansatz nur ein Fenster und nicht der gesamte Bildschirm der Teilnehmer synchronisiert, was den Zuhörern persönliche Notizen parallel zum eigentlichen Vortrag ermöglicht. Insofern ist ein Ansatz der softwareseitigen Synchronisation der Folienübergänge nicht nur für Szenarien der Telepräsentation von Vorträgen, also in virtuellen Treffen, interessant, sondern macht auch für die Unterstützung von face-to-face Situationen Sinn. Als nachteilig erweist sich bei einer Kopplung der Bildschirme die Notwendigkeit der Fixierung der Aufmerksamkeit der Teilnehmer. Hier erweist es sich ohne eine zentrale Projektionseinrichtung als schwierig, z.B. die Aufmerksamkeit der Teilnehmer auf einen zentralen Punkt auf einer Folie zu lenken. Hier ist der Einsatz von so genannten Telepointern sinnvoll, d.h. eines für alle Teilnehmer sichtbaren gemeinsamen Zeigers.

Als zweite, alternative Präsentationstechnik werden die Bildschirme der Teilnehmer nicht mit dem Bildschirminhalt des Vortragenden synchronisiert, sondern eine zentrale Projektionseinrichtung genutzt. Die Bildschirme der Zuhörer sind unabhängig von dem Fortgang der Präsentation nutzbar. Durch den Einsatz eines kooperationsunterstützenden Systems können sich die Teilnehmer begleitend zum Vortrag innerhalb des virtuellen Raumes zum Seminar bewegen und verschiedene Möglichkeiten des Systems nutzen. Dies kann z.B. das Anfertigen persönlicher Notizen zu den eigentlichen Folien sein, wozu die Folien von den Vortragenden zuvor als netzgestützte HTML- oder XML-Version abgelegt wurden und nun annotierbar sind. Auch der Zugriff auf ergänzende Netzseiten oder die vollständige Seminararbeit ist durchaus sinnvoll.

Ersteres bietet sich für eine Seminarsituation wie beschrieben an. So kann z.B. eine Diskussionsstrukturierung gewählt werden, in der nach einem Vortrag die Zuhörer die wesentlichen Fragen, aber auch wesentliche Ergebnisse zu dem jeweiligen Seminarvortrag zusammenfassen müssen. Gleichzeitig können die von den Zuhörern zu einem Seminarvortrag angefertigten Stichpunkte helfen eine nachfolgende Diskussion besser zu strukturieren. Dokumente, die Notizen der Seminarteilnehmer zu den jeweiligen Semi-

narvorträgen enthalten, können entweder in dem zentralen virtuellen Seminarraum abgelegt werden, besser jedoch sind sie in spezielle Unterordner zu den entsprechenden Seminarvorträgen einzufügen. Hierbei kann die Leseberechtigung so gesetzt werden, dass es sich um wirkliche private Notizen (Lesen nur für den Autor) oder um öffentliche, für alle Teilnehmer des Seminars zu lesende Notizen handelt.

Eine Diskussion begleitend oder im Anschluss an jeden einzelnen Seminarvortrag ist als wesentliches Ergebnis der Seminarvorbereitung zu charakterisieren. Die Qualität der Präsentation schlägt sich unmittelbar in dem Niveau der anschließenden Diskussion nieder. Wesentliches Ziel ist daher, die Seminarteilnehmer zu ermuntern, gezielt aber konstruktiv Kritik über Vortragsstil und die erarbeiteten Ergebnisse zu üben. Hierbei kommen während des Seminarvortrages angefertigte Notizen und Annotationen der Folien zugute. Die Vortragenden profitieren von Anregungen und Verbesserungsvorschlägen an ihrem Vortrag unmittelbar für die endgültige Fassung der schriftlichen Seminararbeit, daher müssen auch sie die Kernideen der Diskussion fixieren. Bei einigen Seminarthemen bietet es sich an, als Ergebnis der Diskussion eine Einschätzung zu einem Sachverhalt zu finden und dieses als Ertrag der Seminarveranstaltung mit in den virtuellen Seminarraum aufzunehmen. Dies können z.B. persönliche Erfahrungen der Seminarteilnehmer sein, die sich den Ergebnissen einer vorgestellten Studie gegenüberstellen lassen.

Auf das vorgestellte Thema Open Source bezogen ließe sich nach einem Seminarvortrag eine Erhebung zur Nutzung von Open Source-Produkten im privaten Umfeld durchführen, wie groß der Anteil der Seminarteilnehmer ist, die zumindest ein Open Source-Produkt regelmäßig auf ihrem PC einsetzen. Die Ergebnisse dieser Miniaturbefragung werden innerhalb des virtuellen Seminarraumes abgelegt und könnten fortlaufend aktualisiert werden.⁷⁵

Kooperative Seminararbeit

Die abschließende Phase der kooperativen Anfertigung eines Seminarbandes zum Seminarthema bildet ein typisches Beispiel für die Reduzierung von Medienbrüchen durch den Einsatz von kooperationsunterstützenden Werkzeugen. Der Seminarband enthält die Seminararbeiten der einzelnen Seminarteilnehmer und dient als Basis für nachfolgende Seminare mit ähnlichen Themenstellungen oder auch als Prüfungsvorbereitung für einzelne Seminarteilnehmer, die auch in späteren Phasen des Studiums das Seminar als Prüfungsthema in ihre Diplomprüfung mit einfließen lassen können. Deshalb dient der Seminarband als dauerhafte Fixierung wesentlicher Seminarergebnisse und sollte, wenn nur elektronisch verfügbar, über einen längeren Zeitraum unverändert archiviert werden.

In der klassischen Vorgehensweise wird der Seminarband aus den einzelnen Seminararbeiten zentral durch einen Betreuer des Seminars angefertigt. Meist durch ein Vorwort oder eine Einführung in die Thematik ergänzt, enthält er die Seminararbeiten in thematisch sinnvoller Reihenfolge. Die Erstellung des Seminarbandes ist meist aufwändig, je nach Art und Weise der Herstellung und seiner redaktionellen und typografischen Qualität. (Ein einfaches Fotokopieren von Teilbeiträgen ist zumeist mit

⁷⁵ Hierbei könnte ein Diskussionsbrett zum Thema „Open Source“ mit dieser speziellen Fragestellung innerhalb des kooperationsunterstützenden Systems eingerichtet werden, um auch externe Nutzer über einen längeren Zeitraum in die Diskussion mit einzubeziehen. Die hieraus gewonnenen Erkenntnisse können z.B. in die Seminararbeit oder in auf das Seminar aufsetzende Forschungen mit einfließen, z.B. in Form einer Diplomarbeit.

erheblich weniger Aufwand verbunden als die Einbindung der Teilbeiträge in ein zentrales Dokument.) Zudem sind meist mehrere Überarbeitungen der Beiträge zwischen dem Herausgeber des Seminarbandes und den Autoren der Teilbeiträge auszutauschen. Ohne Unterstützung eines geeigneten Systems sind dazu mehrfach Datenträger oder elektronisch Daten per E-Mail auszuwechseln. Hieraus resultiert ein großes Rationalisierungspotenzial für den Einsatz einer kooperationsunterstützenden Umgebung.

Wie schon in vorherigen Phasen des Seminars dargestellt, dient der virtuelle Seminarraum als Treffpunkt der Teilnehmer im Netz, aber auch als zentraler Ablagepunkt aller gemeinsamen Dokumente. Er besitzt die Funktion eines gemeinsamen Gedächtnisses des Seminars. Neben den Vortragsfolien und verschiedenen Übersichtsdokumenten zum Gesamtthema enthält er für die einzelnen Seminarthemen einen Unterordner bzw. Container, in welchem schon vorbereitend zur Seminararbeit und zum Seminarvortrag Verweise auf ergänzende Informationen im Netz oder weiterführende Dokumente abgelegt werden können. Diese von den Seminarteilnehmern selbst gepflegten Bereiche können im Schritt der gemeinsamen Erstellung einer Seminararbeit zur späteren dauerhaften Semindokumentation ausgebaut werden.

Hierzu legt der Betreuer einen neuen Container „Seminarband Open Source 2001“ an, der später allgemein öffentlich sein wird. Die von den Teilnehmern an der ursprünglichen Position bis zu einem festgelegten Datum abgelegten Seminararbeiten werden von den Betreuern gelesen und kommentiert.

Entsprechen die Seminararbeiten den festgelegten Richtlinien, werden die Teilnehmer des Seminars aufgefordert, ihre Seminararbeiten mit dem Titel des entsprechenden Themas versehen in den neu angelegten Container zu kopieren. (Als Format bietet sich eine HTML-, XML- oder auch PDF-Version der jeweiligen Seminararbeit an. Die Quelldokumente bleiben später nur für die Teilnehmer des Seminars lesbar in einem anderen Bereich des virtuellen Seminarraums gespeichert.)

Ergänzende Informationen zu den jeweiligen Seminararbeiten lassen sich ebenfalls zur öffentlichen Dokumentation des Seminars ablegen. Hierbei sind jedoch etwaige Urheberrechte von ergänzenden Artikeln etc. zu beachten. Referenzen auf im Netz verfügbare Artikel können die Literaturangaben in den Seminararbeiten ergänzen.

Während der Seminarphase von anderen Seminarteilnehmern ergänzte Annotationen, Bemerkungen und die Notizen der Seminarteilnehmer sollten für die Allgemeinheit nicht frei zugänglich sein, deshalb bleibt dieser Teil des virtuellen Seminarraumes für die Öffentlichkeit gesperrt.

Seminarteilnehmer sollten in jedem Fall die Schreibberechtigung an den von ihnen angefertigten Unterlagen behalten, d.h. es sollte für sie auch weiterhin die Möglichkeit bestehen, eigene Ergänzungen vorzunehmen und Anmerkungen anzufügen. Der eigentliche elektronische Seminarband sollte jedoch nur für die Betreuer (die Administratoren des virtuellen Seminarraumes) mit Schreibberechtigung versehen dauerhaft archiviert sein. Es kann zudem sinnvoll sein, den virtuellen Seminarband in einen speziellen, mehrere themenverwandte Seminare umfassenden Raum zu verschieben.

2.5.2 Vorlesung

Im folgenden Abschnitt soll der Ablauf einer Vorlesung am Beispiel der „Softwareergonomie“ mit begleitenden Übungen bzw. Tutorien verdeutlicht werden. Der klassische Übungsbetrieb weist eine Reihe von Medienbrüchen auf, die sich ebenfalls durch den Einsatz kooperativer Wissensräume reduzieren lassen. Ein Schwerpunkt liegt auf der

Identifikation genau dieser existierenden Übergänge der verwendeten Medien, welche die Ansatzstellen für kooperationsunterstützende Systeme bilden.

Zurzeit findet in Paderborn eine Unterstützung der Vorlesung durch lernförderliche Infrastrukturen wie elektronische Hörsäle und Übungsräume sowie durch Ablage von Materialien in einem Dokumentenmanagementsystem (Hyperwave) statt. Der Ablauf der Vorlesung und Übung und die identifizierten Ansatzpunkte von kooperativen Wissensräumen basieren stark auf den in den letzten Jahren gesammelten Erfahrungen in einer Reihe derartiger Lehrveranstaltungen (vgl. Abschnitt 1.4).

Eine Vorlesung zur Softwareergonomie umfasst Vorlesungs- und Übungsstunden. Es sind von den Studierenden Übungsaufgaben zu den Vorlesungen zu bearbeiten, die in den darauf folgenden Übungsveranstaltungen besprochen und vorgestellt werden.

Organisatorisch untergliedert sich eine Veranstaltung zur Softwareergonomie demgemäß in den Vorlesungs-, Übungs- und Klausurteil, wobei die einzelnen Abschnitte stark ineinander verzahnt sind.

Vorlesungsabschnitt

Jede einzelne Vorlesung besitzt ein spezielles Thema, z.B. „Kriterien zur Reduzierung erzwungener Sequentialität“ oder „EU-Bildschirmrichtlinie“. Die Vorlesung orientiert sich an den vom Dozenten angefertigten Foliensätzen. Diese werden den Studierenden ein bis zwei Wochen vor der Vorlesung innerhalb kooperativer Wissensräume zur Verfügung gestellt. Die Vorlesungsfolien geben den Inhalt der vorgestellten Themengebiete in stichpunktartiger Form wieder, insofern sind die Folien ohne ergänzende Erklärungen und Kommentare des Dozenten nicht selbsterklärend. Sie dienen als Gedankenstützen und Orientierungshilfen zur Vorlesung und illustrieren die Ausführungen des Dozenten.

Werden beispielsweise Kriterien für eine softwareergonomisch gute Maskengestaltung vorgestellt, führen die Folien einige Kriterien zur Gestaltung von Bildschirmmasken ein, sie besitzen jedoch ohne illustrierende Beispiele nur wenig Aussagekraft. Deshalb haben eine Vielzahl von Beispielen und Illustrationen einen hohen Stellenwert bei der Vorlesungsgestaltung. Etwa ein Drittel der Vorlesungszeit umfasst die Besprechung von Beispielen als diskursivem Prozess zwischen dem Dozenten und den Studierenden. Dementsprechend ist die Ausrichtung der Paderborner Vorlesungen zur Softwareergonomie als reine Präsenzveranstaltung zu erklären, die zwar weitläufige Vortragspassagen der klassischen Vorlesung enthält, aber einen hohen Anteil an interaktiven Komponenten zwischen Dozenten und Studierenden und den Studierenden untereinander besitzt.

Beispiel: Softwareergonomie

Mit Hilfe kooperativer Wissensräume können Studierende die Vorlesung sowohl physisch anwesend als auch synchron von jedem mit einem Internetzugang versehenen Ort verfolgen.

Konzeptionell stellt eine kooperative Lernumgebung einen virtuellen Vorlesungsraum zum Themengebiet der Vorlesung Softwareergonomie bereit. Der virtuelle Raum wird von den Betreuern der Vorlesung (dem Dozenten und beteiligten Übungsgruppenleitern) gemeinsam administriert. Zu Übungsgruppen und studentischen Lerngruppen zur Vorlesung können jeweils eigene von den Betreuern der Übungsgruppe oder auch von den Studierenden selbst verwaltete Räume existieren.

Wie zurzeit schon praktiziert werden innerhalb dieses virtuellen Vorlesungsraumes neben den von dem Dozenten aktuell vorgetragenen Folienpräsentationen auch Hintergrundinformationen zu der Vorlesung, d.h. sämtliche von dem Referenten hinzugezo-

gene Artikel, Texte, Internetseiten, Grafiken und Programme bereitgestellt. Dieser ständig verfügbare Bestand erlaubt es, z.B. während der Vorlesung auf eine Anzahl von Wissensquellen zurückzugreifen und diese aktiv zu illustrieren bzw. auf Fragen und Anregungen durch die Studierenden direkt Bezug zu nehmen bzw. Modifikationen an den Materialien vorzunehmen.

Schreibberechtigung für die Vorlesungsinhalte besitzen der Dozent und die Übungsgruppenleiter der Vorlesung. Der Zugang zu den Vorlesungsinhalten ist nicht beschränkt, Berechtigungen zur Annotation und Ablage persönlicher Notizen ist jedoch nicht möglich, diese Funktionalitäten sind auf die Bereiche der jeweiligen Übungsgruppen beschränkt.

Die Vorlesungsfolien sind mittels eines Präsentationsprogramms erstellt und in eine HTML- oder XML-Darstellung umgewandelt. Beide Versionen werden von dem Dozenten oder den beteiligten Übungsgruppenbetreuern in dem virtuellen Vortragsraum abgelegt.

Präsentation und Annotation in der Vorlesung

Eine Präsentation der Vorlesungsfolien erfolgt, wie schon unter den Szenarien zum Seminarvortrag dargestellt, mittels einer zentralen Projektionseinrichtung (Beamer) innerhalb des elektronischen Hörsaals, durch Reproduktion der Bildschirminhalte des Dozenten auf die Bildschirme der Zuhörer (didaktisches Netzwerk) oder über einen synchronen Shared Browser, d.h. eine sitzungsbasierte, softwareseitige Kopplung von Fenstern und Bildschirminhalten. Letzteres erlaubt ähnlich einer zentralen Projektionseinrichtung die entkoppelte Arbeit der Hörer begleitend zur Vorlesung und damit das Anfertigen von Notizen oder Anmerkungen an den Vorlesungsfolien innerhalb des virtuellen Vorlesungsraumes.⁷⁶

Die Teilnehmer der Vorlesung haben sich die Folien in einen persönlichen, virtuellen Raum oder in einen gemeinsamen Übungsraum kopiert, um sich während der Vorlesung Notizen und Anmerkungen zu den Folien machen zu können. Hierbei sind verschiedene Techniken zum Anfertigen von persönlichen Notizen an die Vorlesungsfolien denkbar: Anmerkungen können in einer separaten Textdatei zu den jeweiligen Folien abgelegt werden, es kann aber auch eine in das verwendete Präsentationsprogramm integrierte Funktion zum Anfertigen von Notizen an Folien verwendet werden oder es können innerhalb der kooperationsunterstützenden Umgebung integrierte Mechanismen der Annotation genutzt werden.

In der einfachsten Variante, in der von den Studierenden persönliche Anmerkungen in eigenen Textdateien zu dem jeweiligen Foliensatz abgelegt werden, sind kaum spezielle Mechanismen in eine kooperationsunterstützende Umgebung zu integrieren. Begleitend zur Vorlesung wird ein beliebiger Texteditor genutzt, um Anmerkungen und Ergänzungen zu den vorgetragenen Vorlesungsinhalten in einer Reihe von Textdateien innerhalb der persönlichen Bereiche oder dem zur Vorlesung eingerichteten virtuellen Übungsraum zu erstellen. Ein derartiges Vorgehen erlaubt jedoch wenig direkte Verknüpfungsmöglichkeiten zwischen den Folien und den persönlichen Anmerkungen. Anmerkungen werden zunächst als von den Folien unabhängige Objekte verwaltet und

⁷⁶ Der bisher praktizierte Einsatz eines didaktischen Netzwerkes in Zusammenhang mit einer zentralen Projektionseinrichtung erlaubt je nach Vorlesungsphase Bildschirminhalte einzelner Hörer auf die zentrale Projektionseinrichtung zu verschalten oder auch nur den Bildschirm des Vortragenden auf die der Zuhörenden zu replizieren. Hierdurch ergibt sich ähnlich wie in einer sitzungsbasierten Software-Lösung eine große Freiheit in dem Kopplungsgrad der Bildschirme.

sind lediglich durch ihre Position innerhalb der kooperativen Umgebung (z.B. innerhalb des jeweiligen Containers) mit den Vorlesungsfolien verknüpft.

Als zweite Alternative können in den Anwendungsprogrammen integrierte Funktionen zum Anfertigen von Notizen an Folien genutzt werden. (In Microsoft Powerpoint können zu jeder Folie persönliche Notizen angefügt werden.) Hierzu wird von den Hörern der Vorlesung der jeweils aktuell vorgetragene Foliensatz auf den persönlichen Arbeitsplatz geladen, mit dem zugehörigen Präsentationsprogramm bearbeitet und anschließend erneut abgelegt.

Als letzte Alternative können Notizen schließlich als Annotation direkt an die Folien geheftet werden. Derartige Anmerkungen werden direkt mit den entsprechenden Folienelementen verwaltet (z.B. zusammen mit den Folienelementen in einem Ordner oder Container gespeichert bzw. in dem so genannten Environment, der Umgebung eines Objektes, abgelegt, vgl. Abschnitt 3.2.1). Derartige Annotationen bleiben z.B. bei einem Verschieben eines Objektes erhalten und können mit eigenen Benutzerrechten versehen werden. Hierdurch kann die Lesbarkeit von Anmerkungen auf eine Gruppe von Studierenden ausgeweitet und so verstärkt kooperatives Lernen unterstützt werden. (Vgl. hierzu die Szenarien zum Übungsbetrieb in Abschnitt 2.5.3.)

Insgesamt lässt sich bei sämtlichen Vorgehensweisen das Ziel einer Reduzierung eines Medienbruches zwischen gewöhnlich elektronisch präsentierten Folien und nicht mit in ein kooperationsunterstützendes System integrierte handschriftliche Anmerkungen an ausgedruckte Folien oder handschriftlich angefertigte Mitschriften erzielen.

Praxisphasen innerhalb der Vorlesung – kooperative ergonomische Bewertung

An dem Szenario einer Vorlesung zur Maskengestaltung lassen sich weitere grundlegende Kooperationstechniken innerhalb kooperativer Wissensräume illustrieren. Hierbei spielen konkrete Beispiele der Praxis eine wichtige Rolle.

Die Vorlesung beginnt mit der Einführung der sieben Kriterien zur Dialoggestaltung nach EN ISO 9241 Teil 10 (Aufgabenangemessenheit, Selbstbeschreibungsfähigkeit, Steuerbarkeit, Erwartungskonformität, Fehlerrobustheit, Individualisierbarkeit, Lernförderlichkeit). Zu jedem der Kriterien wird auf den zugehörigen Folien eine kurze, beispielhafte Definition in Form einer Maske dargestellt, die das jeweilige Kriterium verletzt. Anschließend wird ein Beispiel für ein Softwaresystem vorgestellt, das von den Studierenden nach den EU-Richtlinien zur Dialoggestaltung zu bewerten ist.

Ein geeignetes Softwaresystem ist ein netzgestütztes Formular zur „Pizza-Bestellung“ über das Internet. Es lässt sich durch eine innerhalb der kooperativen Lernumgebung abgelegte Internetadresse aufrufen und praktisch erproben. Die verwendeten Netzseiten wurden anonymisiert und als Beispiel abgelegt.⁷⁷

Das Beispiel verletzt einige grundlegenden Kriterien der Aufgabenangemessenheit, indem z.B. bei einem Bestellvorgang jeweils nur eine Pizza bestellt werden kann, der Bestellvorgang ist in eine Vielzahl von unnötigen Dialogen unterteilt, die jeweils vollständig ausgefüllt werden müssen, um schließlich auf dem letzten Formular den eigentlichen Bestellvorgang auszulösen. Die Eingabe der Zahlungsweise ist sehr umständlich realisiert und die Eingabe einer Kreditkartennummer unübersichtlich gestaltet. Insgesamt lassen sich für jedes der Kriterien Beispiele innerhalb des Systems finden, die von den Softwareentwicklern bzw. Gestaltern der Netzseite nicht beachtet wurden.

⁷⁷ Es handelt sich um eine HTML-/ Javascript-Applikation, die in Anlehnung an einen existierenden Paderborner Pizzadienst erstellt wurde.

Neben dem (bis auf den eigentlichen Bestellvorgang) vollständig funktionierenden und damit von den Studierenden praktisch erprobten Beispiel des zu bewertenden Softwaresystems finden sich von jedem der relevanten Bildschirmmasken Bildschirmkopien als Grafikobjekte, die in einer späteren Besprechung, z.B. auf einem elektronischen Whiteboard kommentiert und grafisch bearbeitet werden können.

Die praktische Überprüfung der Kriterien beginnt mit einer Erprobungsphase des Softwaresystems durch die Studierenden. In kleinen Gruppen zu zwei bis drei Personen werden einige typische Anwendungsfälle erprobt, jeder versucht die von ihm bevorzugte Pizza zu bestellen. Schnell ergeben sich offensichtliche Probleme und „Unschönheiten“ in der Bedienung, die in dem nächsten Schritt zu notieren sind.

Hierzu kann wiederum eine einfache Textdatei dienen (für jede Gruppe getrennt), in der die Probleme in Stichworten zu den einzelnen Masken notiert werden. Alternativ können die vorbereiteten Bildschirmkopien (Grafikdateien, Screenshots) der Masken verwendet werden, die auf einem in die Lernumgebung integrierten Whiteboard kommentiert werden können. (Hierzu ist zuvor eine Kopie der jeweiligen Bildschirmkopien in einen freien Arbeitsbereich des kooperationsunterstützenden Systems zu verschieben, so dass die einzelnen Gruppen parallel und unabhängig voneinander Notizen und Modifikationen vornehmen können.)

Ein elektronisches Whiteboard eignet sich zu einem derartigen Prozess insbesondere durch die Fähigkeit, verschiedenfarbige Grafikobjekte über den zu kritisierenden Screenshot zu legen. So kann z.B. eine Schaltfläche (Button), die sich an einer ungünstigen Position befindet, rot eingekreist und mit einem Pfeil in Richtung der typischen, gängigen Position markiert werden. Notwendige und wünschenswerte Aktionen sind z.B.:

- Markieren von Bereichen durch Invertieren, Einkreisen oder Platzieren einer transparenten Farbfläche über das hervorzuhebende Objekt. Textobjekte können durch Unterstreichen oder andere Auszeichnungen der Schrift hervorgehoben werden.
- Setzen von Markierungen in Form von kleinen Pfeilen oder sonstigen Grafikobjekten in räumliche Nähe der hervorzuhebenden Objekte.
- Platzieren von verschiedenfarbigen Annotationen auf die Zeichenfläche. Diese können evtl. als „Tooltips“ ausgeführt sein, d.h. beim „Überfahren“ mit der Maus werden die vollständigen Anmerkungen angezeigt.
- Zeichnen von Grafikobjekten: Kreise, Polygonzüge, Freihandzeichnen.
- Platzieren von speziellen grafischen Symbolen – im einfachsten Fall verschiedenfarbiger „+“ oder „-“ Zeichen, um eine einfache Form von Wertungen ausdrücken zu können. Dieses Konzept ist erweiterbar auf die Anordnung einer Anzahl von speziell auf die Lehrmaterialien oder Vorlesungsinhalte abgestimmten Symbole (z.B. Symbole, die für einzelne softwareergonomische Kriterien stehen).
- Überdecken, „Ausradieren“ eines Bereiches.
- Einbringen von weiteren Grafikelementen auf die Zeichenfläche. Hierdurch lässt sich z.B. ein zweiter Gestaltungsvorschlag oder Teile von diesem mit der Gesamtgestaltung in Bezug setzen.
- Ablegen verschiedener Zustände (Snapshots) des Inhaltes eines Whiteboards. Hierdurch lassen sich z.B. verschiedene Probleme auf einer Bildschirmmaske durch unterschiedliche Lösungen oder Beispiele illustrieren.

Sämtliche Funktionen auf einem Whiteboard sind innerhalb einer Sitzung gekoppelt, d.h. arbeitet eine Gruppe von Studierenden auf ein und demselben Whiteboard, sind für jeden Teilnehmer die Aktionen der anderen Mitglieder einer Sitzung sichtbar. Die Aktionen der jeweiligen Teilnehmer sind durch verschiedenfarbige Mauszeiger erkennbar, es sind jedoch eine ganze Anzahl anderer Formen der gegenseitigen Wahrnehmung denkbar: z.B. Versehen der platzierten Grafikobjekte mit einem Namenskürzel oder speziellen, jedem Teilnehmer zugeordneten Symbolen.

Studierende können alle Formen von digitalen Medien, wie Grafiken, Verweise und Dokumente mittels Drag&Drop-Techniken auf diese synchrone, d.h. von allen Beteiligten an einem virtuellen Ort zur gleichen Zeit sichtbare Arbeitsfläche, übertragen. Angelegt als objektorientiertes Whiteboard, d.h. eine Form der Arbeitsfläche, welche auf reinen Objekten agiert, erlaubt ein derartiger Ansatz, digitale Medien in eine synchrone Zusammenarbeit zu überführen. So führt z.B. das Ziehen eines Verweises aus einem Browser unmittelbar zu einem Link-Objekt innerhalb eines dem Whiteboard zugeordneten Raumes.

Auf das obige Beispiel bezogen ist in einem zweiten Schritt das Pizzabestellsystem anhand der sieben Kriterien zur Dialoggestaltung konkret zu bewerten. Es ist für jedes Kriterium kurz festzustellen, inwieweit das untersuchte Softwaresystem dem Kriterium genügt oder es in verschiedener Hinsicht verletzt.

Anschließend werden die Ergebnisse aus den einzelnen Untergruppen zusammengetragen und gegenseitig bewertet. Der Dozent übernimmt die Moderation und fragt gezielt nach Problemen und einer ersten Einschätzung der Ergonomie des zu bewertenden Softwaresystems. Die geschilderten Erfahrungen der Studierenden werden von ihm (oder einem anderen Studierenden bzw. Assistenten) kurz in Stichpunkten festgehalten.

Alternativ kann der Prozess der softwareergonomischen Bewertung durch die gemeinsame Zeichenfläche weiter strukturiert werden. So ist z.B. ein Anwendungsfall denkbar, in dem jede Gruppe von Studierenden oder auch jeder Studierende – in Abhängigkeit der Anzahl der Lernenden – eine Anzahl von Grafiksymbolen (z.B. verschiedenfarbige „+“ oder „-“ Symbole) auf Bildschirmkopien des zu bewertenden Systems platzieren soll. Auf diese Weise ergibt sich ein Prozess des kooperativen Bewertens eines Softwareentwurfs durch eine Anzahl von Lernenden – das resultierende Bild ergibt unmittelbar, und direkt erkennbar, einen Meinungsspiegel der anwesenden Gruppe.

Auf diese Weise identifizierte, softwareergonomische Probleme und Schwachstellen können bereits dem jeweiligen Kriterium zugeordnet werden. So zeigt sich beispielsweise, dass durch eine willkürliche Auflistung der zu bestellenden Sorten von Pizza und keinerlei erklärende Texte Kriterien der Selbstbeschreibungsfähigkeit und Strukturiertheit gleichermaßen verletzt sind.

Derartige Schwachstellen werden gezielt auf den vorhandenen Bildschirmkopien wie beschrieben markiert und den Vorlesungsunterlagen beigelegt. Es entsteht in dieser Form ein gut dokumentiertes, von den Studierenden und dem Dozenten kooperativ bearbeitetes Beispiel zum Vorlesungsthema, welches sich unmittelbar neben den Vorlesungsfolien im virtuellen Vorlesungsraum befindet und als zukünftige Referenz und Bezugspunkt für Übungsaufgaben etc. dienen kann.⁷⁸ Dies ist wichtig für die Entwicklung eines tief greifenden Verständnisses des präsentierten Vorlesungsinhaltes, da sich die

⁷⁸ Nach Ende der Veranstaltung können z.B. im Rahmen einer Übung erarbeitete Beispiele in einen virtuellen „Stöberraum“ übernommen werden, der späteren Generationen von Lernenden als Fundgrube und Sammelsurium für gut dokumentierte Beispiele dienen kann.

Studierenden in ihren Diskussionen im Rahmen einer Übung oder der späteren Prüfungsvorbereitung zu den theoretischen Konzepten auf derartige, gemeinsam bearbeitete Beispiele beziehen können.

Eigenständiger softwareergonomischer Entwurf

Innerhalb einer zweiten Vorlesungseinheit oder als Teil der Übung könnte nun ein eigenständiger softwareergonomischer Entwurf der Benutzerschnittstelle für ein netzgestütztes Bestellsystem angefertigt werden. Studierende verwenden z.B. ein grafisches Benutzerschnittstellenwerkzeug (GUI-Builder), um die wesentlichen Dialoge und Masken zu gestalten und evtl. als Prototyp innerhalb ihres virtuellen Übungsraumes abzulegen.

Anschließend lassen sich derartige Gestaltungsentwürfe gegenseitig vorstellen und rezensieren. Studierende erproben verschiedene Bedienmetaphern, z.B. die eines Warenkorb, mit dem man durch eine Art virtuelle Selbstbedienungspizzeria geht, oder die Metapher eines Interviews oder eines telefonischen Bestellsystems, indem der Kunde von einem virtuellen Pizzabäcker zu seinen Wünschen befragt wird. Verschiedene Alternativen der Realisierung werden mit den grafischen Benutzerschnittstellenwerkzeugen in Form von kleineren Prototypen skizziert und als Vorlesungsdokumentation abgelegt.

Zu ihrer weiteren Vertiefung werden die zuvor eingeführten Kriterien anhand einer Reihe von positiven Einzelbeispielen illustriert. Dies kann durch eine Ansammlung von Entwürfen innerhalb der Lernumgebung geschehen, die etwa durch eine Übersichtskarte (semantische Karte) verknüpft sind. Zu jedem Kriterium findet sich ein positives Beispiel, z.B. zum Kriterium Erwartungskonformität die überarbeitete Maske des oben erwähnten Pizzabestellsystems, welche als klassische Speisekarte gestaltet den Erwartungen eines unbedarften Nutzers entspricht.

Wege durch Vorlesungsmaterialien

Als weitere Alternative zu in der Vorlesung erarbeiteten Beispielen lässt sich ein Weg, ein „Trail“, durch im Internet vorhandene Beispiele ablegen. Studierende bewegen sich anhand von kommentierten Verweisen von Internetadresse zu Internetadresse und können anhand dieser Beispiele aus der Lebenswirklichkeit gewisse softwareergonomische Kriterien von Internetapplikationen erkunden.

Wichtiges Kriterium eines derartigen Weges ist, dass er eine Art grafische Repräsentation besitzt, die zur Navigation genutzt werden kann. Trails sollten einfach erstellt und modifiziert werden können (dies sollte kooperativ, also gemeinsam von den Mitgliedern eines virtuellen Raumes geschehen können). Ebenso sollten Trails als echtes kooperatives Objekt zwischen Studierenden und Dozenten ausgetauscht und weitergegeben werden können. In dieser Form kann insbesondere das Anfertigen von eigenen Trails, einem dokumentierten Weg durch im Netz verfügbare Beispiele, als Teil der Übung zur Vorlesung dienen. Trails können des Weiteren von den Dozenten zur Illustration eines Sachverhaltes genutzt werden, also im Rahmen der Vorlesung vorgestellt werden. Hierbei werden ähnlich dem kooperativen Betrachten (Browsen) Internetseiten in einer definierten Reihenfolge aufgerufen und von dem Dozenten kommentiert.

Dozenten nehmen während der Vorlesung jedoch nicht nur Bezug auf Vorlesungsfolien. Beispiele und Verweise zu externen Netzseiten, auch Artikel, Gesetze und Normen, also sämtliche weiterführenden Texte, können innerhalb der kooperativen Lernumgebung begleitend zur Vorlesung hinzugezogen werden. So kann beispielsweise eine Passage einer im Netz verfügbaren Diplomarbeit zur softwareergonomischen Bewertung

eines größeren Softwaresystems im Rahmen der Vorlesung gezeigt werden und anhand von enthaltenen Grafiken und Illustrationen ein konkreter Praxisbezug zu den vorgestellten theoretischen Konzepten hergestellt werden. Kopien der Arbeiten oder Verweise auf digitale Materialien werden hierzu in den virtuellen Vorlesungsraum aufgenommen. Des Weiteren können z.B. auch wissenschaftliche Artikel zur Vertiefung bestimmter Vorlesungsinhalte im Rahmen der Vorlesung referenziert oder besprochen werden und so den Studierenden Ansatzpunkte für die spätere, eigenständige Lösung von Übungen oder realen Entwurfsproblemen gegeben werden.

Einbindung räumlich entfernter Lernender

Als zweiter interessanter Effekt können trotz der weiterhin starken Prägung der Vorlesung als Präsenzveranstaltung durch entsprechende Kommunikationsmechanismen wie textuellem Chat sowie Audio- und Video-Chat auch räumlich distanzierte Teilnehmer aktiv in das Vorlesungsgeschehen mit eingebunden werden. Über das Internet (im virtuellen Vorlesungsraum) der Vorlesung beiwohnende Studierende oder Dozenten können in dieser Form Rückfragen stellen oder Kommentare ablegen bzw. Dokumente dem Vorlesungsraum zukommen lassen.

2.5.3 Übung

Eng verknüpft mit den vorgestellten Szenarien und Anwendungsbeispielen von kooperationsunterstützenden Systemen in Seminar und Vorlesung ist die Übung als kooperativer, interaktiver Prozess zwischen den Studierenden und den Betreuern bzw. Dozenten der Lehrveranstaltung zu beschreiben. Sie dient maßgeblich der Vertiefung und weiteren Verständnisbildung der in der Vorlesung vorgestellten Sachverhalte und ist von ihrer Grundausrichtung her durch Aktivitäten der Studierenden geprägt. Übungen finden in Gruppen von zehn bis maximal vierzig Personen statt und sind daher von der Gruppengröße her prädestiniert für den Einsatz von lernförderlichen Infrastrukturen wie kooperativen Wissensräumen, elektronischen Hörsälen und elektronischen Seminarräumen. Anwendungsszenarien für einen Übungsbetrieb sind durch den fließenden Wechsel aus Gruppen- und Individualarbeitsphasen geprägt.⁷⁹

Organisatorisch untergliedert sich die Übung zu einer Vorlesung in an festen Terminen stattfindenden realen Treffen der jeweiligen Übungsgruppen und Phasen, in denen die Studierenden vorbereitend eine Anzahl von Übungsaufgaben zu bearbeiten haben. Bearbeitete Übungsaufgaben können bei den Betreuern bis zu einer gewissen wöchentlichen Frist abgegeben werden und dienen der Überprüfung des persönlichen Kenntnisstandes und dem Erwerb einer Anzahl von Bonuspunkten für die später stattfindende Klausur. Übungsblätter können von einer Gruppe von zwei bis drei Studierenden gemeinsam bearbeitet werden. Diese Studierenden bilden typischerweise eine gemeinsame Lerngruppe, welche oft auch nach der eigentlichen Vorlesungsveranstaltung bestehen bleibt und sich gemeinsam auf Prüfungen vorbereitet.

Insofern ist es eine wesentliche Komponente eines kooperationsunterstützenden Systems, diese Struktur aus Lerngruppen und Übungsgruppen in das System abzubilden. Wie schon unter den Szenarien zur Vorlesung angedeutet, benötigen Studierende persönliche Arbeitsbereiche, um Notizen, Anmerkungen oder Beispiele dauerhaft in Bezug

⁷⁹ Auch in der schon heute praktizierten Vorgehensweise einer Übung zur Softwareergonomie werden Studierende aktiv in die Gestaltung der Übung mit eingebunden. Hierbei wird insbesondere die Zusammenarbeit zwischen den Studierenden gefördert.

zu den Vortragsunterlagen setzen zu können. Dies kann in den Arbeitsbereichen der studentischen Lerngruppen geschehen. Insofern kann der Ansatz gewählt werden, dass jede Übungsgruppe einen gemeinsamen virtuellen Übungsraum erhält, zusätzlich aber auch jede studentische Lerngruppe von zwei bis drei Studierenden einen eigenen Raum besitzt.

Anmeldung

Organisatorisch beginnt eine Vorlesungsveranstaltung zu Beginn des Semesters mit dem Aushang der Übungstermine, die entsprechend der Anzahl der an einer Vorlesung teilnehmenden Studierenden innerhalb der kooperationsunterstützenden Umgebung zusammen mit den Namen der betreuenden Assistenten in Form eines Informationsbrettes zur Vorlesung bekannt gegeben werden.

Ähnlich der beschriebenen, netzgestützten Zusammensetzung einer Seminargruppe können sich Studierende innerhalb einer Anzahl von virtuellen Übungsgruppenräumen zur Teilnahme an der entsprechenden Übung anmelden. Hierzu sind Anmelde-mechanismen bereitzustellen, die Studierenden, welche schon eine Kennung innerhalb des Systems besitzen, Zugang zu den Bereichen der entsprechenden Übungsgruppe gewähren oder einen neuen Zugang einrichten.

Die Anmeldung findet in einem zweistufigen Prozess statt, Studierende können sich so lange wie eine Übungsgruppe genügend Plätze bietet in dem entsprechenden virtuellen Übungsgruppenraum der Vorlesung anmelden. Der Betreuer der entsprechenden Übungsgruppe wird von dem System automatisch benachrichtigt.

Anschließend nimmt der Übungsgruppenleiter die neuen Teilnehmer in die jeweiligen, zuvor eingerichteten Benutzergruppen auf. Hierbei lässt sich gegebenenfalls eine Umverteilung der Übungsgruppenstärke vornehmen. Innerhalb des Anmeldeprozesses werden alle weiteren, notwendigen Daten der Studierenden wie Name und Matrikelnummer erfasst, um z.B. später die Bonuspunkte der abgegebenen Übungen zu verwalten (vgl. Abbildung 5-13).

Einrichten – Selbstadministration persönlicher Arbeitsbereiche

Als wesentliches Element der aktiven Arbeit der Studierenden mit den Materialien zu Vorlesung und Übung dienen persönliche Bereiche der Studierenden, welche in der Rechte- und Gruppenstruktur der sozialen Struktur der Lerngruppen entsprechen. In kooperationsunterstützenden Systemen, welche eine zentrale Administration besitzen, kann es mitunter schwierig und aufwändig sein, für kleinere Gruppen von Studierenden eigene Arbeitsbereiche einzurichten und zudem fortlaufend neue Mitglieder einer Lerngruppe einzutragen (und damit Berechtigungen zum Zugriff auf den Arbeitsbereich zu setzen) oder entsprechend Mitglieder zu streichen oder in andere Bereiche zu verschieben.⁸⁰

In einem Szenario zu dem verfolgten Ansatz können Arbeitsbereiche und speziell kleinere Nutzergruppen der Studierenden eigenständig eingerichtet und verwaltet werden. Das zugrunde liegende Konzept der Selbstadministration erlaubt prinzipiell das Einrichten von neuen, virtuellen Räumen und gleichzeitig das Erzeugen neuer Nutzer-

⁸⁰ Es wurden in der Praxis mit dem Hyperwave-System, welches eine rein zentralisierte Administration besitzt, Erfahrungen in dieser Richtung gemacht. So wurden in Lehrveranstaltungen mit einer größeren Anzahl von Teilnehmern nur Gruppenzugänge, d.h. eine gemeinsame Nutzerkennung und Zugangskennwort vergeben.

gruppen und das Hinzufügen von Personen zu diesen durch die Studierenden selbst (vgl. Abbildung 3-6).

Anschließend kann sowohl die gesamte Pflege und Verwaltung der neuen Nutzergruppe als auch des zugehörigen Dokumentenraumes dezentral von den Lernenden selbst vorgenommen werden.⁸¹

Nach der Anmeldung der Studierenden in einer Übungsgruppe sind folgende Schritte zum Einrichten von bzw. Arbeiten mit persönlichen, virtuellen Lerngruppen und Bereichen denkbar: Bei den Leitern der jeweiligen Übungsgruppen ist die Einrichtung einer geeigneten Benutzergruppe und eines Arbeitsbereiches zu einer neuen Lerngruppe zur Vorlesung zu erfragen. Der Übungsgruppenleiter stimmt dem zu und übergibt die Administration, d.h. die Berechtigungen neuer Benutzer zu der Nutzergruppe hinzuzufügen oder Berechtigungen für den Zugriff auf die in dem Arbeitsbereich vorhandenen Objekte zu vergeben, an einen Studierenden oder gleichberechtigt an alle Teilnehmer der neuen Lerngruppe. Es erscheint sinnvoll, bei kleinen, gleichberechtigten Gruppen von Studierenden auch die Berechtigungen möglichst identisch zu setzen. Insgesamt sind zur kooperativen Arbeit in kleinen Lerngruppen kaum gezielt zu spezifizierende, sich gegenseitig einschränkende Berechtigungen notwendig.

Eine einfache Regel der Zusammenarbeit könnte sein, dass jeder auf jedes Dokument der Lerngruppe vollständigen Zugriff erhält. Schreibender Zugriff auf die Materialien der Lerngruppe ist den Mitgliedern anderer Gruppen, wenn nicht explizit zugestanden, verwehrt. Es handelt sich primär um abgeschlossene, persönliche Arbeitsbereiche von Gruppen von Studierenden, insofern können diese nach Wunsch der Studierenden auch vollständig abgeschlossen sein.

In einer Hierarchie aus Benutzergruppen finden sich Lerngruppen unterhalb der zur Vorlesung eingerichteten Übungsgruppen wieder. Hierbei ist es nicht zwingend erforderlich, dass sämtliche Teilnehmer einer Lerngruppe an ein und derselben Übungsgruppe teilnehmen. Organisatorisch vereinfacht die Zuordnung einer Lerngruppe zu einer Übungsgruppe den Zugriff auf im Rahmen der Übung erarbeitete Materialien. Durch Mitgliedschaft der Studierenden in einer Übungsgruppe (auf die Lernumgebung bezogen) lassen sich Berechtigungen global für Übungsgruppen setzen und auf diese Weise Rechte gezielt an die beteiligten Lerngruppen vererben.

Lerngruppen erhalten einen persönlichen, virtuellen Lerngruppenraum, der von der Lerngruppe selbst eingerichtet werden kann. In diesem Raum lassen sich Dokumente oder beliebige andere Materialien zur Vorlesung zusammentragen. Materialien der Lerngruppen sind persistent, d.h. bleiben genau wie die Bereiche der Übungsgruppen durchgängig bestehen. Innerhalb eines virtuellen Lerngruppenraumes können sich Teilnehmer der entsprechenden Lerngruppe virtuell treffen. Mechanismen der gegenseitigen Wahrnehmung (Awareness) erlauben z.B. festzustellen, ob sich ein Teilnehmer in einem virtuellen Lerngruppenraum aufhält. Kommunikationsmechanismen wie Chat sind in ihrer Reichweite auf den virtuellen Raum beschränkt und unterstützen auf diese Weise den Meinungsaustausch der Studierenden einer Lerngruppe untereinander.

Geht eine Lehrveranstaltung mit dem Abschluss des Semesters zu Ende, können die Studierenden selbst über das weitere Bestehen des virtuellen Lerngruppenraumes ent-

⁸¹ Die grundlegenden Mechanismen zu einer Selbstadministration von Benutzergruppen und virtuellen Räumen ist die Weitergabe von Berechtigungen. Hierbei kann das Recht zur Administration einer Benutzergruppe z.B. von einem Betreuer einer Lehrveranstaltung an einen Studierenden weitergegeben werden und damit die Administrationsrechte und Pflichten an die Studierenden delegiert werden (vgl. Abschnitt 3.1.5).

scheiden. Es kann durchaus sinnvoll sein, dass er für die weitere gemeinsame Prüfungsvorbereitung bestehen bleibt oder die Materialien in die privaten Arbeitsbereiche der Studierenden verschoben oder dupliziert werden.

Virtueller Übungsraum – kooperatives Lösen von Übungsaufgaben

Organisatorische Informationen werden innerhalb des virtuellen Vorlesungsraumes und innerhalb der Seminarräume durch eigene Informationsbretter bekannt gegeben. Auf solchen elektronischen Nachrichtenbrettern können z.B. Terminverschiebungen von Übungen oder Klausuren etc. bekannt gegeben werden.

Innerhalb des Übungsgruppenraumes helfen themenbezogene Nachrichtenbretter offene Fragen zur Vorlesung oder zu den Übungsaufgaben zu klären. Ein Studierender kann z.B. eine kurze inhaltliche Frage formulieren wie: „Auf welche Elemente innerhalb eines der Beispiele zum Fakturierungssystem aus der Vorlesung bezieht sich die Selbstbeschreibungsfähigkeit genau und inwieweit grenzt sie sich von der Erwartungskonformität ab?“ Hierdurch initiiert er einen Diskussionsprozess in Form von sich aufeinander beziehenden Antworten, Bemerkungen oder auch weiteren Fragen.

Nachrichtenbretter können von einem Übungsgruppenleiter moderiert werden oder auch ganz in der Verantwortung der Studierenden liegen. Typischerweise ist der Inhalt eines Nachrichtenbrettes oder Diskussionsforums nur für die Vorlesungs- bzw. Übungszeit über das Semester persistent – wichtige Erklärungen können jedoch auch über diesen Zeitraum hinaus bestehen bleiben.

Der Ablauf von Übungsveranstaltungen orientiert sich meist an Übungsaufgaben. Diese werden wöchentlich von den Dozenten und Übungsgruppenleitern erstellt und innerhalb des virtuellen Übungsgruppenraumes abgelegt. Hierzu eignet sich ein in HTML oder XML gespeichertes Textdokument genauso wie eine Grafik oder ein lauffähiges Java-Programm. Ziel der Übung ist es, die Inhalte der Vorlesung zu vertiefen und Beispiele aus der Praxis aufzuzeigen. Übungsaufgaben sind vom Umfang und Schwierigkeitsgrad darauf ausgerichtet, in kleineren Gruppen gelöst zu werden. Da die Übungsaufgaben den eigentlichen Stoff für die spätere Übung bilden, besitzen sie einen hohen durch die Studierenden zu erbringenden kreativen Eigenanteil – es existiert meist auch keine alleinige, eindeutige Lösung. In der Übung werden die Übungsaufgaben besprochen, verschiedene Lösungsvarianten gegenübergestellt und diskutiert.

Ein Beispiel für solch eine Übungsaufgabe ist z.B. ein Text, der den typischen Verfahrensablauf einer Mitgliederverwaltung eines Sportvereins beschreibt. Es werden die erfassten Stammdaten der Mitglieder und ihre Speicherung innerhalb einer elektronischen Mitgliederverwaltung beschrieben. Zu festen Daten werden Teile der Mitgliederdaten an eine lokale Sparkasse zum automatischen Bankeinzug der Mitgliedsbeiträge elektronisch übermittelt. Zur Organisation von gemeinsamen Feierlichkeiten sind auch persönliche Daten wie Geburtsdatum und Beruf gespeichert. Auch führt der Verein Statistiken über Vereinswettkämpfe.

Die Aufgabe für die Teilnehmer der Vorlesung besteht nun darin, den beschriebenen Umgang mit den Daten der Vereinsmitglieder unter Datenschutzgesichtspunkten zu bewerten. Es sind potenzielle Verletzungen des Datenschutzgesetzes und zulässige Verfahrensweisen aufzuzeigen und gegebenenfalls ein modifizierter zulässiger Umgang mit den Mitgliederdaten zu entwerfen. Als Lösung der Übungsaufgabe sind die Probleme und Schwachstellen des beschriebenen Verfahrens mit Verweisen auf die verletzten Abschnitte des zu den Vorlesungsunterlagen abgelegten Datenschutzgesetzes zu versehen und ein modifizierter neuer Umgang mit den Mitgliederdaten zu beschreiben.

Die Studierenden diskutieren in kleinen Gruppen den Umgang mit den Mitgliederdaten und entwerfen gemeinsam eine zulässige und sinnvolle Lösung. Übungsaufgaben und die zugehörigen Lösungen werden zunächst in den virtuellen Lerngruppenräumen abgelegt. Der virtuelle Raum ist hierbei gemeinsamer Ort der Ablage von Entwürfen und Varianten der Lösung.

Auf das obige Beispiel bezogen treffen sich drei Studierende, die die Aufgabe gemeinsam bearbeiten, und diskutieren mögliche Verletzungen des Datenschutzgesetzes der beschriebenen Vereinsverwaltung. Es steht ihnen ein PC mit Zugang zu den Vorlesungsunterlagen zur Verfügung. Das Datenschutzgesetz ist innerhalb des virtuellen Vorlesungsraumes abgelegt. Es ist als Hypertext in einzelne Absätze unterteilt, wobei jeder Absatz eine Aussage oder einen Sachverhalt repräsentiert.

Der zu analysierende Text wird von den Studierenden abschnittsweise durchgearbeitet und analysiert. Findet sich eine Stelle, die einer Verletzung des Datenschutzgesetzes entspricht, wird ein Verweis auf den entsprechenden Absatz des Datenschutzgesetzes eingefügt. Dies kann eine Annotation sein, die einen Verweis enthält oder den entsprechenden Abschnitt des Datenschutzgesetzes zitiert. Ein in der Gruppe strittiger Punkt lässt sich mit Hilfe des Datenschutzgesetzes allein nicht klären, auch eine Reihe von Beispielen, von denen eines in der Vorlesung ausführlich besprochen wurde und unter den Vorlesungsunterlagen abgelegt ist, geben keinen Aufschluss über die Auslegung des strittigen Paragraphen. Eine Studentin aus der Gruppe vermerkt das Problem am Diskussionsbrett innerhalb ihres virtuellen Übungsgruppenraumes. Kann sie ihr Problem nicht mit Hilfe anderer Studierender oder des Übungsgruppenleiters lösen, werden in jedem Fall die auf dem Diskussionsbrett angesprochenen Fragestellungen in der Übungsveranstaltung dieser Lerngruppe diskutiert.

Für die Bearbeitung der zweiten Teilaufgabe, einer überarbeiteten Verfahrensweise des Umgangs mit den Mitgliederdaten, existieren eine Reihe möglicher Lösungen. Die Studierenden beschließen, in den nächsten Tagen jeder für sich ein überarbeitetes Szenario aufzustellen und innerhalb ihres virtuellen Lerngruppenraumes abzulegen. Bis zum eigentlichen Termin der Besprechung der Aufgabe in der Übung oder alternativ der Frist der Abgabe bei einem Betreuer besteht nun die Möglichkeit, Anmerkungen und Kommentare zu den Lösungsvorschlägen der Kommilitonen anzufertigen. Eine Studentin der Lerngruppe entdeckt eine mögliche Schwachstelle im Szenario ihrer Kommilitonin. An einer Stelle ist ihrer Meinung nach das Datenschutzgesetz weiterhin verletzt – sie fügt eine entsprechende Notiz hinzu und schickt ihr eine kurze E-Mail. Der dritte Lernpartner hat zwar alle Paragraphen des Datenschutzgesetzes beachtet, seine Kommilitonin bemerkt jedoch in dem von ihm verfassten Szenario, dass nicht genügend Daten an die Bank übermittelt werden, um die Vereinsbeiträge per Bankeinzugsverfahren zu verbuchen. Auch sie notiert das Problem kurz an seinem Dokument. Außerdem hat sie noch eine ganze Anzahl interessanter Texte im Internet zu einer Diskussion um die Auslegung des Datenschutzgesetzes zu einigen in der Gruppe strittigen Punkten gefunden. Sie beinhalten u.a. ein Gerichtsurteil zur Fragestellung des Umganges mit elektronischen Bankdaten. Beides fügt sie als Referenz in den virtuellen Lerngruppenraum ein und kommentiert die Verweise mit einigen Worten zu ihrer Relevanz in Bezug auf die zu lösende Übungsaufgabe.

Bei einem nächsten Treffen überarbeiten die drei Studierenden ihre Entwürfe und geben sie schließlich im virtuellen Übungsraum ab.

Übungsveranstaltung

In der eigentlichen Übungsveranstaltung werden die angedeuteten Arbeitstechniken aus der Vorlesung weiter in Richtung kooperatives Problemlösen und kooperatives Lernen ausgebaut.

Studierende fügen im Vorfeld der Übung ihre Lösungsvorschläge in den virtuellen Übungsgruppenraum ein bzw. erstellen die Lösung durch eine Verknüpfung vorhandener Dokumente, z.B. Bildschirmmasken oder Kriterienkataloge mit persönlichen Lösungsvorschlägen innerhalb der Lern- und Dokumentenumgebung (ein Prozess, der individuell geschehen kann, jedoch meist in kleinen Gruppen von Lernenden vorgenommen wird). Es werden also im Idealfall kooperativ verschiedene Lösungsalternativen entwickelt und diese innerhalb des virtuellen Übungsgruppenraumes abgelegt.

Die eigentliche Übung lässt sich als Wechselspiel der Präsentation und Diskussion der gefundenen Lösungsalternativen beschreiben. Die Diskussion wird von dem Übungsgruppenleiter moderiert, Ziel ist jedoch die möglichst freie Diskussion der Übungsteilnehmer.

Ziel der Übung ist die Darstellung verschiedener Alternativen und ihre gegenseitige konstruktive Kritik bzw. Bewertung, ein Prozess, der wiederum von dem zu gestaltenden System mittels Techniken der Annotation und Strukturierung, d.h. Reorganisation von Objekten auf einem Shared Whiteboard unterstützt werden kann.

Gemeinsam erfolgt eine Verknüpfung der gefundenen Lösungsvorschläge sowie ihre Speicherung und Archivierung innerhalb der kooperativen Wissensräume.

Es entstehen neben den Lösungen der einzelnen Lerngruppen eine oder mehrere von der Übungsgruppe favorisierte bzw. von den Dozenten kommentierte Lösungen. Diese fungieren als Musterlösungen und können öffentlich bereitgestellt werden. Auf diese Weise entstehen kooperativ Lehrunterlagen, welche eine wichtige Basis zu späteren Nachbereitungen und Prüfungsvorbereitungen darstellen.

Eine durch Computer medierte Übung vermag eine Vielzahl von Arbeitsformen zwischen Studierenden und den Übungsgruppenbetreuern zu unterstützen. Im Folgenden werden einige dieser Arbeitsformen anhand verschiedener Beispiele aus den Vorlesungen zur Softwareergonomie, zur Informatik und Gesellschaft sowie aus der Fachrichtung der Technischen Mechanik näher dargestellt.

Einige Kooperationsformen benötigen speziell gestaltete Zugangswerkzeuge zu der kooperativen Lernumgebung, auf deren spezifische Eigenschaften und Realisierung im Folgenden nicht genauer eingegangen wird. Die in Kapitel 3 vorgenommenen Designanforderungen von kooperationsunterstützenden Systemen wurden jedoch unmittelbar aus den dargestellten Arbeitsformen entwickelt.

a) Arbeiten mit linearem und nicht-linearem Text

Zugangswerkzeuge zu einem kooperationsunterstützenden System sind maßgeblich auf den Prozess der Gegenüberstellung von Alternativen und das Arbeiten mit allen Formen von linearem und nicht-linearem Text auszurichten.

Hierzu ist eine enge Verzahnung der synchronen Kommunikationsmittel (textueller Chat, Audio-Chat), der asynchronen Kommunikationsformen (Annotation, Kommentar, Kritik, E-Mail) und der Dokumentenorientierung (Hypertext, Grafiken) anzustreben. Eine derartige Verschmelzung erlaubt z.B. innerhalb einer Übung zur Softwareergonomie kooperativ verschiedene im WWW verfügbare E-Commerce Lösungen zu evaluieren (Shared Browsing) und diese mit nur für die Gruppe der Übungsteilnehmer sichtbaren Kommentaren zu versehen.

In einem parallelen Arbeitsschritt werden gemeinsam Verweise auf die zugrunde gelegten softwareergonomischen Kriterien bzw. innerhalb der Lernumgebung verfügbaren Gesetze und Normen (z.B. der EU-Bildschirmrichtlinie) angefertigt. Wichtige Passagen aus textuellem Chat oder einem Diskussionsbrett können mittels Drag&Drop-Mechanismen als Annotationen oder neue Dokumente verwendet werden. Hierdurch lassen sich z.B. innerhalb eines Diskussionsbrettes formulierte Fragestellungen unmittelbar mit in die Lösung zu einer Übungsaufgabe aufnehmen. Von den Studierenden angefertigte Gestaltungsvorschläge können innerhalb persönlicher und gemeinsamer Arbeitsbereiche (virtuellen Übungsgruppenräumen) abgelegt werden und damit der Gruppe bereitgestellt werden.

Arbeiten mit nicht-linearem Text bedeutet nicht allein die Nutzung von Hypertexten zur Vertiefung eines Themengebietes oder Sachverhaltes der Vorlesung (hier ist z.B. von den Übungsteilnehmern ein Hypertext zum Thema der Maskengestaltung anhand von Styleguides durchzuarbeiten), vielmehr kann der Hypertext als aktives, von den Übungsteilnehmern zu gestaltendes Medium verstanden werden. So kann es z.B. Ziel einer Übung sein, kooperativ einen Hypertext zu einem Themengebiet aus der Vorlesung zu erstellen. Beispielsweise sind von den Übungsgruppenteilnehmern zur Maskengestaltung einzelne Gestaltungsgesetze anzuwenden und kurz in Form einer einzelnen Textseite zu kommentieren. Diese Teilbeispiele werden in der Übung diskutiert und in einen Hypertext zur Anwendung der Gestaltungsgesetze überführt. Ausgehend von einem einführenden Text zu den Gestaltgesetzen sind zunächst formelle Definitionen korrekt zuzuordnen und anschließend die zuvor erstellten Beispiele zu verknüpfen.

b) Explorationen und interaktive Animations- und Simulationssysteme

Neben dem aktiven Lernen unterstützt durch Texte und Dokumente können interaktive Animationen und Simulationen helfen, Sachverhalte zu illustrieren und zu verdeutlichen. Die Idee der Exploration geht über herkömmliche Animationen und Simulationen um einiges hinaus (vgl. [Hampel et al. 1999b] und [Hampel & Keil-Slawik 1998b]).

Basierend auf Erfahrungen im Rahmen der Entwicklung von interaktiven Visualisierungen zur Technischen Mechanik [Hampel et al. 1998a] wurden eine ganze Anzahl von Beispielen hochgradig interaktiver Visualisierungssysteme entwickelt und erprobt, die es Lernenden erlauben eigenständig und konstruktiv in den Simulationsprozess einzugreifen (Konstruktion eines zu simulierenden Versuchsaufbaus) und eine unmittelbare Rückmeldung über die errechneten Ergebnisse zu erhalten. Eine Bereitstellung verschiedener Sichten auf den dargestellten Sachverhalt (z.B. grafische Darstellung des Versuchsaufbaus, Darstellung der relevanten Formeln und mathematischen Zusammenhänge sowie grafische Visualisierung der relevanten Einflussgrößen) erlauben es den Lernenden, aus unterschiedlichen Blickwinkeln ein Problem zu verstehen und durch individuelles „Explorieren“ mit verschiedenen Parametereinstellungen sich eine Vorstellung und ein Bild der dargestellten Beziehungen zu machen.

Explorationen können für ganz unterschiedliche Themenfelder gestaltet sein und im Grade der Komplexität differieren. Sie werden als Lernbausteine innerhalb von virtuellen Räumen abgelegt und dienen der Veranschaulichung eines speziellen Teilthemas. Auf den Bereich der Softwareergonomie bezogen können einfache Explorationen beispielsweise grundsätzliche Zusammenhänge der Farbenlehre verdeutlichen, also verschiedene Wirkungen von Farben auf das menschliche Auge. So kann es innerhalb der Vorlesung zur Softwareergonomie einen virtuellen Farbraum zur Mischung von Farben geben. Studierende finden dort verschiedene Explorationen zur additiven und subtraktiven Farbmischung sowie zur Wirkung von Farbkombinationen auf das menschliche Auge.

Teil der Übung zur Softwareergonomie ist es nun, mit Hilfe von Explorationen verschiedene, textuell nur schwer zu beschreibende Zusammenhänge zu demonstrieren. Wichtige Beiträge zur Verständnisbildung leisten Explorationen damit insbesondere im Anwendungsfeld von komplexen, ohne experimentellen, konstruktiven Aufbau nur schwer zu beschreibenden Bereichen. Sie können dauerhaft in virtuellen Räumen abgelegt und an andere Studierende weitergegeben werden.

c) Erarbeiten von Beispielen

Virtuelle Räume können als praxisorientierte Erprobungsfelder für die Softwareergonomie dienen. Insbesondere durch kleinere reale Übungsbeispiele sollen Studierende in die Lage versetzt werden, aktiv softwareergonomische Fragestellungen zu klären und in begrenztem Umfang Praxiserfahrungen zu sammeln. Damit ist das primäre Ziel von Praxisbeispielen der Softwareergonomie, aktiv Gestaltungsprozesse von Benutzerschnittstellen zu ermöglichen und verschiedene Bedienungs-Metaphern bzw. Alternativen der grafischen und funktionalen Gestaltung gegenüberzustellen.

Dieser Prozess lässt sich in einer Übung zur Softwareergonomie durch den Einsatz geeigneter lernförderlicher Infrastrukturen gezielt fördern. Werden z.B. als Teil der Vorlesung verschiedene Methoden der Fenstertechnik oder der Gestaltung von Dialogen behandelt, ist es Ziel von zugehörigen Übungen, dass Studierende zum einen vorhandene Beispiele gezielt analysieren und gegebenenfalls Schwachstellen identifizieren, zum anderen sind Gestaltungsalternativen anhand von persönlichen, in Teilen zu implementierenden Entwürfen zu erproben und zu bewerten.

Virtuelle Wissensräume müssen auch derartige Prozesse der Softwareentwicklung und Verwaltung von Beispielen geeignet unterstützen. So entwickeln Studierende beispielsweise mittels Werkzeugen zur Oberflächengestaltung (z.B. durch visuelle Programmierung) Entwürfe und Prototypen von Benutzerschnittstellen in Java. Diese Java-Programme oder Java-Applets werden in den virtuellen Übungsgruppenräumen zur Vorlesung abgelegt und sind damit an den verschiedenen Lernorten Vorlesungsraum, Übungsraum und der persönlichen Arbeitsumgebung der Studierenden etc. verfügbar.

Ähnlich den schon beschriebenen Vorgehensweisen zum Lernen mit Hilfe von linearem und nicht-linearem Text oder Explorationen werden Werkzeuge zur Oberflächengestaltung gezielt in den Ablauf der Übung mit eingebunden. So kann etwa das Ziel einer Übung sein, verteilt auf mehrere Gruppen eine Benutzerschnittstelle exemplarisch für ein kleineres Softwaresystem zu skizzieren. Hierzu dient ein zuvor von den Übungsgruppenleitern vorbereitetes Rahmenprogramm. In der Übungsveranstaltung werden in kleineren Gruppen Teile der Benutzerschnittstelle generiert und zu einem größeren Programm innerhalb eines virtuellen Raumes zusammengefügt. Die einzelnen Teillösungen werden in der Gruppe diskutiert und unter Zuhilfenahme der softwareergonomischen Kriterien der Vorlesung analysiert. Schrittweise können begleitend der Vorlesung schon in einem anderen Kontext entwickelte Beispiele aufgegriffen und ausgebaut werden.

Mit Hilfe eines im virtuellen Übungsgruppenraum integrierten Nachrichtentables lassen sich die Abstimmungsprozesse zur gemeinsamen Gestaltung koordinieren. Hier können bekannte Probleme oder auch beliebige Kommentare der Gruppe zu den einzelnen Beispielen festgehalten werden.

Als hilfreiche Werkzeuge zur Unterstützung kooperativer Diskussionsprozesse zu softwareergonomischen Gestaltungsfragen lassen sich, wie schon in den Anwendungsszenarien zur Vorlesung gezeigt, kooperative Zeichenflächen (Shared Whiteboard) nutzen. In einem derartigen Szenario platziert z.B. ein Diskussionsteilnehmer seinen Gestaltungsentwurf auf die Zeichenfläche. Weitere Teilnehmer der Übung können sich nun

mit dem Whiteboard verbinden, indem sie den entsprechenden virtuellen Raum betreten. Durch synchronisierte Mauszeiger (Telepointer) oder auch die Platzierung von grafischen Objekten ist ein unkompliziertes und gezieltes Besprechen der Bildschirmmasken gegeben.

d) Kooperative Wege durch die Lehrveranstaltungsunterlagen

Eine Grundlage des Erarbeitens wesentlicher Konzepte einer Lehrveranstaltung kann das kooperative Anfertigen von Wegen (Trails) durch Lehrveranstaltungsunterlagen bilden.

Wege durch netzgestützte Materialien dienen der Orientierung der Lernenden und beziehen gezielt externe Materialien mit ein, die nicht im Rahmen der Veranstaltung erstellt und unmittelbar innerhalb der Lernumgebung abgelegt sind.

Als Teil einer Übungsaufgabe zum Datenschutz ist beispielsweise ein Trail, ein Weg durch verschiedene Anbieter zum Thema der Anonymisierung von Datenspuren im Internet, Zertifizierungsstellen für Internetdienste o.ä. zu recherchieren und innerhalb eines virtuellen Raumes abzulegen. Der Übungsgruppenleiter informiert zu Beginn der Übung über das Thema, zu dem in kleineren Gruppen Trails angefertigt werden sollen. Es werden z.B. kurz die Möglichkeiten der gezielten Analyse der Spuren, die ein Surfer im Netz hinterlässt, demonstriert und gezeigt, inwieweit die Zusammenführung von unterschiedlichen Daten zu Nutzungsprofilen führen können (vgl. [Fischer et al. 2001]).

Anschließend werden die Teilnehmer der Übung aufgefordert, in kleinen Gruppen zu zwei bis drei Personen gezielt nach Artikeln und Veröffentlichungen, staatlichen Einrichtungen und Informationsmaterialien zu suchen und diese in Form eines Trails – eines geordneten Weges innerhalb des virtuellen Übungsgruppenraumes – abzulegen. Trails bestehen aus markanten Punkten, den Webseiten, also Texten, Grafiken und Dokumenten, die eine geordnete Reihenfolge besitzen. Wegstellen (Punkte auf dem Trail) können mit Notizen und Kommentaren versehen werden, die anderen Lernenden zur Orientierung auf dem Trail dienen können. Trails können des Weiteren Gabelungen besitzen, d.h. ein Weg verästelt sich in zwei oder mehrere Teilpfade, von denen einige zu einem späteren Zeitpunkt wieder in einem Gesamtweg münden können.

Trails können entweder über ein spezielles Werkzeug realisiert sein und damit auch eine grafische Repräsentation besitzen oder durch die räumliche Anordnung von Verweisobjekten innerhalb der virtuellen Raumstruktur abgebildet sein. In jedem Fall ist ein wichtiges Merkmal eines Trails, dass er von den Lernenden eigenständig erstellt, modifiziert und innerhalb von virtuellen Lernräumen abgelegt werden kann. Insofern ist der Trail weniger ein didaktisches Arrangement bzw. eine inhaltliche Aufbereitung von Lehrmaterialien, die den sekundären Medienfunktionen zuzuordnen wäre (vgl. Differenzierung der Medienfunktionen in Abschnitt 2.1), vielmehr ist er als aktives gemeinsames Medium des Arrangierens und Strukturierens von Lehrmaterialien zu interpretieren.

Auf das obige Beispiel bezogen werden nach einer kurzen Recherchezeit die erstellten Trails in der Gruppe vorgestellt und diskutiert. Hierzu eignen sich insbesondere Techniken der gekoppelten Navigation (Shared Browsing) entlang eines Trails. Hierbei übernimmt ein Lernender die Navigation und stellt die einbezogenen Inhalte vor. Andere Lernende können im Idealfall begleitend persönliche oder für die Lerngruppe sichtbare Notizen (Annotationen) an dem Trail vornehmen.

Anschließend wird aus den individuellen Trails der Lernenden ein exemplarischer Weg der Gruppe zusammengestellt, indem unnötige Knoten gelöscht, fehlende eingefügt oder auch verschiedene Wege zusammengeführt werden. Der so entstandene Weg wird von der Gruppe mit gemeinsam gefundenen Kommentaren versehen und fortlaufend ausgebaut.

Solche kooperativ erstellten Trails werden dauerhaft in den virtuellen Übungsräumen zu den Lehrunterlagen abgelegt und nach Möglichkeit ergänzt und aktualisiert. Genutzt werden sie intensiv zur späteren Prüfungsvorbereitung oder zur Unterstützung der Einarbeitung von völlig neuen Lernenden in das beschriebene Themenfeld.

Trails sind damit nicht nur wichtiges Instrument der Navigation durch verschiedene Lehrmaterialien, sondern auch Lerngegenstand an sich, indem die Erstellung eines Trails, d.h. die sinnvolle Auswahl und Bewertung der Materialien des Trails einen wichtigen Beitrag zur Wissensbildung aufseiten der Studierenden darstellt.

Betreuung neben der Vorlesung und Übung – virtuelle Sprechstunde

Neben der allgemeinen Betreuung der Studierenden in Übung und Vorlesung ist mittels einer kooperationsunterstützenden Umgebung eine neue Form der persönlichen Betreuung realisierbar. Studierenden stehen neben den asynchronen Kommunikationsformen wie Diskussionsbrettern auch so genannte virtuelle Sprechstunden oder virtuelle Übungsgruppentreffen zur Verfügung.

Virtuelle Sprechstunden können als feste Zeiträume der Präsenz von Betreuern innerhalb der Dokument- und Kommunikationsumgebung helfen, inhaltliche und organisatorische Fragen zu klären. Eine Sprechstunde wird in dieser Form als primär synchrone Zusammenkunft einer Gruppe von Studierenden und Dozenten verstanden. Eine Online-Diskussion erlaubt den direkten Bezug auf Materialien, bearbeitete Übungsaufgaben, Vorlesungsunterlagen, Musterlösungen oder externe Dokumente des WWW. Ergebnisse der Sprechstunde sind nicht zwingend flüchtig, stehen vielmehr auch anderen Gruppen oder zeitlich nicht anwesenden Studierenden zur Verfügung. Die enge Verzahnung der synchronen Kommunikationsmechanismen, Informationen der gegenseitigen Wahrnehmung (Awareness) und dieser Dokumentenwelt, schafft eine neue Qualität der vermittelten Kommunikation. Derartige Treffen der Teilnehmer einer Übungsgruppe im Netz sind nicht zwingend anzubahnen, vielmehr kann es durch die bloße Anwesenheit eines Betreuers oder anderer Studierender (z.B. den Mitgliedern einer Lerngruppe) zu ungezwungener Kommunikation über die alltäglichen Fragestellungen und Problemen der Vorlesungsinhalte kommen. Der virtuelle Raum ist hierbei zentraler Treff- und Bezugspunkt der Kommunikation.

Eine Anwesenheit in dem virtuellen Raum signalisiert zunächst Ansprechbereitschaft, weiterführende Formen der bereitgestellten Awareness erlauben, z.B. resultierend aus einer aktuellen Tätigkeit, Informationen darüber einzuschränken.

Automatisierte Abgabe von Übungslösungen

Der virtuelle Raum bietet neben einer Form der Erstellung und Strukturierung von Materialien auch organisatorische Erfordernisse einer Lehrveranstaltung an. Das Anfertigen von Lösungen zu gestellten Übungen ist wesentlicher Bestandteil des Übungsbetriebs. Hierbei ist die Abgabe zum Teil freiwillig. In den meisten Fällen können über die Lösung von Übungsaufgaben Bonuspunkte für die spätere Klausur angesammelt werden. Virtuelle Wissensräume vermögen die termingerechte Abgabe von Übungslösungen durch virtuelle Abgabekästen organisatorisch zu vereinfachen. Dies geschieht z.B. durch Ziehen des Übungsblattes auf den Abgabekasten (vgl. Abbildung 2-2).

Ein virtueller Abgabekasten vermag automatisch die fristgerechte Abgabe zu überprüfen und sich bei erheblicher Überschreitung gegen die Annahme zu sperren. Eine Leerung des Abgabekastens wird entweder manuell durch einen Übungsgruppenbetreuer vorgenommen oder die enthaltenen Übungsgruppenlösungen werden automatisch an diesen übermittelt. In jedem Fall sind Informationen der Bearbeiter mit den Übungslö-

sungen verknüpft. Weiterhin werden durch die Abgabe des Übungsblattes gezielt Berechtigungen an den Korrektor übertragen. So erhält der Betreuer die volle Berechtigung zur Korrektur und Annotation, also Bewertung des Dokuments, ein nachträgliches Ändern des Übungsblattes durch die Bearbeiter ist ausgeschlossen.

Auch die Rückgabe der Übungsblätter an die Bearbeiter kann automatisiert vorgenommen werden. Dies geschieht z.B. durch Verschieben der Übungslösungen in die entsprechenden Lerngruppenräume.

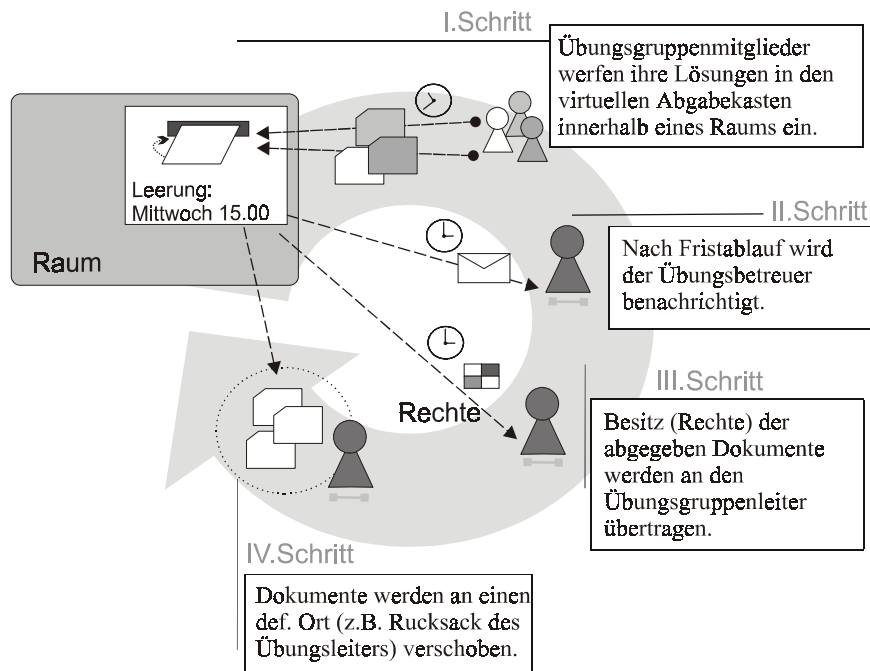


Abbildung 2-2: Automatisierte Übungsabgabe

2.5.4 Kooperative Projektarbeit – Projektgruppe

Anwendungsszenarien für den Einsatz von computerbasierten kooperationsunterstützten Umgebungen finden sich nicht nur in Vorlesung, Übung und Seminar, den klassischen Modellen der Lehre, sondern in gleichem Maße in der Unterstützung von studentischer Projektarbeit.

Moderne Studiengänge sind durch eine Reihe von Praxisphasen geprägt. Bezogen auf das Studium der Informatik fallen hierunter Veranstaltungen zur Projektarbeit in kleineren Entwicklerteams, die so genannten Projektgruppen. Ohne eine zu starke Spezialisierung vorzunehmen, seien im Folgenden die wesentlichen Mechanismen kooperativen Arbeitens am Beispiel einer Projektgruppe zur Informatik analysiert und elementare Unterstützungsfunktionen aufgezeigt.

Projektgruppen sind gegen Ende des Studiums angesiedelte Veranstaltungen, die einen Umfang von ca. sechs Semesterwochenstunden über zwei Semester besitzen. Ziel ist es, Studierende zur gemeinsamen Arbeit im Kontext eines Forschungsprojektes anzuregen. Trainiert werden Fähigkeiten zur Gruppenarbeit und das zielgerichtete Arbeiten an einer Aufgabenstellung über einen längeren Zeitraum. Neben dem Erlernen von Abstimmungsprozessen, wissenschaftlichem Arbeiten und Mechanismen der kooperativen Ideenfindung ist insbesondere ein gutes kooperatives Wissensmanagement notwendig, um Teilergebnisse und Wissen, aber auch Arbeit in der Gruppe zu verteilen. Verschie-

dene Dokumente, wie Anforderungsanalysen, Seminararbeiten, Zwischenberichte und der Projektgruppenabschlussbericht besitzen primär diese Funktion.

Finden im Rahmen der Projektgruppentätigkeit Softwareentwicklungen statt, sind diese in kleineren Programmerteams kooperativ zu gestalten und an möglichst reale, vorbildliche Softwareentwicklungsprozesse anzulehnen. Insbesondere der Umgang mit Quellen, ihre Kommentierung und Dokumentation ist wesentlicher Lerngegenstand der Projektgruppenveranstaltung. Im Rahmen von objektorientierten Softwareentwicklungsprozessen wird großer Wert auf die objektorientierte Analyse und das objektorientierte Design gelegt. Das Entwickeln von Anwendungsfällen und verschiedenen Formen von grafischen Repräsentationen, wie etwa Klassen-, Objekt- und Sequenzdiagrammen gehört u.a. zur Basis eines professionellen Softwareentwurfs [Booch 1994].

In einer Projektgruppe liegt der kreative Anspruch aufseiten der Studierenden. Schrittweise ist eine Lösung für eine bestimmte Fragestellung zu entwickeln, sind Prototypen zu schaffen und eine Gesamtkonzeption zu entwerfen. Formell besteht eine Projektgruppe aus einem Vorlesungs-, Übungs- und Seminarteil. Die Auftaktveranstaltung einer Projektgruppe bildet ein Seminar. Es dient der Einarbeitung der Teilnehmer in die Thematik der Projektgruppe und bereitet wesentliche Grundlagen der in den folgenden Monaten zu bearbeitenden Aufgabenstellung vor. Begleitend zur eigentlichen Projektarbeit und damit verbundener regelmäßiger Treffen kann eine Vorlesungsveranstaltung für die Projektarbeit notwendiges Wissen vertiefen und ausbauen.

Projektgruppen besitzen typischerweise eine Größe von sieben bis fünfzehn Teilnehmern und ein bis zwei Betreuern. Je nach Gruppengröße und Aufgabenstellung kann eine Projektgruppe wiederum in mehrere Untergruppen eingeteilt sein, die jedoch meist eng zusammenarbeiten.

Im Folgenden werden der Ablauf und die wesentlichen Kooperationsstrukturen einer Projektgruppe zur Informatik beschrieben. Übergeordnetes Ziel der als Beispiel hinzugezogenen Projektgruppe ist es, eine Verbesserung der Qualität der Lehre durch den Einsatz von Multimedia-Werkzeugen zu erreichen. Dazu sind, bezogen auf ein konkretes Beispiel (z.B. die Lehre der Maschinentchnik), neue Formen von grafischen Animationen und Simulationen zu entwickeln, zu erproben und in Form eines raumbasierten Informationssystems in die Lehre einzubinden (vgl. [Hampel & Keil-Slawik 1998a], [Hampel & Keil-Slawik 1998b]).

Anmeldung, Seminarphase, Themenfindung und Kennenlernen

Start der Veranstaltung bildet eine kurze Vorstellung der Ziele der Projektgruppe im Rahmen einer Gesamtveranstaltung zur Präsentation aller aktuellen Projektgruppen. Hier werden die organisatorischen Rahmenbedingungen, Zugangsvoraussetzungen und die thematische Ausrichtung vorgestellt. Studierende wählen basierend auf diesen Informationen eine für sie geeignete und ansprechende Projektgruppe. Ergänzende Informationen und weiterführende Texte zum Themengebiet werden in einem virtuellen Raum für Veranstaltungskommentare bereitgestellt.

In einem zweiten Schritt findet, wie schon unter den Szenarien zum Seminar beschrieben, eine Anmeldung der Teilnehmer über den virtuellen Projektgruppenraum statt.

Zentrales Instrument des Informationsaustausches und der Terminabsprache bildet nun der virtuelle Projektgruppenraum. Hier werden beispielsweise wichtige Termine und sämtliche relevante Dokumente und Informationen abgelegt. Ein virtueller Projekt-

gruppenraum dient der Projektgruppe als kooperatives Gedächtnis und ist in seiner Struktur offen angelegt. Alle Teilnehmer der Projektgruppe besitzen im Wesentlichen identische Berechtigungen. Beispielsweise wird jedem Projektgruppenteilnehmer Schreibzugriff auf den Raum, alle darin enthaltenen Dokumente und Nachrichtenbretter gewährt.

Hat sich die Projektgruppe konstituiert, findet im ersten Semester der Projektlaufzeit ein Seminar und eine Vorlesungsveranstaltung in der auf die Unterstützung durch eine kooperative Lernumgebung bezogene schon dargestellte Art und Weise statt. Dies bezieht sich insbesondere auf den Unterstützungsprozess der Auswahl der Seminarthemen und die Betreuung der Seminarteilnehmer in der Vorbereitungsphase ihrer Seminararbeit und des Seminarvortrages.

Die Seminarthemen sind eng an die Zielrichtung der Projektgruppe angelehnt und bilden eine inhaltliche Basis für das geplante Vorhaben. Auf das Beispiel einer Projektgruppe zum Thema „Neue Medien in der Lehre“ bezogen, werden wesentliche Beispiele vorhandener Lehr- und Lernumgebungen recherchiert und innerhalb des virtuellen Projektgruppenraumes abgelegt. Andere Seminarthemen führen in technologische Hintergründe zur Entwicklung von multimedialen Anwendungen ein, wie z.B. in die Programmiersprache Java, ihre wichtigen Bibliotheken oder spezielle Visualisierungstechniken. Des Weiteren illustrieren ausgewählte Seminarthemen den richtigen Gebrauch von Modellierungswerkzeugen und ihren grafischen Repräsentationen (UML) oder den Einsatz von visuellen Programmierwerkzeugen. Ergebnisse sind in allen Fällen Hypertexte, Folien und weiterführende Verweise auf Ressourcen im Netz, die abgelegt im virtuellen Projektgruppenraum den Teilnehmern zur Verfügung stehen.

Wie schon angedeutet, besteht der wesentliche Sinn und Zweck einer Projektgruppe in der Selbstorganisation der Studierenden zur Lösung einer gemeinsamen Aufgabe. Hierbei spielen häufige und längere Treffen eine wichtige Rolle. Umso wichtiger ist jedoch, bedingt durch eine räumliche und zeitliche Verteilung der Projektgruppenmitglieder, die Bereitstellung eines gemeinsamen Dokumentenraumes – dem Gedächtnis der Projektgruppe. Typischerweise belegen die Studierenden parallel zur Projektgruppe weitere Veranstaltungen und sind damit nicht bei jedem Treffen, speziell bei Untergruppentreffen, anwesend. Insofern stellt der virtuelle Projektgruppenraum den Treffpunkt der Gruppe im Netz, aber auch den gemeinsamen Ort der Speicherung aller notwendigen Materialien dar. Dies können neben Texten zur Spezifikation der Projektgruppenaufgabe (auf das Beispiel bezogen sind dies Texte zur Gestaltung von Multimedia, zum Lernen mit Multimedia oder auch inhaltliche Hintergrundinformationen zur Mechanik) auch Programmierdokumentationen und die Dokumentation der Quellen und Strukturbilder sein.

Entwicklung erster Prototypen

Im Verlauf des ersten Semesters der Projektgruppe wird durch eine Reihe von Treffen die bearbeitete Problemstellung analysiert und erste Lösungsvorschläge und Herangehensweisen diskutiert. Als Vorbereitung auf die spätere Entwicklung einer Softwarelösung lassen sich durch die Gestaltung erster, kleinerer Prototypen Realisierungsmöglichkeiten und Anwendungsszenarien erschließen. Auf das Beispiel der multimedialen Lehrunterstützung des Faches Maschinentechnik bezogen werden an Lehrbuchbeispielen orientiert einige kleinere, interaktive Animationen in Java entwickelt. Diese können technische, aber auch inhaltlich-konzeptionelle Realisierungsmöglichkeiten aufzeigen und dienen damit den Projektgruppenteilnehmern der Einarbeitung in die Thematik.

Erste lauffähige Prototypen werden in einem virtuellen Experimentierraum abgelegt, wo sie von den Projektgruppenteilnehmern erprobt und kommentiert werden können. Im Idealfall werden schon in diesem frühen Stadium die späteren Anwender (Dozenten und Studierende der Maschinentechnik) mit den Prototypen konfrontiert und testen deren Einsatzmöglichkeiten. Hierzu ist beispielsweise neben jedem Prototypen ein kurzer einleitender Text sowie ein öffentliches Brett für Kommentare vorzusehen. Die Beispielanimationen sind direkt durch den Client der Lernumgebung, aber auch durch den Zugriff über einen Internetbrowser aus dem virtuellen Experimentierraum startbar. Hierdurch werden aufwändige Testinstallationen vermieden.

Die Entwickler der Prototypen nutzen den virtuellen Projektgruppenraum intensiv zur Ablage aller relevanten technisch-konzeptionellen Informationen. So wird in der Phase der Umsetzung erster Prototypen z.B. eine Dokumentation der erzeugten Java-Klassen als Hypertextdokument abgelegt und fortlaufend aktualisiert. Die an der Entwicklung von Prototypen beteiligten Studierenden können nun die jeweilige Klassen und Methoden mit Anmerkungen versehen und auf diese Weise Verbesserungsvorschläge und Änderungswünsche diskutieren. Die Quellen selbst lassen sich ebenfalls in einem speziellen Experimentierraum (Labor) ablegen und so verteilt weiterentwickeln.⁸²

In den einzelnen Teilgruppenbereichen der Projektgruppe dienen Nachrichtenbretter der Koordination von Design- und Entwicklungsaktivitäten. Etwa bei Änderungen an kooperativ genutzten Modulen werden diese mit dem Änderungsdatum versehen und auf dem entsprechenden Nachrichtenbrett notiert. Diese Mitteilungen können erneut von anderen Teilnehmern kommentiert werden. Auf diese Weise entsteht eine erste Dokumentation der technischen Realisierung der Prototypen während ihrer Realisierung. Neben der Klärung von Fragen oder dem Verwalten von Fehlern kann es spezielle „Kreativbretter“ geben, die beispielsweise Vorschläge zu neuen interessanten Perspektiven des Projektes beinhalten können. Den Entwicklungsprozess begleitende, persistente Foren erlauben insbesondere zu späteren Zeitpunkten, auf zurückliegende Designentscheidungen zurückzugreifen und damit den Entwicklungsprozess kritisch zu hinterfragen. Auf diese Weise sind speziell in realen Treffen der Projektgruppe oder ihrer Untergruppe Protokolle auf diesen Brettern zu führen und damit wichtige Entscheidungen der Gruppe zu archivieren.

Die verteilte Arbeit an der Problemstellung erfordert sowohl in den Bereichen der Dokumentenerstellung, aber insbesondere in der Softwareentwicklung einen hohen Anteil an Koordinierungs- und Abstimmungsprozessen.

Diese resultieren insbesondere aus der räumlichen Verteilung der Projektgruppenteilnehmer, die sich sicherlich nicht täglich zu einer gemeinsamen Entwicklung treffen, sondern an sehr unterschiedlichen Orten (Studentenwohnheim, studentische Poolräume etc.) an dem Projekt arbeiten. Kooperative Wissensräume können durch elementare Funktionen der gegenseitigen Wahrnehmung (z.B. Informationen über die Anwesenheit im virtuellen Projektgruppenraum) die Zusammenarbeit wesentlich unterstützen. Informationen über die aktuelle Themenbearbeitung der Projektgruppenteilnehmer⁸³ gestalten

⁸² Dies wäre eine Alternative zu einem speziellen Verwaltungssystem für Quellen, wie es z.B. das Concurrent Versions System (CVS, vgl. <http://www.cvshome.org/>, Stand 20.9.2001) bietet. Denkbar ist aber auch eine Verknüpfung der Netzschnittstelle eines Quellenarchivierungssystems mit einem entsprechenden Raum der kooperationsunterstützenden Umgebung.

⁸³ Dies kann über einfache Mechanismen, wie die Anwesenheit des Avatars in einem entsprechenden Untergruppenraum, realisiert sein.

die Erörterung eines Designkonflikts oder die Lösung eines Problems um einiges einfacher und effektiver.

Über einen auf einen virtuellen Raum beschränkten Chat können kurze Probleme diskutiert und Verabredungen etc. getroffen werden. Den Betreuern fällt in diesem Zusammenhang insbesondere die Aufgabe zu, organisatorische Rahmenbedingungen im Auge zu behalten, wie z.B. die Festsetzung verschiedener Entwicklungsschritte oder wichtiger Termine. Diese werden auf einem speziellen Organisationsbrett innerhalb des virtuellen Projektgruppenraumes bekannt gegeben.

Ein konkretes Szenario stellt sich in diesem Zusammenhang wie folgt dar: Eine zweiköpfige Lerngruppe ist zuständig für die Entwicklung eines Prototypen zur Visualisierung eines spannungsoptischen Effekts der Maschinenteknik. An diesem Beispiel soll der Einsatz von Java3D für die Visualisierung von Explorationen in der Mechanik erprobt werden.

Von zu Hause aus wählt sich der eine Student in den virtuellen Projektgruppenraum ein und plant, an den von ihm zurzeit entwickelten Quellen eine neue Selektionsmöglichkeit von Objekten zu erproben, z.B. das Teilen des spannungsoptischen Aufbaus. Zuerst werden die relevanten Quellen aus dem virtuellen Experimentierraum geladen. Er vermerkt in dem zugehörigen Diskussionsbrett, an welchen Quellen er Änderungen vornimmt und welche konzeptionellen Änderungen geplant sind.

Anschließend beginnt er mit der Programmierung und ergänzt den lauffähigen Prototypen um die Möglichkeit, einzelne Objekte der optischen Bank mit der Maus zu selektieren, horizontal zu verschieben und zu drehen. In der vorhergehenden Version war eine derartige Verschiebung und Drehung der Filter über Parametereinstellungen gelöst. Parallel zu seinen Entwicklungswerkzeugen ist die im virtuellen Projektgruppenraum abgelegte Dokumentation der mathematisch-physikalischen Zusammenhänge des Polariskops⁸⁴ und eine Java-Dokumentation geöffnet. Er entdeckt einen Vorzeichenfehler in der Beschreibung des Polariskops und markiert ihn durch eine Annotation an das entsprechende Dokument.

Bei der Umsetzung seines Vorhabens ergeben sich zudem einige technische Probleme. So gelingt es ihm nicht, innerhalb der Visualisierung (des Java3D-Szene-Graphen) die richtige Gruppe des selektierten Objektes zu identifizieren. Eine Recherche innerhalb der Java3D-Dokumentation, welche sich innerhalb des virtuellen Entwicklungsraumes befindet, ergibt keine Lösung.

Der Student bewegt seinen Avatar in den Projektgruppenraum der zweiten Untergruppe, die sich mit einem Baukastensystem zur vereinfachten Entwicklung von Java-Animationen beschäftigt, und trifft dort zwei weitere Mitstudenten. Per Chat gibt der Student kurz seine technischen Probleme bekannt und fügt einen Link auf die Stelle im Quelltext an, bei der seiner Meinung nach der Fehler zu suchen sei. Ein Mitstudent nimmt sich des Problems der Selektion des richtigen Knotens im Java3D-Szene-Graphen an, hat einen Verdacht über die Ursache und entdeckt in der Java-Fehlerdatenbank einen diesbezüglich beschriebenen Fehler in der Java3D-Bibliothek mit einem zugehörigen Hinweis zu dessen Umgehung. Er fügt einen Verweis auf den Fehlerreport in das Java-Diskussionsbrett innerhalb des virtuellen Projektgruppenraums ein und meldet sich kurz per Chat zurück.

⁸⁴ Bei einem Polariskop handelt es sich um eine Apparatur für einen spannungsoptischen Versuch, der eine Materialprobe – typischerweise Araldit B – untersucht.

Der Student kann den Fehler tatsächlich auf diese Weise beheben und fügt seine Selektionsmethode als neue Version des Beispiels in den virtuellen Experimentierraum ein. Auf einem zugehörigen Nachrichtenbrett vermerkt er seine Änderungen und stellt insbesondere die neue Art des Drehens der Filter mit der Maus zur allgemeinen Diskussion.

Innerhalb der nächsten Tage erproben verschiedene Projektgruppenmitglieder das neue Verfahren und geben kurze Kommentare ab. Allgemein ist die Resonanz der technischen Realisierung positiv, lediglich die Benutzbarkeit und Handhabbarkeit der neuen Interaktionsmöglichkeit wird von einigen in Frage gestellt.

So ergibt sich durch absolute Navigation innerhalb des spannungsoptischen Aufbaus bei gleichzeitiger Manipulationsmöglichkeit der einzelnen Objekte des Polariskops eine fast unüberschaubare Vielzahl an Möglichkeiten. Der Student und sein Kommilitone beschließen, den Punkt auf dem nächsten realen Projektgruppentreffen in zwei Tagen mit dem Rest der Gruppe und den Betreuern zu diskutieren. Sie signalisieren dies für die anderen Projektgruppenmitglieder auf einem Nachrichtenbrett im virtuellen Projektgruppenraum.

Diese Diskussion ergibt den Plan, die Navigation um das Polariskop einzuschränken und insbesondere eine Navigation in das Polariskop hinein zu vermeiden. Die neuen Interaktionsmöglichkeiten an den Filtern werden beibehalten und ausgebaut, es wird aber gleichzeitig eine zuvor vorhandene numerische Eingabemöglichkeit für die Filterstellung unterstützt.

Der Kommilitone fügt diese Ergebnisse des Projektgruppentreffens kurz in das Dokument der Anforderungsspezifikation zu seinem Teilprojekt ein und bittet einige Studierende des Maschinenbaus um Rückmeldung zu der geänderten Navigationstechnik. Hierzu sendet er ihnen per E-Mail einen Verweis auf die Prototypen des innerhalb des virtuellen Experimentierraums abgelegten Polariskops.

Die geänderte Navigation wird positiv aufgenommen, jedoch merken die Testnutzer an, dass die Drehung der Filter auch unsinnige Konstellationen zulässt, die mechanisch durch Begrenzungen bei dem realen Experiment nicht auftreten können. Der Kommilitone bittet sie, diese Randbedingungen in das Dokument zum mathematisch-physikalischen Hintergrund einzufügen und gewährt ihnen hierzu Schreibberechtigung an dem Dokument. Anschließend überarbeitet er das Programm diesbezüglich und vermerkt den aktuellen Stand auf dem zugehörigen Diskussionsbrett.

Gestaltungsprozess – Evaluation von Prototypen

Im weiteren Verlauf der Projektgruppe wird typischerweise ein Ansatz der evolutionären Softwareentwicklung verfolgt. Hierbei orientiert sich der Gestaltungsprozess an den Anforderungen der späteren Nutzer und einer schrittweisen Verbesserung der Gestaltung anhand einer Reihe von Prototypen. Auch in dem konkret beschriebenen Themenfeld ist die zu gestaltende Lehr-/Lernumgebung zu einem möglichst frühen Zeitpunkt kontinuierlich in der Lehre zu erproben. Hierbei wird die Gruppe der Testnutzer fort-dauernd vergrößert. Ziel erster formativer Evaluationsstudien sind die beschriebenen Animationen/Simulationen oder auch Explorationen. Sowohl in der Gestaltung der Benutzerschnittstelle als auch in der grafischen Realisierung der dargestellten Zusammenhänge ergibt sich ein großes Potenzial an Verbesserungsmöglichkeiten. In der interdisziplinären Zusammenarbeit zwischen der Projektgruppe (Studierende der Informatik) und den späteren Nutzern (Studierende der Maschinentechnik) ist zudem ein großer Anteil von Detailarbeit an der erarbeiteten Lösung erforderlich. Kontinuierliche Evaluationen, d.h. die Nutzung der entwickelten Explorationen durch verschiedene Gruppen von Studierenden der Maschinentechnik, ist ein wichtiges Ziel der Projektgruppenarbeit.

Zur Unterstützung dieser Aufgabe wird ebenfalls der virtuelle Projektgruppenraum genutzt. Ein spezieller Evaluationsraum enthält hierzu die zu erprobenden Prototypen der Explorationen. Ist z.B. das virtuelle Polariskop im Hinblick auf die Bedienbarkeit der Belastungseinrichtung für die Probe zu bewerten, wird der Prototyp in einem virtuellen Labor innerhalb der Lernumgebung abgelegt. Neben der durch einen einfachen Mausklick zu startenden Exploration befindet sich ein kurzer Einführungstext zum Polariskop und dessen Bedienung sowie ein Hypertext zu den spannungsoptischen Zusammenhängen. Weiterhin werden verschiedene Bilder von Proben mit zugehörigen Parametern des realen Versuchs und die resultierenden spannungsoptischen Bilder bereitgestellt.

Den an der Evaluation beteiligten Studierenden wird nun die Aufgabe gestellt, zunächst die Grundfunktionen des Versuchsaufbaus (Verstellung der Filter, Justage der Lichtquelle, Belastung der Probe) zu erkunden und erproben. Studierende, die mit dem optischen Weg der Lichtstrahlen von der Lichtquelle durch die Filter und die Probe noch nicht vertraut sind, kann die Aufgabe gestellt werden, den Weg kurz (z.B. in einer Textdatei) zu erklären und unverständliche Aspekte der entsprechenden Visualisierung aufzuzeigen.

Anschließend sind mit Hilfe des virtuellen Polariskops eine Reihe von Versuchen durchzuführen und die simulierten Ergebnisse mit den Bildern aus dem realen Versuch zu vergleichen. Hierzu sind die gewonnenen Parameterstellungen und das resultierende simulierte Bild innerhalb des virtuellen Evaluationsraums in einem Ordner (Container) zu jedem Studierenden und Versuch abzulegen.

Weiterhin wird jeder Studierende mündlich zu seinen Problemen und seinem Eindruck zur Exploration befragt und die Ergebnisse stichpunktartig in einem Dokument zu den Evaluationsergebnissen abgelegt. An der Evaluation nicht direkt beteiligte Projektgruppenteilnehmer können auf diese Dokumente unmittelbar zugreifen und sie gegebenenfalls direkt kommentieren oder mit Notizen versehen.

Neben dieser mündlichen Befragung ist in dem Evaluationsraum ein automatisiertes Formular abgelegt, in das jeder Testnutzer seine Einschätzungen zu verschiedenen Komponenten der Exploration einfügen kann. Dabei handelt es sich um einen Fragebogen auf Multiple Choice Basis, in dem jedoch auch Kommentarfelder vorhanden sind. Diese Fragebögen können bei größeren Nutzerkreisen durch geeignete Skripte des kooperationsunterstützenden Systems automatisiert ausgewertet werden und zwecks einer kontinuierlichen Rückmeldung auch während der späteren, realen Nutzerphase der Lernumgebung bestehen bleiben.

Basierend auf den Ergebnissen der ersten Nutzung der jeweils untersuchten Prototypen kann nach Diskussion in der Projektgruppe ein entsprechendes Redesign vorgenommen werden. In jedem Fall sind alle Evaluationsergebnisse anonymisiert innerhalb des virtuellen Evaluationsraumes abgelegt. Verbesserungen oder auch Rückentwicklungen, z.B. in der Bedienbarkeit, sind so gut dokumentiert und jederzeit für alle Teilnehmer der Projektgruppe einsehbar.

Sowohl im späteren Einsatz als auch bereits im Rahmen der ersten Erprobung der Exploration sind verschiedene weitergehende Verknüpfungen mittels des Konzepts kooperativer Wissensräume denkbar. Diese äußern sich in erster Linie in neuen Betreuungsmodellen der Lernenden.

Explorationen lassen sich beispielsweise durch Nutzung von synchronen Eigenschaften der Lernumgebung in kooperative Formen überführen, d.h. in Programme mit verschiedenen synchronisierten Eigenschaften. Hierbei lassen sich verschiedene Funktionen koppeln (z.B. die Sicht auf den Versuchsaufbau beim Polariskop bzw. die Ein-

stellungen von Filtern etc.) und wie bei einem Shared Whiteboard synchrone Kooperationsmodelle realisieren.

So kann etwa ein Betreuer über das Netz an einer Sitzung zu einer Exploration teilnehmen und Veränderungen an dem Modell vornehmen – und so beispielsweise einen sinnvollen Versuchsaufbau herstellen. Anschließend wird die Kontrolle der Exploration an einen Studierenden zurückgegeben. Eine Kommunikation findet in räumlich verteilten Szenarien z.B. über einen Chat statt.

Die Zusammensetzung einer Sitzung, d.h. Funktionen der Wahrnehmung und Kommunikation, wird erneut über den virtuellen Experimentier- und Evaluationsraum koordiniert. So können alle Personen eines Raumes an der Sitzung zu einer entsprechenden Exploration teilnehmen.

2.5.5 Kooperatives Selbststudium – Prüfungsvorbereitung

Neben Vorlesung und Übung als wichtige Phasen des Lernens ist die Prüfungsvorbereitung in die Betrachtung der möglichen Unterstützung durch kooperative Wissensräume mit einzubeziehen. Phasen der Prüfungsvorbereitung sind zeitlich nicht zwingend an entsprechende Vorlesungen und Übungen angeschlossen. Auch die Art und Zusammensetzung der Lerngruppe kann sich stark von denen der entsprechenden Übung unterscheiden.

Die Vorbereitungsphase auf eine Prüfung (Klausur oder mündliche Prüfung) ist zunächst als von den Studierenden in hohem Maße selbstorganisierter Prozess zu charakterisieren, d.h. den Studierenden fällt die Aufgabe zu, die Inhalte der entsprechenden Vorlesung in Eigeninitiative nachzubereiten und zu vertiefen. Hierbei ist kein fester Zeitplan oder eine feste Anzahl von Treffen von Seiten der Dozenten vorgegeben. Umso wichtiger ist jedoch ein reger Kontakt zu Übungsgruppenleitern, Dozenten und Mitstudierenden.

Kooperationsunterstützende Systeme können im Rahmen der Prüfungsvorbereitung eine enorme Unterstützungsleistung erbringen. Schwerpunkt liegt ähnlich den schon beschriebenen Anwendungsfällen zur Unterstützung von Übungen auf der Bereitstellung eines gemeinsamen Gedächtnisses für die Prüflinge. Diese müssen von allen Lernorten aus Zugriff auf sämtliche von ihnen und kooperativ mit Kommilitonen erstellten Materialien und Vorlesungsinhalte haben. Hierbei sind insbesondere die zurückliegend, im Rahmen der Lehrveranstaltung erstellten Materialien zu berücksichtigen. Es ist also eine veranstaltungsbezogene Archivierung der Lehrmaterialien vorzunehmen und der Zugriff für ehemalige Teilnehmer zu öffnen, d.h. der volle Funktionsumfang des kooperationsunterstützenden Systems bleibt für die Phasen der späteren Nutzung erhalten.

Auch unter der Perspektive des lebenslangen Lernens ist eine dauerhafte Bereitstellung der Vorlesungsunterlagen wünschenswert. Studierende sollten auch nach Abschluss ihres Studiums im Berufsleben den Kontakt zu ihren Studienunterlagen nicht abrupt verlieren, vielmehr sollten die Vorlesungsunterlagen weiterhin aktiv nutzbar bereitstehen und der Kontakt zu ehemaligen Teilnehmern einer Vorlesung und den Dozenten bei Bedarf wiederhergestellt werden können.

Als zweite wichtige Komponente der Prüfungsvorbereitung sollte ein Vorlesungsraum so genannte Audio-Annotationen [Grimm & Hoff-Holtmanns 1999] zu den jeweiligen Vorlesungsfolien enthalten. Ein Konzept, welches durch Bereitstellung von CD-ROMs für Studierende in der Lehre der Arbeitsgruppe Informatik und Gesellschaft in Paderborn schon seit einiger Zeit praktiziert wird. Audio-Annotationen beinhalten den vom Dozenten gesprochenen Text der entsprechenden Folie als Audio-Strom. Erfahrungen der Arbeitsgruppe belegen, dass Audio-Annotationen wertvolle Dienste in der

Nachbereitung bestimmter Vorlesungsinhalte leisten. Hierbei wird weniger Akzent auf die professionelle Aufnahme und redaktionelle Aufbereitung der Inhalte gelegt als vielmehr ein direkter Bezug zu den früheren Vorlesungen hergestellt. In diesem Sinne sollen die Audio-Annotationen den qualitativen Stand der Vorlesung widerspiegeln und werden in jedem Semester erneut aufgezeichnet.

In Szenarien der kooperativen Nachbereitung und Prüfungsvorbereitung ist das Konzept der Audio-Annotationen nahtlos in den Ansatz der Annotierbarkeit und kooperativen Nutzung und Strukturierung innerhalb von virtuellen Räumen eingefügt, d.h. Audio-Annotationen können zu den einzelnen Vorlesungsfolien abgerufen werden und sind innerhalb der virtuellen Übungsgruppenräume als den anderen digitalen Medien (Dokumente, Grafiken, Referenzen) gleichwertige Materialien gekennzeichnet.

2.6 Wissensräume – funktionale Anforderungen

Die oben beschriebenen Anwendungsbereiche kooperativer Wissensräume konzentrieren sich auf die infrastrukturelle Verbesserung der Lehre. Durch die Bereitstellung der primären Medienfunktionen in der individuellen Arbeit und Gruppenarbeit werden verschiedene, neue Lernformen und Qualitäten im Umgang mit digitalen Medien sichtbar. Diese resultieren zum großen Teil aus der Reduzierung von Medienbrüchen in kooperativen Wissensräumen.

Im Folgenden werden die oben skizzierten Anwendungsfälle, die einen Prozess wie Vorlesung, Übung oder Seminar beschreiben, in eine Anzahl von kleineren Elementartätigkeiten aufgeschlüsselt.

Es wird der Versuch unternommen, diese Einzeltätigkeiten den wesentlichen primären Medienfunktionen zuzuordnen und hierdurch eine gewisse Systematik zu erzielen. Hierbei wird keine strenge Trennung der individuellen primären Medienfunktionen und kooperativen primären Medienfunktionen vorgenommen, es werden vorrangig kooperative Teilbereiche sämtlicher primärer Medienfunktionen berücksichtigt.

Weiterhin ist zu beachten, dass eine große Anzahl der geschilderten Teil-Szenarien eine Vielzahl von Medienfunktionen betreffen und somit auch die eindeutige Zuordnung strittig erscheint. In diesem Fall wurde die entsprechende Tätigkeit einer offensichtlich zutreffenden Medienfunktion zugeordnet.

Als zweites Kriterium für eine Systematisierung der gefundenen Szenarien dienen die Kommunikationsfunktionen innerhalb der Szenarien. Hier wird gezielt nach Unterstützungsfunktionen der Kommunikation zwischen Lernenden und Dozenten untersucht, die durch eine kooperative Lernumgebung geleistet werden können.

Für beide Unterscheidungskriterien ist festzuhalten, dass sie sowohl synchron als auch asynchron einzuordnen sind, d.h. der Wechsel zwischen sitzungsorientierten Anwendungsformen und eher asynchronen Arbeitstechniken ist als fließend zu charakterisieren.

Schließlich helfen neben einer Systematisierung entlang der primären Medienfunktionen eine Anzahl von Metaphern (z.B. Metaphern des kooperativen Wissensraums) die jeweiligen Kooperationssituationen geeignet zu beschreiben. Derartige Kennzeichen, wie z.B. „Foyer“, das einen freien Diskussionsraum beschreiben könnte, dienen insbesondere den Lernenden zur Orientierung in der kooperationsunterstützenden Umgebung und machen durch ihre realweltlichen Bezüge Regeln der Kooperationsstrukturen deutlich.

2.6.1 Metaphern virtueller Wissensräume

Im Folgenden werden beispielhaft einige mögliche Metaphern virtueller Wissensräume genannt. Auf eine detaillierte Beschreibung der jeweiligen Kooperationsstrukturen wird verzichtet, sie resultieren unmittelbar aus den skizzierten Anwendungsszenarien. Zudem ist die Wahl und konkrete Einbettung von Metaphern des virtuellen Raums an der Grenze von den primären zu den sekundären Medienfunktionen anzusiedeln und dahingehend für die verfolgte konzeptionelle und technische Rahmenarchitektur nur von untergeordneter Bedeutung.

Auf der Seite der Inhaltskomponente eines virtuellen Raums lassen sich eine ganze Reihe von eingebetteten Werkzeugen und aktiven Elementen ausmachen. Hierzu zählen Protokollwerkzeuge (Rekorder) einer Diskussion, aktive Abgabekästen für Übungsblätter, gemeinsame Zeichenflächen (Shared Whiteboards) oder kooperative Terminkalender. Ihre Beschreibung folgt im Rahmen der Skizze einer technischen Umsetzung des sTeam-Systems (Kapitel 5).

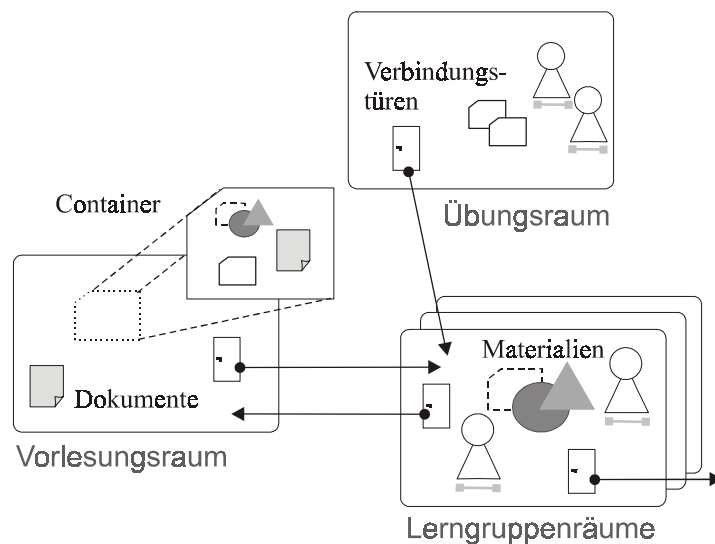


Abbildung 2-3: Struktur aus virtuellen Räumen

Beispiele möglicher Metaphern für den virtuellen Raum sind:

- *Virtuelle „Studentenbude“*: Der private virtuelle Arbeitsraum kann von den Studierenden frei genutzt werden. Hier werden persönliche Lehrmaterialien abgelegt. In erster Linie besitzt der Lernende alleinigen Zugang. Für andere Lernende und Dozenten bleibt der Zugang zunächst verwehrt.
- *Virtueller Vorlesungsraum*: Wie in den Szenarien beschrieben, werden im virtuellen Vorlesungsraum verschiedene Vorlesungsmaterialien abgelegt. In ihm besitzen der Dozent und die Übungsgruppenleiter sämtliche Berechtigungen. Speziell in der Situation der Vorlesung werden genutzte Werkzeuge, z.B. zur Folienpräsentation, von den Dozenten gesteuert. Annotationen und Anmerkungen können von den einzelnen Lernenden an die Materialien angebracht werden (evtl. müssen diese hierzu in einen Übungsgruppenraum oder Lerngruppenraum dupliziert werden).
- *Foyer*: Freier Diskussionsraum zur Vorlesung. Zu jeder Veranstaltung wird neben Vorlesungs- und Übungsräumen auch ein freier Diskussionsraum zur Vorlesung eingerichtet. Eine Anzahl, für jeden Vorlesungsteilnehmer zugreifbarer

Diskussionsbretter dienen der asynchronen Kommunikation von Vorlesungsinhalten und organisatorischen Fragestellungen.

- *Virtueller Fachgruppenflur und virtuelles Informationsbrett*: Ein dem Foyer der jeweiligen Fachgruppen und Vorlesungen zugeordnetes Informationsbrett oder Nachrichtenbrett enthält aktuelle Aushänge und Ankündigungen zu Vorlesung und Übung. Hier werden Verschiebungen der Übungstermine, Ankündigungen von Klausur- und Tutoriumsterminen bekannt gegeben. Anmeldungen zu Lehrveranstaltungen werden ebenfalls über das virtuelle Informationsbrett vorgenommen.
- *Virtueller Lerngruppenraum*: Den von den Studierenden selbstverwalteten Lerngruppenräumen fällt die wichtige Funktion der Strukturierung und Ablage von Lehrmaterialien zu. Hier können von Studierenden erstellte Unterlagen wie Lösungen zu Übungsaufgaben, weiterführende Literatur oder Zusammenfassungen abgelegt werden. Die virtuellen Räume der einzelnen Lerngruppen liegen vollständig in der Verwaltung der Studierenden und bleiben auch über mehrere Veranstaltungen hinweg erhalten.
- *Virtueller Übungsraum*: Der virtuelle Übungsgruppenraum dient der Strukturierung der Lehrmaterialien im Rahmen der Übungsveranstaltung. Die Teilnehmer der Übung besitzen einen gemeinsamen Schreibzugriff auf die abgelegten Materialien. Neue Materialien dürfen von allen Teilnehmern der Übungsgruppe in den virtuellen Übungsgruppenraum eingebracht werden. Sämtliche Materialien können für die gesamte Gruppe lesbar annotiert werden. Für die Abgabe von Übungsblättern sind spezielle Ordner (Briefkästen) vorgesehen.
- *Virtueller „Stöberraum“*: Eine ganze Anzahl von Anwendungsszenarien zielen auf ein freies Erkunden und Durcharbeiten von Materialien durch Lernende. Konzepte wie Trails beschreiben Wege durch Lehrmaterialien, die zum großen Teil in einem speziellen Raum abgelegt sein können. Virtuelle Stöberräume können eine große Anzahl von Beispielen und Materialien zu einem Themengebiet enthalten, in dem sich Lernende frei bewegen und eigene Strukturierungen, wie etwa Wege durch das Material oder Annotationen an den Materialien, anbringen können.
- *Virtueller Experimentierraum*: Ein virtueller Experimentierraum enthält interaktive, kooperative Animationen, Visualisierungen oder Explorationen. Mit ihrer Hilfe können Zusammenhänge der Vorlesung und Übung nachvollzogen und experimentell „entdeckt“ werden (vgl. die Szenarien zur Projektgruppe).

Den wesentlichen Charakter eines virtuellen Raums machen neben der tragenden Metapher speziell seine Handlungsmöglichkeiten aus. In den folgenden Abschnitten werden hierzu die Beziehungen des virtuellen Raums und seiner Medienfunktionen untersucht.

2.6.2 Konzeptuelle Einbettung der primären Medienfunktionen

Wesentlicher Aspekt des in der vorliegenden Arbeit entwickelten Rahmenkonzepts nimmt die Einbettung der primären Medienfunktionen in die Metapher des virtuellen Wissensraums ein. Anhand der entwickelten Szenarien werden in den folgenden Abschnitten kooperative Handlungen identifiziert und den primären Medienfunktionen zugeordnet. Ein derartiges Vorgehen erlaubt eine differenzierte Festlegung notwendiger konzeptioneller Anforderungen an eine kooperative Lernumgebung aus medientechnischer Perspektive.

Erzeugen

Tätigkeiten des Erzeugens und Kreierens von Materialien nehmen einen großen Anteil der vorgestellten Szenarien ein. Hierbei besteht natürlicherweise ein enger Bezug zur Medienfunktion des Zugreifens. Erzeugen ist zumeist an Berechtigungen an weiteren Materialien (z.B. dem Ort, an welchem etwas erzeugt wird) gekoppelt. Derartige Berechtigungen werden durch die Medienfunktion des Zugreifens ausgedrückt.

Teilszenarien oder Arbeitstechniken, die dem Erzeugen zuzuordnen sind, sind weiterhin eng mit der Medienfunktion des Verknüpfens verwandt. So kann z.B. das Anbringen einer Annotation als eine Variante des Verknüpfens verstanden werden, sie geht einher mit dem Erzeugen eines neuen Verweisobjekts, ist also auch eine Tätigkeit des Erzeugens. Ausgangspunkt der Identifikation von Tätigkeiten des Erzeugens sind die Fragestellungen: *Wann wird etwas kreiert?* und *Wann wird etwas gesammelt? Werden neue Instanzen oder Kopien von Materialien erzeugt oder gesammelt?*

Tätigkeiten aus den vorgestellten Szenarien, in dessen Zusammenhang digitale Medien kreiert werden, sind beispielsweise:

- *Erzeugen eines Raums:* Das Erzeugen eines Lerngruppenraums durch die Lernenden oder Dozenten kann zunächst der Medienfunktion des Erzeugens zugeordnet werden. Gleichzeitig wird durch das Erzeugen von Räumen und Ordern (Containern) die Struktur der Lernumgebung verändert. Es handelt sich in dieser Form gleichfalls um die Ausübung der Medienfunktion des Arrangierens.
- *Verteiltes Bearbeiten eines Übungsblattes:* Eine Gruppe von Lernenden teilt ein Übungsblatt in eine Anzahl von Teilaufgaben auf, die auf die Gruppenmitglieder verteilt gelöst werden. Es entstehen Teillösungen zu den einzelnen Aufgaben, die zu einer Gesamtlösung kombiniert werden.
- *Individuelles oder kooperatives Erzeugen von Notizen:* Studierende erzeugen persönliche Notizen für sich oder die Gruppe (z.B. die Lerngruppe). Diese werden in allen Stufen des Lernprozesses hinzugezogen, modifiziert oder zwischen Lernenden ausgetauscht.
- *Kooperatives Editieren eines Textes:* Studierende entwerfen zu einem Themengebiet der Vorlesung gemeinsam einen Text. Kooperative Elemente der Texterstellung ergeben sich beispielsweise durch geeignete mündliche Abstimmungsprozesse oder durch Aufteilung in Teildokumente.
- *Erzeugen von Vorlesungsunterlagen, speziell Folien und Texten:* Dozenten, aber auch beteiligte Übungsgruppenleiter und Lernende entwerfen im Zusammenhang von Vorlesung und Seminar Textdokumente, einzelne Folien oder auch Folienvorträge, die innerhalb von virtuellen Räumen abgelegt werden.
- *Erzeugen von Grafiken:* Dozenten entwerfen Grafiken zur Illustration von Inhalten der Vorlesung. In gleicher Weise werden als Teil der Übung oder persönlichen Strukturierung von Lehrmaterialien auch von Studierenden Grafiken entworfen. Dies können u.a. Gestaltungsvorschläge, Übersichten oder Illustrationen sein.
- *Audio-Annotationen:* Audio-Annotationen werden begleitend der Vorlesung erzeugt und geben wesentliche Abschnitte von Vorlesung oder Übung wortgetreu wieder. Sie werden mit entsprechenden Folien oder weiteren Vorlesungsmaterialien verknüpft.

Löschen

Obwohl kaum explizit genannt, ist die Medienfunktion des Löschens in gleicher Weise Teil der dargestellten Szenarien kooperativer Wissensräume wie die Medienfunktion des Erzeugens. Das Löschen von Materialien findet in sehr unterschiedlichen Kontexten bzw. Formen statt. Löschen ist an Nutzerrechte geknüpft und demgemäß eng mit der Medienfunktion des Zugreifens verbunden.

Speziell im Zusammenhang mit den Medienfunktionen des Verknüpfens und Arrangierens beeinflusst das Löschen maßgeblich die Gestalt und Anordnung der Materialien eines Wissensraums. Dies bezieht sich insbesondere auf das Löschen von Elementen, die weitere Materialien strukturieren bzw. einen Zugriff auf diese ermöglichen. Dementsprechend müssen verschiedene Strategien in der Behandlung von Beeinflussungsfunktionen des Löschens gefunden werden.

Es gilt eine ganze Anzahl von Situationen der dargestellten Szenarien zu systematisieren, die im Zusammenhang mit der Medienfunktion des Löschens zu berücksichtigen sind:

- *Löschen von Dokumenten, Grafiken, Text:* Das Löschen von verschiedenen Materialien einer Lernumgebung ist von den zugesprochenen Zugriffsrechten (z.B. der Schreibberechtigung) abhängig. Festzulegen ist, inwieweit das Löschen eines Dokuments nachträglich rückgängig gemacht werden kann oder entsprechende Inhalte unwiederbringlich verloren sind.
- *Löschen von Objekten, die andere Objekte beinhalten:* Das Löschen von Strukturierungselementen einer Lernumgebung, also Objekten, die andere Objekte kapseln oder beinhalten, bedarf einer besonderen Behandlung. Beispielsweise sind besondere Nutzerrechte und Strategien für das Löschen eines Raums oder eines Ordners bzw. Containers zu entwickeln. So können enthaltene Elemente automatisch mit entfernt werden oder eine gesonderte Behandlung erfahren (z.B. Verschieben an einen anderweitigen Ort etc.).
- *Löschen von Verweisen und Referenzen:* Generell verlangen primäre Medienfunktionen von Verweisen und Referenzen besondere Beachtung. So kann sich das Löschen eines Referenzobjekts in Abhängigkeit der Medienfunktion des Zugriffs (Berechtigungen) auf das Referenzobjekt oder den referenzierten Inhalt (das entsprechende Objekt) beziehen. Demgemäß führt das Löschen einer Referenz entweder zum Löschen des Referenzobjekts (das referenzierte Dokument bleibt bestehen) oder das referenzierte Objekt selbst wird entfernt. Kooperative Wissensräume müssen hierbei klare Strategien entwickeln bzw. differenzierte Funktionen bereitstellen.
- *Löschen von Annotationen:* Annotationen müssen unabhängig von dem annotierten Objekt löscher sein. Zu überlegen gilt, inwieweit der Autor eines Dokumentes es vermag, unerwünschte Annotationen an seinen Dokumenten zu löschen (dies sollte sicherlich der Fall sein).

Arrangieren

Der Prozess des Arrangierens kennzeichnet Tätigkeiten des Anordnens und Strukturierens von Lehrmaterialien. Auch im Falle des Arrangierens besteht eine enge Verknüpfung zur Medienfunktion des Zugreifens. Arrangieren setzt in vielen Fällen den Zugriff auf eine gemeinsame Darstellung bzw. den Zugriff auf die zu arrangierenden Objekte oder Verweise bzw. Referenzen zu diesen voraus.

Weiterhin besteht, wie in den Szenarien gekennzeichnet, eine enge Beziehung zu den Tätigkeiten des Verknüpfens, also des In-Beziehung-Setzens von Lehrunterlagen.

So dienen Tätigkeiten des Arrangierens, wie beispielsweise das Anordnen von Elementen auf einer gemeinsamen Zeichenfläche (Shared Whiteboard), in vielen Fällen einer Verknüpfung von digitalen Medien. Gleichzeitig kann das Erzeugen eines Verweises (z.B. auf einer Überblickskarte) der Tätigkeit des Arrangierens von Lehrmaterialien zugeordnet werden. Auf die beschriebenen Szenarien bezogen stellt sich in dieser Form insbesondere die Fragestellung: *Wann wird etwas arrangiert und strukturiert?*

Beispiele für Handlungen des Arrangierens durch Lernende finden sich in mannigfaltigen Formen der gezeigten Szenarien. Beispiele sind:

- *Strukturierung von Lehrmaterialien durch Räume und Container:* Räume und Container dienen der thematischen Strukturierung von Lehrunterlagen. In dieser Form ist das Kreieren eines Raums nicht alleinig der Medienfunktion des Erzeugens, sondern insbesondere der Medienfunktion des Arrangierens (Strukturierens) zuzuordnen. Bei der Kapselung von Objekten in Ordnern (Containern) wird dieses um einiges deutlicher. Container dienen der thematischen Ordnung von digitalen Medien und fassen Lehrmaterialien unter einem spezifischen Gesichtspunkt zusammen.
- *Arbeiten mit Duplikaten/Referenzen von Objekten:* In vielen Fällen ist die Struktur der Lehrmaterialien zunächst fest vorgegeben. Lernende schaffen durch Referenzen oder Duplikate persönliche Sichten auf Materialien. Hierzu werden Teile vervielfältigt (z.B. Vorlesungsfolien) und in den persönlichen Lern- bzw. Übungsgruppenräumen abgelegt.
- *Strukturierung von Wissen durch semantische Karten oder räumliche Arrangements:* Der Raum selbst, aber auch spezielle grafische Repräsentationen werden von Lernenden zur Strukturierung von Lehrmaterialien genutzt. Durch spezielle grafische Darstellungen lassen sich Relationsbeziehungen zwischen Materialien visualisieren.
- *Vorstrukturierung von Wissen durch Dozenten und Übungsgruppenleiter (Experten), spätere Feinstrukturierung durch die Lernenden:* Auf das Szenario eines Seminars bezogen werden beispielweise Artikel nach Themengebieten vorsortiert bereitgestellt. Den Gruppen der Lernenden fällt anschließend die Aufgabe zu, diese thematisch feiner zu untergliedern und in eine geeignete Struktur zu überführen. Als eine Variante wird das Klassifikations- bzw. Strukturierungsschema der Lehrunterlagen vorgegeben. Aufgabe der Übungsteilnehmer ist es anschließend, die eigentliche Strukturierung vorzunehmen.
- *Kooperatives Browsen, Arrangieren von Verweisen:* In Situationen der Recherche und des Sichtens von Lehrmaterialien lassen sich Referenzen oder Verweise sammeln und innerhalb von virtuellen Räumen in eine Ordnung bringen. Verweise sind hierbei themenbezogen einzusortieren. Ein derartiger Prozess wird in der Gruppe umgesetzt. Gefundene Referenzen werden gegenseitig vorgestellt und diskutiert.
- *Kooperatives Browsen, Erzeugen und Nachvollziehen von Trails:* Neben dem thematischen Arrangement von Verweisen können diese durch eine geeignete Struktur innerhalb von Ordnern, Containern oder auch durch ein spezielles Werkzeug in eine zeitliche Abfolge (einen Weg) gebracht werden.
- *Diskussionsunterstützung: Unterstützung einer realen Diskussion (reales Treffen) durch eine gemeinsame Zeichenfläche (Shared Whiteboard):* Ein Shared Whiteboard kann als Strukturierungshilfe eines realen oder computergestützten Diskussionsprozesses dienen. Hierbei werden kooperativ von verschiedenen Diskussionsteilnehmern Notizen angefertigt.

- *Räumliche Zuordnung von Personen zu Inhalten, Definition von Zuständigkeiten:* Durch die räumliche Darstellung eines virtuellen Raums lassen sich Beziehungen zwischen anwesenden Avataren (Piktogramme der Nutzer) und thematischen Strukturierungen von Lehrmaterialien (Ordner, Container) ausmachen. In einem derartigen Prozess positionieren sich beispielsweise die Betreuer einer Lehrveranstaltung jeweils in die räumliche Nähe von Themengebieten, für die sie ein potenzieller Ansprechpartner sind.

Verknüpfen

Die Medienfunktion des Verknüpfens untersucht das Herstellen von Beziehungen zwischen verschiedenen Materialien. Eine Verknüpfung von Dokumenten kann durch unterschiedliche Mechanismen erreicht werden. Hierzu zählen insbesondere Annotationen, aber auch direkte Verbindungen, wie sie durch Verweise oder Strukturierungen hervorgerufen werden. Aus diesem Grunde ist speziell eine Abgrenzung gegen die Medienfunktion des Arrangierens nicht in jedem Zusammenhang notwendig.

Verknüpfen ist zudem in einer Vielzahl der Szenarien eng verbunden mit dem Zugriff (Medienfunktion des Zugreifens) auf Objekte oder Dokumente. Hierbei dient eine Referenz als Zugriff auf ein Objekt.

In den gezeigten Szenarien finden sich verschiedene Beispiele des Verknüpfens von digitalen Medien. Hierzu zählen:

- *Verweisobjekte:* Die sicherlich offensichtlichste Tätigkeit des Verknüpfens von Materialien findet sich in dem Erzeugen von Verweisobjekten. Diese umfasst die Verknüpfung von Textbausteinen untereinander und mit beliebigen anderen digitalen Medien, also das Erzeugen von Hypertexten. Hinzu kommt das explizite Erzeugen eines Verweisobjekts auf weitere Materialien. Als Teil der aufgezeigten Szenarien legen Lernende sowohl zwischen internen als auch zu externen Materialien Verweise an. Hier zeigt sich die enge Verwandtschaft der Medienfunktion des Verknüpfens zu den Medienfunktionen des Arrangierens und Zugreifens. Verweisobjekte (Verknüpfungen) unterliegen einer eigenen Struktur von Zugriffsrechten, d.h. Verweisobjekte können ähnlich wie Annotationen nur für eine bestimmte Gruppe von Lernenden sichtbar sein, aber auch unter existierenden Einschränkungen erzeugt werden, z.B. auch wenn keine Schreibberechtigung für ein Objekt vorhanden ist.
- *Erzeugen von Verweisen – Kooperatives Bearbeiten/Strukturieren von Text:* Wie in den Szenarien zum Umgang mit Textdokumenten gezeigt, kann die Restrukturierung eines existierenden Textes, z.B. durch Verknüpfen von Textabschnitten, ein wichtiges Lernziel sein. Kooperatives Erstellen oder auch die Umwandlung eines linearen Textes in einen Hypertext stellen typische Beispiele für derartige Lernszenarien dar. Diese können individuell, aber auch kooperativ gestaltet sein. So kann es beispielsweise Teil der Übung sein, in kleinen Gruppen von Lernenden eine Anzahl von Textbausteinen geeignet zu verknüpfen.
- *Verknüpfung durch Erzeugen einer Annotation:* Neben der expliziten Erzeugung eines Verweisobjekts bildet die Annotation eine sehr direkte Verknüpfung eines Objekts mit einem Text oder einer Anzahl von anderen Objekttypen. Aus den skizzierten Szenarien ergeben sich eine Vielzahl von Formen von Annotationen. Diese umfassen beispielsweise: Kooperatives Annotieren von Materialien – die Annotationen sind für die Gruppe sichtbar; privates Annotieren von Materialien – die Annotationen sind für den privaten Gebrauch bestimmt und nur für den Einzelnen sichtbar.

Übertragen

Die Medienfunktion des Übertragens, also der Kommunikation von Medien bzw. der Weitergabe von Lehrmaterialien, ist an einer Vielzahl von Stellen innerhalb der Szenarien zu erkennen und, wie schon in Abschnitt 2.3.1 dargestellt, ebenfalls eng verbunden mit der Medienfunktion des Zugreifens.

Die Art und Weise des Transports von Medien ist insbesondere beeinflusst von der Anzahl der Rezipienten. So handelt es sich bei der Übergabe eines Dokumentes durch einen Rucksack meist um eine Eins-zu-Eins-Beziehung. Soll ein Dokument an eine Gruppe von Lernenden weitergegeben werden, eignet sich die Ablage innerhalb eines Gruppenraums, was die Medienfunktion des Zugreifens kennzeichnet.

Beispiele für den direkten Transport von Medien finden sich u.a. in Szenarien der Abgabe von Übungslösungen oder der Übertragung von Rechten an einem Dokument und seiner Weitergabe in einem Rucksack eines Nutzers.

Die entscheidenden Fragestellungen im Zusammenhang mit der Medienfunktion des Übertragens lauten demgemäß: *Wann werden Medien von einem Lernenden oder Dozenten an einen anderen Lernenden weitergegeben?*

Mit Hilfe unterschiedlicher Mechanismen werden Dokumente von einem Lernenden zu einem anderen Lernenden transportiert: Hierbei ist der Prozess der Übergabe und die damit verbundene Konstellation von Zugriffsrechten durch verschiedene Metaphern charakterisierbar.

Sie umfassen beispielsweise das Verschieben von Dokumenten in den Rucksack eines anderen Lernenden. Aber auch das Ablegen eines Dokuments an einem definierten Ort (z.B. innerhalb eines Lernraums) und die Übereignung eines Dokumentes an einen anderen Lernenden durch Setzen entsprechender Zugriffsrechte (Schreibrechte) kann als eine Medienfunktion des Übertragens gedeutet werden. – Es zählt der Akt der gezielten Weitergabe von Materialien an einen anderen Lernenden.

Im ersten Fall wird ein Dokument direkt an einen weiteren Lernenden übereignet, im zweiten Fall ist zuvor die Struktur von Zugriffsrechten geeignet festzulegen, um dem Rezipienten das Dokument vollständig zu übertragen, also insbesondere schreibenden Zugriff zu ermöglichen. Es ist ggf. das Recht zur Weitergabe von Berechtigungen entsprechend zu setzen, um ein Dokument vollständig zu übereignen.⁸⁵

Zusammengefasst lassen sich folgende Tätigkeiten als Übertragen von Materialien im Sinne der primären Medienfunktionen darstellen:

- *Rucksack*: Übertragen eines Dokumentes durch den Rucksack eines Nutzers. Über seinen Rucksack wird ein anderer Lernender explizit adressiert, ein Dokument wird einem Lernenden übereignet bzw. schreibender Zugriff gewährt. (Das Verhalten des Rucksackes ist entsprechend festzulegen.)
- *Gezieltes Setzen von Rechten*: Ein Dokument wird von seinem Besitzer explizit für den Zugriff eines anderen Nutzers (Lernenden) vorbereitet. Kriterium ist hier das explizite Setzen von Berechtigungen bzw. das Übertragen von Berechtigungen an einen weiteren Nutzer (vgl. Abschnitt 3.1.5).

⁸⁵ Das Zusprechen von Leseberechtigung an einem Dokument wird entsprechend als Medienfunktion des Zugreifens interpretiert.

Synchronisieren (Ereignisse und Repräsentationen) – Wahrnehmen (Personen und Handlungen)

Die Medienfunktion des Synchronisierens steht bei den gezeigten Szenarien kooperativer Wissensräume nicht unmittelbar im Vordergrund.⁸⁶ Nichtsdestotrotz lassen sich eine ganze Reihe von Situationen ausmachen, in denen Informationen oder Sichten auf gemeinsame Objekte synchronisiert werden bzw. Informationen zur gegenseitigen Wahrnehmung bereitgestellt werden.

Es finden sich die folgenden Bereiche und Formen des Synchronisierens in den dargestellten Szenarien:

- *Synchronisation von Raumdarstellungen:* Gemeinsame Darstellungen auf einen virtuellen Raum (z.B. einen virtuellen Lerngruppenraum) werden zwischen den einzelnen Teilnehmern (ihren Clients) synchronisiert. Verschiebt beispielsweise ein Lernender ein Objekt innerhalb des Raums (z.B. in einen Ordner), werden die Darstellungen der Nutzer, die ebenfalls eine Sicht auf den entsprechenden Raum besitzen, aktualisiert.
- *Synchronisation von Informationen der gegenseitigen Wahrnehmung:* Informationen der gegenseitigen Wahrnehmung, wie beispielsweise die anwesenden Lernenden oder ihre Handlungen werden innerhalb des virtuellen Wissensraums dargestellt. Insbesondere innerhalb einer Chat-Sitzung oder auf einer gemeinsamen Zeichenfläche werden die beteiligten Nutzer genannt. Neben reinen passiven Wahrnehmungsinformationen können auch aktive Informationen über die Handlungsschritte der Lernenden übermittelt werden. Hierzu zählt beispielsweise das Erscheinen oder Verlassen eines Nutzers innerhalb eines Raums.
- *Synchronisation von gemeinsamen Zeichenflächen:* Gemeinsame Zeichenflächen bieten eine enge Kopplung der Sichten auf die bearbeiteten Dokumente. Entsprechend werden Handlungsschritte der beteiligten Nutzer innerhalb einer Sitzung synchronisiert. Gemeinsame Mauszeiger (Telepointer) sind prototypisch für eine enge Kopplung derartiger Sichten.
- *Synchronisation gemeinsamer Informationen (Shared Browser, gemeinsame Folienpräsentation):* Ähnlich einem Shared Whiteboard werden an vielen Stellen innerhalb der genannten Szenarien Informationen synchron präsentiert bzw. gemeinsam betrachtet. In beiden Fällen liegt die Steuerung eines Shared Browsers bzw. eines kooperativen Präsentationswerkzeuges bei einem Lernenden bzw. bei einer Gruppe von Lernenden. Mechanismen der Koordinationsunterstützung wie Telepointer oder synchronisierte Bedienelemente (z.B. ein kooperativ bedienbarer Vor-/Zurück-Knopf auf einem Folienpräsentationswerkzeug) unterstützen derartige Prozesse.
- *Synchronisation gemeinsamer Handlungen:* Verschiedene Materialien, wie z.B. eine kooperative Exploration (vgl. die Szenarien in Abschnitt 2.5.4) oder auch Werkzeuge des kooperativen Schreibens erlauben eine Synchronisation ihres Inhaltes und ihrer Darstellung. So werden gekoppelte Sichten auf eine Exploration synchronisiert bzw. im Falle eines kooperativen Editors Modifikationen anderer Lernender an einem Dokument visualisiert und abgeglichen.

⁸⁶ Wie schon dargestellt, resultiert dies aus der vorrangigen Berücksichtigung von Situationen der Präsenzlehre.

Zugreifen (Berechtigungen)

Nur an wenigen Stellen der genannten Szenarien erfolgt die direkte Übertragung, d.h. der Transport von Lehrmaterialien durch explizite Adressierung eines Lernenden. So wird beispielsweise ein Skript nicht unmittelbar an einen Lernenden übergeben, es wird die Berechtigung zum Lesen eingeräumt und damit den Lernenden das Recht eröffnet, es in einen anderen Raum zu überführen und neu zu strukturieren.

Die Medienfunktion des Zugreifens steht für die Sichtbarkeit von Dokumenten und untersucht die Fragestellung: *An welchen Stellen erfolgt der Zugriff auf Materialien und wo wird der Zugriff explizit gewährt oder eingeschränkt?*

Zugreifen auf Materialien ist kennzeichnend für synchrone und insbesondere asynchrone Formen kooperativen Lernens. Lernende greifen auf Dokumente zu, die von anderen Lernenden oder Lehrenden bereitgestellt und mit entsprechenden Berechtigungen versehen werden. Damit ist die Medienfunktion des Zugreifens Teil aller schon genannten primären Medienfunktionen. Beispiele finden sich in vielfältiger Zahl in den dargestellten Szenarien:

- *Zugreifen auf fremde Dokumente:* Materialien werden von Lernenden innerhalb eines virtuellen Raums abgelegt. Auf diese kann in Abhängigkeit von entsprechenden Berechtigungen von Mitlernenden zugegriffen werden. Lesender Zugriff entspricht der Sichtbarkeit des Dokuments, schreibender Zugriff ermöglicht seine Modifikation.
- *Zugreifen auf externe Dokumente:* Über Referenzen bzw. Verweisobjekte kann der Zugriff auf externe, nicht innerhalb eines virtuellen Raums abgelegte Materialien erfolgen. Verweisobjekte können anderen Lernenden weitergegeben werden, die durch die entsprechende Adresse den Zugriff erhalten.
- *Zugreifen auf Annotationen:* Annotationen sind von Berechtigungen des Zugriffs abhängig, d.h. Berechtigungen steuern die Annotierbarkeit eines Dokumentes, aber auch den Zugriff auf vorhandene Annotationen.
- *Zugreifen auf Informationen eines Nachrichtenbrettes:* In analoger Weise zu den genannten Annotationen entspricht das Lesen einer Nachricht auf einem Informationsbrett dem Zugriff auf die enthaltenen Informationen und ist von den entsprechenden Berechtigungen abhängig.

Basierend auf im Rahmen der Untersuchung von typischen Szenarien des kooperativen Lernens identifizierten Medienfunktionen erfolgt in den folgenden Abschnitten eine Analyse existierender kooperationsunterstützender Systeme. Diese Analyse erfolgt in Fortsetzung der gewonnenen medientechnischen Perspektive und berücksichtigt insbesondere Elemente der Selbstorganisation und Selbstadministration durch die Lernenden.

2.7 Analyse einer Reihe von Systemen

Wichtiger Schritt auf dem Weg zu einer Architekturkonzeption kooperativer Wissensräume ist es, verschiedene existierende Ansätze auf ihre Einsetzbarkeit hin zu untersuchen. Wie schon einleitend beschrieben, werden hierzu gezielt Systeme ausgewählt, die einen hohen Innovationsgrad in Bezug auf die konzeptionelle Umsetzung aufweisen und wesentliche Impulse für das zu entwickelnde Rahmenkonzept liefern können.

Eine Umsetzung der im vorliegenden Kapitel skizzierten Anwendungsfälle ist mit Hilfe verfügbarer Systeme durch eine Vielzahl auftretender Medienbrüche nur in sehr begrenztem Maße möglich. Speziell der Bereich der Web-gestützten kooperativen Dokumentenverarbeitung kann in dieser Form lediglich in Teilaspekten mit Hilfe von netz-gestützten Dokumentenmanagement-Systemen wie Hyperwave umgesetzt werden (vgl.

[Brennecke & Selke 2000]). Eine Übersicht verschiedener asynchroner Dokumentenmanagement-Systeme, teils mit synchronen Elementen, im Einsatzfeld der Schule und universitären Lehre liegt bei [Schulte et al. 1999] vor.

In [Ho 2000] wurde der erste Versuch dokumentiert, einige typische (jedoch sehr eingeschränkte) Lehrszenarien mit Hilfe von raumbasierten und sitzungsbasierten Systemen abzubilden. Im Folgenden werden die Werkzeuge TeamRooms [Roseman & Greenberg 1996a], Habanero [Chabert et al. 1998] und CVW [Spellman et al. 1997] zu einer Umsetzung der Szenarien berücksichtigt und praktisch erprobt.

Jedes der drei gewählten Systeme steht prototypisch für eine Klasse von kooperationsunterstützenden Ansätzen und macht damit Teilaspekte der im Rahmen dieser Arbeit entwickelten kooperativen Wissensräume deutlich. TeamRooms steht für die Klasse synchroner, aber persistenter, kooperationsunterstützender Umgebungen. Innerhalb eines virtuellen Raums werden Aktionen der Kooperationspartner synchronisiert und Formen kooperativen Arbeitens bereitgestellt. Habanero unterstreicht den Aspekt synchroner kooperativer Werkzeuge, ist jedoch primär sitzungsorientiert. CVW basiert auf der Metapher virtueller Räume und dokumentiert Aspekte kooperativer Dokumentenverwaltung.

Es werden bewusst Systeme gewählt, die in Teilaspekten herausragende Eigenschaften in Bezug auf die Umsetzung der Szenarien zum kooperativen Lernen aufweisen. Die Wahl fällt auf die oben genannten Systeme, obwohl keines explizit zur Unterstützung im Lehr-/Lernumfeld konzipiert wurde. Keines der Systeme erfüllt für sich allein die Anforderungen einer offenen und selbstverwalteten Unterstützung der Lehre, wie in den dargestellten Szenarien gefordert. Insofern erscheint es sinnvoll, insbesondere die Leistungsfähigkeit und Innovation von Teilaspekten der einzelnen Systeme hervorzuheben. Im Folgenden werden die einzelnen Systeme kurz vorgestellt und ihr Einsatz unter Berücksichtigung der obigen Szenarien erläutert.

2.7.1 TeamWave

Das TeamWave-System [Greenberg & Roseman 1998] bildet die kommerzielle Weiterentwicklung von TeamRooms [Roseman & Greenberg 1996a], welches an der Universität Calgary entwickelt wurde. TeamWave ist ein rein netzgestütztes, jedoch nicht WWW-basiertes System und orientiert sich als zentrales Konzept an der Metapher des virtuellen Raumes.

Jeder virtuelle TeamWave-Raum ist zugleich als Shared Whiteboard konzipiert, d.h. der virtuelle Raum wird als planare Fläche verstanden und bildet zugleich eine synchrone Zeichenfläche. Verschiedene Objekte, wie Grafiken und Dokumente, aber auch Verbindungen zu anderen Räumen oder Werkzeuge werden auf dieser Arbeitsfläche positioniert. TeamWave ist von seiner Grundidee sitzungsbasiert. Es ermöglicht zwar in begrenztem Maße asynchrones Arbeiten, fokussiert jedoch synchrone Formen der Zusammenarbeit in kleinen Nutzergruppen. Teilnehmer können auf eine Liste der vorhandenen Räume zugreifen und sich in einer Sitzung zu verschiedenen Räumen verbinden. Die Aktionen der Teilnehmer einer derartigen Sitzung werden zwischen den einzelnen Akteuren synchronisiert, d.h. TeamWave eignet sich primär für kleinere Gruppen von Kooperationspartnern.

Innerhalb eines Raumes stehen neben einem synchronen Chat und einer Liste der aktuell mit dem Raum verbundenen Teilnehmer auch eine Übersichtskarte (Radar View) des Rauminhaltes bereit. Inhalte eines Raumes sind persistent, d.h. Sitzungen können jederzeit unterbrochen und wieder aufgenommen werden. Weiterhin bietet TeamWave

eine Versionsverwaltung für den Rauminhalt, d.h. der Zustand eines Raumes lässt sich zu einem beliebigen Zeitpunkt speichern und später rekonstruieren.

Die Realisierung des TeamWave-Raumes als gemeinsame synchrone Arbeitsfläche bedingt eine enge Ausrichtung der bereitgestellten Werkzeuge auf dieses Konzept. In Form von speziellen Applikationen werden beispielsweise ein Präsentationswerkzeug, ein Betrachtungswerkzeug für unterschiedliche Dokumentenformate (File Viewer), ein Concept Mapping-Werkzeug, eine Art Nachrichtenbrett und verschiedene weitere kooperative Werkzeuge bereitgestellt.

Derartige Werkzeuge weisen einen hohen Integrationsgrad in die TeamWave-Umgebung auf und sind in Bedienung und Darstellung eng synchron gekoppelt. So besitzt z.B. der kooperative File Viewer einen eigenen Rollbalken (Scroll Bar) für jeden Teilnehmer der Sitzung. Werkzeuge werden der Whiteboard-Metapher entsprechend auf der Fläche des Raumes ausgeführt und sind damit für alle Teilnehmer der Sitzung sichtbar. Sie lassen sich innerhalb des Raumes frei anordnen, auf ein Symbol verkleinern bzw. auf die benötigte Größe skalieren. Verschiebungen der einzelnen Fenster auf dem Raum und die Bewegungen der Mauszeiger (Telepointer) sind für alle Teilnehmer einer Sitzung sichtbar. Hierbei werden die einzelnen Mauszeiger durch unterschiedliche Farbgebung dargestellt. Die Arbeitsfläche (der Hintergrund des Raumes) und auch die meisten Werkzeuge unterstützen eine Reihe von einfachen Zeichenfunktionen (verschiedene geometrische Objekte, Text), welche ebenfalls für alle Teilnehmer einer Sitzung gekoppelt dargestellt werden.

Verknüpfungen zwischen Räumen werden in TeamWave über eine eigene „Doorway“-Applikation realisiert. Sie verknüpft zwei Räume durch eine feste Verbindung und erlaubt den schnellen Wechsel des Raumes ohne den Umweg über die Liste aller Räume gehen zu müssen.

TeamWave besitzt ein recht fortschrittliches Konzept der Vergabe von Berechtigungen für Räume. Berechtigungen sind an den Raum selbst und nicht an die Objekte innerhalb der Räume geknüpft. Dies zeugt von der grundsätzlichen Idee des TeamWave-Systems, möglichst gleichberechtigte Kooperationspartner innerhalb zu haben.

Benutzer müssen sich vor einer Sitzung identifizieren und erhalten in Abhängigkeit ihrer Gruppe bestimmte Rechte. Rechte drücken konkrete Aktionen innerhalb von Räumen aus, wie z.B. der Zugang zu einem Raum, die Modifikation des Whiteboards oder das Erzeugen und Löschen von Werkzeugen. Neben einer Anzahl von vordefinierten Benutzergruppen können über ein spezielles Administrationswerkzeug weitere Gruppen eingerichtet werden.

Für jeden Nutzer kann das Recht Räume erzeugen zu dürfen explizit festgelegt werden. Das Erscheinungsbild eines neuen Raumes kann hierbei über ein spezielles Template spezifiziert werden. Hierdurch können verschiedene Grundtypen von Räumen wie Übungsraum, Vorlesungsraum etc. bereitgestellt werden, die eine Vorauswahl von Werkzeugen enthalten. Wie schon angedeutet, kann der gesamte Inhalt eines Raumes mit einem Zeitstempel versehen archiviert und zu einem späteren Zeitpunkt rekonstruiert werden. Bei einer derartigen Sicherung werden auch die Inhalte der aktuell geöffneten Werkzeuge (z.B. der Inhalt des Bildbetrachters) gesichert. Sicherungen von Räumen werden automatisch vorgenommen, wenn der letzte Nutzer den Raum verlässt. Zu jedem Zeitpunkt können verschiedene Stadien des Rauminhaltes rekonstruiert werden.

2.7.2 Habanero

Synchrone Anteile der Szenarien aus dem Bereich der Vorlesung und der Übung lassen sich mit Habanero [Chabert et al. 1998] erproben.

Habanero wird am National Center for Supercomputing Applications (NCSCA) in den USA entwickelt und ist als rein sitzungsbasiertes System konzipiert. Ziel ist die Unterstützung verschiedener Formen der synchronen Zusammenarbeit, wie gemeinsames Betrachten von Netzinhalten, kooperatives Editieren oder spezielle kooperative Anwendungen, beispielsweise ein Shared Whiteboard oder ein kooperatives Bildbetrachtungswerkzeug. Habanero-Werkzeuge besitzen keinerlei Mechanismen, um Daten dauerhaft innerhalb der Umgebung abzulegen, ausgenommen der Möglichkeit Daten lokal zu speichern oder Aktionen einer Sitzung aufzuzeichnen (Aufzeichnung und anschließende Wiedergabe einer Sitzung). Dies bedingt ein sehr eingeschränktes Einsatzspektrum auf zeitlich begrenzte, unmittelbare Formen der Zusammenarbeit. Die Persistenz der Arbeitsergebnisse ist von den Teilnehmern selbst zu realisieren und zu verwalten.

Als vollständig netzgestütztes (Java-) System bietet Habanero verschiedene kooperative Applikationen wie ein Shared Whiteboard, einen Shared Browser, ein Chat und einen Multi-User Editor. Eine offene Systemarchitektur erlaubt das flexible Hinzufügen weiterer kooperativer Applets (Hablets) – insofern sehen die Entwickler Habanero primär in der Bereitstellung einer Rahmenarchitektur für sitzungsbasierte kooperative Werkzeuge. Verschiedene Institutionen entwickeln zurzeit basierend auf Habanero eine Reihe von kooperativen Applikationen, wie z.B. ein kooperatives Visualisierungswerkzeug für wissenschaftliche Daten oder ein kooperatives Lernprogramm zur Geometrie.

Aus Nutzersicht ist Habanero sitzungs- und werkzeugbasiert. Ein Benutzer kann sich nach Anmeldung an das System mit einer oder mehreren Sitzungen verbinden. (Aktuell existierende Sitzungen werden in einer Auswahlliste gezeigt.) Anschließend werden alle in einer ausgewählten Sitzung aktiven Werkzeuge präsentiert. Mit dem Ausführen (Starten) einer Applikation sind automatisch alle folgenden Aktionen innerhalb des Werkzeugs gekoppelt. Hierbei ist der Kopplungsgrad von der Applikation selbst bestimmt.

Zu jeder aktiven Sitzung werden die aktuell verbundenen Nutzer dargestellt. Jeder Habanero-Nutzer besitzt eine spezielle Identifikationskarte, welche persönliche Informationen über die entsprechende Person enthält.

Habanero besitzt keinerlei weitergehenden Rechte- oder Gruppenstrukturen. Ausgerichtet auf kleinere Gruppen von Kooperationspartnern und ein offenes Kooperationsmodell sind Interaktionen innerhalb der verschiedenen Applikationen nicht eingeschränkt und werden nur durch die Applikation selbst kontrolliert. Beispielsweise besitzen sämtliche Teilnehmer der Sitzung eines kooperativen Browsers identische Berechtigungen für die Schaltflächen zum Seitenwechsel. Mechanismen des Floor Control sind in Habanero nicht vorgesehen. Rechte zum Erzeugen oder Schließen einer Sitzung sind ebenfalls nicht vorhanden. Das Schließen des Fensters eines Werkzeugs führt zur Beendigung der gesamten Sitzung.

2.7.3 Collaborative Virtual Workspace

Der Collaborative Virtual Workspace (CVW) [Spellman et al. 1997] bildet die dritte der untersuchten Umgebungen. Ursprünglich von der Mitre Corporation entwickelt und anschließend in ein Open Source-Produkt überführt, versteht es sich als eine Mischung aus sitzungs- und dokumentenzentrierten, asynchronen Werkzeugen. Hierbei wurden wesentliche Konzepte aus dem Bereich des MOO entnommen. CVW kommt damit der im Rahmen dieser Arbeit vorgestellten Rahmenarchitektur sowohl konzeptionell als auch architektonisch sehr nahe.

CVW ist als raumbasiertes, netzgestütztes System konzipiert. Räume und die in ihnen enthaltenen Objekte (Dokumente) sind persistent und repräsentieren die Gruppenar-

beitsumgebung. Diese bildet ein virtuelles Gebäude bestehend aus verschiedenen Etagen und Räumen, welche jeweils an einem zentralen Flur angesiedelt sind. Ein Plan über die Etagen und Räume vermittelt dem Benutzer einen Überblick dieser Struktur. Ein CVW-Client erzeugt eine Sicht auf den virtuellen Raum. Dieser beinhaltet im Wesentlichen drei Bereiche: einen textuellen Chat, eine Übersicht der aktuell anwesenden Personen (dargestellt durch eine Liste von Piktogrammen) sowie eine Visualisierung des virtuellen Raums (Content) mit abgelegten Dokumenten und Objekten.

Räume sind thematisch ausgerichtet und können spezifische Werkzeuge zur Unterstützung ihrer Funktion beinhalten. Ein Beispiel hierzu ist der Diskussionsraum, welcher einen Rekorder zur Aufzeichnung von textuellen Diskussionsbeiträgen besitzt. Einen derartigen Rekorder setzte schon Carlstrom zur Aufzeichnung von Aktivitäten von Nutzern im LambdaMOO-System ein (vgl. [Carlstrom 1992]).

Neben einem herkömmlichen Chat beinhaltet die CVW-Umgebung Kommunikationsmechanismen wie Audio- und Videokonferenz und ein Shared Whiteboard. Dieses beschränkt sich auf elementare Zeichenfunktionen und bietet keinerlei weiterführende synchrone Mechanismen wie z.B. Telepointer. Interessanterweise kann das Whiteboard sowohl synchron genutzt werden, unterstützt aber auch asynchrone Formen der Zusammenarbeit, indem Objekte in Abwesenheit der Kooperationspartner platziert werden. Weiterführende Werkzeuge wie kooperative Editoren sind nicht vorhanden, zur Anzeige von Dokumenten dient ein Webbrowser.

CVW adaptiert eine Reihe von aus dem Bereich der MOOs bekannten Konzepte. Hierzu zählt ein persönlicher Rucksack, den jeder Nutzer mit sich führt. In ihm können Dokumente abgelegt und so von einem Raum zum anderen transportiert werden. Das Einbringen von Dokumenten in die CVW-Umgebung wird ebenfalls über den Rucksack realisiert. Neu eingefügte Objekte finden sich zunächst ebenfalls in dem Rucksack eines Benutzers und können anschließend innerhalb eines Raums abgelegt werden.

Der Zugang zu einem Raum und der Zugriff auf einzelne Dokumente werden über Benutzer- und Gruppenrechte spezifiziert. Eine Liste der berechtigten Personen, die „room access list“, wird von dem Raumverwalter, dem „room manager“ verwaltet. Dieser bestimmt zudem, welchem Nutzer Zutritt gewährt wird. Der Zugriff auf Dokumente wird ebenfalls über einfache Zugriffslisten kontrolliert. Hier legt der Besitzer eines Dokumentes fest, welche Nutzer und Gruppen Zugriff auf das entsprechende Dokument erhalten sollen.

Nutzergruppen können von jedem Nutzer erzeugt werden. Hierbei werden existierende Nutzer zu einer neuen Nutzergruppe zusammengefasst, die Administration der Nutzer selbst unterliegt jedoch einem zentralen Administrator.

CVW stellt verschiedene Formen der gegenseitigen Wahrnehmung bereit. Wie in einem klassischen MUD/MOO üblich, werden die Aktionen der Nutzer innerhalb des Textfensters (Chat) ausgegeben. Dies umfasst das Erscheinen und Verlassen der Arbeitsumgebung, aber auch die Interaktionen der Nutzer an Dokumenten. So ist jederzeit ersichtlich, welche Personen sich aktuell ansprechbar in der Arbeitsumgebung befinden.

Ein weiteres interessantes Konzept der Interaktion im virtuellen Raum verwirklicht CVW durch so genannte „Proxies“. Ein CVW-Nutzer kann zugleich in mehreren Räumen anwesend sein. Neben seiner realen Position erlauben es verschiedene „Proxy“-Sichten, das Geschehen in anderweitigen Räumen zu beobachten und sich sogar an Diskussionen (Chat) zu beteiligen. Der Zugriff auf Dokumente bleibt jedoch der ursprünglichen Position vorbehalten.

CVW wurde primär firmenintern (innerhalb der Mitre Cooperation) in Bereichen der Koordinations- und Arbeitsunterstützung erprobt und hat den Status eines Prototypen nie verlassen. Die Aufgabe der Prototypen wird speziell in der Erprobung der Verknüp-

fung von synchronen Kommunikationsmechanismen und einfachen Funktionen des Dokumentmanagements gesehen.

2.7.4 Ergebnisse aus der Umsetzung

Die genannten Systeme werden im Folgenden auf ihre Umsetzbarkeit bzgl. der im vorliegenden Kapitel spezifizierten Szenarien des kooperativen Lernens hin untersucht. Dies ist, wie schon angedeutet, aufgrund verschiedener Einschränkungen der berücksichtigten Systeme nur in Ansätzen erreichbar und für einen realen Lehrbetrieb kaum möglich.

Eine der Grundfunktionen eines kooperationsunterstützenden Systems liegt in der Bereitstellung eines gemeinsamen Gedächtnisses der Lerngemeinschaft. Diese Funktion wird durch den virtuellen Raum, wie ihn der Collaborative Virtual Workspace (CVW) mit seinen verschiedenen Funktionen des Erzeugens und Arrangierens von Medien bietet, in wesentlichen Teilen erfüllt. Bei genauer Analyse werden jedoch eine Reihe von Medienbrüchen sichtbar, die fruchtbare Kooperationsprozesse erschweren.

Anmeldevorgang

Analog den gezeigten Szenarien zum Übungs- und Vorlesungsbetriebs erfolgt zunächst eine Anmeldung der Übungsteilnehmer bei dem Betreuer der Übungsgruppe. Keines der untersuchten Systeme unterstützt einen automatisierten (oder halbautomatisierten) Anmeldeprozess der Studierenden. Aus diesem Grunde sind die persönlichen Daten der Teilnehmer manuell zu erfassen und über ein Administrationswerkzeug in die Umgebung einzubringen.

Das Erzeugen neuer Räume und das Einrichten zusätzlicher Nutzergruppen ist mittels CVW einfach umsetzbar. Studierende können z.B. für ihre Lerngruppe einen eigenen CVW-Raum einrichten und eine entsprechende Nutzergruppe erzeugen.

Erzeugen

Lehrmaterialien aus Vorlesung und Übung werden in verschiedenen virtuellen Räumen abgelegt, dies sind z.B. virtuelle Vorlesungs- und Übungsräume für die einzelnen Übungsgruppen. Da CVW keine Mechanismen besitzt, um HTML-Dokumente in der Umgebung selbst zu verwalten (keine Bereitstellung von konsistenten Verweisen in der Umgebung), sind Hypertextdokumente, die aus einer größeren Anzahl von Dateien bestehen (z.B. ein Skript, welches eine Anzahl von Grafiken enthält), auf einem externen Webserver abzulegen und über einen Verweis in die Umgebung einzubringen. Diese Einschränkung trifft auf sämtliche betrachteten Werkzeuge zu. Speziell TeamWave und Habanero, welche Szenarien des kooperativen Browsens unterstützen, bieten keinerlei Mechanismen, HTML- oder XML-Dokumente direkt innerhalb eines virtuellen Raumes auf dem Whiteboard zu platzieren. So werden z.B. die Unterlagen zu einer Vorlesung in ein HTML-Format überführt und auf einem herkömmlichen Webserver abgelegt. Anschließend wird ein Verweis von dem zugehörigen Raum zur Vorlesung auf die entsprechenden Folien eingefügt. Ziel ist es, diese Tätigkeit Dozenten und Übungsgruppenleitern zu ermöglichen.

Anderweitige Materialien und Dokumentenformate wie Texte und Grafiken werden direkt innerhalb eines Raumes platziert. So können z.B. Übungsaufgaben und die zugehörigen Lösungen über den Rucksack des Übungsgruppenleiters oder eines Studierenden in dem Raum als Datei gespeichert und mittels einer beliebigen Applikation angezeigt werden.

Der Zugriff auf die CVW-Umgebung von allen (typischen) Lernorten ist durch die Nutzung des Java-Clients sichergestellt.

Durch die Spezifikation geeigneter Zugriffsrechte lassen sich Vorlesungsmaterialien gegenüber unautorisiertem Schreibzugriff schützen. Hierbei sind Berechtigungen für einzelne Dokumente festlegbar und nicht alleinig der Zugang zum gesamten Raum kontrollierbar. Werden z.B. Folien auf einem externen Webserver abgelegt, sind die Zugriffsrechte der CVW-Umgebung für diesen Server nicht relevant. Dafür werden spezielle Berechtigungen auf dem externen Server benötigt. Hierdurch wird das Konzept der Selbstadministration entscheidend verletzt, da die Rechteverwaltung des externen Webserver entkoppelt von der Rechteverwaltung des CVW-Systems ist und damit Studierende ohne explizite Berechtigung für die Administration des Webserver keine eigenen Netzmaterialien abzulegen vermögen. Insofern muss in vielen Fällen auf einfache Dokumentenformate (linearer Text) zurückgegriffen werden, die jedoch nicht den Einsatz von kooperativen Werkzeugen, wie z.B. den eines Shared Browsers unterstützen. Die Medienfunktion des Erzeugens und Übertragens (der Austausch) von Lehrmaterialien ist in dieser Form nicht für alle Teilnehmer der Lernumgebung vollständig umgesetzt bzw. nur unter Umwegen (Medienbrüchen) möglich.

Verknüpfen

Neben dem Erzeugen und Speichern von Materialien sind deren Verknüpfung und das Arrangieren wichtige Kernkonzepte kooperativer Wissensräume.

Die Anforderung des persönlichen Strukturierens und Verknüpfens parallel zu Vorlesung und Übung wird durch CVW nur in Teilen gewährleistet. Während der Vorlesung oder Übung können von den Studierenden eigene Verweise auf externe (durch eine WWW-Adresse spezifizierte Referenzen) angefertigt werden. Leider sind keine Verweise auf Objekte möglich, die sich innerhalb der virtuellen Lernräume befinden. Zudem erlaubt das System die Verknüpfung von virtuellen Räumen (Erzeugen von Verbindungen, Türen zwischen Räumen).

Mittels CVW und TeamWave können persönliche Notizen oder Anmerkungen parallel zu Vorlesung oder Übung erstellt und als eigenständige Dateien in den virtuellen Räumen abgelegt werden. CVW bietet keinerlei weiterführende Mechanismen der Annotation. TeamWave erlaubt zumindest Mechanismen der Annotation bezogen auf die Nutzung des synchronen Shared Whiteboard (Platzieren von textuellen Kommentaren, Markieren von Bildschirmbereichen etc.).

Arrangieren

Zunächst erstreckt sich das Schaffen von Bedeutungszusammenhängen in CVW auf die Gruppierung von Objekten innerhalb von virtuellen Räumen. Es können Dokumente in Ordnern zusammengefasst und hierdurch semantische Beziehungen hergestellt werden. Damit sind Grundelemente der Medienfunktion des Arrangierens vorhanden.

Eine Realisierung des Übungsbetriebs gelingt mittels TeamWave insbesondere im Bereich der Unterstützung von synchronen Szenarien des Arrangierens sowie der synchronen Kommunikation. Untersucht werden im Folgenden Szenarien der Lehre der Arbeitsgruppe Informatik und Gesellschaft (Präsenzsituationen).

Auf das Beispiel zur Softwareergonomie bezogen eignet sich TeamWave insbesondere zu einer Unterstützung der softwareergonomischen Bewertung einer Bildschirmmaske: Mittels des zentralen Administrationswerkzeugs wird zunächst eine neue Nutzergruppe zur Übung erzeugt, für jeden Teilnehmer einer Übung eine Zugangsberechtigung eingerichtet und der entsprechenden Nutzergruppe zugefügt. Die Zugangsdaten

werden von den Teilnehmern selbst eingebracht – TeamWave unterstützt in diesem Punkt einen geringen Anteil an Selbstadministration.

Der Übungsgruppenleiter positioniert eine Bildschirmkopie (Grafik) innerhalb des TeamWave Raums. Durch verschiedenfarbige Zeichenfunktionen kann diese kooperativ kritisiert und können z.B. mögliche Verbesserungsvorschläge diskutiert werden. Hierbei werden unter anderem kurze Texte als Annotationen auf die Grafik platziert.

So kreist z.B. ein Studierender einen Bereich der Bildschirmmaske in rot ein, um eine unübersichtliche Darstellung der gezeigten Tabelle deutlich zu machen. Ein Kommilitone annotiert die entsprechende Stelle durch einen kurzen Text, der eine günstigere Tabellenüberschrift vorschlägt. Mittels verschiedenfarbiger Telepointer können Studierende und Dozenten in ihren Ausführungen auf verschiedene Bereiche innerhalb der diskutierten Bildschirmmaske verweisen und im Zuge einer Diskussion oder Frage Markierungen oder Kommentare anbringen. Hierbei werden recht freie Kooperationsstrukturen verwirklicht.

TeamWave unterstützt in dieser Form speziell die Medienfunktion des Arrangierens in kleinen Gruppen von Studierenden, z.B. einer Lerngruppe. Das bereitgestellte kooperative Whiteboard repräsentiert eine Sicht auf einen virtuellen Raum, welcher zwar als Ganzes persistent ist, jedoch aufgrund fehlender Mechanismen des Zugriffsschutzes nicht für komplexere Gruppenstrukturen geeignet scheint. Komplexere Gruppenprozesse, wie die Abbildung einer Lehrveranstaltungsstruktur, benötigen eine differenzierte Vergabe von Zugriffsrechten. Eine Reduzierung des Zugriffsschutzes auf ein globales Recht zum Zugang zu einem Raum bietet beispielsweise keinerlei Schutz der von dem Dozenten bereitgestellten Vorlesungsmaterialien.

Im Bereich der Selbstadministration werden mittels TeamWave einige interessante Konzepte verfolgt, die sich unmittelbar in freien Lernformen innerhalb der umgesetzten Szenarien niederschlagen: Ein Übungsgruppenleiter stellt beispielsweise einen für die Lehrveranstaltung zugeschnittenen virtuellen Lerngruppenraum als Vorlage (Template) bereit. Dieser enthält Verweise auf weiterführende Materialien (im WWW) und wesentliche Materialien der Übung. Anschließend können Lerngruppen mit wenigen Aktionen einen neuen Lernraum von der bereitgestellten Vorlage ableiten. Dieser Raum ist vollständig von der Gruppe selbstverwaltet.

Neuen Mitgliedern wird von der Lerngruppe selbst der Zugang zum virtuellen Lerngruppenraum gewährt. Das Eintragen von neuen Lernenden in die zugehörige Nutzergruppe wird von der Gruppe oder dem Gruppenverwalter selbst vorgenommen.

CVW ist in Hinsicht des Einrichtens von Lerngruppenräumen um einiges restriktiver, hier können neue Räume nur durch einen zentralen Administrator eingefügt werden.

Zugreifen, Navigation, gegenseitige Wahrnehmung

Lernende navigieren zwischen verschiedenen virtuellen Räumen (Vorlesungsraum, Übungsräume, Lerngruppenraum) durch den CVW-Client, der jeweils eine Sicht auf einen Raum bietet. Ein Raum besitzt eine eindeutige Bezeichnung, z.B. „Übungsraum 1 zur Softwareergonomie WS2000/2001“ und einen Beschreibungstext, der den Raum weiter charakterisiert. Ein eigenes Recht kontrolliert den Zugang zu einem Raum. Lerngruppen benötigen einen abgeschlossenen Raum, um beispielsweise eine private Besprechung einer Übungslösung abzuhalten oder einen Entwurf eines Übungsblattes abzulegen.

Informationen der Anwesenheit anderer Lernender sind in der CVW-Umgebung gegeben. So sieht ein Lernender, der einen Übungsraum betritt, unmittelbar, welche Kommilitonen und Dozenten sich in dem Raum befinden. Durch die schon beschriebene

Möglichkeit, in mehreren Räumen passiv anwesend zu sein, wird der kooperative Lernprozess um einiges vereinfacht. Ein Lernender kann sich in seinem Übungsgruppenraum befinden und gleichzeitig (über seinen Proxy) in einem Lerngruppenraum anwesend sein. Wird in dem Übungsgruppenraum aktiv mit Materialien gearbeitet, es sind z.B. verschiedene Verweise zu erzeugen, können gleichzeitig Rückfragen in der entsprechenden Lerngruppe gestellt werden.

Mittels des CVW-Systems sind zudem eine Reihe von Szenarien zur Betreuung der Studierenden begleitend zu Vorlesung und Übung umsetzbar. Hierzu ist für jeden Übungsgruppenleiter neben dem Übungsraum ein virtueller „Bürraum“ einzurichten. Büroräume können den Fachgruppen entsprechend auf einem gemeinsamen virtuellen Flur angesiedelt sein. Virtuelle Büroräume erfüllen eine ganze Reihe von Funktionen. Hier legt jeder Übungsgruppenbetreuer z.B. Informationen zu seiner Erreichbarkeit ab. Die Anwesenheit eines Übungsgruppenleiters ist in dieser Form über ein entsprechendes Piktogramm zu erkennen. Er kann in einer virtuellen Sprechstunde mittels eines Chats angesprochen werden. Ein weiterer typischer Anwendungsfall ist die Unterstützung einer netzgestützten Sprechstunde durch ein Whiteboard. Auf diesem wird der zu diskutierende Sachverhalt vertiefend erläutert. Derartige Szenarien gelingen speziell mit Unterstützung des TeamWave-Systems. Telepointer vereinfachen hier z.B. im Kontext der gemeinsamen Diskussion einer Bildschirmmaske das Fokussieren der Diskussion durch Deuten auf einen bestimmten Bereich innerhalb der Maske.

Übertragen

Das Übertragen (Person zu Person) von Lehrmaterialien, zum Beispiel der Lösung eines Übungsblattes, wird über den CVW-Rucksack eines Nutzers vorgenommen. Er nimmt verschiedene Dokumente auf, die anschließend in einem zweiten Raum abgelegt werden können. Die Abgabe der Lösung eines Übungsblattes lässt sich demgemäß durch Transport eines Dokumentes von z.B. dem virtuellen Lerngruppenraum in den Übungsraum oder einen speziellen Abgaberaum für die Übungsaufgaben realisieren. Hierbei wäre die Möglichkeit der Weitergabe, des Abtretens der Berechtigungen an der entsprechenden Datei im Falle der Abgabe des Übungsblattes an den Betreuer oder Übungsgruppenleiter sinnvoll und wünschenswert.

Die Weitergabe von Objekten zwischen Lernenden kann zudem über die Ablage in einem Raum erfolgen.

Synchronisieren und Kommunikation

Neben den oben skizzierten klassischen Medienfunktionen, die zu einem großen Anteil auf asynchrone Kooperationstechniken beruhen, nehmen eine Reihe von synchronen Elementen einen hohen Stellenwert in den aufgezeigten Szenarien zu einem erfolgreichen kooperativen Lernen ein.

Unter dem Blickwinkel des vornehmlichen Anwendungskontextes der Präsenzlehre wurden computergestützte synchrone Kommunikationsmechanismen nur zweitrangig untersucht. Diese sind in sämtlichen hinzugezogenen Systemen zumindest als textueller Chat und Shared Whiteboard vorhanden. Der Collaborative Virtual Workspace bietet zudem Möglichkeiten der computergestützten Audiokommunikation, welche in Anwendungsfällen der Arbeit von Lerngruppen oder der virtuellen Sprechstunde nutzbar sind. Sie spielen jedoch nur eine untergeordnete Rolle in den Bereichen der Unterstützung von Vorlesung und Übung. Dies liegt insbesondere an der sehr eingeschränkten Anzahl möglicher Teilnehmer an einer Audiokonferenz.

Umso wichtiger erscheint es, diskursive Elemente einer kooperationsunterstützenden Umgebung auf die Fähigkeiten der Persistenz und Strukturierungsmöglichkeiten der Diskussion hin zu untersuchen. So erlaubt es CVW innerhalb eines speziellen Diskussionsraumes Teile der Diskussion textuell aufzuzeichnen, was z.B. externen oder verhinderten Teilnehmern einer Übung zugute kommt. Anschließend können wichtige Elemente der Übung bzw. Vorlesung nachbereitet werden. Während der Übung werden hierzu beispielsweise wichtige Fragen oder Aussagen innerhalb des Diskussionsbereichs des virtuellen Übungsgruppenraums in Form von kurzen Textpassagen protokolliert. Der Diskussionsverlauf wird auf diese Weise in Stichpunkten dokumentiert.

Zur Reduzierung von Medienbrüchen im Bereich der Präsentation von Lehrmaterialien oder des gemeinsamen Betrachtens von Materialien in Vorlesung, Übung oder Seminarvortrag nehmen kooperative Präsentationswerkzeuge und Shared Browser einen hohen Stellenwert ein. An vielen Stellen der beschriebenen Anwendungsfälle aus Vorlesung und Übung sind Lehrmaterialien von einem Vortragenden der Gruppe zu präsentieren, wobei den Zuhörern Möglichkeiten der Annotation und des persönlichen Anfertigen von Notizen bereitgestellt werden sollen. Synonym derartiger Anwendungsszenarien ist der Seminarvortrag, in dem ein Studierender einen Themenbereich unter Einbeziehung von Folien und weiterführenden Materialien (Texte, Verweise, Grafiken) der Gruppe näher bringt. Die Präsentation findet in einem virtuellen Vortragsraum (dem Seminar-, Vorlesungs- oder Übungsraum) statt. Aktuell anwesende Lernende können der Präsentation durch Teilnahme an der entsprechenden Sitzung folgen.

Gemeinsame Werkzeuge zur Navigation wie ein Shared Browser erweitern das Nutzungsszenario z.B. durch das kooperative Betrachten eines Skriptes oder kooperatives Durcharbeiten von Netzseiten. Hier kann die Steuerung der Navigation bei verschiedenen Teilnehmern der Sitzung liegen.

Sowohl mittels TeamWave als auch Habanero sind Anwendungsfälle des synchronisierten Folienvortrags umsetzbar. TeamWave beinhaltet ein kooperatives Folienbetrachtungswerkzeug (Slide Viewer) und einen einfachen Shared Browser.

Zur Durchführung eines Seminarvortrages scheint der Shared Browser des Habanero-Systems besonders geeignet. Es wird beispielsweise ein typischer Foliensatz einer Vorlesung zur Softwareergonomie auf einem Webserver abgelegt. Anschließend werden die Folien in einer kooperativen Sitzung von einem Vortragenden vorgestellt. Die in einem elektronischen Hörsaal anwesenden Zuhörer verfolgen den Vortrag in der entsprechenden Sitzung. Die Steuerung der Browser der Zuhörer liegt bei dem Vortragenden. Durch den Einsatz eines zentralen Telepointers kann ähnlich einem Shared Whiteboard der Fokus der Zuhörer auf einen zentralen Punkt gelenkt und durch Einrahmungen, Umrandungen etc. der Vortrag weiter strukturiert werden.

In allen Anwendungsfällen ergibt sich eine prinzipielle Eignung des Konzepts des Shared Browsers bzw. des speziellen kooperativen Folienvortragswerkzeugs, wie es TeamWave bietet, für eine Umsetzung von Präsentationssituationen.

Ein kooperatives Betrachten der Folien in einem Shared Browser ähnelt der zentralen Projektion des Folienvortrages. Wie in den Szenarien gezeigt, liegt die neue Qualität in den Möglichkeiten, begleitend dem Vortrag Anmerkungen, Notizen und Annotationen an Folien anfertigen zu können. Diese Techniken sind mit beiden Werkzeugen in Ansätzen umsetzbar.

In allen Fällen erscheint der Kopplungsgrad der untersuchten Werkzeuge für die Szenarien aus Seminarvortrag und Vorlesung als zu eng gefasst. Die Steuerung des kooperativen Browsers ist von allen Teilnehmern der Sitzung gleichermaßen möglich, was

zu erheblichen Irritationen und Problemen während eines Vortrages führen kann. Ein Teilnehmer betätigt z.B. bewusst oder unbewusst die Zurück-Taste des Browsers. Hier wäre eine entkoppelbare zentrale Präsentation sinnvoll. Auch die Telepointer der einzelnen Sitzungsteilnehmer müssen während des zentralen Vortrages unsichtbar schaltbar sein. Zu viele Telepointer erweisen sich als für alle Teilnehmer irreführend und stören den Gesamt Ablauf.

Mit einem weiter gefassten Kopplungsgrad ist zudem die Möglichkeit des persönlichen Anfertigen von Notizen und Annotationen etc. intendiert: Idealerweise können Lernende Folien annotieren und sich persönliche Notizen anfertigen ohne die zentral gesteuerte Präsentation verlassen zu müssen. Dies ist innerhalb der untersuchten Systeme nur durch Öffnen eines zweiten Fensters (über Umwege) möglich. Die vorhandenen Anmerkungen und Notizen sind zudem innerhalb des Habanero-Systems nicht persistent.

Möglichkeiten der Kommunikation (Chat) über einen Textkanal begleitend der Vorlesung oder des Seminarvortrages wurden durch Ausrichtung auf die Präsenzsituation nicht weiter untersucht. Hier würden Tipgeräusche und speziell der motorische Aufwand der textuellen Kommunikation den Gesamt Ablauf erheblich stören. Aus ähnlichen Gründen wurden die Audio-Fähigkeiten des TeamWave-Systems in derartigen Szenarien ebenfalls nicht berücksichtigt.

Für die Umsetzung von spezifischen kooperativen Applikationen (Explorationen) ist Habanero prädestiniert. Dies erfordert jedoch einen erheblichen Entwicklungsaufwand in der Erstellung derartiger Applikationen.

Ergebnisse aus der Untersuchung der genannten Systeme

Resümierend ist festzuhalten, dass eine Umsetzung der Szenarien zum computergestützten kooperativen Lernen auch in Kombination von drei fortschrittlichen Systemen aus verschiedenen Gründen nicht gelingt.

Dies liegt zunächst weniger an einer mangelnden Ausrichtung der gewählten Systeme auf das Anwendungsfeld menschlichen Lernens, vielmehr werden eine Vielzahl von Medienbrüchen und technologisch bedingten Einschränkungen in der Selbstverwaltung der Studierenden sichtbar. Medienfunktionen des kooperativen Umgangs mit Materialien werden lediglich von CVW ansatzweise bereitgestellt. Diese sind jedoch sehr eingeschränkt vorhanden und unterstützen größere Lehrunterlagen in Form von Hypertextdokumenten nicht. Gleichzeitig besitzt keines der untersuchten Systeme eine gute Integration von WWW-basierten Dokumenten. Unterstützt werden Verweise auf WWW-Seiten, das Einbringen von XML- oder HTML-Dokumenten (konsistente Verweise) ist nicht vorgesehen.

Auch wenn TeamWave und Habanero gute Fähigkeiten der Annotation im Kontext eines Shared Whiteboard besitzen, sind diese im Umgang mit Materialien nur sehr eingeschränkt vorhanden. An einigen wenigen Stellen sind innovative Ansätze der Selbstverwaltung der Studierenden zu erkennen. Dies zeigt sich in den Möglichkeiten eigene Räume zu erstellen und der vereinfachten Anmeldung zu einer kooperativen Sitzung. Ein durchgängiges Konzept einer auf Selbstadministration der Studierenden abgestimmten Benutzer- und Gruppenstruktur ist jedoch nicht vorgesehen.

CVW besitzt seine Stärken in der Verbindung von Konzepten der Dokumentenverwaltung und Mechanismen der synchronen Kommunikation und der gegenseitigen Wahrnehmung. Es verknüpft die kommunikativen Fähigkeiten des MOO mit rudimentären Möglichkeiten der Bereitstellung primärer Medienfunktionen.

Habanero und TeamWave konzentrieren sich auf die Umsetzung synchroner Kooperations-situationen. Habanero ist rein sitzungsbasiert und zeigt gute Eigenschaften in der netzgestützten Präsentation von Vorlesungsinhalten sowie dem kooperativen Betrachten von Netzseiten. TeamWave entfaltet seine Stärken unter der Metapher eines Shared Whiteboard als persistente gemeinsame Zeichenfläche mit guten Fähigkeiten zur gegenseitigen Annotation. Sowohl Habanero als auch TeamWave eignen sich naturgemäß nur für Kooperationssituationen kleinerer Gruppen von Studierenden.

Basierend auf den genannten Einschränkungen verspricht eine weitere Klassifizierung und Untersuchung von Einzelsystemen keinen erheblichen Erkenntnisfortschritt. Innovative Ansätze sind in ihrem Leistungsumfang sehr heterogen, eine Bewertung und Verbindung gelingt dementsprechend nur teilweise. Keine der genannten Klassen von Systemen vermag Anteile primärer individueller und kooperativer Medienfunktionen bei gleichzeitiger Selbstorganisation der Studierenden sicherzustellen. Insofern erscheint die konzeptionelle und technische Umsetzung eines Systems zur Bereitstellung kooperativer Wissensräume viel versprechend.

Hierzu gilt es zunächst, wesentliche konzeptuelle Elemente des gewonnenen Rahmenkonzepts gegenüber dem Erkenntnisstand der Forschung abzugrenzen bzw. Überschneidungen aufzuzeigen. Die Darstellung wesentlicher Eigenschaften eines Systems zur Bereitstellung kooperativer Wissensräume in der Metapher des virtuellen Raums (Abschnitt 3.1) orientiert sich entsprechend erneut an der gefundenen Differenzierung der primären Medienfunktionen (Abschnitt 3.2). Ihre konzeptuelle Einbettung in den virtuellen Raum bedarf einer Vielzahl von Faktoren, beispielsweise einer detaillierten Berücksichtigung von verschiedenen Sichten auf den kooperativen Raum (Abschnitt 3.1.3), Benutzerrechten (Abschnitt 3.1.5) und einer spezifischen Repräsentation von Materialien innerhalb der Umgebung (Abschnitt 3.2.1).

3. Konzeptionelle Aspekte kooperativer Wissensräume

Die Analyse existierender Ansätze und Systeme unter dem Blickwinkel der vorgestellten Szenarien haben zu einer Reihe von wichtigen Erkenntnissen geführt:

Zum einen wurde deutlich, dass eine Vielzahl elaborierter Ansätze existieren, die in Einzelaspekten gute Konzepte und Ideen verdeutlichen. – Leider besteht die Schwierigkeit, diese Konzepte auf den Kontext selbstverwalteter und nutzerzentrierter kooperativer Wissensräume zu übertragen. Zum anderen verkörpern existierende Ansätze eine Vielzahl von Designkonflikten, die auch die in der vorliegenden Arbeit entwickelte Konzeption kooperativer Wissensräume prägen. So gilt es zu jedem Zeitpunkt, einen gesunden Kompromiss aus komplexen und leistungsfähigen Systemen und einer unter alltäglichen Bedingungen praktikablen Lösung zu schaffen. Beispielsweise existieren weit entwickelte, umfangreiche Modelle von Benutzerrechten. Diese können kooperativen Lernprozessen auch hinderlich sein, wenn sie nicht die Selbstorganisation der Lernenden unterstützen und für Nutzer verständlich umsetzbar sind. Die Konzeption kooperativer Wissensräume versucht hier einen gesunden Mittelweg zu gehen.

Kooperative Wissensräume basieren auf der Einbettung primärer Medienfunktionen in den virtuellen Raum. Speziell die Anforderung der Realisierung *kooperativer* Medienfunktionen macht ihre differenzierte Umsetzung und die Berücksichtigung einer Vielzahl technischer und konzeptioneller Randbedingungen notwendig.

Das vorliegende Kapitel untersucht die Frage nach den grundlegenden Strukturen kooperativer Wissensräume sowie ihrer Eigenschaften und Nutzungsmöglichkeiten. Ergebnis ist die Repräsentation von Materialien als Objekte und Attribute. Basierend auf den primären Medienfunktionen des Kreierens, Arrangierens, Verknüpfens, Übertragens, Synchronisierens und Zugreifens werden Modelle von Objekten und Zugriffsrechten entwickelt. Es gilt zu untersuchen, welche Sichten auf Objekte zu realisieren sind, welche Möglichkeiten unterschiedlicher Kopplungsgrade der Sichten auf gemeinsam genutzte Materialien existieren und wie Handlungen an Materialien in virtuelle Wissensräume einzubetten sind. Dabei sind die Medienfunktionen des Arrangierens, Verknüpfens und Übertragens in Systemfunktionalitäten des Kopierens, des Erzeugens von Referenzen oder des Zugriffs auf Objekte zu übertragen. Insofern präsentiert die vorliegende Rahmenkonzeption eine konsequent medientechnische Perspektive. Es werden ausschließlich Konzepte betrachtet, die der Umsetzung der primären Medienfunktionen dienen. Verfahren und Ansätze, bei denen die Strukturierung von Lernprozessen innerhalb der Lernumgebung (Turn-Taking, Floor Control etc.) im Vordergrund stehen,

sind den sekundären Medienfunktionen zuzuordnen und werden nicht oder nur am Rande berücksichtigt.

Ausgehend von der Metapher des kooperativen Raums (Abschnitt 3.1) werden zunächst einige Grundmerkmale kooperativer Wissensräume diskutiert. Hierzu zählen das Kriterium der Persistenz enthaltener Objekte (Abschnitt 3.1.2), verschiedene Kopplungsgrade der Sichten sowie mögliche grafische Repräsentationsformen des kooperativen Raums (Abschnitt 3.1.3). In einem nächsten Schritt werden Elemente der gegenseitigen Wahrnehmung und Koordination integriert, also die Medienfunktion des Synchronisierens berücksichtigt (Abschnitt 3.1.4). Abschnitt 3.1.5 beschreibt die Entwicklung eines Modells der Zugangskontrolle zu virtuellen Räumen und Materialien (Gruppen und Rechte). Es folgt die Darstellung grundsätzlicher Interaktionstechniken zwischen Lernenden untereinander und zwischen Lernenden und den Materialien der Lernumgebung (Abschnitt 3.2). Strukturiert entlang der primären Medienfunktionen werden Handlungen an Objekten und Attributen (z.B. Formen der Annotation von Materialien) konzeptionell in virtuelle Wissensräume eingebettet und in ein entsprechendes Objektmodell überführt. Hierzu zählen Themenbereiche wie das Erzeugen von kooperativen Objekten mit entsprechenden Attributen, das Gruppieren von Objekten im Raum, das Erzeugen von Verweisen und vor allem kooperative Interaktionsformen wie das Einbringen von Lehrmaterialien in die Lernumgebung, der Austausch von Materialien zwischen Lernenden sowie kooperatives Betrachten und Präsentieren. Abschließend werden verschiedene Kommunikationsmechanismen in das Rahmenkonzept eingefügt und insbesondere Möglichkeiten der Strukturierung von Kommunikation in Relation zu den Materialien der Lernumgebung dargestellt (Abschnitt 3.3).

3.1 Der kooperative virtuelle Raum

Zentrale konzeptionelle Grundlage der skizzierten kooperativen Lern- und Arbeitsumgebung ist der virtuelle Raum. Im Folgenden werden neben einer genaueren Betrachtung der zugrunde liegenden Metapher und mögliche Sichten (Repräsentationsformen) insbesondere Rechte, Rollen und Gruppen als wesentliche Elemente kooperativer Interaktionsformen des virtuellen Raums dargestellt.

Neben der technischen und medienfunktionalen Sichtweise existiert eine soziale Komponente des kooperativen virtuellen Raums. Speziell aus dem Bereich der MUDs und MOOs und der CVEs (vgl. Abschnitt 1.7.1) ist bekannt, dass der virtuelle Raum ein soziales Kraftfeld aufweist. Schon in den Anwendungen früherer MUDs als reine Rollenspiele hat sich gezeigt, dass sich Spieler zu bestimmten Zeiten in festgelegten virtuellen Räumen aufhalten, um andere Spieler zu treffen und sich auszutauschen. Innerhalb des virtuellen Raumes entstehen hierbei eigene soziale Konventionen und Regeln, aber auch ein Zusammengehörigkeitsgefühl und ein Gefühl des Zuhause-seins.

In dieser Form besitzt der virtuelle Raum den Charakter eines sozialen Treffpunktes.⁸⁷ Die vorliegende Arbeit wird den Anspruch virtueller Wissensräume als soziales System nicht weiter intensivieren. Themen wie Interaktions- und Kommunikationsformen, soziale Konventionen und verbale und nonverbale Kommunikationsmechanismen

⁸⁷ Eine Vielzahl von Forschern haben die sozialen Interaktionen von Nutzern innerhalb von virtuellen Umgebungen und virtuellen Gemeinschaften untersucht (vgl. [Turkle 1995], [Turkle 1998] oder [Becker & Mark 1998]). Turkle spricht sogar von „social MUDs“ und definiert sie primär über Interaktionen zwischen Nutzern und der den Nutzern verliehenen Fähigkeit die virtuelle Welt eigenständig zu gestalten (vgl. [Turkle 1998, S. ix]).

werden unter dem Anspruch der Konzeption eines primär technischen Rahmenkonzepts nicht weiter berücksichtigt.

Dennoch ist es für die Gestaltung eines kooperativen Lernsystems wichtig, die Metapher des virtuellen Raumes ernst zu nehmen und auch soziale Aspekte des Lernens zu berücksichtigen. Der virtuelle Lern- und Arbeitsraum ist mit einer konkreten Bedeutung und einer Botschaft zu versehen. Eine ganze Anzahl von Autoren betonen in diesem Zusammenhang, dass sich der virtuelle Raum zu einem „place“⁸⁸ entwickelt, also einem Ort, der Menschen das Gefühl des Zuhauseesens vermittelt (vgl. [Harrison & Dourish 1996]). Greenberg und Roseman stellen eine Anzahl von aus ihrer Sicht wichtigen Kriterien auf, um den virtuellen Raum zum „social place“ [Greenberg & Roseman 1998, S.21ff.] werden zu lassen.⁸⁹ Schon Curtis hat diesen Effekt, dass sich Nutzer des MUD an bestimmten, belebten Stellen treffen, soziale Schwerkraft getauft.⁹⁰

Auf das Feld des menschlichen Lernens bezogen erscheint der Anspruch des virtuellen Raums als Zusammenhalt und „soziale Klammer“ etwas überbewertet. Im Folgenden wird das Verständnis des virtuellen Raums primär auf die Idee kooperativer Wissensräume reduziert, also auf einen gemeinsamen Ort kooperativen Handelns verdichtet: Primär dient der Raum als konzeptionelle Einheit, um Lernende an einem definierten Ort und in einem abgegrenzten Kontext die Möglichkeit zu eröffnen, aktiv mit Materialien zu arbeiten, zu interagieren und mit anderen Lernenden zu kommunizieren.

Es ist insbesondere diese Kommunikationsmöglichkeit in Verbindung mit Medienfunktionen (einer medientechnischen Perspektive), die den virtuellen Raum um soziale Komponenten aufwertet.

3.1.1 Raummetapher

In der Literatur und im praktischen Einsatz finden sich eine große Anzahl von Ansätzen und konkreten Systemen, die sich auf die Metapher eines virtuellen Raums berufen. Hierbei wird nur in wenigen Fällen eine präzise Abgrenzung zwischen einer wirklichen Raummetapher und einer Schreibtischmetapher vorgenommen.⁹¹

Erste Ansätze, die dem hier entwickelten Verständnis eines virtuellen Raums in weiten Teilen entsprechen, stellen die MUDs und MOOs dar. Der Raum wird als eine virtuelle Umgebung verstanden, der mit anderen virtuellen Räumen über Verbindungen (Exits) verknüpft ist und neben Personen (Avataren) auch verschiedene Objekte enthalten kann. Räume besitzen meist eine reduzierte, betont einfache Repräsentation, sind in ihrer Anzahl und thematischen Ausrichtung nicht begrenzt und können beliebig miteinander verknüpft werden. Dies geschieht vorwiegend angelehnt an geografische, realweltliche Gegebenheiten. Spätere Systeme greifen diese Idee auf und reduzieren sie auf

⁸⁸ Miao und Haake definieren in ihrem „virtual institute model“ [Miao & Haake 2001, S. 5ff.] vier grundlegende Einheiten des virtuellen Raums, den „place“, „agent“, „document“ und „tool“. Hierbei wird der „place“ als Ort definiert, an dem zum einen verschiedene Objekte abgelegt werden können, zum anderen aber auch soziale Interaktionen stattfinden.

⁸⁹ Ähnliche Aspekte untersuchen Bannon und Bødker in ihren Forschungen zur Bereitstellung von gemeinsamen Informationsräumen, dem „Common Information Space“ [Bannon & Bødker 1997].

⁹⁰ „[...] players are thus more attracted to more populated areas. I call this phenomenon ‘social gravity’; it has a real-world analog in the tendency of people to be attracted to conspicuous crowds, such as two or more people at the door of a colleague’s office.“ [Curtis 1992, S. 12ff.]

⁹¹ Eine viel zitierte Arbeit ist in diesem Zusammenhang das Rooms-System von Henderson und Card (vgl. [Henderson & Card 1985]). Es handelt sich um einen der ersten Versuche, eine raumbasierte Arbeitsumgebung bereitzustellen, die aus heutiger Sicht jedoch einer Schreibtischmetapher zuzuordnen wäre.

konkrete Anwendungsszenarien; sie sind in ihren Handlungsmöglichkeiten innerhalb des virtuellen Raums jedoch dann zumeist recht eingeschränkt.⁹²

Auf die Konzeption kooperativer Wissensräume übertragen lässt sich aus den zugrunde liegenden Einsatzszenarien ein konkretes Verständnis des virtuellen Lernraums entwickeln. Dieses Verständnis der Metapher des virtuellen Lern- und Arbeitsraums ist durch eine Anzahl von Merkmalen gekennzeichnet:

- *Persistenz*: Virtuelle Lernräume benötigen eine Dauerhaftigkeit und müssen auch über einen längeren Lernprozess in ihrer Gestalt mit den enthaltenen Objekten unverändert Bestand haben.
- *Verknüpfbarkeit*: Räume müssen in ihrer Anordnung frei gestaltbar sein, d.h. Verknüpfungen zwischen Räumen werden im Rahmen der Nutzung erstellt und repräsentieren eine semantische Struktur (vgl. Abbildung 2-3).
- *Kreierbarkeit*: Räume müssen durch einfache Prozesse, auch von den Lernenden selbst erzeugt und ggf. wieder entfernt werden können.
- *Strukturierungsfunktion*: Der Raum selbst wirkt als Instrument der Strukturierung für enthaltene Objekte. Hierzu zählen Dokumente, aber auch Personen, die sich innerhalb des virtuellen Raumes aufhalten. So wirkt sich beispielsweise eine textuelle Kommunikation zunächst auf den Bereich des jeweiligen Raumes aus.
- *Universelle Zugreifbarkeit*: Der virtuelle Raum besitzt eine fest verankerte Position im Netz, d.h. er ist von verschiedenen Lernorten aus zugreifbar. Die Struktur des Raumes ist in diesem Zusammenhang zunächst unabhängig von der gewählten Repräsentation durch ein entsprechendes Zugriffswerkzeug (Client-/Browser).
- *Zugriffsschutz*: Der Zugang zu einem virtuellen Lernraum und die in ihm enthaltenen Objekte lässt sich gezielt einschränken.
- *Gegenseitige Wahrnehmung (Awareness)*: Es existieren Mechanismen, die Teilnehmern eines virtuellen Raumes über die Anwesenheit anderer Personen informieren und Möglichkeiten bieten, evtl. etwas über die Aktivitäten dieser Personen zu erfahren (Koordinationsfunktion).
- *Kooperative Ausübung der Medienfunktionen*: Der virtuelle Raum ist die konzeptionelle Grundlage zu einer Bereitstellung von kooperativen Medienfunktionen. Lernende müssen eigene Objekte erzeugen, Verknüpfungen zwischen Objekten herstellen können, diese arrangieren und transportieren können. Weiterhin müssen Handlungen zwischen Lernenden in Bezug auf die jeweiligen Objekte synchronisierbar sein.

Unter den obigen Kriterien ist speziell die Verknüpfbarkeit ein wesentliches Element der konzeptuellen Ausgestaltung virtueller Wissensräume. Verbindungen zwischen virtuellen Räumen lassen sich zunächst über eine Enthaltensein-Beziehung ausdrücken, d.h. das Erzeugen eines neuen Raums macht ihn zum Unterraum des entsprechenden Eltern-Raums. In dieser Weise kann ein Raum Unterraum eines beliebigen anderen Raums sein. Räume beinhalten Materialien oder Ordner (Container). Diese kapseln wiederum die Dokumente eines Raums (vgl. Abbildung 3-1).

⁹² Beispiele für die Integration verschiedener Materialien in den virtuellen Raum sind zu finden bei [Roseman & Greenberg 1996a], [Sohlenkamp & Chwelos 1994], [Miao & Haake 2000, S. 5ff.] oder [Spellman et al. 1997, S. 198].

Der im Rahmen dieser Arbeit verfolgte Ansatz kooperativer Wissensräume sieht vor, für jeden Nutzer und jede Nutzergruppe einen eigenen Arbeitsraum (Workroom) bereitzustellen. Dieser ist Wurzel eines Baumes mit weiteren Unterräumen, Containern und Dokumenten. Über Türen oder „Exits“ können Räume miteinander verbunden werden. (Diese Verbindungen können nur in eine Richtung begehbar oder bidirektional angelegt sein.)

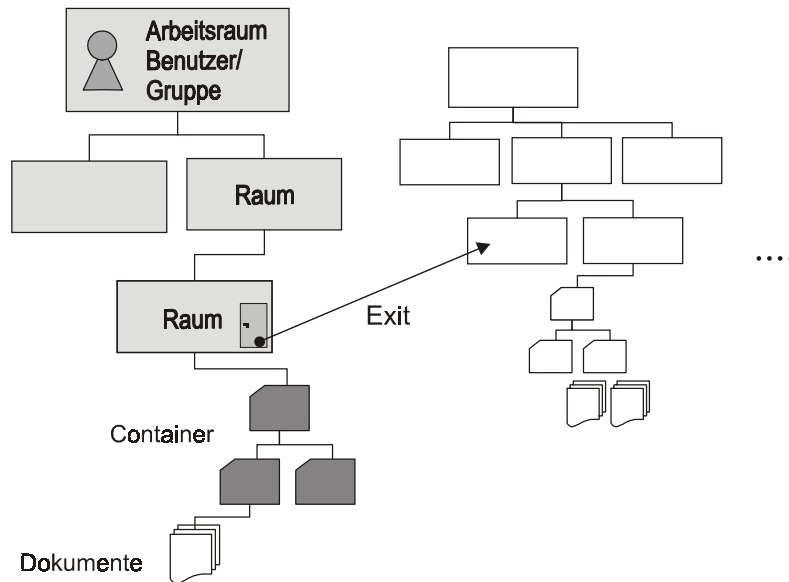


Abbildung 3-1: Raumstruktur

3.1.2 Persistenzeigenschaft

Neben der Persistenz des Raums an sich ist die Persistenz der enthaltenen Materialien und Werkzeuge wichtige Grundeigenschaft virtueller Wissensräume. Zentrale Eigenschaft primärer Medienfunktionen ist damit auch die Sicherstellung der Persistenz der von Lernenden erzeugten, arrangierten und verknüpften Materialien.

Die verfolgte Konzeption grenzt sich in dieser Eigenschaft von einer ganzen Klasse von Systemen ab, die Persistenz für virtuelle Räume nur über eine Sitzung hinweg bereitstellen.⁹³

Ziel und zentrale Botschaft der Verschmelzung von ereignisbasierten MUD/MOO-Kommunikations- und Kooperationsformen mit den Eigenschaften netzgestützten Dokumentenmanagements ist damit die Integration von Persistenzeigenschaften in den virtuellen Raum. Die Persistenz innerhalb des virtuellen Raumes muss einer Anzahl von Anforderungen genügen, die in dieser Form von existierenden Systemen nur in Teilaspekten erfüllt werden.⁹⁴

Wesentliche Eigenschaften der Persistenz sind damit:

- *Nachhaltigkeit der Objekt-Persistenz über einen längeren Nutzungszyklus:* Objekte (Lehrmaterialien) müssen durch Ablage in einem virtuellen Lernraum eine

⁹³ Ein Vertreter dieser Klasse ist beispielsweise das Habanero-System, vgl. Abschnitt 2.7.2.

⁹⁴ Einen ähnlichen Standpunkt verfolgen Spellman et al.; sie führen den Begriff der „Place-based systems“ [Spellman et al. 1997, S. 198] ein, also Systeme, die sich auf die Bereitstellung eines persistenten Raums konzentrieren. Unter den Charakteristika für eine derartige Umgebung steht die Persistenz neben der Unabhängigkeit des Ortes („location transparent“, universelle Zugreifbarkeit) an oberster Stelle.

dauerhafte Speicherung erfahren. Auch zu erheblich späteren Zeitpunkten oder in andern Kontexten müssen diese unverändert zur Verfügung stehen.

- *Persistenz von Räumen*: Räume müssen genau wie Objekte eine Persistenz besitzen. Diese erstreckt sich auf die enthaltenen Objekte und ihre semantische Struktur. Anders als in vielen Lernumgebungen, die den Lernenden gezielt durch Materialien führen oder auch in einigen MUDs und MOOs, wird die Raumdarstellung nicht regelmäßig in den Ursprungszustand zurückversetzt. Sie repräsentiert vielmehr einen von den Lernenden geschaffenen Zustand.
- *Transparenz der Persistenz*: Persistenz muss für die Nutzer/Lernenden transparent ohne explizite Handlungsschritte erzeugt werden. Sämtliche Medienfunktionen, wie das Erzeugen neuer Objekte, die Ablage existierender Objekte und beliebige Objektmodifikationen müssen sich, ohne notwendige Aktionen des Speicherns etc. direkt auswirken. Dies kann sich auf das entsprechende Objekt selbst oder auf eine Version des Objektes im Falle einer Versionsverwaltung beziehen.
- *Skalierbarkeit der Persistenz in Bezug auf Objektanzahl und Größe*: Die Anzahl und Größe der Objekte darf nur durch praktische Handhabbarkeit bzw. durch die zur Verfügung stehenden Ressourcen begrenzt sein. Über eine Erweiterung dieser Ressourcen muss die Objektgröße und Objektanzahl skalierbar sein.

3.1.3 Grafische Repräsentationsformen

Die Metapher des kooperativen virtuellen Raums ist maßgeblich durch die für die jeweiligen Benutzer erzeugte Sicht auf diesen geprägt.

Im Gegensatz zu einer alleinig genutzten Benutzerschnittstelle eines herkömmlichen Softwaresystems gestaltet sich die kooperativ genutzte Darstellung des virtuellen Raums um einiges schwieriger. Es gilt den Raum selbst geeignet und dem Anwendungsfall entsprechend darzustellen, verschiedene Kopplungsgrade der Sichten der jeweiligen Nutzer zu unterscheiden, Informationen über die Anwesenheit der Nutzer und ihrer Schritte in der Umgebung bereitzustellen und die vorhandenen Objekte zu visualisieren. Generell muss ein Client Sichten auf gemeinsame Objekte in Abhängigkeit der aktuellen Zugriffsbedingungen (Rechte des Benutzers, Position des Benutzers innerhalb des Systems) bereitstellen (vgl. [Greif & Sarin 1987, S. 197ff.]). Wie Trevor et al. richtig feststellen, sollten dies in jedem Fall alternative Sichten auf ein Objekt sein [Trevor et al. 1994, S. 220ff.]. Ein Client muss also nicht nur verschiedene Sichten auf den Inhalt der Objekte bieten, sondern auch Informationen über das Objekt selbst (Awareness-Informationen) und über den Zustand des Objektes bereitstellen. (Wer greift auf ein Objekt zu, welche Umgebung, welche Position besitzt ein Objekt etc.)

Kopplungsgrade, Sichten (Synchronisieren)

Das Akronym WYSIWIS („What You See Is What I See“) [Stefik et al. 1987] steht für die Kopplung von Sichten von Koperationspartnern auf gemeinsame Objekte. Motiviert durch frühe Groupware-Applikationen, die meist durch einen Application-Sharing-Ansatz geprägt waren, entwickelten sich Ideen des strict-WYSIWIS, d.h. sämtliche Interaktionen an einem Objekt werden synchron an alle Teilnehmer einer kooperativen Sitzung (Session) übermittelt. Dies bezieht sich insbesondere auf das Rollen (Scrollen) von Fenstern, gekoppelte Mauszeiger (Telepointer) und das Bewegen von Materialien. Jeder Teilnehmer erhält eine identische Sicht auf die manipulierten Materialien.

Im Gegensatz hierzu bildet das relaxed-WYSIWIS eine Entkopplung der spezifischen Sichten auf den gemeinsamen Datenraum (die gemeinsamen Materialien). Es

werden zwar ebenfalls alle Aktionen der Kooperationspartner übermittelt, diese können sich jedoch persönliche Sichten schaffen (beispielsweise eigene Bildschirmausschnitte wählen) und sich innerhalb eines kooperativ editierten Textes auf eine bestimmte Position bewegen. Eine Sondervariante des relaxed-WYSIWIS stellen Beaudouin-Lafon und Karsenty mit ihrem Konzept des Time-Relaxed-WYSIWIS [Beaudouin-Lafon & Karsenty 1992, S. 172ff.] vor. Mit der Privatsphäre des einzelnen Benutzers im Blick, führen die Autoren einen Kooperationsmodus ein, in dem Benutzer für eine gewisse Zeit persönlich an Objekten arbeiten und diese modifizieren können, erst nach abgeschlossener erfolgreicher Modifikation wird ihr Ergebnis anderen Kooperationspartnern mitgeteilt. Zu diesem Zeitpunkt werden die entsprechenden Sichten der anderen Kooperationspartner aktualisiert. Eine derartige Vorgehensweise ist damit einer Form der asynchronen Arbeit in Verbindung mit der Bereitstellung geeigneter Aktivitäts-Wahrnehmungsinformationen zuzuordnen.

Historisch betrachtet besitzt die Erforschung der Verknüpfung von synchronen und asynchronen Arbeitsformen eine lange Tradition, wird jedoch nur von wenigen Autoren als wichtiges Problem der Kooperationsunterstützung erkannt.⁹⁵ Zumeist sind existierende Ansätze alleinig durch ihre Eigenschaften der Bereitstellung von Persistenz entweder den asynchronen oder rein synchronen Systemen zuzuordnen.

Interessanterweise entwickelten sich schon innerhalb textueller MUDs Mechanismen der Verschmelzung von synchronen und asynchronen Interaktionstechniken. Während im MUD eine Konversation (Chat) meist synchron innerhalb eines Raumes stattfindet, lassen sich durch einfache Mechanismen auch asynchrone Elemente (Mitteilungen) in die Umgebung integrieren.⁹⁶ So bietet das MUD neben der Qualität der synchronen Kommunikation und dem daraus resultierenden Gefühl der gegenseitigen Nähe (Co-Presence) auch die Möglichkeit einer effektiven Kommunikation in temporärer Abwesenheit.

Die Einbettung von synchronen und asynchronen Interaktionstechniken in kooperativen Wissensräumen muss sehr sorgsam vorgenommen werden.⁹⁷ Die im Rahmen dieser Arbeit entwickelte Konzeption schränkt den Einsatz synchroner Elemente gezielt ein. Die Medienfunktion des Synchronisierens wird speziell in Verbindung mit der Kopplung von Raumdarstellungen oder Szenarien der Nutzung von Shared Whiteboards bereitgestellt – eine durchgängige Kopplung aller Medienfunktionen (z.B. kooperatives Editieren) erscheint in der Praxis (speziell in mittleren bis großen Gruppen von Lernenden) nur wenig realisierbar und sinnvoll.

Darstellungsformen

Visualisierungstechniken des virtuellen Raums spielen für die Einbettung in reale Lernprozesse eine entscheidende Rolle. Es finden sich eine Vielzahl unterschiedlicher An-

⁹⁵ Vgl. hierzu auch den von Ellis et al. im Rahmen ihrer Arbeiten zum kooperativen Editor GROVE entwickelte Begriff eines „shared context“ [Ellis et al. 1991, S. 45]. Die Entwickler unterschieden sehr differenziert verschiedene (private, gemeinsame und öffentliche) Sichten auf einen gemeinsamen Kontext. (Zu einem ähnlichen Ansatz vgl. auch [Shiozawa et al. 1999].)

⁹⁶ So berichten Churchill und Bly von der regen Nutzung des „poking“ [Churchill & Bly 1999, S. 43], d.h. eines Mechanismus, eine direkte Mitteilung im Falle der Abwesenheit eines Nutzers hinterlassen zu können. Dieser erhält bei Rückkehr in das MUD eine Aufstellung der Kommunikationspartner und der Versuche, die betreffende Person zu kontaktieren.

⁹⁷ Sohlenkamp und Chwelos haben durch ihren Prototypen der Arbeitsumgebung DIVA [Sohlenkamp & Chwelos 1994] gezeigt, dass eine Integration von synchronen und asynchronen Arbeitsformen durch gut gewählte Metaphern (Ablegen und Bearbeiten von Dokumenten auf virtuellen Tischen) möglich ist.

sätze, die zugleich eine Reihe von Designkonflikten deutlich machen. Die vorliegende Arbeit bietet durch ein geeignetes Rahmenkonzept die Möglichkeit der Umsetzung (und Erprobung) verschiedener Alternativen in der Darstellung virtueller Wissensräume, trifft also zunächst keine Festlegung bei der Visualisierung und Interaktionsform. Die folgenden Abschnitte geben einen kurzen Überblick möglicher Darstellungsformen. Zu unterscheiden sind textuelle, eindimensionale, zweidimensionale und dreidimensionale Darstellungsformen des virtuellen Raums.

Verschiedene Darstellungen können in der vorgestellten Architektur kooperativer Wissensräume parallel, dem Anwendungsfall entsprechend, koexistieren (vgl. hierzu die Architektur des sTeam-Systems in Kapitel 5).⁹⁸ So kann beispielsweise ein und derselbe Lernraum von verschiedenen Nutzern zur selben Zeit durch unterschiedliche Zugangswerkzeuge betrachtet werden.

Textuelle Darstellungen des virtuellen Raums sind historisch bedingt durch MUDs und MOOs entwickelt. Der Raum selbst, wie auch sämtliche in ihm enthaltenen Objekte, werden in Textform beschrieben. Über Kommandos lassen sich beispielsweise Listen des Inhalts (Inventory) eines Raumes erzeugen. Wie schon in der Analyse der MUDs und MOOs gezeigt, setzen eine ganze Reihe von Umgebungen textuelle Darstellungsformen des virtuellen Raums für die Unterstützung realer Kooperations- und Lernprozesse ein. Derartige Ansätze nutzen das MUD oder MOO entweder als reinen Treffpunkt bzw. Kommunikationsplattform oder vermitteln Wissen in Form von Rollenspielen oder Erkundungswelten (vgl. [Hill & Slator 2000]). Formen textueller Objektbeschreibungen sind in der vorgestellten Konzeption und angestrebten technischen Lösung umsetzbar, zielen jedoch nicht auf die primäre Ausrichtung der vorliegenden Arbeit.

Eindimensionale Darstellungen des virtuellen Raumes finden sich in der Visualisierung von Auflistungen⁹⁹, wie sie z.B. die sTeam-Webschnittstelle bietet. Hier werden die Objekte des Raumes in Form einer serverseitig generierten Webseite dargestellt. Durch Aktivieren von Kommandos (hierzu wird eine Menüstruktur abgebildet) kann navigiert werden bzw. können elementare Interaktionen innerhalb des Raumes ausgeübt werden (vgl. die Realisierung der sTeam-Webschnittstelle in Abschnitt 5.3.2). Wichtiges Unterscheidungsmerkmal gegenüber textuellen, aber auch zwei- und dreidimensionalen Darstellungsformen ist, dass eine Web-Schnittstelle technisch bedingt keinerlei synchrone Elemente aufweist. Der Kopplungsgrad der Sichten der Teilnehmer des virtuellen Raums ist sehr gering. Informationen über Veränderungen am Raum werden erst durch eine explizite Aktualisierung der Benutzerschnittstelle, z.B. durch einen Navigationsschritt oder durch eine Interaktion vorgenommen. So kann beispielsweise nicht automatisiert eine Aktualisierung der Awareness-Information im Falle des Erscheinens eines neuen Lernenden innerhalb des Raumes stattfinden. Strukturierungen bezüglich der Objekte eines Lernraumes lassen sich durch Ordner (Container), also Kapselungen von Objekten innerhalb anderer Objekte verwirklichen.

Die Integration von eindimensionalen Darstellungen des virtuellen Raums in die Konzeption eines kooperationsunterstützenden Systems ist insofern unumgänglich, da

⁹⁸ Architektonisch betrachtet realisiert der kooperative sTeam-Server die Objektstruktur eines virtuellen Raumes mit den in ihm enthaltenen Objekten. Zugangswerkzeuge (Clients) wie ein Webbrowser oder auch spezielle kooperative Clients erzeugen lediglich verschiedene Sichten auf virtuelle Räume und schließen sich gegenseitig nicht aus.

⁹⁹ Eindimensionale Darstellungen von virtuellen Räumen finden sich in einer Vielzahl von Ansätzen, wie z.B. in dem BSCW-System (vgl. [Bentley & Appelt 1997]). Hier wird jedoch nicht von virtuellen Räumen, sondern von Arbeitsbereichen gesprochen.

notwendige externe Schnittstellen, wie z.B. eine FTP- oder WebDAV-Schnittstelle¹⁰⁰ eine Verzeichnisstruktur eines sTeam-Raums erzeugen. Eine verzeichnisorientierte „Ordnung“ von Objekten hat sich historisch bedingt durch die Verschmelzung von klassischen Betriebssystemparadigmen mit dem ebenfalls verzeichnisorientierten Konzept des WWW als Standard für die Ablage von Dokumenten im Netz durchgesetzt. Eine virtuelle Lernumgebung wie sTeam muss trotz ihres raumbasierten Ansatzes eine Kompatibilitätsschnittstelle hierzu aufweisen. Diese erlaubt z.B. die direkte Speicherung von Objekten aus herkömmlichen Applikationen auf einem sTeam-Server.

Zweidimensionale Darstellungen des virtuellen Raumes erlauben über lineare Listen oder Ordnerstrukturen hinausgehende Darstellungen. In diesem Zusammenhang ist das Konzept der semantischen Karte [Klemme et al. 1998] hin zum Konzept des semantischen Raums [Hampel & Selke 1999] entwickelt worden. Ziel ist es, semantische Beziehungen zwischen Objekten des virtuellen Raums abzubilden und diese Darstellung von den Lernenden arrangierbar zu gestalten. Objekte besitzen zu diesem Zweck Attribute, wie beispielsweise eine Position innerhalb des Raums, die von den Lernenden genutzt werden, um z.B. semantische Nähe durch räumliche Zuordnung oder Zuständigkeiten einer Person zu Dokumenten durch die Position eines Avatars in der Nähe von Dokumenten auszudrücken. Verstärkt werden kann dieser Effekt durch eine hinterlegte grafische Darstellung abstrakter Beziehungen zwischen der Raumtopologie und den arrangierten Objekten (vgl. [Preiß 2000, S. 41ff.]).

Der Kopplungsgrad eines zweidimensionalen Clients ist durch die gewählte technische Umsetzung gegenüber einer eindimensionalen Darstellung zumeist enger möglich, muss jedoch auf das konkrete Lernszenario abgestimmt werden. Wie in Kapitel 2 gezeigt, wird in der Konzeption virtueller Wissensräume von einer primär asynchronen Kooperationsform ausgegangen, welche jedoch in Bezug auf die Kommunikation und Interaktion auf einem Shared Whiteboard auch synchrone Elemente enthält.

Dreidimensionale Darstellungen werden primär in so genannten Avatarwelten [Damer 1998] oder auch CVEs [Churchill et al. 2001] gewählt. Ziel ist es, den Benutzern durch eine möglichst detailgetreue Darstellung des Raumes realweltliche Navigationswege und Interaktionsformen zu erlauben. Speziell auf den Aspekt der räumlichen Anordnung von Objekten in der dreidimensionalen Darstellung einer Arbeitsumgebung konzentrieren sich die Arbeiten von Büscher et al. in dem Manufaktur-System [Büscher et al. 2000]. Es verfolgt einen Application-Sharing-Ansatz und sucht gezielt nach dreidimensionalen Darstellungen eines kooperativen Arbeitsplatzes („workspaces“). Insbesondere die Entwicklung und Manipulation (z.B. Zoom) von persönlichen und gemeinsamen Sichten steht hierbei im Vordergrund.¹⁰¹

Die vorliegende Konzeption konzentriert sich durch ihre Fokussierung auf digitale Materialien und Szenarien der Präsenzlehre weniger auf die Bereitstellung von dreidimensionalen Zugangswerkzeugen. Diese können jedoch in Kombination mit einem klassischen zweidimensionalen Client oder einer Web-Schnittstelle interessante Möglichkeiten der Interaktion im virtuellen Raum bieten. Erfahrungen einer studentischen Projektgruppe [Hampel & Keil-Slawik 2001c] haben gezeigt, dass sich mittels einer isomet-

¹⁰⁰ Web Distribution and Versioning Protokoll, vgl. Abschnitt 5.2.8.

¹⁰¹ Weitere Visualisierungstechniken des virtuellen Raums wie die Verschmelzung von persönlichen und gemeinsamen Arbeitsräumen in dem von Shiozawa et al. vorgestellten Ansatz [Shiozawa et al. 1999] oder die Visualisierung einer Büroumgebung [Sohlenkamp & Chwelos 1994] werden im Folgenden nicht weiter berücksichtigt.

rischen dreidimensionalen Darstellung des Raumes eine große Anzahl von Objekten (Dokumenten) anordnen lassen und sich durch Nutzung der Räumlichkeit speziell die Kommunikation zwischen Avataren strukturieren lässt.¹⁰²

3.1.4 Awareness und Koordination

Gegenseitige Wahrnehmung (Awareness) und Koordination sind wesentliche Eigenschaften des virtuellen Raums. In ihrer konzeptionellen Ausgestaltung sind diese den sekundären Medienfunktionen zuzuordnen. Gleichzeitig beeinflussen Methoden der Awareness einen wesentlichen Anteil der strukturierten und koordinierten Ausübung der primären Medienfunktionen in kooperativen Formen der Zusammenarbeit und werden aus diesem Grunde als Teil der Medienfunktion des Synchronisierens verstanden. In den folgenden Abschnitten werden grundlegende Funktionen gegenseitiger Wahrnehmung im Kontext kooperativer Wissensräume diskutiert. Hierzu zählen die Untersuchung von Awareness-Informationen im WWW, Formen der Aktivitäts-Awareness und die Funktion von Awareness in Bezug auf die Strukturierung von Interaktion und Kommunikation.

Gegenseitige Wahrnehmung

Die Information der gegenseitigen Wahrnehmung ist für das Zustandekommen und die Koordination eines kooperativen Lernprozesses grundlegend notwendig. Verschiedene Rollen und Zugriffsrechte lassen sich erst durch ausreichende Awareness-Informationen sinnvoll nutzen.

Die im Rahmen dieser Arbeit dargestellte Form kooperativer Wissensräume konzentriert sich, wie aus den aufgezeigten Szenarien ersichtlich, primär auf die Unterstützung von Situationen der Präsenzlehre. Aus diesem Grunde ist zumeist eine gegenseitige Wahrnehmung der Anwesenheit von Mitlernenden und in gewissem Maße ihrer Aktionen (Kommunikation) gegeben. Die Bereitstellung von Awareness-Informationen innerhalb von kooperativen Lernräumen ist jedoch auch für Präsenzsituationen hilfreich. Beispielsweise lässt sich die Information, welcher Lernende sich in welchem virtuellen Lernraum aufhält, ohne jegliche Awareness-Mechanismen nur über explizite Nachfrage (z.B. innerhalb eines Kommunikationskanals) ermitteln.

Im Falle der Integration von nicht physisch anwesenden Lernenden in den Kooperationsprozess und natürlich in allen Situationen von virtuellen Treffen (virtuelle Sprechstunde, virtuelles Treffen zur Besprechung einer Übungsaufgabe etc.) sind Informationen der gegenseitigen Wahrnehmung unersetzlich. Trotzdem ist anzumerken, dass die vorliegende Arbeit das Problem der Bereitstellung von verschiedenen Formen der Awareness nicht überbewerten möchte. Als primär asynchron genutztes System lassen sich durch die Lernenden selbst ein Großteil von Wahrnehmungsinformationen z.B. durch Annotationen oder kurze Mitteilungen bereitstellen.

Der folgende Abschnitt stellt verschiedene Möglichkeiten des Verfügbarmachens von Wahrnehmungsinformationen in virtuellen Räumen vor, spiegelt sie an dem Stand der Forschung auf diesem Gebiet und versucht Rückschlüsse und Anforderungen an die Konzeption virtueller Wissensräume zu formulieren.

Awareness lässt sich in eine ganze Reihe von Arten und Formen der gegenseitigen Wahrnehmung unterteilen: Zunächst ist es als Vorbedingung für einen kooperativen

¹⁰² Vgl. hierzu auch die Arbeiten zur MASSIVE-Umgebung [Greenhalgh & Benford 1995], in der räumliche Nähe zwischen Avataren zur Strukturierung von Audio-Kommunikation genutzt wird.

Lernprozess unerlässlich zu erfahren, wer sich zu welchem Zeitpunkt an einem Punkt der gemeinsamen virtuellen Umgebung aufhält und ansprechbar ist.

Awareness-Informationen sind also das Instrument, den Benutzern ein Gefühl der gegenseitigen Wahrnehmung zu vermitteln und damit Koordination zu ermöglichen.¹⁰³

Eine von Tollmar, Sandor und Schömer angebrachte Definition charakterisiert diese Form von Awareness durch ihre Eigenschaft, das soziale Umfeld des Lernenden zu beleuchten, als „social awareness“.¹⁰⁴

Eine typische Form der Awareness-Informationen ist aus einer Vielzahl von Ansätzen, z.B. aus dem Bereich der CVEs oder Avatar-Umgebungen bekannt. Avatare bewegen sich in einer (oftmals dreidimensional) visualisierten virtuellen Welt und sind häufig selbst aufwändig grafisch gestaltet. Nutzer des CVE steuern über den Avatar ihre Aktionen in der virtuellen Welt. Durch die Integration von Mimik und Gestik sowie Audio- und Video-Kommunikationsmechanismen ergeben sich realweltliche Interaktions-, spezielle Kommunikationsformen zwischen den Nutzern.¹⁰⁵

Ähnlich weisen schon rein textuelle MUDs und MOOs verschiedene Wahrnehmungsmechanismen auf. Benutzer werden über das Erscheinen neuer Nutzer innerhalb eines Raumes informiert und es lässt sich zu jedem Zeitpunkt feststellen, welche Mitspieler sich in einem Raum aufhalten.

In der Literatur findet sich eine weitere Differenzierung des Begriffs der Awareness in die folgenden Kategorien: Activity Awareness (Wahrnehmung der Aktivitäten der Kooperationspartner), Availability Awareness (Informationen über die Ansprechbarkeit der Kooperationspartner), Process Awareness (Informationen über den Gesamtprozess in den sich eine Arbeit einfügt, z.B. in dem Bereich des Workflow Managements), Perspective Awareness (Hintergrundinformationen über die Kooperationspartner, ihre früheren Aktivitäten etc.) und Environmental Awareness (Informationen über Ereignisse, die außerhalb der eigentlichen virtuellen Arbeitsumgebung stattfinden), vgl. [Steinfeld et al. 1999, S. 83ff.].

Awareness im WWW

Das WWW besitzt in seiner Konzeption keinerlei Mechanismen der Awareness, der Nutzer bewegt sich allein, ohne Wissen anderer Nutzer durch das Netz. Konzeptionell bieten die Protokolle des WWW (HTTP) keine Ansatzpunkte der Übermittlung von Awareness-Informationen. Eine Reihe von Ansätzen beheben dieses Manko durch Entwicklung eines speziellen Servers, welcher z.B. Informationen über andere Nutzer auf einer speziellen Website bereitstellt.¹⁰⁶ Allen Ansätzen ist gemein, dass Nutzer den entsprechenden Awareness-Server als Ausgangspunkt ihrer Aktivitäten im WWW nutzen sollen und auf diese Weise Kontakt zu anderweitigen, an ähnlichen Themenbereichen interessierten Nutzern finden sollen. Aufgrund ihres Charakters eines spezifischen Diensteanbieters und keines allgemeinen Standards ist sämtlichen Systemen kein übermäßiger Erfolg beschieden.

Ähnliche Ansätze haben zum Ziel, Benutzern eine Art persönlichen Assistenten bei ihren Bewegungen durch das WWW an die Hand zu geben. Dieser weist gezielt auf an-

¹⁰³ Vgl. [Beaudouin-Lafon & Karsenty 1992, S. 171].

¹⁰⁴ Vgl. [Tollmar et al. 1996, S. 298ff.].

¹⁰⁵ Unter dem Anspruch des Betrachtens eines Lernprozesses primär aus medientechnischer Perspektive werden derartige Ansätze nicht weiter verfolgt.

¹⁰⁶ Klassische Systeme aus diesem Bereich sind CoBrow und Virtual Places (vgl. <http://www.cobrow.com/>, und <http://www.vplaces.net/>, Stand 1.12.2001) bzw. Lluna, Gooey und Third Voice. Bis auf CoBrow und Virtual Places sind alle genannten Dienste zurzeit eingestellt.

dere Mitglieder der Gemeinschaft hin, die sich auf besuchten Webseiten befinden. Hierbei wird neben dem Besuch von Webseiten auch die kooperative „Informationssammlung“ (vgl. [Prinz & Gräther 2000, S. 129]) unterstützt, d.h. durch eine geschickte Kopplung aus Gruppenarbeitsumgebung (BSCW), verschiedener zusätzlicher Werkzeuge und natürlich eines Webbrowsers werden kooperative Formen der Wissensorganisation bereitgestellt.¹⁰⁷

Das entwickelte Rahmenkonzept kooperativer Wissensräume erhebt nicht den Anspruch, einen allgemeinen Awareness-Mechanismus für das WWW anbieten zu wollen. Gegenseitige Wahrnehmung wird über die Anwesenheit der Nutzer in einem virtuellen Raum umgesetzt. Wahrnehmung des gemeinsamen Besuchs von Webseiten lässt sich über die Integration einer Awareness-Komponente in einem Shared Browser realisieren.

Aktivitätsawareness

Für einen kooperativen Gruppenprozess sind Informationen über die Aktivitäten der Mitlernenden von hohem Interesse, um mögliche Gemeinsamkeiten zu erkennen oder geeignete Ansprechpartner zu finden. Fragen, die ein kooperationsunterstützendes System in diesem Zusammenhang beantworten sollte, sind z.B. „Wer beschäftigt sich mit welchem Thema, wer bearbeitet welches Dokument?“ Die Forschung bezeichnet auch diese Form der Aktivitätsinformation als „activity awareness“.

Schon in den frühen Arbeiten der gegenseitigen Gruppenwahrnehmung wird diese Information über die Aktionen der Mitnutzer als wesentliches Kriterium und Kontext für die eigenen Interaktionen innerhalb einer kooperativen Umgebung erachtet.¹⁰⁸ Sohlenkamp und Chwelos differenzieren die Aktivitätsinformation noch eine Dimension weiter hinsichtlich des zeitlichen Bezugs der Arbeit. Es wird zwischen synchroner und asynchroner Awareness unterschieden [Sohlenkamp & Chwelos 1994].

Synchrone Awareness stellt z.B. Informationen bereit: Wo befinden sich einzelne Gruppenmitglieder? – Wer spricht mit wem? – Wer ist aktuell ansprechbar? – Wer arbeitet an welchen Dokumenten? – Wer kooperiert in synchronen Sitzungen?

Asynchrone Awareness bietet Informationen über den Status verschiedener Dokumente: – Wer hat welches Dokument modifiziert? – Was haben andere getan, das für den einzelnen Anwender interessant sein könnte?

Synchrone Awareness-Informationen

In der Forschung existieren eine ganze Reihe von Lösungen zum Problem der Wahrnehmung von Aktivitätsinformationen.¹⁰⁹ In [Fuchs 1999] werden durch die grafische Gestaltung von Piktogrammen, ihre Gestalt und Größe, Informationen visualisiert, wie z.B. Awareness für Öffnen und Schließen von Ordnern oder Editieren und Lesen von Dokumenten. Benutzer können spezifizieren, an welchen Ereignissen sie interessiert

¹⁰⁷ Prinz und Gräber kombinieren das WWW mit ihrem Konzept der Arbeitsbereiche. Arbeitsbereiche lassen sich aus dem „Cockpit“ direkt zu den besuchten Webseiten anzeigen. Gleichzeitig kann von den Nutzern eine Bewertung in Form eines numerischen Wertes zu der jeweiligen Netzseite abgegeben werden. Dieser Wert wird bei einer Suche über die integrierte Suchmaschine für eine Entscheidung über die Bedeutung der jeweiligen Seiten berücksichtigt. Es handelt sich um ein Konzept der gegenseitigen Bewertung von bereitgestellten Wissensquellen, also um eine spezifische Form von Awareness.

¹⁰⁸ „In these terms, awareness is an understanding of activities of others, which provides a context for your own activity“, [Dourish & Bellotti 1992, S. 107].

¹⁰⁹ Es findet sich beispielsweise die NESSI-Awareness Engine [Prinz 1999], eine dreidimensionale Visualisierung der Aktivitäten von Avataren in Bezug auf die Bearbeitung von Dokumenten oder der Aktivitäts-Awareness Client für das PoliTeam-Projekt [Fuchs 1999], [Sohlenkamp et al. 1998].

sind und in welcher Form bzw. in welchem Grade sie über diese informiert werden möchten.¹¹⁰

Neben der Frage der Auswahl von Informationen der gegenseitigen Wahrnehmung nehmen sich eine Reihe von Autoren insbesondere der Form der Präsentation an. Shiozawa, Okada und Matsushita führen beispielsweise in ihren Arbeiten zu speziellen Visualisierungstechniken von privaten und Gruppenräumen die Idee der „Background Awareness“, also einer unaufdringlichen „Hintergrund-Awareness“, ein [Shiozawa et al. 1999, S. 73]. Hierbei lassen sich Aktionen der Gruppe durch eine transparente Fenster-Visualisierungstechnik unaufdringlich wahrnehmen und beobachten, ohne explizit an der virtuellen Sitzung teilnehmen zu müssen.

Ähnlich schlagen Dourish und Bellotti einen „shared feedback approach“ [Dourish & Bellotti 1992, S. 109] vor, in dem Awareness-Informationen nicht separiert von den Materialien der eigentlichen Arbeit dargestellt werden, sondern in der gemeinsamen Arbeitsumgebung möglichst unaufdringlich integriert dargestellt werden.

Eine weitere Form der Aktivitäts-Awareness bilden kooperative Mauszeiger (Telepointer) oder Übersichtsdarstellungen (Radar Views). Diese sind aus verschiedenen Shared Whiteboard-Ansätzen bekannt (vgl. [Gutwin et al. 1996], [Roseman & Greenberg 1996a]) und visualisieren die Aktivitäten der Mit-Benutzer in Form einer engen Kopplung der Benutzerschnittstellen.

Erfahrungen im Umgang mit kooperativen Mauszeigern haben gezeigt (vgl. [Ho 2000]), dass durch eine zu große Anzahl dargestellter Telepointer die Übersichtlichkeit stark eingeschränkt wird. Aus diesem Grunde sieht die im Rahmen dieser Arbeit gewonnene Konzeption kooperative Zeigerinstrumente alleinig für Präsentationssituationen vor. Dieses kann z.B. die Besprechung eines grafischen Entwurfs oder im Rahmen eines Vortrages sein (vgl. Lernszenarien des synchronen Präsentierens auf einem Shared Whiteboard aus Kapitel 2).

Asynchrone Awareness-Informationen

Der verfolgte Ansatz konzentriert sich in seiner Ausrichtung als primär asynchron genutzte kooperative Lernumgebung auf die Bereitstellung von leistungsfähigen Mechanismen zu einer asynchronen Form von Awareness. Hierzu sind zu einem großen Teil keine speziellen, automatisch bereitgestellten Systemfunktionalitäten notwendig, Awareness wird vielmehr in großen Teilen durch die Lernenden selbst, z.B. durch Annotationen, verfügbar gemacht. Synchrone Formen von Awareness beschränken sich auf Informationen der Anwesenheit von Nutzern im Raum und in beschränktem Maße auf die Bereitstellung von Informationen der gegenseitigen Wahrnehmung im Bereich des Shared Whiteboard.

Dieses Vorgehen löst ein weiteres wesentliches Problem der automatisch von Seiten des Systems bereitgestellten Aktivitätsinformationen: Innerhalb eines kooperativen Lernprozesses ist es fast nicht zu entscheiden, welche Informationen über die Aktivitäten anderer für einen Lernenden von Wichtigkeit sind. Automatisch generierte Awareness-Informationen können hier schnell zu einer Vielzahl von unnötigen Tätigkeiten und einer Überfrachtung des Lernenden mit Informationen über die Aktivitäten anderer führen. Klassische Gruppenwahrnehmungsmechanismen nehmen beispielsweise Ereignisse aus der Umwelt des Benutzers auf, verarbeiten diese und geben sie an andere Gruppen-

¹¹⁰ Weitere ähnliche Ansätze sind die DIVA-Umgebung [Sohlenkamp & Chwelos 1994], in welcher Awareness-Informationen durch farbliche Markierungen an den Dokumenten deutlich gemacht werden, oder das Interlocus-System [Nomura et. al. 1998], welches ebenfalls verschiedene Formen von Aktivitäts-Awareness demonstriert.

mitglieder weiter. Dies geschieht unabhängig von dem Kontext, d.h. unabhängig des Sinnzusammenhanges, in dem sich der jeweilige Kooperationspartner befindet. Ziel ist nun, eine weitgehende Kopplung von Entstehungskontext eines Ereignisses und dem aktuellen Arbeitskontext des Benutzers, an den ein bestimmtes Ereignis weitergeleitet werden soll, zu erzielen. Ein Beispiel wäre z.B. das gemeinsame Editieren eines Dokumentes. Hier erscheint es wenig sinnvoll, dass ein Benutzer über die Änderung eines Dokumentes benachrichtigt wird, wenn er es gar nicht geöffnet hat, sich also in einem vollständig anderweitigen Kontext seiner Arbeit befindet.¹¹¹

Neben der beschriebenen Kopplung des Arbeits- und Entstehungskontextes ist es also wichtig, dass der Benutzer bestimmen kann, in welchem Arbeitskontext er über Ereignisse aus unterschiedlichen Entstehungskontexten informiert werden will. Awareness-Mechanismen dürfen der Koordination der Lernenden also nicht hinderlich sein, wie es z.B. Restriktionen der Gruppe wie z.B. Floor Control sein können (vgl. [Beaudouin-Lafon & Karsenty 1992, S. 171]).

Kooperative Wissensräume sollten daher sehr vorsichtig mit den Möglichkeiten verfahren, den Kontext des Nutzers gezielt automatisch zu erkennen und dahingehend Awareness-Informationen bereitzustellen. Zum einen ergibt sich ein gewisser Kontext zwangsläufig durch die Anwesenheit eines Nutzers innerhalb eines virtuellen Lernraums, zum anderen liegt es in der Natur von asynchronen Kooperationsformen, dem Kooperationspartner gezielt Informationen (z.B. über den eigenen Arbeitsfortschritt) bereitzustellen.

Awareness zur Strukturierung von Interaktion und Kommunikation

In der Literatur und Forschung finden sich eine unüberschaubare Anzahl von Versuchen und Herangehensweisen Awareness-Informationen innerhalb von kooperativen Umgebungen bereitzustellen. Die Mehrzahl dieser Ansätze fallen in den Bereich der Strukturierung von Interaktion und Kommunikation zwischen Nutzern (Avataren). Die vorliegende Arbeit kann in ihrer Ausrichtung der Skizze eines praktikablen und alltagstauglichen Werkzeugs nicht sämtliche derartigen Strömungen und Forschungsfragen berücksichtigen, vielmehr soll ein erster Eindruck der Möglichkeiten vermittelt werden, die für das entwickelte Rahmenkonzept eine technische und konzeptuelle Basis bereitstellen.

Interessant sind sicherlich zunächst einige durch räumliche („spatial“) Orientierung geprägte Vorstellungen von gegenseitiger Wahrnehmung in virtuellen Räumen. MUDs und MOOs sind dieser Klasse zuzurechnen.¹¹² Auf ähnlichem Gebiet haben Benford und Fahlén in ihren Arbeiten zur Kommunikation in Collaborative Virtual Environments Neuland beschritten (vgl. [Benford & Fahlén 1993])¹¹³.

¹¹¹ Gross und Prinz haben derartige Erfahrungen gemacht und betonen die Wichtigkeit des Kontextes von Aktivitäts-Awareness [Gross & Prinz 2000, S. 118ff.].

¹¹² Der virtuelle Raum wird rein textuell beschrieben, Spieler können eine Position innerhalb des Raums einnehmen. Die Reichweite des Chat bezieht sich zumeist auf die Ausdehnung des jeweiligen Raums. Es haben sich zwischen den Nutzern sehr eigene Mechanismen, meist in Form von kurzen Akronymen entwickelt, um Wahrnehmungsmechanismen in die Umgebung zu integrieren. (Ein schönes Beispiel hier ist die Abkürzung „brb“ - „be right back“, mit welcher eine kurzzeitige Abwesenheit signalisiert wird.)

¹¹³ Ein Avatar besitzt eine „aura“, einen „nimbus“ und einen „focus“. Der Nimbus bildet den Bereich ab, in dem das jeweilige Objekt (der Avatar) wahrgenommen wird. Der Fokus ist eine Blickrichtung, in der Objekte andere Objekte (Avatare) wahrnehmen. Als Aura wird der Interaktionsbereich eines Objektes in dem jeweiligen Medium bezeichnet. Das MASSIVE-System [Greenhalgh & Benford 1995]

Auswirkungen auf die Gestaltung kooperativer Wissensräume

Der vorgeschlagene Ansatz kooperativer Wissensräume legt die Betonung auf die Bereitstellung einer sorgfältigen Kombination aus asynchronen und synchronen Awareness-Informationen. Die Auswahl und der Umfang der verfügbaren Awareness-Informationen innerhalb einer kooperativen Lern- und Arbeitsumgebung ist aufgrund der mit ihnen verbundenen didaktischen und sozialen Fragestellungen den sekundären Medienfunktionen zuzuordnen. Auswahl und Umfang der notwendigen Wahrnehmungsmechanismen sind dem jeweiligen Einsatzkontext anzupassen.

Die im Rahmen dieser Arbeit entwickelte Rahmenarchitektur erlaubt durch ihr ereignisbasiertes Gesamtdesign (vgl. Kapitel 5) eine freie Spezifikation der zu protokollierenden Ereignisse und damit die prinzipielle Möglichkeit, eine Vielzahl an Awareness-Informationen zu erzeugen. In der Entwicklung erster Prototypen werden bewusst nur eine begrenzte Anzahl von Informationen über die Handlungen der Mitlernenden bereitgestellt, um die Umgebung einfach und beherrschbar zu gestalten.

Wichtiges Kriterium für die Alltagstauglichkeit einer Lernumgebung ist somit, eine möglichst freie Konfigurierbarkeit der jeweiligen Übermittlung von Awareness-Informationen zu ermöglichen.

Unter dem Gesichtspunkt der praktizierten, meist asynchronen Interaktionsformen zwischen Lernenden lassen sich Wahrnehmungsmechanismen über die Aktivitäten der Mitlernenden weitgehend durch die Kooperationspartner selbst bereitstellen. Annotationen an Dokumenten oder kurze, manuelle Mitteilungen strukturieren einen Lernprozess besser als eine Vielzahl an automatisch erzeugten Systemmitteilungen über die Aktivitäten von Mitlernenden, die vielleicht trotz Anwesenheit in demselben Raum nicht unmittelbar von Interesse sind.

Unabhängig von der dargestellten „Eigenverantwortung“ der Wahrnehmungsinformationen zwischen Lernenden existiert eine Anzahl von Informationen, die direkt oder auf Anfrage an die Lernenden zu übermitteln sind. Hierzu zählen insbesondere Rückmeldungen in Bezug auf gemeinsame Handlungen an Materialien.

Dies bezieht sich insbesondere auf die Abgrenzung von Zugriffsrechten und die Ausübung von Rollen. Awareness-Informationen sind in ihrer Form und Ausprägung stark an die Umsetzung des jeweiligen Zugangswerkzeugs gekoppelt. Aus diesem Grunde werden im Folgenden losgelöst von der konkreten Umsetzung die grundlegenden Konzepte der Einbettung von Funktionen der gegenseitigen Wahrnehmung in kooperative Wissensräume (am Beispiel von sTeam) aufgezeigt. Hierzu zählen beispielsweise:

- *Handlungen an Materialien:* Anzeige der Aktivitäten anderer Lernender innerhalb desselben Raumes in Bezug auf die Darstellung von Materialien, die sich im eigenen Sichtbarkeitsbereich befinden: Die Auswirkungen auf Materialien, die sich im Sichtbarkeitsbereich eines Nutzers befinden, wie z.B. der Inhalt eines Containers, müssen unmittelbar visualisiert werden. Verschiebt beispielsweise ein Nutzer ein Objekt innerhalb eines Raums oder löscht dieses, ist die Raumdarstellung der neuen Situation anzupassen (ebenso im Falle des Erzeugens neuer Objekte, Verschiebens, Kopierens, räumlichen Arrangierens von Objekten).

experimentiert hier mit verschiedenen Größen der Aura. Objekte können nur miteinander kommunizieren, wenn sich ihre Auren überlappen [ebd. S. 241 ff.].

- *Hervorhebung neuer Materialien*: Neu-erzeugte Materialien können durch eine farbliche oder textuelle Auszeichnung kenntlich gemacht werden. Beispiele sind das Einfügen von Materialien in den Rucksack eines Benutzers oder die Existenz neuer Meldungen in einem Postfach etc.
- *Modifikation von Materialien*: Ein sinnvolles Vorgehen ist es, Modifikationen an Materialien nicht direkt als Ereignis an die Raumteilnehmer weiterzugeben, sondern innerhalb von Attributen (z.B. „letzte Änderung durch“, „Liste der Modifikationen“, Versionsverwaltung etc.) zu visualisieren. Auch wenn synchrones Bearbeiten von Dokumenten in der vorgestellten Konzeption nicht unmittelbar berücksichtigt wird, kann eine Bereitstellung der Awareness-Information über die Bearbeitung eines Dokumentes durch einen Nutzer notwendig sein.¹¹⁴
- *Darstellung von Annotationen oder Mitteilungen an Materialien*: Werden Annotationen an Materialien angefügt, ist diese Information unmittelbar an dem Objekt selbst darzustellen (z.B. Anzahl der vorhandenen Annotationen, Hervorhebung einer neuen, ungelesenen Annotation).
- *Kommunikations-Awareness – Darstellung neuer Diskussionsbeiträge, des Diskussionsflusses*: Innerhalb eines Chat-Fensters muss klar erkennbar sein, welcher Nutzer einen Diskussionsbeitrag formuliert hat (z.B. durch den Namen des Nutzers). Sowohl in synchronen als auch asynchronen Kommunikationssituationen ist die Existenz eines neuen Beitrages hervorzuheben und die Kennung bzw. der Name des Autors mitzuteilen.
- *Visualisierung der Rollen von Nutzern*: Zugriffsrechte, die Benutzer an Objekten der Lernumgebung besitzen, können in Form von Rollen gekapselt sein. So besitzt beispielsweise eine Rolle „Moderator“ weiterführende Berechtigungen innerhalb eines kooperativen Prozesses gegenüber einem „Gast“. Derartige, auch temporär mögliche Zuordnungen zwischen Personen und Rollen sind geeignet darzustellen. Hierbei ist besonders der Wechsel einer Rolle anderen Nutzern mitzuteilen.
- *Zugriffskontrolle und Zugangskontrolle*: Rechte bzgl. des Zugriffes an Dokumenten oder der Zugang zu einem Raum bilden eine Form der darzustellenden Gruppenwahrnehmungsinformationen. So kann beispielsweise der gestattete Zugang zu einem virtuellen Raum durch eine geöffnete Tür symbolisiert werden, analog eine geschlossene Tür für einen nicht erlaubten Zugang.
- *Kooperative Zeigerinstrumente (Telepointer)*: Kooperative Zeigerinstrumente können speziell in Präsentationssituationen oder kooperativen Diskussionsphasen in Verbindung mit einem Shared Whiteboard oder einer Raumdarstellung wichtige Wahrnehmungsinformationen vermitteln. Ziel ist es, nur eine geringe Anzahl simultaner Telepointer zuzulassen und eine möglichst klare Zuordnung zwischen einer Person und dem zugehörigen Telepointer zu visualisieren. Dies kann beispielsweise durch eine farbliche Zuordnung geschehen.¹¹⁵
- *Thematische Beziehungen und Zuständigkeiten zwischen Nutzern und Dokumenten*: Im Falle der Visualisierung eines Raums auf einem Shared Whiteboard

¹¹⁴ Für zukünftige Erweiterungen des dargestellten Modells der gegenseitigen Wahrnehmung können zudem Informationen über die Historie von kooperativen Handlungen bereitgestellt werden (vgl. [Simone & Bandini 1997]).

¹¹⁵ Zeigerinstrumente sind bewusst nur für stark kooperative Präsentations- oder Vortragssituationen zuzulassen. Hierbei wird z.B. dem Vortragenden und den Zuhörern jeweils ein Telepointer zugeordnet. Eine Visualisierung sämtlicher Aktivitäten auf z.B. einem Shared Whiteboard oder innerhalb der Raumdarstellung ist aus Gründen der Übersichtlichkeit nicht sinnvoll.

kann die Darstellung der räumlichen Position der Nutzer in Bezug zu anderen Objekten, wie Dokumenten oder Grafiken erfolgen. Hierdurch lassen sich beispielsweise thematische Beziehungen und Zuständigkeiten zwischen Nutzern und Dokumenten ausdrücken.

Aus der Darstellung einer Reihe von sinnvollen bzw. notwendigen Formen der gegenseitigen Wahrnehmung und im kooperativen Umfeld werden eine Reihe von Designkonflikten deutlich:

Gegenseitige Wahrnehmung erfordert einen sinnvollen Kontext ihrer Bereitstellung. Zu viele Informationen verunsichern Lernende und führen zu einem Überangebot an Informationen, zu wenige Awareness-Informationen lassen den Lernenden für sich allein stehen, schaffen wenig Bezüge zu anderen Lernenden und kooperativ genutzten Materialien. Der einzig sinnvolle Weg scheint zu sein, zum einen eine bewusste Auswahl notwendiger Awareness-Informationen zu treffen und sich in dieser Auswahl sehr zurückhaltend zu verhalten (wie oben geschehen). Zum anderen ist die nutzerzentrierte Konfiguration und Bereitstellung von Informationen der gegenseitigen Wahrnehmung ein entscheidendes Kriterium. Hierbei ist davon auszugehen, dass Lernende für ihre persönlichen Formen der Zusammenarbeit sehr differenziert über notwendige Awareness-Mechanismen entscheiden. Das im Rahmen dieser Arbeit entwickelte Kernkonzept bietet die technischen Möglichkeiten, um verschiedene Formen der gegenseitigen Wahrnehmung umzusetzen und unter alltagspraktischen Bedingungen zu erproben. In ähnlicher Weise sind Zugriffskontrolle und Zugriffsschutz Beispiele einer diesbezüglichen Kernanforderung an die zugrunde liegende Architektur.

Awareness ist Teil der Medienfunktion des Synchronisierens. Ähnlich bilden im folgenden Abschnitt Benutzerrechte und Rollen die Grundbausteine der Medienfunktion des Zugreifens.

3.1.5 Zugriffskontrolle und Zugriffsschutz – Selbstadministration

Benutzergruppen und Rechte sind eine der Schlüsselkonzepte zu geordneten kooperativen Handlungen an Materialien. Es lassen sich Objekte einer Gruppe von Benutzern bzw. Lernenden zuordnen und klare Abgrenzungen gegenüber anderen Nutzern schaffen. Damit ist der Prozess der Vergabe von Benutzerrechten als maßgebliches Kriterium der Selbstadministration und Selbstorganisation der kooperativen Lernumgebung zu charakterisieren. Mittels leistungsfähiger Mechanismen der Vergabe und Weitergabe von Berechtigungen (Prozess des Übertragens von Berechtigungen) sind die Grundelemente einer durch die Lernenden weitgehend selbstorganisierbaren und selbststrukturierbaren und damit nutzerzentrierten Lernumgebung gegeben.

Schon Ellis et al. kommen Anfang der 90er Jahre zu der Erkenntnis, dass sämtliche Konzepte existierender CSCW-Systeme zu komplex erscheinen und keinerlei Aussage über die Abbildung der zugrunde liegenden Rechtemodelle in die Benutzerschnittstelle treffen (vgl. [Ellis et al. 1991]). Auf die heutige Situation bezogen muss diese Aussage relativiert werden, besitzt aber weiterhin ihre Gültigkeit: Existierende Rechtemodelle von kooperativen Systemen sind entweder trivial oder sehr komplex. Dieser Gedanke ist umso aussagekräftiger, wenn man eine Reihe von explizit für kooperative Applikationen vorgeschlagenen Modellen näher betrachtet.¹¹⁶

¹¹⁶ So führen Shen und Dewan ein äußerst fein granulierbares, aber äußerst komplexes Modell von Zugriffsrechten ein. Es existieren darin mehr als 50 Detailrechte für alle Facetten kooperativen Arbeitens (vgl. [Shen & Dewan 1992, S. 54ff.]).

Die vorliegende Arbeit konzentriert sich speziell auf Aspekte von Rechten und Gruppen, die die Nutzerzentriertheit kooperativer Wissensräume ermöglichen. Es steht wiederum nicht die Realisierung eines komplexen Systems, sondern die alltagstaugliche Lösung im Vordergrund.

Rollen sind neben Gruppenzugehörigkeiten eine Möglichkeit, Konfigurationen von Rechten gegenüber den Nutzern systematisch zu strukturieren und bilden damit eine Form der gegenseitigen Wahrnehmung in Bezug auf Handlungen an kooperativ genutzten Objekten. Im Rahmen dieser Arbeit erfolgt eine gezielte Einschränkung auf die technische Fundierung von Rollen. Eine Auswahl und Spezifikation geeigneter Rollen für einen kooperativen Lernprozess ist den sekundären Medienfunktionen zuzuordnen und an dieser Stelle von untergeordneter Relevanz.

Gleichzeitig ergeben sich innerhalb von Lernprozessen zwangsläufig Konfliktsituationen im Zugriff auf Materialien. Derartige Probleme der Strukturierung des Zugriffs (Concurrency Control) werden aus selbigen Gründen nicht behandelt.

Im Folgenden sind die grundlegenden Konzepte der Benutzergruppen und Rechtemodelle unter dem Blickwinkel kooperativer Lernszenarien dargestellt.

Es existieren eine Vielzahl von generischen Modellen und Konzepten für die Verwaltung von Benutzerrechten in nicht-kooperativen Anwendungen. Leider lassen sich diese Modelle nicht auf den Fall der Unterstützung von kooperativen Lehr- und Lern- oder Arbeitsumgebungen übertragen. Dies könnte ein Grund dafür sein, dass in den letzten Jahren nur wenige kooperative Lernumgebungen mit leistungsfähigen Modellen von Nutzerrechten ausgestattet worden sind (vgl. [Shen & Dewan 1992, S. 51]). Rechtssysteme von kooperativen Umgebungen können äußerst komplex sein. – Anforderungen der Übersichtlichkeit (Verständlichkeit) und eine geeignete Benutzerschnittstelle treten damit in den Vordergrund. Endbenutzern muss sowohl das Erkennen vorhandener oder verwehrter Berechtigungen als auch eine Modifikation von Berechtigungen leicht möglich sein. Gleichzeitig muss das zugrunde liegende Rechtemodell eines Systems (sein struktureller Aufbau) mit den Vorstellungen des Benutzers (Lernenden) konform entwickelt sein.¹¹⁷

Die folgenden Abschnitte dokumentieren das im Rahmen der vorliegenden Arbeit gewählte Modell aus Nutzergruppen und Zugriffsrechten. Es schließen sich eine Diskussion sowie Lösungsvorschläge zum Bereich der Negativ-Rechte (Ausschlussrechte) an. Kernkonzepte des gewählten Rechtemodells sind die Vererbung und Weitergabe (Delegation) von Rechten und das Übertragen der Verantwortung für Materialien.

Nutzergruppen

Die Zusammenfassung von Nutzern (Lernenden) in so genannte Nutzergruppen ist sicherlich eine nahe liegende Eigenschaft von Mehrbenutzersystemen aller Art. Nutzergruppen oder auch Rollen sind mit äußerster Sorgfalt zu wählen und beeinflussen die Grundstruktur kooperativer Prozesse. (Entsprechend charakterisiert die Flexibilität der Zuordnung von Nutzern zu Nutzergruppen den Grad der Selbstorganisation und Nutzerzentriertheit kooperativer Wissensräume.)

Für die folgenden Ausführungen sei betont, dass es sich bei der Struktur der Benutzergruppen nicht um eine wirkliche Teilmengenrelation handelt.¹¹⁸ Es werden von den

¹¹⁷ Vgl. [Ellis et al. 1991, S. 55].

¹¹⁸ Für eine formale Schreibweise vgl. [Sikkel 1997].

Eltern- auf die Kindergruppen reine Berechtigungen vererbt. – Obergruppen beinhalten dementsprechend nicht zwingend die Nutzer ihrer Untergruppen (oder umgekehrt).

Im Hinblick auf die Gestaltung eines offenen und flexiblen Gruppen- und Rechte-modells für die Umsetzung kooperativer Wissensräume lassen sich auch ohne eine vollständig formale Entwicklung der Beziehungen aus Nutzern, Nutzergruppen und Berechtigungen die folgenden Aussagen treffen:

Die Menge der Nutzer eines kooperationsunterstützenden Systems sei mit U bezeichnet. Die Menge U enthält sämtliche registrierte Benutzer, d.h. Teilnehmer, die über einen Benutzernamen und ein Kennwort verfügen.¹¹⁹

Es kann also geschrieben werden:

$$U = \{u_1, u_2, \dots, u_n\}, \text{ z.B. } PG = \{\text{Student 1, Student 2, Student 3, ...}\}$$

Eine Benutzergruppe ist typischerweise eine Menge von Nutzern und anderen Gruppen, kann also geschrieben werden als:

$$G = \{g_1, g_2, \dots, g_m, u_1, u_2, \dots, u_n\},$$

$$\text{z.B. } \textit{GruppeD} = \{\textit{GruppeA, GruppeB, ...}, \text{Nutzer1, ...}\}$$

Auf das Beispiel bezogen enthält also die *Gruppe D* verschiedene Untergruppen und eine Anzahl von Nutzern (z.B. alle Untergruppen von Personen, die den jeweiligen Forschungsgebieten zugeordnet sind, sowie den Leiter des Instituts).

Nutzer können laut Definition als Gruppen mit einem Element, also einem Nutzer aufgefasst werden. Gruppen, die mehr als einen Nutzer enthalten, seien als *echte* Gruppen bezeichnet.

Obergruppen sind Gruppen, die nur echte Gruppen als Elemente enthalten, also keine weiteren Nutzer. Auch hier ist zu beachten, dass g_i ein Element der Obergruppe ist und nicht alle Benutzer, die in der Untergruppe sind, auch in der Obergruppe enthalten sind:

$$G_{\text{Obergruppe}} = \{g_1, g_2, \dots, g_n \mid \text{alle } g_i \text{ sind echte Gruppen}\}$$

Analog kann eine Untergruppe als Gruppe definiert werden, welche in der Obergruppe in dem Sinne enthalten ist, dass sie gewisse Eigenschaften der Obergruppe erbt.

Es ergibt sich eine hierarchische Strukturierung von Benutzern und Gruppen, wobei die Nutzer typischerweise in den Blättern des resultierenden Baumes zu finden sind (vgl. Abbildung 3-2).

In der Literatur wird oftmals eine „Mitglieder-“ oder „members-Funktion“ auf eine Gruppe definiert, welche die Menge aller Nutzer der jeweiligen Gruppe, also auch die Mitglieder der jeweiligen Untergruppen, bildet. Eine derartige Funktion liefert die Blätter des zugehörigen Teilbaumes der Gruppe. Es lässt sich schreiben:

$$G_{\text{Mitglieder}} = \{u \in U \mid u \text{ ist eine Untergruppe von } G\}$$

Aus der hierarchischen Strukturierung und der Enthaltensein-Beziehung zwischen verschiedenen Benutzergruppen ergeben sich eine Reihe von notwendigen Festlegungen und konzeptionellen Fragestellungen:

- *Sollen Rechte und Attribute von einer übergeordneten auf untergeordnete Gruppen vererbt werden?* – Auf die in Kapitel 5 vorgestellte sTeam-Konzeption bezogen erben Untergruppen einer Gruppe kumulativ die mit der Gruppe verbundenen Rechte. Ein Benutzer einer Untergruppe ist damit faktisch auch Mitglied

¹¹⁹ Es sind keine anonymen Nutzer des Systems gemeint – für diese Nutzer existiert im Falle des sTeam-Systems eine spezielle anonyme Gruppe, die „world user group“.

in den entsprechenden Obergruppen. Attribute von Gruppen werden an die Untergruppen nicht vererbt.

- *Inwieweit können Benutzer mehreren Gruppen zugehörig sein? – Wenn ja, was geschieht mit aus den Gruppen zugesprochenen oder verweigerten Berechtigungen? Gibt es eine Hierarchie in der Gruppenzugehörigkeit, zählt z.B. die Zugehörigkeit zu einer speziellen Gruppe mehr als die Zugehörigkeit zu untergeordneten Gruppen?* – Die im Rahmen der vorliegenden Arbeit gewonnene Gruppenstruktur sieht die Möglichkeit eines Benutzers vor, Mitglied in beliebig vielen Gruppen sein zu können. Konflikte zwischen zugesprochenen oder verwehrtten Berechtigungen werden entsprechend den in den folgenden Abschnitten diskutierten Mechanismen aufgelöst.
- *Was geschieht im Falle der Entfernung von Gruppen mit untergeordneten Gruppen? – Werden diese an übergeordnete Gruppen angehängt oder gelöscht? – Innerhalb der gewählten Konzeption bleiben Untergruppen bei Löschen einer Elterngruppe unbeeinflusst. Sie werden zurzeit nicht an eine neue Obergruppe angehängt und stehen wie eine neue Wurzel „losgelöst“ von der Gruppenhierarchie.*
- *Müssen Gruppen in eine Hierarchie also Baumstruktur eingefügt werden oder können Gruppen ohne weitere Verknüpfungen oder Enthaltensein-Beziehungen nebeneinander stehen?* – Die sTeam-Konzeption sieht eine beliebige Anzahl von Wurzelknoten der Gruppenhierarchie vor, d.h. Gruppen können beliebig neben anderweitigen Gruppen stehen.

Das aus den Szenarien zu kooperativen Lernprozessen gewonnene Modell sieht ein weitgehend offenes Konzept von Gruppen und Rechten vor. Es ist dahingehend als recht universell zu bezeichnen, da von der Architektur her nur wenige Festlegungen getroffen werden. Vielmehr sind spezifische Clients und ihre Benutzerschnittstellen für eine Reduktion und Skalierung der Funktionalität (geeignete Abbildung von Teilen des Rechtemodells) verantwortlich.

Es erscheint sinnvoll eine hierarchische Strukturierung von Gruppen zuzulassen. Eine Baumstruktur der Gruppen und Nutzer bildet die meisten natürlichen Strukturen von Organisationen und Lernkontexten (Universität, Schule etc.) ab. So lassen sich beliebige studentische Lerngruppen gut in eine virtuelle Campusstruktur einfügen. Benutzergruppen sind zunächst unabhängig voneinander, können aber beliebige Untergruppen beinhalten. Benutzer können gleichzeitig Mitglied in einer Anzahl von Nutzergruppen sein (vgl. Abbildung 3-2). In dem typischen Beispiel sind alle Nutzer eines Systems automatisch einer Gruppe „Nutzer des Systems“ zugeordnet. Dies erleichtert das Setzen von voreingestellten Berechtigungen für alle Nutzer eines Systems.

Um einiges entscheidender als eine besonders umfangreiche Benutzer- und Gruppenstruktur erscheint insbesondere unter dem Aspekt der Selbstorganisation die Fragestellung einer möglichst flexiblen und dynamischen Administration der Gruppenstruktur. Typischerweise ist die Gruppenstruktur bei existierenden Systemen recht fest und starr. Für das Eintragen eines Nutzers in eine Benutzergruppe sind zumeist Administrator-Berechtigungen notwendig. Dementsprechend findet die Zuordnung eines Benutzers zu einer Gruppe oder einer Untergruppe zu einer Obergruppe meist für einen längeren Zeitraum statt. Die im Rahmen dieser Arbeit entwickelte Gruppenstruktur ist durch eine Reihe von Mechanismen flexibel an den jeweiligen Lernkontext anpassbar. Diese Anpassungen können von berechtigten Nutzern vorgenommen werden (nutzerzentrierte

Administration – Selbstorganisation). Dies geschieht über so genannte Gruppenkennwörter, die für jede Nutzergruppe das Hinzufügen und Entfernen von Nutzern und Gruppen erlauben. Gleichzeitig unterstützen automatisierte Anmeldeprozesse den Zugang zu einer Nutzergruppe (vgl. Abbildung 5-13).

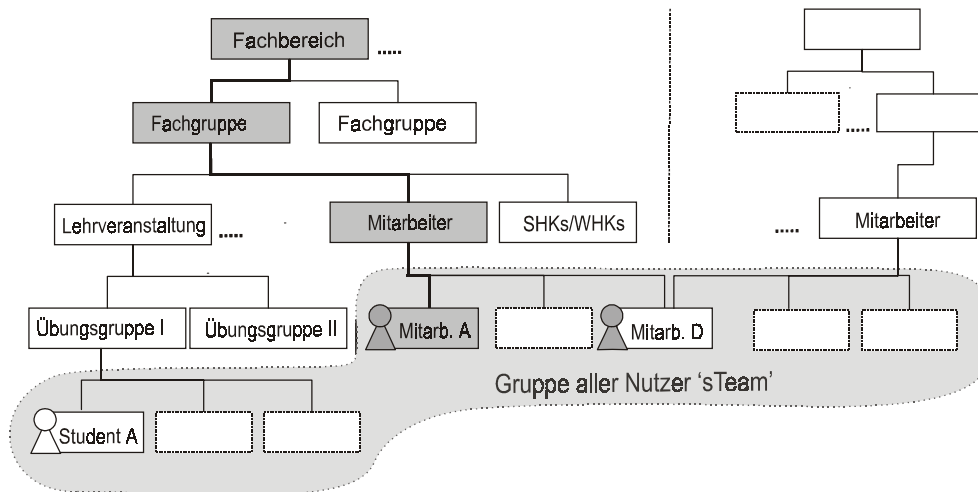


Abbildung 3-2: Typische Benutzer- und Gruppenstruktur

In der Literatur wird der Begriff Rolle und Gruppe meist synonym füreinander verwendet. Im Folgenden sei durch Verwendung der Bezeichnung Rolle eine primär zeitlich begrenzte Zuordnung einer Gruppe zu einer Person intendiert, also eine tätigkeits- oder funktionsbezogene Zuordnung getroffen. Beispielsweise erhält der Moderator in einer Diskussion erweiterte Befugnisse in der Möglichkeit, Dokumente innerhalb eines virtuellen Arbeitsraums zu arrangieren, z.B. das Recht zum Verschieben (move) von Dokumenten. Die Zugehörigkeit zu einer derartigen Moderatorengruppe ist jedoch zeitlich auf die Ausübung der Tätigkeit beschränkt. So kann die Funktion des Moderators flexibel über einen längeren Diskussionszeitraum rotieren. Eine wirkliche Gruppenzugehörigkeit zu der Gruppe der Raumverwalter würde dieses Recht für einen vorerst unbegrenzten Zeitraum zusprechen.

Von Seiten der Implementierung sind Rollen und Gruppen fast identisch umzusetzen. Lediglich von ihrer konzeptionellen und ideologischen Ausrichtung sind beide Konzepte gleichwohl zu differenzieren.

Bei den meisten verfügbaren Systemen ist die Möglichkeit der Veränderung der Benutzer- und Gruppenstruktur an einen oder eine Gruppe von spezifischen Benutzern oder die Administratoren geknüpft. In der sTeam-Konzeption wird der Ansatz verfolgt, das Recht, zu einer bestimmten Gruppe Benutzer hinzuzufügen, Benutzer zu löschen oder allgemein neue Gruppen einzurichten, an der entsprechenden Gruppe abgebildet wird bzw. an spezielle Rollen geknüpft ist. In diesem Sinne ist eine Benutzergruppe oder systeminterne Repräsentation eines Benutzers mit einem gewöhnlichen Objekt innerhalb des Servers gleichzusetzen. Es kann genauso mit Rechten wie jedes andere Objekt, z.B. ein Container-Objekt, versehen werden und dadurch eine dezentrale Administration erfolgen. Wichtige Schlüsselidee in diesem Zusammenhang ist die Weitergabe von Berechtigungen, also z.B. neue Benutzer einer Gruppe hinzuzufügen zu können, auch einem einfachen Benutzer zu gestatten, der hierdurch den Status eines Unter-Administrators erhält.

Zugriffsrechte

Es ist generell zu unterscheiden zwischen Authentifizierung oder Identifizierung (Authentication), also der Anmeldung an ein System und damit der Sicherstellung eines Status in Abhängigkeit eines bestimmten Benutzers („Wer bist du?“, „Wer hat das gesagt?“), und der Autorisierung, d.h. der Überprüfung von Berechtigungen zwischen Dokumenten, Objekten und einer Person („Was darfst du?“, „Wer darf auf dieses Dokument zugreifen?“). Weiterhin werden die Akronyme „Zugriffsrechte“ (Access Rights) und „Zugriffskontrolle“ (Access Control) synonym zueinander verwendet.¹²⁰

Betriebssysteme stellen klassischerweise Rechte zu einer Datei in Relation der Besitzverhältnisse und Zugehörigkeit zu einer Benutzergruppe dar. Im UNIX-Betriebssystem etwa lassen sich Rechte für eine Datei für den Nutzer bzw. die Nutzergruppe des Systems lediglich durch drei Attribute von Rechten, *Lesen*, *Schreiben*, *Ausführen*, spezifizieren. Diese Rechte werden entsprechend der Zugehörigkeit eines Benutzers zu bestimmten Gruppen ausgewertet.

Das klassische Modell der Zuordnung von Rechten von Nutzern und Gruppen (domains) und Objekten ist die Access Control List (ACL) nach [Lampson 1974]. Lampson benutzt die Notation von einer Menge von „objects“ O , eine Menge von „domains“ S und einer Zugriffsmatrix A .

Objects sind die Objekte des Systems, die einem Zugriffsschutz unterworfen werden, also beispielsweise die Materialien innerhalb kooperativer Wissensräume. Als Domain werden alle Einheiten („entities“), z.B. Benutzer oder Gruppen beschrieben, die durch die Zugriffsfunktion f über der Matrix A auf Objekte Berechtigungen zugewiesen bekommen. Shen und Dewan verwenden statt des Begriffes der Domain den Terminus des „subjects“ S , also eine Instanz, die Zugriff auf ein Objekt erhalten möchte [Shen & Dewan 1992, S. 52ff.]. Im Folgenden sei analog der Sicherheitsstatus des Modells durch ein Tripel (S, O, A) beschrieben.

Die Zugriffsmatrix A schafft einen Zusammenhang zwischen Objekten und Subjekten, also Nutzern und Gruppen. Jedes Element der Matrix spezifiziert über eine Menge von Berechtigungen R (oftmals implementiert durch einen Bitstring), welche Zugriffsrechte „access types“ r_1, \dots, r_n zwischen S und O vorhanden sind, z.B. *Lesen*, *Schreiben*, *Löschen*. Rechte spezifizieren den Zugriff eines Nutzers oder einer Gruppe auf das entsprechende Objekt. In der Literatur wird eine Kombination aus derartigen Berechtigungen als „view“ bezeichnet.

Die Anzahl der möglichen Rechte innerhalb eines View ist in den verschiedenen Systemen sehr unterschiedlich. Einige Systeme nutzen sehr viele Einzelberechtigungen, um jeden Aspekt kooperativer Zusammenarbeit durch ein eigenes Recht beeinflussen zu können (typischerweise Systeme, die Funktionen des Dokumentenmanagements bereitstellen). Andere Systeme benötigen lediglich einige wenige Berechtigungen, die einen eher universellen Charakter besitzen (Systeme, die primär synchrone Kooperationsunterstützung bereitstellen).¹²¹ Natürlich erfordern letztere durch eine begrenzte Anzahl von Kooperationsteilnehmern in einer Sitzung und ihre relativ enge Kopplung kein der-

¹²⁰ Der Bereich der Authentifizierung wird z.B. sehr ausführlich in [Lampson et al. 1992] dargestellt und ist primär für die technische Umsetzung in der Client-Server Kommunikation von Interesse.

¹²¹ Shirmohammadi et al. genügen beispielsweise die Rechte „no-access“, „view-only“, und „view/ interact“ für die Zugriffssteuerung auf Objekte auf z.B. einem Shared Whiteboard (vgl. [Shirmohammadi et al. 1998]).

art komplexes System von Rechten, wie es in typischen asynchronen Systemen notwendig ist. Das verfolgte Konzept ist eher der zweiten Gruppe von Systemen mit einer geringen Anzahl Berechtigungen zuzuordnen. – Nicht jede Möglichkeit (Handlung) wird innerhalb eines kooperativen Wissensraums durch ein eigenes Recht kontrolliert. Eine Vielzahl von Berechtigungen lassen sich beispielsweise zu den Grundrechten *Lesen*, *Schreiben*, *Löschen*, *Verschieben* und *Annotieren* verdichten.¹²²

Betrachtet man nun die Spalten der Matrix, so wird in Form einer Liste zu jedem Objekt spezifiziert, welche Subjekte Zugriff auf das jeweilige Objekt besitzen und damit welchem Nutzer und welcher Nutzergruppen (oder auch Rollen) Zugriff auf das entsprechende Objekt erlaubt ist. Jedem Objekt ist in dieser Form eine Access Control List (ACL) zugeordnet.

Daraus resultierend lässt sich leicht eine Funktion f definieren und implementieren, die für jeden Nutzer bzw. jede Nutzergruppe $u, g \in S$ und jedes Objekt $o \in O$ in Abhängigkeit der Menge der Zugriffsrechte $r_i \in R$ die Werte „erlaubt“ oder „verweigert“ ermittelt:

$$\begin{aligned}
 f &: S \times O \times R \rightarrow \{\text{erlaubt}, \text{verweigert}\} \\
 R &= \{\text{lesen}, \text{schreiben}, \text{löschen}, \dots\} \\
 S &= \{\text{Nutzer1}, \text{Nutzer2}, \text{Nutzer3}, \dots, \text{Gruppe1}, \text{Gruppe2}, \dots, \text{Rolle1}, \text{Rolle2}, \dots\} \\
 O &= \{\text{obj1}, \text{obj2}, \text{obj3}, \dots\}
 \end{aligned}$$

Es lässt sich damit für jeden Zugriffstypen $r_i \in R$ feststellen, inwieweit ein Nutzer $u \in S$ oder eine Gruppe $g \in S$ die Berechtigung $r_i \in R$ (z.B. *Lesen*) auf das Objekt $o \in O$ besitzt.

Die Zugriffsfunktion f überprüft im obigen Fall, ob sich in der ACL des Objektes o ein Element r_i für den Nutzer u befindet

$$f(u, o, r) = \begin{cases} \text{erlaubt}, & \text{wenn } u \in o.ACL \text{ für } r \\ \text{verweigert}, & \text{wenn } u \notin o.ACL \text{ für } r \end{cases}$$

oder in einer leicht anderen Schreibweise:

$$f(u, o, r) = \begin{cases} \text{erlaubt}, & \text{wenn } r \in A[o, u] \\ \text{verweigert}, & \text{wenn } r \notin A[o, u] \end{cases}$$

Die Implementierung von ACLs lässt sich durch den sTeam-Datenbankansatz relativ leicht realisieren (vgl. Abschnitt 5.2.6).

Für jedes Objekt innerhalb der vorgestellten Konzeption sind prinzipiell die Rechte: *Lesen*, *Schreiben*, *Löschen* (entspricht der Schreibberechtigung), *Verschieben*, *Ausführen*, *Einfügen (Insert)*¹²³ und *Annotieren* sowie die Weitergabe von Rechten vorgesehen.

Die Weitergabe von Rechten ist ein erweitertes Konzept des Administrationsrecht, wie es [Satyanarayanan 1989, S. 262] vorschlägt.¹²⁴ Dieses Recht erlaubt die explizite

¹²² Zu viele Berechtigungen erhöhen die Unübersichtlichkeit und hemmen offene Kooperationsstrukturen.

¹²³ Das Einfügen (Insert)-Recht bezieht sich auf das Recht, Elemente in die Umgebung eines Objekts einbringen zu dürfen, beispielsweise das Einfügen von Materialien innerhalb eines Raums. Dieses wird zunächst von der Schreibberechtigung auf das Raumobjekt getrennt, um die Modifikation des Raums an sich (z.B. Ändern von Attributen) kontrollieren zu können.

Modifikation einer ACL und ist so ein erster wichtiger Schritt hin zu einer dezentralen Administration.

Basierend auf der Berechtigung zur Weitergabe von Zugriffsrechten lassen sich kooperative Strukturen, insbesondere das Verfügbarmachen von Materialien für Mitlernende umsetzen. Hierbei wird ein Dokument entweder vollständig dem Besitz des Mitlernenden unterstellt (Weitergabe der Administration des Dokuments) oder lediglich die Berechtigung zum Setzen von Zugriffsrechten vergeben (ein Dokument verbleibt im Besitz des Nutzers und Mitlernende erhalten lediglich das Recht, beliebig Berechtigungen zu vergeben).

Im Folgenden werden insbesondere die Weitergabe (Delegation) und explizite Negativrechte diskutiert.

Negative Zugriffsrechte

Eine sinnvolle Strategie der Vergabe von Zugriffsrechten ist das Binden von Rechten primär an Benutzergruppen. Bei einem derartigen Vorgehen werden Berechtigungen an zugehörige Benutzergruppen geknüpft und nur in seltenen Fällen einzelne Benutzer mit Rechten versehen.¹²⁵ In ACLs zu einem Objekt befinden sich demgemäß in den meisten Praxisfällen nur Benutzergruppen. Zielsetzung ist die Wahrung von Übersichtlichkeit durch Vergabe von möglichst wenigen Rechtespezifikationen. Ein derartiges Vorgehen impliziert, dass es in vielen Fällen schwierig, wenn nicht unmöglich, erscheint, einen gewissen Sachverhalt durch eine Konfiguration von rein positiven Rechten auszudrücken.

Ist beispielsweise einem einzelnen Benutzer der Zugriff auf ein Objekt zu verwehren, erscheint es bei obigem Ansatz unumgänglich, ihn aus der entsprechenden Gruppe, der das jeweilige Recht zugesprochen ist, zu entfernen und alle sonstigen mit der Gruppe verknüpften Berechtigungen explizit neu zu setzen. Es ist sicherlich ein hoher Aufwand, um einen einfachen Sachverhalt wie „Person X darf auf Objekt Y nicht zugreifen“ auszudrücken.

Das explizite Setzen eines derartigen Negativrechtes kann auf unterschiedliche Weise realisiert werden. Es ergeben sich im Wesentlichen die folgenden Möglichkeiten der Einführung von Negativrechten:

a) Pauschales Ausschlussrecht

Für jedes Objekt ist als Attribut oder Recht eine Gruppe mit Benutzern zu führen, die keinen Zugriff auf das jeweilige Objekt erhalten. Die Art des Zugriffs kann nicht genauer spezifiziert werden. Das Setzen eines negativen Ausschlussrechts überlagert alle positiv gesetzten Rechte.

b) Negative Gruppenzugehörigkeit

Im Konzept der negativen Gruppenzugehörigkeit können Personen explizit von der Zugehörigkeit einer Gruppe ausgeschlossen werden. Insbesondere unter dem Gesichts-

¹²⁴ In der Literatur wurde die Weitergabe (Delegation) von Berechtigungen erstmals im Zusammenhang mit den Rechtemodellen zu relationalen Datenbanken diskutiert (vgl. [Griffiths & Wade 1976]). Autoren wie Bertino et al. greifen diese Ideen auf und ergänzen derartige Konzepte um Negativberechtigungen und erweiterte Mechanismen zur Delegation und Widerruf der Weitergabe von Rechten (vgl. [Bertino et al. 1997]).

¹²⁵ Die Benutzerschnittstelle des aktuellen sTeam-Systems schränkt die Vergabe von Berechtigungen bewusst auf Benutzergruppen ein. Rechte können aus Gründen der Übersichtlichkeit nicht für einzelne Nutzer vergeben werden.

punkt der möglichen Zugehörigkeit von Personen zu mehreren Nutzergruppen kann auf diese Weise erreicht werden, dass eine Person niemals indirekt Mitglied einer speziellen Gruppe werden kann. Die Person ist dann nicht nur „Nicht-Mitglied“ der Gruppe, sie ist explizit von der Mitgliedschaft ausgeschlossen (vgl. [Sikkel 1997]).

c) Explizite Negativrechte

Zu jeder positiven Berechtigung wie *Lesen*, *Schreiben* und *Löschen* existiert ein zugeordnetes Negativrecht. So fügt Satyanarayanan in seinem System zu jedem Dateiojekt neben der typischen ACL mit Positivrechten eine zweite ACL mit Negativrechten an (vgl. [Satyanarayanan 1989]). Der Ansatz der expliziten Negativrechte muss nicht zwingend zwei getrennte ACLs haben. Es können auch Negativ- und Positivrechte gemischt vorliegen. Es lassen sich in jedem Fall für jedes Recht separiert in der ACL zu jedem Objekt diejenigen Benutzer oder Gruppen spezifizieren, denen das entsprechende Recht zu entziehen ist. Eine sinnvolle Festlegung ist, dass negative Rechte ein im Rahmen einer Gruppenzugehörigkeit gesetztes positives Recht überschreiben.

Problematisch erscheint lediglich die Evaluation der jeweils gültigen Berechtigung für den Nutzer. Hier bietet sich in der Benutzerschnittstelle eine Visualisierung an, die die jeweils gültigen Berechtigungen an einem Objekt für einen spezifischen Nutzer oder eine Gruppe darzustellen vermag.¹²⁶

d) Negativrechte mit hierarchischer Vererbung

Werden Negativrechte (Ausschlussrecht) mit dem Konzept der Vererbung von Rechten durch Gruppen gekoppelt, sind einige Prioritätsregeln zu definieren. Es erscheint aufgrund der Übersichtlichkeit (Intention der Nutzer) ein sinnvolles Dogma zu sein, Festlegungen an den Blättern des Hierarchiebaumes mit höchster Priorität zu belegen. Das bedeutet, ein explizit für einen Benutzer gesetztes Positivrecht überschreibt ein für den Benutzer durch eine Gruppenzugehörigkeit vererbtes Negativrecht. Gleichwohl ist ein expliziter, für einen Nutzer definierter Ausschluss höher als ein von einer Gruppe vererbtes Recht zu setzen.

Innerhalb des Hierarchiebaumes werden Positiv- und Negativrechte nicht abgestuft. Es ist eine Festlegung zu treffen, wie eine (oder mehrere) gleichzeitig vorhandene Positiv- und Negativberechtigungen innerhalb der Gruppenhierarchie zu werten sind. Hier führt sowohl die Vorgehensweise der höheren Bewertung von Ausschlussrechten gegenüber Positivberechtigungen zu Schwierigkeiten als auch die umgekehrte nahe liegende Strategie der höheren Bewertung von vorhandenen Positivberechtigungen gegenüber Negativberechtigungen. In jedem Fall ist der Sinngehalt einer getroffenen Festlegung von der Semantik des aufgebauten Hierarchiebaumes abhängig.¹²⁷

Lediglich unstrittig erscheint die Festlegung, dass ein explizit an eine Person geknüpftes Recht, sei es ein Positiv- oder Ausschlussrecht, einem durch eine Gruppenzugehörigkeit vererbten Recht vorgelagert ist. Hier erscheint die Semantik eindeutig, dass durch das explizit an die Person geknüpfte Recht eine durch die administrative Person gewollte gewisse Priorität intendiert erscheint.

¹²⁶ Das BSCW-System bietet für diesen Fall einen expliziten Überprüfungsmechanismus der vorliegenden Berechtigungen, einen „Evaluate“-Knopf (vgl. [REP-FIT-1999-64]).

¹²⁷ Es ist schwer allgemein zu entscheiden, inwieweit jemand, der z.B. Mitglied in drei durch Positivberechtigung ausgestatteten Gruppen ist und gleichzeitig aber auch einer durch ein Negativrecht belegten Gruppe angehört, seine Berechtigung für ein Dokument lediglich durch eine Gruppenzugehörigkeit verlieren soll. Umgekehrt können Negativberechtigungen, welche durch eine Gruppenzugehörigkeit außer Kraft gesetzt werden, zu erheblicher Verwirrung führen.

Damit ergibt sich die folgende Regel für die Auswertung von Positiv- und Negativrechten als Modell der höchsten Bewertung von Positivberechtigungen: *Es ist eine Wertung des direkt an einen Benutzer vergebenen Rechts in der Priorität vor ein durch Vererbung aus einer Gruppenzugehörigkeit resultierendes Recht zu stellen.*

Hierzu ist von der Gruppe der Subjekte (Nutzer und Gruppen) explizit die Gruppe U der Nutzer abzuspalten:

$$S = \{\text{Nutzer1, Nutzer2, Nutzer3, ..., Gruppe1, Gruppe2, ..., Rolle1, Rolle2}\}$$

$$S_{\text{echteGruppen}} = \{\text{Gruppe1, Gruppe2, ..., Rolle1, Rolle2}\}$$

$$U = \{u_1, u_2, \dots, u_n\}$$

Aus der Konstruktion von Negativrechten ist die Funktion zur Überprüfung von Rechten in folgender Weise zu modifizieren:

- I. Soll überprüft werden, ob ein Nutzer u Zugriff auf ein Objekt o mit der Berechtigung r erhält, so ist zuerst zu überprüfen, ob an dem Objekt o ein explizites Negativrecht $-r$ oder Positivrecht r für den Nutzer u vorliegt. Wenn dies der Fall ist, wird entsprechend der Zugriff *erlaubt* oder *abgelehnt*.
- II. Ist dies nicht der Fall, wird über die Gruppenzugehörigkeit des Nutzers geprüft, inwieweit ein Positivrecht für das zu überprüfende Recht r am Objekt o existiert. Existiert dieses Positivrecht, hat es höchste Priorität. In dem beschriebenen Modell findet eine Wertung von Individual- vor Gruppenberechtigung statt.¹²⁸

e) Negative Blattrechte

Um das obige Modell negativer Rechte um einiges zu vereinfachen und insbesondere semantische Konflikte in der Auslegung von Prioritätsregeln in Bezug auf die Wertung von Positiv- gegenüber Negativberechtigungen zu umgehen, wird für die sTeam-Konzeption ein Modell vorgeschlagen, in dem lediglich Negativberechtigungen für einzelne Benutzer und nicht für ganze Gruppen von Benutzern zulässig sind. In einem derartigen Vorgehen wird zwar die Freiheit und Mächtigkeit des Modells der Negativrechte um einiges eingeschränkt, umso mehr wächst jedoch die Übersichtlichkeit und Eindeutigkeit der Semantik von vergebenen Berechtigungen in Konfliktsituationen.

Vererbung von Rechten

Berechtigungen und Rechte sind in unserer alltäglichen Umwelt ein dynamisches und gleichzeitig aber auch ein durch soziale Gesetzmäßigkeiten und Normen geprägtes Feld. So erscheint es uns ganz natürlich, dass Materialien bei ihrem Weg von einer Person zu einer anderen Person ihren Status in vielerlei Hinsicht an den jeweiligen Verwendungskontext, die Handelnden oder auch den Ort anpassen können. So verändert z.B. ein Klausurblatt seinen Status von einem hoch sensiblen, nur für den Dozenten zu lesenden Dokument nach der Klausur zu einem öffentlichen, frei kommentier- und diskutierbaren Dokument. Insbesondere in kooperativen Lehr- und Lernprozessen erscheint es unumgänglich, sich von einer starren Zuordnung von Berechtigungen an Dokumenten zu lösen.

¹²⁸ Diese Festlegung ist in Abhängigkeit der Semantik der gesamten Gruppenstruktur zu wählen, so trifft z.B. [Satyanarayanan 1989, S. 261] eine genau konträre Festlegung in Bezug auf die Bewertung von Positiv- und Negativberechtigungen, indem negative Rechte in jedem Fall vor Positivrechten gewertet werden.

Es lassen sich vier Dimensionen der dynamischen Vergabe von Berechtigungen ausmachen: Vergabe von Berechtigungen aus der sozialen Gruppenstruktur, Vergabe von Berechtigungen aus der räumlichen oder kontextuellen Anordnung von Objekten, explizite Weitergabe (Delegation) von Berechtigungen und die Übertragung der Verantwortlichkeit von Berechtigungen.

Vererbung von Berechtigungen aus der Gruppenstruktur

Der sicherlich einfachste Fall des Erlangens von Berechtigungen an Dokumenten und Materialien ist das schon beschriebene Erlangen von Berechtigungen aus der Benutzer- und Gruppenstruktur (Abbildung 3-2). Werden Berechtigungen pauschal für eine Benutzergruppe erteilt, erlangen alle Mitglieder dieser Benutzergruppe und auch sämtliche Mitglieder von Untergruppen entsprechende Berechtigungen. Über einen derartigen Mechanismus lassen sich Berechtigungen resultierend aus dem sozialen Status, also der dauerhaften Zugehörigkeit eines Nutzers zu einer Benutzergruppe, abbilden, aber auch dynamische Prozesse wie die Rollenzugehörigkeit ausdrücken. Wird beispielsweise ein Teilnehmer einer Diskussionsrunde im Verlauf des Treffens zu einem Moderator ernannt und in eine zugehörige Gruppe aufgenommen, erhält er automatisch die mit der Moderatorenrolle verknüpften Berechtigungen an Dokumenten und Materialien bzw. an der Umgebung des Diskussionsprozesses.

Insofern ist das Rechtemodell kooperativer Wissensräume flexibel in Bezug auf die Zuordnung von Personen zu Gruppen bzw. Rollen auszuführen. Das sTeam-System sieht eine Übertragung des flexiblen Rechtekonzepts auf die Verwaltung und Administration der Gruppenstruktur vor. Hierdurch lassen sich Berechtigungen an Gruppen dezentral administrieren und flexibel anpassen. Rechte werden durch die Gruppenhierarchie von Eltern- auf Kindergruppen weitergegeben.

Neben der Vererbung aus der Gruppenstruktur sind im Rahmen eines Rechtemodells kooperativer Wissensräume als weiteres wesentliches Verfahren für die Vergabe von Berechtigungen zu nennen. Hierzu kann für ein Objekt spezifiziert werden, ob seine Berechtigungen aus dem Umfeld (Environment) des Objekts (kontextuelle Ableitung von Rechten) oder von einem explizit angegebenen Objekt abgeleitet werden sollen (semantische Ableitung von Berechtigungen).

Vererbung von Berechtigungen aus der Umgebung (Environment) – kontextuelle Ableitung von Rechten

Das Verfahren der Vererbung von Berechtigungen aus der Umgebung erlaubt es, Rechte an Ordnern (Containern) zu spezifizieren, welche ihre Berechtigungen an die in ihnen enthaltenen Materialien weitergeben. Hierdurch lassen sich Kooperationsprozesse einfach und flexibel gestalten, indem Materialien durch ihre Strukturierungselemente, also Umgebungen mit den notwendigen Berechtigungen für einen Gruppenarbeits- oder Lernprozess versehen werden. In einem derartigen Ansatz bewirkt die Verschiebung oder Weitergabe eines Dokumentes gleichzeitig die Anpassung der Berechtigungen des Dokumentes an die entsprechenden neuen Gegebenheiten.¹²⁹

Bei dem Kontext, aus welchem Berechtigungen geerbt werden, kann es sich bedingt durch die architektonisch-zugrundegelegte Objektstruktur (vgl. Kapitel 5) sowohl um

¹²⁹ Im Gegensatz zur kontextuellen Vererbung werden bei der semantischen Vererbung Rechte von einem festen Referenzobjekt abgeleitet. Dabei werden Rechte unabhängig von der Position, von dem Kontext, in dem sich ein Objekt befindet, abgeleitet. Demgemäß werden die Rechte eines Objekts beispielsweise beim Verschieben des Objekts nicht dem jeweiligen Kontext angepasst.

einen Ordner (Container) als auch um einen virtuellen Raum, aber auch um ein beliebiges anderweitiges Objekt (z.B. Rucksack eines Nutzers) handeln.

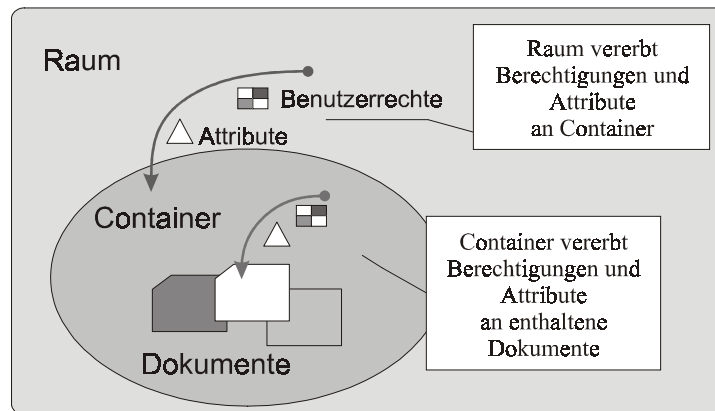


Abbildung 3-3: Kontextuelle Vererbung

Des Weiteren ist die Vererbung von Berechtigungen nicht auf eine Hierarchiestufe beschränkt; so kann ausgehend von der Vererbung von Berechtigungen aus dem unmittelbaren lokalen Umfeld auch eine Vererbung von beliebig höheren Hierarchiestufen durch mehrfache Vererbung von Rechten vorgenommen werden. Ein derartiges Konzept macht speziell bei der kontextuellen Vererbung aus der jeweiligen Umgebung (Environment) Sinn: Das Objekt erbt vom Container, der Container erbt vom Raum (vgl. Abbildung 3-3).

Gleichzeitig lassen sich über den kontextuellen Vererbungsmechanismus sehr elegant Kooperationsmodelle in die Struktur von Benutzerrechten abbilden.¹³⁰ So kann ein Dokument beispielsweise, wenn es in den Rucksack oder die Arbeitsmappe eines anderen Benutzers verschoben wird, automatisch die Berechtigungen des entsprechenden Objektes erben, es ist keine explizite aufwändige Anpassung der Berechtigungen notwendig (vgl. Abbildung 3-4).

Ableitung von Berechtigungen von einem festen Referenzobjekt

Typische Dokumentenmanagementsysteme¹³¹ setzen bei dem Erzeugen neuer Dokumente oder dem Einfügen von Dokumenten die Berechtigungen auf eine fest vorgegebene Voreinstellung (Default-Einstellung). Erfahrungen in der täglichen Praxis mit verschiedenen Systemen haben gezeigt, dass durch ein derartiges pauschalisiertes Setzen von Rechten kontinuierlich globale Anpassungen der Berechtigungen notwendig werden. Zu diesem Zweck bieten zwar einige Systeme die Möglichkeit, Berechtigungen für eine Anzahl von Objekten in einem Schritt zu setzen, diese muss jedoch explizit nach jeder Modifikation an der Gesamtstruktur angewandt werden.

Diese Notwendigkeit wirkt sich beispielsweise sehr negativ im Umgang mit Materialien (z.B. eine Präsentation/Vorlesung) aus, die eine große Anzahl von Dateien enthalten. Es ist bei jedem Vorgang des Verschiebens oder der Weitergabe notwendig, für alle

¹³⁰ Hierbei erscheint es wenig sinnvoll das „variable“-Konzept von Sikkel zu adaptieren, welches erlaubt, die Weitergabe für jedes Teilrecht separat zu spezifizieren (vgl. [Sikkel 1997]). Eine Vererbung von einzelnen Teilberechtigungen aus der Umgebung von Objekten wirkt sich sicherlich negativ auf die Übersichtlichkeit und Benutzbarkeit des Systems aus.

¹³¹ Ein Beispiel ist das Hyperwave System.

Dateien die Berechtigungen explizit anzupassen. – Ein Delegationsmechanismus von Rechten eröffnet im Gegensatz hierzu die Möglichkeit, alle Berechtigungen von einem zentralen Objekt abzuleiten.

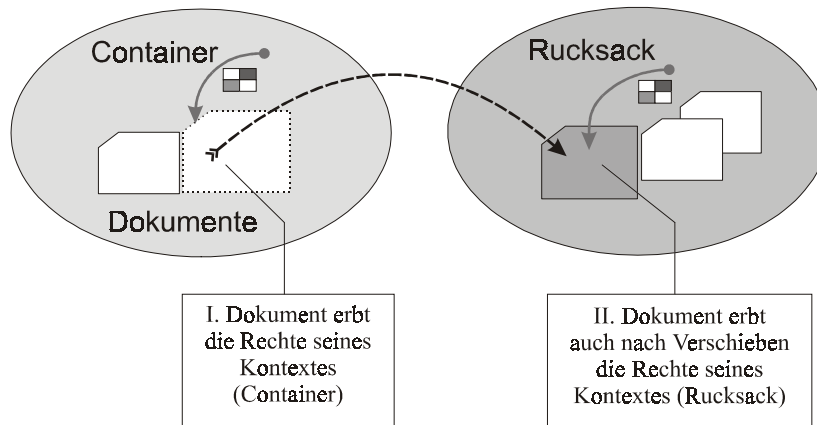


Abbildung 3-4: Kontextuelle Vererbung – Verschieben eines Dokuments in den Rucksack eines Benutzers

Die Möglichkeit der Ableitung von einem anderen spezifizierten Objekt bindet die Berechtigungen unabhängig vom Ort des Dokumentes an die Rechte eines festen Referenzobjekts (Vererbungsobjekts). Hierbei kann es sich um einen Raum, einen Container, ein beliebiges anderes Objekt oder einen Benutzer handeln. Eine Ableitung von Berechtigungen ist insbesondere unter dem Aspekt einer möglichst einfachen und eines für Teilnehmer eines Kooperationsprozesses reibungslosen Umganges mit Materialien innerhalb von Räumen wichtig. Idealerweise befinden sich in der Praxis nur sehr wenige direkt an den Dokumenten spezifizierte Berechtigungen. Ein derartiges Vorgehen macht es den Lernenden umso leichter, die Struktur von Lehrmaterialien flexibel an ihre konkreten Bedürfnisse anzupassen.

Das Erzeugen neuer Objekte lässt sich zusätzlich erheblich vereinfachen, indem durch eine geschickte Voreinstellung der Rechte an neuen Dokumenten, z.B. durch Spezifikation eines Containers oder Raums als Rechte-Vererbungsobjekt, keinerlei Berechtigungen explizit neu zu spezifizieren oder anzupassen sind.

Verantwortung für ein Objekt

Der klassische Ansatz einer Zuordnung zwischen Objekten und Benutzern besteht aus einem einfachen Besitzverhältnis. Viele Systeme legen den Erzeuger einer Datei als „creator“ oder „owner“, also den Verantwortlichen der entsprechenden Datei fest. Als direkte Konsequenz übt ein „owner“ das Recht aus, beliebige Zugriffsrechte an dem entsprechenden Objekt zu modifizieren. Hierzu kann etwa das UNIX-Betriebssystem als Beispiel genannt werden, welches den Besitzer einer Datei explizit kennzeichnet und zudem Mechanismen bereitstellt, den Besitz einer Datei an einen anderen Benutzer zu übertragen. Zu beachten ist, dass der Besitzer nicht automatisch über alle Rechte an seinen Objekten verfügen muss – er kann sie sich jedoch in jedem Fall selbst gewähren.

Die Administration von Objekten ist in drei wesentliche Klassen von Grundkonzepten zu unterteilen, die Möglichkeit der Ausübung von Administrationsaufgaben an einem Objekt (z.B. das Einstellen von Berechtigungen), die Übertragung von Administrationsberechtigungen an andere Personen (Delegation von Berechtigungen) sowie die Übertragung der Verantwortlichkeit an weitere Personen.

Weitergabe (Delegation) von Berechtigungen

Der einfachste Fall der Ausübung von Administrationsaufgaben an Objekten kann in den meisten Fällen von dem Besitzer des Objektes vorgenommen werden. Verallgemeinern lässt sich dieses Vorgehen, indem nicht ein einzelner Besitzer, sondern eine ganze Gruppe von vertrauenswürdigen Personen/Administratoren (Trusted Group) an einem Objekt spezifizierbar ist.¹³²

Das zugrunde liegende Konzept wird im Folgenden als Delegation von Rechten bezeichnet. In aktuellen Forschungsarbeiten finden sich verschiedene Ansätze, die sich auf die Grundarchitektur der Delegation von Berechtigungen zurückführen lassen. So schlagen Stiernerling und Wulf ein Konzept vor, in dem Benutzerberechtigungen um die Eigenschaften von „vertrauenswürdigen Personen“ erweitert werden (vgl. [Stiernerling & Wulf 1998]). Hierbei handelt es sich um eine Art Protokoll zur Erweiterung einer Gruppe von vertrauenswürdigen Personen. Die grundsätzliche Idee wurde zwar in dem Bereich der Groupware entwickelt, lässt sich aber durchaus auf Szenarien der Lehr- und Lernunterstützung transponieren. Versucht eine Person ein Dokument mit nicht vorhandener Leseberechtigung zu öffnen (bzw. ein nicht für Schreibzugriff vorgesehenes Dokument zu schreiben), werden über einen Benachrichtigungsmechanismus eine Anzahl von zuvor spezifizierten Personen mit der Frage kontaktiert, inwieweit dieses Recht noch zugesprochen werden kann. Hier sind verschiedene Strategien denkbar – z.B. eine Art Widerspruchsfrist. Ein derartiges Konzept erlaubt eine größere Freiheit in der späteren Arbeit mit Materialien, da Rechte durch die Nutzung der Dokumente dynamisch angepasst und erweitert werden.

Aus dem Workflow Management bekannte Vorgehensweisen der Weitergabe von Dokumenten von einem Bearbeiter zum nächsten sind analog auf das Konzept der Weitergabe von Berechtigungen abbildbar. Es existieren Vorschläge, neben der expliziten Weitergabe eines Dokumentes an eine Person auch die Möglichkeit einzuräumen, einen derartigen Delegationsschritt wieder rückgängig machen zu können (Revoke). Das vorgestellte Rechtemodell würde diese Vorgehensweise durch Entfernen der entsprechenden Person aus der Gruppe der für das Objekt Administrationsberechtigten realisieren.

Die Gruppe der vertrauenswürdigen Personen ist nicht in jedem Fall homogen. So kann beispielsweise weiterhin der Besitzer des Dokumentes innerhalb der Gruppe der vertrauenswürdigen Personen ausgemacht werden. Er besitzt weiterhin das Recht, zusätzlichen Nutzern Rechte an dem Dokument zu übertragen.

Konzeptionell fügt sich die Idee der Möglichkeit zur Definition von vertrauenswürdigen Gruppen nahtlos in die Konzepte zum kooperativen Arbeiten und Arrangieren von Dokumenten ein. Auf den Gedanken der Selbstorganisation von Lerngruppen bezogen lassen sich über einen derartigen Mechanismus ganze Gruppen von zur Administration an Materialien befähigten Personen bilden.

Recht zur Weitergabe von Zugriffsrechten

Die Weitergabe von Berechtigungen löst nicht vollständig das Problem der dezentralen Administration bzw. selbstorganisierten Verwaltung von Materialien. Erst das Übertragen des Rechts, neue Berechtigungen setzen zu können (ein Recht was typischerweise

¹³² Die hierzu notwendige Entkoppelung von dem Besitzen eines Dokumentes und der Möglichkeit zu dessen Administration wurde schon durch Satyanarayanan mittels eines eigenen „administer“-Rechts an Verzeichnissen eingeführt (vgl. [Satyanarayanan 1989, S. 262]). Besitzer dieser Berechtigung sind in der Lage, die ACL zu einem Verzeichnis zu modifizieren und damit Berechtigungen neu zu setzen. Eine Erweiterung dieser Berechtigung auf sämtliche Typen von Objekten ist in der vorliegenden Konzeption vorgesehen.

nur der Besitzer/Erzeuger eines Dokumentes ausübt), erlaubt das Schaffen freier Strukturen in Handlungen an Materialien. So kann es innerhalb kooperativer Prozesse notwendig sein, Materialien vollständig an einen anderen Nutzer zu übergeben bzw. die Administration an diesem Objekt zu übertragen.

Einen interessanten Aspekt und sicherlich wichtigen Schritt hin zu einem dezentralisierten Umgang mit Dokumenten ist neben der Weitergabe der Administrationsmöglichkeiten von Berechtigungen damit der Transfer der Verantwortlichkeit für ein Dokument. Klassischerweise ist der Erzeuger eines Dokumentes zugleich der Besitzer des Dokumentes. In Bezug auf die Weitergabe der Besitzverhältnisse eines Dokumentes sind verschiedene Ansätze denkbar. Die meisten Varianten sind primär von einer geschickten Umsetzung in eine geeignete Benutzerführung auf Schnittstellenebene abhängig.

Das im Rahmen der vorliegenden Arbeit vorgeschlagene Modell führt ein eigenes Recht für die Weitergabe von Berechtigungen ein. Damit wird für jedes Objekt ein Recht für den Transfer des Besitzes eingeführt.

Dieses Recht kann an einzelne Personen oder Gruppen von Personen vergeben werden. Inhaber eines derartigen Rechtes sind in die Lage versetzt, den Besitz des Dokumentes an dritte Personen weiterzugeben und damit das Recht zur Administration von Berechtigungen zu erweitern.

Berechtigungen außerhalb des Modells – Administratorengruppe

Neben der sorgfältigen Strukturierung der Benutzer- und Gruppenstruktur erscheint es in der Praxis zwingend erforderlich eine Gruppe von Benutzern einzuführen, die ohne explizites Setzen von Berechtigungen in ACLs zu Objekten allgemein Zugriff auf jedes Objekt innerhalb der Umgebung ausüben können. Dies ist aus Gründen der Administration dringend erforderlich – bedenkt man, dass sich Personen den Zugriff zu ihren eigenen Dateien entziehen könnten – auch wenn ein derartiges pauschalisiertes Aussprechen von Rechten aus Datenschutzgründen durchaus problematisch erscheint.

In der sTeam-Konzeption wird ein Ansatz von spezifischen Gruppenrechten gewählt. Hierbei können für jede *Benutzergruppe* Rechte an allen Objekten innerhalb des Systems spezifiziert werden. Es handelt sich damit um ein vollständig konträres Vorgehen zur Spezifikation von Berechtigungen an Dokumenten für Benutzer und Gruppen.

Gruppenrechte erlauben es in dieser Form z.B. einer Gruppe von Administratoren, das Recht zur Modifikation von Berechtigungen an allen Dokumenten innerhalb des Systems zu gewähren. Gleichzeitig lassen sich verschiedene Formen von Kooperationsregeln definieren. So kann für eine Gruppe festgelegt werden, dass das Recht zum Lesen von Dokumenten für jedes Gruppenmitglied vorhanden ist.

Konkret sind eine Reihe von Berechtigungen für Nutzergruppen festzulegen, die elementare Auswirkungen auf die Beschaffenheit kooperativer Wissensräume haben können (z.B. die Leseberechtigung für alle Dokumente innerhalb eines Gruppenraums).¹³³

¹³³ Weiterhin lassen sich mit Hilfe der Gruppenberechtigung Administrations- und Erweiterungsmöglichkeiten des Systems festmachen.

Rechtegruppen

Motiviert durch eine große Anzahl an „collaboration rights“ [Shen & Dewan 1992], d. h. eine große Anzahl von Einzelberechtigungen, die in einem kooperationsunterstützenden System auftreten können, führen Shen und Dewan das Konzept der Rechtegruppen (Rights Groups) [ebd., S. 53ff.] ein. Hierbei wird eine transitive Enthaltensein-Beziehung für einzelne Berechtigungen definiert. Komplexe Berechtigungen lassen sich hierdurch auf Elementarrechte zurückführen.

Innerhalb des im Rahmen dieser Arbeit vorgeschlagenen Modells existiert keinerlei Beschränkung auf die Anzahl der Elementarrechte. Aus Gründen der Übersichtlichkeit und Verständlichkeit für die Benutzer sind jedoch die Anzahl der Elementarrechte möglichst klein zu halten, etwa von einer maximalen Anzahl von bis zu acht Berechtigungen.

Komplexere Berechtigungen werden durch spezielle Anwendungen oder Clients implementiert, z.B. die Berechtigung auf einen kooperativen Terminkalender schreiben zu können. Derartige Berechtigungen lassen sich gezielt auf Schreib- bzw. Ausführungsberechtigungen an speziellen Objekten zurückführen. Diese können von den Benutzerschnittstellen der kooperativen Applikationen spezifisch dargestellt werden. Das Kernsystem benötigt nicht zwingend ein neues, getrennt verwaltetes Recht.

Weitere Verfahren: z.B. geografische Verfahren

Es existieren vielfältige Möglichkeiten der Zuordnung von Berechtigungen zu räumlichen Strukturen einer kooperationsunterstützenden Umgebung. Hierzu zählen beispielsweise eine Reihe von Modellen, die sich an räumlichen Gegebenheiten orientieren.¹³⁴

Leider lässt sich über ein rein geografisches Verfahren der Vergabe von Berechtigungen nur der Zugang zu Objekten steuern, nicht die Handlungen an Dokumenten durch fein unterteilte Berechtigungen. Sinnvoll sind derartige Modelle sicherlich für den Zugang zu Räumen an sich. Historisch gesehen besteht eine Verbindung zum Level-Konzept der MUDs und MOOs, welche über einen Benutzer-Level den Zugriff auf verschiedene Ressourcen kontrollieren.¹³⁵

Für das Konzept virtueller Wissensräume lassen sich zukünftig basierend auf der im Rahmen dieser Arbeit entwickelten Kernarchitektur verschiedene Verfahren erproben. So können basierend auf Konzepten der Vererbung von Rechten aus dem Kontext Rechte lediglich an Räume gebunden und auf diese Weise ein einfaches geografisches/räumliches Modell des Setzens von Berechtigungen verwirklicht werden. Zukünftige Forschungsfragen ergeben sich zum einen in der Wahl geeigneter Vererbungsstrukturen für Berechtigungen (Vererbung vom Raum/Container), zum anderen aber auch in der Anzahl notwendiger Einzelberechtigungen.

¹³⁴ Vgl. das Verfahren von Bullock und Benford zur Vergabe von Zugangsrechten in virtuellen Umgebungen. Ihr „SPACE“-Modell [Bullock & Benford 1999] teilt die virtuelle Umgebung in geografische, durch Begrenzungen separierte Bereiche auf. Jede Grenze besitzt einen numerischen Wert, der von einem einzelnen oder einer Gruppe erreicht werden muss, um die Grenze überschreiten zu können. Konkret wenden Bullock und Benford ihr Verfahren auf verschiedene kooperative Umgebungen wie Orbit [Mansfield et al. 1999] und TeamRooms [Roseman & Greenberg 1996a] an.

¹³⁵ Weitere Modelle der Vergabe von Rechten orientieren sich an dynamischen Regeln des Kontextes, in dem sich Objekte befinden (vgl. [Trevor et al. 1994]).

Darstellung von Rechten und Rollen in der Benutzerschnittstelle

Neben der Formulierung eines möglichst einfachen, leicht verständlichen, aber gleichzeitig ausreichend komplexen Rechtemodells erscheint es unumgänglich, Berechtigungen fließend in die Benutzerschnittstelle der Lernumgebung zu integrieren. In der Literatur finden sich eine Reihe von interessanten Ansätzen der Abbildung von Berechtigungen auf einfache und einsichtige Metaphern. Ellis et al. stellen beispielsweise im Zusammenhang kooperativen Edierens ein Konzept der farblichen Abgrenzung von gesperrten Berechtigungen (Textbausteinen) vor.¹³⁶ Analog entwickeln Sohlenkamp und Chwelos ein einfaches, aber von der metaphorischen Umsetzung her interessantes Rechtemodell des Verschließens von Räumen.¹³⁷

Ziel verschiedener Ansätze ist es, möglichst einsichtige Metaphern für den Zugang zu Materialien zu entwickeln. Ein geeigneter Ansatz wäre beispielsweise die Vergabe eines Schlüssels zu Räumen und Dokumenten. Ein derartiges Modell ist für die Erprobung im Rahmen einfacher Einsatzszenarien geeignet, erweist sich jedoch im Bereich der Dokumentenverwaltung (Verwaltung größerer Bestände von Lehrmaterialien) als nicht ausreichend.

Nichtsdestotrotz erscheint der Anspruch, möglichst wenige Elementar-Rechte zu benötigen und möglichst einfache Metaphern zur Spezifikation von Berechtigungen zu entwickeln, als anstrebenswertes Kriterium kooperativer Wissensräume.

Rollen – an Benutzer gebundene Berechtigungen

Greif und Sarin diskutieren schon recht früh in ihren Arbeiten zum kooperativen Editor CES und kooperativen Kalendersystem PCAL den Einsatz von Rollen als Abstraktion verschiedener Zugriffsrechte entkoppelt von den Rechtemodellen des zugrunde liegenden Betriebssystems (vgl. [Greif & Sarin 1987, S. 199ff.]). Rollen sind geprägt durch Rechte zum Zugriff bzw. der Modifikation von Dokumenten und Objekten.¹³⁸ Analog definieren Ellis et al. Rollen als zugesprochene Privilegien, Verantwortlichkeiten und Attribute einer Person.¹³⁹

Eine Vielzahl von Content Management Systemen oder auch Hypertextsystemen verwenden Rollen als reine Abstraktion der Definition einer Kombination von Zugriffsrechten.¹⁴⁰ Das Rollenkonzept wird aber selbst in betriebswirtschaftlichen Anwendungskontexten als sinnvoll erachtet.¹⁴¹

¹³⁶ Hier erkennt ein Benutzer anhand „ausgegrauter“ Bereiche sofort, dass er keine Berechtigung zur Modifikation besitzt [Ellis et al. 1991].

¹³⁷ Berechtigungen für das Betreten von Räumen und den Zugriff auf Dokumente werden über einfache Zugriffslisten organisiert. Räume können den Status „open“, „locked“ und „shuttered“ besitzen. Bei dem Status „shuttered“ handelt es sich um eine Mischform eines Raums, der zwar verschlossen ist, aber nach Information der Personen innerhalb des Raumes durch ein akustisches Signal geöffnet werden kann.

¹³⁸ „A role [...] is an attribute of the user's computation that enters into the determination of whether or not to allow a given operation“ [Greif & Sarin 1987, S. 200].

¹³⁹ „A role is a set of privileges and responsibilities attributed to a person [...]. Roles can be formally or informally attributed.“ [Ellis et al. 1991, S. 46]

¹⁴⁰ Als Beispiel wäre z.B. das DAPHNE-System zu nennen, welches die Rollen „author“, „content examiner“, „publisher“ und „system administrator“ bereitstellt und in Form einer Matrix die einer Rolle zugesprochenen Berechtigungen festlegt [Zhang et al. 1999, S. 195].

¹⁴¹ Klienen et al. schlagen beispielsweise resultierend aus einem Verständnisproblem vieler Endanwender vor, in ihrem speziellen Anwendungsfall Benutzergruppen durch die Rollen „Kunde“, „Produktion“, „Litho“ und „Agentur“ abzubilden [Klienen 1999], [Klienen et al. 2000].

Zu einer Umsetzung der Rollen innerhalb kooperativer Wissensräume müssen Assoziationen zwischen Berechtigungen und Rollen sowie Assoziationen zwischen Rollen und Nutzern verwaltet werden.

Bezogen auf das entwickelte Modell von Benutzergruppen und Zugriffsrechten lassen sich Rollen durch flexibles Erzeugen spezieller Benutzergruppen bzw. flexibler Zuordnung von Nutzern zu Nutzergruppen realisieren. Einer entsprechenden Benutzergruppe zugesprochene Gruppenberechtigungen stimmen mit pauschalisiert vergebenen Rollenberechtigungen überein. Einer Rolle zugesprochene Berechtigungen an einzelnen Materialien werden durch Rechte der zugehörigen Benutzergruppe an den Objekten umgesetzt. Rollen können flexibel durch Zufügen einer Person zu einer Benutzergruppe vergeben und entsprechend entzogen werden. (Hierzu eignen sich die schon beschriebenen Fähigkeiten der Administration von Gruppen durch Gruppenkennwörter etc.)

Dahingehend liegt der wesentliche Aufwand der Bereitstellung von Rollen in einer Visualisierung in der entsprechenden Benutzerschnittstelle sowie der Abstimmung der gewählten Rollen auf die Struktur des Lernprozesses.¹⁴²

Gruppenprotokolle, Floor Control, Concurrency Control

Bei der Interaktion von Lernenden oder allgemein von Kooperationspartnern innerhalb einer Umgebung spielt die Definition von sozialen Regeln und Verhaltensweisen eine wichtige Rolle. Zum einen müssen Mechanismen bereitgestellt werden, die es Menschen erlauben ihre gewohnten, sozialen Verhaltensweisen und Regeln in eine virtuelle Umgebung und die entsprechenden Kommunikationskanäle zu übertragen. Gleichzeitig sind eine Reihe von Protokollen bereitzustellen, die es einer Gruppe erlauben sich zu koordinieren. Ellis et al. sprechen in diesem Zusammenhang von „social protocols“ (vgl. [Ellis et al. 1991, S. 51]).

Allgemein definiert die Vermeidung von Konfliktsituationen in der Gruppeninteraktion das Feld des Floor Control, also den koordinierten Zugriff auf von der Gruppe gemeinsam zu nutzenden Ressourcen.¹⁴³ Floor Control umfasst also sämtliche Mechanismen der Steuerung von Koordinations- und Interaktionsprozessen innerhalb eines kooperationsunterstützenden Systems.¹⁴⁴

Interessanterweise wenden sich insbesondere Nutzer von kooperationsunterstützenden MUDs in der Lehre, wie z.B. Fanderclai, gegen spezielle Turn-Taking-Protokolle und jegliche Mechanismen der Steuerung der Kommunikation innerhalb eines MUD (vgl. [Fanderclai 1995]). Für sie wird speziell durch Kontrollen der Redemöglichkeit oder Mechanismen der Wortmeldung („Hand Raising“) der Anspruch einer frei fließenden Kommunikation bzw. Diskussion empfindlich gestört.

Entsprechend dem Floor Control treten in kooperativen Handlungen Konfliktsituationen des gemeinsamen Zugriffs auf Materialien auf (Concurrency Control).

Die Forschungen zum verteilten Zugriff auf gemeinsame Daten wurden in der Vergangenheit zu einem großen Teil aus den Bereichen der Entwicklung kooperativer Editoren (Collaborative Editing Systems), speziell kooperativer Textsysteme oder koopera-

¹⁴² Von einer weiterführenden theoretischen Betrachtung des Rollenkonzepts z.B. in einer formalen Definitionssprache bzw. mathematischen Mengenschreibweise sei an dieser Stelle abgesehen (vgl. hierzu [Li & Muntz 1998] und [Gavrila & Barkley 1998]).

¹⁴³ Vgl. die Definition von Dommel und Aceves: „Floor control is a technology to deal with conflicts within shared workspaces. It helps to coordinate joint and competing activities among people and their interacting computational processes, such as regulating turn-taking in conversations or write-updates on shared files, preserving coherency of local and remote information“ [Dommel & Aceves 1997, S. 24].

¹⁴⁴ Eine Zusammenstellung verschiedener möglicher Protokolle des Floor Control („Free Floor“, „Pre-emptive“, „Explicit Release“, „First In First Out“) findet sich in [Greenberg 1991, S. 22].

tiver grafischer Editoren, geprägt. Hier treten Konfliktsituationen des konkurrierenden Zugriffs (Concurrency Control) durch eine enge Form der gegenseitigen Zusammenarbeit besonders in Erscheinung.¹⁴⁵ In der Literatur finden sich eine ganze Reihe von zum Teil technischen Konzepten zur Vermeidung von Konfliktsituationen beim Zugriff auf gemeinsam genutzte Objekte.

Hierzu zählen: „Simple Locking“, „Transaction Mechanisms“, „Turn-Taking Protocols“, „Centralized Controller“, „Dependency Detection“, „Reversible Execution“ und „Operation Transformations“ (vgl. [Ellis et al. 1991, S. 53] und [Ellis et al. 1989]).

Basierend auf der Erkenntnis, dass es sich bei der Implementierung von Verhaltensweisen und Kontrollstrukturen innerhalb von kooperativen Lernprozessen um eine Implementierung von sekundären Medienfunktionen handelt, sind Mechanismen des Floor Control für das technische Rahmenkonzept eines kooperationsunterstützenden Systems nicht von vorrangiger Bedeutung und werden im Folgenden nicht weiter untersucht. Auf die Praxis kooperativer Wissensräume übertragen erscheint das einfache Blockieren von Materialien („Simple Locking“) zum Zeitpunkt des Zugriffs durch einen Nutzer praktikabel. Ein diesbezügliches optimistisches Verfahren im Umgang mit kooperativen Objekten genügt sowohl in primär asynchronen Lernumgebungen, aber auch in synchronen Teilbereichen, z.B. auf einem Shared Whiteboard.¹⁴⁶

Nachfolgend der Untersuchung möglicher Metaphern des virtuellen Raums, Konzepten der Awareness und einem Modell von Benutzerrechten ist ein nächster wesentlicher Schritt die Einbettung von Materialien innerhalb kooperativer Wissensräume. Wesentliche Aufgabe fällt hierbei der objektorientierten Repräsentation von Materialien in Objekten und Attributen zu. Die nächsten Abschnitte untersuchen speziell die bereitgestellten Medienfunktionen im Hinblick auf ein derartiges Modell.

3.2 Kernkonzept kooperativer Medienfunktionen

Wie aus den vorgestellten Szenarien in Kapitel 2 ersichtlich, lassen sich eine Reihe von Grundfunktionalitäten und Metaphern isolieren, die kooperative Wissensräume charakterisieren. Hierzu zählt insbesondere die Integration von synchronen und asynchronen Kommunikations- und Interaktionsformen. Synchrone Interaktionsformen finden sich beispielsweise in Situationen der Präsentation von Vorlesungsmaterialien, der Nutzung eines Shared Whiteboards oder der synchronen Kommunikation (Chat). Asynchrone Interaktionsformen sind in allen kooperativen Handlungen an Materialien innerhalb der Lernumgebung anzutreffen.

Zum Verständnis des Konzeptes der Einbettung primärer Medienfunktionen innerhalb von kooperativen Lern- und Arbeitsräumen ist eine weitere Differenzierung des zeitlichen Ablaufs (synchron/asynchron) und der existierenden Interaktionsformen notwendig:

- I. Differenzierung: Es gilt zu unterscheiden, ob kooperativ zu bearbeitende Materialien durch denselben Nutzer zu verschiedenen Zeiten bearbeitet werden (individuelle, asynchrone Interaktion zwischen Nutzer und Objekt), Materialien zu verschiedenen Zeitpunkten von verschiedenen Nutzern bearbeitet werden (kooperative, asynchrone Interaktion zwischen Nutzern und Objekt) oder eine

¹⁴⁵ Vgl. [Ellis et al. 1989] und [Chen & Sun 1999].

¹⁴⁶ Vgl. hierzu die Systeme von [Schuckmann et al 1996, S. 31] und [Sohlenkamp & Chwelos 1994], die jeweils ebenfalls ein optimistisches Zugriffsprotokoll implementieren.

Anzahl von Nutzern zeitgleich auf dieselben Materialien zugreifen (kooperative, synchrone Interaktion mit Objekten).

- II. Differenzierung: Die verschiedenen Fälle möglicher Interaktionsformen sind des Weiteren entlang der ausübenden oder betroffenen Interaktionspartner zu differenzieren. Hier gilt es zu unterscheiden, ob Interaktionen zwischen einem Nutzer (Lernenden) und Materialien stattfinden oder Interaktionen zwischen den Nutzern untereinander erfolgen. Im ersten Fall der Interaktion von Nutzern und Materialien kann zudem der Fall ergänzt werden, dass eine Interaktion des Nutzers mit aktiven Elementen der Lernumgebung, z.B. einem elektronischen Briefkasten, einem elektronischen Abgabefach für Übungsblätter oder Protokollwerkzeugen stattfindet.

Ziel der vorliegenden Rahmenkonzeption ist es, sowohl die konzeptionelle als auch technische Umsetzung der Verwaltung von Materialien innerhalb kooperativer Wissensräume zu vereinheitlichen. Die funktionalen Anforderungen an virtuelle Wissensräume (Abschnitt 2.6) erfordern ein leistungsfähiges Konzept der Einbettung von Materialien (und primären Medienfunktionen), Nutzern und Interaktionsformen innerhalb des virtuellen Raums. Der im Rahmen dieser Arbeit entwickelte Ansatz beinhaltet hierzu ein fortschrittliches Objekt- und Attributkonzept: Beliebige Materialien werden als Objekte repräsentiert. Sie besitzen einen Inhalt, der von einem einfachen Text im Falle eines Textdokuments bis zu einer Anzahl weiterer Objekte im Falle eines Ordners/Containers reichen kann. Materialien (Objekte) besitzen zudem eine Reihe von Attributen und eine Liste von Zugriffsrechten.

Nicht nur existierende Funktionen, sondern auch durch Nutzer ergänzte Möglichkeiten orientieren sich an den Rahmenbedingungen und einem Verständnis der Lernumgebung als Verbund kooperierender Objekte. Das gesamte System, seine virtuellen Räume und die jeweiligen Rauminhalte werden in dieser Form als eine Struktur aus Objekten abgebildet. Ein Objekt kann sowohl ein einzelner Raum als auch ein Container oder ein Textdokument sein. Zugangswerkzeuge erzeugen verschiedene Sichten auf die erzeugte Objektstruktur.

Im Folgenden werden exemplarisch verschiedene Handlungen an Materialien innerhalb kooperativer Wissensräume konzeptionell umgesetzt. Es werden die Medienfunktionen des Kreierens, Arrangierens, Verknüpfens und Übertragens auf den konkreten Umgang mit Objekten abgebildet.

Zu einer weiterführenden technischen Ausformulierung dieser Medienfunktionen vergleiche die in Abschnitt 5.2.1 gezeigte Objektstruktur.

3.2.1 Zugrundegelegtes Objektmodell

Die Medienfunktion des Kreierens als Basis eines Wahrnehmungsraums für den Lernenden bezieht sich auf eine Vielzahl unterschiedlicher Materialien. Dies können symbolische Beschreibungen, Grafiken, Zeichnungen, Visualisierungen von mathematischen Zusammenhängen, aber auch Texte sein. Diese Vielfalt der möglichen Medientypen macht es schwierig, das Erzeugen von verschiedenen Materialien vollständig in ein System einzubetten. So ist das Erstellen eines virtuellen Raums genauso eine Medienfunktion des Kreierens wie beispielsweise das Schreiben eines Kommentars zu einem Dokument oder das Anlegen eines Textdokumentes.

Hinzu kommt, dass kooperativ genutzte Materialien, wie sie Teil kooperativer Wissensräume sind, in ihrer Struktur schwieriger zu realisieren sind als individuell genutzte Materialien. Kooperative Materialien sind durch Nutzerrechte gegeneinander abzugren-

zen und besitzen eine viel höhere Zahl an Attributen gegenüber einem lediglich individuell genutzten Dokument.

Das im Rahmen dieser Arbeit entwickelte Konzept virtueller Wissensräume bildet Materialien, aber auch virtuelle Räume und semantische Strukturen über Objekte ab. Ein Objekt besteht aus einer Objektkapsel, einem Inhalt, einer Umgebung, Attributen und Benutzerrechten (vgl. Abbildung 3-5). Die Objektkapsel repräsentiert das jeweilige Objekt innerhalb der Lernumgebung. Der Inhalt ist mit der Objektkapsel verknüpft und nimmt entsprechend verschiedene digitale Medien auf. Diese werden persistent innerhalb einer Datenbank abgelegt. Eine Anzahl von Attributen charakterisieren das Objekt. Benutzerrechte kontrollieren den Zugriff auf das Objekt in einer kooperativen Zugriffsstruktur.

Damit ist eine wichtige Grundstruktur zur Selbstadministration einer Lernumgebung durch die Lernenden gelegt. Bezieht sich die Idee der Selbstorganisation in einer ganzen Reihe von Systemen lediglich auf das Einrichten neuer Diskussionsgruppen oder Einladen von Nutzern zu Arbeitsbereichen, erstreckt sich Selbstorganisation im Rahmen der vorgestellten Konzeption ebenfalls auf das Erzeugen von beliebigen Materialien und Handlungen an Objekten. Wesentliches Element ist die Verwaltung und Weitergabe von Berechtigungen an Objekten. Durch eine Abbildung aller wesentlichen Elemente eines kooperationsunterstützenden Systems durch Objekte (Räume, Ordner, Materialien) ist die Möglichkeit zur verteilten nutzerzentrierten Selbstorganisation gegeben.

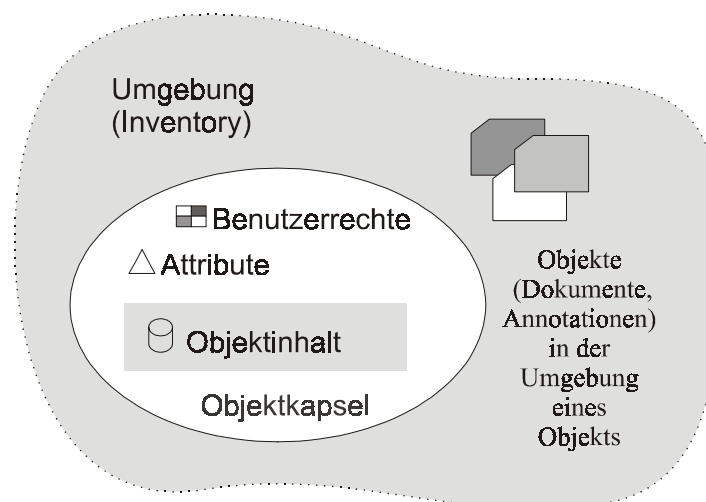


Abbildung 3-5: Objektkapsel, Inhalt und Umgebung

Objektkapsel

Die Objektkapsel bildet eine Umschließung (Hülle) für das jeweilige Objekt. In ihr sind Attribute und Objektinhalt enthalten. Jedes Objekt besitzt einen Namen, eine eindeutige Nummer und eine WWW-konforme Adresse. Diese ist unabhängig von dem Ablageort des Objektes frei spezifizierbar (vgl. Abschnitt 5.2.7). Alle genannten elementaren Objekteigenschaften sind innerhalb von Attributen abgelegt (vgl. Abschnitt 5.2.3).

Umgebung

Eine Reihe von Objekttypen können ähnlich einem Behältnis in ihrer unmittelbaren Umgebung weitere Objekte kapseln.¹⁴⁷ Eine Vielzahl der innerhalb der vorgestellten Rahmenarchitektur realisierten Metaphern und Vorgehensweisen basieren auf diesem Konzept. Beispielsweise handelt es sich bei einem Container-Objekt, welches weitere Objekte zusammenfasst, lediglich um eine Variante eines herkömmlichen Objekts. Ein Raumobjekt ist eine leicht abgewandelte Ableitung des Container-Objekts. Auch der Rucksack eines Benutzers, in dem beliebige andere Objekte aufbewahrt werden können, ist damit eine Eigenschaft des Benutzerobjekts (vgl. Abbildung 5-1).

Eine mögliche Realisierung von Annotationen an Materialien ist damit, Annotationen in der Umgebung des entsprechenden Objekts abzulegen und sie in dieser Form auf natürliche Weise mit den jeweiligen Materialien zu verknüpfen.¹⁴⁸

Attribute

Attribute speichern Strukturinformationen und Metainformationen zu dem jeweiligen Objekt. Es existieren eine Reihe von Attributen, die jedes Objekt besitzt, wie Name, Erstellungsdatum, Objekttyp etc., aber auch Attribute, die sehr speziell dem jeweiligen Objekttyp zugeordnet sind, wie z.B. eine Internetadresse im Falle eines Referenzobjektes.

Attribute beschreiben in dieser Form das Objekt in Abhängigkeit einer Anzahl von Charakteristika. Diese Metainformationen zu dem jeweiligen Objekt erlauben die Strukturierung und Koordination kooperativer Prozesse.

Beliebige Attribute können von den Schaffern eines neuen Objekttyps den Standard-Attributen hinzugefügt werden (vgl. hierzu die Factories in Abschnitt 5.2.2). Damit ist der Attributmechanismus der Schlüssel für eine Vielzahl von Funktionen innerhalb kooperativer Wissensräume (sTeam-Umgebung).

Zugangswerkzeuge, wie grafische Clients, nutzen die Attribute der jeweiligen Objekte, um beispielsweise Informationen über die Position des Objektes innerhalb des Raumes abzubilden. Ebenfalls innerhalb der Attribute spezifizierte Schlüsselwörter zu Materialien erlauben z.B. eine gezielte Suche von Objekten.

Objektinhalt

Der Inhalt (Content) eines Objekts enthält die digitale Repräsentation des zugeordneten Mediums. Verschiedene Materialien werden prinzipiell in ein und demselben Objekttyp „Dokument“ abgelegt. Dieser ist damit vorerst nicht auf eine feste Anzahl von Medientypen beschränkt. Unterschiedliche Typen von Materialien lassen sich anhand spezieller Attribute differenzieren. In seiner Struktur unterscheiden sich verschiedene Materialien damit zunächst nicht. Erst durch ein Typ-Attribut werden entsprechende Betrachtungs- oder Bearbeitungswerkzeuge mit dem Inhalt verknüpft. Die Persistenzeigenschaften des Servers wirken sich weitgehend unabhängig von dem jeweiligen Inhalt auf das entsprechende Objekt aus.

Durch spezifische Objektattribute, wie z.B. einem Hypertextdokument (HTML), lassen sich spezielle Behandlungen erzielen.¹⁴⁹

¹⁴⁷ Das Konzept der Umgebung von Objekten stammt aus dem „Inventory“-Ansatz der MUDs und MOOs. Hier trägt beispielsweise jeder Benutzer eine Anzahl von Gegenständen mit sich.

¹⁴⁸ In der gewählten Konzeption werden Annotationen jedoch in Form einer Verweisliste an das entsprechende Objekt gelöst, um sie unabhängig von dem entsprechenden Objekt verwalten zu können (vgl. Abschnitt 5.2.4).

Zugriffsrechte

Innerhalb einer kooperativen Lern- und Arbeitsumgebung birgt die gemeinsame Nutzung von Objekten eine weitere Dimension der Komplexität in sich. Nutzungsrechte und Zugriffsschutz sind, wie schon gezeigt, zentrale Themen kooperativer Wissensräume.

Auf das Objektmodell übertragen kontrolliert eine Liste von Benutzer- und Gruppenrechten den Zugriff auf das entsprechende Objekt (die Materialien). Zugriffsrechte sind entweder direkt an das Objekt gebunden oder werden aus der Objekthierarchie vererbt. Sie steuern die Einschränkung/Bereitstellung der primären Medienfunktionen auf das zugehörige Objekt und auf die Objekte innerhalb seiner Umgebung. Dies bedeutet beispielsweise, dass der Zugang zu einem Raum gleichbedeutend einer zugebilligten Leseberechtigung für das Raumobjekt ist. Das Erzeugen eines neuen Objekts, z.B. durch Ablage einer existierenden Datei in einem Raum, entspricht einer Einfügeberechtigung (Insert-Recht) auf das Raumobjekt bzw. auf ein zugeordnetes Container-Objekt. Durch die universelle Verknüpfung von Berechtigungen an sämtliche Objekttypen sind beliebige Medienfunktionen kontrollierbar. Damit lässt sich auch das Kreieren von wesentlichen Elementen (z.B. das Erzeugen eines neuen Raumes) in die Hände von autorisierten Lernenden legen und das Ziel der Selbstadministration und Selbstorganisation der Lernumgebung umsetzen.

Die folgenden Ansätze untersuchen die konkrete Einbettung primärer Medienfunktionen in das Objektmodell kooperativer Wissensräume.

3.2.2 Kreieren und Löschen

Wie aus den skizzierten Szenarien ersichtlich, existieren eine Vielzahl von Tätigkeiten im Umgang mit Lehrmaterialien, die der Medienfunktion des Kreierens entsprechen. In kooperativen Szenarien gilt es im Bereich des Kreierens eine Reihe von Designkonflikten zu lösen: Kreieren von Materialien oder auch Erzeugen von Strukturierungselementen kooperativer Wissensräume (Ordner/Container oder ganze Räume) kann die Struktur der Lernumgebung erheblich verändern und damit starke Auswirkungen auf die Arbeitsumgebung anderer Mitlernender haben. – In vielen Situationen ist ein starkes Verändern der Lernumgebung nicht erwünscht, um die Übersichtlichkeit/vorgegebene Struktur nicht zu gefährden.

Gleichzeitig ist es erklärtes Ziel eines nutzerzentrierten Ansatzes, den Lernenden genau diese Freiheiten des Schaffens eigener virtueller Räume und Strukturen zu ermöglichen. Eine zunächst technische Lösung kann den obigen Designkonflikt für den konkreten Anwendungskontext nicht vorgeben, sie kann aber zumindest die technischen Grundlagen zu einem nutzerzentrierten Kreieren von Materialien, Containern und Räumen liefern. In Verbindung mit dem schon dargestellten Rechtemodell (Abschnitt 3.1.5), z.B. der Weitergabe von Berechtigungen bzw. der Verantwortung für Objekte, erfüllt der gewählte Architekturansatz diesen Anspruch.

¹⁴⁹ Dies ist etwa für Hypertextdokumente (WWW) notwendig, um die festen Adressen integrierter Verweise auf dynamische Adressen innerhalb des sTeam-Systems umsetzen zu können. Die hierzu notwendige Funktionalität ist in die Persistenzschicht des Servers eingelagert und ist für den Lernenden transparent.

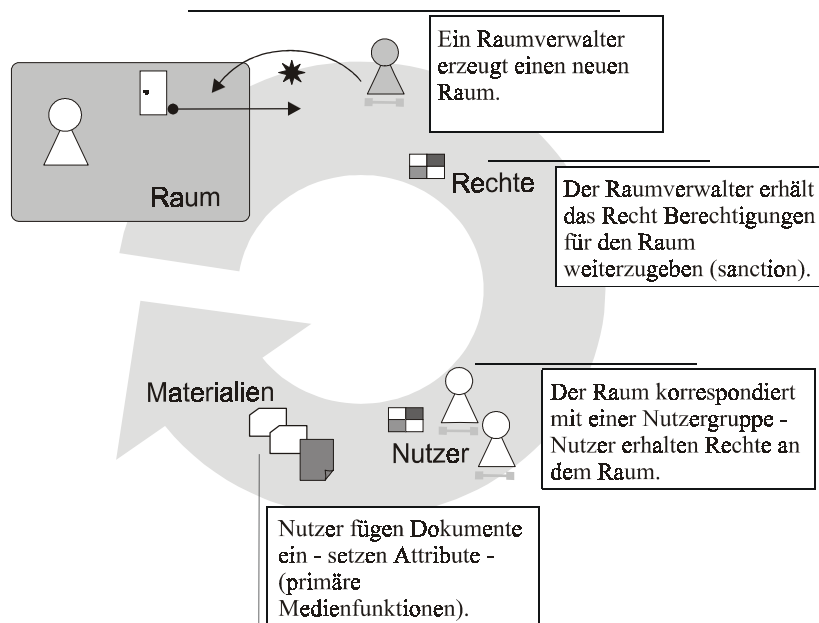


Abbildung 3-6: Selbstadministration – Erzeugen eines Raums
– Weitergabe von Berechtigungen

Das vorgestellte Objektmodell bildet die Medienfunktion des Kreierens und Löschens von Materialien durch das Erzeugen und Entfernen eines Objektes mit entsprechenden Attributbelegungen und Konstellationen der Benutzerrechte ab. Folgende Interaktionsformen im virtuellen Raum erzeugen beispielsweise neue Objekte:

- Erzeugen eines Raums (vgl. Abbildung 3-6)
- Erzeugen eines Containers/Ordners
- Ablegen eines Dokumentes in einen Raum (Upload)¹⁵⁰
- Erzeugen von Annotationen zu Materialien
- Erzeugen einer Referenz auf ein externes WWW-Dokument
- Erzeugen einer Referenz auf innerhalb der Lernumgebung abgelegte Materialien
- Kopieren eines Objekts (Grafik, Textdokument etc.)

Die obige Liste kann beliebig fortgesetzt werden. Objekte entstehen in einer Vielzahl von Situationen kooperativer Lernprozesse, in denen neue Materialien erzeugt werden; aber auch Tätigkeiten des Verknüpfens von Materialien (Annotation, Referenz) erzeugen spezifische neue Objekte.

Wie im Bereich der Factories (vgl. Abschnitt 5.2.2) noch gezeigt wird, ist das Erzeugen von Objekten an eine Reihe von Bedingungen geknüpft. Hierzu zählen zum einen Berechtigungen der entsprechenden Klasse, zum anderen aber auch Berechtigungen an der Umgebung, in dessen Kontext ein Objekt erzeugt wird. In Verbindung mit dem in Abschnitt 3.1.5 dargestellten Modell von Nutzergruppen und Rechten lassen sich die in Kapitel 2 formulierten Anforderungen der Selbstorganisation kooperativer Wissensräume in Bezug auf die Medienfunktion des Kreierens umsetzen.

¹⁵⁰ Dies kann beispielsweise über eine Web-Schnittstelle, ein Zugangswerkzeug (Java-Client) oder eine Netz-Schnittstelle (FTP/ WebDAV-Zugang) geschehen, vgl. Abschnitt 5.3.

Nutzern ist es prinzipiell möglich, beliebige Materialien zu erzeugen; dieser Vorgang entspricht dem Erzeugen eines entsprechenden Dokument-Objekts. Gleichzeitig wird das Kreieren von Strukturierungselementen kooperativer Wissensräume, wie das Erzeugen eines Ordners (Containers) oder eines neuen virtuellen Raums, in gleicher Weise auf das Kreieren eines Container- bzw. Raum-Objekts mit entsprechenden Attributen abgebildet. Der Prozess des Kreierens entsprechender Objekte, beispielsweise eines Raumobjekts, löst automatisch notwendige Aktionen in Bezug zu anderweitigen Objekten (Materialien) aus. So führt das Erzeugen eines neuen Raums automatisch zu dessen Verknüpfung mit dem Ursprungsraum über Verbindungstüren (Exits).

Das Ablegen von Dokumenten innerhalb kooperativer Wissensräume entspricht ebenfalls dem Kreieren von Objekten.¹⁵¹ Hierbei ist es erforderlich, Attribute und Benutzerrechte voreingestellt zu wählen bzw. nachfolgend durch die Nutzer zu setzen. Die Mehrzahl von Attributen lässt sich aus den zur Verfügung stehenden Informationen (Namen, Autor, Datum, Rechte des Autors etc.) automatisiert wählen bzw. durch das Zugangswerkzeug erzeugen. Benutzerrechte werden zunächst entsprechend den Rechten des angemeldeten Nutzers und geeigneten Voreinstellungen gesetzt.

Konkrete Beispiele für das Ablegen von Dokumenten in virtuellen Wissensräumen sind das Laden (Upload) eines Dokumentes per Browser in einen virtuellen Raum, das Ablegen eines Dokumentes (z.B. einer Folienpräsentation) aus einer Anwendung (Office-Applikationen) oder das Übertragen von Dateien (beispielsweise eine in HTML oder XML vorliegende Folienpräsentation) über einen dateiorientierten (z.B. FTP-) Client. – In allen Fällen werden neu eingefügte Materialien als Objekte mit entsprechenden Attributen repräsentiert und persistent innerhalb kooperativer Wissensräume gespeichert.

3.2.3 Arrangieren

Die Tätigkeit des Arrangierens setzt verschiedene Materialien in Beziehung. Sie hilft Lernenden Sinnzusammenhänge herzustellen. Hierzu sind Materialien nach Möglichkeit gleichzeitig in das Wahrnehmungsfeld des Lernenden zu bringen. In der Mehrzahl der gewählten Repräsentationsformen lassen sich logische Zusammenhänge, wie eine Verbindung zwischen zwei Themengebieten, durch räumliche Nähe der entsprechenden Objekte (Piktogramme etc.) abbilden.

Die Medienfunktion des Arrangierens findet sich in einer ganzen Reihe von Tätigkeiten innerhalb kooperativer Wissensräume und damit an wesentlichen Stellen der technischen Konzeption. Beispiele sind das Gruppieren von Materialien in Ordnern/-Containern, sind aber auch die kooperative Strukturierung von Verweisobjekten auf einem Shared Whiteboard oder die Raumdarstellung.

Im Bereich der formulierten Lernszenarien (vgl. Kapitel 2) sind eine ganze Reihe von Tätigkeiten und Interaktionsformen zwischen Lernenden identifiziert, die sich der Ausübung der Medienfunktion des Arrangierens zuordnen lassen. Im Folgenden werden diese aus konzeptioneller Sicht in Bezug auf die Bereitstellung von Ausdrucksformen der Medienfunktion des Arrangierens hin untersucht:

- *Gruppieren von Objekten in Ordnern/Containern und Räumen:* Materialien werden durch Container-Objekte logisch zusammengefasst. Ein Raum ist eine spezielle Form des Containers. Objekte innerhalb eines Containers sind gemein-

¹⁵¹ Schnittstellen und Zugangswerkzeuge des Servers (vgl. Kapitel 5) ermöglichen es, verschiedene elektronische Materialien aus einer lokalen Arbeitsumgebung des Lernenden innerhalb kooperativer Wissensräume abzulegen. Hierzu werden durch eine Reihe möglicher Zugangswerkzeuge Sichten auf einen virtuellen Raum erzeugt.

sam manipulierbar, d.h. eine Zusammenfassung von Objekten stellt verschiedene primäre Medienfunktionen bereit. Ein Container-Objekt besitzt Attribute und Rechte, es speichert eine beliebige Anzahl von anderen Objekten in seiner Umgebung. Container-Objekte sind persistent und besitzen bis auf wenige Ausnahmen eine grafische oder textuelle Repräsentation, auf die mittels eines Zugangswerkzeugs Sichten erzeugt werden. Container-Objekte können Eigenschaften auf die enthaltenen Objekte vererben und können beliebig geschachtelt werden, d.h. ein Container kann eine Anzahl weiterer Container enthalten.

- *Räumliches Arrangieren von Objekten innerhalb von Containern oder Räumen (asynchron/synchron):* Die Repräsentation eines Containers kann sowohl in linearer bzw. eindimensionaler Weise (z.B. eine Liste des Inhalts) als auch mehrdimensional erfolgen (z.B. planare, räumliche Anordnung von Objekten innerhalb eines Containers). Dokumente und Grafiken werden beispielsweise im Falle einer räumlich-grafischen Darstellungsform durch ihre räumliche Nähe zueinander arrangiert. Hier ergibt sich neben der Darstellung von räumlichen Beziehungen der Materialien zueinander speziell die Anforderung der Navigierbarkeit und Rückkopplung von Navigationsschritten an den Benutzer.
- *Gruppieren von Materialien in der Umgebung von speziellen Objekten (Gruppierungsobjekte):* Die Umgebung eines speziell ausgewiesenen Objekts erlaubt das Gruppieren, Arrangieren von weiteren Objekten. Prominentes Beispiel ist der Rucksack, welcher es dem Lernenden erlaubt, eine Anzahl von Objekten mit sich zu „tragen“ und gezielt zu manipulieren. Weitere Beispiele sind Briefkästen zur Abgabe von Übungsblättern oder ein Whiteboard-Objekt, welches z.B. den Inhalt (Materialien) einer gemeinsamen Zeichenfläche kapselt. Eine derartige universelle Gruppierungs- und Arrangierungsfunktionalität ist auf Objektebene des Kernkonzepts kooperativer Wissensräume verankert.
- *Strukturieren/Arrangieren von Kommunikation (asynchron/synchron):* Speziell in asynchronen Diskussionssituationen ist eine thematische Zuordnung von Diskussionsbeiträgen und Materialien (Objekten) notwendig. Zum einen können Annotationsmechanismen an Objekten den Charakter einer fortlaufenden Diskussion erhalten (asynchroner Fall), zum anderen ist der virtuelle Raum die natürliche Begrenzung für eine synchrone Kommunikation, d.h. er gruppiert die Diskussionsbeiträge der anwesenden Lernenden. Elementares Strukturierungsinstrument der Kommunikation ist damit ein beliebiges Objekt (Raum, Container, Dokument oder Grafik), welches im asynchronen Fall durch fortlaufende Annotation eine Anzahl von Diskussionsbeiträgen gruppiert.

3.2.4 Verknüpfen

Das Verknüpfen von Objekten ist eng mit der Tätigkeit des Arrangierens verbunden. Die dargestellten Szenarien zum kooperativen Lernen haben gezeigt, dass die Medienfunktion des Verknüpfens auf einer Reihe von Bedingungen basiert, die so von den meisten untersuchten Systemen (vgl. Abschnitt 2.7) nicht oder nur in Teilaspekten erfüllt werden.

Verknüpfen bezieht sich ähnlich wie Arrangieren auf die Herstellung semantischer Beziehungen zwischen Objekten. Gegenüber dem Arrangieren schafft eine Verknüpfung eine um einiges festere, dauerhafte Verbindung zwischen Materialien (Objekten).

Als Anforderungen an die konzeptionelle Umsetzung der Medienfunktion des Verknüpfens sind eine Reihe von Eigenschaften zu nennen, welche das verfolgte Objektmodell architektonisch umsetzt:

- *Herstellen von Verbindungen zwischen einem Ausgangs- und einem Zielobjekt:* Verknüpfungen müssen unabhängig von der Beschaffenheit des Zielobjekts zu erstellen sein, es darf keinerlei Abhängigkeiten von der Art des Zielobjekts geben. Dies bezieht sich insbesondere auf innerhalb und außerhalb kooperativer Wissensräume angesiedelte Objekte, d.h. es muss möglich sein, neben Referenzen auf Objekte innerhalb von virtuellen Räumen auch Referenzen auf beliebige Objekte des WWW abzulegen. Verknüpfungen zwischen virtuellen Räumen werden als „Exits“ bezeichnet und verbinden einen Ausgangsraum mit einem Zielraum.
- *Primäre Medienfunktionen eines Verknüpfungsobjekts:* Bei dem Verknüpfungsobjekt handelt es sich um ein eigenständiges Objekt, welches in seiner Inhaltskomponente eine Referenz auf das Zielobjekt enthält. Damit implementiert es wie jedes anderweitige Objekt (Materialien) primäre Medienfunktionen, d.h. Verknüpfungen können erzeugt, arrangiert, wiederum verknüpft, übertragen, zugegriffen und synchronisiert werden.
- *Verknüpfungen sind unabhängig der schreibenden Berechtigungen für das Zielobjekt:* Verknüpfungen sind an ein explizites Recht zur Verknüpfung oder aber an die Leseberechtigung des Zielobjektes gebunden.
- *Verknüpfungen stellen ein vollwertiges Objekt dar, d.h. besitzen Attribute und Berechtigungen:* Ein Verknüpfungsobjekt ist als vollständig eigenständiges Objekt zu verstehen. Es besitzt eine Reihe von Attributen, hierzu zählen z.B. Metainformationen, wie Schlüsselworte oder der Autor der Verknüpfung. Verknüpfungsobjekte besitzen eigene, zunächst von den Berechtigungen des Zielobjektes unabhängige Rechte.
- *Verknüpfungen sind unabhängig von den Modifikationen des Zielobjektes:* Durch eine schon genannte Entkopplung von Berechtigungen existieren primäre Medienfunktionen unabhängig voneinander für Verknüpfungsobjekt und Zielobjekt. Über das Verknüpfungsobjekt erfolgt ein Zugriff auf das Zielobjekt.
- *Verknüpfungen sind unabhängig von der eigenen Position und der Position des Zielobjektes:* Ein Verknüpfungsobjekt stellt eine semantische Verbindung zwischen Zielobjekt und dem Verknüpfungsobjekt her, die unabhängig von der Position der beiden Objekte innerhalb der Umgebung ist.¹⁵²
- *Verknüpfungen sind bidirektional, besitzen einen Mechanismus der „Überwachung“ des Zielobjekts:* Verknüpfungen besitzen den Charakter einer bidirektionalen Verbindung zwischen zwei Objekten (Materialien). Nicht nur das Verknüpfungsobjekt ‚kennt‘ sein Zielobjekt, es lassen sich ebenfalls für jedes Objekt die auf ihn verweisenden Verknüpfungsobjekte ermitteln. Verknüpfungsobjekte können beispielsweise über Modifikationen, wie Löschen des Zielobjekts, informiert werden; auf diese Weise werden Inkonsistenzen durch ungültige Verknüpfungsobjekte vermieden.¹⁵³

Eine ganze Reihe von Möglichkeiten des vorgestellten Rahmenkonzepts einer kooperationsunterstützenden Umgebung basieren, wie schon in den verschiedenen Szenarien zu Lehrunterstützung genannt, auf der Verknüpfung von Objekten. Beispiele sind:

¹⁵² Eine Umsetzung dieser Anforderungen für Verknüpfungen auf Objekte innerhalb des WWW ist technisch aufgrund der konzeptionellen Einschränkungen des zugrunde liegenden HTTP-Protokolls nur bedingt möglich (vgl. Abschnitt 5.3.2).

¹⁵³ s. vorherige Fußnote.

- *Anlegen von Trails, d.h. von geführten Wegen durch Materialien:* Spezifische Zugangswerkzeuge können Referenzen auf externe Web-Seiten durch Verknüpfungen realisieren. Damit ist die konzeptionelle Basis für einen geführten Weg (Trail) durch Materialien gegeben.
- *Kooperatives Browsen – Anlegen von Verweisen:* Verweise können, wie in Szenarien des kooperativen Browsens beschrieben, in der Gruppe von Lernenden angelegt und ausgetauscht werden. Verweisobjekte besitzen in diesem Fall Lese- und Schreibberechtigung für die jeweilige Gruppe.
- *Erzeugen von persönlichen Referenzlisten (Bookmarks) oder kooperativ genutzten Verweisen:* Von den Nutzern werden individuell oder in der Gruppe Sammlungen von Verweisen angelegt. Hierbei lassen sich beispielsweise Verweise auf Vorlesungsmaterialien erzeugen, ohne eine Schreibberechtigung für die entsprechenden Materialien besitzen zu müssen.
- *Erzeugen von semantischen Karten – Einbeziehen von externen Materialien:* Semantische Karten werden durch räumlich angeordnete Verweisobjekte umgesetzt. Die kooperative Arrangierbarkeit ergibt sich durch den Einsatz eines Shared Whiteboard. Hierbei ist die Navigierbarkeit sicherzustellen, es ist zudem eine Rückmeldung über die Navigationsschritte der Lernenden auf der Karte zu visualisieren.
- *Erzeugen von Annotationen:* Annotationen können durch verschiedene angeheftete Objekte (oder auch Verknüpfungsobjekte) semantische Beziehungen zwischen verschiedenen Materialien ausdrücken (vgl. Abbildung 3-7).

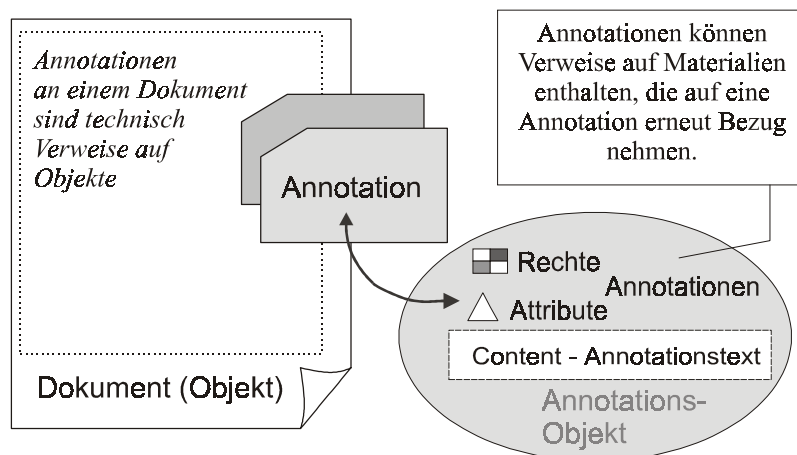


Abbildung 3-7: Annotationen sind Verweise auf Objekte

3.2.5 Annotieren

Eng verbunden mit der primären Medienfunktion des Verknüpfens ist das stark genutzte Konzept der Annotation.

Auf das dargestellte Rahmenkonzept bezogen werden Annotationen ähnlich einem Verknüpfungsobjekt als eigenständiges Objekt realisiert. Sie besitzen Attribute und Zugriffsrechte und können dementsprechend mit spezifischen Funktionalitäten ausgestattet werden. Das Annotieren eines Objekts ist als eigenes Recht abgebildet; es spezifiziert für einen Nutzer oder eine Gruppe die Berechtigung, Annotationen an das zugehörige Objekt anfügen zu dürfen.

Technisch werden Annotationen in Form einer Liste von Referenzen an Materialien auf Annotationsobjekte gespeichert. Das Annotationsobjekt selbst besitzt eine Referenz auf das annotierte Objekt (vgl. Abbildung 3-7).

Durch den konzeptionellen Ansatz der Realisierung von Annotationen als eigenständige Verweisobjekte kann der Inhalt der Annotationskomponente aus einer Referenz auf einen beliebigen anderen Objekttyp bestehen oder auch direkt verschiedene Inhaltskomponenten enthalten.¹⁵⁴ Damit sind Audio-Annotationen oder grafische Annotationen umsetzbar.

Ähnlich zu Verweisobjekten können Annotationsobjekte sich nicht nur auf Materialien beziehen, die sich innerhalb des kooperationsunterstützenden Systems befinden, sondern auch auf externe Webseiten. Dieser Fall ist mit einer Reihe von konzeptionellen Randbedingungen versehen, die zu einem großen Teil aus Einschränkungen der WWW-Technologie erwachsen.¹⁵⁵

Zusammenfassend besitzen Annotationen folgende konzeptionelle Eigenschaften:

- *Inhaltskomponente*: Ein Annotationsobjekt weist eine Inhaltskomponente auf (in den meisten Fällen einen Text) und eine Referenz auf das annotierte Objekt.
- *Medienfunktionen*: Das Annotationsobjekt stellt wiederum primäre Medienfunktionen bereit, d.h. Annotationen können arrangiert und verknüpft werden.
- *Attribute/Metainformationen*: Annotationen besitzen wie jedes Objekt Attribute. Diese können Metainformationen beinhalten. Über einen Mechanismus der Spezifikation von Schlüsselworten zu einer Annotation kann beispielsweise eine nachträgliche Verschlagwortung zu einem Objekt hinzugefügt werden, für das selbst keine Schreibberechtigung existiert.
- *Zugriffsrechte*: Annotationsobjekte besitzen Zugriffsrechte, d.h. Annotationen können in ihrer Sichtbarkeit oder in der Fähigkeit sie entfernen zu können auf eine Gruppe von Nutzern eingeschränkt werden.
- *Benachrichtigungsmechanismen*: Annotationsobjekte sind durch einen Benachrichtigungsmechanismus an das annotierte Objekt gebunden. Werden Operationen auf das annotierte Objekt ausgeführt, welche das Annotationsobjekt elementar beeinflussen, kann dieses entsprechend benachrichtigt werden. In diesem Zusammenhang sind verschiedene Strategien im Falle der Ungültigkeit von Annotationsobjekten denkbar. Beispielsweise können bei Löschen eines annotierten Objekts Annotationen ebenfalls gelöscht oder als ungültig gekennzeichnet werden.
- *Gegenseitige Bezüge*: Annotationen innerhalb eines Objektes können sich aufeinander beziehen oder linear an das Objekt angefügt sein. Es ergibt sich eine einfache Baumstruktur aus Diskussionsbeiträgen einzelner Nutzer. Diese kann beispielsweise durch eine interaktiv „aufklappbare“ Liste visualisiert sein, die zu jedem Eintrag Titel und Autor enthält. Ziel ist es, ein beliebiges Objekt zu einer Keimzelle eines Diskussionsverlaufs werden zu lassen. Der resultierende Diskussionsverlauf besitzt den Charakter einer News-Gruppe.

Neben den primären individuellen Medienfunktionen sind speziell primäre kooperative Medienfunktionen in das Objektmodell kooperativer Wissensräume einzubetten. Me-

¹⁵⁴ In der derzeitigen Realisierung erster Prototypen ermöglichen Annotationsobjekte die Verknüpfung eines kurzen Textes mit einem beliebigen Objekt, d.h. Annotationsobjekte enthalten zusätzlich zu einer Referenz auf das annotierte Objekt einen Text als Inhaltskomponente.

¹⁵⁵ Eine weiterführende Untersuchung von verschiedenen Annotationstechniken in Bezug auf die Einbeziehung von beliebigen Webseiten wurde in [Tappe 2000] vorgenommen.

dienfunktionen des Zugreifens, Übertragens und Synchronisierens von Materialien verlangen spezifische konzeptuelle Anforderungen in Bezug auf kooperative Interaktionen zwischen Nutzern. Die konzeptuellen Grundlagen zu ihrer Verwirklichung wurden schon durch das gezeigte Objektmodell gelegt. Es schafft die Basis, um Materialien auch in kooperativen Lernprozessen geeignet zu repräsentieren.

3.2.6 Zugreifen, Übertragen und Synchronisieren

Lernen ist ein verteilter, kooperativer Prozess. Wie schon in Abschnitt 1.4 gezeigt, werden Materialien an verschiedenen Orten zu verschiedenen Zeitpunkten abgelegt, abgerufen und bearbeitet. Gleichzeitig werden Materialien zwischen Lernenden ausgetauscht und in neue Lernkontexte gebracht. Kooperative Wissensräume müssen mit dem Anspruch der Bereitstellung primärer Medienfunktionen das Zugreifen auf Materialien in verschiedenen Konstellationen unterstützen. Ziel ist es, auftretende Medienbrüche weitgehend zu minimieren, d.h. eine durchgängige Verfügbarkeit der Lehrmaterialien zu erzielen.

Die zu berücksichtigenden Dimensionen der Zugreifbarkeit und des Übertragens von Lehrmaterialien sind zum einen der Ort der Materialien in Bezug auf den Lernenden und zum anderen das Verhältnis der Anzahl aus Sender und Rezipienten. Letzteres unterscheidet beispielsweise zwischen der Übertragung eines Objektes von einem zu mehreren Nutzern oder der Übertragung von genau einem Lernenden zu einem anderen Lernenden.

Im Folgenden werden verschiedene konkrete Formen der Medienfunktionen des Zugreifens, Übertragens und Synchronisierens innerhalb kooperativer Wissensräume diskutiert und an den Eigenschaften des gewählten Objektmodells deutlich gemacht.

Aktualisieren von Materialien (eines Objekts) durch einen neuen Inhalt (Zugreifen)

Kooperative Prozesse im Umgang mit Lehrmaterialien verlangen Tätigkeiten wie Aktualisieren, Modifizieren und Austauschen von Teilen eines Dokumentes. Hierzu dienen meist externe Werkzeuge, die nicht in den kooperativen Wissensraum integriert sind. Asynchrones Arbeiten bedeutet aus diesem Grunde, Dokumente aus der Umgebung zu laden, in eine Anwendungsapplikation (z.B. ein Office-Produkt) zu übernehmen und anschließend wieder in die kooperative Lernumgebung einzufügen.¹⁵⁶

Ein Aktualisieren und Austauschen des Inhalts eines Dokumentes bedeutet, ein existierendes Objekt mit bestehenden Attributen und Benutzerrechten mit einer neuen Inhaltskomponente zu versehen, diese zu aktualisieren oder eine weitere Inhaltskomponente bei weiterhin möglichem Zugriff auf ältere Versionen hinzuzufügen (Versionsmanagement). Der Prozess der Aktualisierung einer Inhaltskomponente eines existierenden kooperativen Objektes erfordert die Beachtung einer Reihe von Punkten:

- *Rechte*: Ein Austausch von Inhalten existierender Materialien wird über die Rechte der bestehenden Objekte kontrolliert.
- *Awareness*: Der Austausch einer existierenden durch eine neue Inhaltskomponente muss für andere Lernende deutlich erkennbar visualisiert werden (Awareness). Dies bezieht sich auch auf Verweisobjekte, die eine Referenz zu dem entsprechenden Objekt bilden.

¹⁵⁶ Eine ganze Reihe von existierenden Systemen forcieren die Entwicklung von in die Lernumgebung integrierten Werkzeugen (vgl. [Wessner et al. 1998]). Die vorgeschlagene Konzeption sieht sich als möglichst schlanke Lernumgebung; sie erhebt den Anspruch, durch standardisierte Schnittstellen möglichst integrativ zu bestehenden Werkzeugen zu wirken – die Entwicklung von proprietären Werkzeugen innerhalb des Systems widerspricht diesem Ansatz.

- *Annotationen:* Im Falle der Aktualisierung einer Inhaltskomponente eines Objekts sind für anhaftende Annotationen verschiedene Strategien der Aktualisierung denkbar. Diese können vollständig entfernt (speziell wenn es sich um Annotationen innerhalb von Texten handelt), an einem definierten Ort gesammelt oder als nicht aktuell gekennzeichnet werden. Verschiedene Strategien der Handhabung von Annotationen werden primär von den Zugangswerkzeugen bzw. der Webschnittstelle implementiert.

Verfügbarmachen und Präsentation von Materialien aus dem virtuellen Raum (Synchronisieren, Übertragen)

Das Laden bzw. Anzeigen von Objekten ist eine implizite Funktion des Zugreifens. Die Darstellung der Inhaltskomponente eines Objekts wird durch das Zugangswerkzeug (meist den Browser) vorgenommen.¹⁵⁷ Eine Visualisierung von Attributen oder verschiedenen Awareness-Informationen wird ebenfalls in Abhängigkeit des dargestellten Kontextes von den Zugangswerkzeugen realisiert. Innerhalb der entwickelten Szenarien sind Situationen des gemeinsamen Betrachtens von Lehrmaterialien oder Präsentationen unerlässlicher Bestandteil.¹⁵⁸ Diese werden durch spezielle Shared Browser (gekoppelte Darstellung der Sichten einer Gruppe von Lernenden) oder durch manuelles Synchronisieren herkömmlicher Browser umgesetzt.¹⁵⁹

Ein Shared Whiteboard kann in diesem Kontext als ein spezielles Instrument des Synchronisierens von Sichten auf Materialien für eine Anzahl von Lernenden verstanden und aus diesem Grunde speziell zur Umsetzung von Szenarien der Präsentation genutzt werden. Ähnlich kann ein Shared Browser der Medienfunktion des Synchronisierens zugeordnet werden. Innerhalb eines sTeam-Raumes abgelegte Verweise werden beispielsweise über gekoppelte Zugangswerkzeuge präsentiert.

Aus konzeptueller Sicht entspricht das Synchronisieren von Objekten dem Schaffen gekoppelter (synchronisierter) Sichten auf diese. Änderungen an dem Inhalt oder der Struktur der Objekte werden über Ereignisse (Events) in den verschiedenen Sichten aktualisiert.

Übertragen von Materialien zwischen Lernenden (Übertragen eines Objekts)

Der Austausch von Materialien zwischen Lernenden oder Dozenten und Lernenden erfolgt über den virtuellen Wissensraum. Es lassen sich zwei grundsätzliche Vorgehensweisen des Übertragens von Materialien zwischen Nutzern identifizieren:

Zum einen kann ein Objekt einem Lernenden oder einer Gruppe von Lernenden zur Ansicht oder Bearbeitung übertragen werden, in diesem Fall bleiben die Rechte bei dem Besitzer des Dokuments. Zum anderen können Dokumente vollständig einem anderen Lernenden übereignet werden, in diesem Fall geht das Objekt in den Besitz des Rezipienten über.

Die erste Form ist in seiner Ausprägung sicherlich die geläufigste Form.¹⁶⁰ Ein Dokument wird in einem virtuellen Raum abgelegt, Zugriffsrechte werden entsprechend der angestrebten Kooperationsform gesetzt. Die Mitlernenden erhalten Lese- bzw. Schreibberechtigung. Eine zugesprochene Schreibberechtigung erlaubt das Kopieren des entsprechenden Dokuments.

¹⁵⁷ Eine Darstellung erfolgt über geeignete Komponenten innerhalb des Browsers, so genannte Plugins.

¹⁵⁸ Vgl. hierzu die Szenarien zur Vorlesung in Kapitel 2.

¹⁵⁹ Lernende sprechen sich über die jeweils diskutierten Materialien ab und synchronisieren ihre Sichten auf die Lehrmaterialien manuell.

¹⁶⁰ Sie ist eng verbunden mit der Medienfunktion des Zugreifens (Sichtweise des Rezipienten).

Im zweiten Fall wird ein Objekt explizit in die Verwaltung eines anderen Lernenden oder einer Gruppe von Lernenden gelegt. Es wird das Recht zur Weitergabe von Berechtigungen gesetzt, d.h. das Dokument wird vollständig an den Rezipienten übereignet. Konzeptionelle Basis zu diesem Vorgehen ist der Mechanismus zur Weitergabe von Berechtigungen (vgl. Abschnitt 3.1.5).

Der schon beschriebene Mechanismus der Delegation von Zugriffsrechten bietet die Möglichkeit, Objekte ohne explizit gesetzte Zugriffsliste auszustatten und die Rechte von einem Elternobjekt (Delegationsobjekt) zu erben. Hierbei kann unterschieden werden zwischen einem festen Elternobjekt oder einer Vererbung in Abhängigkeit der Umgebung des Objekts. Letzteres entspricht einem Übertragen von Berechtigungen durch Umbewegen oder Verschieben des Dokumentes.

Zugangswerkzeuge (Client, Webschnittstelle) kleiden die beiden grundsätzlichen Mechanismen der Weitergabe von Dokumenten in verschiedene Metaphern. Diese orientieren sich insbesondere im Zuspreehen einer Konstellation aus Zugriffsrechten und der Weitergabe von Berechtigungen an verschiedenen Bildern wie Rollen, Briefkästen, Rucksäcken etc.

Beispiele geeigneter Metaphern zur Medienfunktionen des Übertragens sind:

- Verschieben eines Dokumentes in den Rucksack eines anderen Nutzers. Dies entspricht einem Übertragen der Verantwortung (Weitergabe von Berechtigungen) für das entsprechende Objekt.
- Verschieben eines Dokumentes in einen speziellen Abgabebriefkasten für Übungsblätter. Dies entspricht einer Übereignung des Dokumentes an den Übungsgruppenleiter. Die Zugriffsrechte werden vollständig an den Übungsgruppenleiter übertragen (vgl. Abbildung 2-2).
- Versenden eines Dokumentes in den Rucksack eines Nutzers. Der Rucksack eines Nutzers kann ein Postfach für den Empfang von Nachrichten (z.B. E-Mail) und angefügten Dokumenten enthalten. Über einen derartigen Mechanismus kann beispielsweise ebenfalls die Abgabe von Übungslösungen an einen Betreuer erfolgen. Analog können im umgekehrten Fall Nachrichten über ein herkömmliches E-Mail-Programm aus dem Postkasten abgerufen werden. Letzteres ist bereits in den derzeitigen sTeam-Prototypen umgesetzt.¹⁶¹

Synchronisieren, gemeinsame Präsentation, gemeinsames Browsen

In der Literatur und in der praktischen Anwendung finden sich eine ganze Reihe von Applikationen, die gezielt das kooperative Browsen und die netzgestützte Präsentation ermöglichen.¹⁶² Weiterhin existieren verschiedene Techniken der Kopplung von Sichten auf gemeinsam genutzte Applikationen (Application Sharing). Defizitär erweist sich in den meisten Ansätzen ihr Charakter als unverbundene einzelne Werkzeuge – es fehlt eine verbindende konzeptuelle Basis.

Die Notwendigkeit und Leistungsfähigkeit eines Konzepts kooperativer gemeinsamer Objekte wird insbesondere in den Bereichen des gemeinsamen Betrachtens von Netzseiten (Shared Browsen) und der synchronen Präsentation deutlich: In der Konzeption kooperativer Wissensräume finden sich zwei grundlegende Elemente, die eine

¹⁶¹ Der sTeam-Server implementiert einen Mailserver, der den Abruf von E-Mail-Nachrichten aus den Postfächern der einzelnen Nutzer gestattet.

¹⁶² Vgl. beispielsweise [Cabri et al. 1999] und [Marais & Bharat 1997].

Kopplung der Sichten auf Objekten und damit eine Fokussierung von Aufmerksamkeit zulassen:

- *Gemeinsames Browsen*: Die Sichten verschiedener Lernender auf Elemente innerhalb oder außerhalb der Lernumgebung sind koppelbar. Eine Synchronizität der Darstellungen ergibt sich durch Aktionen der einzelnen Nutzer. Wird beispielsweise in einem Browser ein Navigationsschritt durchgeführt, werden die Darstellungen der gekoppelten Browser aktualisiert. Die Kopplung wird durch einen ausgezeichneten Browser vorgenommen oder sämtliche Browser sind gleichberechtigt.¹⁶³
- *Präsentation von Materialien (z.B. Vorlesungsinhalten)*: Eine Präsentation von Materialien lässt sich zum einen über den Mechanismus des Shared Browsers realisieren, zum anderen eignet sich ein Shared Whiteboard zur synchronen Präsentation. Im letzteren Fall erlauben gemeinsame Mauszeiger (Telepointer) oder grafische Elemente eine direkte, synchrone Kopplung der Sichten auf gemeinsame Objekte.

Interaktion zwischen Lernenden und aktiven Elementen (Übertragen)

Interaktion findet zwischen Personen (Kommunikation, Weitergabe von Objekten, gegenseitige Präsentation), aber auch zwischen Personen und aktiven Elementen innerhalb eines Raumes statt.¹⁶⁴

Es lassen sich eine ganze Reihe von Elementen aus den skizzierten Szenarien ableiten, die ein aktives Verhalten im Sinne einer Kommunikation und Interaktion zwischen Lernenden aufweisen. Hierbei handelt es sich in den meisten Fällen um eine Kette des Übertragens von Materialien zwischen Nutzern über ein aktives Objekt, d.h. ein Dokument wird in einen elektronischen Briefkasten o.ä. gelegt und damit einer zweiten Person indirekt übergeben. Aus diesem Grunde erscheint eine Zuordnung zu der Medienfunktion des Übertragens sinnvoll. Aktive Elemente innerhalb kooperativer Wissensräume werden durch ein herkömmliches Objekt realisiert, welches eine Reihe von speziellen Eigenschaften aufweist:

Es besitzt ein Verhalten, welches auf eintretende Ereignisse reagiert. Ereignisse sind beispielsweise das Erscheinen eines Benutzers in einem Raum, das Eintreffen eines Dokumentes in der Umgebung oder ein Chat-Beitrag innerhalb des Raumes (vgl. Abschnitt 5.2.5).

Aktive Objekte sind persistent und besitzen Attribute und Rechte, die eine Benutzung der Objekte kontrollieren. Sie können von beliebigen (berechtigten) Benutzern innerhalb von Räumen erzeugt und abgelegt werden.¹⁶⁵ Der übliche Weg ist sicherlich das Kopieren und Anpassen eines vorhandenen aktiven Objekts durch einen Raum- bzw. Gruppenverwalter. Es können jedoch ebenfalls vollständig neue aktive Objekttypen eingebracht werden.

Es lassen sich eine ganze Reihe von möglichen Objekttypen und Anwendungsfeldern ausmachen. Exemplarisch genannt seien beispielsweise:

¹⁶³ Abschnitt 5.3.4 diskutiert einige architektonische Realisierungsformen synchronisierter Browser und Clients. Die Kernkonzepte hierzu sind in der sTeam-Architektur vorgesehen.

¹⁶⁴ Schon Curtis schildert in seinen Erfahrungen zu dem am Xerox Parc installierten LambdaMOO-System erste Ansätze derartiger Interaktionsformen (vgl. [Curtis 1992]).

¹⁶⁵ Es handelt sich in diesem Sinne nicht um eine systemseitige Erweiterung des Servers, sondern um die benutzerseitige Ergänzung von virtuellen Räumen durch spezifische Funktionalitäten.

- *Abgabe-Briefkasten*: Ein Briefkasten reagiert auf „eingegangene“ Dokumente, protokolliert das Eintreffen und erlaubt Abgabefristen zu definieren. – Der Briefkasten „verschließt“ sich nach Ablauf der Abgabefrist. Optional lassen sich E-Mails zur Bestätigung des Empfangs des Dokumentes versenden (vgl. Abbildung 2-2).
- *Elektronische Wahlurne*: Ein aktives Element in Form einer Wahlurne protokolliert abgegebene Stimmen zu einer zu treffenden Entscheidung. Jeder Lernende hat genau eine Stimme, der Wahlleiter vermag den Abstimmungsvorgang zu initiieren und zu beenden. Das Ergebnis der Abstimmung wird automatisch aus den abgegebenen Stimmen ermittelt.
- *Konversationsrekorder*: Ein Rekorder protokolliert Nachrichten (Abschnitte einer Diskussion) innerhalb eines Raumes. Er kann definiert ein- und ausgeschaltet werden und vermag Abschnitte einer Diskussion wiederzugeben.

Sämtliche der oben genannten aktiven Elemente kooperativer Wissensräume basieren auf aktiven Objekten. In den meisten Fällen genügt ein herkömmliches Container-Objekt (vgl. Abschnitt 5.2.1), welches auf verschiedene Ereignisse reagiert und entsprechend neue Ereignisse auslöst. Statusinformationen etc. werden in speziellen Attributen gespeichert. Beispielsweise reagieren der elektronische Abgabe-Briefkasten oder die elektronische Wahlurne auf das Ereignis eines eintreffenden Objekts (Dokumentes) und lösen entsprechende Aktionen aus. Attribute verwalten Abgabefristen oder das aktuelle Wahlergebnis. Im Falle eines Konversationsrekorders reagiert ein aktives Objekt auf die Chat-Kommandos innerhalb des Raums. Über ein Attribut wird der Protokollmodus ein- und ausgeschaltet, die Diskussion wird ebenfalls in einem Attribut protokolliert.

Neben der gegenseitigen Wahrnehmung und der Medienfunktion des Synchronisierens (vgl. Abschnitt 2.3.3) ist Kommunikation das zweite Instrument der Koordination zwischen Lernenden in der Nutzung kooperativer Medienfunktionen. Lernende kommunizieren mit anderen Lernenden über relevante Lehrmaterialien. Handlungen an gemeinsam genutzten Materialien werden durch verschiedene Formen der Kommunikation zwischen Lernenden koordiniert. Abstimmungs- und Koordinationsprozesse sind damit wesentlicher Teil der Kommunikation und zunächst Bestandteil kooperativer primärer Medienfunktionen. Gleichzeitig gilt es jedoch zu beachten, dass Mechanismen zur Strukturierung und Koordination von Kommunikationsprozessen, speziell wenn es sich um thematische Strukturierungen oder vorgegebene Abläufe handelt, den sekundären Medienfunktionen zuzuordnen sind.

3.3 Kommunikation

Computergestützte Kommunikation dient maßgeblich der Koordination von Aktivitäten innerhalb virtueller Wissensräume und ist damit zugleich Instrument der gegenseitigen Wahrnehmung. Kommunikationsunterstützung ist in dieser Form ein zentraler Baustein virtueller Lerngemeinschaften.¹⁶⁶

¹⁶⁶ Historisch betrachtet ist die Bereitstellung von netzgestützten Kommunikationsmechanismen die zentrale Motivation früherer MUD- und MOO-Systeme. Weitere Beispiele aus der Gründerzeit des Internet sind synchrone, kanalorientierte Systeme wie der Internet Relay Chat (IRC) oder asynchrone, nachrichtenorientierte Systeme wie NetNews. Avatarwelten und WWW-gestützte Chat-Umgebungen können als moderne Ausdruckformen dieser klassischen Mediendienste bezeichnet werden.

Die folgenden Abschnitte beleuchten die wesentlichen Komponenten und Formen einer computermedierten Kommunikation und stellen die entwickelten Mechanismen der Einbettung von Kommunikationsmechanismen in virtuelle Wissensräume vor. Hierzu zählen asynchrone und synchrone Kommunikationsformen, Strukturierungsformen von Diskussionsbeiträgen, Gruppierungsmechanismen von Kommunikationspartnern und Zugangswerkzeuge zur Kommunikationsunterstützung.

Eine ganze Reihe psychologisch-kognitiver Ansätze unterteilen Kommunikation in face-to-face Kommunikation („Angesicht-zu-Angesicht“-Kommunikation), synchrone computermedierte Kommunikation und asynchrone computermedierte Kommunikation. Eine derartige Unterscheidung von face-to-face-Kommunikation im Gegensatz zu computergestützter Kommunikation sei an dieser Stelle nicht weiter untersucht. Dies resultiert aus dem Anspruch, kooperative Lernprozesse sowohl in Präsenzsituationen durch Computereinsatz unterstützen zu wollen und gleichermaßen synchrone und asynchrone Kommunikationsformen zu erlauben.

Das ausgehend von dem innerhalb einer „Person-zu-Person“-Kommunikation (Chat) zunächst triviale Problem der gegenseitigen Wahrnehmung entwickelt sich schon bei einer kanalorientierten Kommunikation mit einer Vielzahl von Teilnehmern zu einer komplexen Problemstellung.¹⁶⁷ Wahrnehmungskomponenten erlangen insbesondere in einem synchronen Umfeld einen hohen Grad an Bedeutung. Zu visualisieren sind z.B. Aspekte wie:

- die Anwesenheit von Kommunikationspartnern – hierbei ist eine Unterscheidung nach physischer Anwesenheit, also einer Ansprechbarkeit, und einer Anwesenheit in dem Kommunikationsmedium zu treffen.¹⁶⁸
- Kommunikationstätigkeit – die Erkennbarkeit bzw. Sichtbarkeit des Autors eines Diskussionsbeitrages im Diskussionsverlauf.

Eine erste Klassifizierung verschiedener Kommunikationsformen gelingt entlang der Kriterien Synchronizität (synchron – asynchron), Persistenz (flüchtig – persistent) und der Orientierung (kanalorientiert – objektbezogen).

3.3.1 Synchron – asynchron, kanalorientiert – objektbezogen

Basierend auf der obigen Unterscheidung ist die Synchronizität von Kommunikation im virtuellen Raum ein bestimmendes Element. Synchrone Kommunikation verlangt im Gegensatz zu asynchronen Kommunikationsformen die Anwesenheit der Kommunikationspartner. Zudem ist synchrone Kommunikation meist flüchtig, d.h. es können nur wenige Abschnitte einer Kommunikation zurückverfolgt werden.¹⁶⁹

Findet Kommunikation synchron statt, kann dies natürlichsprachig umgesetzt sein (Audio-/Video-Konferenz) oder rein textuell (Chat). Die erste Klasse von Systemen wird in der Literatur zumeist als Media Spaces bezeichnet.¹⁷⁰

¹⁶⁷ Insbesondere aus dem Bereich der Avatarwelten konnten in den letzten Jahren eine Reihe von soziologischen und psychologischen Erkenntnissen über die Verhaltensweisen in der Interaktion und der Kommunikation in virtuellen Welten gewonnen werden (vgl. [Turkle 1995]).

¹⁶⁸ Schon in MUDs und MOOs werden Nutzer über die Anwesenheit, insbesondere das Eintreffen neuer Spieler in der Umgebung und das Verlassen von Nutzern aus dem virtuellen Raum informiert.

¹⁶⁹ Innerhalb eines üblichen Chat-Systems lässt sich nur ein gewisser Abschnitt, beispielsweise nur einige Bildschirmseiten, zurückverfolgen.

¹⁷⁰ Schon Curtis sieht die Möglichkeit, eine Audio-Kommunikation in das Konzept des MUD-Raumes zu integrieren; d.h. die Teilnehmer eines Raumes nehmen an einem Audio-Chat teil, vgl. das Jupiter-System [Curtis et al. 1995].

Asynchrone Kommunikationsformen besitzen keine unmittelbare zeitliche Abfolge von Redebeiträgen, obwohl diese auch zeitnah erfolgen können. Hauptmerkmal ist die Persistenz der Beiträge. Diese Dauerhaftigkeit bedingt eine spezielle Metapher im Umgang mit asynchronen Kommunikationsformen. Sie sind weniger kanalorientiert, d.h. nicht flüchtig und orientieren sich zumeist an mehreren Themen und Materialien.¹⁷¹

Unter dem Blickwinkel kooperativer Wissensräume, welche sowohl synchrone als auch asynchrone Kommunikationsmechanismen beinhalten, ist weniger die zeitliche Abfolge von Kommunikationsbeiträgen zu untersuchen, vielmehr sind die Möglichkeiten ihrer Strukturierung begleitend einem kooperativen Lernprozess von hohem Interesse.

3.3.2 Problembereiche der Strukturierung von Kommunikation

Innerhalb virtueller Welten erstreckt sich die Kommunikation der anwesenden Nutzer zunächst auf den Raum. Eine Kontrolle des Zugangs zu einem Raum schränkt Kommunikation auf eine Gruppe von Nutzern ein – der Raum wirkt als natürliches *Begrenzungsinstrument* von Kommunikation.¹⁷² Gleichzeitig wirkt der Raum als *Strukturierungshilfe* einer Kommunikation, indem er thematisch orientiert ist und durch die Anwesenheit eines bestimmten Nutzerkreises eine gewisse soziale Anziehungskraft ausübt.

Nur wenige Audio-Chat-Ansätze beziehen sich in ihrer Konzeption auf eine Raummetapher oder eine ihr gleichwertigen Metapher.¹⁷³

Nicht nur der Raum an sich, sondern auch in ihm enthaltene virtuelle Gegenstände, wie Tische oder spezielle Bereiche können Kommunikation strukturieren. So kann beispielsweise ein virtueller Tisch metaphorisch eine Anzahl von Nutzern zu einer Diskussionsgruppe zusammenfassen.¹⁷⁴ Leider orientieren sich derartige Mechanismen der Kommunikationsunterstützung zumeist an grafischen, nur für kleine Diskussionsgruppen geeigneten Metaphern.

Schwierigkeiten der fehlenden Struktur bei existierenden Kommunikationsmodellen wie beispielsweise das Zuordnen von Redebeiträgen in einer Gruppe eines gemeinsamen Audio-Chat-Kanals treten ebenfalls im Umfeld textueller Chats auf und werden von einer Reihe von Autoren identifiziert¹⁷⁵.

Synchrone computergestützte Kommunikationsmechanismen sind mit einer Reihe weiterer Schwierigkeiten verbunden. So ist ein synchroner Chat zunächst nur für die innerhalb des Kanals oder eines Raumes anwesenden Personen sichtbar und besitzt aufgrund meist fehlender Persistenz nur wenig Verbindlichkeit. Wird die räumliche Position eines Avatars an die Möglichkeit zur Teilnahme an einem Chat geknüpft, entstehen ähnliche Einschränkungen. Die zwingend erforderliche Anwesenheit von Diskussions-

¹⁷¹ Der entsprechende klassische Internetdienst ist das nachrichtenorientierte NetNews-System, welches sich über themenorientierte Gruppen, den News-Gruppen definiert. Eine Reihe von Systemen greifen die News-Metapher auf und erweitern sie um spezielle Funktionalitäten. Beispiele sind BSCW [Appelt et al. 1998] und HyperNews (vgl. [Hunt 1996]).

¹⁷² Von der Notwendigkeit privater Diskussionsräume berichtet schon Curtis (vgl. [Curtis 1992, S. 10ff.]).

¹⁷³ Beispiele sind die MASSIVE-Umgebung [Greenhalgh & Benford 1995] oder das Mediazine-System [Schiffner & Chodura 2000].

¹⁷⁴ Vgl. [Sohlenkamp & Chwelos 1994]. Benutzer, die sich an einem gemeinsamen virtuellen Schreibtisch befinden, können durch Verschieben ihres Piktogramms (Icons) auf einen Kommunikationspartner eine private Kommunikation initiieren. Eine private Unterhaltung ist für andere Teilnehmer der Umgebung nur in reduzierter Lautstärke zu hören.

¹⁷⁵ Guernsey merkt beispielsweise an, dass das Verfolgen mehrerer Linien einer Diskussion innerhalb eines Chats erhebliche Probleme aufwirft (vgl. [Guernsey 1996]).

partnern einer virtuellen Umgebung in einem Raum führt beispielsweise dazu, dass sich Diskussionen nicht über eine Anzahl von Räumen hinweg führen lassen. Existieren z.B. mehrere Lernräume zu einem Themenbereich, kann es keine parallele globale Diskussion zu diesem Thema geben, ohne dass die Lernenden ihre einzelnen Lerngruppenräume verlassen.

3.3.3 Implikationen für die gewonnene Konzeption – synchron

Alle Formen synchroner Kommunikation sind in ihrer Reichweite zunächst durch die Umschließung des virtuellen Raums begrenzt. Teilnehmer von Chat und Audiokommunikation sind die anwesenden Lernenden des jeweiligen virtuellen Raums. Im Falle einer Audio-Kommunikation ist die Zusammenführung aller Audio-Quellen eines Raums die nahe liegende Vorgehensweise, d.h. jeder Lernende kann das Gesprochene der anderen Teilnehmer im Raum hören.

Die Kopplung aus Anwesenheit in einem virtuellen Raum und Teilnahme an einer Diskussion ist mit einer Reihe von Einschränkungen verbunden (z.B. fehlende Möglichkeit des Verfolgens mehrerer in verschiedenen Räumen stattfindender Diskussionsstränge). Denkbare Lösungsansätze hierzu sind die Möglichkeit zur passiven Teilnahme an weiteren Chat-Sitzungen parallel zur aktiven Anwesenheit in einem Raum oder das Kanalkonzept. In beiden Fällen ermöglicht das schon beschriebene Kernkonzept kooperativer Wissensräume das Binden von Kommunikation an Materialien:

- *Raumbezogene Chat-Gruppen*: Raumbezogene Chat-Gruppen lassen eine Anzahl von beliebigen Diskussionsgruppen innerhalb virtueller Wissensräume zu. Diese können durch räumliche Nähe der Avatare innerhalb des virtuellen Raumes initiiert sein, d.h. ein Avatar nimmt an maximal einer Diskussionsgruppe teil, oder öffentlich sein, d.h. Lernende können parallel an mehreren Gruppen innerhalb ein und desselben Raumes teilnehmen. In beiden Fällen ergibt sich eine Unterstrukturierung für die innerhalb eines Raumes stattfindende synchrone Kommunikation.
- *Kanal-Objekt-Konzept*: Basierend auf der Metapher eines Kanals können Lernende eigene synchrone Diskussionskanäle an bestehende oder neu eingerichtete Objekte virtueller Wissensräume knüpfen. Ein Lernender kann sich durch seine Rechte an dem entsprechenden Objekt kontrolliert mit dem Kanal verbinden und so an der Diskussion teilnehmen. Kanäle können von Lernenden eigenständig an ein beliebiges Objekt geknüpft oder alternativ durch spezielle Kanalobjekte erzeugt werden. Die Anwesenheit in einem Kanal ist nicht an die Präsenz in dem entsprechenden Raum gebunden, damit ist die Teilnahme parallel an verschiedenen Diskussionssträngen möglich. Eine Integration von kanalorientierter Kommunikation in den virtuellen Raum kommt insbesondere der Nutzung externer Chat-Werkzeuge, z.B. einem IRC-Client im Zusammenhang mit dem sTeam-System, zugute (vgl. Abschnitt 5.2.8).¹⁷⁶
- *Multiple Avatare („Geister“)*: Die Kopplung aus Benutzer und Avatar lässt sich durch das Modell von weiteren Repräsentationen eines Nutzers (Avataren) auflösen. Hierbei ist denkbar, dass ein Benutzer eine Reihe von Avataren besitzt, die beispielsweise der Anzahl von Fenstern des Zugangswerkzeugs entsprechen. Um das damit einhergehende Problem der Einschränkung von Wahrnehmungs-

¹⁷⁶ Ein neu eingerichteter Kanal kann beispielsweise durch einen externen IRC-Client angesprochen werden und auf diese Weise ohne ein spezielles Zugangswerkzeug genutzt werden. Zurzeit fungiert jeder einzelne Raum gleichbedeutend einem für ein IRC-Client ansprechbaren Kanal.

informationen der realen Position eines Lernenden zu mindern, können zusätzliche Avatare „Geister“ in ihren Möglichkeiten herabgesetzt werden und ein bestimmtes Erscheinungsbild erhalten. Das Konzept der „Geister“ reduziert zusätzliche Avatare auf passives Verfolgen einer Diskussion ohne sich an ihrem Verlauf aktiv beteiligen zu können. „Geister“ sind als ein Kompromiss zu verstehen zwischen Beibehaltung des Konzepts der eindeutigen Awareness-Informationen über die Position eines Nutzers und der Möglichkeit, parallel eine Reihe von Diskussionen verfolgen zu können.

- *Spatial Audio – räumliche Zuordnung von Audio-Quellen*: In Abgrenzung zu existierenden Ansätzen sieht das Konzept virtueller Wissensräume die Integration von Audio-Kommunikationsmechanismen vor, untersucht jedoch keine speziellen Mechanismen der Aufbereitung der Audio-Signale zur räumlichen Zuordbarkeit der Kommunikationspartner innerhalb des Raums.¹⁷⁷
- *Spezifische Chat-Zugänge*: Durch eine Definition möglichst universeller technischer Schnittstellen zu den Kommunikationsprozessen kooperativer Wissensräume lassen sich auch existierende spezielle Chat-Clients integrieren.¹⁷⁸

Fehlende Persistenz computergestützter synchroner Kommunikationsmechanismen kann durch eine Reihe von Metaphern aufgelöst werden:

- *Raumrekorder*: Protokollwerkzeuge können in der Metapher eines Rekorders in virtuelle Räume eingebracht werden. Realisiert als eigenständiges aktives Objekt kann er zum gewünschten Zeitpunkt aktiviert werden und protokolliert die stattfindende, synchrone Kommunikation. Teile einer Kommunikation können anschließend wiedergegeben werden bzw. stehen in textueller Form zur Verfügung.
- *Protokoll-Avatare*: Avatare können mit einer Protokoll-Funktion ausgestattet werden, die es erlaubt, Teile einer Diskussion aufzuzeichnen und zu einem späteren Zeitpunkt wiederzugeben. Auf diese Weise lässt sich bei physischer Abwesenheit eines Nutzers der Verlauf einer Diskussion verfolgen.

3.3.4 Implikationen für die gewonnene Konzeption – asynchron

Wesentliches Defizit existierender Kommunikationsmechanismen ist die fehlende Verknüpfung von Materialien einer Lernumgebung und der Struktur von Kommunikationsmechanismen.

Eine These, welche die vorliegende Arbeit in diesem Zusammenhang verfolgt, lautet, dass sich Abstimmungs- und Lernprozesse innerhalb einer kooperativen Umgebung entscheidend an dem Maß an Strukturierungsvielfalt von Kommunikation in Bezug auf die Materialien des Lernens orientieren. Damit findet Kommunikation zwar in den natürlichen Schranken des virtuellen Raums statt, ist aber mehr und mehr thematisch an einzelne Objekte geknüpft und mit diesen Objekten arrangierbar.

Ohne im Folgenden gezielt eine Liste von Prioritäten vorgeben zu wollen, lassen sich eine ganze Reihe von konzeptionellen Merkmalen identifizieren, die eine Verbesserung der Verknüpfung aus Materialien und Kommunikationsmechanismen erzielen. Schlüsseltechnologie ist die Idee, jedes Objekt annotierbar und zur Keimzelle synchro-

¹⁷⁷ Technische Studien zum Einsatz von räumlichen Audioquellen finden sich zumeist im Zusammenhang mit der Entwicklung von Collaborative Virtual Environments (CVEs) [Greenhalgh & Benford 1995]) und dem Bereich des „spatial audio“ (vgl. [Yamazaki & Herder 2000]).

¹⁷⁸ Vgl. die Entwicklung eines speziellen Chat-Clients für sehbehinderte Nutzer [Hampel et al. 1999a].

ner Kommunikation werden zu lassen. Abstrakt ausgedrückt ist das Ziel die Verschmelzung von Annotationskonzept, Mechanismen der asynchronen und synchronen Kommunikationsformen auf Basis der Herstellung von Objektbezügen.

Annotationskonzepte aus dem Bereich der Forschung zum Umfeld Hypertext beschäftigen sich primär mit der Fragestellung der Integration von Anmerkungen in das Medium selbst. Annotationen besitzen eine feste Position innerhalb eines Textes oder einer Grafik und werden durch Markierungen o.ä. visualisiert.

Das Kernkonzept der vorliegenden Arbeit reduziert das Problem der Annotation zunächst auf die Strukturierung von Annotationen *an dem* Objekt. Natürlich können von Seiten der technischen Umsetzung Kommentare und Anmerkungen sowohl an innerhalb der Umgebung vorliegenden Textmaterialien erzeugt werden als auch externen Hypertexten, wenn auch mit gewissen Einschränkungen.¹⁷⁹ Eine Annotation an dem Objekt erlaubt gegenseitige Bezüge von Annotationen aufeinander und damit die Entwicklung von Diskussionsverläufen an Materialien.

Asynchrone Kommunikation definiert sich damit primär über das Konzept des Verknüpfens von Materialien (Texte, Grafiken, Illustrationen) mit Diskussionsbeiträgen. Diese Idee kann in ihrer Grundstruktur erheblich weiter ausformuliert werden und spezifische Darstellungen der Beziehungen zwischen Materialien und Kommunikationsmechanismen entwickeln.¹⁸⁰

Das vorliegende Kapitel hat gezeigt: Die konzeptionelle Einbettung primärer individueller und primärer kooperativer Medienfunktionen in die Metapher kooperativer Wissensräume ist trotz einer Vielzahl auftretender Designkonflikte realisierbar. Basierend auf einem universellen Objektmodell konnten wesentliche Anteile kooperativer Lernprozesse in technische Unterstützungsfunktionen übertragen werden. Gleichzeitig wurde ein Modell aus Benutzergruppen und Rechten als Basis kontrollierter kooperativer Handlungen an Materialien formuliert.

In einem nächsten Schritt sind die gewonnenen, konzeptionellen Modelle in ein konkretes, *technisches* Konzept zu übertragen. Im Folgenden werden zunächst potenzielle Systeme und Frameworks zur Kooperationsunterstützung aus technischer Sicht untersucht. Damit findet sich eine Grundstruktur möglicher kooperationsunterstützender Architekturen. In Abgrenzung zu der in Kapitel 5 folgenden Architekturlösung kooperativer Wissensräume werden die relevanten existierenden Architekturansätze kooperationsunterstützender Systeme auf die technische Umsetzbarkeit der im vorliegenden Kapitel formulierten Designanforderungen hin untersucht.

¹⁷⁹ Vgl. hierzu die technische Analyse von Konzepten der Annotation von Fremdmaterialien [Tappe 2000] und [Hampel & Bopp 2001].

¹⁸⁰ In [Preiß 2000] wurde beispielsweise das Konzept des „strukturierten Chats“ entwickelt und erprobt. Das Erzeugen von Annotationen und sich aufeinander beziehenden Diskussionsbeiträgen an Objekten führt zu textuellen Einträgen in einer Baumdarstellung. Umgekehrt können Annotationen in der Baumdarstellung vorgenommen werden und erzeugen entsprechende Annotationen auf der grafischen Ebene.

4. Architekturkonzepte kooperativer Wissensräume

Die technische Einbettung und architektonische Umsetzung des Konzepts virtueller Wissensräume macht eine Reihe von Schwierigkeiten und Designkonflikten deutlich: Wie schon dargestellt, besitzt die Umsetzung von Anforderungen der Selbstadministration und kooperativen Medienfunktionen für die Lernenden ein hohes Potenzial, gleichzeitig aber verursacht die architektonische Fundierung des zugrunde liegenden kooperationsunterstützenden Systems erhebliche Probleme. Dies liegt zu einem großen Teil an oft gegensätzlichen Designanforderungen, z.B. der Bereitstellung offener und freier Kooperationsformen (eines flexiblen Benutzer- und Rollenkonzepts) bei gleichzeitiger Wahrung elementarer Grundsätze des Zugriffsschutzes.

Eine Umsetzung des Konzepts kooperativer Wissensräume stößt zudem auf eine Reihe von technischen-infrastrukturellen Problemen. Zunächst gilt es, verfügbare Standards und Entwicklungslinien zu berücksichtigen. Diese sind zwingende Voraussetzung für die Integrierbarkeit der angestrebten Lösung in die tägliche Arbeitsumgebung von Lernenden und Lehrenden. Gleichzeitig sind genau diese verbreiteten Standards (wie beispielsweise das HTTP-Protokoll) mit Einschränkungen verbunden, die eine Umsetzbarkeit der angestrebten Anforderungen erschweren. Die vorliegende Arbeit unternimmt hier eine Gratwanderung und verbindet technisch-konzeptuell eine Reihe grundsätzlicher Entwicklungslinien. Diese gilt es im vorliegenden Kapitel vorzustellen sowie erste Abgrenzungen zur in Kapitel 5 vorgenommenen Darstellung der Architektur zu treffen.

Im Folgenden wird zunächst über wesentliche Architekturmerkmale kooperativer Systeme informiert. Hierzu zählen verteilte und zentralisierte Architekturen bzw. eine Übertragung des Model View Controller (MVC) -Konzepts auf zentralisierte Applikationen. Gleichzeitig werden verschiedene technologisch relevante Ansätze zum Themenbereich der benutzerseitigen Programmier- und Erweiterbarkeit eines kooperationsunterstützenden Systems dargestellt.

Ein sich anschließender detaillierter Überblick über existierende technische Ansätze klassifiziert diese nach den Kriterien zentralisierter oder replizierter Architekturen, objektorientierter Frameworks, WWW-basierter Architekturkonzepte sowie MUD- und MOO-basierter Architekturen. Letztere bilden den wesentlichen architektonischen Kern der im Rahmen dieser Arbeit geschaffenen Lösung. Kapitel 5 schließt sich unmittelbar an die Analyse und Abgrenzung möglicher alternativer Architekturen an. Aufbauend auf den in Kapitel 3 formulierten Designanforderungen an kooperative Wissensräume und

der im Folgenden vorgenommenen Darstellung möglicher Basisarchitekturen zu ihrer Umsetzung wird der resultierende Architekturansatz vorgestellt.¹⁸¹

4.1 Technologische Grundkonzepte

In Betrachtung der technischen Konzeption zur architektonischen Umsetzung des angestrebten kooperationsunterstützenden Systems sind die Grundmerkmale zentralistisch, repliziert, asynchron und synchron von vorrangiger Bedeutung. Auch wenn es sich bei der angestrebten Lernumgebung um ein primär asynchrones System handelt, enthält es doch wesentliche Merkmale synchroner Systeme. Gleichzeitig werden sowohl Elemente zentralistischer Architekturen mit replizierten Konzepten verbunden.

In den folgenden Abschnitten werden drei wesentliche Merkmale kooperativer Applikationen diskutiert: Es handelt sich um die Abgrenzung zwischen zentralisierten und replizierten Architekturen, dem Merkmal der Transparenz gegenüber spezieller kooperationsunterstützender Technologie, d.h. inwieweit die Semantik einer Applikation mit den kooperationsunterstützenden Eigenschaften verbunden sein muss, sowie der Fragestellung nach der architektonischen Berücksichtigung benutzerseitiger Konfigurier- und Erweiterbarkeit.

4.1.1 Zentralisiert und repliziert

Eine erste und sicherlich nahe liegende Klassifizierung von Architekturen von kooperativen Applikationen oder ganzen kooperationsunterstützenden Umgebungen gelingt durch die Unterscheidung nach einer für einen bestimmten Einsatzzweck entwickelten Applikation, z.B. einer Lernumgebung, oder einer Rahmenarchitektur (Framework) für die Entwicklung von kooperativen Applikationen. Derartige Frameworks haben in der CSCW-Forschung eine lange Tradition und dienen in vielen Fällen primär der Entwicklung und Erprobung von technischen Konzepten und bereitgestellten Funktionalitäten.

Eine Klassifizierung *technischer* Grundkonzepte zur Entwicklung kooperationsunterstützender Systeme fällt schwer. Zum einen handelt es sich um Protokolle und Architekturen, die Kommunikation von Objekten oder Applikationen über Netzwerke ermöglichen, zum anderen um Mechanismen, die einen gemeinsamen Datenraum zwischen unterschiedlichen Applikationen bereitstellen.

Wesentliches Unterscheidungsmerkmal ist der architektonische Grundaufbau als *replizierte* oder *zentralisierte* Architektur. Hierbei ist die Architektur zur Laufzeit der Applikation von Bedeutung.

Replizierte Architekturen bestehen aus einer Anzahl prinzipiell identischer kooperativer Instanzen einer Applikation, die über das Netz eigenverantwortlich kommunizieren (vgl. Abbildung 4-3). Diese Kommunikation geschieht zwar über einen gemeinsamen Datenbus, ist jedoch nicht durch einen zentralen Knoten, also einen Server geprägt. Verschiedene Ansätze besitzen zwar einen ausgewiesenen Client, der spezifische Aufgaben wie die Verwaltung der einzelnen Sitzungen vornimmt, die Kommunikation der Clients ist jedoch meist nicht zwingend über diesen kontrolliert.

¹⁸¹ Lesern, die sich unmittelbar für diese technische Umsetzung des sTeam-Systems und weniger für verschiedene Architekturansätze kooperativer Wissensräume interessieren, wird an dieser Stelle der Abschnitt 4.2.5 oder auch der unmittelbare Übergang zu Kapitel 5 empfohlen.

Zentralisierte Architekturen (vgl. Abbildung 4-2) sind durch einen zentralen Prozess bzw. ein zentrales Programm geprägt, welches untergeordnete Clients mit Informationen versorgt.¹⁸² Im Falle von Application-Sharing-Anwendungen werden sogar ganze Bildschirmausgaben übermittelt. Von Seiten der Grundstruktur entsprechen zentralisierte Architekturen den klassischen Client-Server-Strukturen, wie sie seit je her aus den Bereichen des Datenbankumfeldes bzw. aus der üblichen Webserver-/Browser-Architektur bekannt sind. Aus historischer Betrachtungsweise sind zentralisierte Strukturen durch die Entwicklung von mehrbenutzerfähigen Betriebssystemen geprägt, wie z.B. im Falle des UNIX-Betriebssystems, auf dem sich durch Application-Sharing-Konzepte eine Art von synchroner Groupware gestalten lässt, d.h. Verteilung der Bildschirmausgaben, Eingabemöglichkeiten auf eine Anzahl von über das Rechnernetzwerk verbundenen Benutzern.

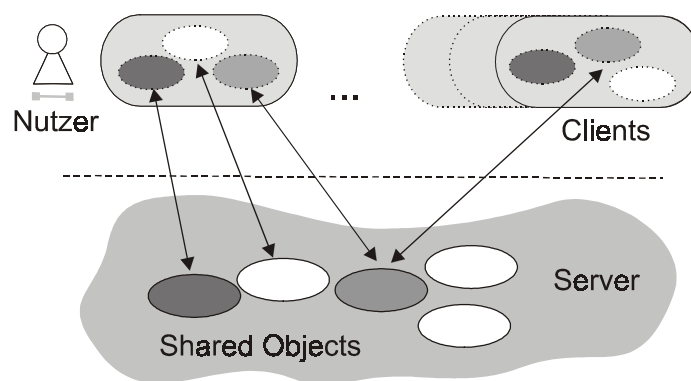


Abbildung 4-1: Gemeinsame Objekte – Shared Objects

Problematisch erscheint eine Einordnung moderner, verteilter Architekturen, die zwar meist einen zentral ausgewiesenen Server besitzen, jedoch durchaus Teile der Applikation in den Clients besitzen können, also Ansätze einer replizierten Architektur aufweisen. Insofern handelt es sich um eine Klasse von hybriden Strukturen, die Elemente sowohl rein replizierter Architekturen als auch Teile aus zentralisierten Strukturen besitzen.

Ein wichtiges Element bei der Betrachtung von Architekturen eines CSCL- oder CSCW-Systems ist die Frage der Abbildung von Objekten innerhalb des Systems. Frühe Systeme konzentrieren sich primär auf das Ausstatten der Benutzerschnittstelle mit kooperativen Fähigkeiten.¹⁸³ Erst in einem zweiten Schritt setzt sich ein Verständnis durch, welches kooperative Prozesse stark von der Art und Weise der Ablage von Informationen abhängig macht. Insofern ist es wesentlich zu unterscheiden, welche Möglichkeiten eine Groupware-Applikation oder ein Toolkit bereitstellt, um auf der einen Seite Benutzerschnittstellen mit kooperativen Fähigkeiten (z.B. synchronisierte Fenster, Telepointer etc.) auszustatten und auf der anderen Seite die gemeinsam genutzten Objekten („shared objects“) bereitzustellen. Von Trevor et al. wird die obige Unterscheidung als das zur Verfügung-stellen von „shared interface“-Diensten [Trevor et al. 1994] und einem „shared object service“ charakterisiert (vgl. Abbildung 4-1). Nur wenige Groupware-

¹⁸² Eine geradlinige Definition von zentralisierter Groupware liefern Greenberg und Roseman insofern, als sie die Clients lediglich für die Aufbereitung von Ein- und Ausgaben durch den Benutzer vorsehen (vgl. [Greenberg & Roseman 1999, S. 137]).

¹⁸³ Hierbei handelt es sich vielfach um einfache Screen Sharing-Ansätze.

Anwendungen und spezielle Toolkits sind unter Paradigma der Bereitstellung gemeinsamer Objekte gestaltet.

Unter dem Aspekt der Bereitstellung kooperativ genutzter Daten innerhalb einer Applikation gilt es des Weiteren zu unterscheiden, mit welcher Granularität die Daten zwischen kooperierenden Instanzen ausgetauscht werden. Im Wesentlichen sind drei grundsätzliche Strategien möglich:

- I. Verteilung und Synchronisation eines gemeinsamen Zustandes zwischen kooperierenden Instanzen, z.B. durch die Replikation von Objekten oder Applikationsdaten.
- II. Synchronisation der Benutzerschnittstellen von verteilten Applikationen – Umverteilung von Ein- und Ausgaben. Dies kann z.B. durch ein Ereigniskonzept (Event Sharing) geschehen.
- III. Umverteilung der Benutzerschnittstelle nur einer Applikation – typisches zentralisiertes Konzept.

Neben reinen Lernumgebungen existieren eine ganze Reihe von Baukästen zur Konstruktion von CSCL-Applikationen. Diese Baukästen oder auch Frameworks für die Gestaltung kooperativer Applikationen sind eng mit der Begrifflichkeit der Anpassbarkeit oder auch „tailorability“ verbunden. Im Kommenden sei primär auf die technische Sichtweise der Entwicklung von komponentenbasierter, kooperationsunterstützender Software eingegangen.

4.1.2 Collaboration Unaware/Aware und Collaboration Transparent

Aus technischer Sicht lassen sich kooperative Applikationen nach dem Grad der Verschmelzung der Funktionen einer Kooperationsunterstützung mit der Applikationssemantik differenzieren. Diese von Dommel und Aceves gewählte und beispielsweise von Li et al. aufgegriffene Klassifizierung unterteilt kooperative Applikationen in die Klassen „collaboration unaware“, „collaboration aware“ und „collaboration transparent“ (vgl. [Dommel & Aceves 1997, S. 31] und [Li et al. 1999]).

Bei „collaboration unaware“-Applikationen handelt es sich um gewöhnliche Ein-Benutzerapplikationen, beispielsweise eine herkömmliche Textverarbeitung, welche durch spezielle Mechanismen einer Groupware-Umgebung mit kooperativen Fähigkeiten ausgestattet wird. Kooperative Funktionalitäten sind nicht in die eigentliche Applikation integriert.

So genannte „collaboration aware“-Applikationen, wie z.B. ein kooperativer Editor oder ein Shared Whiteboard, sind speziell auf den Einsatz in einem kooperativen Prozess abgestimmt, d.h. die Funktionalität der Applikation wurde für das kooperationsunterstützende Umfeld entwickelt und ist technisch auf kooperative Mehrbenutzer-Situationen ausgelegt. Sie verfügen über Informationen zum aktuellen Kollaborationskontext und können diesen gezielt nutzen.

Die Klasse der „collaboration transparent“-Applikationen (vgl. hierzu auch [Lantz 1990]) schließlich bilden ein Hybrid aus Applikationen, welche eigentlich keine Groupware-Funktionalität besitzen, und einem speziellem Framework, welches die kooperative Funktionalität in die Applikation integriert.¹⁸⁴

¹⁸⁴ Der sTeam-Ansatz bildet in seiner Grundstruktur eine „collaboration aware“-Applikation. Wesentliche Komponenten der Umgebung sind auf den Kooperationsprozess abgestimmt und nutzen gezielt den Kontext des Kooperationsprozesses. Gleichzeitig erhält das sTeam-System durch Client-Framework und nutzerseitige Programmierbarkeit des Servers den Charakter eines „collaboration

4.1.3 Anpassbarkeit – Customisation

Kooperative Wissensräume konzentrieren sich in wesentlichen Punkten auf die nutzerseitige Strukturierbarkeit, aber auch auf die nutzerseitige Anpassbarkeit der Lernumgebung.

Eine architektonische Umsetzung des Anspruchs einer offenen, erweiterbaren und von den Nutzern gestaltbaren Umgebung setzt eine detaillierte Analyse verschiedener existierender Konzepte anpassbarer Software voraus, speziell unter dem Aspekt anpassbarer kooperationsunterstützender Softwaresysteme.¹⁸⁵

Sicht des Nutzers

Nutzerseitige Konfigurier- und Erweiterbarkeit ist sicherlich eines der vorrangigen diskutierten, aber wenig erreichten Ziele bei der Gestaltung kooperativer Systeme.

Anpassbarkeit und Konfigurierbarkeit hat bei der Gestaltung von Einzelplatzanwendungen eine lange Tradition.¹⁸⁶ Die Anpassbarkeit („customisation“) und Zuschneidbarkeit von Softwaresystemen ist schon seit einiger Zeit ein viel diskutiertes Thema in der Softwareentwicklung und Softwareergonomie (vgl. [Bannon 1992]). Schon früh wird der Einsatz ihrer Techniken auch in dem Bereich des CSCW geprüft (vgl. [Robinson 1992]).¹⁸⁷ Auch die Anpassbarkeit von CSCL-Systemen wird vereinzelt als wesentlich erachtet (vgl. [Dourish et al. 1999a]).

Wichtiges Kriterium für ein erfolgreich anpassbares System ist die Anpassbarkeit durch den „einfachen“, im Sinne von nicht mit Interna des Systems vertrauten Nutzer (vgl. [MacLean et al. 1990]). Gleichzeitig ist jedoch eine möglichst universelle Programmierbarkeit des Systems und damit das Dilemma einer hohen Komplexität der Anpassbarkeit unumgänglich.

Nach Henderson und Kyng existieren drei Ebenen der Komplexität von Anpassungen: Das Wählen von Alternativen von möglichem Verhalten, die Konstruktion und Komposition von neuem Verhalten aus existierenden Bausteinen und die wirkliche Modifikation, Veränderung, was einer Um- oder Neuprogrammierung gleichkommt (vgl. [Henderson & Kyng 1991, Wulf 1999]). Mørch et al. fügen der obigen Unterscheidung die Erweiterung einer Applikation durch Programmierschnittstellen wie APIs hinzu (vgl. [Mørch et al. 1998]).

Ein Verständnis von „Anpassbarkeit“ und „Konfigurierbarkeit“ im Zusammenhang der Beschreibung von kooperationsunterstützenden Umgebungen erstreckt sich in vielen Fällen lediglich auf zwei Dimensionen in dessen Nutzung. Zunächst wird die Möglichkeit betont, Teile der Benutzerschnittstelle den persönlichen Bedürfnissen der Nutzer anpassbar zu gestalten.¹⁸⁸ Hierbei handelt es sich aber nur um eine Konfigurierbarkeit

transparent“-Frameworks, d.h. spezielle, den kooperativen Lernprozess unterstützende Werkzeuge wie ein Shared Whiteboard lassen sich mit technisch überschaubarem Aufwand in die Umgebung integrieren.

¹⁸⁵ Das grundsätzliche Softwaredesign des sTeam-Systems ist in wesentlichen Elementen der nutzerseitigen Konfigurierbarkeit durch Architekturlösungen der MUDs und MOOs beeinflusst.

¹⁸⁶ Meilensteine auf dem Weg zu anpassbaren oder auch programmierbaren Systemen wie z.B. Apples HyperCard-System (vgl. [Atkinson 1987] und [Williams 1987]) oder auch der EMACS-Editor dokumentieren die Idee, Software nutzerseitig gestaltbar zu machen.

¹⁸⁷ Bentley und Dourish betonen in ihrer Definition von „customisation“ insbesondere die aktive Rolle des Benutzers eines kooperationsunterstützenden Systems (vgl. [Bentley & Dourish 1995, S. 147]).

¹⁸⁸ Greenberg betont in seiner Definition von anpassbarer, „personalizable“ Groupware primär die Individualisierung von Erscheinungsbild und Verhalten der Benutzerschnittstellen. Ihm geht es aber

der Benutzerschnittstelle oder einer Modifikation der Sicht des Nutzers auf Objekte und Materialien. Das Konzept des relaxed-WYSIWIS ist in dieser Form lediglich eine andere Metapher für eine Möglichkeit der Konfigurierbarkeit der Sicht auf Objekte, also die Individualisierbarkeit des Wahrnehmungsbereichs auf die gemeinsamen Objekte.

Als zweite weitaus interessantere Dimension der Anpassbarkeit wird die Individualisierung der Struktur der Inhalte verstanden. In diesem Sinne geht das hier vertretene Verständnis von Anpassbarkeit über das der Konfiguration von Benutzerschnittstellen oder des Schaffens einer persönlichen Sicht auf die Materialien des Kooperationsprozesses hinaus: Anpassbarkeit muss es den Nutzern erlauben, dauerhafte persönliche Strukturierungen und Sichten auf einen Wissensraum anzufertigen. Insbesondere sind diese Sichten durch einen kooperativen Prozess schrittweise zu verfeinern und müssen sich in der persistenten Struktur der Umgebung niederschlagen (Medienfunktion des Arrangierens und Verknüpfens).

Dies schließt ebenfalls die Möglichkeit für die Nutzer mit ein, eigene Werkzeuge zu erzeugen (Medienfunktion des Erzeugens), existierende Werkzeuge und ihre Benutzerschnittstellen zu modifizieren (Skalierbarkeit der Benutzerschnittstelle) und das Erscheinungsbild und die Funktionalität der Umgebung dauerhaft ihren persönlichen Bedürfnissen und denen der Gruppe anzupassen. In diesem Sinne ist Anpassbarkeit besser charakterisiert als Individualisierbarkeit und, wie Dourish et al. richtig bemerken, als ein Wechselspiel aus Bedürfnissen aller Anwender einer Umgebung.¹⁸⁹ Trotz dieses Verständnisses der Berücksichtigung sämtlicher Interessen und „Anpassungs“-wünsche einer Umgebung spiegelt sich in fast allen existierenden Systemen zumeist der Anspruch der Anpassung des Systems und weniger der Anspruch der individuellen und kooperativen Strukturierung von Materialien wider.

Eine zweite Begrifflichkeit aus dem Bereich der komponentenbasierten Softwareentwicklung ist das Konzept der „tailorability“. Eine eindeutige Bedeutung und Definition des Begriffes der „tailorability“ ist sicherlich nur schwerlich zu finden. Eine begriffliche Abgrenzung von „tailoring“ liefert Mørch, indem er es als einen der eigentlichen Applikationsentwicklung nachgestellten Prozess beschreibt [Mørch et al. 1998]. Tailoring findet also nach der eigentlichen Entwicklungsphase eines Produktes oder Werkzeuges statt und dient nach seinem Verständnis der Anpassung an neue Bedürfnisse und Anforderungen, die in der Designphase nicht berücksichtigt wurden. Biermans und Hofte verstehen unter „tailorability“ die Fähigkeit eines Softwaresystems, an die Vorlieben eines Benutzers angepasst zu werden (resultierend aus der zu bearbeitenden Aufgabe bzw. dessen Kontext, den konkreten Bedingungen der Umwelt oder persönlichen Anforderungen). Diese Anpassungen müssen auch nach Implementierung eines Softwarewerkzeuges durch Dritte dauerhaft möglich sein [Biermans & Hofte 1999].

Im Rahmen dieser Arbeit seien die genannten Kriterien, welche „tailorability“ als die Fähigkeit eines Softwaresystems charakterisieren an persönliche Vorlieben angepasst zu werden, bewusst um das Kriterium der Erweiterbarkeit und des Ausbaus ergänzt. Das landläufige Verständnis von Anpassbarkeit erscheint unter dem Anforderungshorizont der Unterstützung grundsätzlich verschiedener Anwendungsszenarien und Lernsituationen zu schwach.

ebenfalls um eine Form der Skalierbarkeit der Funktionalität [Greenberg 1991, S. 19], vgl. hierzu auch [Wulf 1999, S. 52ff.].

¹⁸⁹ „[...] that customisation is a collaborative phenomenon.“ [Dourish et al. 1999a, S. 11].

Die Anpassbarkeit einer Benutzerschnittstelle¹⁹⁰ oder auch der Funktionalität eines ganzen Systems hängt neben der Möglichkeit der Konfiguration des Erscheinungsbildes oder der Erweiterbarkeit von Verhalten und Umfang des Werkzeuges entscheidend von der Akzeptanz und Motivation der Nutzer hierzu ab. MacLean nennt dies *eine Kultur der Anpassbarkeit* („tailoring culture“) [MacLean et al. 1990, S. 175]. Sicherlich richtig ist seine Forderung, Anpassbarkeit nicht nur erfahrenen und der Programmierung fähigen Anwendern zu ermöglichen.¹⁹¹

Eine wirkliche Herausforderung ist es gewiss, dem „einfachen“ Nutzer oder im Kontext des kooperativen Lernens dem Lernenden als Anwender des Systems die Möglichkeit der Anpassung von Benutzerschnittstelle und Funktionalität zu ermöglichen.¹⁹²

Sicht der Systemarchitektur

Gegenüber einer Betrachtung anpassbarer kooperationsunterstützender Systeme aus Sicht des Benutzers ist die Untersuchung von flexiblen Architekturkonzepten kooperationsunterstützender Systeme unumgänglich. Ziel hierbei ist es, den Ausbau von Clients und Server weitgehend ohne gegenseitige Einflussfaktoren voneinander vornehmen zu können.

Roseman und Greenberg schlagen hierzu ein Konzept der „open protocol“-Technologie und eine konkrete Systemarchitektur vor, die es erlaubt, auch nach Systemeinführung einer Groupware-Umgebung diese um weitere Module zu ergänzen [Roseman & Greenberg 1993]. Dies geschieht durch eine Variante des Client-Server-Ansatzes, in welcher der Client die wesentliche Interpretation und die semantischen Festlegungen (das Verhalten) der im Server gespeicherten Daten trifft.¹⁹³ Das Konzept bedingt, dass es verschiedene Clients geben kann, die Objektzustände des Servers auf unterschiedliche Weise interpretieren.

Als Anwendungen dieses Konzepts stellen Roseman und Greenberg verschiedene Strategien des Floor Control vor, die über verschiedene Clients („controller“) umgesetzt werden. Diese modifizieren gezielt Statusvariablen des Zugriffs im Server über das Protokoll.

Erweiterbarkeit ist traditionell ein wichtiges Thema der Systemarchitektur einer Reihe von MUDs und MOOs. Schon Curtis betont, dass MUDs von den Nutzern selbst erweitert werden, und macht diese Eigenschaft zu einem wichtigen Kriterium der Definition und des Erfolges seines MUD.¹⁹⁴ Ein Beispiel für diese Fähigkeiten, MUDs und MOOs um spezielle Funktionalitäten zu ergänzen, ist sicherlich das ProgrammingLand

¹⁹⁰ Der Aspekt der Anpassung von Benutzerschnittstellen („customisation“) wird auch als „adaptability“ bzw. „adaptivity“ gekennzeichnet. „Adaptability“ bedeutet die Anpassbarkeit der Benutzerschnittstelle durch den Benutzer und „adaptivity“ die automatische Anpassung auf bestimmte Aktionen des Nutzers.

¹⁹¹ MacLean et al. unterscheiden Nutzer nach „Worker“, „Tinkerer“ und „Programmer“. Hierbei wird gezielt nach dem technischen Verständnis im Umgang, dem Experimentiergeist und den Programmierfähigkeiten der Anwender unterschieden (vgl. [ebd., S. 175]).

¹⁹² Eine Idee, die Appelt et al. in ihrem BSCW-System aufgreifen (vgl. [Appelt et al. 1998]).

¹⁹³ Es interagieren ein „controlled object“ (Server) und ein „controller“ (Client) über ein spezielles Protokoll. Hierbei bildet das „controlled object“ lediglich die Persistenzschicht für Daten, seine Zustände können nur über das Protokoll modifiziert werden. Die Bedeutung dieser Zustände wird erst durch den „controller“ definiert, d.h. das Verhalten, die Interpretation von Zuständen, wird ausschließlich in dem Client vorgenommen.

¹⁹⁴ „A MUD is extensible from within; a user can add new objects to the database such as rooms, exits, 'things', and notes. Certain MUDs, [...] even support an embedded programming language in which a user can describe whole new kinds of behavior for the objects they create. [...] Finally, one of the great strengths of MUDs lies in the users' ability to customize them, to extend them, and to specialize them to the users' particular needs“ [Curtis 1992, S. 2].

MOO [Hill & Slator 2000]. Hier werden von Lehrenden und Lernenden eine ganze Anzahl interaktiver Demonstrationsobjekte geschaffen, die sich mit dem Erlernen von Programmierfähigkeiten beschäftigen.

Die architektonischen Grundlagen der Gestaltung eines programmierbaren, erweiterbaren Systems beeinflussen basierend auf den Erfahrungen der MUD-Technologie die Konzeption des vorgeschlagenen Rahmenkonzepts nachhaltig.¹⁹⁵

Ohne den konkreten Architekturbetrachtungen vorgreifen zu wollen, ist das Konzept der Anpassbarkeit in sTeam in einer Reihe von Architekturentscheidungen verwirklicht:

- *Interne Programmierbarkeit:* Durch den Aufbau des Systems in einer interpretierten Programmiersprache können virtuelle Wissensräume dynamisch erweitert werden (vgl. Abschnitt 5.2).
- *Modulare Server-Architektur:* Der Server ist modular in Form von Grundmodulen aufgebaut. Zusätzliche Module können in die Gesamtarchitektur eingebracht werden (vgl. Abschnitt 5.2).
- *Freie Serverobjekte:* Die Struktur von Objekten des Servers ist flexibel anpassbar und erweiterbar. Das Hinzufügen neuer Objekte ist zur Laufzeit des Systems möglich (vgl. Abschnitt 5.2.2).¹⁹⁶
- *Freie Attributstruktur:* Attribute können flexibel an Objekten, auch während des Betriebs, registriert werden (vgl. Abschnitt 5.2.3).
- *Universelle Client-Server-Schnittstellen:* Die Schnittstellen zwischen Client und Server reduzieren sich auf abstrakte Funktionen der Modifikation und des Austauschs von Objekten und Attributen. Funktionen können direkt in den Objekten des Servers angesprochen werden (vgl. Abschnitt 5.2.8).

4.2 Architekturansätze

Ein vollständiger Überblick über existierende technische Ansätze ist aufgrund ihrer Vielfalt und mangelnden Hintergrundinformationen zu den einzelnen Architekturkonzepten nicht möglich und zugleich nicht sinnvoll.

In der folgenden Betrachtung ausgewählter Architekturen werden neben zentralisierten und replizierten Ansätzen speziell Frameworks (Baukästen) zur Entwicklung kooperationsunterstützender Applikationen dargestellt. Diese unterstützen im einfachen Fall die Kommunikation netzgestützter Applikationen, können jedoch ebenfalls auf weitaus höherer Ebene angesiedelt sein, beispielsweise erweiterte Mechanismen zur Kooperationsunterstützung besitzen.

Weiteres Schwergewicht liegt auf der Betrachtung WWW-basierter Systeme zur Kooperationsunterstützung. An allen genannten Ansatzpunkten werden primär die kooperativen Aspekte der jeweiligen Systeme betont.

4.2.1 Zentralisierte Architekturen – kooperatives MVC-Konzept

Die klassische Motivation für die Idee einer zentralisierten Applikation ist das MVC (Model View Controller)-Konzept von Krasner und Pope (vgl. [Krasner & Pope 1988]).

¹⁹⁵ Das von Roseman und Greenberg genannte Verfahren lässt sich auf das Modifizieren von Attributen innerhalb von kooperativ genutzten Objekten übertragen. Die sTeam-Architektur ergänzt die Client-Server-Schnittstelle um die Möglichkeiten beliebige Funktionen (Methoden) innerhalb des Servers zu aktivieren.

¹⁹⁶ In der Literatur finden sich einige wenige vergleichbare Ansatzpunkte, z.B. die Architektur der EVOLVE-Plattform, die es erlauben soll, zur Laufzeit Komponenten einer Groupware-Applikation zu arrangieren und kombinieren (vgl. [Stiemerling 1999]).

Die Idee der Entkopplung der eigentlichen Applikation (Model) von Programmbereichen, welche Ein- und Ausgaben bearbeiten (Controller), und Programmabschnitten, welche sich mit der grafischen Repräsentation des Models befassen (View), legt die Übertragung auf ein Architekturmodell für die Entwicklung von synchronen Groupware-Anwendungen nahe.

Basierend auf einer zentralisierten Architektur, d.h. einer zentralen Instanz, welche die eigentliche Applikationssemantik bereitstellt (dem Model), werden View- und Controller-Komponenten auf allen an einer synchronen Sitzung teilnehmenden Instanzen verteilt. Es handelt sich in dieser Form um eine semi-replizierte Architektur, in der zumindest die View-/Controller-Instanzen auf die Clients repliziert werden und über einen Benachrichtigungsservice mit dem Server verbunden sind (vgl. Abbildung 4-2). Es wird nicht immer eine strenge Trennung von View und Controller vorgenommen, was den Grundgedanken der Trennung der eigentlichen Applikation von der grafischen Aufbereitung einer Sicht auf diese nicht negativ beeinflusst. Durch eine Verteilung von Ereignissen (Events) über das Netzwerk sorgt die eigentliche Applikation für eine stetige Konsistenz der verbundenen View-/Controller-Kombinationen auf den Clients und gewährleistet so in der einfachsten Form Synchronizität (WYSIWIS) der verbundenen Clients.

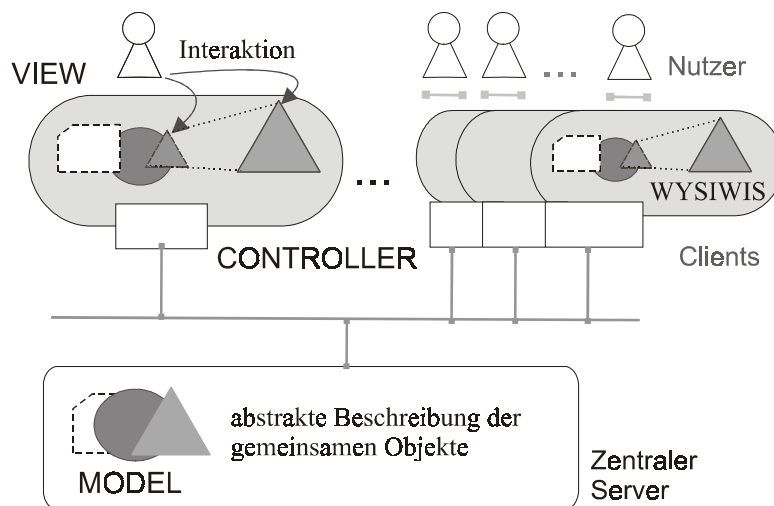


Abbildung 4-2: Zentralisierte Model View Controller (MVC)-Architektur

Eine konsequente Entwicklung und Übertragung des MVC-Konzeptes auf verteilte Anwendungen und damit auf die Entwicklung von Groupware findet sich in der Architektur des Rendezvous-Systems (vgl. [Patterson et al. 1990] und [Hill et al. 1994]).¹⁹⁷ Historisch bedingt stand bei den frühen Bemühungen der Entwickler von CSCW-Systemen und Toolkits primär die Bereitstellung von Diensten zur gemeinsamen Sicht auf eine Applikation im Vordergrund. Eine nahe liegende Umsetzung hierzu ist es, eine Applikation mit einer Anzahl von synchronisierten, verteilten Fenstern auszustatten. Rendezvous leistet diesen Schritt.

¹⁹⁷ Rendezvous ist realisiert in Common Lisp, einem Grafiksystem unter X-Windows, und vertritt die Klasse der zentralisierten Architekturen. Es handelt sich um ein frühes Groupware-Toolkit zur Entwicklung von verteilten Benutzerschnittstellen, fällt also nach Trevor et al. in die Klasse der „shared interface service“-Umgebungen [Trevor et al. 1994].

Zentralisierte Architekturen besitzen in dieser Form den Charakter einer isolierten Applikation, welche im Wesentlichen ein und denselben Kontext für alle ihre Nutzer bereitstellt. Es wird für jeden Teilnehmer einer kooperativen Sitzung ein und derselbe Zustand repliziert.

Damit eignen sich zentralisierte Architekturen für die Entwicklung von vorrangig synchron geprägten Systemen. Extreme Beispiele, welche einen hohen Kopplungsgrad der synchronen Applikationen aufweisen,¹⁹⁸ werten keinerlei Kontextinformationen der kooperierenden Teilnehmer aus und eignen sich damit nur beschränkt für die im Rahmen dieser Arbeit verfolgten kooperativen Arbeitsformen (vgl. [Li et al. 1999, S. 334]).

Historisch bedingt handelt es sich bei existierenden Ansätzen, welche auf einer zentralistischen Architektur basieren, zumeist um autarke und nicht WWW-basierte Lösungen. In Bezug auf die technische Realisierung des vorgestellten Rahmenkonzepts spielen rein zentralisierte Architekturen damit nur eine begrenzte Rolle.

Die Mehrzahl existierender Ansätze wählen eine replizierte Architektur. Die Spannweite hierbei reicht von vollständig replizierten Systemen, bei denen auf jedem Client eine exakt identische Instanz der gemeinsamen Objekte vorliegt, bis zu replizierten Ansätzen, in denen nur ein Teil der gemeinsamen Objekte und Daten auf den jeweiligen Client übertragen werden. In sämtlichen Fällen existiert ein zentraler Server zur Verwaltung der kooperativen Sitzung bzw. ein ausgewiesener Client übernimmt diese Aufgabe.

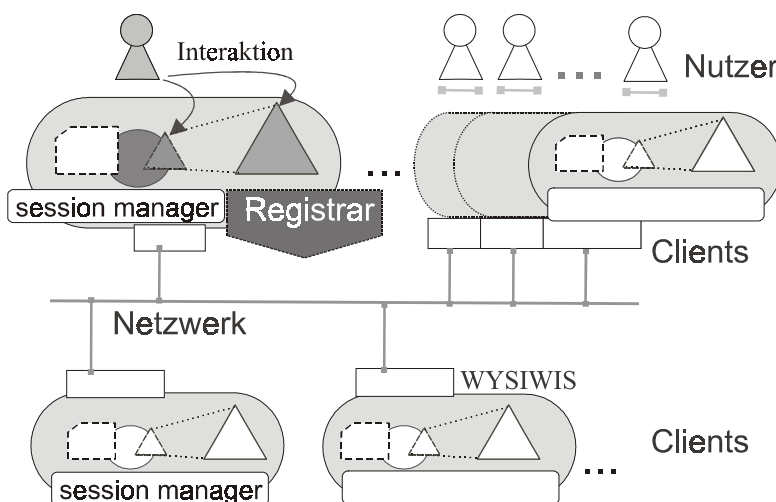


Abbildung 4-3: Replizierte Architektur

4.2.2 Replizierte Architekturen

Prototypisch für den Aufbau einer synchronen replizierten Architektur sind fast identische Instanzen einer Applikation (vgl. Abbildung 4-3). Eine Gruppe von Nutzern (Applikationen) nimmt jeweils an einer Sitzung (Session) teil, diese wird von einer ausgewiesenen Instanz verwaltet.

Wesentliche Elemente einer derartigen Architektur sind nach Greenberg und Roseman der „registrar“, der „session manager“ und die eigentliche, kooperative Applika-

¹⁹⁸ Engl. „state replication“, den Zustand einer kooperativ genutzten Applikation auf alle verbundenen Clients replizieren.

tion, der „conference application“ (vgl. [Greenberg & Roseman 1999, S. 142]). Hierbei bildet als einzig zentralisiertes Element der „registrar“ als Prozess den Verbindungspunkt einer kooperativen Sitzung. Seine Adresse ist bekannt, neue Teilnehmer einer Sitzung melden sich über ihre „session manager“ an diesem zentralen Prozess an. Jede Instanz der verteilten Applikation besitzt einen eigenen Sitzungsmanager, der die Teilnahme an einer Gruppensitzung mit dem zentralen „registrar“ organisiert und verwaltet. Schließlich bildet die „conference application“ die eigentliche kooperative Applikation, die als typisches Merkmal einer replizierten Architektur als vollständige Instanz auf den einzelnen Arbeitsstationen ausgeführt wird. Über Kommunikationskanäle werden zwischen den Applikationen Daten ausgetauscht.

Als einer der ersten Vertreter der Gattung der replizierten Architekturen gilt das GroupKit-System zur Entwicklung von Groupware-Anwendungen [Roseman & Greenberg 1992]. Konzeptionell reiht sich GroupKit in die Klasse der ersten Laufzeitumgebungen und Bibliotheken ein, die die Entwicklung von Groupware-Applikationen ermöglichen sollten.¹⁹⁹

Ähnliche Ansätze, wie z.B. das DistView-System [Prakash & Shim 1994], unterstützen die Entwicklung von synchronen kooperativen Applikationen auf Basis des Application-Sharing, also der gemeinsamen Nutzung einer Applikation durch Bereitstellung verschiedener Sichten. DistView ist vollständig objektorientiert entworfen und bietet die Möglichkeit, eine kooperative Applikation durch ein so genanntes „object replication scheme“, also durch Spezifikation von gemeinsam genutzten Objekten, zu entwerfen. Diese werden zu den entsprechenden Clients der verteilten Anwendung repliziert. Spezielles Augenmerk legt DistView auf die Bereitstellung verschiedener Kopplungsgrade der Kooperation in Form von privaten oder gemeinsam genutzten Fenstern für eine Groupware-Applikation.

Von Seiten der Architektur betrachtet besitzt DistView eine ganze Anzahl von interessanten Merkmalen wie Proxy-Objekte und Replikationsmechanismen von Oberflächenelementen (beinhaltet in dieser Form auch Merkmale zentralisierter Architekturkonzepte).

Zu einer Abgrenzung der vorgeschlagenen Architektur gegenüber etablierten technischen Ansätzen ist eine Betrachtung des an der Fraunhofer Gesellschaft (ehemals GMD) entwickelten COAST (Cooperative Applications Systems Toolkit, [Schuckmann et al. 1996]) unerlässlich.²⁰⁰ COAST ist in die Klasse der vollständig replizierten Frameworks zur Entwicklung von kooperativen Applikationen einzuordnen. Es zählt als Plattform für eine Reihe von CSCL-Lösungen zu den über lange Jahre entwickelten und ausgebauten Systemen. COAST ist durchgängig objektorientiert konzipiert und bietet verschiedene Komponenten zur Gestaltung von kooperativen Applikationen durch für den Entwickler transparente Replikations- und Kopplungsmechanismen.

Zielrichtung von COAST ist die Entwicklung verteilter, synchroner (Smalltalk-) Applikationen. Diese sind zunächst nicht WWW-basiert.²⁰¹

¹⁹⁹ GroupKit wurde Anfang der 90er Jahre an der Universität Calgary auf der Basis von C und später in Tcl/Tk. implementiert. Es bildet die Laufzeitumgebung einer Groupware-Applikation und organisiert den Aufbau der jeweiligen Netzwerkverbindungen.

²⁰⁰ Das COAST-System ist als neueste Entwicklung Open-Source, vgl. <http://www.opencoast.org/>, Stand 1.12.2001.

²⁰¹ Basierend auf COAST wurden in den letzten Jahren eine ganze Reihe von Beispielapplikationen entwickelt (vgl. [Wessner et al. 1998] und [Wessner & Pfister 2001, S. 254ff.]).

Architektonisch prägt COAST eine ganze Klasse von replizierten synchronen Systemen. Eine COAST-Applikation besteht aus kooperierenden Objekten, die auf den jeweiligen Kooperationspartnern verteilt vorliegen. Ausgehend von diesem Ansatz verwaltet eine COAST-Applikation die gemeinsamen Informationen in Form von Dokumenten (Shared Documents) oder Teilen von diesen, den „document parts“. Der Zugriff von Benutzern auf die gemeinsamen Dokumente wird auf den einzelnen Clients über Sitzungsobjekte (Session Objects) koordiniert. In ihnen wird eine Liste der Objekte verwaltet, an denen ein Benutzer zurzeit partizipiert, Awareness-Information bereitgestellt und der Kopplungsgrad (Coupling) gemeinsamer Informationen bestimmt. Ein Sitzungsobjekt ist über den Transaktionsmanager und den Replikationsmanager mit dem Netzwerk verbunden (vgl. [Schuckmann et al. 1996, S. 32]). Der Transaktionsmanager koordiniert den Zugriff auf die einzelnen gemeinsamen Objekte, welche als gekapselte Aktionen (Transaktionen) an den Objekten verarbeitet werden. Der jedem Client zugeordnete Replikationsmanager gibt z.B. Veränderungen an den Objekten in Transaktionen gekapselt an den Transaktionsmanager weiter. Jede Sitzung wird über einen Sessionmanager verwaltet. Benutzer können über ihn neue Sitzungen erzeugen bzw. an existierenden Sitzungen teilnehmen.

Als konsequent replizierter Ansatz werden Objekte auf den jeweiligen Clients verwaltet, diese sind über so genannte „mediators“ für die Persistenz der Objekte verantwortlich. Jedes Dokument ist genau an einer Stelle gespeichert, es existiert kein zentraler Server zur Ablage von Objekten.

In Bezug auf den objektorientierten Ansatz, d.h. der Modellierung einer kooperativen Applikation als Verbund replizierter Objekte, ist die COAST-Architektur als sehr modern zu charakterisieren und lässt direkte Parallelen zur sTeam-Konzeption erkennen.

Die grundlegende Idee der Modellierung lässt sich in dieser Form auf moderne Java-Ansätze abbilden.²⁰²

In Abgrenzung gegenüber anderen Ansätzen ist zu betonen, dass es sich bei COAST um ein Framework zur Gestaltung von kooperativen Anwendungen für relativ synchrone Kommunikationsformen handelt. COAST betont zwar, ebenfalls ein Dokumentenmanagementsystem zu sein, und bietet entsprechend auch asynchrone Komponenten, aber die Verwaltung und Ablage von kooperativ bearbeiteten Dokumenten steht im Gegensatz zu weiteren, eher der Klasse der Dokumentenmanagementsysteme zuzuordnenden Systeme jedoch weniger im Vordergrund. Mit COAST entwickelte Anwendungen liegen monolithisch auf den einzelnen Client-Rechnern vor und sind zur Laufzeit nicht um neue Komponenten erweiterbar.

Replikation kann nicht alleinig als Konzept der verteilten Systementwicklung verstanden werden (Replikation von Objekten und Daten zwischen Clients), es kann speziell im Umfeld verteilter Datenräume Verwendung finden. Hierzu ist Lotus Notes [Kawell et al. 1988] ein prominentes Beispiel.²⁰³

Lotus Notes besitzt eine verteilte Systemarchitektur mit einem konsequent datenbankorientierten Ansatz. Aus Sicht der Nutzer betrachtet ist das Arbeiten mit Objekten

²⁰² In dieser Form ist die ebenfalls an der GMD entwickelte Java-basierte DyCE-Architektur [Tietze & Steinmetz 2000] in vielen Punkten an COAST angelehnt, obwohl DyCE auf einem semi-replizierten Ansatz mit einem zentralen Server basiert (vgl. Abschnitt 4.2.3).

²⁰³ Lotus Notes ist trotz einer Anzahl von synchronen Eigenschaften primär ein asynchrones Workflow-orientiertes System. Obwohl in den letzten Jahren Schnittstellen für das WWW entwickelt wurden, ist der Lotus Notes-Client weiterhin das typische Werkzeug zum Zugriff auf den Notes-Server.

in Notes durch Replizieren von Dokumenten aus Datenbanken geprägt. Lotus Notes entwickelt ein vollständig anderes Verständnis von Replikation als z.B. COAST oder Rendezvous. Clients können auch ohne Verbindung zum Server auf lokalen Datenbanken arbeiten und diese bei späterer Verbindung zum Server mit der Master-Datenbank abgleichen. Durch seine leistungsfähigen Replikationsmechanismen ist Notes primär als mehrbenutzerfähiges Dokumentenmanagementsystem zu charakterisieren [Mohan 1999].

Notes-Dokumente werden auch in der Metapher der Bedienung in Notes-Datenbanken abgelegt und bei Bearbeitung lokal repliziert. Jedes Dokument wird als so genannte Master-Replik in einer Notes-Datenbank verwaltet. Benutzer replizieren zur Bearbeitung das Dokument als ganzes oder Teile von ihm. Damit ist eine recht grobe Granularität in der Möglichkeit des kooperativen Bearbeitens von Dokumenten gegeben. Dokumente werden nach Bearbeitung in die Datenbank zurück überführt. Bei Konflikten (z.B. zwei Benutzer haben gleichzeitig ein Dokument modifiziert) wird auf Strategien der manuellen Konfliktlösung (Benachrichtigung der Nutzer) zurückgegriffen.

Ursprünglich als internet-, aber nicht WWW-gestütztes System entwickelt, bietet Lotus mit der Domino-Technologie (die Bezeichnung Lotus Notes Domino steht für die Servertechnologie) die Möglichkeit, internetgestützte Groupware-Applikationen basierend auf Notes-Datenbanken zu entwickeln [Thomas & Peasley 1998]. Damit ist Lotus Notes als echtes Groupware-Toolkit zu charakterisieren. Zur Programmierung von primär asynchronen Notes-Applikationen stehen verschiedene Ebenen von Programmierschnittstellen zur Verfügung, die eng mit dem jeweiligen Betriebssystem des Clients verknüpft sind.

Die Architektur des Notes-Systems bietet wenig bekannte Eigenschaften anderer Groupware-Toolkits. Sie soll wegen ihres Charakters als verteilte Datenbankarchitektur an dieser Stelle nicht genauer untersucht werden. Die Replikation auf Dateiebene bietet von der Programmierenebene betrachtet eine gänzlich unterschiedliche Art, kooperative Applikationen zu entwickeln, wie sie z.B. sTeam erlaubt. So sind Benutzerrechte unmittelbar an die Berechtigungen zum Zugriff auf entsprechende Datenbanken geknüpft.

Die sTeam-Architektur verbindet die Vorteile replizierter und zentralisierter Architekturkonzepte. Sie besitzt als Client-Server-Applikation einen zentralen Server, welcher gemeinsame Objekte bereitstellt, Ereignisse bearbeitet, diese umverteilt und Kommunikationsflüsse koordiniert. Gleichzeitig werden wesentliche Elemente, z.B. gemeinsam genutzte Objekte, auf die Clients repliziert. Hierbei findet jedoch keine vollständige Replikation kooperativer Objekte statt, es werden lediglich Objekt-„Hülsen“, so genannte Proxy-Objekte bereitgestellt, die eine für die jeweiligen Clients transparente Kommunikation mit dem Server bereitstellen (vgl. Abschnitt 5.3 und Abbildung 5-9). In dieser Form besitzt sTeam aus Sicht der Clients den Charakter einer verteilten, replizierten Architektur, aus Sicht des Server handelt es sich jedoch um einen zentralisierten Ansatz.

Auch zu einem auf die Mehrbenutzersituation übertragenen MVC-Modell lassen sich eine ganze Reihe von Parallelen erkennen. Es findet eine Anwendung des MVC-Prinzips auf der Ebene von einzelnen kooperativen Objekten statt. Objekte besitzen ein definiertes Verhalten (Funktionalität und Methoden, mit denen sie auf Ereignisse reagieren) und eine Repräsentationsebene bzw. ein Erscheinungsbild. Das kooperative MVC-Prinzip ist in dieser Form sowohl auf Ebene des Servers als auch auf Ebene des Clients umgesetzt.

4.2.3 Objektorientierte Frameworks kooperativer Applikationen

Im Hinblick auf die Umsetzung des skizzierten Rahmenkonzepts ist eine genaue Betrachtung existierender, zumeist Java-basierter Rahmenarchitekturen aus dem Bereich synchroner CSCW- und CSCL-Systeme sinnvoll. Eine Analyse vorhandener Systeme dokumentiert weite Teile der Realisierung möglicher Clients sowie die Umsetzung gemeinsamer Objekte kooperativer Wissensräume. Auch wenn eine ganze Anzahl der im Folgenden untersuchten Systeme aus Sicht ihrer Anwendung nur eine untergeordnete Rolle spielen, liefern sie doch wichtige Impulse für die sTeam-Architektur und grenzen diese gegenüber vorhandenen Lösungen ab.

In den letzten Jahren sind eine größere Anzahl von objektorientierten und speziell Java-basierten Rahmenarchitekturen für die Entwicklung von kooperativen Applikationen entstanden. Sie bilden den Trend der Nutzung des WWW als Kommunikationsmedium von kooperativen, sitzungsorientierten Werkzeugen. Frühe Konzeptionen nutzen zwar allgemein das Internet als Kommunikationsschicht für synchrone Applikationen; Clientwerkzeuge, Adressierung von Sitzungen/Knotenpunkten von Kooperationen und die Integration externer Ressourcen sind jedoch zumeist proprietär gelöst, d.h. sie verhalten sich sehr eingeschränkt in den unterstützten Formaten und Protokollen.

Motiviert durch einen generellen Trend in der Softwareentwicklung hin zu verteilten Architekturen existieren eine Reihe von verbreiteten Konzepten, wie z.B. RMI²⁰⁴ oder der CORBA-Standard²⁰⁵, die wichtige Bausteine zur Entwicklung verteilter, netzbasierter Applikationen bilden. Da es sich um alleinige Mechanismen der Inter-Applikationskommunikation handelt, werden sie im Folgenden im Umfeld der technischen Kooperationsunterstützung auf höherer Ebene nicht berücksichtigt bzw. nicht isoliert betrachtet.

Eine erste wichtige Unterscheidung in dem Versuch, existierende Frameworks für kooperative Applikationen zu klassifizieren, ist der Grad an Komplexität in der Bereitstellung kooperativer Funktionalitäten.

Während einige Frameworks wie z.B. das JSDT²⁰⁶ oder Pavilion [McKinley et al. 1999] eher elementare Kommunikationsmechanismen zum Datenaustausch der kooperativen Applikationselemente über das Netz bereitstellen (vgl. hierzu ebenfalls HOUCOM, der Fraunhofer-IGD [Einhoff et al. 2000]), bieten andere Frameworks, wie z.B. DyCE [Tietze & Steinmetz 2000] durch die Bereitstellung von Kooperationsunterstützung auf weit höherer Ebene und komplexe, objektorientierte Strukturen eine ganz andere Art von Unterstützung. Liegt bei den ersten das Augenmerk zumeist auf effizienten Kommunikationsprotokollen und Mechanismen des Datenverkehrs, versuchen letztere durch möglichst weit entwickelte Komponenten dem Entwickler Unterstützung auf hoher Ebene zu leisten.

Im Folgenden seien exemplarisch eine Reihe von Rahmenarchitekturen zur Entwicklung von CSCW- und CSCL-Systemen vorgestellt. Verbindendes Element aller berücksichtigten Ansätze ist ihre objektorientierte Architektur; also spezifische Mechanismen der Verwaltung kooperativer Objekte. Mit der Nutzung der Java-Technologie ergibt sich zudem ein plattformunabhängiges Einsatzfeld. Interessanterweise finden sich sowohl zentralisierte als auch verteilte Architekturen. Im Falle der zentralisierten Architekturen verschiebt sich diesbezüglich das Verständnis einer zentralisierten CSCW-

²⁰⁴ RMI (Remote Method Invocation) bezeichnet ein Paradigma des Objektzugriffs in verteilten Systemen (vgl. [Coulouris et al. 2001, S. 169ff.]).

²⁰⁵ CORBA wurde von der Object-Management-Group (OMG) verabschiedet. Vgl. <http://www.omg.org/corba/>, Stand 1.12.2001.

²⁰⁶ Java Shared Data Toolkit, vgl. <http://java.sun.com/products/java-media/jsdt/>, Stand 1.12.2001.

Architektur in Richtung der Reduzierung von Funktionalität auf der Clientseite, beispielsweise durch Verlagerung von minimalen Zugriffsfunktionen auf das kooperative System in einen Webbrowser.

Java-basierte Frameworks auf niedriger Ebene der Kooperationsunterstützung

Im Folgenden werden exemplarisch TANGO, Pavilion, JETS, Promondia, FLEXI-BEANS und MultiTEL als Vertreter von Systemen zur Bereitstellung kooperativer Funktionalität auf niedriger Ebene der Kooperationsunterstützung vorgestellt.²⁰⁷

Das an der Syracuse University entwickelte TANGO-System [Beca et al.1997] ist in vielerlei Hinsicht für die Realisierung der angestrebten kooperationsunterstützten Umgebung von Interesse. Die Entwickler setzen erstmals auf eine rein WWW-basierte zentralisierte Architektur. Ein zentraler, mit einer objektorientierten Datenbank verbundener Server kommuniziert mit speziellen, in herkömmliche Webbrowser eingebrachte „daemon Plug-ins“ (als Netscape Plugin bzw. ActiveX Control realisiert). Letztere gewährleisten eine enge Kopplung des Browsers mit dem TANGO-Server. Der TANGO-Server (eine eigenständige Java-Applikation) enthält Komponenten zur Sitzungskontrolle (Session Management), zur Persistenz von den Zuständen der verbundenen Applikationen, zur Kommunikation zwischen TANGO-Applikationen und zur Ereignisverarbeitung (Event-Management). TANGO-Applikationen sind auf einem herkömmlichen WWW-Server abgelegt. Es existieren eine Reihe von Programmierschnittstellen zur Entwicklung von kooperativen TANGO-Applikationen für unterschiedliche Programmiersprachen sowie ein Java-API zur Entwicklung von mit dem TANGO-Server verbundenen Applets.²⁰⁸

Die technische Umsetzung des Systems ist auf den Datenaustausch der in einer Sitzung verbundenen Applikationen ausgerichtet. Ein Event-System erlaubt Java-Applikationen verschiedene Ereignisse, wie z.B. das Betreten einer Session oder das Laden einer Applikation oder Ereignisse aus dem Bereich des Floor Control zu registrieren und darauf mit entsprechenden Aktionen zu reagieren. Zur Kommunikation zwischen den TANGO-Clients (Applets/Applikationen) und dem Java-Server finden keine Mechanismen wie RMI oder CORBA Verwendung. – TANGO basiert nicht auf einer Systemarchitektur von kooperativen verteilten Objekten.

Aus Sicht des verfolgten Ansatzes ist insbesondere die gelungene Browser-seitige Integration des Java-Systems in das WWW zu erwähnen. TANGO-Clients können auf mannigfaltige Weise auf Aktionen des Nutzers in der Browserbedienung reagieren. Weiterhin ist das Java-API zur Entwicklung von TANGO-Clients von hohem Interesse. Es bietet eine leistungsfähige Programmierschnittstelle zur Entwicklung kooperativer Clients.

Analog zu der beschriebenen TANGO-Architektur demonstrieren Gall und Hauck die Entwicklung einer WWW-gestützten kooperativen Architektur in ihrem Promondia-Prototypen [Gall & Hauck 1997]. Der zentrale Java-Server kommuniziert jedoch allein mit in Webseiten eingebundenen Java-Applets. In dieser Form ist Promondia ein

²⁰⁷ Natürlich kann eine derartige Klassifizierung nur willkürlich gewählt sein, sie soll lediglich einer ersten Einordnung aus technischer Sichtweise dienlich sein.

²⁰⁸ TANGO sieht keine speziellen Mechanismen der Dokumentenverwaltung vor. Die Ausrichtung der Systemkonzeption konzentriert sich auf klassische Sitzungsunterstützung in Form von kooperativen Applikationen. TANGO ist insofern ein System zum Datenaustausch zwischen netzgestützten Applikationen und weniger (wie die Entwickler behaupten) ein kooperativer gemeinsamer Informationsraum.

Beispiel für eine reine Java-Lösung aus der Klasse der synchronen Systeme. Wert wurde bei diesem Ansatz auf eine möglichst effiziente Implementierung der zugrunde liegenden Kommunikationsprotokolle zum synchronen Datenaustausch gelegt. Es handelt sich hierbei um ein durchaus typisches Vorgehen bei der Fokussierung auf primär synchrone Szenarien wie die Unterstützung von Shared Whiteboard-Anwendungen.

Einen ähnlichen Ansatz wählen McKinley et al. in der Pavilion-Architektur [McKinley et al. 1999]. Es handelt sich um einen replizierten, Java-basierten Ansatz. Pavilion unterstützt die Entwicklung von kooperativen Java-Applikationen. Zielrichtung ist die Idee des kooperativen Browsers, also die Integration von kooperationsunterstützenden Funktionen in einen WWW-Browser. Die Kommunikation zwischen Pavilion-Server und dem Browser erfolgt anders als z.B. in TANGO über eine DDE-Schnittstelle bzw. einen Proxy-Server.

Seine Stärken beweist Pavilion in der Realisierung eines Multicast-Protokolls [Multicast 2000], d.h. der Bereitstellung von relativ elementaren, aber leistungsfähigen Kommunikationsmöglichkeiten.²⁰⁹ Es handelt sich in dieser Form nicht um ein Framework zur Bereitstellung von Kooperationseigenschaften auf hoher Ebene, die Entwickler fokussieren die Kommunikationsunterstützung zwischen Applikationen im WWW.

In die Klasse der Rahmenarchitekturen zur Kommunikation zwischen Java-Applikationen fallen sowohl JETS [Shirmohammadi et al. 1998] als auch MultiTEL [Fuentes & Troya 2000] oder der FLEXIBEANS [Stiemerling 1999].

FLEXIBEANS ähnelt dem JavaBeans-Ansatz²¹⁰ und erweitert diesen um kooperative Elemente. JavaBeans sind nicht für die Entwicklung von verteilten Applikationen konzipiert. Sie erlauben z.B. nicht die Verarbeitung von Ereignissen auf verschiedenen Netzwerkschnittstellen. Als verteilte Client-Client-Applikation bietet FLEXIBEANS so genannte „named ports“, „shared objects“ und „remote interaction“. Hierbei werden JavaBeans um die Fähigkeiten einer auf verschiedenen Kanälen arbeitenden Ereignisbehandlung, um gemeinsam genutzte Objekte (Shared Objects) und verteilte Methodenaufrufe (RMI) erweitert.²¹¹

Die JETS-Umgebung ist ein Beispiel für einen zentralisierten Ansatz zur Kooperation/Kommunikation zwischen Java-Applets. Konzipiert als klassische Client-Server-Architektur fokussieren die Autoren eine Unterstützung von kooperativen multimedialen Applikationen (Shared Whiteboard, Shared Video etc.). Interessant ist die Umsetzung einer parallelen Ereignisverteilung (Event-Multicasting) ausgehend von dem Server zu den einzelnen in einer Sitzung synchronisierten Applikationen. Damit ist JETS ein Beispiel für die Adaption von Multicast-Technologie auf Event-Ebene. – Hierbei handelt es sich sicherlich auch für die Umsetzung des angestrebten Ereigniskonzepts um eine interessante Perspektive.

MultiTEL demonstriert Verfahren der dynamischen Erweiterung einer verteilten Applikation. Es handelt sich in dieser Form um ein „middleware“-Framework für die

²⁰⁹ Im Hinblick auf die sTeam-Architektur erscheint eine Verwendung von Multicast-Protokollen zum gegebenen Zeitpunkt nicht zwingend notwendig. Multicast-Protokolle eignen sich insbesondere im Umfeld stark synchroner Anwendungen, wo eine große Anzahl von relativ kleinen Informationen umzuverteilen sind (z.B. Aktualisierungen, Aktionen auf einem Shared Whiteboard).

²¹⁰ Vgl. <http://java.sun.com/products/javabeans/resources.html>, Stand 1.12.2001.

²¹¹ Diese Vorgehensweise entspricht dem System von Banavar et al. (vgl. [Banavar et al. 1998]). Eine synchrone Anwendung wird durch kooperative JavaBeans gestaltet, welche grafisch arrangiert und verknüpft werden können.

Unterstützung von Kommunikation zwischen in Java realisierten Applikationen. Die Entwickler verfolgen einen konsequent komponentenbasierten Ansatz. Komponenten kommunizieren über so genannte „connector“-Elemente, die ein Protokoll zwischen den Komponenten implementieren und zugehörige Ereignisse (Events) verarbeiten.²¹² Multi-TEL unterstützt als relativ modernes Framework ähnlich wie DyCE [Tietze & Steinmetz 2000] das dynamische Binden von Komponenten zu existierenden Applikationen.

Auf der Schwelle zur Kooperationsunterstützung auf hoher Ebene ist das DISCIPLINE-Framework [Li et al. 1999] anzusiedeln. Es erlaubt existierende Java-Beans mit eng gekoppeltem, synchronem Verhalten auszustatten. Dazu wird eine Java-Swing GlassPane²¹³ über die Applikation gelegt. Diese erlaubt es alle Interaktionen eines Nutzers an der entsprechenden Komponente zu verarbeiten und an die Kooperationspartner umzuverteilen. DISCIPLINE arbeitet in dieser Form wie ein „collaboration bus“ [ebd., S. 328].²¹⁴

Damit dokumentieren sowohl MultiTEL als auch JETS, obwohl es sich bei beiden nicht um wirkliche CSCW- oder CSCL-Frameworks wie COAST oder DyCE handelt, innovative Technologien der Kommunikation von verteilten kooperativen Applikationen. Speziell MultiTEL bietet ähnlich dem sTeam-Java-Client eine komponentenbasierte Architektur zur dynamischen, clientseitigen Erweiterung um kooperative Komponenten.

Java-basierte Frameworks auf hoher Ebene der Kommunikationsunterstützung

Eine Klassifizierung existierender Frameworks nach dem Grad der Kooperationsunterstützung auf niedriger Kommunikationsebene bzw. hoher Ebene in der Bereitstellung kooperativer Objekte oder Komponenten ist in jedem Fall nicht eindeutig. Im Folgenden seien insbesondere Habanero²¹⁵ und DyCE als prototypische Vertreter der Klasse der Frameworks auf hoher Ebene der Kommunikationsunterstützung definiert.²¹⁶

Von der konzeptionellen Ausrichtung ähnelt Habanero dem Pavilion-System; dient demnach ebenfalls der Bereitstellung von Kommunikationsmechanismen für Java-Applets oder Java-Applikationen. Habanero besitzt eine Client-Server-Architektur, also einen zentralen Server, der den zentralen Datenraum bereitstellt und die Ereignisverwaltung (Event-Management) zwischen den einzelnen kooperativen Applikationen übernimmt. Kooperative Applikationen, die so genannten „Habanero-Tools“ oder auch „Hablets“, bilden die Clients des Habanero-Systems. Diese werden in eine Rahmenap-

²¹² Die technische Konzeption des sTeam-Clients verfolgt einen ähnlichen Ansatz. Auch hier hält ein so genannter sTeam-Connector [Enklaar 2001, S. 26ff.] die Kommunikation zum Server aufrecht (vgl. Abschnitt 5.2.8).

²¹³ Bei der Java-Swing-GlassPane handelt es sich um ein spezielles Oberflächenelement, welches unsichtbar für den Nutzer Interaktionen mit herkömmlichen unterliegenden Oberflächenelementen, wie Buttons etc., abzufangen und ggf. zu verarbeiten vermag.

²¹⁴ In seiner Ausrichtung auf stark synchrone Anwendungsfälle und dem Anspruch „collaboration unaware“-Applikationen mit kooperativen Eigenschaften ausstatten zu wollen, grenzt sich DISCIPLINE von der sTeam-Konzeption ab.

²¹⁵ Habanero [Chabert et al. 1998a] erhebt den Anspruch eines Frameworks für die Entwicklung kooperativer Web-Applikationen. Es wird am National Center for Supercomputing Applications (NCSCA) in den USA entwickelt.

²¹⁶ COAST besitzt in seiner objektorientierten Architektur effektive kooperative Komponenten auf hoher Ebene, wird jedoch aufgrund seiner Realisierung in Smalltalk an dieser Stelle nicht detaillierter durchleuchtet.

pplikation eingebracht, das „Habanero User Environment“, welches eine Reihe von für alle Hablets identische Funktionen bereitstellt. Benutzer können z.B. eine Sitzung initiieren oder einer bestehenden Sitzung beitreten.

Habanero unterstützt sehr elegant die Portierung existierender Applikationen oder Applets in kooperative „Hablets“.²¹⁷ Hierbei ist die Gradualität der verarbeiteten Ereignisse frei wählbar. Es lassen sich auf diese Weise hochgradig synchrone Applikationen wie z.B. Whiteboards mit gemeinsamen Mauszeigern entwickeln, aber auch gering gekoppelte Applikationen realisieren, welche lediglich über gemeinsame Datenräume verfügen. Wichtiges Unterscheidungsmerkmal gegenüber anderen Ansätzen ist hierbei, dass Habanero keine Replikation von Objekten zwischen dem Habanero-Server und den kooperierenden Applikationen vornimmt. – Sämtliche Kommunikationen sowie der Datenaustausch sind über entsprechende Java-Interfaces abzuwickeln. Des Weiteren besitzt Habanero als rein sitzungsbasierter Ansatz keinerlei Möglichkeiten, gemeinsame Inhalte einer Session ausgenommen durch Speicherung auf den Clients persistent abzulegen – DyCE bietet hier eine leistungsfähige Persistenzschicht für die verwalteten Objekte.

Insofern handelt es sich zwar um ein spezielles Framework für die Entwicklung von synchronen Applikationen und bietet spezielle Mechanismen zur Sitzungsunterstützung, wie etwa Formen von Awareness, erlaubt jedoch keinerlei Ansätze der Persistenz von gemeinsamen Objekten.

Im Hinblick auf die technische Konzeption der entwickelten Architektur sind insbesondere die Fähigkeiten von Habanero (Habanero-Framework) von Interesse, die es erlauben mit einfachen Mitteln kooperative, synchrone Applets gestalten zu können. Hier ergeben sich Überschneidungspunkte zur Integration von synchronen Werkzeugen in virtuelle Räume (z.B. die Einbindung eines Shared Whiteboards).

In den genannten Punkten der Konstruktion von CSCL-/CSCW-Applikationen aus kooperativen Komponenten geht die DyCE-Architektur („Dynamic Collaboration Environment“) [Tietze & Steinmetz 2000] um einiges weiter. Als Framework für die Entwicklung von kooperativen Werkzeugen wird es seit einiger Zeit an der Fraunhofer Gesellschaft als Nachfolger des COAST-Systems entwickelt. Zielrichtung ist der Aufbau einer durchgängig Java-basierten Client-Server-Architektur, welche die Entwicklung synchroner Groupware-Applikationen unterstützt.

Wesentliches architektonisches Merkmal von DyCE sind die so genannten „groupware components“ als kooperative Komponenten zur Konstruktion von CSCW-Applikationen. Als verteilter Ansatz mit einem ausgewiesenen Server werden Daten und Zustände der Komponenten durch Replikation auf den jeweiligen Clients konsistent gehalten. Zentrales Instrument der synchronen Kommunikation von Java-Objekten innerhalb der DyCE-Architektur ist die Java-RMI. Asynchrone Kommunikation findet in Transaktionen gekapselt über eine eigene Netzwerkschicht statt.

Auch die gemeinsamen Daten-Objekte einer DyCE-Applikation werden durch Replikation auf die Clients verteilt. Hierbei werden jedoch nur die von dem jeweiligen Client aktuell genutzten Objekte repliziert (semi-replizierte Architektur). Objekte werden durch so genannte „Object Manager“ persistent gespeichert und durch „Transaktionsmanager“ auf die jeweiligen Clients umverteilt. Erfolgt eine Änderung an dem Mo-

²¹⁷ Hierzu ist z.B. im Falle eines Applets die Vererbung nicht von der Klasse „applet“, sondern von der Klasse „hablet“ vorzunehmen. Das neue Applet enthält sämtliche Mechanismen zur Kommunikation mit dem Habanero-Server.

dell²¹⁸, werden alle betroffenen Objekte auf anderen Clients benachrichtigt und aktualisiert.

DyCE erlaubt ähnlich der im Rahmen dieser Arbeit dargestellten Konzeption neben der flexiblen Kombination von unterschiedlichen Komponenten auch eine Erweiterung, d.h. eine Instanziierung von Komponenten in einer Gruppensitzung. Dieser Prozess des dynamischen Hinzufügens von Komponenten wird durch den so genannten „Component Broker“ vorgenommen.

Wichtiges Merkmal zur Abgrenzung gegenüber anderen Ansätzen ist, dass DyCE Groupware-Components speziell als interaktive Komponenten versteht, die ein eigenes Datenmodell sowie zumeist eine grafische Darstellung besitzen (vgl. [Tietze & Steinmetz 2000, S. 51]). Sie erzeugen eine konsistente Sicht auf kooperative Daten und erlauben es diese zu manipulieren. Damit verwirklicht DyCE auf höchster Ebene die komponentenbasierte Entwicklung von CSCW-/CSCL-Systemen.

Neben den eigentlichen Komponenten stellt DyCE einen so genannten „Groupware Desktop“ bereit, um verschiedene Komponenten zu verwalten und flexibel zuzuordnen.

Die Systemarchitektur des DyCE-Ansatzes ist Client-Server-orientiert, wobei es die Aufgaben des Servers sind, die Verwaltung der Objekte vorzunehmen, eine der Konsistenzen von Operationen durch Kapselung der Aktionen der Clients in Transaktionen sicherzustellen (Transaktionsvalidierung), die Verwaltung und Umverteilung der Komponenten auf die Clients zu koordinieren sowie die Sitzung an sich zu organisieren (Session Management).²¹⁹ Architektonische Ähnlichkeiten weist der DyCE-Ansatz insbesondere zu den TeamComponents (vgl. [Roth & Unger 1998] und [Roth & Unger 2000]) auf.²²⁰

Zusammenfassend ist festzuhalten, dass DyCE in Bezug auf sein Modell kooperativer Objekte architektonische Ähnlichkeiten zu dem im Rahmen dieser Arbeit konzipierten Framework kooperativer Wissensräume aufweist. DyCE ist vollständig Java-basiert, an seine Architektur werden ähnliche Anforderungen in Form von Komponentenbasertheit und dynamischer Erweiterbarkeit gestellt. Damit demonstriert DyCE weite Teile der Möglichkeiten ebenfalls synchroner sTeam-Clients. In den Bereichen der Dokumentenverwaltung hebt sich das entwickelte Rahmenkonzept jedoch weitgehend von DyCE ab. DyCE bietet nur rudimentäre Fähigkeiten der Dokumentenverwaltung und z.B. keinerlei Mechanismen der Integration von Hypertextdokumenten (z.B. keinerlei konsistente Verweisstrukturen etc.). Auch eine konzeptionelle Nähe zum WWW wie etwa die Zugreifbarkeit von in DyCE abgelegten Dokumenten über einen Browser ist nicht vorhanden.²²¹

DyCE eignet sich in der derzeitigen Form weniger für den Aufbau offener kooperativer Lernumgebungen als Teil des WWW, es eignet sich vielmehr für die Unterstützung

²¹⁸ Ein Modellobjekt ist eine Kapsel für kooperativ genutzte Objekte innerhalb von DyCE, es besitzt standardisierte Zugriffsmethoden auf die Daten des Objekts.

²¹⁹ Als Beispielapplikation für die Validierung der implementierten DyCE-Komponenten wurde von der GMD ein „kooperativer Hypermedia Workspace“ (vgl. [Tietze & Steinmetz 2000, S. 57]) realisiert. Für verschiedene Anwendungen der DyCE-Umgebung vgl. [Haake 1999].

²²⁰ Es wird ein konsequent dezentraler, also replizierter Ansatz verfolgt. Im Gegensatz zu einer zentralisierten Architektur existiert kein zentraler Gruppenserver. Informationen können zwischen den kooperativen Applikationen serialisiert übertragen werden. Eine zentrale Persistenzschicht ist nicht vorhanden. Für die Übertragung der Daten zwischen den Shared-Applications findet eine spezielle Multicast-Technologie Verwendung.

²²¹ DyCE bietet eine HTTP-Unterstützung, diese wird jedoch primär zum Nachladen von Modulen genutzt.

sehr gezielter Kooperationssituationen auf Basis einzelner Werkzeuge. In diesem Punkt demonstriert es technisch Machbares und brilliert durch eine moderne objektorientierte Architektur.

4.2.4 Webserver-basierte Architekturkonzepte

Im Hinblick auf die Zielsetzung des vorgestellten Rahmenkonzepts als Synthese aus Fähigkeiten des Dokumentenmanagements und synchronen kooperativen Elementen in der Metapher kooperativer Wissensräume ist eine Betrachtung von auf Webserver-Technologie basierenden Ansätzen unumgänglich.

Eine ganze Reihe von Ansätzen erweitern herkömmliche Webserver um gewisse Anteile kooperationsunterstützender Funktionalität. Hierzu werden unterschiedliche Dienste wie E-Mail, News oder Chat zusammen mit Mechanismen der Dokumentenverwaltung und der gegenseitigen Wahrnehmung in einen Webserver integriert und über WWW-Zugangswerkzeuge (Browser) zugreifbar gemacht.²²²

Trotz eines nur geringen Anteiles an synchronen Elementen sei als typischer Vertreter dieser Klasse das BSCW-System (vgl. [Bentley et al. 1997] und [Appelt et al. 1998]) architektonisch beleuchtet. BSCW erweitert einen herkömmlichen Webserver um eine Benutzer- und Gruppenverwaltung sowie so genannte kooperative Arbeitsbereiche. Diese werden architektonisch durch eine Reihe von an den Webserver angefügten Skripten bereitgestellt. Daten werden direkt auf dem Webserver abgelegt, es ist keine hiervon unabhängige Persistenzschicht notwendig. Im Vergleich zu den im Vorfeld betrachteten Ansätzen existiert in dieser Form kein eigenständiger kooperationsunterstützender Server.

BSCW ist sicherlich kein klassisches Framework für die Gestaltung kooperativer Umgebungen, besitzt jedoch durch seine Architektur gute Voraussetzungen der nutzerseitigen Konfiguration und Erweiterbarkeit. Dies liegt insbesondere an seiner Grundstruktur des um eine Reihe von Python-Skripten erweiterten Webservers. In der auf diese Weise erzielten Stabilität und der nutzerseitigen Erweiterbarkeit begründet sich sicherlich ein guter Teil des Erfolges des BSCW-Systems. Erfahrungen der Entwickler belegen in diesem Zusammenhang, dass eine große Anzahl von Modifikationen, z.B. die Übersetzung der Benutzerschnittstelle in verschiedene Sprachen, von den Anwendern selbst vorgenommen wurden (vgl. [Appelt et al. 1998, Abschnitt 4.1]).

Einschränkungen ergeben sich in diesem Zusammenhang in der fehlenden Kopplung von Objekten an Nutzer und den von ihnen genutzten Rechnersystemen. So schlagen sich Änderungen an dem Frontend und speziell an dessen Funktionalität auf alle Nutzer eines Servers nieder. Die Entwickler des BSCW-Systems reduzieren dieses Problem durch vorgefertigte Benutzerschnittstellen für „Anfänger“, „Fortgeschrittene“ und „Experten“ sowie die Möglichkeit, ein Benutzerprofil zu erstellen. In einem derartigen Benutzerprofil lassen sich das Erscheinungsbild und der Funktionsumfang der Benutzerschnittstelle spezifizieren.

Bedingt durch seine enge Anlehnung an die übliche Realisierung eines WWW-Servers ergeben sich primär asynchrone Nutzungsszenarien des BSCW-Systems. Die zunächst verbindungslose Eigenschaft des HTTP-Protokolls – Browser Requests sind voneinander unabhängig, Statusinformationen lassen sich lediglich in begrenztem Umfang

²²² Die im Rahmen dieser Arbeit vorgestellte Architektur weist in seiner Grundstruktur als primär asynchrones System mit engen Verbindungen zum WWW auf den ersten Blick eine gewisse Ähnlichkeit zu dieser Klasse von Systemen auf. Es grenzt sich jedoch in seiner Feinstruktur als Verbund kooperativer, ereignisgesteuerter Objekte erheblich von derartigen Herangehensweisen ab.

über Cookies²²³ herstellen – erlaubt weder leistungsfähige Awareness-Mechanismen noch eine Integration von synchronen Kommunikationsmechanismen.²²⁴

Die nutzerseitige Anpassbarkeit von Arbeitsbereichen fokussieren zudem die frühen Forschungen von Bentley und Dourish, die in ihrem Prototypen eines Clients für das BSCW-System den Begriff der „incremental customisation“ verwenden [Bentley & Dourish 1995].²²⁵

Die architektonischen Stärken von BSCW liegen in seiner Einfachheit und engen Anknüpfung an herkömmliche WWW-Technologie begründet. Kooperativ genutzte Dokumente werden über die Arbeitsbereiche auf dem Webserver abgelegt und sind über einen herkömmlichen Browser zugreifbar.

Nachteilig erweist sich diese Einfachheit in Bezug auf die Integration von Awareness-Informationen, der Bereitstellung einer unmittelbaren Rückmeldung von Aktivitäten der Benutzer und der Integration synchroner Kommunikationsformen.

Im architektonischen Vergleich zu CSCW-/CSCL-Werkzeugen wie DyCE, COAST oder auch der im Rahmen dieser Arbeit vorgestellten Konzeption lassen sich zudem Schwächen im Umgang mit kooperativen Objekten ausmachen. So besitzt BSCW im Gegensatz zu den genannten Ansätzen keine durchgängig objektorientierte Architektur, in der gemeinsam genutzte Objekte repliziert werden oder Benachrichtigungsmechanismen für Modifikationen an diesen existieren.

Im direkten Vergleich zum verfolgten Ansatz existiert zudem keine logische Entkopplung der Struktur von Hypertextdokumenten und ihrer Speicherung auf dem Webserver (fehlende Konsistenz von eingebetteten Verweisen). Damit lassen sich Dokumente auf einem BSCW-Server nur schwer in wirkliche kooperative Prozesse einbinden (z.B. individuell arrangieren).

4.2.5 MUD- und MOO-basierte Architekturkonzepte

Das im Weiteren skizzierte Rahmenkonzept weist in seiner Kernarchitektur enge Parallelen und Analogien zum Bereich der MUDs und MOOs auf. Aus diesem Grunde erscheint speziell ein Blick auf existierende MUD-Umsetzungen und damit auf die unmittelbaren technischen Vorläufer des sTeam-Ansatzes (HyperMUD) [Bollmeyer 1997] sinnvoll.

MUD-Architektur: Driver und MUDLIB

Von Seiten der Architektur betrachtet handelt es sich bei einem MUD um ein spezielles Programm, welches Netzwerkverbindungen aufbaut und verwaltet und auf diese Weise Personen erlaubt, mit Objekten in einer virtuellen Welt zu interagieren. Über einfache, meist textuelle Kommandos und Kommunikations- sowie Awareness-Mechanismen (Avatare) bewegen sich die Nutzer in Gruppen durch die virtuelle Welt. In seiner Grundstruktur verarbeitet das MUD die eintreffenden Kommandos eines Nutzers, modifiziert entsprechend die vorhandenen Objekte und die Raumstruktur sowie informiert andere Nutzer über eintretende Ereignisse.

²²³ Vgl. http://home.netscape.com/legal_notices/cookies.html, Stand 1.12.2001.

²²⁴ Eine ganze Anzahl von Ansätzen versucht die Architektur des BSCW-Systems im Hinblick auf die oben genannten Einschränkungen zu erweitern. Hierzu zählt der an der GMD zeitweise entwickelte BSCW-Java-Client (die Entwicklungen sind nach mündlicher Auskunft der Entwickler zurzeit eingestellt), eine Awareness-Visualisierung [Prinz 1999] oder eine Kopplung von BSCW mit der virtuellen Umgebung Active Worlds zur Umsetzung von virtuellen Räumen [Huxor 1998].

²²⁵ Ziel ist es, Benutzern eines gemeinsamen Arbeitsbereichs zu erlauben, flexibel die Darstellung und Interaktionsmöglichkeiten anzupassen.

Das spezifische Erscheinungsbild eines MUD wird durch die MUDLIB geprägt. Sie enthält den anwendungsspezifischen Code für Räume, Spieler bzw. Avatare, Objekte in der virtuellen Welt und aktive Elemente des MUD, so genannte „Non Player Characters“ (NPCs), gleichbedeutend Elementen des MUD, die nicht von einem Spieler gesteuert werden.

In frühen MUDs ist eine MUDLIB zumeist in C implementiert. Die Implementierungen der einzelnen Objekte ist fest und eng mit den systemspezifischen Funktionen des MUD verknüpft.

Eine ganze Reihe von späteren MUDs nehmen in der MUDLIB eine mehr oder weniger strikte Trennung von Systemfunktionalitäten (der Laufzeitumgebung) in Form eines so genannten Drivers und inhaltlichen Komponenten des MUD vor.

Driver können damit als eine rudimentäre Laufzeitumgebung eines MUD charakterisiert werden. Als wesentliche Aufgabe bieten sie den technologischen Rahmen für die MUDLIB, stellen in dieser Form eine Persistenzschicht bereit und interpretieren eintreffende Kommandos aus entsprechenden Netzpaketen. Die MUDLIB bildet die Semantik des Spiels entlang festgeschriebener Regeln, den „Spielregeln“ des MUD.

Innovativer Schritt zu einer einfachen Entwicklung von speziellen Ausprägungen eines MUD ist die Realisierung der MUDLIB in einer speziell interpretierten Programmiersprache und die Integration eines entsprechenden Interpreters in den Driver des MUD.²²⁶

Eine Vielzahl sehr unterschiedlicher Einsatzkontexte der MOO-Technologie beweist, dass sich seine konsequent objektorientierte Technologie sehr gut für den Aufbau von synchronen Kommunikations- und Interaktionsumgebungen eignet.

Bemerkenswert ist, dass im Umfeld der architektonischen Weiterentwicklung des MUD schon früh Anstrengungen unternommen wurden, die MUD-Technologie um Anteile des Dokumentenaustausches zu erweitern.²²⁷

Interpretierte MUDs – LPMUD

Den Prozess der Integration von speziellen Programmiersprachen in das MUD hat Pensjö mit seinem seit Mitte der 80er Jahre verfügbaren LPMUD²²⁸ maßgeblich geprägt. Der Driver eines LPMUD besitzt im Wesentlichen eine Laufzeitumgebung und einen Interpreter für LPC (Lars Pensjö C), eine interpretierte Variante von objektorientiertem C. In den letzten Jahren sind neben dem von Pensjö entworfenen und von Amylaar weiterentwickelten LPMUD-Driver eine ganze Reihe weiterer MUD-Driver entstanden, welche sich auf LPC stützen. Beispiele sind der CD-Driver aus dem Genesis-Projekt

²²⁶ Damit entstanden eine ganze Reihe von Strömungen und Weiterentwicklungen des ursprünglichen AberMUD. Hierzu zählen TinyMUD, welches zunächst keine eigene, interne Programmiersprache enthält, TinyMUCK, TinyMOO und DikuMUD, welches erst in C entwickelt und später mit einer internen Programmiersprache ausgestattet wurde. Weitere populäre MUD-Typen sind MUSH, TinyMUSH, welches eine LISP-ähnliche interne Programmiersprache aufweist, und natürlich das MOO.

Für eine umfangreiche Aufstellung verschiedener MUD-Systeme vergleiche Abschnitt 1.7.1 und [Bartle 1990b, S. 71ff.].

²²⁷ Ein besonders herausragendes Beispiel ist die von Masinter und Ostrom realisierte Verknüpfung eines MUD mit dem Gopher-System (vgl. [Masinter & Ostrom 1993]). Hier werden schon sehr früh die kommunikationsunterstützenden Aspekte des MUD mit den Ideen des Internet-basierten Hypertext verknüpft.

²²⁸ Vgl. <http://www.neosoft.com/genesis/>, Stand 1.12.2001.

oder Pike und DGD, bei denen es sich um vollständig eigenständige Entwicklungen handelt.

Der von Croes entwickelte DGD (Dworkins' Generic Driver)²²⁹ gilt als besonders effiziente und performante Implementierung von LPC und weist leistungsfähige Mechanismen der Erweiterbarkeit einer MUDLIB zur Laufzeit des Systems auf. Die ersten Vorläufer des sTeam-Systems (vgl. [Bollmeyer 1997], [Hampel 1999], [Bopp 2000]) wurden auf Basis des nur kommerziell verfügbaren DGD entwickelt, später jedoch aufgrund des Anspruchs der Entwicklung einer Open Source-Umgebung verworfen.

Neben DGD ist Pike²³⁰ die bekannteste LPC-Implementierung. Als Open Source-Produkt genießt es durch seine große Anzahl verfügbarer Bibliotheken (Beispiele sind Module zur Datenbankanbindung, Bildverarbeitung, Kryptographie etc.) den Ruf einer nicht alleinig für MUD-Anwendungen geeigneten Umgebung, sondern als universelle Sprache. So nutzt beispielsweise der Roxen-Webserver Pike und LPC als Implementierungsbasis (vgl. Abschnitt 5.1).²³¹

MOO-Architektur

Trotz ihrer für kooperationsunterstützende Systeme langen Tradition weisen MOOs eine erstaunlich moderne Architektur auf. Diese wurde Anfang der 90er Jahre von White basierend auf dem TinyMUD entworfen. Kurze Zeit später wurde die Entwicklung von Curtis in seinem LambdaMOO Projekt (vgl. [Curtis 1992] und [Carlstrom 1992]) fortgesetzt.

Wesentliche Komponenten eines MOO sind der eigentliche Server und eine integrierte Datenbank. Der Server verarbeitet, wie in einem gewöhnlichen MUD, Ein- und Ausgabeoperationen, stellt Netzwerkverbindungen bereit und implementiert eine interne Programmiersprache. Die neue Qualität gegenüber herkömmlichen MUD-Architekturen liegt in der Beschreibung des MOO (seiner Objekte) in einer integrierten Datenbank. Jedes Objekt ist als ableitbare Klasse in der Datenbank spezifiziert und persistent abgelegt. Funktionen eines MOO werden „verbs“ genannt und erlauben Operationen auf den verschiedenen Objekten. In dieser Form sind auch die Kommandos von Nutzern als „verbs“ realisiert und in der Datenbank abgelegt. Objekte besitzen flexible Attribute, so genannte „Properties“, welche die wesentlichen Merkmale eines Objektes beschreiben.

Mit der Datenbank-zentrierten Architektur ist eines der vorrangig verfolgten Ziele von Curtis erreicht: Es ist die einfache, nutzerseitige Erweiterbarkeit oder besser Programmierbarkeit einer kooperationsunterstützenden Umgebung.

Neben dem reinen Einsatz in unterschiedlichen Anwendungskontexten (vgl. Abschnitt 1.7.2) finden sich eine ganze Anzahl von Bestrebungen auf Basis des populären LambdaMOO-Systems, neue kooperationsunterstützende Architekturen zu entwerfen. Hierzu zählen Ansätze, LambdaMOO durch spezielle Clients und Erweiterungen von Protokollen als Lernsystem einzusetzen. Ein Beispiel ist das frei verfügbare enCore-System²³² [Haynes & Holmevik 1998] der Universität Dallas, Texas, USA. Basierend auf enCore entwickeln Hill und Slator kooperative netzgestützte Lernsysteme [Hill & Slator 2000] (Entwicklung des WWW-basierten Clients enCore Xpress).

²²⁹ Vgl. <http://dworkin.nl/dgd>, Stand 1.12.2001.

²³⁰ Vgl. <http://pike.idonex.se>, Stand 1.12.2001.

²³¹ Aktuelle Prototypen der sTeam-Architektur (vgl. [Hampel & Keil-Slawik 2001a]) sind fast vollständig in Pike (LPC) umgesetzt. Ihre Implementierung nutzt eine ganze Anzahl vorhandener Pike-Module, z.B. das SQL-Datenbankmodul zur Anbindung an eine relationale Datenbank oder einen XML-Parser.

²³² Vgl. <http://lingua.utdallas.edu/encore/home.html>, Stand 1.12.2001.

Sowohl aus konzeptioneller (vgl. Abschnitt 2.7.3) als auch aus architektonischer Sicht ist der Collaborative Virtual Workspace (CVW) [Spellman et al. 1997] für die Realisierung der angestrebten Architektur von großem Interesse.²³³

CVW setzt auf LambdaMOO auf und integriert einen Webserver und einen Dokumentenserver in die synchrone MOO-Umgebung. Clients sind für verschiedene Plattformen verfügbar (Tcl/Tk und Java) und kommunizieren mit dem Server über ein HTTP-basiertes einfaches Protokoll (MOO-Client-Protocol – MCP).

Spezielles Augenmerk wurde bei der Konzeption der CVW-Architektur auf die Integration synchroner Werkzeuge und Kommunikationsformen in das MOO-System gelegt. Audio- und Videokommunikation ist genauso vorhanden wie ein synchrones Shared Whiteboard.

Die im Rahmen der vorliegenden Arbeit entwickelte Kernarchitektur grenzt sich trotz auf den ersten Blick großer architektonischer Ähnlichkeiten in einer ganzen Reihe von entscheidenden Punkten von CVW ab.²³⁴ Zu einem großen Teil resultieren die aufgezeigten Unterschiede aus der engen Anlehnung des CVW an herkömmliche MOO-Technologie. In diesem Zusammenhang sind folgende Punkte zu nennen:

- *Registrierung neuer Klassen:* CVW-Objekte sind resultierend aus dem MOO-Ansatz auf den Funktionsumfang des MOO begrenzt. Erweiterte Eigenschaften, wie neue Klassen und Attribute, speziell zusätzliche Attributtypen sind in dieser Form nicht in das System integrierbar. Die sTeam-Architektur bietet in diesem Punkt einen speziellen Mechanismus der Registrierung von neuen Attributen in Factories (vgl. Abschnitt 5.2.2).
- *Aktualisierung laufender Instanzen:* Instanzen von CVW-Objekten können nicht zur Laufzeit des Systems durch eine neue Version des Objekts/der Klasse ersetzt werden. Ein derartiger Mechanismus ist für die dynamische Erweiterung bzw. Aktualisierung des Systems (z.B. das Einbringen neuer Objekte in die Umgebung) von großer Wichtigkeit. Der DGD war als erste MUD-Architektur mit derartigen Eigenschaften ausgestattet. Die sTeam-Architektur sieht ebenfalls die Möglichkeit vor, laufende Instanzen von Klassen durch neue Versionen dynamisch zu ersetzen.
- *Multimediale Dokumententypen:* CVW implementiert eine ganze Reihe multimedialer Datentypen. Trotz seiner Ausrichtung als Dokumentenmanagementsystem existiert historisch bedingt keine vollständige Unterstützung von Hypertext- bzw. WWW-basierten Dokumententypen. Die MOO-Architektur erlaubt beispielsweise nur schlecht die Implementierung aufwändiger Konvertierungs- und Parser-Routinen zur dynamischen Extraktion und Ersetzung von Verweisen aus HTML-Dokumenten. Das in der sTeam-Architektur genutzte Pike-Modulkonzept bietet performante Möglichkeiten, derartige Routinen als eigenständige Prozesse in einer nicht-interpretierten Programmiersprache zu implementieren. Basierend auf einer dynamischen Ersetzung von Verweisen ergeben sich Möglichkeiten, WWW-gestützte HTML-/XML-Dokumententypen als vollwertige kooperative Dokumente nutzbar zu machen (Möglichkeiten des Arrangierens und Übertragens).

²³³ CVW wurde von der Mitre Cooperation im Rahmen eines Forschungsprojektes prototypisch konzipiert und ist seit kurzem in eine Open Source-Entwicklung überführt, vgl. <http://cvw.sourceforge.net/>, Stand 1.12.2001.

²³⁴ Zu einer ersten Abgrenzung sind im Vorgriff auf Kapitel 5 einige Unterschiede und Ähnlichkeiten aufgezeigt.

- *Clients*: CVW demonstriert eine langjährige Erfahrung in der Entwicklung von Clients für unterschiedliche Plattformen. Sowohl der Java-Client als auch der Tcl/Tk-Client sind ausgereift und bieten sowohl synchrone als auch asynchrone Interaktionsformen. Architektonisch werden Konzepte des MOO, wie das Inventory, der persönliche Rucksack eines Nutzers, in die Architektur des Clients eingebunden. Beispielsweise ist das Einfügen von beliebigen Dokumenten metaphorisch über den Rucksack eines Nutzers umgesetzt. Ein Client realisiert lediglich eine Sicht auf den sonst nur textuell beschriebenen virtuellen Raum. Hierbei ist im Falle des CVW-Systems die architektonische Kopplung aus klassischem MOO und dem Client als Sicht auf den virtuellen Raum sehr eng. Ein CVW-Client bildet im Wesentlichen alle Kommandos eines MOO (Funktionen der Kommunikation und des Umganges mit Dokumenten) in eine grafische Benutzerschnittstelle ab. Dies zeigt sich speziell durch die Möglichkeit, CVW weiterhin durch Kommandos in Textform zu bedienen. Die vorgeschlagene Konzeption kooperativer Wissensräume sieht im Gegensatz hierzu eine weitaus abstraktere Darstellung virtueller Räume vor.
- *Nutzerrechte*: MUDs und MOOs weisen von Haus aus keine ausgefeilten Systeme aus Gruppen- und Nutzerrechten auf. Dies liegt zum großen Teil an der fehlenden Notwendigkeit zur Dokumentenverwaltung. Berechtigungen innerhalb eines MUD oder MOO sind zumeist an den Status eines Nutzers, wie Spieler, Wizard etc., geknüpft und erstrecken sich auf globale Fähigkeiten der Modifikation und Interaktion in der virtuellen Welt. Dahingehend wurden innerhalb des CVW zwar Möglichkeiten der Spezifikation von Nutzerrechten vorgesehen, die Fähigkeiten der Selbstadministration sind jedoch bedingt aus der MOO-Architektur nur sehr rudimentär ausgebildet.
- *Erweiterbarkeit/Programmierbarkeit*: Resultierend aus dem Anwendungsszenario eines MOO zur Realisierung von textuellen, virtuellen Welten sind die Möglichkeiten der Programmierung in einem MOO sehr begrenzt. MUD-Driver wie DGD oder insbesondere Pike, welche keine primäre Ausrichtung auf klassische Rollenspiel- oder Kommunikationsumgebungen aufweisen, besitzen hier erweiterte Funktionalitäten. Der sTeam-Ansatz profitiert in seiner Architektur von den Möglichkeiten der recht universellen Programmiersprache LPC und ist in sich gut strukturierbar und erweiterbar. Objekte können von autorisierten Benutzern erstellt und in das System eingebracht werden.

Zusammenfassend ist festzuhalten, dass CVW durch seinen Anspruch, Dokumentenmanagement mit den Ideen eines MOO systematisch zu verknüpfen, eine ausgesprochene Nähe zu der im Rahmen dieser Arbeit entwickelten Konzeption aufweist, aber erhebliche Einschränkungen in den Bereichen der Nutzerrechte und Selbstadministration der Lernumgebung aufweist.

Auch die Integration verschiedener Materialien und konzeptuelle Nähe zum WWW (beispielsweise im Umgang mit Hypertextdokumenten) sind nicht entsprechend weit entwickelt. In der technisch-architektonischen Ausprägung präsentiert sich CVW historisch bedingt als primär eigenständiges System. Zudem erlaubt es CVW nur bedingt beliebige semantische Strukturen (der Materialien und Räume) aufzubauen. Im Vergleich zu Systemen wie COAST oder DyCE dokumentiert CVW den Anspruch, weniger Synchronizität einer kooperationsunterstützenden Umgebung auf der Ebene von gekoppelten Oberflächenelementen und sehr speziell ausgerichteten Werkzeugen bereitzustellen. Vielmehr macht es sich die aus der Sparte der MUDs und MOOs erprobten Ideen des

virtuellen gemeinsamen Raums zunutze. Architektonisch wird dieses Ziel durch ein Konzept von kooperativen Objekten umgesetzt, was eine im Vergleich zu herkömmlichen CSCW-Architekturen aufwändige Konstruktion des Servers notwendig macht.

4.2.6 Experimentelle Ansätze, Prototypen der Forschung

Es existieren eine ganze Reihe experimenteller Implementierungen von CSCW-Architekturen. Sie weisen sehr unterschiedliche Herangehensweisen auf und besitzen im Hinblick auf die Bereitstellung kooperativer Medienfunktionen in der Metapher virtueller Wissensräume meist nur wenig Relevanz.

Beispiele derartiger Systeme sind die DIVA-Umgebung [Sohlenkamp & Chwelos 1994], welche basierend auf dem Application Framework GINA in Common Lisp und CLOS (Common Lisp Object System) implementiert ist. Es handelt sich um eine zentralisierte Architektur, in der alle Applikationen von einem im Mittelpunkt stehenden Prozess aus gestartet werden. Auch die Datenhaltung ist zentralisiert in einer Datenbank vorgenommen. Nach Aussage der Autoren ist eine Umstellung auf eine replizierte Architektur vorgesehen.

Ein weiteres Beispiel ist das Clock Toolkit [Graham et al. 1996]. Es nimmt in den dargestellten Basistechnologien zur Entwicklung von kooperativen Anwendungen eine Sonderstellung ein. Die unter der Federführung von Graham entwickelte Programmiersprache Clock ist als funktionale Programmiersprache speziell auf die Entwicklung von verteilten Anwendungen hin ausgerichtet und konzentriert sich wie Rendezvous oder COAST auf die Übertragung des MVC-Konzeptes auf die Entwicklung von Groupware. Durch die spezielle Programmierumgebung ClockWorks lässt sich die Architektur einer Groupware-Applikation entwerfen und wesentliche Komponenten modellieren. Clock-Applikationen weisen bedingt durch den MVC-Ansatz eine strikt zentralisierte oder semi-zentralisierte Architektur auf. Clock ist also in dieser Form eine sehr spezialisierte Sprache zur Entwicklung von verteilter, interaktiver Software.²³⁵

Basierend auf der Erkenntnis, dass sich keines der untersuchten technischen Rahmenkonzepte zur Umsetzung selbststrukturierter und in Eigenverantwortung durch die Lernenden gesteuerten Lernprozessen eignet, wird im Folgenden die Umsetzbarkeit des im Rahmen dieser Arbeit entwickelten Architekturmodells dokumentiert. Umsetzbarkeit bedeutet in diesem Kontext das Aufzeigen wesentlicher konzeptioneller und technischer Eigenschaften der gefundenen Lösung.

Es ist festzuhalten, dass die im Folgenden vorgestellte Architektur kooperativer Wissensräume eine Reihe vorhandener innovativer Konzepte zusammenführt und diese um spezifische Möglichkeiten erweitert. In diesem Zusammenhang ist insbesondere die Verknüpfung ereignisbasierter Technologien mit den Fähigkeiten der WWW-basierten Dokumentenverwaltung einmalig.

Eine Darstellung des entwickelten Rahmenkonzepts unterteilt sich in die Architekturen von Server und Client sowie Schnittstellen des Servers. Aufseiten des Servers werden Klassenstruktur, Attribute, die Ereignisbehandlung und Persistenzebene desselben untersucht. Auf Seite des Clients stehen die Weiterführung der Architektur des Servers im Vordergrund sowie verschiedene, auch synchrone Client-Technologien mit ihrer Verknüpfung zum WWW.

²³⁵ Clock basiert auf der Programmiersprache Haskell.

5. Technische Umsetzung

Die technische Umsetzung des Konzepts kooperativer Wissensräume ist mit einer Reihe von Schwierigkeiten verbunden. Zum einen hat das vorherige Kapitel gezeigt, dass nur wenige Frameworks und Architekturkonzepte existieren, die in Teilbereichen den verfolgten Ideen eines nutzerzentrierten Ansatzes entgegenkommen und die Umsetzung primärer Medienfunktionen ermöglichen. Derartige Architekturen orientieren zumeist nicht an den de facto Standards des WWW und haben dementsprechend keine größere Verbreitung erlangt.

Auf der anderen Seite besitzen MUDs und MOOs viel versprechende Architekturkonzepte, lassen jedoch elementare Fähigkeiten des netzgestützten Dokumentenmanagements (im Umgang mit Lehrmaterialien) vermissen.

Der vorliegende Ansatz verbindet die schon konzeptuell untersuchten Entwicklungslinien (bürowirtschaftlich motivierter) dokumentenorientierter CSCW-/CSCL-Systeme mit den ereignisorientierten Architekturen der MUDs und MOOs. Es entsteht eine flexible Struktur kooperativer virtueller Wissensräume, die eine enge Nähe zum WWW besitzt, aber gleichzeitig neue Mechanismen der Verwaltung und Bereitstellung von Materialien entwickelt. Interaktionen zwischen Nutzern untereinander und Dokumenten werden als Interaktionen zwischen kooperierenden gemeinsamen (shared) Objekten abgebildet. Gemeinsame Objekte sind persistent und orientieren sich entlang einer Raum- und Dokumentenstruktur. Rechte kontrollieren die Instanziierung neuer Objekte und die Interaktionen zwischen Objekten. Letztere werden über Ereignisse (Events) zwischen Objekten (in Client und Server) abgebildet.

Eine weitere Schwierigkeit in dem Entwurf einer Rahmenarchitektur kooperativer Wissensräume liegt in dem Anspruch, eine erweiterbare und flexibel an existierende und zukünftige Standards anpassbare Architektur zu schaffen. Die Qualität der Erweiterbarkeit und Wartbarkeit einer kooperativen, verteilten Architektur liegt insbesondere in der Entwicklung entsprechender Softwarekomponenten. Hierzu ist die Aufteilung und Vererbungshierarchie von Klassen mit der richtigen Granularität zu wählen, Klassen-Schnittstellen (-Interfaces) müssen die nötige Universalität besitzen und es gilt, grundlegende Entwurfsmuster (Design-Patterns)²³⁶ zu entwickeln. Design-Patterns beschreiben wiederkehrende Muster eines objektorientierten Softwareentwurfs, die sich in einer An-

²³⁶ Die Idee der Entwurfsmuster – Design-Patterns geht auf Alexander zurück (vgl. [Alexander et al. 1977]).

zahl kommunizierender spezieller Klassen und Objekte ausdrücken (vgl. [Gamma et al. 2000, S. 3]).²³⁷

Der im Rahmen der vorliegenden Arbeit gestaltete Architekturforschung orientiert sich an einer Reihe existierender Design-Patterns aus den Bereichen des CSCW/CSCL, der MUDs und MOOs und des objektorientierten Softwareentwurfs. Diese werden erweitert, um neue Konzepte ergänzt und mit bestehenden Softwarekomponenten zur Implementierung der entsprechenden Architektur angereichert.²³⁸ Auf diese Weise entsteht eine Rahmenarchitektur, die weniger den Charakter eines in sich geschlossenen Produkts annimmt als den eines Frameworks zur Anpassung und Entwicklung zukünftiger Clients und Werkzeuge.

Das vorliegende Kapitel orientiert sich entlang der architektonischen Eigenheiten der gewählten Umsetzung. Zunächst wird ein kurzer Überblick über die Gesamtarchitektur der Client-Server-Umgebung gegeben (Abschnitt 5.1). In seinen Kernpunkten untergliedert sich die folgende Darstellung in die Betrachtung von Server (Abschnitt 5.2) und Client (Abschnitt 5.3). Wichtige Merkmale der serverseitigen Umsetzung sind eine spezielle Klassen- und Objektstruktur. Hierzu zählen das dynamische Registrieren von Attributen an Objekten und die gesteuerte (durch Rechte kontrollierte) Instanziierung von Objekten an Factories. Anschließend werden die Ereignisbehandlung des Servers (Event Handling) und seine Persistenzebene (Datenbankanbindung) beschrieben. Schnittstellen des Servers werden im Rahmen verschiedener Protokolle und der Bereitstellung einer Reihe von Object Request Broker (ORB) betrachtet. Clientseitig werden insbesondere Merkmale der Integration verschiedener Client-Technologien und die Anbindung an existierende Webbrowser untersucht.

Ein Überblick über die Elemente benutzerseitiger Anpassbarkeit und Strukturierbarkeit, also des nutzerzentrierten Ansatzes (Abschnitt 5.4), sowie eine Darstellung alternativer Möglichkeiten der Implementierung (Abschnitt 5.5) runden das Kapitel ab.

5.1 Überblick

Die vorgeschlagene Rahmenarchitektur untergliedert sich in den sTeam-Server, einen externen Webserver, eine relationale Datenbank und verschiedene Clients.

Der sTeam-Server koordiniert die Verwaltung von Benutzern und Gruppen, stellt die Raumstruktur mit den enthaltenen Objekten bereit, organisiert die Kommunikation zwischen Nutzern und realisiert die Synchronizität der dargestellten Informationen zwischen den Clients. Die Repräsentation von Objekten erfolgt über einen externen Webserver. Dieser ist mittels eines integrierten speziellen Moduls mit dem Server verbunden. Damit erfolgt der Zugriff eines Browsers auf den Server grundsätzlich zunächst über den Webserver. Letzterer verarbeitet eintreffende Anfragen, dient aber nicht der Speicherung und Ablage von Objekten.

In frühen Prototypen des sTeam-Ansatzes war der Webserver in den eigentlichen Server integriert (vgl. [Hampel 1999] und [Bopp 2000]). Die im Rahmen dieser Arbeit vorgestellte Architektur verbindet einen externen Webserver (Roxen)²³⁹ mit dem sTeam-Server. Auf diese Weise können innerhalb des Roxen implementierte Protokolle (z.B.

²³⁷ Design Patterns sind in dieser Weise keine komplexen Bausteine eines objektorientierten Entwurfs, sie beschreiben wiederkehrende grundsätzliche Strategien der Implementierung und Lösung eines Design-Problems.

²³⁸ Beispiele genutzter Softwarekomponenten sind HTML- bzw. XML-Parser und ein LPC-Interpreter. Genutzte Design Patterns des objektorientierten Softwareentwurfs sind der Proxy-Pattern, Factory-Pattern oder Command-Pattern (vgl. [Gamma et al. 2000, S. 6ff.]).

²³⁹ Vgl. <http://www.roxen.com>, Stand 1.12.2001.

das durch den Secure Socket Layer (SSL)²⁴⁰ verschlüsselte HTTP-Protokoll – HTTPS) unmittelbar genutzt werden. Hierbei verhält sich der sTeam-Server wie ein virtuelles Dateisystem an dem Webserver. Eine Behandlung von WWW-Zugriffen erfolgt durch den Webserver. Sie werden in Anfragen an den sTeam-Server umgesetzt.

In dem zurzeit realisierten Prototypen fiel die Wahl auf den frei verfügbaren Roxen-Webserver. Dieser ist wie der sTeam-Server in der Sprache LPC implementiert. Es handelt sich um einen modernen Webserver, der alle wichtigen Eigenschaften wie verschiedene Sicherheitskonzepte, Javascript und serverseitiges Java unterstützt. Die Wahl des Roxen-Servers ist nicht als Festlegung zu sehen, alternativ kann beispielsweise eine Anbindung des ebenfalls frei verfügbaren und weit verbreiteten Apache-Servers²⁴¹ entwickelt werden.

Kooperative Objekte (Materialien, Repräsentationen von Benutzern etc.) werden in der mittels einer SQL-Schnittstelle mit dem Server verbundenen Datenbank abgelegt.

Als Clients des sTeam-Systems werden zunächst gängige Internetbrowser verstanden. Spezifische Funktionalitäten wie synchrone Kommunikation (Chat) oder kooperative Applikationen werden durch spezielle, synchrone Clients bereitgestellt.

Der eigentliche sTeam-Server umfasst den sTeam-Kernserver, eine Anzahl von Erweiterungsmodulen (Packages) und eine externe SQL-Datenbank als Ort der Ablage von Klassen und Objekten (Object-Repository). Erweiterungsmodule lassen sich flexibel in den sTeam-Server einbringen. Zurzeit existiert z.B. ein erstes Erweiterungsmodul zwecks Bereitstellung einer externen Webschnittstelle (Webfrontend). Diese erlaubt den Zugriff auf den Server über einen gewöhnlichen Internetbrowser.

Die in Kapitel 2 aufgezeigten Anforderungen an die Architektur im Hinblick auf die Interaktion von Nutzern in virtuellen Wissensräumen, notwendige Mechanismen der gegenseitigen Wahrnehmung und des kooperativen Umgangs mit Lehrmaterialien oder beliebigen digitalen Medien verlangt eine spezielle Verwaltung von Objekten, Benutzern und verschiedenen Ereignissen (Events).

Unter dem Blickwinkel der technologischen Rahmenbedingungen basiert der sTeam-Server auf einer Implementierung der Klassen und Objektstruktur in der Interpretersprache LPC. Als Laufzeitumgebung findet der frei verfügbare²⁴² LPC-Interpreter Pike Verwendung. Pike besitzt neben einer guten Performanz eine breite Codebasis und weist eine große Anzahl existierender Bibliotheken auf, die in der Programmiersprache C implementiert sein können. Beispiele hierzu sind die Pike-SQL-Datenbankschnittstelle und ein Parser, welche beide in die Umsetzung der sTeam-Architektur einfließen.

Die sTeam-Architektur ist vollständig objektorientiert aufgebaut (vgl. [Hampel & Keil-Slawik 2001a]). Der überwiegende Teil ist in Form von Klassen und Methoden in LPC umgesetzt. Objekte besitzen verschiedene Attribute und ggf. einen Inhalt (Content) wie z.B. Dokumenttypen.

Interaktionen zwischen Objekten werden über gegenseitige Methodenaufrufe oder Events vorgenommen. Hierbei geht das entwickelte Konzept der Verarbeitung von Ereignissen in einer zentralen Event-Queue und der Differenzierung zwischen lokalen und globalen Events um einiges über das der MUDs und MOOs hinaus²⁴³, wenn auch in der

²⁴⁰ SSL 3.0 Specification, 26. 8. 1996, <http://home.netscape.com/eng/ssl3/>, Stand 1.12.2001.

²⁴¹ Vgl. <http://www.apache.org>, Stand 1.12.2001.

²⁴² GNU Public Licence, GPL.

²⁴³ MUDs und MOOs organisieren Ereignisaustausch zwischen Objekten über gewöhnliche Callback-Funktionen, d.h. bei Eintreten eines Ereignisses aufzurufende Funktionen.

Idee, Kooperationsprozesse durch Ereignis-/Nachrichtenaustausch zwischen Objekten zu realisieren, die verwandtschaftliche Nähe zu den MOOs und MUDs zu finden ist (vgl. Abschnitt 4.2.5).

Die Kommunikation zwischen Server und Clients wird über eine spezifische Kommunikationsschicht, den Client Object Access Layer (COAL) implementiert. COAL dient sowohl der Kommunikation zwischen Server und Webserver (dem integrierten Schnittstellenmodul)²⁴⁴, aber auch der Kommunikation zwischen Server und verbundenen Clients. Zurzeit ist ein COAL-API für die Programmiersprache Java implementiert.²⁴⁵

Das COAL-/Java-API erlaubt neben der An- und Abmeldung eines Benutzers (Clients) das Abonnieren und den Empfang von Events aus dem Server, das Ansprechen serverseitiger Objekte (Methodenaufrufe) sowie den Transfer (Up- bzw. Download) von Dateien.

Die Objektstruktur des Servers wird auf der Seite des Clients analog abgebildet. Für jeden Server-Objekttyp existiert eine korrespondierende Klasse im COAL-/Java-API. Instanzen dieser Klassen entsprechen realen Objekten im Server, beinhalten jedoch nicht die Daten und Inhalte des Objekts. Eine Replikation der Objekte bezieht sich jeweils auf den Ausschnitt der betrachteten oder benötigten Objekte. Über die auf diese Weise erzeugten Objekthüllen²⁴⁶ kann durch Methodenaufruf auf Objekte im Server direkt zugegriffen werden (z.B. im Falle der Änderung eines Attributs). In diesem Punkt enthält der sTeam-Ansatz Anteile einer semi-replizierten Architektur. Die gesamte Ereignisbehandlung und Persistenzhaltung von gemeinsamen Objekten ist für die Clients damit transparent.

Weiteres Schlüsselkonzept des Servers sind die so genannten Object Request Broker (ORBs). Dieses aus dem CORBA-Umfeld abgeleitete Vorgehen erlaubt es, unterschiedliche Strukturen und Sichten auf die Raum- und Objektstruktur des Servers zu erzeugen. Dies ist z.B. notwendig, um über unterschiedliche Schnittstellen (Clients) auf Räume und Dokumente zugreifen zu können. Beispielsweise benötigt ein Dateisystem-orientierter FTP- oder WebDAV-Client eine gänzlich andere Aufbereitung der Raum- und Objektstruktur als beispielsweise ein spezieller Java-Client.

In ähnlicher Weise wird die sTeam-Webschnittstelle aus Objektrepräsentationen erzeugt. Objekte (Inhalt und Attribute) werden in XML beschrieben und per Stylesheet in eine HTML-Repräsentation konvertiert und zugreifbar gemacht.

In dieser Form ist das Erscheinungsbild eines Raums mit den enthaltenen Objekten im WWW (über einen Browser) nicht fest programmiert, sondern wird dynamisch entlang einer Reihe von Beschreibungen aus den Objekten eines Raums erzeugt.

5.2 Server

Die vorliegende Arbeit sieht ihre primäre Aufgabe in dem innovativen Schritt vom theoretischen Modell der Medienfunktionen zu einem konkreten technischen Rahmenkonzept. In diesem Gestaltungsprozess nimmt der Server eine exponierte Stellung ein. Zu-

²⁴⁴ Die Realisierung des COAL-Moduls für den Roxen-Webserver erfolgt ebenfalls in Pike (LPC).

²⁴⁵ Die Konzeption und Implementierung eines COAL-APIs für C++ befindet sich zurzeit in der Umsetzungsphase.

²⁴⁶ Typen von Objekthüllen (clientseitige Proxy-Objekte) werden zurzeit nicht automatisch erzeugt, sondern müssen auf Seite des Clients nachgepflegt werden. Hier wäre eine Erweiterung denkbar, so dass vollständig neue (zunächst unbekannte) Objekttypen in den Client übertragen werden.

grundelegte Klassen- und Objektstrukturen bestimmen die Gestaltungsmöglichkeiten späterer anwendungsspezifischer Clients. Aus diesem Grunde erfolgt im Rahmen der vorliegenden Betrachtung zunächst eine sorgsame konzeptuelle Darstellung der Server-Architektur.

Diese untergliedert sich zunächst in die Darstellung der Klassenstruktur (Abschnitt 5.2.1), der gesteuerten Instanziierung von Objekten in Factories (Abschnitt 5.2.2) sowie der Vorstellung des flexiblen Attribut-Konzepts (Abschnitt 5.2.3) und der Realisierung von Annotationen (Abschnitt 5.2.4). Anschließend folgt die Darstellung der Ereignisbehandlung (Abschnitt 5.2.5) und der Persistenzebene des Servers (Abschnitt 5.2.6). Objekt Request Broker (ORBs) bereiten die Dokumenten- und Raumstruktur unter verschiedenen Gesichtspunkten auf (Abschnitt 5.2.7). Die Überleitung zur Darstellung der Client-Architektur findet sich in der Vorstellung des COAL-Protokolls (Abschnitt 5.2.8).

5.2.1 Klassenstruktur des Servers

Die fundamentale Idee der vorgeschlagenen Konzeption ist die Abbildung virtueller Wissensräume mit den beteiligten Personen und Materialien in Form einer Struktur aus kooperierenden Objekten.

Speziell der Anspruch der Umsetzung von primären Medienfunktionen in einer kooperativen Lernumgebung ist maßgeblich von der zugrunde liegenden Klassen- bzw. Objektstruktur abhängig (vgl. Abbildung 5-1).

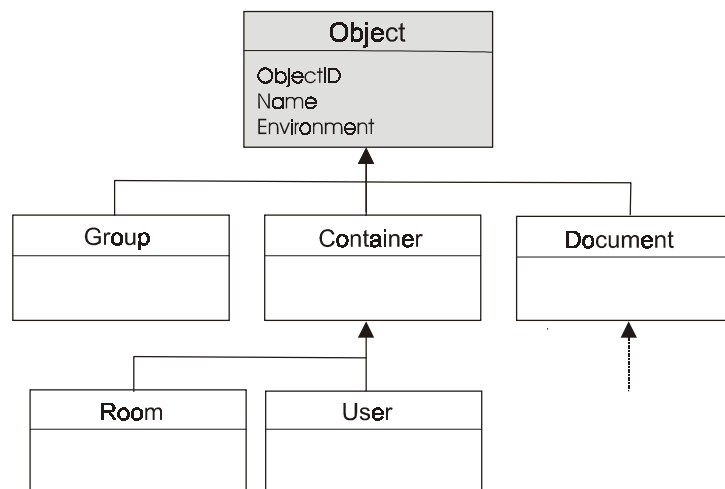


Abbildung 5-1: Klassenstruktur

In dieser Form ist ein Objekt eine persistente Instanz seiner sTeam-Klasse. Damit ist beispielsweise jedes Dokument innerhalb eines Raumes, aber auch der Raum selbst eine Instanz der zugehörigen Klassen „Document“ bzw. „Raum“. Objekte sind über die gesamte Lebensdauer des Systems persistent und werden in der Datenbank gespeichert. Objekte repräsentieren damit nicht nur den internen Aufbau des Systems, sie verkörpern auch die kooperativen Strukturen und Daten der Lernumgebung.

Objekte des Servers kapseln einen Zustand, ein Verhalten und eine Anzahl von Attributen. Objekte besitzen eine eindeutige Schnittstelle des Zugriffs (Methoden und At-

tribute), diese wird in der Literatur zumeist als Interface gekennzeichnet (vgl. [Trevor et al. 1994, S. 221]).

Für die Integration von Sicherheitsmechanismen in das kooperationsunterstützende System ist es unumgänglich, dass die Instanziierung von Objekten unter kontrollierten Bedingungen vorgenommen wird. Es darf weder Nutzern noch Programmen möglich sein, Objekte zu schaffen, die über die eigenen existierenden Ermächtigungen hinausgehende Zugriffsberechtigungen zugebilligt bekommen. Zudem verlangt die Persistenzhaltung von Objekten (speziell Mechanismen der Aktualisierung von Objektinstanzen) ein registriertes Erzeugen von Objekten.

Aus den obigen Gründen werden in der entwickelten Architektur Objekte durch eine eindeutige Instanz bereitgestellt. Diese Instanz wird als Factory bezeichnet. Das Konzept der Factory ist an das gleichnamige Entwurfsmuster (Design-Pattern) objektorientierten Designs angelehnt (vgl. [Gamma et al. 2000, S. 87ff.]) und erweitert dieses um eine Reihe von Eigenschaften wie beispielsweise das Registrieren von Attributen an Factories. Factories erzeugen neue Instanzen von existierenden Klassen nach Überprüfung aller notwendigen Berechtigungen durch ein so genanntes Security Object. Entstehende Objektinstanzen werden in der Datenbank registriert (vgl. Abbildung 5-2).

Eine zweite Aufgabe der Factories ist die Registrierung von einer weiteren, über einen festen Satz von Standard-Attributen hinausgehende Anzahl von Objektattributen. Attribute werden vornehmlich von spezifischen Clients registriert, um eine zusätzliche Funktionalität an Objekten bereitstellen zu können. (Ein Beispiel wäre ein Attribut „Verfallsdatum des Dokumentes“, welches z.B. für Ankündigungen oder kurze Mitteilungen verwendet werden kann.) Typische registrierte, zusätzliche Attribute sind Koordinatenangaben zu Dokumenten, die z.B. eine Positionsbestimmung des Dokumentes im virtuellen Raum zulassen.

Clients können über festgelegte Funktionen erfragen, welche Schnittstellen und Funktionalität ein Objekt bereitstellt. Anschließend erfolgt der Zugriff über in die Clients replizierte Proxy-Objekte (vgl. Abschnitt 5.3.4). Dabei wird nicht der Objektinhalt in die Clients repliziert, es werden lediglich für den Client transparente Schnittstellen des Zugriffs bereitgestellt. Auf diese Weise können Clients mit neu in das System eingebrachten Objekt- und Attributtypen interagieren, die so in der Schnittstellen-Spezifikation (COAL) nicht vorgesehen sind.²⁴⁷

Für ein Verständnis der spezifischen Eigenschaften der Server-Architektur wird im Folgenden die Klassen- und Objektstruktur vereinfacht dargestellt sowie ihre grundlegenden Objekttypen erläutert.

Der wesentliche Teil der Klassenstruktur repräsentiert die Benutzer- und Raumstruktur bzw. die kooperativen Objekte (Materialien) der Lernumgebung. Zusätzliche, zur Implementierung des Systems selbst notwendige Klassen orientieren sich ebenfalls an dieser Klassenstruktur. (So ist beispielsweise die Sicherheitsinstanz der sTeam-Umgebung, das „Security-Object“, als eigenes Objekt (Modul) implementiert.)

Die Architektur des Servers setzt sich aus den Klassen „Object“, „Container“, „User“, „Group“ und „Document“ zusammen. Alle Klassen sind von der Basisklasse „Object“ abgeleitet (vgl. Abbildung 5-1). Dies ist insofern zwingend erforderlich, als

²⁴⁷ In vorliegenden Prototypen ist das Konzept des automatischen Erzeugens neuer Objekttypen aufseiten des Clients noch nicht vollständig entwickelt. Zurzeit müssen Schnittstellen neuer Objekttypen manuell nachgepflegt werden. Attribute und Methoden werden jedoch an den Factories registriert und liegen in den entsprechenden Instanzen vor.

elementare Sicherheitskonzepte für das System aufrechtzuerhalten sind, die über eine Instanziierung der Klasse „Object“ realisiert sind. Die Klasse „Object“ enthält als Basis-Klasse alle wesentlichen Methoden und Attribute eines kooperativen Elements. Alle weiteren, abgeleiteten Klassen erweitern sie um spezielle Funktionalität.

Klasse Objekt (Object)

Die Klasse „Object“ ist die Basis jedes Objekts des sTeam-Systems. Wesentliche Eigenschaft der Klasse „Object“ ist zunächst die Persistenzeigenschaft jeder Instanz, welche für das abgeleitete Objekt transparent vorgenommen wird. Objekte werden in der relationalen Datenbank persistent verwaltet. Sie werden von dem objektorientierten Modell in relationale Tabellenstrukturen überführt bzw. serialisiert. Zu einem beschleunigten Zugriff werden Objektinstanzen in einem Cache verwaltet. Die für den Server eindeutige Objekt-Identifikationsnummer (ID) dient als primärer Schlüssel der abgebildeten Datenbankstruktur.

Die Klasse „Object“ enthält zudem alle Attribute, Methoden und Ereignisse (Events), die mit Objekten verknüpft werden können. Attribute werden im Zuge der Instanziierung von Objekten spezifiziert bzw. neu registriert (vgl. Abschnitt 5.2.3).

Objekte besitzen eine eindeutige Umgebung, die im Folgenden als „Environment“ bezeichnet wird. Das Environment eines Objektes kann beliebige andere persistente Objekte aufnehmen (vgl. Abbildung 3-5). In dieser Form besitzt jede Instanz der Klasse „Object“ zugleich eine gruppierende/strukturierende Wirkung auf anderweitige Objektinstanzen. Beispielsweise lässt sich die Struktur virtueller Räume über einen derartigen Mechanismus architektonisch abbilden: Jedes Objekt ist genau an einer Stelle innerhalb der Raumstruktur vorhanden und kann weitere Objekte kapseln. Räume und Container fassen als spezielle Varianten des Objektes weitere Objekte zusammen. Jeder einer Gruppe oder einem Benutzer zugeordnete Raum („Arbeitsraum“ oder „Workroom“) ist Wurzelknoten einer derartigen Struktur (vgl. Abbildung 3-1).²⁴⁸

Klasse Container

Eine einfache Ableitung von der Klasse „Object“ ist der Container. Er dient der logischen Kapselung von Objekten und ist die Strukturierungshilfe für einen Raum. Der objektorientierten Denkweise entsprechend kann ein Container andere Objekte beinhalten. In diesem Sinne ist die interne Objektrepräsentation analog einer semantischen Gruppierung von Materialien in Gruppen (Containern) vorgenommen. Entsprechend ist auch der Rucksack eines Benutzers, in dem ein Lernender Materialien mit sich führen und mit anderen Nutzern austauschen kann, von der Objektrepräsentation als Container realisiert. Recht universelle Eigenschaften von Containern, wie das Gruppieren beliebiger Objekte, resultieren unmittelbar aus dieser Eigenschaft.

Aus einem derartigen, recht allgemeinen, objektorientierten Ansatz ergibt sich die Realisierung des Raums als spezielle Variante eines Containers. Ein Raum gruppiert Personen, Dokumente und Werkzeuge, also sämtliche sTeam-Objekte. Hinzu kommen lediglich zusätzliche Methoden (beispielsweise zur Kommunikation innerhalb des Raums).

²⁴⁸ Die zugrunde gelegte Raumstruktur lässt sich demgemäß nicht auf einen einzigen Baum reduzieren, ist vielmehr ein „Wald“ von Bäumen. (Der sTeam-Wurzelraum ist ein eigener Baum und Ausgangspunkt für den Zugriff auf innerhalb des Systems abgelegte technische Objekte und Skripte, er entspricht dem „/“ im WWW, vgl. Abschnitt 5.2.7).

Verbindungen zwischen Räumen sind als Objekte realisiert, für deren Benutzung spezifische Rechte notwendig sind. Einen Ausgang oder einen anderen Raum zu betreten bedeutet das Recht zu besitzen, ein Raumobjekt lesen zu dürfen.

Klasse Benutzer (User)

Für jeden angemeldeten Benutzer der sTeam-Umgebung existiert genau ein Benutzerobjekt (User-Object). Das Benutzerobjekt befindet sich abhängig von der „Position“ des Benutzers innerhalb eines bestimmten Raums und repräsentiert den realen Nutzer innerhalb eines virtuellen Raums.

Benutzer können sich als bewusste konzeptionelle Einschränkung lediglich in Raum-Objekten, nicht in Containern aufhalten. Die Kommunikation zwischen einem realen Anwender, der sich über einen Client an dem System anmeldet, und anderen Benutzern in der Lernumgebung wird über ein an das Benutzerobjekt angeheftetes Kommunikationsobjekt vorgenommen. Da die Möglichkeit besteht, dass ein Benutzer zur selben Zeit über mehrere Clients mit dem System verbunden ist, kann ein Benutzerobjekt gleichzeitig mehrere Kommunikationsobjekte besitzen. Sämtliche Kommunikation zwischen Nutzern und zwischen der Lernumgebung und den Nutzern wird über die Benutzerobjekte koordiniert. Sie sind damit zugleich der Schlüssel zu einer Bereitstellung von geeigneten Awareness-Informationen.

Ausgehend von der Klassenstruktur handelt es sich bei dem Benutzerobjekt um einen mit verschiedenen Attributen und Methoden erweiterten Container. Damit erbt das Benutzerobjekt die Möglichkeit, verschiedene Objekte zu kapseln. Der Rucksack, das „Inventory“ eines Benutzers, befähigt ihn auf diese Weise, in der MUD-Terminologie gesprochen, Gegenstände (andere Objekte) mit sich zu führen.

Spezielle Eigenschaften des Benutzers sind z.B. Name, Zugangskennwort etc. und werden zum Teil als Attribute in dem entsprechenden Benutzerobjekt verwaltet. Der Name eines Benutzers ist eindeutig zu wählen und dient der Identifikation des Benutzerobjektes innerhalb der Lernumgebung.

Synchrone Kommunikation zwischen den Benutzern (den Benutzerobjekten) eines virtuellen Raumes wird über Ereignisse realisiert. Hierbei geht das verwirklichte Konzept der Ereignisbehandlung über das gängige MUD-Event-Modell hinaus. Beispielsweise löst eine Chat-Mitteilung eines Benutzers das Event „*Benutzer X sagt: Text*“ aus. Der entsprechende Event wird von den Empfängern des Ereignisses verarbeitet.

Architektonisch nimmt damit das Benutzerobjekt eine wichtige Position in der Repräsentation kooperativer Prozesse zwischen verschiedenen Lernenden und den anderweitigen Objekten ein. Es ist damit das virtuelle Gegenstück eines realen Nutzers in der kooperativen Lernumgebung (es erlaubt beispielsweise die Identifikation eines Lernenden) und bildet die entscheidende Instanz zur Koordination von Kommunikationsmechanismen aller Art.

Klasse Benutzergruppe (Group)

Die Benutzergruppe ist eine logische Kapsel für eine Anzahl von Benutzern. Die Klasse „Group“ ist direkt von der Klasse „Object“ abgeleitet und besitzt einen ähnlichen Status wie ein Benutzerobjekt. Benutzergruppen können Untergruppen und weitere Benutzer enthalten. Die Klasse einer Benutzergruppe bildet die konzeptionelle Einheit zur Abbildung von Benutzer- und Gruppenstrukturen. Auf diese Weise ist der Aufbau einer Hierarchie von Gruppen und Benutzern möglich (vgl. Abbildung 3-2).

Eine Benutzergruppe kann dementsprechend verschiedene Nutzer logisch zu einer Gruppe mit gleichartigen Eigenschaften zusammenfassen. Durch die konzeptionelle

Nähe der Klassen „User“ und „Group“ ist die Mehrzahl von Methoden und Ereignissen von Objekten sowohl auf einzelne Benutzer als auch auf Benutzergruppen anwendbar. Beispielsweise kann über die Anwendung eines Chat-Kommandos auf ein Gruppenobjekt eine ganze Gruppe von Nutzern angesprochen werden.

Benutzergruppen nehmen insbesondere im Rahmen der Spezifikation von Berechtigungen eine exponierte Stellung ein. Ein Gruppenobjekt steht gleichbedeutend einem Benutzerobjekt in einer Access-Control-List (ACL) für die Organisation von Berechtigungen an einer ganzen Gruppe von Personen. Damit schlägt sich die logische Struktur aus Lernern und zu speziellen Gruppen zusammengefassten Lernenden in der internen Repräsentation von persistent abgelegten Benutzern und Gruppenobjekten nieder.

Klasse Dokument (Document)

Alle Interaktionen innerhalb kooperativer Wissensräume basieren im Wesentlichen auf dem Erzeugen, der Manipulation und dem Austausch von Materialien. Sie bilden die architektonische Basis der Realisierung von primären Medienfunktionen in der kooperativen Lernumgebung.

Alle Formen von digitalen Medien werden über Instanzen der Klasse „Document“ abgebildet. Ein Dokument zeichnet sich gegenüber einem herkömmlichen Objekt durch seine Inhaltskomponente, dem „Content“, aus. Dieser repräsentiert die in der Datenbank abgelegten digitalen Medien, z.B. eine Grafik, ein elektronisches Dokument oder auch ein Hypertextdokument (WWW-Dokument). Unterschieden werden verschiedene Dokumententypen durch den innerhalb eines Attributes gespeicherten MIME-Type. – Es sind also zunächst nur wenige spezifische Dokumentklassen für unterschiedliche in das System eingebrachte Typen von Lehrmaterialien oder Anwendungsobjekten notwendig. Ausnahmen bilden hier lediglich die Verwaltung von Hypertextdokumenten, in deren Umfeld spezielle Verarbeitungsprozesse notwendig sind, (konsistente Verweise) oder Grafiken, die spezifische Mechanismen der Verwaltung wie Vorschaubilder (Thumbs) besitzen.²⁴⁹ Alle Dokumententypen, die spezielle Verarbeitungsprozesse erfordern, werden jedoch von ein und derselben Factory erzeugt.

Der Objekttyp Dokument ist direkt von der Basisklasse „Object“ abgeleitet und besitzt auf diese Weise alle Fähigkeiten eines sTeam-Objekts bzgl. der Verarbeitung von Ereignissen oder der Annotierbarkeit. Jedes Objekt (alle Lehrmaterialien) sind annotierbar. Annotationen sind als Verweise auf Annotationsobjekte realisiert und werden als weiterer Teil der Inhaltskomponente innerhalb eines Objekts gespeichert. Jedes Objekt enthält entsprechend eine Liste mit verbundenen Annotationsobjekten.

Eine weitere interessante architektonische Perspektive ist die Ablage von unterschiedlichen Versionen (Versionsmanagement) eines Dokumentes in der Umgebung des Ursprungsdokumentes. Damit lassen sich Technologien der Versionsverwaltung oder Mehrsprachigkeit von Dokumenten in Zukunft elegant über einfache Mechanismen in die Architektur integrieren.

Die in der dargestellten Klassenstruktur realisierte strenge Trennung zwischen einem Objekt, seiner Dokumenteigenschaften (Attribute) und dem eigentlichen Inhalt bietet den Vorteil einer flexiblen Verarbeitung unterschiedlicher Dokumentinhalte ohne der Notwendigkeit eines zusätzlichen hohen Verarbeitungsaufwandes. Aktuelle Prototypen des sTeam-Systems speichern den Inhalt (content) von Dokumenten in der verbundenen relationalen Datenbank. Denkbar wäre in diesem Zusammenhang, spezielle Multimedia-

²⁴⁹ Objekttypen, die eine Sonderbehandlung erfahren, sind dementsprechend Hypertextdokumente (DocHTML), Grafiken (DocGraphics) und Audio-Dateien (DocMP3).

Datenbanken oder auch einfache Dateisysteme zur effizienten Speicherung des Dokumentinhaltes hinzuzuziehen.

Die Trennung zwischen dem eigentlichen Dokumentenobjekt und seinem Inhalt erlaubt zudem eine einfache und effiziente Verwaltung von Dokumenten innerhalb von Clients. Erst im Falle des eigentlichen Zugriffs auf den Inhalt durch den Benutzer (z.B. durch eine Upload- oder Download-Operation) ist der jeweilige Dokumentinhalt aus der Datenbank zu laden.

Eine Sonderbehandlung unter den unterstützten Dokumenttypen (Instanzen der Klasse „Document“) nehmen hierbei die schon genannten verschiedenen Hypertextdokumente (wie HTML- oder XML-Dokumente) ein. Diese können innerhalb des Dokumentes (also innerhalb des Object-Content) Verweise auf externe Dokumente (URLs) besitzen. Um Inkonsistenzen dieser Verweise durch den Prozess des Einfügens in die sTeam-Umgebung oder bei späteren Verschiebungen des Dokumentes zu vermeiden, sind Verweise einer Sonderbehandlung zu unterziehen.

In der jetzigen Umsetzung werden Verweise aus HTML-Dokumenten extrahiert und durch eindeutige Referenzen ersetzt. Eine derartige Ersetzung findet beim Einfügen und Auslesen des Dokumentinhaltes aus der Datenbank durch Überschreiben der hierzu im Objekt vorhandenen Funktionen statt.

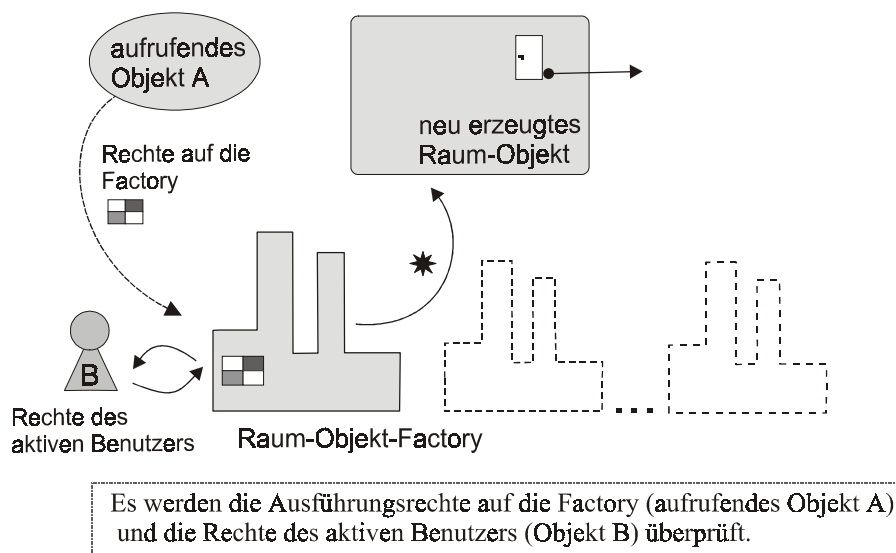


Abbildung 5-2: Factories – Instanzieren von Objekten

5.2.2 Factories

Wichtiges Merkmal der entwickelten Architektur ist die kontrollierte Instanziierung neuer Objekte über speziell zu diesem Zweck bereitgestellte Objekte, den Factories.

Factories entsprechen weitgehend dem Factory-Pattern²⁵⁰, erweitern dieses jedoch um eine Reihe von Eigenschaften. Hierzu zählen die dynamische Registrierung von Attributen an Factories und eine Implementierung von Sicherheitsfunktionen an diesen.

Für jede ableitbare Klasse der sTeam-Objektypen (vgl. Abschnitt 5.2.1) existiert genau eine Factory, d.h. jede Klasse besitzt ein zugeordnetes Objekt, welches eine Funk-

²⁵⁰ „Abstract Factory: Provide an interface for creating families of related or dependent objects without specifying their concrete classes“ [Gamma et al. 2000, S. 87ff.].

tion „execute ()“ zur Bereitstellung von Instanzen der Klasse besitzt (vgl. Abbildung 5-2).

Das Konzept einer eindeutigen Stelle für die Erzeugung von neuen Objekten birgt eine ganze Anzahl von Vorteilen:

- I. Factories erzeugen Instanzen von Klassen sämtlicher sTeam-Objekttypen. Hierbei wird nicht alleine die entsprechende Instanz zu einer Klasse erzeugt, sondern auch ggf. Initialisierungen vorgenommen. So erzeugt beispielsweise die Factory eines neuen Benutzer-Objekts (User-Factory) einen zugehörigen Arbeitsraum sowie eine Mailbox. Des Weiteren wird der Nutzer in die Gruppe aller sTeam-Nutzer eingetragen und in seinen Arbeitsraum bewegt.
- II. Es findet eine Registrierung von Attributen zu Objekten innerhalb der Factory statt. Mittels eines derartigen Mechanismus können Clients neue Attribute zu existierenden Objekten registrieren und ohne Modifikation des Kernservers spezifische Merkmale implementieren. Neu hinzugefügte Attribute werden allen laufenden Objektinstanzen automatisch angefügt.
- III. Objektinstanzen sind (über eine Datenbanktabelle) zentral registriert und können einfach und unkompliziert zur Laufzeit aktualisiert werden. Factories dienen u.a. der Aktualisierung von Objektinstanzen. Dabei können neue Attribute oder Methoden hinzugefügt werden, ohne den Server übersetzen oder erneut starten zu müssen. Laufende Instanzen eines modifizierten Objekttyps werden dynamisch aktualisiert.

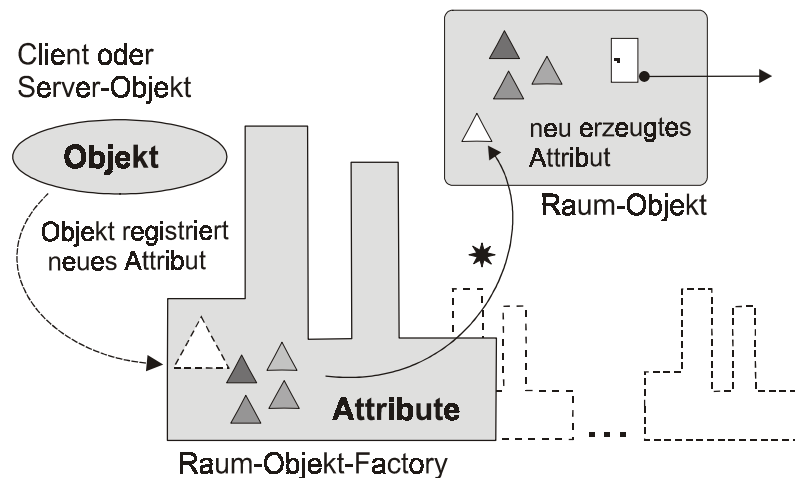


Abbildung 5-3: Registrieren von Attributen an Factories

Besonderes Merkmal des im Rahmen der sTeam-Architektur gewählten Factory-Ansatzes ist die Registrierung von Attributen zu Objekten an den Factories. Benötigt ein Client ein spezielles Attribut, wie beispielsweise den räumlichen Abstand einer Person zu einer virtuellen Tafel (z.B. für eine Darstellung einer Art Sitzordnung), kann der entsprechende Client dieses neue Attribut bei der Factory der Klasse Raum registrieren und fortan nutzen (vgl. Abbildung 5-3). Attribute sind damit von der Implementierung einer Klasse entkoppelt und können dynamisch den Gegebenheiten der Clients angepasst werden. Resultat ist eine „flache“ Klassenstruktur des Servers, die nur mit sehr wenigen Basisklassen realisiert ist und nicht für jedes zusätzliche Attribut eines neuen Objekttypen eine zusätzliche Klasse benötigt.

Wesentliches Kernkonzept der Factories ist die Ausweitung des Sicherheitskonzeptes auf die Objekt- und Klassenstruktur des Servers. Nur berechtigte Benutzer oder Benutzergruppen sind in der Lage, neue Instanzen einer Klasse zu erzeugen und damit das System dauerhaft zu modifizieren. Es findet also keine Sicherheitsüberprüfung allein auf Ebene der Benutzerschnittstelle statt, das Sicherheitskonzept gewährleistet die Integrität des Systems bis auf die Ebene der Erzeugung von einzelnen Objekten. Factories sind damit Grundvoraussetzung für den Ausbau einer kooperativen Applikation zur Laufzeit und die Erweiterung der Umgebung durch die Nutzer des Systems selbst.

Das verfolgte Konzept der Selbstadministration, d.h. die Möglichkeit, den Nutzern des Systems Rechte an der Verwaltung, Umgestaltung und Erweiterung der Umgebung einzuräumen, basiert in seiner architektonischen Fundierung auf einer derartigen geschützten Objekt-Instanziierung.

Beachtenswert ist in diesem Zusammenhang, dass eine Anwendung von Berechtigungen auf das Erzeugen neuer Klassen und die Instanziierung existierender Klassen differenziert erfolgt. Anders als in einer Reihe von Systemen, in denen oftmals lediglich unterschieden wird, ob ein Benutzer den Status besitzt, neue Klassen erzeugen zu dürfen, erfolgt in der sTeam-Architektur eine Überprüfung von Berechtigungen für jede einzelne Factory. Für die Instanziierung eines Objektes sind die Rechte des aktiven Benutzers sowie die Rechte des aufrufenden Objekts erforderlich. Auf diese Weise kann differenziert das Recht vergeben werden, neue Objekte erzeugen zu dürfen. Unter dem Gesichtspunkt der Selbstadministration können Nutzer damit den virtuellen Raum entsprechend den zugesprochenen Berechtigungen aktiv um- oder neu gestalten. Beispielsweise entspricht das Erzeugen eines neuen Raums der Berechtigung zur Ausführung der entsprechenden Raum-Factory.

Das Erzeugen eines gänzlich neuen Objekttyps (beispielsweise eines Protokollwerkzeuges) benötigt minimal die Berechtigung auf die Objekt-Factory. Jedes neue Objekt, welches mit dem System bzw. anderen Objekten oder Benutzern interagieren muss, benötigt eine Ableitung von der Klasse „Object“, um Ereignisse (Events) versenden oder abonnieren zu können bzw. auf sicherheitsrelevante Funktionen zugreifen zu können.²⁵¹ Die Sicherstellung der Integrität der kooperationsunterstützenden Umgebung ist in Bezug auf die Klassen- und Objektstruktur durch eine Entkopplung von Berechtigungen von konkreten Objekten gewährleistet. Berechtigungen werden dementsprechend nicht direkt an das Objekt geheftet, sondern sind von dem aktiven Nutzer und Besitzer des Objekts abhängig. Damit wird jedem Objekt neben dem aktiven Benutzer, der aktuell mit dem Objekt interagiert, ein eindeutiger Besitzer (typischerweise der Erzeuger des Objekts) zugeordnet.

Soll eine sicherheitskritische Operation, wie z.B. der Aufruf einer Methode oder das Auslösen eines Ereignisses, durchgeführt werden, sind die Berechtigungen sowohl von dem aktiven Benutzer (also demjenigen, der die Operation ausüben möchte) als auch von dem Besitzer des Objektes notwendig (also dem Erschaffer der entsprechenden Klasse, vgl. Abbildung 5-2). Ein derartiges Vorgehen verhindert mögliche Sicherheitslücken, in denen einem autorisierten Benutzer ein Objekt „untergeschoben“ wird und auf diese Weise der Besitzer des Objektes weitreichende Berechtigungen erhält. Diese zentrale, konzeptionelle Eigenschaft der sTeam-Architektur erlaubt einen flexiblen Umgang mit dem Problem, Prozesse der Selbstadministration unterstützen zu wollen bei gleichzeitiger Sicherstellung der Integrität des Systems.

²⁵¹ Alle wesentlichen Sicherheitsfunktionen des sTeam-Systems sind in der Klasse „object“ implementiert und nicht überschreibbar. Zu jedem Zeitpunkt findet eine Überprüfung vorhandener Objektinstanzen auf ihre Ableitung von Objekt statt.

Benutzer können neue Instanzen von existierenden Objektklassen im Rahmen ihrer persönlichen Berechtigungen an dem Objekt erzeugen. Gleichzeitig stehen die erzeugten Objekte unter der Kontrolle der Berechtigungen des Besitzers der entsprechenden Klasse. Damit existiert kein Weg, durch das Erzeugen neuer Objekte erweiterte Rechte gegenüber den eigenen (des ausführenden Objekts) und denen des aktiven Nutzers zu erlangen.

5.2.3 Attribute

Wesentliches Merkmal jeden Objekts innerhalb kooperativer Wissensräume sind neben seinem Inhalt (Content) zugehörige Attribute. Auch jeder virtuelle Raum (das zugehörige Objekt) oder ein Benutzer (das entsprechende Nutzerobjekt) besitzt eine Reihe von Attributen. Clients interpretieren Objekte in unterschiedlichen Kontexten. Hierbei spielen Attribute zu Objekten eine entscheidende Rolle. Sie dienen der Zuordnung (Identifikation) eines Objekts, der Ablage von Metadaten oder der Speicherung von technischen Eigenschaften.²⁵² Attribute sind in dieser Form als universeller Mechanismus der Zuordnung von spezifischen Informationen zu Objekten zu kennzeichnen.

Unterscheidbar sind zwei grundsätzliche Formen von Attributen: Semantische Attribute, welche das zugehörige Objekt um semantische Informationen bzw. Metadaten erweitern, sowie technische Attribute, welche die Realisierung bestimmter architektonischer Eigenschaften zum Zweck haben. Letztere sind zudem zu differenzieren in serverseitige Systemattribute, welche unmittelbar zur Umsetzung des Servers notwendig sind, und clientseitige Attribute, welche zur Bereitstellung spezifischer Eigenschaften eines Clients dienen (vgl. Abbildung 5-4).

Attribute besitzen im objektorientierten Softwaredesign eine lange Tradition. Das in diesem Zusammenhang entwickelte Verständnis von Attributen beschränkt sich jedoch zumeist auf den objektorientierten Entwurf von Klassen. Eine objektorientierte Denkweise für den Umgang mit den Daten einer Applikation findet sich nur in modernen Ansätzen objektorientierter Datenbanken.

Zugleich erlebt die Standardisierung von Metadaten zu wissenschaftlichen Datensätzen (und Lehrmaterialien) zurzeit eine Renaissance; dies ist sicherlich ein Hinweis auf die Notwendigkeit objektorientierter Strukturen in der Repräsentation von Daten.²⁵³ Im Zusammenhang mit der Entwicklung von kooperationsunterstützenden Systemen ergeben sich nur wenige Anhaltspunkte für die Ausgestaltung eines universellen Attributkonzepts bezogen auf kooperative gemeinsame Daten.²⁵⁴

Die Leistungsfähigkeit im Umgang mit Attributen an Dokumenten ist zumeist ein Indikator für die Anpassbarkeit und Erweiterbarkeit einer Architektur im Hinblick auf verschiedene Anwendungsszenarien. Zugleich ergibt sich durch die Eleganz der Reprä-

²⁵² Annotationen könnten ebenfalls innerhalb spezieller Attribute zu einem Objekt gespeichert werden, werden jedoch z.Z. in einem Bereich der Inhaltskomponente abgelegt.

²⁵³ Vgl. hierzu die Dublin Core Metadata Initiative: <http://dublincore.org/>, Stand 1.12.2001.

²⁵⁴ Erfahrungen der Notwendigkeit eines leistungsfähigen Attributkonzepts wurden in ersten Prototypen der sTeam-Umgebung gemacht (vgl. [Bopp 2000]). Eines der wenigen kooperationsunterstützenden Systeme, welche schon früh ein offenes Attributkonzept verfolgten, ist das COAST-System. Vereinzelt finden sich weitere Beispiele aus dem Bereich der CSCW-Systeme: So wird beispielsweise resultierend aus einem zu starren Attributkonzepts der Presto-Umgebung [Dourish et al. 1999b] in dem nachfolgenden „Macadam“-Prototypen ein Ansatz der Kopplung von Presto-Attributen/ Dokumenten und eigenständigen Java-Objekten gewählt. Eine derartige Erweiterung der Presto-Dokumentenstruktur durch frei instanziierebare Klassen (welche Methoden und Attribute besitzen) erlaubt die gewünschte flexible Kategorisierung und Hierarchisierung von Dokumenten (vgl. [Dourish et al. 1999a]).

sensation von Attributen eine schlanke Gesamtarchitektur (speziell des objektorientierten Entwurfs).

Das im Rahmen dieser Arbeit vorgestellte Attributkonzept ist maßgeblich durch die Repräsentation von gemeinsamen Materialien (Dokumenten) in kooperativen Objekten und die Registrierung von Attributen an Factories gekennzeichnet.

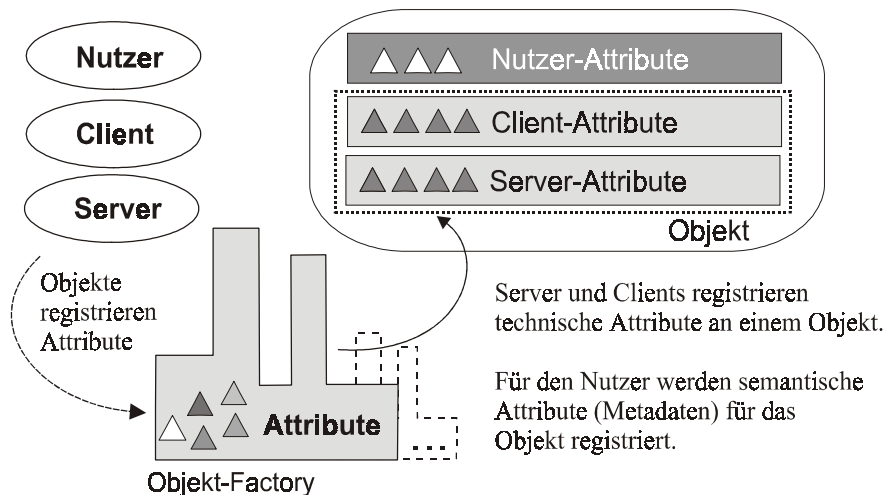


Abbildung 5-4: Nutzer-, Client- und Server-Attribute

Tätigkeiten wie das Einfügen neuer digitaler Materialien in einen Container oder Raum, also das Erzeugen einer Anzahl neuer Objekte, macht die Spezifikation einer Vielzahl von Attributen notwendig. Eine denkbare Lösung ist das Wählen geeigneter Voreinstellungen für Attribute. Zu Problemen führt dieses Vorgehen, wenn Objekte kurze Zeit nach ihrem Erzeugen verschoben, also in einen neuen Kontext gebracht werden müssen. In diesem Falle ist das nachträgliche Ändern einer großen Anzahl von Attributen notwendig.

Der verfolgte Ansatz wählt die Strategie der Vererbung von Attributen aus dem jeweiligen Kontext (Environment) oder einem spezifizierbarem Objekt. Objekte können jedes einzelne Attribut (seinen Wert) von einem anderen Objekt erben oder durch die Auswertung einer zu spezifizierenden Funktion erhalten. Genauer gesagt existiert für die Bestimmung jedes Attributs eine Funktion, die typischerweise auf das Environment zur Ermittlung des Attributs verweist, d.h. Attribute werden aus der Umgebung eines Objekts, beispielsweise für ein Dokument von seinem umgebenden Container, geerbt. Natürlich kann die entsprechende Funktion anwendungsspezifisch implementiert werden, d.h. der Wert eines Attributs wird dynamisch durch Funktionsauswertung ermittelt.

Attribute sind durch ihre Registrierung an den Factories hierarchisch spezifizier- und registrierbar. Subklassen (Kind-Objekte) erben die relevanten Attribute von ihren Elternklassen, können diese jedoch ggf. überschreiben.

Jedes Objekt besitzt eine Reihe automatisch registrierter Attribute an seiner zugehörigen Factory. Eine Factory registriert Attribute mit einem Schlüssel (Key), einer kurzen Beschreibung, dem Typ des Attributes, einem voreingestellten (Default-) Wert und der Angabe einer Vererbungsinstanz.

Der Schlüssel gestattet die eindeutige Identifikation jedes Attributs. Über ihn werden Attributänderungen vorgenommen. Die Vererbungsinstanz spezifiziert, von welchem Objekt bzw. welcher Funktion das Attribut gegebenenfalls geerbt wird.

An einer Factory registrierte Attribute sind in jeder neuen Instanz des entsprechenden Objekttyps vorhanden. Der Zugriff auf Attribute zu Objekten erfolgt über einheitliche Zugriffsfunktionen (`set_attribute`, `query_attribute`), welche zugleich Typüberprüfungen vornehmen.

Registrierungsdaten zu Attributen werden allein in der Factory gespeichert. Zu den Daten zählen neben den oben genannten Eigenschaften auch das auszulösende Lese- und Schreib-Event. In den meisten Fällen wird lediglich das Ereignis „Attributänderung“ bei schreibendem Zugriff auf das Attribut ausgelöst. Ein Lesezugriff wird aus Gründen der Effizienz (Performanz des Systems) nur in wenigen Ausnahmen ausgelöst. In den meisten Fällen ist das Lesen von Attributen als unkritisch einzustufen, lediglich persönliche Attribute eines Nutzers, wie die E-Mail Adresse sind mit Leseberechtigungen zu versehen.

Über die registrierten Ereignisse an Attributen erfolgt zugleich die Sicherheitsüberprüfung auf vorhandene Berechtigungen zu einer Modifikation des entsprechenden Attributs. Aus diesem Grunde ist es zwingend notwendig, registrierte Attribute einer Zugriffskontrolle durch ein entsprechendes Event zu unterwerfen.

Attribute können auch ohne Registrierung an der Factory einem Objekt zugefügt werden. Clients nutzen diesen Mechanismus beispielsweise, um nur für die existierende Instanz relevante Informationen abzulegen. Attribute, die einem Objekt ohne Registrierung an der Factory zugefügt werden, besitzen keinen persistenten Charakter für andere Instanzen der zugehörigen Klasse. Sie existieren nur über die Lebenszeit des jeweiligen Objekts.

Anwendungen des Attributkonzepts

Aus Anwendungsseite betrachtet sind nur wenige Attribute zu einem Objekt von den Nutzern explizit wählbar. Diese werden clientseitig bei der Erzeugung eines neuen Objekts in Form von Metadaten von dem Benutzer erfragt und können zum Teil optional angegeben werden. Derartige Attribute erlauben insbesondere präzise Suchoperationen in den Materialien der Lernumgebung. Sinnvolle Attribute zu Lehrmaterialien sind in diesem Zusammenhang beispielsweise Schlüsselworte, die Sprache des Dokumentes und eine kurze, textuelle Inhaltsangabe. Attribute wie der Titel des Dokuments oder sein Besitzer werden unmittelbar zur Koordination von kooperativen Prozessen benötigt und definieren elementare Grundeigenschaften des jeweiligen Objekts. Attribute können durch Clients bzw. die entsprechende Factory voreingestellte Werte besitzen.²⁵⁵

Eine ganze Reihe von konzeptionellen Anforderungen an die sTeam-Umgebung sind auf Basis von technischen Attributen lösbar. Clients nutzen systemseitig erzeugte Attribute an Objekten zu einer Realisierung verschiedener Werkzeuge und Formen ihrer Darstellung. Ein Shared Whiteboard vermag auf diese Weise die räumliche Position eines Objekts auf einer gemeinsamen Zeichenfläche zu ermitteln und zu speichern. Aber auch serverseitige Werkzeuge legen kooperative Daten in Attributen von gemeinsamen Objekten ab.

Ein elektronischer Briefkasten beispielsweise speichert den Zeitpunkt seiner „Leerung“ in einem Attribut an dem entsprechenden Objekt. Ein Protokollwerkzeug speichert protokollierte Diskussionsverläufe in einfachen Textattributen.

²⁵⁵ Eine interessante Erweiterung und Vereinfachung des expliziten Setzens von Attributen durch die Benutzer ist der Ansatz von Dourish et al., in dem durch so genannte Attribut-„Templates“ eine Anzahl von Attributen simultan in Abhängigkeit des neu eingebrachten Objekttyps gesetzt werden können (vgl. [Dourish et al. 1999b]). Attribute werden hierbei aus den Voreinstellungen der Nutzer zu verschiedenen Objekttypen ermittelt.

Als zweite wesentliche Anwendung von systemseitigen Attributen an Objekten werden Verweisobjekte durch die Speicherung einer Web-Adresse (einer URL) in einem Attribut des Verweisobjekts umgesetzt. Interne Verweisobjekte, wie z.B. Verbindungen zwischen Räumen, enthalten eine Objektreferenz, welche ebenfalls in einem Attribut gespeichert ist.

5.2.4 Annotationen

Annotationen nehmen, wie schon dargestellt, eine wesentliche Stellung in allen Formen kooperativer Lernprozesse ein. Mit dem Ziel, die Annotierbarkeit sämtlicher Materialien kooperativer Wissensräume sicherzustellen, sind diese technisch auf relativ niedriger Ebene in der Systemarchitektur zu verankern.

Des Weiteren besteht die Anforderung, den Typ einer Annotation nicht auf ein spezielles Format (lediglich Text) festzulegen, sondern prinzipiell sämtliche Objekttypen als eine Annotation zuzulassen. (Eine erste Anwendung sind beispielsweise Audio-Annotationen oder grafische Annotationen.)

Wie schon in Abschnitt 3.2.5 gezeigt, sind Annotationen als eigenständige Objekttypen realisiert. Sämtliche Formen von digitalen Materialien können mit Annotationen versehen werden. Hierzu ist eine Verknüpfung zwischen dem zu annotierenden Objekt und der eigentlichen Annotation (dem Annotationsobjekt) herzustellen.

Annotationen werden in Form einer Liste von Objektidentifikatoren (ObjectID) an dem jeweiligen Objekt abgelegt. Hierzu existiert eine spezielle Umgebung zu jedem Objekt (ein eigenes Inventory). Das Erzeugen einer Annotation entspricht demgemäß dem Erzeugen eines neuen Annotations-Objekts und dem Einfügen einer Referenz auf dieses in die Umgebung des zu annotierenden Objekts. Eine Annotation ist damit eine Referenz auf ein eigenständiges Annotationsobjekt.

Die Umsetzung von Annotationen als eigenständige Objekte bietet eine ganze Anzahl von Vorteilen. Hierzu zählen die Freiheit des Typs (Text, Grafik etc.) sowie die freie Manipulierbarkeit von Annotationen (primäre Medienfunktionen) unabhängig des annotierten Objekts. Auch Nutzerrechte können für Annotationsobjekte wie für jedes anderweitige Objekt spezifiziert werden.

5.2.5 Ereignisbehandlung

Neben der Repräsentation von virtuellen Wissensräumen, Materialien und Nutzern durch Objekte und Attribute sind Interaktionen zwischen diesen ebenfalls durch ein geeignetes Modell abzubilden. Eine derartige Aufgabe fällt der Ereignisbehandlung des sTeam-Servers zu – Objekte kommunizieren über Ereignisse (Events).

Das in der sTeam-Architektur entwickelte Ereignis-Modell geht über das der MUDs und MOOs um einiges hinaus. MUDs und MOOs orientieren sich gewöhnlich an Callback-Funktionen, d.h. der Befehl eines Nutzers löst beispielsweise einen Methodenauf-ruf in einem zugehörigen Objekt aus. Demgegenüber ermöglicht eine Ereignisbehandlung nach dem Publish-Subscribe-Paradigma²⁵⁶ den Objekten verschiedene Events zu abonnieren, d.h. die Reichweite eines Ereignisses einzuschränken (auf die Abonnenten des Events zu beschränken) und Ereignisse gleichzeitig in einer Reihe von Objekten auszulösen. Sowohl der sTeam-Server als auch die Client-Server-Kommunikation sind vollständig nachrichten- und ereignisgesteuert.

²⁵⁶ Das Publish-Subscribe-Paradigma beschreibt ein Ereignismodell, in welchem verschiedene Objekte Ereignisse erzeugen können. Diese werden von den Objekten bekannt gegeben (Publish) und können von anderen Objekten abonniert (Subscribe) werden (vgl. [Coulouris et al. 2001, S. 187ff.]).

Ereignisse nehmen damit eine exponierte Stellung in der Gestaltung der Gesamtarchitektur ein. Sie treten in allen synchronen und asynchronen Interaktionsformen zwischen Objekten und zwischen Nutzern und Objekten innerhalb des Servers auf. Events werden durch verschiedene an ein Objekt geknüpfte Ereignisse, wie z.B. „Benutzer X sagt etwas in den Raum“ oder „das Attribut ‚Titel‘ wurde verändert“ ausgelöst.

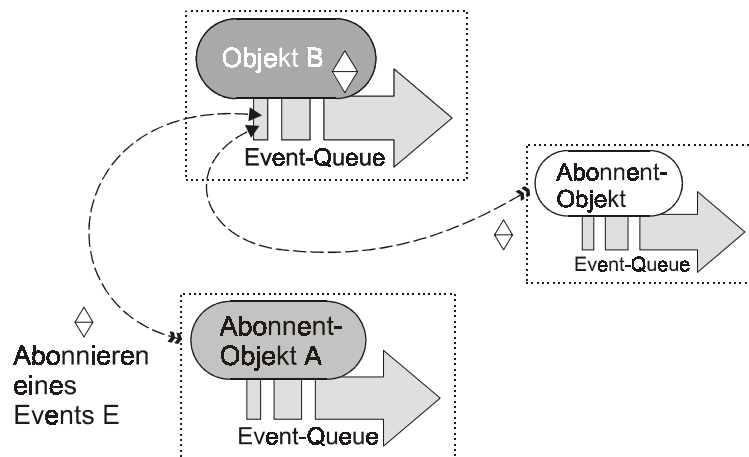


Abbildung 5-5: Abonnieren von Events

Jedes sTeam-Objekt besitzt eine eigene Event-Queue²⁵⁷ (vgl. Abbildung 5-5), d.h. in einem Objekt ausgelöste Events werden zunächst über die lokale Event-Queue an alle Abonnenten verteilt. Gleichzeitig wird jedes Event an die zentrale globale Event-Queue übermittelt. Eine zentrale Instanz der Server-Architektur ist damit die globale Ereignisverarbeitung (globale Event-Queue). Die globale Event-Queue ist als Singleton²⁵⁸ realisiert und ist innerhalb des Kernservers angesiedelt. Die globale Event-Queue ermöglicht es, insbesondere Sicherheitsüberprüfungen vornehmen zu können, d.h. jeder ausgelöste Event wird an ein Sicherheitsobjekt (Security-Objekt) übermittelt. Events werden unmittelbar im Moment ihres Eintreffens verarbeitet und an betroffene Objekte (Abonnenten) weitergeleitet (vgl. Abbildung 5-6).

Die sTeam-Architektur basiert auf einem flexiblen System von Events: Events werden durch beliebige Objekte ausgelöst und können von anderen beliebigen Objekten verarbeitet werden.

Die Unterscheidung zwischen globalen und lokalen Events ist für die Systemarchitektur entscheidend. Events sind in jedem einzelnen Objekt getrennt registriert, d.h. will ein Objekt „A“ einen Event „E“ eines anderen Objekts „B“ verarbeiten, trägt sich Objekt „A“ in die Event-Queue des Objekts „B“ für das Ereignis „E“ ein (vgl. Abbildung 5-5). Bei Auslösen des Events „E“ wird eine zu spezifizierende „notify“-Funktion in-

²⁵⁷ Der Begriff einer „Queue“ ist in diesem Zusammenhang etwas missverständlich gewählt, Ereignisse werden unmittelbar im Moment des Eintreffens ausgelöst. Da es sich in diesem Bereich des Servers um ein Ein-Prozess-Modell handelt, müssen eintreffende Ereignisse nicht in einer Warteschlange verwaltet werden. Im Falle einer zukünftigen Multi-Prozess-Implementierung des Kernservers ist allerdings eine echte Warteschlange notwendig.

²⁵⁸ Ein Singleton steht für das Design Pattern eines Objekts, welches in genau einer Instanz vorliegt und globale Zugriffsfunktionen besitzt (vgl. [Gamma et al. 2000, S. 127ff.]).

nerhalb des Abonnenten-Objekts „B“ aufgerufen und für jeden Event spezifische Parameter übermittelt.

Jedes lokale Event löst gleichzeitig ein globales Event aus und kann damit von globalen Objekten (Modulen), wie z.B. dem Sicherheitsmodul, verarbeitet oder abgefangen werden. Globale Events werden automatisch an alle Objekte der Umgebung übermittelt, d.h. alle Objekte verarbeiten gegebenenfalls alle eintreffenden Ereignisse. Hierbei erfolgt eine Sicherheitsüberprüfung zum Zeitpunkt der Registrierung eines Events. Beispielsweise benötigt das hörende Objekt Leseberechtigung zum Empfangen eines Events und Schreibberechtigung zum Blockieren (Abfangen) eines Events.

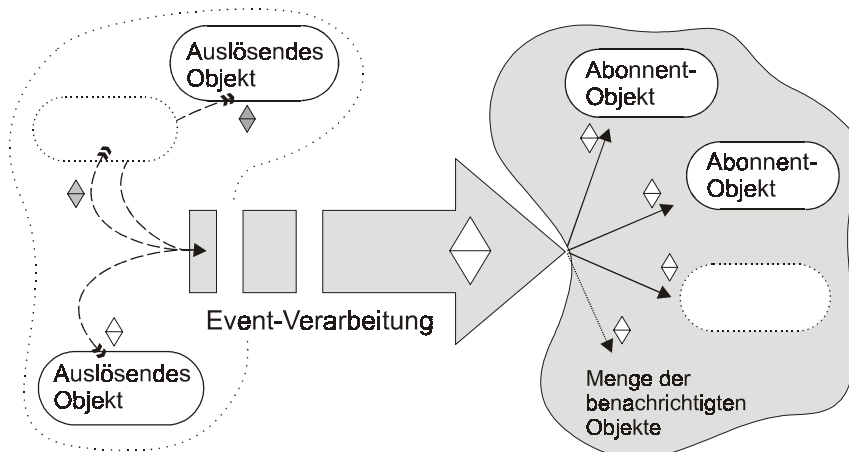


Abbildung 5-6: Benachrichtigung der Abonnent-Objekte

Basierend auf der Möglichkeit, Events flexibel an beliebigen Objekten registrieren zu können, werden wesentliche Interaktions-Mechanismen zwischen Objekten realisiert. Hierzu zählen alle Funktionen im Umgang mit Materialien (Bewegen, Verschieben von Objekten), aber auch Attributänderungen an Objekten und Kommunikationsfunktionen zwischen Objekten.

Das Auslösen eines Events geschieht in drei Phasen (vgl. Abbildung 5-7). Beispiel sei das Verschieben eines Objekts in einen Container.

- I. Die Try-Phase übermittelt den Wunsch zur Auslösung des Events an alle Abonnenten des Events bzw. an globale Sicherheitsmodule wie das Security-Modul. Bereits zu diesem Zeitpunkt kann das Auslösen des Events ggf. verhindert werden. Es wird das Recht zum Verschieben des Objekts in den entsprechenden Container geprüft (Berechtigung des aktiven Nutzers und Berechtigung des ausführenden Objekts).
- II. Die Run-Phase führt die eigentliche, mit der Auslösung des Events verbundene Aktion aus. Beispielsweise wird nun ein Objekt in einen Container bewegt.
- III. Die Notify-Phase schließlich benachrichtigt alle Abonnenten des entsprechenden Events über die Ausführung der mit dem Event verbundenen Aktion. Auf das Beispiel bezogen haben verschiedene Clients den obigen Event zur Verschiebung eines Objekts abonniert und können nun ihre Darstellungen aktualisieren.

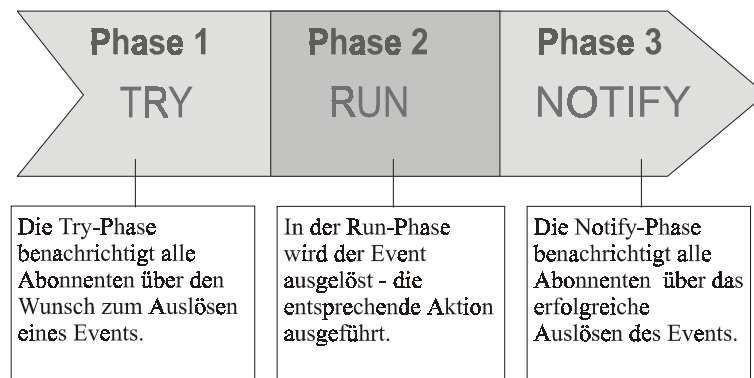


Abbildung 5-7: Phasen der Auslösung eines Events

Die sTeam-Kernarchitektur besitzt bereits eine ganze Anzahl von festgelegten Events. (Diese können mit einfachen Mitteln um weitere spezielle Ereignisse ergänzt werden.)

Zunächst festgelegte Events sind:

- `enter_inventory`, `leave_inventory` für das Betreten und Verlassen eines Objekts in die Umgebung eines anderen Objekts,
- `attributes_change` und `attributes_lock`, `set/get_attribute` für den Zugriff auf Attribute eines Objekts,
- `upload` und `download` für das Einfügen von Daten-Objekten in den Server,
- `move`, `delete`, `duplicate`, `arrange_object` als elementare Manipulationsfunktionen der Position eines Objekts,
- `register_attribute` und `register_factory` zur Registrierung eines neuen Attributs bzw. zum Anlegen einer neuen Klasse,
- `say`, `tell` zur synchronen Kommunikation zwischen Nutzern.

Prinzipiell sind für alle Formen primärer Medienfunktionen und damit für sämtliche in Kapitel 3 identifizierten Teilaspekte kooperativen Arbeitens getrennt zu verarbeitende Ereignisse notwendig. Um eine zu große und unüberschaubare Anzahl möglicher Events zu vermeiden, werden eine Reihe von generischen Events gewählt. Ein Beispiel aus der obigen Liste ist der `attributes_change`-Event, welcher sich auf sämtliche Attributtypen bezieht. Aus selbigem Grund nimmt das `execute`-Event eine besondere Stellung unter den definierten Ereignissen ein. Es findet in unterschiedlichen Kontexten Verwendung, z.B. im Zusammenhang mit der Ausführung von Skripten und dem Erzeugen neuer Objekte in einer Factory. Um bei einem Skript²⁵⁹ seine Ausführung zu bewirken, ist `execute` die einzige Funktion, die von einem Client aufgerufen werden kann. Das zentrale Sicherheitsobjekt (Security) verarbeitet die Auslösung eines `execute`-Events. Auf diese Weise wird überprüft, inwieweit der aufrufende Benutzer über das notwendige Ausführungsrecht an dem Objekt (Skript) verfügt.

Ähnlich wird im Falle der Factories das `execute`-Event benutzt, um zu entscheiden, ob ein Benutzer das Recht besitzt, Instanzen der jeweiligen Klasse zu erzeugen. Anschließend kann es analog genutzt werden, um die neu entstehenden Objekte zu überwachen. Die Bedeutung des `execute`-Events ist damit von der Klasse abhängig.

²⁵⁹ Skripte sind LPC-Programme, über die beispielsweise die Funktionalität der Webschnittstelle bereitgestellt wird.

Es kann eine Funktion im Sinne von „erzeugen“, aber auch „ausführen“ besitzen und kontrolliert sehr allgemein das Erzeugen neuer Objekte, aber auch den Umgang mit existierenden Objekten.

Im Gegensatz zu einem starren Ereignissystem besitzt der vorgestellte Ansatz die Möglichkeit, den bestehenden Events beliebig neue Ereignisse zuzufügen. Damit können serverseitige und clientseitige Ereignisse definiert und verarbeitet werden. Mit ihrer Hilfe lassen sich spezifische Kooperationsstrukturen architektonisch abbilden. Speziell im Rahmen des im Server verfolgten Modulkonzepts können hinzugefügte Module weitere Ereignisse definieren und der globalen Ereignisverarbeitung zuführen. Des Weiteren können verschiedene Clients zusätzliche Ereignisse notwendig machen.

Als weiteres, wesentliches Merkmal des vorgestellten Ansatzes gegenüber herkömmlichen ereignisbasierten Systemen können Ereignisse durch das Konzept der lokalen Events nur an eine begrenzte Anzahl von Objekten, den Abonnenten des Events, übermittelt werden. Die Übermittlung eines Events an eine ausgewählte Anzahl von Objekten (Einschränken der Reichweite eines Events) ist ein wichtiges Kriterium für die Skalierbarkeit des Gesamtsystems. Nehmen beispielsweise eine große Anzahl Nutzer an einer kooperativen Sitzung teil, ist es aus Gründen der Effizienz unumgänglich, Events nur auf die an kooperativen Handlungen teilhabenden Objekte (Nutzer, Dokumente) einzuschränken.

Gleichzeitig erlaubt die feine Granularität der Reichweite eines Events (Objekte müssen lokale Events explizit abonnieren), Strukturen kooperativer Handlungen unmittelbar aufseiten der Architektur umzusetzen. Beispiele sind Gruppen von Diskussionspartnern bzw. ein Chat, der sich thematisch an ein Dokument anschließt. Eine Chat-Gruppe lässt sich direkt durch das Abonnieren von Chat-Events für eine Anzahl von Nutzerobjekten an ein Objekt binden (z.B. alle Benutzer innerhalb eines Raums oder global alle Objekte der Umgebung). Damit können von den Clients so genannte Listener (Methoden, welche bei Auslösung eines Events aufgerufen werden) an ein beliebiges Objekt angefügt werden und die für eine spezifische Funktionalität notwendigen Aktionen vornehmen.

Serverseitige Nutzung von Events

Sämtliche internen Strukturen des Servers sind ebenfalls ereignisgesteuert realisiert. So abonniert z.B. das Modul Security, die zentrale Sicherheitsinstanz, alle globalen Events und leitet sie nach einer notwendigen Sicherheitsprüfung an die entsprechenden Objekte weiter. Das gesamte Konzept der Überprüfung von Benutzerrechten basiert auf diesem Mechanismus.

Die Leistungsfähigkeit des Ereignissystems zeigt sich insbesondere in den Möglichkeiten des Schaffens neuer Werkzeuge oder Dokumenttypen. Kooperative Applikationen, wie beispielsweise ein Shared Whiteboard, lassen sich mittels eines leistungsfähigen Event-Systems leicht umsetzen.

Auf das obige Beispiel bezogen wird innerhalb eines virtuellen Raums ein neues Shared Whiteboard-Objekt erzeugt (z.B. abgeleitet von einem Container). Dieses verwaltet die kooperativ genutzten Grafik- und Dokumentobjekte. Gleichzeitig werden Methoden bereitgestellt, um auf die Daten innerhalb der Shared Whiteboard-Ablage zugreifen zu können. Benutzer, die an einer kooperativen Sitzung teilnehmen, registrieren über ihren Client, der für sie den Inhalt der gemeinsamen Zeichenfläche geeignet aufbereitet, alle Events an dem Shared Whiteboard-Objekt.

Greift nun ein Benutzer über einen Client auf den Datenbestand zu, zeichnet z.B. ein neues Element auf die Zeichenfläche, werden diese Zustandsänderungen über Events an alle Abonnenten, also an die Teilnehmer der Whiteboard-Sitzung umverteilt. Kooperative Applikationen sind in dieser Form maßgeblich durch einen gemeinsamen persistenten Datenraum und die Umverteilung von Ereignissen auf die jeweiligen Teilnehmer einer kooperativen Sitzung (Session) gekennzeichnet.

Speziell die nutzerseitige Programmierbarkeit, d.h. die Erweiterung einer kooperativen Umgebung um aktive Objekte (Werkzeuge) basiert maßgeblich auf der Registrierung von Events. So lässt sich beispielsweise ein Protokollwerkzeug zum Mitschnitt von Chat-Diskussionen innerhalb eines Raumes ebenfalls durch ein Container-Objekt realisieren, welches die entsprechenden Kommunikationsereignisse der Teilnehmer einer Diskussion abonniert und z.B. in Attributen oder Objekten ablegt.

Clientseitige Nutzung von Ereignissen

Das dargestellte Ereigniskonzept wird bis auf die Ebene des Clients konsequent fortgeführt.

Clients können analog zu beliebigen Objekten des Servers Events verarbeiten und Funktionen im Server aufrufen und damit Events auslösen. Hierzu wird ein spezielles COAL-gestütztes Server-API entworfen, welches Events über die sTeam-Connector-Objekte an den Java-Client weiterleitet (vgl. Abbildung 5-8).

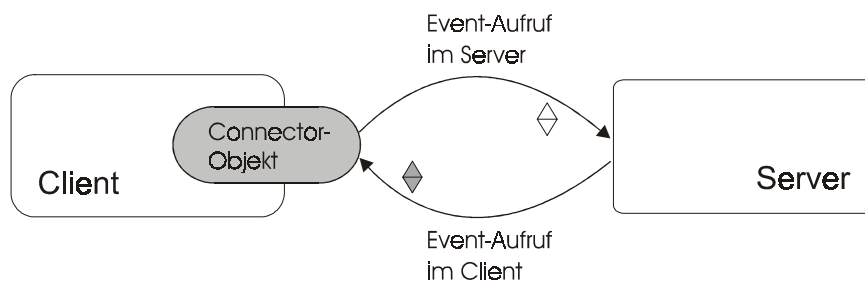


Abbildung 5-8: Connector-Objekte

Auf diese Weise gelingt eine vollständig objektorientierte Repräsentation von Server-Objekten innerhalb der Clients. Die Leistungsfähigkeit des Protokolls beeinflusst damit unmittelbar den Mechanismus der Übermittlung von Events zwischen Client und Server. Wichtig ist es, das Protokoll unabhängig von den transportierten Events zu gestalten. Damit lassen sich serverseitig neue Events definieren, ohne das Protokoll und die Umsetzung der Clients anpassen zu müssen. Genauso vermögen Clients zusätzliche Events innerhalb des Servers zu abonnieren, ohne ihre Kommunikationsschicht zum Server anpassen zu müssen.

Auch für die Performanz der gesamten kooperationsunterstützenden Umgebung ist die Umsetzung der Ereignisverarbeitung in der Client-Server-Schnittstelle von vorrangiger Bedeutung. Hier gilt es beispielsweise durch eine gebündelte Übermittlung der Events von und an den Server, diesen nicht durch ein massives Auslösen von Events zu blockieren. Ein derartiger Effekt tritt z.B. bei dem Raumwechsel eines Benutzers innerhalb des Clients auf. Werden alle Anfragen zur Attributänderung separat, in nicht gebündelten Anfragen gestellt, kann es schnell zu einer Überlastung in der serverseitigen Ereignisverarbeitung kommen. In diesem Zusammenhang können spezielle Cache-Mechanismen entscheidende Verbesserungen in der Geschwindigkeit der Verarbeitung bewirken.

Vorteile des objektorientierten Designs, welche oftmals reinen Java-Client-Server-Lösungen zugesprochen werden, können so in der hybriden sTeam-Architektur umgesetzt werden.

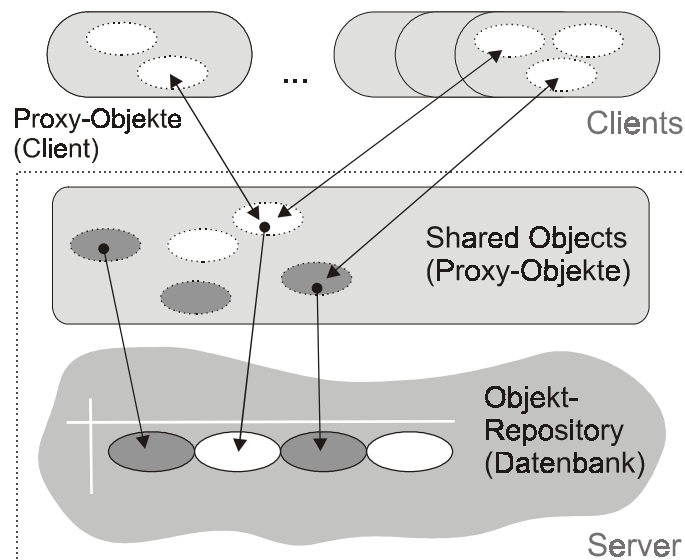


Abbildung 5-9: Client- und serverseitige Proxy-Objekte

5.2.6 Persistenzebene – Datenbankschnittstellen

Maßgebliches Kriterium gemeinsam genutzter, kooperativer Materialien ist ihre dauerhafte und gesicherte Ablagemöglichkeit. Damit resultiert die Anforderung der Persistenz unmittelbar aus dem Anspruch der Bereitstellung der primären Medienfunktionen.

Wie schon in Kapitel 4 gezeigt, lösen existierende kooperationsunterstützende Systeme das Problem der Persistenz von Objekten sehr unterschiedlich. Einige sitzungsorientierte Systeme erlauben keinerlei gemeinsame Ablage von Objekten auf einem zentralen Server, Objekte können in vielen Fällen lediglich auf den Client-Rechnern der Nutzer gespeichert werden (vgl. die Beschreibung des Habanero-Systems in Abschnitt 2.7.2). Andere Systeme nutzen ein einfaches Dateisystem zur dauerhaften Archivierung der Daten einer gemeinsamen Zusammenarbeit. Nur in wenigen Fällen erfolgt diese Speicherung mittels einer Datenbank.

Eine der wesentlichen architektonischen Grundlagen eines jeden verteilten objektorientierten Systems ist damit die Umsetzung der Persistenzebene für die Objekte des Servers. Wurden in deklarativen Ansätzen Funktionen genutzt, um einzelne Datenfelder in Dateien oder Datenbanken zu speichern, verlangt ein objektorientierter Ansatz eine neue, erweiterte Form von Persistenz. Ein Objekt ist mit allen existierenden Attributen und Methoden in einen persistenten Zustand zu überführen.²⁶⁰ In diesem Zusammenhang ist insbesondere das Problem der Vererbung von Attributen und Objekten zu lösen.

Es ergeben sich eine Reihe von Möglichkeiten Persistenz für Objekte eines Systems zu erzielen. Auf ein kooperationsunterstützendes System bezogen bedeutet Persistenz, die Objekte des Servers dauerhaft in einer Datenbank abzulegen, diese bei Bedarf jeder Zeit flexibel aus der Datenbank bereitzustellen und ihre Aktualisierung dynamisch vor-

²⁶⁰ In aktuellen sTeam-Prototypen werden zu jedem Objekt der Quelltext und zugehörige Attribute in der Datenbank abgelegt.

zunehmen. Hieraus ergibt sich eine Qualität der Transparenz für den Programmierer auf Objektebene. Die konkrete Ablage der Objekte in einer Datenbank wird von der Laufzeitumgebung realisiert.

Die sicherlich einfachste Lösung des Problems der Persistenz ist das Einfügen von SQL-Kommandos in die Klasse eines zugehörigen Objektes, um es bei Bedarf in einer Datenbank zu sichern. Dieser Ansatz ist mit geringem Aufwand umzusetzen, bedingt jedoch eine enge Kopplung des Objekt-Quelltextes an das vorhandene Datenbankschema, also der Organisation der Datenbank in Tabellen und Indexe. Auch die gängige Praxis der Kapselung der Persistenzfunktionen in eigene Klassen führt nicht zu einer echten Entkopplung der Persistenzschicht von den Objekten einer kooperativen Applikation. In einem derartigen Vorgehen ist es den Entwicklern beispielsweise nicht möglich, neue Klassen einer Applikation hinzuzufügen, ohne die Persistenzschicht anzupassen oder zu erweitern. Insbesondere das Hinzufügen von Klassen zur Laufzeit ist jedoch eine wesentliche Anforderung an die skizzierte Architektur.

Weiterhin von großem Interesse ist die Frage des Zeitpunktes der Persistenz-Werderung von Objekten. In einem frühen Prototypen des sTeam-Systems werden vergleichbar einem MUD bei Start des Systems sämtliche Objekte in den Speicher des Servers geladen. Das System arbeitet lediglich auf den im Speicher befindlichen Objekten; sie bilden die kooperative Applikation. In gewissen Zeitabständen, z.B. bei einem Neustart der Lernumgebung werden alle Objekte auf dem Dateisystem des Servers serialisiert. Eine derartige Vorgehensweise wird als „state-dump“ bezeichnet und bietet sich durch die Architektur des Systems als Verbund von kooperativen Objekten an. Leider ist die Anzahl der Objekte theoretisch durch den Arbeitsspeicher begrenzt. Der in dem frühen sTeam-Prototypen eingesetzte DGD-Driver bietet zwar die Möglichkeit des dynamischen, für den Nutzer oder Programmierer transparenten Auslagerns von Objekten aus dem Speicher, diese werden jedoch lediglich auf dem Dateisystem des Servers serialisiert.

Eine Lösung bietet der für die vorgestellte Architektur adaptierte und erweiterte Ansatz von Ambler. Es wird eine so genannte robuste Persistenzschicht („robust persistence layer“) [Ambler 2000a] eingeführt. Ziel dieser Vorgehensweise ist es, eine eigene Systemschicht bereitzustellen, welche die Persistenz der Objekte einer Applikation vollständig abdeckt. Die Implementierung der Objekte sollte unabhängig von der Realisierung der Persistenzschicht sein und unabhängig von dem verwendeten Datenbankschema sowie der verbundenen Datenbank erfolgen.

Idealerweise erlaubt es die Persistenzschicht nach Ambler durch Versenden von wenigen einfachen Nachrichten, Objekte persistent in der Datenbank abzulegen. Hierbei handelt es sich im Wesentlichen um die Befehle `create` zur Erzeugung eines neuen Objektes, `save` zur Speicherung/Ablage eines Objektes, `delete` zur Löschung eines Objektes und `retrieve` zum Laden eines Objektes aus der Datenbank. Die elementaren Funktionen der Datenbankschicht werden in der Literatur typischerweise, nach Hinzufügen einer Funktion `update` zur Aktualisierung von Objekten, mit dem Akronym „CRUD“ für Create, Retrieve, Update, Delete belegt.²⁶¹

Auf die vorgestellte Konzeption bezogen sind so genannte Proxy-Objekte das zentrale Element der Persistenzschicht. Proxy-Objekte entsprechen dem aus der objektori-

²⁶¹ Für eine ausführliche Beschreibung von weiteren Kriterien und Anforderungen an eine robuste Persistenzschicht und die sich hieraus ergebenden Implementierungsdetails vgl. [Ambler 2000a].

entierten Programmierung bekannten Proxy-Pattern (vgl. [Gamma et al. 2000, S. 207ff.]. Proxy-Objekte sind als Platzhalter für reale Objekte innerhalb des Servers angesiedelt und bilden die Schnittstelle zur Persistenzschicht des Servers. Die Proxy-Objekte innerhalb des Servers sind nicht zu verwechseln mit den durch Replikation in den Clients bereitgestellten Proxy-Objekten. In beiden Fällen bilden sie jedoch zunächst einen Platzhalter, eine leere „Hülle“ zum Zugriff auf das reale Objekt (vgl. Abbildung 5-9).

Proxy-Objekte bestehen prinzipiell nur aus der Server-eindeutigen Objekt-ID (OID) und einem Zeiger auf das wirkliche Objekt in der Datenbank. Durch die Persistenzschicht werden Objekte bei Ansprechen des Proxy-Objekts für den Programmierer bzw. Nutzer transparent aus der Datenbank bereitgestellt. Nach dem Laden des „echten“ Objekts werden Änderungen an dem Objekt ebenfalls transparent für den Nutzer in der Datenbank abgelegt. Sämtliche Zugriffe und Modifikationen an Objekten finden in dieser Form über die Proxy-Objekte statt. Für den Programmierer bzw. Nutzer ist es nicht notwendig zu wissen, inwieweit sich ein Objekt zum Zeitpunkt des Zugriffs im Speicher befindet oder es für ihn transparent bereitgestellt wird. In analoger Weise muss keine explizite Ablage von Objekten in der Datenbank initiiert werden.

Die Einführung von Proxy-Objekten gewährleistet eine gute Skalierbarkeit des Gesamtsystems. Es werden nur die jeweils benötigten Objekte in den Speicher geladen. Sie bieten in dieser Form eine elegante und einfache Möglichkeit, Persistenz für den Benutzer oder Programmierer unsichtbar bereitzustellen.

Die Abbildung von Objekten in ein relationales Datenbankmodell ist von sich aus ein Bruch von Paradigmen. Objektorientiertes Design basiert auf Paradigmen der Vererbung und Kapselung, während das relationale Datenbankmodell im Wesentlichen auf mathematischen Verknüpfungen von Mengenrelationen beruht. Trotzdem ist die Verwendung einer relationalen gegenüber einer objektorientierten Datenbank im Rahmen der vorgeschlagenen Konzeption ein vernünftiger Schritt:

Relationale Datenbanken haben eine weite Verbreitung gefunden und liegen in stabilen, robusten Versionen vor. Es sind eine ganze Reihe von professionell entwickelten Produkten frei verfügbar, wie z.B. die MySQL-Datenbank²⁶², die der GPL unterliegt. Dies kann im Bereich der objektorientierten Datenbanken nicht uneingeschränkt behauptet werden. Objektorientierte Datenbanken sind wenig standardisiert und liegen nicht als eigenständige Applikationen, sondern zumeist als Bibliotheken zu einer Anbindung an Java- oder C++-Programme vor.

Ambler demonstriert das Gelingen der Abbildung von objektorientierten Paradigmen in das relationale Datenbankmodell (vgl. [Ambler 2000b]). Hierbei bereiten insbesondere Konzepte wie multiple Vererbung innerhalb des relationalen Modells Probleme, diese müssen jedoch in der vorliegenden Konzeption nicht verwendet werden.

Auf die sTeam-Architektur bezogen genügt zunächst eine einfache Organisation sämtlicher Objekte in einer Reihe von Tabellen. Die einzelnen Tabellen sind nicht untereinander verknüpft. Zugriff auf die Tabellen erfolgt über entsprechende Schlüsselemente, z.B. ein eindeutiger Objektschlüssel (OID).

Insofern gestaltet sich die reine Abbildung unkompliziert, erweitertes Augenmerk ist jedoch auf den oben beschriebenen Mechanismus der Einführung einer robusten Per-

²⁶² Vgl. <http://www.mysql.org/>, Stand 1.12.2001.

sistenzschicht über Proxy-Objekte zu legen. Nur so lässt sich eine für den Nutzer transparente Persistenz von Objekten aller Teile des Systems erzielen.

Innerhalb des Servers erfolgt eine Verkettung von im Speicher befindlichen Objekten über die zugehörigen Proxy-Objekte. Sämtliche Zugriffe auf Objekte werden zunächst über die entsprechenden Proxy-Objekte vorgenommen. Diese delegieren Methodenaufrufe und Attributzugriffe an die realen Objekte weiter. Die Persistenzschicht sorgt für die Bereitstellung der Objekte aus der Datenbank. Werden Objekte aus dem Speicher entfernt, bleiben die Proxy-Objekte weiterhin im Speicher geladen, um einen erneuten Zugriff auf das Objekt zu ermöglichen. Die Persistenzschicht realisiert auf diese Weise einen effizienten Objekt-Cache.

Das zugrunde liegende Datenbankschema des Servers sei an dieser Stelle nicht weiter vertieft. Es enthält im Wesentlichen zwei Tabellen, die Objekt-Tabelle und eine Daten (Content)-Tabelle. Erstere speichert sämtliche Objekte des Systems, d.h. es werden zu jedem Objekt eine eindeutige Objekt-ID, der Pfad der abzuleitenden Klasse (URL) und eine Zeichenkette der serialisierten Attribute abgelegt (Object-ID – Class URL – serialized Attributes).

Die zweite Daten-Tabelle speichert zugreifbar über die Objektidentifikationsnummer die Daten eines Objekts in einer Liste von Datenblöcken, also den Inhalt beliebiger kooperativer Materialien (Object-ID, Data-blocks).

Eine Reihe von Datenbankindizes erlauben den Zugriff auf die gespeicherten Objekte. Diese sind speziell auf die Anforderungen der Object Request Broker (ORBs) zugeschnitten. ORBs stellen verschiedene Sichten auf die sTeam-Objektstruktur bereit und werden im Folgenden noch detailliert behandelt (vgl. Abschnitt 5.2.7). Für die beschriebene kooperative Lernumgebung notwendige Datenbankindizes dienen u.a. dem Abbilden von Beziehungen zwischen Benutzer-ID und Benutzer- oder Nutzergruppennamen und dem Zugriff auf Objekte über eine externe URL.

Proxy-Objekte bilden zugleich die notwendige Instanz zu einer Zugriffskontrolle auf Objekte im kooperativen Umfeld. Sie vermögen Konfliktsituationen im zeitgleichen Zugriff auf ein und dasselbe Objekt zu handhaben (Concurrency Control). In den meisten Fällen genügt ein einfaches Sperren eines Objektes während der Zeit eines Zugriffs. Bedingt durch nur kurze auftretende Zugriffszeiten auf einzelne Objekte und nur wenige nebenläufige Operationen genügt diese optimistische Strategie des Sperrens (Locking). Weiterführende Mechanismen, wie in Transaktionen gekapselte Zugriffe, sind nicht notwendig.

5.2.7 Objekt-Strukturen – Object Request Broker

Sämtliche Materialien wie Dokumente, Grafiken und auch die Benutzer selbst sind in Form von Objekten innerhalb des Servers repräsentiert. Eine erste Ordnung in einer derartigen freien Struktur bilden Räume und Container als Kapseln für Dokumente und Benutzer. Materialien und Nutzer sind auf diese Weise einer eindeutigen Position innerhalb eines virtuellen Raums zuordbar.

Jede Benutzergruppe und jeder Benutzer besitzen einen persönlichen Arbeitsraum (Workroom), welcher zugleich die Wurzel eines Baumes enthaltener weiterer Räume, Dokumente und Container ist (vgl. Abbildung 3-1). Dementsprechend existiert keine Hierarchie aus den Räumen eines sTeam-Servers, vielmehr ein „Wald“ aus einzelnen Bäumen. Der so genannte sTeam-Wurzelraum („root“-Raum) bildet eine eigene Wurzel eines Baumes. Er enthält technische Komponenten des sTeam-Systems und fungiert als

Wurzel für den Zugriff auf interne Skripte und Objekte (beispielsweise für FTP- oder HTTP-Zugriffe).

Räume sind durch Verbindungstüren mit anderen Räumen verknüpft. Diese können bidirektional, aber auch nur in einer Richtung begehbar sein.²⁶³

Die Eigenschaft, dass (bis auf die Wurzelräume) jeder Raum Unterraum eines Elternraumes ist, führt dazu, dass jedes Objekt eine eindeutige Umgebung (Environment) besitzt, aus der verschiedene Eigenschaften (Attribute) vererbt werden können (vgl. Abschnitt 5.2.3).

Es treten verschiedene Situationen auf, in denen die Struktur von Räumen und Objekten aus unterschiedlichen Sichtweisen organisiert vorliegen muss. Werden beispielsweise Dokumente über ein dateisystemorientiertes Protokoll wie FTP in den Server übertragen, ist eine gänzlich andere Struktur abzubilden, wie sie eine Suchabfrage nach Objekten innerhalb des Servers erforderlich macht.

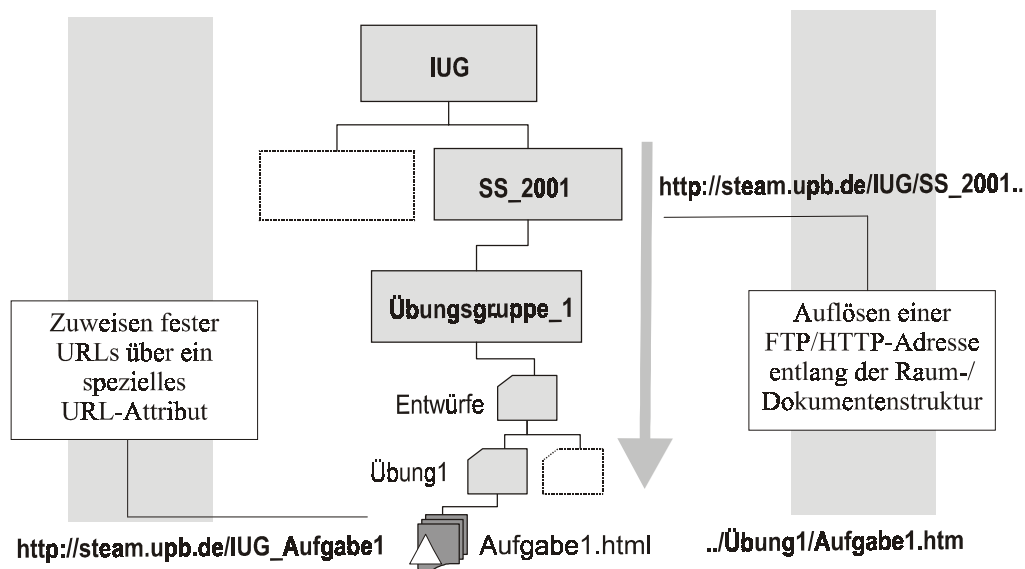


Abbildung 5-10: Auflösen von Pfaden aus der Raum-/Dokumentenstruktur

Die Aufgabe der Aufbereitung der Struktur unter verschiedenen Blickwinkeln und Sichtweisen übernehmen eigene Instanzen, die Object Request Broker.

Jeglicher Zugriff auf die Objektstruktur erfolgt über einen entsprechenden ORB. Es existieren zurzeit drei verschiedene ORBs:

- I. *Objekt-ID – Objekt:* Es wird eine Relation zwischen der Objektidentifikationsnummer und dem eigentlichen Objekt bereitgestellt. Eine derartige Zuordnung wird für die meisten Objektzugriffe innerhalb des Servers benötigt, z.B. im Umfeld der Suche von Objekten.
- II. *Objekt-URL – Objekt:* Der zweite ORB ordnet jedem Objekt eine durch die Position des Objekts eindeutig spezifizierbare URL zu, wie sie typischerweise auf gewöhnlichen Webservern Verwendung findet. Zu diesem Zweck können belie-

²⁶³ In der klassischen MUD-Konzeption sind Verbindungen zwischen Räumen zunächst nur in einer Richtung begehbar. Konzeptuell scheint diese Eigenschaft von Verbindungstüren im Umfeld kooperativer Wissensräume zunächst nicht erforderlich.

bige sTeam-Objekte mit einem URL-Attribut versehen werden, welches eine Art Alias für den Zugriff auf das Dokument über einen WWW-Client repräsentiert. Der URL-ORB ist insbesondere für die Ansprechbarkeit von Objekten über das WWW, z.B. im Rahmen der Realisierung der Webschnittstelle, notwendig.

- III. *Objekt-Environment – Objekt*: Ein spezieller ORB organisiert die Abbildung der sTeam-Struktur durch eine Hierarchie aus Räumen, Containern und Objekten. Hierbei besitzt jedes Objekt eine eindeutige Umgebung (Environment). Der Environment-ORB liefert zu jedem Objekt seine Umgebung. Verschiedene Clients nutzen ihn zur Darstellung der Raumstruktur (vgl. Abbildung 3-2).

Nur der ORB, welcher eine Zuordnung zwischen Objekten und ihrer eindeutigen Identifikationskennung (ID) bereitstellt, kann auf alle Objekte innerhalb des Servers zugreifen. Die hierarchische Raumstruktur und die Identifikation über Namens-URLs bilden jeweils nur einen Teil der Serverstruktur ab.

So wirkt die schon dargestellte Struktur aus Räumen, Containern und Objekten (vgl. Abbildung 3-1) als gemäß dem HTTP-/FTP-Protokoll konforme Adressierung der einzelnen Objekte. Pfade lassen sich durch eine entsprechende Enthaltensein-Beziehung auflösen. Es gilt also zwischen einem Objekt-Environment- und einem Objekt-URL-ORB zu differenzieren: Ersterer vermag Objekte resultierend ihrer Position innerhalb der Hierarchie aus Räumen und Objekten anzusprechen (Auflösen von Pfaden), letzterer adressiert ein Objekt über eine optional zugewiesene URL. Beide Darstellungsformen lassen sich unabhängig voneinander auf verschiedenen Netzwerkadressen (Ports) anbieten (vgl. Abbildung 5-10).

Durch das Konzept der Object Request Broker ergibt sich eine einfache und effiziente Möglichkeit, Objekte unter sehr unterschiedlichen Blickwinkeln zu betrachten. Mittels zusätzlicher ORB lassen sich spezielle Kompatibilitätsschnittstellen zu existierenden Systemen und Protokollen schaffen. In dieser Form macht eine zusätzliche Schnittstelle des Servers, beispielsweise eine WebDAV-Schnittstelle (vgl. Abschnitt 5.2.8), lediglich die Anpassung oder das Hinzufügen eines entsprechenden ORB notwendig.

5.2.8 Schnittstellen des Servers – sTeam-API – COAL

Zu einer Kommunikation zwischen Server und synchronen Clients wurde im Rahmen der sTeam-Architektur das TCP/IP-basierte COAL-Protokoll entwickelt. Es ist speziell zu einer effizienten, ereignisbasierten Kommunikation zwischen Client und Server konzipiert, dient aber auch der Verbindung zwischen sTeam-Server und einem Webserver (in diesem Fall dem Roxen-Webserver).

In Verbindung mit einem speziellen Java-API realisiert COAL eine Form des Objektzugriffs (Methodenaufrufs) über das Netzwerk, kann also als ein auf RMI (Remote Method Invocation)²⁶⁴ basierendes Protokoll charakterisiert werden. Damit ähnelt COAL konzeptionell dem CORBA-Standard. CORBA erlaubt die Inter-Applikationskommunikation unabhängig von verwendeten Programmiersprachen und Plattformen. Hierzu werden Applikationen basierend auf CORBA-Objekten aufgebaut, welche ihrerseits Schnittstellen (Interfaces) in IDL (Interface Definition Language), der CORBA-

²⁶⁴ RMI bezeichnet in diesem Zusammenhang das grundsätzliche Paradigma von Methodenaufrufen in verteilten Systemen (Distributed Objects). Es darf nicht mit einer konkreten Implementierung von RMI in einer konkreten Programmiersprache, z.B. Java-RMI, verwechselt werden.

Schnittstellenspezifikationsprache, definieren. Verteilte Zugriffe auf Schnittstellen von CORBA-Objekten erfolgen über RMI. Damit erlaubt CORBA den Zugriff auf Objekte einer Applikation, von der zum Zeitpunkt der Kompilierung nur abstrakte Schnittstellen bekannt sind.²⁶⁵

Entsprechend RMI erlaubt die COAL-Schnittstelle das direkte Ansprechen von Server-Objekten durch den Client. Aufseiten des Clients organisiert ein Verbindungsobjekt, der so genannte `sTeam-Connector`, den Aufbau und die Aufrechterhaltung einer Verbindung zum Server (vgl. Abbildung 5-8).

Die Kommunikation zwischen Server und Client erfolgt über Nachrichten (Messages). Hier sind nur wenige elementare Nachrichten definiert. Diese umfassen beispielsweise das An- und Abmelden (`login`, `logout`), den Up- und Download (`upload`, `download`) und eine Übermittlung von Fehlermeldungen (`error`). Eine besondere Bedeutung kommen den Nachrichten zum Erfragen möglicher Befehle (`query_commands`), dem Kommando (`command`) und dem Ereignis (`event`) zu.

Über die Kommandos `query_commands` und `command` können Methoden in Objekten des Servers direkt aufgerufen werden. Der Client ermittelt über `query_commands` mögliche Methoden (Befehle) eines Objekts und ruft diese über `command` in Verbindung mit einer Reihe von Attributen auf.

Damit besteht jede an den Server übermittelte Message aus dem Nachrichten-Typ, einem Objekt (bzw. der ID), auf die sich die Nachricht bezieht, und einer Anzahl von Daten (Attributen).

Eine zugehörige Kommunikationsschicht (API) sorgt für eine Kapselung von Nachrichten in Transaktionen. Jede Nachricht erhält eine eindeutige Transaktionsidentifikationsnummer. Elementare Zugriffe auf den Server sind zur Vermeidung von Synchronisationsproblemen und Zugriffskonflikten in Transaktionen zu kapseln (vgl. [Roseman & Greenberg 1993]).

Über die Nachricht `event` werden von Seiten des Servers Events an den Client übermittelt. Clients müssen zuvor ein Event für ein Objekt abonnieren und erhalten im Falle des Eintretens eine Nachricht mit den zugehörigen Parametern, wie die Objekt-Nummer und die Parameter des Ereignisses. Das Java-API stellt zur vereinfachten Handhabung von Events so genannte `steamListener`-Objekte bereit. Diese können mit beliebigen Objekten verknüpft sein und werden im Falle des Eintretens eines entsprechenden Events aktiviert.

Eine Verarbeitung von Nachrichten im Server erfolgt in der Reihenfolge des Eintreffens. Einkommende Nachrichten werden von verschiedenen Verbindungsobjekten entgegengenommen und in einer zentralen Ereignisschleife innerhalb des Servers verarbeitet, d.h. an die entsprechenden Serverobjekte weitergeleitet. Ein derartiger Mechanismus entspricht der Vorgehensweise innerhalb eines typischen Ein-Prozess- (single threaded) Servers wie z.B. ein HTTP- oder FTP-Server.

Bedingt durch die Grundarchitektur des Servers als Ereignis-Schleife (Event-Loop) müssen Kommandos kurz sein, d.h. sie dürfen keine langen Verarbeitungsprozesse im Server erfordern. Ein Aufruf, welcher zu viel Zeit benötigt, würde andere Aufrufe im Server blockieren.

²⁶⁵ Zu einer detaillierten Übersicht zum CORBA-Standard vergleiche [Coulouris et al. 2001].

In diesem Punkt sind aufwändigere, nebenläufige Architekturösungen denkbar. Diese sind komplex in ihrer Umsetzung und erfordern insbesondere einen hohen Aufwand in der Verwaltung der Nebenläufigkeitskontrolle.

Eine ganze Reihe von Situationen innerhalb des Clients machen eine große Anzahl von Anfragen und damit Nachrichten an den Server erforderlich. Hierzu zählen beispielsweise das Betreten eines neuen Raums oder das Öffnen eines Containers, welcher eine beträchtliche Anzahl von Objekten enthält.

Aus Gründen der Geschwindigkeitsoptimierung lassen sich derartige Anfragen zu einer komplexeren Anfrage bündeln. Gleichzeitig können Anfragen von der Kommunikationsschicht, dem API, gesammelt an den Server geleitet werden, ohne zunächst die Reaktion des Servers auf eine frühere Anfrage abwarten zu müssen.

5.3 Client

Im Folgenden wird ein kurzer Überblick über die Fortsetzung der vorgestellten Architektur aufseiten der Clients gegeben. Die vorliegende Arbeit konzentriert sich auf die Konzeption einer Kernarchitektur für die Bereitstellung kooperativer Wissensräume. Der Client ist stark von den Merkmalen des Servers beeinflusst, kann aber, wie in Kapitel 3 gezeigt, sehr unterschiedliche Darstellungsformen entwickeln. Aus diesem Grunde erfolgt zunächst eine konzeptionelle Beschreibung der clientseitigen Fortsetzung der gefundenen Architektur, ohne auf spezifische Details in der Realisierung der einzelnen Clients eingehen zu wollen.

Der vorliegende Abschnitt umreißt zunächst verschiedene Client-Realisierungen (Abschnitt 5.3.1), anschließend werden Alternativen der Verknüpfung von Browser-Technologie und synchroner sTeam-Clients vorgestellt (Abschnitt 5.3.2). In diesem Zusammenhang werden spezielle Konzepte verschiedener Proxy-Server und der Annotea-Standard dargestellt. In Abschnitt 5.3.3 werden die Integrationsbemühungen verschiedener Client-Technologien diskutiert, hierzu zählt die Bereitstellung einer Web-Schnittstelle. Abschließend wird die Architektur eines synchronen (Java-) Clients untersucht (Abschnitt 5.3.4).

5.3.1 Client-Realisierungen

Als Clients sind alle Applikationen zu kennzeichnen, die eine Verbindung mit dem Server aufnehmen. Hierzu zählen Webbrowser, spezifische synchrone Clients (Java), aber auch FTP- oder WebDAV-Werkzeuge. Da der Server einen Teil seiner Funktionalität in Form von gängigen Internetdiensten abbildet, können weitere, verbreitete Clients genutzt werden. Beispiele sind eine IRC-, POP3-, Annotea- oder News-Schnittstelle (vgl. Abbildung 5-11).

Clients besitzen eine spezifische Ausrichtung und bilden die eigentliche Benutzerschnittstelle kooperativer Wissensräume. Sie sind insbesondere nach dem Grad ihrer Unterstützung synchroner Funktionalität und nach der Art ihrer Verbindung zum Server (verbindungslos – sitzungsorientierte Verbindung) zu differenzieren. Es sind demgemäß die Klassen synchron, asynchron sowie Mischformen auszumachen (vgl. Abbildung 5-12).

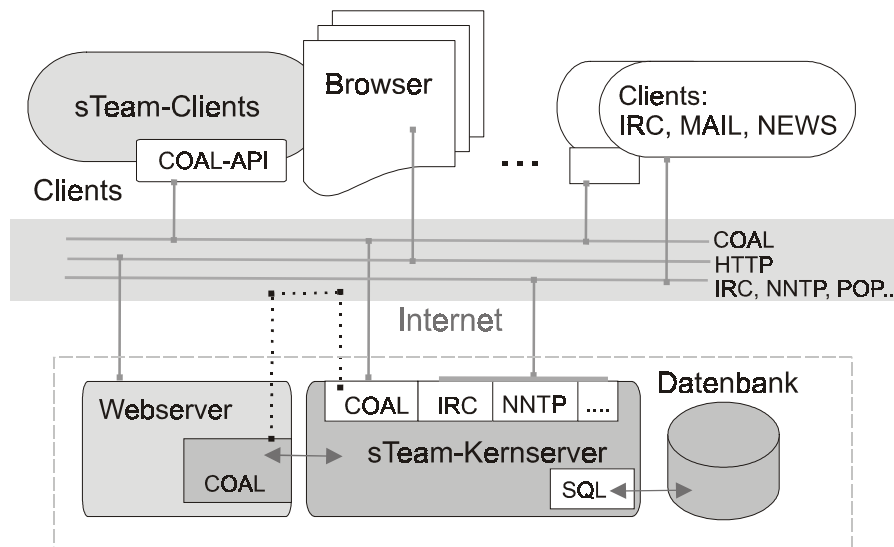


Abbildung 5-11: Client- Server-Schnittstellen

Von Seiten der Anwendung lassen sich die folgenden Grundklassen von Clients nennen:

- *Asynchrone WWW-Clients*: WWW-basierte Clients greifen über eine Web-Schnittstelle auf die kooperative Lernumgebung zu. Begründet durch die Verbindungslosigkeit des verwendeten HTTP-Protokolls ergibt sich eine eingeschränkte Funktionalität derartiger Clients speziell in den Bereichen synchroner Kommunikation und Interaktion. Die sTeam-Web-Schnittstelle wird über den Roxen-Webserver in Verbindung mit einer Anzahl von Pike-Skripten realisiert (vgl. Abschnitt 5.3.3).
- *Synchrone Java-/C++-Clients*: In Verbindung mit einem Java-API bilden synchrone Java-Clients die gesamte synchrone und asynchrone Funktionalität des sTeam-Servers ab. Spezielle, synchrone Kommunikations- und Interaktionsformen wie Chat oder ein Shared Whiteboard lassen sich auf diese Weise umsetzen. Synchrone Clients erhalten über das API und die COAL-Schnittstelle eine direkte Verbindung zum Server. Es können Nachrichten versendet und Events von Objekten des Servers empfangen werden. Synchrone Clients replizieren eine Anzahl genutzter Server-Objekte.
- *Dateisystem-orientierte Clients*: Zum vereinfachten Datenaustausch mit existierenden Applikationen ist es unerlässlich, den Server mit standardisierten Schnittstellen zur Datenablage auszustatten. In der vorgeschlagenen Architektur des Servers ist die Implementierung von FTP- und WebDAV-Protokollen vorgesehen. Mit ihrer Hilfe können beliebige FTP-Clients (der Microsoft Explorer bietet beispielsweise FTP-Funktionalität) oder WebDAV-fähige Applikationen (z.B. Microsoft Office-Produkte) zur Ablage von Objekten auf dem Server genutzt werden.
- *News-Clients*: Die Bereitstellung des NNTP-Protokolls (Network News Transport Protocol) bietet die Möglichkeit, mittels eines verbreiteten News-Readers (z.B. Netscape Navigator, Microsoft Outlook) auf Objekte ähnlich einer Diskussionsgruppe (Newsgroup) zuzugreifen und Annotationen an den Objekten zu erstellen. Annotationen werden in dieser Form als asynchrone Diskussionsgruppen interpretiert.

- *Synchrone Chat-Clients (IRC)*: Die Server-Architektur sieht Kompatibilität zu einem IRC-Server vor. Ein beliebiges Objekt, beispielsweise ein virtueller Raum, kann zugleich einen IRC-Kanal repräsentieren. Der sTeam-Benutzername und das Kennwort entsprechen der einzutragenden IRC-Kennung. Damit kann neben der Nutzung spezifischer synchroner Clients auch über einen gewöhnlichen IRC-Client an einer synchronen Kommunikation innerhalb eines virtuellen Raums teilgenommen werden.
- *E-Mail-Clients*: Durch Implementierung eines E-Mail-Protokolls (POP/IMAP) können Nutzer über ein gewöhnliches E-Mail-Programm (z.B. Pine, Netscape Navigator, Microsoft Outlook etc.) direkt auf ein persönliches Postfach in ihrem Rucksack zugreifen. Auf diese Weise können Nachrichten an einen Nutzer versandt werden und per E-Mail-Werkzeug gelesen werden, aber auch Objekte aus diesem Postfach „abgeholt“ werden.
- *Annotea-Clients*: Im Rahmen der „Semantic Web Activity“ des W3C ist ein Standard zur Annotation von Dokumenten auf Webservern in Vorbereitung bzw. Erprobung.²⁶⁶ Verhält sich der Server gemäß den festgelegten Annotea-Standards, können auch von fremden Annotea-Clients Annotationen auf einem sTeam-Server abgelegt werden. Zurzeit existiert lediglich der experimentelle Amaya-Browser als Annotationswerkzeug, welches den Annotea-Standard unterstützt.²⁶⁷

Eine vertiefende Beschreibung möglicher Clients ist gemäß der Ausrichtung der vorliegenden Arbeit auf die Kernarchitektur der sTeam-Umgebung im Folgenden nicht vorgesehen.²⁶⁸ Es folgt die Skizze des konzeptionellen Aufbaus eines synchronen Java-Clients und einer zugehörigen Webschnittstelle zur Darstellung WWW-basierter Lehrmaterialien. In diesem Zusammenhang sind insbesondere einige Bemerkungen zur architektonischen Einbindung eines Proxy-Servers und die Beachtung zukünftiger Annotea-Standards sinnvoll.

5.3.2 Verknüpfung: Browser – synchroner Java-Client – Proxy/Annotea

Grundlegendes Merkmal der clientseitigen Architektur ist eine enge Verzahnung verschiedener Client-Technologien. Aus der in Kapitel 2 geforderten durchgängigen Verfügbarkeit kooperativer Wissensräume ergibt sich die Notwendigkeit einer direkten Verknüpfung von WWW-Browsern und synchronen Clients. Verschiedene Anforderungen beeinflussen hierbei den Kopplungsgrad zwischen Browser und Client-Applikation. Für die Darstellung von Netzseiten, Grafiken und Dokumenten ist der Browser lediglich zur Anzeige des entsprechenden Objekts heranzuziehen (lose Kopplung). Für Bereiche des kooperativen Betrachtens von Materialien sind die Navigationselemente im Browser fernzusteuern (enge Kopplung), was eine spezielle Schnittstelle aufseiten des Browsers notwendig macht.

Szenarien der Aufzeichnung von Wegen, die Nutzer im Netz zurücklegen, erfordern die Verfügbarmachung und Übermittlung der aktuell im Browser angezeigten Netzseiten bzw. Navigationsschritte. Konzepte wie die Annotation von nicht auf dem Server

²⁶⁶ Vgl. <http://www.w3c.org/2001/Annotea>, Stand 1.12.2001.

²⁶⁷ Vgl. <http://www.w3c.org/Amaya>, Stand 1.12.2001.

²⁶⁸ Detaillierte technische Darstellungen Java-basierter Clients finden sich in [Sinemus 2000] und [Enklaar 2001].

abgelegten Websites machen weiterführende Mechanismen der Verarbeitung von Seiten notwendig.

Kopplung aus Client und Browser

Eine Kopplung des synchronen Clients mit dem Webbrowser ist über sehr unterschiedliche Technologien denkbar: Hierbei ergeben sich aus der technischen Lösung verschiedene Möglichkeiten der Synchronisation von Browserinhalten und seiner gezielten Steuerung durch den Client (z.B. im Anwendungsfall eines Shared-Browsers).

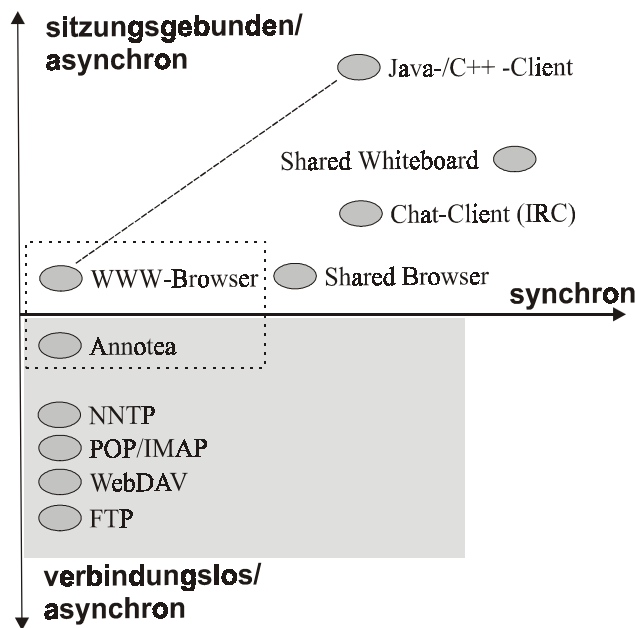


Abbildung 5-12: Klassifizierung von Client-Technologie

Erste Prototypen von sTeam-Clients sind sehr eng an den Browser gekoppelt und über eine spezifische Schnittstelle mit dem Netscape Navigator verbunden (LiveConnect). Später folgt die Integration einer Browser Java Komponente (Bean) in den Java-Client (vgl. [Sinemus 2000]). Der Browser wird Teil des Java-Clients, was insbesondere Probleme des nicht ausreichenden Leistungsumfangs und der Wartbarkeit der verwendeten Browser-Komponente aufwirft. Enklaar erprobt die Kopplung aus Browser und synchronem Java-Client über eine so genannte Browser-Launcher-API (vgl. [Enklaar 2001, S. 25]). Der Browser wird über eine speziell entwickelte Schnittstelle aus der Java-Applikation heraus gestartet.

Damit ergeben sich eine Reihe grundsätzlicher architektonischer Strategien zu einer Kopplung von Browser und synchronem Client bzw. der Kopplung von Browsern untereinander.²⁶⁹ Minimales Ziel ist in jedem Fall die Anzeige von Web-Seiten und extern abgelegten Dokumenten:

²⁶⁹ Anderweitige CSCL-/CSCW-Systeme lösen eine Kopplung von Browser und synchronem Client auf recht unterschiedliche Wege. McKinley et al. demonstrieren die Anbindung von Client-Applikationen durch eine (jedoch nur auf Microsoft Windows-Plattformen verfügbare) DDE-Verbindung (vgl. [McKinley et al. 1999]). Ihr Pavilion-System koppelt die Webbrowser der Kooperationspartner über eine DDE-Verbindung. Gleichzeitig werden lokale Proxy-Server bereitgestellt, um Dokumente im Browser anzeigen zu können. Das TANGO-System geht den umgekehrten Weg, hier wird der synchrone Anteil eines Clients direkt in den Browser integriert. Dies geschieht durch ein geeignetes Plugin/ ActiveX-Control (vgl. [Beca et al. 1997]).

- *Integration einer Browser-Komponente in einen spezifischen kooperationsunterstützenden Client:* Verschiedene, verfügbare Java- oder C++-Komponenten stellen Teile der Funktionalität eines Webbrowsers bereit. Sie eignen sich leider aufgrund fehlender Unterstützung aller Leistungsmerkmale eines herkömmlichen Browsers nur begrenzt zu einer vollständigen Implementierung als Teil eines kooperationsunterstützenden Clients.
- *Integration der Verbindung zu einem CSCL-Server in den Browser über ein Plugin bzw. ActiveX Control:* Synchrone kooperationsunterstützende Funktionalität wird direkt in den Browser integriert. Der Browser ist die Laufzeitumgebung des Clients eines CSCL-Systems. In naher Zukunft kann hierzu innerhalb des Mozilla-Browsers²⁷⁰ die XML-based User Interface Language (XUL)²⁷¹ Verwendung finden.
- *Ansprechen des Browsers über eine hierfür vorgesehene proprietäre Schnittstelle:* Es wird eine spezielle im Browser vorhandene Schnittstelle zu dessen Kopplung mit einer kooperationsunterstützenden Applikation genutzt (z.B. DDE, LiveConnect, Java BrowserLauncher etc.). Es existieren zurzeit keine plattformübergreifenden standardisierten Browserschnittstellen.
- *Integration einer Browser-Komponente in den kooperativen Client (z.B. durch eine geeignete Browser-Bean):* Der Browser wird selbst als Komponente in einen Client eingebracht. Ein herkömmlicher Browser eignet sich hierzu nicht, es muss auf speziell für diesen Zweck entwickelte Browser-Komponenten zurückgegriffen werden.
- *Nutzung eines lokalen an den Client oder Server gebundenen Proxy-Servers:* Serverseitige Dokumente werden über den Proxy-Server in dem verwendeten Browser dargestellt. Der Client veranlasst die Anzeige von Dokumenten über den Server. Der Browser muss in keiner Form modifiziert oder über eine spezifische Schnittstelle angesprochen werden. Es ergeben sich eine Reihe von Nachteilen aus der Verwendung eines Proxy-Servers. Über ihn müssen alle vom Browser gestellten Anfragen übermittelt werden.

Sämtliche der oben genannten Lösungen bieten eine Reihe von Vor- und Nachteilen. Mit dem Ziel, den Kopplungsgrad zwischen Client und Browser möglichst eng zu wählen, gleichzeitig aber auf proprietäre und eingeschränkte Lösungen zu verzichten, wurden eine Anzahl der oben gezeigten Varianten prototypenhaft umgesetzt (vgl. [Hampel & Bopp 2001]). Aus den gewonnenen Erfahrungen hat sich gezeigt, dass eine hybride Architektur verschiedener Lösungen sinnvoll erscheint.

Hierbei wird zunächst der Browser ohne eine implizite Kopplung an den Server angebunden. Resultierend können in einem ersten Schritt beliebige Seiten und Dokumente angezeigt werden. Zurzeit erfolgt die diesbezügliche Kopplung über die Java Browser-Launcher bzw. Java-Quickstart-Schnittstelle. Der Java-Client öffnet den Browser über eine Adresse (URL), die einem Dokument auf dem sTeam-Server entspricht. Der Roxen-Webserver ist für die Darstellung des Dokuments verantwortlich. In einem zweiten Schritt können Proxy-Lösungen zur Realisierung spezieller Funktionalitäten, wie dem Aufzeichnen von Wegen von Benutzern hinzugezogen werden.

Die architektonische Verbindung aus synchronem Client und einem Webbrowser löst eine Reihe grundsätzlicher Probleme, die speziell aus dem Umfeld der Darstellung

²⁷⁰ Vgl. <http://www.mozilla.org/>, Stand 1.12.2001.

²⁷¹ Vgl. <http://www.mozilla.org/xpfe/xulref/>, Stand 1.12.2001.

verschiedener WWW-gestützter Dokumentenformate erwachsen. Verbreitete WWW-Browser eignen sich durch eine große Anzahl verfügbarer herstellerepezifischer Plugins für die Anzeige unterschiedlicher, innerhalb der kooperativen Lernumgebung abgelegter Dokumentenformate.²⁷² Neben den üblichen HTML- und XML-Formaten erlauben eine Reihe von Browsern die Anzeige von PDF- oder Office-Dokumenten sowie verschiedener Grafikformate.

Architekturlösungen der Integration von Proxy-Servern – Anwendungen

Szenarien des gemeinsamen Betrachtens von Netzseiten (vgl. Abschnitt 2.5.2) lassen sich durch eine lose Kopplung aus Client und Browser nur bedingt lösen. Es besteht zunächst keinerlei Möglichkeit, die Navigationsschritte des Browsers (Navigationselemente) von außerhalb durch den Server zu beeinflussen.

Erfolgt die Navigation hingegen über den Client, beispielsweise durch die Anzeige von Objekten, die sich innerhalb eines virtuellen Raums befinden, sind die Sichten der kooperierenden Nutzer über den sTeam-Server synchronisierbar. Eine Synchronisation erfolgt über Events, die zwischen dem Server und den Clients ausgetauscht werden. Die explizite Aktualisierung der Browserinhalte wird anschließend durch die Clients vorgenommen. Eine Kopplung der Sichten von Nutzern, die sich nicht auf den innerhalb der kooperativen Lernumgebung abgelegten Materialien bewegen, ist auf diese Weise nicht möglich. Der Server vermag nur Sichten auf eigene Objekte und Referenzen auf WWW-Seiten zu synchronisieren. Bewegt sich der Nutzer auf beliebigen Netzseiten, erfolgt keinerlei Rückmeldung des Browsers über dessen aktuellen Standort. Informationen über Aktionen eines Nutzers im Browser können, ohne auf spezifische Schnittstellen des Browsers zurückgreifen zu müssen, lediglich über den Mechanismus eines Proxy-Servers ermittelt werden.

Eine ganze Reihe spezieller Eigenschaften kooperativer Wissensräume legen die Integration eines Proxy-Servers in die angestrebte Architektur nahe.²⁷³ Hierzu zählen Szenarien des gemeinsamen Betrachtens von WWW-Seiten und auf externen Servern abgelegten Lehrmaterialien. Auch die Annotation von beliebigen WWW-Seiten und geführte Wege durch das WWW (Trails) machen eine derartige architektonische Lösung notwendig.

Abgrenzung serverseitiger und clientseitiger Proxy

Sowohl der Server als auch der Client bieten mögliche Integrationspunkte eines Proxy-Servers. Der Proxy-Server ist in dem verwendeten Browser einzutragen und filtert fortan die dargestellten Seiten. Eine derartige Architektur löst das Problem der Möglichkeit der Beeinflussung von nicht innerhalb der Lernumgebung abgelegten Seiten. Durch den Proxy-Ansatz kann der Server auch WWW-Seiten aktiv beeinflussen, deren Objekte sich nicht auf dem sTeam-Server befinden.

In der ersten architektonischen Lösung erfolgt die Interpretation und Modifikation von Netzseiten durch einen direkt an den Server (bzw. den Roxen-Webserver) angebun-

²⁷² Vgl. <http://www.mozilla.org/docs/plugin.html>, Stand 1.12.2001.

²⁷³ Aus selbigem Grund greifen verfügbare Frameworks zur Konstruktion von Shared-Browser-Applikationen zumeist auf eine Proxy-Lösung zurück. Ein Beispiel, welches genau diesen Ansatz verfolgt, ist das Pavilion-System [McKinley et al. 1999]. Pavilion bietet die Anbindung eines Proxy-Servers an einen herkömmlichen Browser und die Kommunikation von verteilten (replizierten) Applikationen über ein zentrales Protokoll. Auf diese Weise lassen sich Applikationen entwickeln, die es erlauben, den Inhalt von Webbrowsern über das Netz zu anderen Teilnehmern einer kooperativen Sitzung zu übermitteln.

denen Proxy-Server. Hierzu ist das Authentifizierungsprotokoll für Proxy-Server an Webservern zu verwenden. Nutzer bewegen sich fortan über diesen Proxy-Server durch das Netz, der sTeam-Server vermag die Navigationsschritte des Benutzers zu erkennen und Seiten geeignet aufzubereiten. Auf diese Weise werden mit Annotationen versehene Seiten erkannt und korrekt dargestellt.

Mehr oder weniger aufwändig und nicht für jede Webseite lösbar ist das Problem der genauen Bestimmung des Ankerpunktes einer Annotation auf einer beliebigen WWW-Seite. Eine detaillierte technische Beschreibung des Konzepts der Annotation von beliebigen Seiten des WWW und den damit verbundenen Schwierigkeiten findet sich in [Tappe 2000].

Das Konzept der Aufzeichnung von Wegen von Benutzern im Netz, so genannte Trails, oder auch Darstellungen der Navigationsschritte eines Nutzers auf einer semantischen Karte erfordern ebenfalls die Integration eines Proxy-Servers in die angestrebte Gesamtarchitektur, legen jedoch die Integration eines lokalen, an den Client gebundenen Proxy nahe. Bei diesem Ansatz müssen nicht alle Nutzer einer Lernumgebung zwangsweise ihre Browser-Anfragen über den zentralen Proxy des Servers senden und empfangen. Ein lokaler Proxy befindet sich auf dem Rechner des Nutzers und nimmt durch den Client gesteuert die Aufbereitung der Seiten und Erfassung der Navigationsschritte vor, z.B. für die Aufzeichnung von Trails.

Die detaillierte technische Konzeption eines lokalen Proxy-Servers zur Anbindung an einen Client wurde in [Wiesner 2001] vorgenommen. Ein Pfad (Trail) durch das WWW wird mit Hilfe des lokalen Proxy aufgezeichnet und zusammen mit einer Reihe von Metadaten in einer XML-Repräsentation abgelegt. Anschließend kann der Trail editiert, z.B. mit Kommentaren versehen werden, und zur Navigation genutzt werden. Ähnliche Funktionalität lässt sich unmittelbar in einem Client zur skizzierten Lernumgebung integrieren. Trails werden als eigene Objekttypen im Server abgelegt und können wie jedes Objekt kooperativ genutzt werden.

Annotationen – Annotea-Ansatz

Neben einer proprietären Umsetzung von Annotationen auf beliebige Dokumente des WWW (z.B. über ein Proxy-Konzept) bildet der „Semantic Web“-Ansatz des W3C eine zukunftsweisende Alternative. Der in Vorbereitung befindliche offene Annotea-Standard sieht die Möglichkeit vor, jedes Dokument des WWW mit Kommentaren, Anmerkungen und persönlichen Notizen versehen zu können, ohne das entsprechende Dokument modifizieren zu müssen. Derartige Annotationen werden über die XML-Pointer-Language (Xpointer)²⁷⁴ in das jeweilige Dokument integriert und in Form von RDF-Metadaten (Resource Description Framework, vgl. [Bray 2001]) auf speziellen Annotationsservern (Annotea-Servern) gespeichert.

Der Annotea-Ansatz ordnet sich unmittelbar in die Bestrebungen des W3C ein, den primär leserzentrierten Umgang mit dem WWW hin zu aktiveren und autorenzentrierten Umgangsformen zu entwickeln. Der XML-Standard mit seiner Möglichkeit, bidirektionale Verweise in der XML-Linking-Language (XLINK)²⁷⁵ zu erstellen, oder die Standardisierung von Metadaten für das WWW (RDF-Standard) dokumentieren diesen Anspruch.

²⁷⁴ Vgl. <http://www.w3.org/TR/xptr/>, Stand 1.12.2001.

²⁷⁵ Vgl. <http://www.w3.org/TR/xlink/>, Stand 1.12.2001.

Für die Architektur des sTeam-Servers ergibt sich die Perspektive einen Annotea-Server zu implementieren. Dieser erlaubt standardisierte Annotation von WWW-Dokumenten und zugleich die Annotation der innerhalb der kooperativen Lernumgebung abgelegten Materialien. Architektonisch handelt es sich um eine Erweiterung des HTTP-Protokolls in Form eines RDF-Annotationsschemas.

Annotationen können mit einem beliebigen Annotea-Client vorgenommen werden. Zurzeit existiert lediglich das experimentelle Amaya-Annotationswerkzeug des W3C, es ist jedoch zu hoffen, dass verschiedene herkömmliche Browser ebenfalls den Annotea-Standard bereitstellen und damit zur Annotation genutzt werden könnten.

5.3.3 Integration verschiedener Clients

Architektonisch findet die Integration von WWW-Diensten und Standards in die angestrebte Lernumgebung seine Fortsetzung in der Bereitstellung der Webschnittstelle. Sie erlaubt den Zugriff auf einen maßgeblichen Anteil der asynchronen Funktionalität über einen herkömmlichen Webbrowser. Synchroner Clients nutzen die Webschnittstelle neben der Darstellung von Objekten (Materialien der Lernumgebung) auch als Benutzerschnittstelle für bereitgestellte Funktionalität kooperativer Wissensräume. Hierzu zählen beispielsweise Dialoge zur Änderung von Attributen oder Zugriffsrechten. Bedingt durch die Verbindungslosigkeit des HTTP-Protokolls ist die über die Webschnittstelle verfügbare Funktionalität vergleichbar einem rein asynchronen Dokumentenmanagement-System, wie z.B. dem Hyperwave-System (vgl. [Andrews et al. 1995]).²⁷⁶

Die Webschnittstelle wird über den mit dem sTeam-Server verbundenen Roxen-Webserver bereitgestellt. Sie bietet eine Reihe von Funktionen, die jedoch im Rahmen ihrer Architekturbetrachtung nicht detaillierter vorgestellt werden. Ziel ist es, weite Teile der Möglichkeiten einer Lernumgebung auch über einen herkömmlichen Browser verfügbar zu machen. Neben der Darstellung eines virtuellen Raums mit den enthaltenen Objekten und der Anzeige von Dokumenten sind auch elementare Informationen der gegenseitigen Wahrnehmung umsetzbar. Möglichkeiten wie das Editieren von Attributen an Objekten oder das Erzeugen neuer Objekte lassen sich in intuitiver Weise abbilden. Zu den wesentlichen Eigenschaften der Webschnittstelle zählen die folgenden Funktionen:

- Darstellung eines virtuellen sTeam-Raums als eine Liste von Objekten oder Ordern
- Darstellung von Materialien in dem Browser
- Navigation zwischen verschiedenen virtuellen Räumen über Verbindungstüren und direktes Ansprechen eines Raums über eine entsprechende URL
- Erzeugen von neuen virtuellen Räumen und Containern
- Erzeugen bzw. Einfügen neuer Objekte (Materialien)
- Erzeugen von Referenzen auf Materialien (auch außerhalb des Servers abgelegter Materialien)
- Einfügen und Anzeige von Annotationen zu beliebigen Objekten
- Manipulation von Materialien, wie z.B. Verschieben, Löschen und Kopieren von Objekten
- Modifikation von Benutzer- und Gruppenberechtigungen

²⁷⁶ Es existieren verschiedene Technologien, wie z.B. Cookies oder der Einsatz von Proxy-Servern, um die Verbindungslosigkeit begrenzt aufzuheben (vgl. Abbildung 5-12).

- Darstellung der Attribute zu Objekten (Materialien), Attributänderungen an Objekten
- Darstellung des Rucksacks eines Nutzers, Einfügen und Entfernen von Dokumenten
- Funktionen der Awareness – Anzeige der aktuell anwesenden Benutzer innerhalb eines virtuellen Raums
- Anmelden neuer Nutzer innerhalb eines Raums (nutzergestützte automatisierte Anmeldung)

Wesentliche Anforderungen der Webschnittstelle des sTeam-Systems resultieren aus dem Anspruch der durchgängigen Verfügbarkeit kooperativer Wissensräume. Die Webschnittstelle soll sowohl Anforderungen der Performanz, aber auch ausreichende Handlungsmöglichkeiten mit den Materialien eines virtuellen Raums bieten.

Hinzu kommt, dass eine Webschnittstelle mit begrenztem Aufwand an verschiedene Anwendungsszenarien anpassbar gestaltet werden muss. Dies umfasst sowohl das Erscheinungsbild und die Form der dargestellten Materialien als auch die Skalierbarkeit der Funktionalität. Letztere Anforderung ist nur durch ein ausgefeiltes Architekturkonzept zu leisten.

Architektur der Webschnittstelle

Architektonisch wird die Webschnittstelle des sTeam-Systems durch spezielle, für die Form der dargestellten Materialien verantwortliche Stylesheets und eine Reihe von Skripten realisiert. Die Darstellung und Verarbeitung der WWW-Zugriffe wird durch den mit dem sTeam-Server verbundenen Webserver umgesetzt.

Trotz seiner Sonderstellung als eigenständiger Webserver ohne direkte Eigenschaften eines Clients fungiert ein extern mit dem Server verbundener Webserver wie ein gewöhnlicher Client. Zugriffe auf den Webserver (Requests) werden interpretiert und in entsprechende Objektanfragen umgesetzt. Der sTeam-Server kontrolliert die Authentifizierung von Netzzugriffen, organisiert den präsentierten Inhalt von angeforderten Websites und simuliert eine Struktur der URL-Adressen (vgl. Abschnitt 5.2.7).

Entsprechend erzeugt der Up- und Download von Dokumenten Events im Server. Durch den universellen Charakter der COAL-Schnittstelle sind damit aufseiten des Servers keine spezifisch aufbereiteten Schnittstellen zur Anbindung des Webserver an den sTeam-Server notwendig. Lediglich der verwendete Roxen-Webserver ist durch ein geeignetes Erweiterungsmodul der COAL-Schnittstelle anzupassen (vgl. Abbildung 5-11). Dieses bildet den sTeam-Server in Form eines virtuellen Dateisystems ab. Anfragen des Webserver werden in Operationen an Objekten der Lernumgebung umgesetzt.

Architektonisch ist die Webschnittstelle durch eine Anzahl von LPC-Skripten und XSL-Transformations-Stylesheets (XSLT-Stylesheet)²⁷⁷ realisiert. Materialien bzw. Dokumente der Lernumgebung besitzen eine XML-Repräsentation ihres Inhalts und der Attribute des zugehörigen Objekts.

Aus der XML-Repräsentation eines Objekts und eines für das Erscheinungsbild der Webschnittstelle verantwortlichen XSLT-Stylesheets werden über einen XSLT-Prozessor HTML-Seiten generiert und auf dem Roxen-Webserver dargestellt.

Als XSLT-Prozessor eignet sich z.B. der Sablotron-XSLT-Parser²⁷⁸. Zukünftige Browser-Technologien, wie z.B. der Mozilla-Browser, besitzen einen integrierten

²⁷⁷ Vgl. <http://www.w3.org/TR/xslt>, Stand 1.12.2001.

²⁷⁸ Vgl. <http://www.gingerall.com/>, Stand 1.12.2001.

XSLT-Prozessor und können daher die XML-Repräsentation eines Objekts direkt interpretieren, d.h. in Verbindung mit dem entsprechenden Stylesheet darstellen. Ein derartiges Vorgehen spart die Prozessorzeit für die Umwandlung von XML-Repräsentation in eine HTML-Darstellung.

Wesentliches Element der HTTP-Schnittstelle des sTeam-Systems ist eine hierarchische Abbildung der Raumstruktur. Zu diesem Zweck ist ein eigener Object Request Broker vorgesehen (vgl. Abschnitt 5.2.2). Er weist jedem Objekt eine eindeutige Umgebung zu – ein Vorgang, der der typischen dateorientierten Syntax des Zugriffs auf Webadressen entspricht. Auf diese Weise lässt sich ein virtueller Raum über die typische WWW-Syntax ansprechen.²⁷⁹

Eine zweite Möglichkeit des Zugriffs bildet die für jedes Objekt eindeutige Objekt-Nummer. Sie wird in die URL-Pfadangabe aufgenommen.

Interaktive Elemente der Webschnittstelle werden durch LPC-Skripte erzeugt, welche als eigenständige Objekte direkt im Server aufgerufen werden. Einfache Interaktionsformen, beispielsweise die Aktivierung von kontextsensitiven Menüs, werden über Javascript realisiert.²⁸⁰

Linkkonsistenz (XML)

Die Darstellung netzgestützter Materialien ist neben der Bereitstellung einer plattformunabhängigen Benutzerschnittstelle die primäre Aufgabe der Webschnittstelle.

Isoliert voneinander darstellbare Dokumenttypen, wie z.B. ein Übungsblatt als PDF-Dokument oder eine Illustration als Grafikdatei, werden als eigenständige Objekte verwaltet. Ihre Anzeige in der Webschnittstelle erfolgt mit Hilfe des Webservers. Hierzu wird ein entsprechender Objekthinhalte über ein Dateisystem-Modul in den Roxen-Webserver übertragen. Im Detail wird dem Roxen-Webserver ermöglicht, die Objekthinhalte direkt aus der mit dem sTeam-Server verbundenen Datenbank zu laden.

Im Falle der Verwaltung von Hypertextdokumenten (Websites) ist die Verarbeitung der dargestellten Inhalte aufseiten des sTeam-Servers mit erheblich höherem Aufwand verbunden:

Hypertextdokumente können aus einer ganzen Anzahl von Dateien sowie multimedialen Objekten bestehen und enthalten sowohl interne als auch externe Verweise. Eine Anzahl von Dateien ist dabei meist durch Verweise verknüpft. Speziell die Struktur des WWW (HREF-Protokoll) kodiert einen Verweis über eine feste Position auf einem Webserver. Ein derartiger dateorientierter Ansatz bietet zunächst wenig Flexibilität im Umgang mit Hypertextdokumenten, ein Verschieben des Ursprungs- oder Zieldokumentes eines Verweises ist ohne seine explizite Anpassung nicht möglich.

Moderne Websites generieren aus diesen Gründen Verweise dynamisch mit Hilfe von Datenbanken und serverseitigen Skripten. Verschiedene Anstrengungen konsistente Verweise für das Internet bereitzustellen, finden sich schon parallel zur Entwicklung des WWW (vgl. das HyperG-System, [Andrews et al. 1995]). Das Erzielen von Linkkonsistenz für das WWW auf Basis bestehender Protokolle wird beispielsweise von [Moreau & Gray 1998] forciert.

²⁷⁹ Beispielsweise (fiktive Adresse): <http://steam.upb.de/vorlesungsraum/softwareergonomie>.

²⁸⁰ Frühe Prototypen des sTeam-Systems verarbeiten eintreffende HTTP-Anfragen direkt über ein eigenständiges Port-Objekt. Auf diese Weise wird der Webserver als Teil des sTeam-Servers implementiert. Gleichzeitig wird die HTML-Darstellung von Objekten aus in Stylesheets integrierten LPC-Kommandos erzeugt (vgl. [Bopp 2000, S. 66ff.]).

Im Rahmen der Semantic Web Initiative des W3C ist es das erklärte Ziel der XML- und XLINK-Standards, konsistente, bidirektionale Verweise in das WWW zu integrieren (vgl. die Bestrebungen im Zusammenhang mit der XML-Pointer-Language²⁸¹).

Die entwickelte Architektur kooperativer Wissensräume sieht sowohl XML-Hypertextdokumente mit ihrer zukünftigen Linkkonsistenz vor, konzentriert sich aber gleichzeitig auf die gegenwärtige Struktur des WWW. Eine derartige Vorgehensweise bietet Kompatibilität zu den in breitem Maße verfügbaren Werkzeugen (Vermeidung von Medienbrüchen).

Aus dem Anspruch des kooperativen Umgangs mit Hypertextdokumenten (Möglichkeiten der primären Medienfunktionen) ergeben sich eine Reihe von architektonischen Anforderungen, die in der Serverarchitektur umgesetzt sind. Diese umfassen:

- *Verweiskonsistenz für im Server abgelegte Dokumente:* Verweise innerhalb von Hypertextdokumenten, welche sich in einem virtuellen Raum befinden, müssen auch nach Ausübung der primären Medienfunktionen (speziell der Medienfunktion des Transports zwischen verschiedenen Orten innerhalb der Lernumgebung – Verschieben eines Dokuments) ihre Gültigkeit besitzen. Dies bezieht sich ebenfalls auf die Modifikation von Dokumenten, auf welche sich ein Verweis bezieht, die also Zielpunkt eines Verweises sind. Im Falle des Löschens eines Zieldokumentes müssen die entsprechenden Verweise im einfachsten Fall entfernt oder kenntlich gemacht werden. Der sTeam-Server realisiert Verweiskonsistenz durch einen speziellen Ersetzungsmechanismus sämtlicher Verweise eines in den Server eingebrachten Dokuments.
- *Verweiskonsistenz für WWW-Adressen, die auf ein Objekt im Server verweisen:* Verweise auf Dokumente in sTeam müssen auch nachfolgend verschiedener Handlungen an diesen (kooperativer Medienfunktionen) Gültigkeit besitzen. Es wird einem Objekt eine URL zugewiesen, die für die Lebensdauer des Objekts eindeutig ist und die unabhängig von der logischen Position des Objekts in der Lernumgebung ist. Der sTeam-Server realisiert die Zuordnung fester Adressen zu Objekten über ein spezielles Attribut und zugehörigen ORBs (vgl. Abbildung 5-10).
- *Verweiskonsistenz für Referenzen auf nicht im Server abgelegte Webseiten:* Referenzen auf externe Webseiten sollten soweit möglich von der Gültigkeit der Referenz auf die Webseite informiert sein. Im idealen Fall können ungültige Referenzen auf externe Webadressen aktualisiert oder entfernt werden. Der sTeam-Server erkennt in der vorliegenden Implementierung nicht automatisch ungültige (nicht verfügbare) URLs. Ungültige URLs müssen manuell durch den Nutzer entfernt werden.

Architektonisch wird die Verweiskonsistenz für im Server abgelegte Dokumente mittels Ersetzungsmechanismen der Verweise durchgeführt. Hierbei wird zwischen internen, auf andere Objekte des Servers referenzierende Verweise und externen, auf eine beliebige Webseite außerhalb des Servers deutende Verweise unterschieden.

Eine Verweisersetzung wird im Rahmen des Einfügens von HTML-Dokumenten (Upload) in den Server vorgenommen. Hierzu findet ein als Pike-Modul verfügbarer SGML-Parser (HTML-Modus) Verwendung (vgl. [Hübinette 2001]). Sämtliche HTML-

²⁸¹ Vgl. <http://www.w3.org/TR/xptr/>, Stand 1.12.2001.

Tags (HREF, IMG etc.), die eine Referenz auf ein ebenfalls im Server befindliches Dokument erzeugen bzw. auf das einzufügende Dokument verweisen, werden durch einen Skriptaufruf ersetzt. Dieser Skriptaufruf enthält einen Befehl zum Laden des entsprechenden sTeam-Objekts, also eine Referenz auf das Ziel des Verweises.

Durch Integration von Pike-Skriptaufrufen in HTML-Dokumente ist bei der Darstellung eines Dokuments über die Webschnittstelle keine explizite Rückerstattung notwendig. Diese wird bei Aktivierung des Verweises implizit durch Nachladen der referenzierten Objekte vorgenommen.

Eine Linkkonsistenz für WWW-Adressen, welche auf ein sTeam-Objekt verweisen, wird unmittelbar durch Bereitstellung eines URL-Object Request Brokers (ORB) realisiert. Jedes Objekt kann unabhängig von seiner Position innerhalb der Raumstruktur eine feste URL-Adresse erhalten. Diese wird nur im Falle einer expliziten Attributänderung aufseiten des Benutzers modifiziert.

Die dritte Anforderung der Linkkonsistenz für Referenzen auf nicht im sTeam-Server vorhandene Webseiten wird zurzeit nicht umgesetzt. Die Architektur des WWW bietet keinen Benachrichtigungsmechanismus für die Gültigkeit von Verweisen, erst im Falle der Aktivierung wird ein ungültiger Verweis erkannt. Konzeptionell lässt sich resultierend aus den architektonischen Einschränkungen des WWW in seiner jetzigen Form lediglich feststellen, inwieweit ein Verweis Gültigkeit besitzt. Wird das Ziel eines Verweises umbenannt, aktualisiert oder verschoben, kann keine automatische Aktualisierung des entsprechenden Verweises erfolgen.

Einfügen von Dokumenten über die Webschnittstelle

Eine wesentliche, umzusetzende Anforderung kooperativer Wissensräume ist das Einfügen und Erzeugen von Materialien. Es sollte ohne spezielle Zusatzwerkzeuge und ohne spezifische (nicht allgemein verfügbare) Clients möglich sein. Zu diesem Zweck geht die im Rahmen der vorgeschlagenen Architektur gestaltete Webschnittstelle eine enge Symbiose mit der FTP-Schnittstelle des Servers ein. Letztere gestattet das Einfügen von Dokumenten oder Objekten in einen virtuellen Raum. Ähnlich der Webschnittstelle bildet ein ORB die Struktur aus Räumen, Containern und Dokumenten in eine hierarchische dateiorientierte Struktur ab.

Die Webschnittstelle erlaubt einen direkten Wechsel zwischen HTTP- und FTP-Darstellung. Im idealen Fall wird automatisch eine entsprechende Authentifizierung am Server vorgenommen. Es können Objekte direkt von dem Browser in den Server übertragen werden (die meisten Browser unterstützen ebenfalls das FTP-Protokoll).

Konzeptionell wird die FTP-Schnittstelle in ähnlicher Weise wie die HTTP-Schnittstelle (Webschnittstelle) implementiert. Das in den Roxen-Webserver integrierte Dateisystemmodul setzt sowohl HTTP-Anfragen als auch FTP-Anfragen in entsprechende Aufrufe an den Server um. Damit werden sämtliche FTP-Kommandos zunächst von dem Roxen-Webserver bearbeitet und an den sTeam-Server weitergeleitet.

Alternativ zum populären FTP-Protokoll besteht die Möglichkeit, auf ähnliche Art und Weise das von der „Internet Engineering Task Force“ (IETF) vorgeschlagene WebDAV-Protokoll (Web Distribution and Versioning Protokoll)²⁸² zu unterstützen.

²⁸² Vgl. <http://www.webdav.org/>, Stand 1.12.2001.

WebDAV ist eine Erweiterung des HTTP-Protokolls, die den netzgestützten Transfer von Dateien und Metadaten (Attributen) zwischen verschiedenen Applikationen unterstützt. Es eignet sich in dieser Form insbesondere für das Einbringen von Materialien in die Lernumgebung. Architektonisch erfordert WebDAV die serverseitige Implementierung einer Protokollerweiterung. Verschiedene Kommandos wie das Anlegen eines Verzeichnisses (`mkcol`), Operationen zur Manipulation von Dateien (`copy`, `move`) sowie zum Einfügen und Laden von Objekten (`put`, `get`) sind als Teil des HTTP-Protokolls umzusetzen. Daten einer WebDAV-Transaktion werden im XML-Format vorgenommen.

Spezielle WebDAV-Attribute erlauben es, bestimmte Metadaten, wie den Autor einer Datei oder Schlüsselworte, schon beim Einfügen eines Objekts in den Server zu übertragen. Derartige Informationen werden von einem WebDAV-Client direkt in die XML-Repräsentation der zu transferierenden Daten aufgenommen. Auch das Ergebnis einer WebDAV-Transaktion wird als XML-Repräsentation bereitgestellt und erlaubt eine gezielte Fehlererkennung und Bearbeitung.

Die Qualität einer Unterstützung des WebDAV-Standards liegt in zwei Bereichen: Zum einen bietet WebDAV ähnlich dem FTP-basierten Transfer Funktionen des Erzeugens von Verzeichnissen und des parallelen Übertragens einer größeren Anzahl von Objekten, d.h. Lehrmaterialien können schon während des Einfügeprozesses in die Lernumgebung vorstrukturiert abgelegt werden. Gleichzeitig ist ein Übertragen einer größeren Anzahl von Objekten in einem Schritt möglich.

Zum anderen eröffnet der WebDAV-Standard durch eine größere Anzahl möglicher Clients die Perspektive, aus beliebigen Applikationen direkt Daten auf dem kooperationsunterstützenden System ablegen zu können. Microsoft Office-Applikationen unterstützen schon zum gegebenen Zeitpunkt den WebDAV-Standard, d.h. verhalten sich konform einem WebDAV-Client. Damit kann beispielsweise eine Präsentation oder ein Textverarbeitungsdokument direkt und ohne Umwege auf einem sTeam-Server abgelegt werden. Metadaten wie Schlüsselworte oder begrenzte Einstellungen von Benutzerrechten können dabei direkt aus den Anwenderapplikationen übernommen werden.

Eine Reihe verfügbarer, in Java realisierter WebDAV-Clients erlauben den plattformabhängigen Transfer von Objekten in den Server. Gleichzeitig muss jedoch angemerkt werden, dass zurzeit nur wenig ausgereifte Implementierungen von WebDAV-Clients vorliegen. Weitgehende Unterstützung des WebDAV-Protokolls bieten die aktuellen Versionen der Microsoft-Betriebssysteme.

Ansatzpunkt der Integration von WebDAV-Unterstützung in die dargestellte Serverarchitektur ist der Roxen-Webserver. Ähnlich dem FTP-Protokoll erfolgt eine Verarbeitung von WebDAV-Anfragen zunächst über den Webserver. Anfragen werden anschließend an den sTeam-Server und die verbundene Datenbank weitergeleitet. Eine erste Implementierung des WebDAV-Protokolls für den Roxen-Webserver befindet sich zurzeit in Vorbereitung und ist in ersten Prototypen in die sTeam-Architektur integriert.

Webschnittstelle, FTP- oder WebDAV-Clients repräsentieren primär den asynchronen architektonischen Aufbau eines sTeam-Clients. In den folgenden Abschnitten wird die Kernarchitektur eines synchronen Clients vorgestellt.

5.3.4 Architektur synchroner Java-Clients

Von Seiten der grundsätzlichen architektonischen Merkmale weisen verschiedene synchrone Clients in dem gezeigten Ansatz kooperativer Wissensräume große Ähnlichkeiten auf. Dies liegt zum einen an dem objektorientierten Aufbau des Servers und seiner

Klassenstruktur (vgl. Abschnitt 5.2.1), dem COAL-Protokoll und dem zugehörigen API (vgl. Abschnitt 5.2.8). Zum anderen lassen sich verschiedene Clients mittels eines einheitlichen Modul-Konzepts gestalten. Dieser Entwicklungsschritt wird durch ein Framework unterstützt. Letzteres stellt Grundklassen der Funktionalität bereit, auf dessen Basis spezielle Clients entworfen werden können (vgl. [Enklaar 2001 S. 32ff.]).

Insofern kann die im Folgenden skizzierte Grundstruktur für die Realisierung eines Clients zur Darstellung des virtuellen Raums, Funktionen der gegenseitigen Wahrnehmung und den Umgang mit Lehrmaterialien stehen. Sie bildet aber ebenfalls die architektonische Grundlage zu einer Realisierung von eng gekoppelten Shared Whiteboard-Clients.²⁸³

Framework zur Gestaltung synchroner Clients – Überblick verschiedener synchroner Clients

Aus architektonischer Sicht betrachtet setzt sich ein synchroner sTeam-Client aus dem Framework, einer Reihe von Anwendungsmodulen und dem API zur Kommunikation mit dem Server zusammen.

Das Framework bildet die Grundlage eines modularen Aufbaus von Clients. Es führt den Anmeldungsprozess eines Benutzers am Server durch und verwaltet verschiedene Anwendungs- und Clientmodule. Letztere werden auf dem Server abgelegt und können sehr unterschiedliche Funktionalitäten, wie z.B. eine spezifische Raumdarstellung, erzeugen. Das Framework bildet in dieser Form einen Rahmen für die Kombination und Ausführung einer Anzahl von Clientmodulen. Gleichzeitig werden Basisklassen zur vereinfachten Entwicklung neuer zusätzlicher Clientmodule bereitgestellt. Ziel ist es, dem Entwickler und Endbenutzer eine weitgehend flexible Gestaltung seines synchronen Zugangswerkzeugs zu virtuellen Wissensräumen zu bieten.

Diese Flexibilität beginnt aufseiten der Entwickler in der Gestaltung neuer Clientmodule unter Einbeziehung des Framework bzw. API und erstreckt sich bis auf die Anwendungsebene in der Auswahl geeigneter Clientmodule bzw. verschiedener Alternativen der Darstellung.

Als zweiter wesentlicher Vorteil der Modultechnologie erhält der Client während des Anmeldevorgangs durch Ablage der Module aufseiten des Servers die jeweils aktuellsten Versionen eines Clientmoduls. Dabei können ebenfalls gezielt ältere Versionen von Clientmodulen beibehalten werden. Eine Distribution neuer Versionsstände oder zusätzlicher Clientfunktionalität ist damit weitgehend automatisiert bzw. kann von den Betreibern des Servers vorgenommen werden.

Der Umfang von Clientmodulen beschränkt sich im Hinblick auf die Architektur weitgehend auf die Semantik und die Benutzerschnittstelle der realisierten Applikation. Die Kommunikation mit dem Server wird über das Server-API realisiert, Kommunikation und Interaktion mit weiteren Clientmodulen werden über das Framework bereitgestellt. Hierzu zählen beispielsweise Mechanismen wie Drag&Drop oder eine Zwischenablage zum Austausch von Daten zwischen verschiedenen Clientmodulen.

In Bezug auf die in Kapitel 2 gezeigten Szenarien und Anforderungen sind eine ganze Anzahl von Clientmodulen notwendig und sinnvoll. Hierzu zählen:

²⁸³ Fragestellungen der Gestaltung von Benutzerschnittstellen eines Clients oder der technischen Realisierung eines Shared Whiteboards werden im Rahmen der vorliegenden Arbeit nicht weiter untersucht. Die Konzeption und Gestaltung eines Shared Whiteboards basierend auf der vorgestellten Architektur wird in [Büse 2001] vorgenommen.

- *Raumdarstellung – semantischer Raum*: Visualisierung eines virtuellen Raums mit seinen Containern, Dokumenten und Benutzern. Die räumliche Nähe zwischen Dokumenten und Benutzern und Dokumenten untereinander drückt eine semantische Struktur und Nähe aus (vgl. [Enklaar 2001] und [Sinemus 2000]).
- *Kommunikationsmodule – Chat*: Repräsentation verschiedener Chat-Gruppen innerhalb eines Raums. – Bilden neuer Gruppen, Möglichkeit der Teilnahme in mehreren simultanen Gruppen (vgl. [Enklaar 2001]). Exemplarisch wurde bereits das Konzept des strukturierten Chats erprobt – hierbei sind Diskussionsbeiträge eng an Objekte auf einem Shared Whiteboard geknüpft. Materialien einer Lernumgebung werden zudem zusammen mit Chat-Beiträgen in einer persistenten Baumdarstellung vereinigt (vgl. [Preiß 2000]).
- *Kommunikationsmodule – Audio-Chat (Voice Chat)*: Bereitstellung von Audio-Kommunikationsfunktionalität in einem virtuellen Raum. Es sind sehr unterschiedliche Metaphern der Verarbeitung von Audiosignalen in Bezug auf die räumliche Zuordnung der Kommunikationspartner zueinander denkbar (vgl. hierzu die verschiedenen Konzepte der räumlichen Aufbereitung von Audiosignalen, z.B. [Chodura 2000]). Im einfachsten Fall werden Audio-Daten allen Teilnehmern des Raums bereitgestellt, d.h. jeder Teilnehmer eines virtuellen Raums hört alle Äußerungen anderer Teilnehmer desselben Raums.²⁸⁴
- *Raumstruktur – Darstellung der semantischen Beziehungen zwischen Räumen*: Eine Karte des virtuellen Raums und seiner benachbarten, erreichbaren Räume erzeugt eine semantische Karte über ein thematisch oder organisatorisch abgebildetes Fachgebiet. Es lassen sich spezifische Darstellungsformen der Raumstruktur entwickeln (vgl. [Chen et al. 1999] und [Spence 2001, S. 134ff.]). Wichtiges Kriterium für die Repräsentation einer semantischen Karte nach [Klemme et al. 1998] und [Hampel & Selke 1999] ist es, die Navigationsschritte eines Nutzers zu jedem Zeitpunkt auf der Karte darzustellen.
- *Whiteboard/semantischer Raum*: Eine synchronisierte Darstellung von sTeam-Objekten oder -Dokumenten zusammen mit Benutzern und grafischen Objekten auf einer planaren Zeichenfläche schafft eine Symbiose aus Funktionen des Shared Whiteboards (synchrone Präsentation, Interaktion, Telepointer) und der gemeinsamen Ausübung von Medienfunktionen auf Materialien einer Lernumgebung. Das Modul eines Whiteboard-Raums erlaubt speziell die kooperative Präsentation von Vorlesungsmaterialien (vgl. [Büse 2001]). Materialien können auf dem Whiteboard angeordnet und um verschiedene grafische Elemente ergänzt werden. Elemente können in mehreren Ebene platziert werden und es stehen Telepointer zur Verfügung. Das Whiteboard kann zur Navigation (Bewegen durch die Raum-Dokumentenstruktur) und zur Darstellung der Materialien im Browser genutzt werden.
- *Administrationsmodul*: Aufgaben der zentralen Administration, wie beispielsweise das Erzeugen neuer Räume, die Spezifikation von Benutzerrechten an Objekten oder eine Verwaltung von Benutzerzugängen und Benutzergruppen, können ein spezifisches Administrationsmodul mit entsprechenden grafischen Darstellungen notwendig machen. Auch Konzepte der Selbstadministration und die Möglichkeit der Administration über die Webschnittstelle schließen eine derartige zentrale Administrationsschnittstelle nicht aus. Speziell im Umfeld der Verwaltung einer Vielzahl von Benutzergruppen, die teils hierarchische Abhän-

²⁸⁴ Vgl. auch [Greenhalgh & Benford 1995] und [Schiffner & Chodura 2000].

gigkeiten aufweisen, sind grafische Darstellungen der Gruppenstruktur sinnvoll. Auch Aktivitäten der automatischen Anmeldung von Nutzern an das System sind über eine derartige Schnittstelle entsprechend koordinierbar.

Zur Verifikation des Modulkonzepts und Frameworks ist zudem eine dreidimensionale Repräsentation des virtuellen Raums als eigenständiges Modul prototypisch umgesetzt (vgl. [Hampel & Keil-Slawik 2001c]). Diese wird jedoch aufgrund der aufwändigen grafischen Darstellung und Problemen bei der Navigation in der Dokumenten- und Raumstruktur nicht weiter verfolgt.

Kernarchitektur eines synchronen Clients

Eine wesentliche Designentscheidung der gezeigten Gesamtarchitektur ist es, die Client-Server-Schnittstelle möglichst universell zu gestalten und damit eine weitgehend entkoppelte Entwicklung und Erweiterung von Server und Clients zu gewährleisten. Eine wesentliche Designanforderung ist aus diesem Grunde der Entwurf einer schlanken und möglichst universellen Client-Server-Schnittstelle.

COAL erfüllt diese Anforderungen. So umfasst COAL lediglich Kommandos zum An- und Abmelden eines Clients am Server, zum Up- und Download von Dateien, zur Übermittlung von Server-Ereignissen und eine universelle Funktion zum Aufruf von Methoden innerhalb des Servers.

Ein eigenes API kapselt die im Abschnitt zu COAL (vgl. Abschnitt 5.2.8) angesprochenen Funktionen in entsprechenden Klassen und organisiert eine dauerhafte (für den Zeitraum der Sitzung währende) Verbindung. Synchroner Clients (wie oben aufgeführt) können mit wenigen Bestandteilen des Frameworks eine Verbindung zum sTeam-Server aufnehmen, auf die Objekte des Servers zugreifen und Events vom Server empfangen. Die Paketstruktur eines synchronen, in Java entwickelten Clients macht die konsequente Trennung von Funktionsbereichen deutlich. Hierbei existieren die Pakete:

- *Verbindungsinstanz zum Server* (`steam.api`): Der `SteamConnector` organisiert die Verbindung mit dem Server. Er nimmt gleichzeitig die Repräsentation von Serverobjekten in entsprechende clientseitige Proxy-Objekte (Platzhalter für Objekte des Servers) vor.
- *Verarbeitung serverseitiger Ereignisse* (`steam.api.event`): Die für Objekte eines Clients relevanten Events können direkt an den zugehörigen Serverobjekten abonniert werden. Serverseitige Events werden von Klassen des APIs verarbeitet und in entsprechende Ereignisse des Clients umgewandelt. In analoger Weise können Objekte des Clients Events im Server auslösen. Die Kapselung von Events des Servers in zugehörige Java-Klassen bildet die architektonische Grundlage kooperativer Prozesse zwischen verschiedenen Clients (verschiedene Sichten auf Objekte des Servers).
- *Kommunikation zwischen Client und Server (COAL)* (`steam.api.io`): Die Implementierung des COAL-Protokolls auf unterster Ebene der Serverkommunikation enthält nur wenige, universell gestaltete Kommandos. Hierzu zählt die Authentifizierung eines Clients am Server, das Übertragen von Dateien und die eigentliche Transportschicht von Server-Ereignissen (Events).
- *Framework-Klassen* (`steam.framework`): Framework-Klassen bilden die Grundstruktur zur Erzeugung von Clientmodulen. Sie enthalten alle wesentlichen Eigenschaften eines Java-Clients aus Sicht der Benutzerschnittstelle, beispielsweise die grundsätzliche Menüstruktur, sowie verschiedene Fensterklassen.

- *Module – Realisierung verschiedener Clients:* (`steam.modules.chat`; `steam.modules.roomdisplay`): Realisierung jeden Moduls als eigenständiges Java-Paket. Funktionsbereiche eines Clients, wie z.B. ein Fenster zur synchronen Kommunikation (Chat) oder die Visualisierung der Raumstruktur (Raumkarte), werden als unabhängige Module gestaltet. Module sind in einfacher Weise austauschbar und enthalten im Wesentlichen die Benutzerschnittstelle und Semantik der gewünschten Funktionalität. Die Grundstruktur der umgebenden Benutzerschnittstelle und kooperative Funktionsbereiche werden aus dem Framework entnommen.

Grundlegendes Merkmal der Architektur eines Clients ist es, Wege der Replikation bzw. des Zugreifens auf die Objekte des Servers, also Materialien, Räume und Benutzerobjekte, zu schaffen. Die sTeam-Architektur wählt das Konzept der Proxy-Objekte, um in den Clients notwendige Objekte des Servers verfügbar zu machen.

Clientseitige Proxy-Objekte und Events

Für jedes Server-Objekt lässt sich innerhalb der Clients eine entsprechende Java-„Objekthülse“ replizieren. Dies geschieht für die innerhalb des Kontextes eines Clients notwendigen Objekte, beispielsweise die Objekte, die sich im Sichtbarkeitsbereich einer Clientrepräsentation befinden. Zugriffe auf Attribute und Methoden eines Objekts werden für den Client transparent auf den Server übertragen. Damit liegen in dem Client Proxy-Objekte²⁸⁵ der relevanten Server-Objekte vor (vgl. Abbildung 5-9).

Der Zugriff auf kooperative Objekte des Servers wird über den `SteamConnector` realisiert. Er nimmt als unabhängiges Objekt eine exponierte Stellung in der Architektur des Clients ein. Ein Connector kann allgemein als eigenständige Instanz aufgefasst werden, der den Austausch von Daten (Messages) über ein Protokoll zwischen verschiedenen Komponenten koordiniert (vgl. Abbildung 5-8).²⁸⁶

Innerhalb der Architektur des Clients ist der `SteamConnector` zugleich Ansatzpunkt für Geschwindigkeitsoptimierungen wie das Zwischenspeichern von Attributen oder ganzen Objekten (Cache).

Die zweite wesentliche Aufgabe einer Rahmenarchitektur zur Entwicklung von synchronen Clients liegt in der leistungsfähigen Verarbeitung von Events. Events synchronisieren Darstellungen zwischen Clients auf den virtuellen Wissensraum. Der Server übermittelt in kooperativen Objekten ausgelöste Events an die verbundenen Clients. Hierbei ist die Eingrenzung des Kontextes eines Events ein wesentliches Kriterium. Ein Client kann sämtliche auftretenden Events eines Typs an den Objekten in seinem Sichtbarkeitsbereich überwachen, kann sich aber auch sehr gezielt auf einzelne Events an einem Objekt beschränken.

Die erste Möglichkeit bezieht sich beispielsweise auf die Überwachung notwendiger Events in Bezug auf die Aktualität der Darstellung eines Clients. So können z.B. global alle Änderungen der Position von Objekten innerhalb der Raumdarstellung überwacht werden. Die zweite Möglichkeit eignet sich für die gezielte Überwachung einzelner Events, z.B. die Attributänderung eines Objekts. Zu diesem Zweck kann ein Ereignis über einen so genannten `steamListener` überwacht werden. Die `steamListener`

²⁸⁵ Proxy-Objekte sind ein klassisches Konzept des Designs verteilter Anwendungen (vgl. [Dave et al. 1992]). Sie entsprechen weitgehend dem Proxy-Design Pattern [Gamma et al. 2000, S. 207ff.].

²⁸⁶ Vgl. ebenfalls [Fuentes & Troya 2000].

sind eigenständige Objekte, welche an ein beliebiges kooperatives Objekt innerhalb des Clients geknüpft werden.

Im gleichen Zuge wird auch serverseitig ein zugehöriger Listener an das entsprechende Objekt geknüpft. Bei Auslösung eines Ereignisses, welches mit dem überwachten Objekt in Zusammenhang steht, versendet der Server eine Mitteilung an den Client (im Server sind die Ereignisse in diesem Fall an den Benutzer gebunden). Auf diese Weise werden Ereignisse gezielt abonniert und vom Server an die betroffenen Clients gemeldet. Beispielsweise lassen sich für die Realisierung eines Chats innerhalb des Clients so genannte `chatListener` an die entsprechenden Benutzerobjekte binden, welche im Falle einer Chat-Mitteilung eine zugehörige Funktion zur Darstellung der Mitteilung aufrufen.

Ein Client erzeugt in dieser Form eine persönliche Sicht auf Objekte innerhalb des Servers. Diese Sicht ist gezielt auf relevante Objekte und Ereignisse (Events) eingeschränkt.

Gleichzeitig wird das Grundmerkmal der Architektur des Servers, bestehend aus persistenten Objekten und ihren Attributen, die durch Events verknüpft werden, bis auf die Ebene der Clients durchgängig weitergeführt.

Grafische Benutzerschnittstellen

Die grafische Umsetzung (Benutzerschnittstelle) eines Clients baut unmittelbar auf der serverseitigen Struktur aus kooperativen Objekten auf. Kooperative Elemente aufseiten der Benutzerschnittstelle werden durch Komponenten in der Java-Swing-Terminologie `JComponents` umgesetzt. Diese können eigene Ereignisse auf der Ebene der Benutzerschnittstelle verarbeiten und bilden auf diese Weise das grafische Gegenstück zu den Objekten des Servers bzw. entsprechender clientseitiger Proxy-Objekte.

Ein derartiger Architekturansatz bietet einen hohen Freiheitsgrad in den Interaktionsformen zwischen Benutzern und der grafischen Darstellung eines virtuellen Raums. Mögliche Interaktionsformen sind beispielsweise das Selektieren und Verschieben von Objekten, aber auch Drag&Drop zwischen verschiedenen Fenstern und Applikationen. Zu einer konzeptionellen Darstellung der Realisierung eines Shared Whiteboard mit Hilfe grafischer Komponenten, die Objekten innerhalb des sTeam-Servers entsprechen, vgl. [Büse 2001].

5.4 Zusammenfassung: nutzerzentriert und selbstadministriert

Die vorgestellte architektonische Umsetzung virtueller Wissensräume konzentriert sich gemäß der in Kapitel 2 dargestellten Szenarien und in Kapitel 3 formulierten Anforderungen im Wesentlichen auf ein Objektmodell virtueller Wissensräume. Hierzu wird eine Benutzer- und Raumstruktur aufgebaut; Materialien werden durch gemeinsame Objekte und Attribute repräsentiert. Gleichzeitig steht die Selbstverwaltung der Studierenden, d.h. eine dezentrale Organisation der Lernumgebung im Vordergrund.

Im Folgenden sind die wesentlichen Merkmale benutzerseitiger Konfigurierbarkeit und Erweiterbarkeit sowie die Selbstadministration an konkreten architektonischen Entscheidungen zusammengefasst dargestellt.

Zunächst zeigt sich der nutzerzentrierte Charakter der skizzierten Umsetzung kooperativer Wissensräume in der prinzipiellen, nur durch Benutzerrechte eingeschränkten Möglichkeit, beliebige Objekte zu erzeugen. Damit kann prinzipiell jedem Nutzer das Recht eingeräumt werden, neue Räume oder Container erzeugen zu dürfen (vgl.

Abbildung 3-6) bzw. beliebige Objekte innerhalb eines virtuellen Raums ablegen zu dürfen.

In analoger Weise werden alle kooperativen Medienfunktionen durch ein hierarchisches Konzept von Benutzer- und Gruppenberechtigungen an gemeinsam genutzten Objekten koordiniert. Damit können Nutzern beliebige Möglichkeiten des Erzeugens, Verschiebens, Übertragens und Verknüpfens von Materialien eingeräumt werden. Im einfachsten Fall werden Berechtigungen aus der Struktur des Raums oder Containers geerbt. Aber auch Situationen des expliziten Setzens von Berechtigungen an einzelnen Dokumenten sind eine sinnvolle Option – beispielsweise falls Dokumente auch in der Weitergabe von einem Nutzer zu einem anderen Nutzer ihre ursprünglichen Zugriffsberechtigungen beibehalten müssen.

Verschiedene Typen von Materialien der kooperativen Lernumgebung unterscheiden sich lediglich durch ihren Inhalt und einige wenige spezielle Attribute (Metadaten). Letztere dienen in dieser Form der Verwaltung unterschiedlicher Dokumenttypen. Nutzer können demgemäß beliebige, auch nicht speziell angepasste Dokumente einbringen und mittels externer Werkzeuge manipulieren. Eine Visualisierung verschiedener Materialien erfolgt entweder über einen gewöhnlichen Browser oder eine spezielle Applikation.

Verbindendes Element der Möglichkeit zur Selbstorganisation ist damit in vielfältiger Form ein offenes Konzept aus Benutzerrechten und Benutzergruppen.

Die Möglichkeit der Weitergabe von Berechtigungen an beliebigen Objekten kann durch entsprechende Zugangswerkzeuge in verschiedener Weise abgebildet werden. So kann beispielsweise die Administration eines virtuellen Raums, also das Recht zur Vergabe beliebiger Berechtigungen, einem Raumverwalter (Gruppenverwalter) unterstellt werden. In analoger Weise kann der Gruppenverwalter die Berechtigung einer Kollektion von Lehrmaterialien einem Lehrenden oder Übungsgruppenleiter unterstellen.

In Verbindung mit der Vererbung von Berechtigungen lassen sich auf diese Weise flexible Szenarien der Verwaltung von Lehrmaterialien abbilden. Die Administration für einen Dokumentenbereich (z.B. eine Kollektion) wird einem Lernenden in Selbstverwaltung unterstellt. In einem derartigen Bereich abgelegte Materialien besitzen keinerlei explizite Berechtigungen, sie erben die Berechtigungen aus dem übergeordneten Container oder Raum. Damit erhalten neu erzeugte Dokumente, aber auch verschobene Materialien automatisch die übergeordneten Berechtigungen.

Auch die Administration von Benutzergruppen, also der Gruppenstruktur einer Lernumgebung, ist durch einfache Zugriffsrechte dezentral organisiert. Besitzt ein Nutzer die Berechtigung zur Administration einer Benutzergruppe, können beliebige neue Nutzer einer Gruppe zugefügt werden. Mechanismen der automatischen Anmeldung von Nutzern zu neuen Lerngruppen basieren auf diesem Mechanismus. Über die Webchnittstelle oder einen Java-Client können sich Lernende, die lediglich einen anonymen Zugang zur Lernumgebung besitzen oder Zugang zu einer bestimmten Nutzergruppe wünschen, für die Mitgliedschaft innerhalb einer Gruppe anmelden. Die Nutzerdaten werden an den Verwalter der entsprechenden Gruppe mit Bitte um Bestätigung gesandt. Dieser kann das automatische Eintragen von Nutzern in eine seiner Verwaltung unterstehenden Nutzergruppe billigen oder abweisen.

Architektonisch weist ein derartiges Vorgehen keinerlei über die Mechanismen der dezentralen Gruppenverwaltung hinausgehende Eigenschaften auf. Durch ein spezielles Objekt (aktives Skript) werden Benutzerdaten erfasst und in den Briefkasten des Gruppenverwalters versandt. Hier erfolgt nach manueller Bestätigung die ebenfalls durch ein Skript automatisierte Aktualisierung der Nutzergruppe (vgl. Abbildung 5-13).

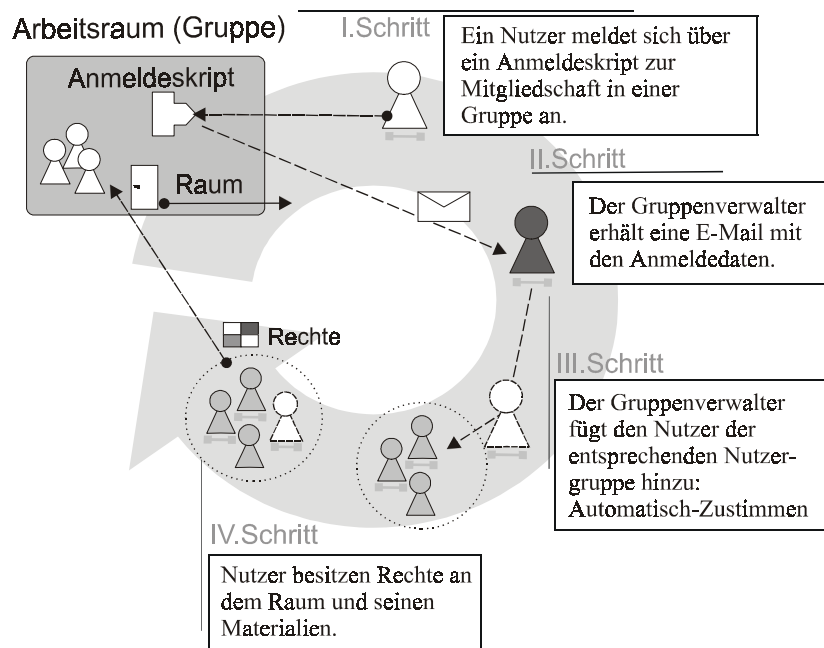


Abbildung 5-13: Selbstadministration – Anmeldung

Werkzeuge, wie die automatisierte Anmeldung zu einer Lehrveranstaltung oder die automatische Abgabe von Übungslösungen, demonstrieren die weiteren Anwendungsfelder des Konzepts einer nutzerzentrierten (selbstverwalteten) Lernumgebung.

Werkzeuge basieren architektonisch in weiten Teilen auf aktiven Objekten, welche spezifische Attribute besitzen und Ereignisse zu verarbeiten vermögen. Auf diese Weise kann ein elektronischer Briefkasten (Abgabefach) eine automatische Leerung durch Auswertung eines Datums bzw. Uhrzeitattributs veranlassen und Objekte ereignisgesteuert in das Postfach (den Rucksack) eines Übungsgruppenbetreuers verschieben. In Verbindung mit dem Konzept der Weitergabe von Berechtigungen löst ein in den Briefkasten verschobenes Dokument (Übungsblatt) ein Ereignis (Eintreten in die Umgebung des Objekts) aus. Anschließend wird das Objekt mittels eines durch das entsprechende Event ausgelösten Skripts in den zuvor spezifizierten Ordner eines Übungsgruppenleiters verschoben (vgl. Abbildung 2-2).

In ähnlicher Weise stützen sich mögliche weitere aktive Elemente auf ihre Fähigkeiten zur Verarbeitung von Ereignissen in Verbindung mit der Ablage von Daten und der Manipulation von Attributen. Aktive Objekte können prinzipiell von jedem Nutzer in kooperative Wissensräume eingebracht werden. Im Zuge der Ableitung zugehöriger Klassen werden in den Factories sowohl die Berechtigung des aktiven Nutzers als auch die Berechtigung des aufrufenden Objekts überprüft. Eine Überprüfung von Berechtigungen auf Ebene der Factories und von Events (Sicherheitsobjekten) ist notwendig, um die Integrität des kooperationsunterstützenden Systems zu wahren. Elemente, wie eine gemeinsame Zeichenfläche oder ein Protokollwerkzeug, erweitern ein gewöhnliches Objekt um spezielle Attribute zur Organisation der Sitzung, Speicherung von Referenzen auf gemeinsam genutzte Objekte oder die Verwaltung einer Statusinformation (Rekorder ein-/ausgeschaltet).

Die für die Realisierung eines neuen Werkzeugs notwendigen Attribute können an den Factories der zugehörigen Klassen flexibel registriert werden. Durch die Persistenzeigenschaft jedes Objekts ergibt sich eine einfache Speicherung von objektbezogenen

Daten und Eigenschaften. Spezielle Clients oder einem Client hinzugefügte Module vermögen durch das Registrieren zusätzlicher Attribute zur Laufzeit des Systems eine erweiterte Funktionalität zu implementieren bzw. zusätzliche Metadaten an Objekten bereitzustellen. Dies geschieht bei gleichzeitiger Kompatibilität zu früheren Clients, da nicht die Klasse eines Objekts modifiziert wird, sondern ihr lediglich neue Attribute zugefügt werden (vgl. Abbildung 5-3).

Auf diese Weise können sehr unterschiedliche Sichten auf den kooperativen Raum entwickelt und parallel bereitgestellt werden. Es liegt in der Freiheit des Lernenden, primär synchrone Clients oder beispielsweise die Webschnittstelle mit ihrem asynchronen Charakter zu verwenden.

Clients können zugleich eine Benutzerschnittstelle zu aktiven Objekten (Werkzeugen) des Servers bilden. Beispielsweise kann mit einfachen Mitteln eine spezielle Visualisierung eines Gesprächsrekorders entwickelt werden. Dieser wird als zusätzliches Modul in das Framework des Clients eingebracht. Die für das Rekorderobjekt relevanten Events werden über das Server-API an den Client weitergeleitet, verschiedene Attribute des Objekts werden in der Benutzerschnittstelle manipulierbar bereitgestellt.

In analoger Weise arbeitet ein Shared Whiteboard als Darstellungsform eines virtuellen Raums. Objekte werden mit den Attributen ihrer Position auf einer Zeichenfläche versehen. Events der Veränderung von Positionsattributen werden verarbeitet und entsprechende Auswirkungen in der Darstellung visualisiert.

Eng verbunden mit dem Konzept, dem Lernenden eine größtmögliche Freiheit in der Wahl seiner Zugangswerkzeuge zu gewähren, ist der Anspruch, den Server mit verschiedenen, standardisierten Schnittstellen zu seiner Funktionalität auszustatten.

WWW-, WebDAV-, FTP-, News- oder IRC-Schnittstellen erlauben es, weit verbreitete Zugangswerkzeuge zu Teilaspekten der kooperativen Lernumgebung zu nutzen. So kann beispielsweise ein Lernender Diskussionsbeiträge innerhalb eines virtuellen Raums mit einem herkömmlichen Newsreader verfassen oder zukünftig ein innerhalb eines WWW-Browsers integriertes Annotea-Werkzeug zur Annotation von beliebigen Materialien nutzen.

Auf diese Weise wird eine größtmögliche Integration zu bestehenden Werkzeugen und Umgebungen sichergestellt und der Vorliebe der Nutzer zu bestimmten Werkzeugen Rechnung getragen.

5.5 Alternative Implementierungen

Es bestehen vielfältige Möglichkeiten alternativer Implementierungen der entwickelten Architektur. Sie besitzen jeweils Vor- und Nachteile und sind durch gänzlich anderweitige Herangehensweisen zum Teil nur wenig vergleichbar zu dem verfolgten Ansatz. Ohne Anspruch auf Vollständigkeit und detaillierte, technische Beschreibungen sind im Folgenden einige grundsätzliche Möglichkeiten alternativer Umsetzungen der in Kapitel 4 formulierten Anforderungen genannt.

Zunächst bietet sich die Möglichkeit der Nutzung einer alternativen Laufzeitumgebung/Interpreter (Driver) zur Implementierung der LPC-Sprache des Servers. Frühere Prototypen des HyperMUD- und sTeam-Ansatzes (vgl. [Bollmeyer 1997] und [Hampel 1999]) nutzen beispielsweise den DGD-Driver. DGD bietet Mechanismen der Aktualisierung von Objektinstanzen zur Laufzeit des Systems, ist sehr effizient und eignet sich zur Abbildung ähnlicher Klassen- und Objektstrukturen wie in der vorgestellten Konzeption. Eine exponierte Stellung in der Umsetzung der gezeigten Architekturmerkmale nehmen ein Driver-Object, Auto-Object und Master-Object ein (vgl. [Bopp 2000]). Lei-

der ist der DGD nicht frei verfügbar und besitzt nicht in dem Umfang vorhandene Erweiterungsbibliotheken wie das gewählte Pike-Produkt.

Eine weitere Option ist es, die Eigenschaften des LPC-Drivers vollständig in einer eigenen Programmiersprache abzubilden. Hierzu eignet sich C++ mit Entwicklung eines entsprechenden LPC- oder C-Interpreters oder auch interpretierte Sprachen wie Python²⁸⁷. Python bietet keinerlei Fähigkeiten der Kapselung von Daten (erlaubt nur globale Variablen) und scheidet dahingehend für die Implementierung eines Sicherheitskonzepts aus. Die Entwicklung eines eigenen Interpreters ist aufwändig in der Umsetzung. In jedem Fall sind erhebliche Entwicklungsanstrengungen in der Implementierung standardisierter Kommunikationsprotokolle und der Ereignisverarbeitung zu erwarten.

Java bietet vielfältige Möglichkeiten der Entwicklung einer verteilten, semi-replizierten Architektur. Ein derartiger Ansatz wurde unter dem Blickwinkel der formulierten Anforderungen an eine kooperationsunterstützende Umgebung in [Preiß 2000] erprobt. Das Java-Shared-Data-Toolkit wird hier zu einer verteilten Architektur (Peer-to-Peer-Architektur) genutzt. Ein ausgewiesener Client übernimmt die Persistenzhaltung von kooperativen Objekten. Resümierend ist festzustellen, dass ein derartiger Architekturansatz nur schwer die Leistungsmerkmale in Bezug auf nutzerseitige Erweiterbarkeit und der Bereitstellung von Fähigkeiten des Dokumentenmanagements zu erreichen vermag. Das JSDT bietet lediglich grundlegende Kommunikationsformen (Nachrichten, gemeinsame Speicherbereiche und Semaphoren). Es eignet sich daher nur bedingt für die Umsetzung der gezeigten Architektur.

Eine zentralisierte Architektur ist mit Hilfe eines reinen Java-Servers und geeigneter, auf Enterprise JavaBeans²⁸⁸ oder CORBA aufsetzender Clients denkbar. Eine derartige Architektur weicht konzeptionell erheblich von dem im Rahmen dieser Arbeit verfolgten Ansatz ab und erfordert einen hohen Implementierungsaufwand. Sie würde Ähnlichkeiten zum Aufbau von MultiTEL [Fuentes & Troya 2000] oder des DyCE-Servers (vgl. Abschnitt 4.2.3) aufweisen.

Auch die Anpassung bzw. Nutzung eines reinen Java-basierten MUD-Servers in Verbindung mit existierenden Java-Bibliotheken zur verteilten Applikationsentwicklung bietet eine nennenswerte Option möglicher Umsetzungen. Hier könnte ein frei verfügbares, in Java implementiertes MUD Ausgangspunkt der Konzeption sein. Auch die Verknüpfung eines angepassten MUD-Servers mit einem frei verfügbaren, WWW-basierten Dokumentenmanagementsystem wäre in diesem Zusammenhang denkbar. Die Wahl eines geeigneten, frei bereitgestellten und vor allem WWW-basierten, leistungsfähigen Dokumentenmanagementsystems erscheint jedoch problematisch.

Insgesamt ist festzuhalten, dass sich eine Vielzahl möglicher Alternativen zu einer Umsetzung der Grundkonzeption bezüglich der formulierten Anforderungen bieten. Im Einzelnen ist zu untersuchen, inwieweit lizenzrechtliche und technologische Faktoren eine Realisierung der skizzierten Architektur beeinflussen. Speziell die Leistungsfähigkeit interpretierter Programmiersprachen in Verbindung mit Bibliotheken zur netzgestützten Kommunikation und Ereignisverarbeitung müssen die Performanz der späteren Lernumgebung im täglichen Einsatz sicherstellen. Auch Robustheit der verwendeten Bibliotheken und Laufzeitumgebungen ist ein wichtiges Kriterium. Insbesondere in den

²⁸⁷ Vgl. <http://www.python.org/>, Stand 1.12.2001.

²⁸⁸ Vgl. <http://java.sun.com/products/ejb/index.html>, Stand 1.12.2001.

letzten beiden Punkten konnten gute Erfahrungen mit dem verfolgten Pike-Ansatz gemacht werden.

Eine notwendige Instanziierung von Objekten zur Laufzeit des Systems und kontrolliertes Ableiten von Klassen ist wie gezeigt wesentliches Element der Selbstadministration und Erweiterbarkeit der gewählten Architektur. Hier machen alle Java-gestützten Ansätze erhebliche Eingriffe in die Laufzeitumgebung und die Sicherheitsmechanismen der Systeme notwendig. Ähnliche Probleme ergeben sich mittels einer Umsetzung in den meisten anderen, interpretierten Programmiersprachen.

Zusammenfassend ist festzustellen, dass die gewählte architektonische Verbindung aus Eigenschaften des Dokumentenmanagements und ereignisorientierten Strukturen eine geeignete Lösung darstellt. Die gezeigte Kernarchitektur kooperativer Wissensräume bietet sowohl die gewünschten Merkmale der Nutzerzentriertheit, erreicht aber auch in Bezug auf durchgängige Verfügbarkeit und der Bereitstellung von primären Medienfunktionen an Materialien die abgesteckten Ziele. Die vorliegende Arbeit charakterisiert in diesem Zusammenhang die Kernkomponenten der Gesamtarchitektur; speziell im Bereich spezieller Clients, wie kooperativer semantischer Raumdarstellungen, eröffnen sich eine ganze Reihe viel versprechender architektonischer Möglichkeiten.

Zusammenfassung und Ausblick

Nachfolgend der Entwicklung des Konzepts kooperativer Wissensräume und der architektonischen Ausformulierung eines entsprechenden technischen Systems gilt es eine Reihe von Fragen zu beantworten: Zunächst stellt sich die Frage nach dem Erreichten, den Lösungen und Antworten, die sich durch das Konzept kooperativer Wissensräume ergeben. In einem nächsten Schritt sind die gewonnenen Ergebnisse sowohl konzeptuell als auch technisch kritisch zu bewerten. Schließlich lassen sich eine Reihe von Perspektiven und Möglichkeiten ausmachen, die sich auf der Basis der vorliegenden Arbeit weiter bearbeiten und erforschen lassen. Hierbei sind zum einen Perspektiven zu nennen, die sich aus dem konzeptuellen Modell kooperativer Wissensräume ergeben, zum anderen Möglichkeiten zu untersuchen, die das konkrete technische System und seine zugrunde liegende Architektur bieten.

Ausgangspunkt der vorliegenden Arbeit ist der primär serverzentrierte Umgang mit digitalen Medien. Hier bildet das WWW den Quasi-Standard der verwendeten Werkzeuge und Technologien. Eine computergestützte Lehre mit Hilfe des WWW reduziert sich für die Lernenden auf das Verfolgen von Verweisen und Herunterladen von Materialien. Veränderungen an den Lehrmaterialien, wie das Schaffen persönlicher Strukturierungen oder ihre Umgestaltung, sind nur wenigen Lehrenden, nicht aber den Lernenden vorbehalten. Kooperative Bearbeitungsprozesse an Materialien, wie gegenseitige Kommentierung (Annotation), deren Austausch, das Schaffen gemeinsamer Sichten und kooperatives Arbeiten daran sind entsprechend nicht bzw. nur durch isoliert von der Gesamtstruktur vorliegende Werkzeuge möglich. Derartige Werkzeuge, wie auch Hypertextsysteme oder multimediale Lernumgebungen bilden isolierte Insellösungen, die sich unter dem Anspruch der durchgängigen Verfügbarkeit nur schwer in moderne Lernprozesse integrieren lassen.

Entsprechend ist das Dilemma deutlich zu erkennen: Hypertextsysteme und kooperationsunterstützende Werkzeuge bieten innovative Ideen und Konzepte, diese sind jedoch für die WWW-gestützte Lehre nicht hinreichend verwirklicht und in bestehende Lehrstrukturen nur unzureichend integrierbar.

Gleichzeitig hat es im Hinblick auf eine Vielzahl von technischen Lösungen und Ansätzen an einem theoretischen Modell gefehlt, um den technischen Mehrwert einzelner Systeme gegeneinander abgrenzen und bewerten zu können.

Die vorliegende Arbeit hat die genannten Problembereiche schrittweise analysiert und durch eine konstruktive Vorgehensweise bearbeitet.

Ausgehend von dem genannten Defizit wurde zunächst das theoretische Modell der Medienfunktionen gewählt und zum Modell der kooperativen Medienfunktionen weiterentwickelt. Hierzu wurden zunächst die primären Medienfunktionen weiter systematisiert und eine Differenzierung in primäre, individuelle sowie primäre, kooperative Medienfunktionen vorgenommen.

Die primären, individuellen Medienfunktionen wurden um die Medienfunktion des Lösens ergänzt und kooperative Tätigkeiten des Übertragens, Synchronisierens und Zugreifens als grundlegende kooperative Medienfunktionen definiert.

Damit lag eine erste Bewertungsgrundlage vor, um den technischen Mehrwert einer gefundenen Lösung aus medientechnischer Perspektive bestimmen – und gleichzeitig technische von didaktischen oder pädagogischen Fragestellungen differenzieren zu können.

Als zweites wesentliches Instrument wurde das Konzept der Medienbrüche aufbauend auf dem Kriterium der durchgängigen Verfügbarkeit von Materialien gewählt. Es eignet sich, um Brüche in den verwendeten Medien begleitend einem Lernprozess aufzeigen und damit die technischen Unterstützungsfunktionen eines Systems bewerten zu können.

Wesentlicher Schritt auf dem Weg zum Konzept kooperativer Wissensräume war nun, eine Reihe von typischen Szenarien der Einbettung kooperativer Medienfunktionen in die Lehre zu entwickeln. Diese orientieren sich an der realen Praxis basierend auf den Erfahrungen der Arbeitsgruppe Informatik und Gesellschaft, Paderborn, und identifizieren wesentliche Grundfunktionen und Handlungen kooperativer Lernprozesse mit Hilfe virtueller Wissensräume.

Im Vordergrund der Szenarien zu Vorlesung, Übung, Seminar und Projektgruppe standen Möglichkeiten, kooperativ mit Materialien umgehen zu können. Lernende sollen von allen Stätten des Lernens auf Materialien zugreifen und diese aktiv bearbeiten, austauschen und in einen persönlichen Kontext setzen können. Hierbei zeigen sich Tätigkeiten wie Verknüpfen und Arrangieren, aber auch Übertragen und Zugreifen als wesentliche Grundfunktionen. Hinzu kommt die Notwendigkeit, die eigene Lernumgebung selbst mit- und umgestalten zu können. Das Erzeugen neuer virtueller Wissensräume, aber auch die Strukturierung der Lernprozesse obliegt den Lernenden und Lehrenden gleichermaßen.

Eine erste konzeptuelle Vorstellung kooperativer Wissensräume, d.h. der bereitzustellenden Handlungsmöglichkeiten im Umgang mit Materialien, folgte unmittelbar aus den formulierten Szenarien – es ergab sich ein erstes Bild notwendiger Tätigkeiten in Bezug auf die bereitzustellenden primären Medienfunktionen. Basierend auf dieser Vorstellung konnten zudem eine Reihe von existierenden und zunächst viel versprechenden Systemen auf ihre Eignung zur Umsetzung der formulierten Szenarien geprüft werden.

Hierbei wurden zu Teilbereichen der Szenarien jeweils für eine Klasse von Systemen stellvertretend innovative Ansätze auf ihre Umsetzbarkeit geprüft. Eine erste Analyse machte deutlich, dass vorhandene Systeme verschiedene innovative Ansätze und Ideen verkörpern, sich diese jedoch aus verschiedenen Gründen nicht für einen alltäglichen, praktischen Einsatz in der Lehre eignen. Keines der untersuchten Systeme bietet für die Lernenden ausreichende, primäre Medienfunktionen bei gleichzeitiger durchgängiger Verfügbarkeit der Lehrmaterialien. Das kooperative Erzeugen, Arrangieren oder Übertragen von Materialien bei gleichzeitiger Strukturierung der Lernumgebung durch die Lernenden selbst ist zumeist nur sehr rudimentär entwickelt. Nur in wenigen Aspekten sind Funktionen erkennbar, beliebige, eigene Strukturen und Gestaltungen erzeugen und diese mit anderen Lernenden teilen zu können. Gleichzeitig existieren kaum Ansätze, die wesentliche Entwicklungen und Standards des WWW ausreichend berück-

sichtigen. Das WWW hat sich durch seine universelle Verfügbarkeit und allgemeine Akzeptanz als prädestinierte Plattform zur Gestaltung von netzgestützten Diensten entwickelt und gibt damit wesentliche technische Rahmenbedingungen vor.

Aufbauend auf den formulierten Szenarien und der Analyse existierender Systeme wurde in einem nächsten Schritt das Kernkonzept kooperativer Wissensräume entwickelt. Hierbei wurden die primären Medienfunktionen in die Metapher des virtuellen Raums eingebettet und ihre konkreten Handlungsmöglichkeiten dargestellt. Das Schwergewicht dieser Einbettung oblag der Entwicklung eines geeigneten Modells von Benutzergruppen und Rechten und der Auswahl geeigneter Repräsentationsformen des virtuellen Raums, aber insbesondere der konkreten Ausformulierung von Tätigkeiten des Kreierens, Löschens, Arrangierens, Verknüpfens, Annotierens und Zugreifens anhand eines konkreten Objektmodells der Repräsentation von Materialien.

Ziel war dabei weniger die Vielfalt existierender Forschungen und Ideen zu repräsentieren, vielmehr wesentliche Designkonflikte in der Ausformulierung des Konzepts kooperativer Wissensräume zu lösen und damit eine den Alltagsbedingungen standhaltende Konzeption zu schaffen. Es galt also weniger, die Breite möglicher Lösungen und existierender Ansätze vollständig zu bearbeiten, als vielmehr die wesentlichen Aspekte im kooperativen Umgang mit Materialien aus medientechnischer Perspektive zu destillieren.

Den technischen Schwerpunkt der vorliegenden Arbeit bildet die Konzeption und die architektonische Ausformulierung kooperativer Wissensräume. Hierzu mussten zunächst verschiedene architektonische Alternativen und Realisierungsformen gegeneinander abgegrenzt und bewertet werden.

Grundidee der gewonnenen Architektur und in dieser Form neu ist die Verknüpfung verschiedener Technologien ereignisbasierter virtueller Welten mit Fähigkeiten zur netzgestützten, kooperativen Dokumentenverarbeitung. Kernkonzept der gewonnenen Client-Server-Architektur ist die Repräsentation von Lernmaterialien, von Nutzern, aber auch der Struktur kooperativer Wissensräume durch kooperative Objekte und Attribute.

Das sTeam-System, die technische Umsetzung der im Rahmen der vorliegenden Arbeit entwickelten Kernarchitektur, bietet damit eine Laufzeitumgebung und eine Persistenzebene für Objekte. Objekte sind die technische Repräsentation der Materialien und Kommunikationsmittel kooperativer Wissensräume. Sie kommunizieren durch Nachrichtenaustausch und Ereignisse (Events). Events werden mittels verschiedener Schnittstellen auch zwischen dem Server und den Clients übermittelt und schaffen auf diese Weise eine Synchronizität der bereitgestellten Sichten auf kooperative Objekte.

Ein offenes und leistungsfähiges Konzept aus Benutzerrechten und Benutzergruppen schafft die Voraussetzung für eine dezentrale Organisation und Selbstverwaltung der Lernumgebung. Die Klassen- und Objektstruktur erlauben die kontrollierte Instanziierung von Objekten zur Laufzeit des Systems und bieten die Registrierung von Attributen an spezifischen Verwaltungsinstanzen (Factories).

Auf diese Weise verleiht das leistungsfähige Event-System in Verbindung mit der Definition verschiedener Schnittstellen des Zugriffs auf kooperative Objekte (Materialien) sowie Mechanismen der flexiblen Registrierung von Attributen an Factories der Kernarchitektur einen weitgehend universellen Charakter.

Das Vorhaben der Arbeit, eine konzeptuelle und technische Basis für kooperative Wissensräume zu schaffen, ist mit der Umsetzung einer geeigneten Kernarchitektur verwirklicht worden. Das Modell kooperativer Wissensräume wurde in ein technisches Rahmenkonzept und System übertragen.

Letzteres konnte zum gegebenen Zeitpunkt erstmalig im Rahmen einer Lehrveranstaltung im praktischen Einsatz erprobt werden. Hierzu wurden in einem ersten Schritt vorrangig WWW-konforme Benutzerschnittstellen entwickelt, um die durchgängige Verfügbarkeit des Systems sicherzustellen. Parallel entstanden spezielle Clients, wie eine Shared Whiteboard, die auch synchrone Handlungen in kooperativen Wissensräumen zulassen.

Damit ist als wesentliches Ergebnis der vorliegenden Arbeit, neben der Ausformulierung des Konzepts kooperativer Wissensräume, die Entwicklung einer flexiblen Rahmenarchitektur zu nennen. Bestimmte Forschungsfragen und Erfahrungen der alltagspraktischen Nutzung lassen sich erst mit Hilfe dieser Rahmenarchitektur weiter untersuchen.

In dieser Form liefert die Arbeit das Fundament eines Systems, das sich im alltäglichen Einsatz befindet, aber gleichzeitig Plattform zur Erörterung weiterer Forschungsfragen bildet – also eines Systems, das unter alltagspraktischen Bedingungen selbstständig weiter zu wachsen vermag.

Ausgehend von dem formulierten Anspruch einer alltagspraktischen Nutzung lassen sich zwei Kernfragen zukünftiger Ausbaumöglichkeiten und Perspektiven nennen. Zunächst ist die Untersuchung der Tragweite und der Möglichkeiten des *Konzepts virtueller Wissensräume* zu hinterfragen. Eng mit dieser konzeptuellen Erkundung verzahnt ist anschließend die Frage nach den *technischen Ausbaumöglichkeiten der Rahmenarchitektur virtueller Wissensräume* zu stellen.

Für beide Kernfragen sind im Folgenden erste Überlegungen zusammengetragen, die Ansatzpunkte für weitergehende Forschungen und Entwicklungen darstellen.

Perspektiven für das Konzept virtueller Wissensräume

Auf den erst genannten Bereich möglicher zukünftiger Weiterentwicklungen des Konzepts kooperativer Wissensräume bezogen, lassen sich perspektivisch eine ganze Reihe von Forschungsbereichen nennen. Hierzu zählen Modelle der Navigation in kooperativen Wissensräumen, Formen der Repräsentation von Wissensstrukturen, Möglichkeiten der Strukturierung von Kommunikationsprozessen entlang der Materialien eines Lernprozesses oder Fragen des Datenschutzes in kooperativen Wissensräumen.

Mit Hilfe der Kernarchitektur und des zugehörigen Theoriemodells können erstmals Fragen der Navigation in virtuellen, kooperativen Wissensräumen in einem konkreten Anwendungsfeld erforscht werden. Hierzu eignet sich das entwickelte Kernkonzept kooperativer Wissensräume insbesondere durch sein flexibles Objektmodell, d.h. der Repräsentation von Materialien in Objekten und Attributen. Virtuelle Räume und Container sind flexible Strukturierungselemente von Materialien. Verschiedene Verfahren und Mechanismen der Navigation in diesen Strukturen lassen sich nun durch unterschiedliche Sichten (Clients) auf die Materialien umsetzen und in der Praxis erproben. Basierend auf dem entwickelten Konzept und der technischen Ausformulierung kooperativer Wissensräume lassen sich notwendige Strukturierungsinformationen in Attributen zu den jeweiligen Materialien (Objekten) ablegen und auf diese Weise auch eine Anzahl von Sichten parallel anbieten.

Eng verbunden ist ein derartiges Vorgehen mit der Entwicklung neuer Formen der Vorstrukturierung und späteren Feinstrukturierung von Wissen. Aktuell in der Forschung diskutierte Standards der Repräsentation und Verknüpfung von Wissen, wie die Topic Maps, gilt es in diesem Zusammenhang zu berücksichtigen. Letztere sind Ansatzpunkt einer normierten Repräsentation von Assoziationen und Abhängigkeiten (semantischen Beziehungen) zwischen Informationsknoten. Zukünftig gilt es, Topic Maps in kooperative Szenarien der gemeinsamen Erstellung und Nutzung begleitend von Lernpro-

zessen zu überführen und auf diese Weise Lernenden die Möglichkeit zum *kooperativen* Aufbau besserer Navigationsstrukturen in Lehrmaterialien zu eröffnen.

Auch in letzterem Punkt bietet das Konzept kooperativer Wissensräume in der Praxis ein Erprobungsfeld zur schrittweisen Erforschung notwendiger Assoziationen zwischen einzelnen Materialien (Knoten). Zum einen kann vorstrukturiertes Wissen als Teil des Lernprozesses ausgebaut und durch die Lernenden neu strukturiert werden, zum anderen eignen sich derartige Strukturierungen, um Lernenden den Einstieg in einen Wissensraum zu erleichtern.

Ebenfalls weitgehend unerforscht, aber für das Konzept kooperativer Wissensräume unerlässlich, ist die gemeinsame Konstruktion und Entwicklung grafischer Repräsentationsformen kooperativer Wissensstrukturen. Wie an einigen Stellen der vorliegenden Arbeit angedeutet, können semantische Strukturen als Ausdrucksformen der Medienfunktionen des Arrangierens und Verknüpfens wesentliche Orientierungshilfen und Navigationselemente eines Lernprozesses darstellen. Forschungen über die Gestalt und Interaktionsformen von semantischen Wissensrepräsentationen bieten zurzeit noch wenig in die Praxis überführbare Lösungen. Dies liegt zu einem guten Teil neben fehlender, konzeptioneller Ansätze auch an einer nicht vorhandenen, technischen Unterfütterung durch geeignete, kooperative Umgebungen. Kooperative Wissensräume erlauben mit begrenztem Aufwand die Entwicklung möglicher grafischer Repräsentationen und notwendiger Zugangswerkzeuge. Diese können auf der Basis kooperativer Wissensräume konzipiert, technisch umgesetzt und in der Praxis erprobt werden.

Weitere wesentliche konzeptuelle Perspektiven kooperativer Wissensräume beziehen sich auf neue Verfahren der Strukturierung von Kommunikationsprozessen innerhalb des virtuellen Raums und ihrer Zuordnung zu den Lehrmaterialien eines Lernprozesses. Existierende Ansätze weisen meist keinerlei übergreifende Mechanismen des Knüpfens von Kommunikationsprozessen an die Materialien eines Lernprozesses auf.

Mit Hilfe kooperativer Wissensräume können sowohl synchrone als auch asynchrone Formen der Kommunikation an die Objekte eines virtuellen Raums gebunden werden. Dabei ist die Struktur und Größe einer computergestützten Diskussion nicht durch einen Kanal oder ein Nachrichtenbrett begrenzt, sondern von den Lernenden selbst organisierbar und an die Struktur der Materialien eines Lernprozesses geknüpft. So kann das Erzeugen eines neuen Dokuments zugleich der Ausgangspunkt zu einer Diskussion in Form von sich aufeinander beziehenden Annotationen sein. Gleichzeitig bietet jedes Dokument potenziell einen Kanal für synchrone Kommunikation. Auch können „virtuelle Diskussionstische“ Lernende zu flexiblen Diskussionsgruppen zusammenfassen. Das Platzieren von Materialien bzw. einem Dokument auf einem virtuellen Tisch kann zugleich die Kopplungsmodi synchroner Formen der Zusammenarbeit steuern. Es lassen sich auf diese Weise Fragen der nötigen Kopplungsgrade beim gemeinsamen Bearbeiten von Materialien untersuchen oder neue Mechanismen der Strukturierung von Kommunikation durch die Materialien einer Lernumgebung erproben.

Schließlich wirft das Konzept kooperativer Wissensräume neue Fragestellungen in dem Bereich des Datenschutzes und der Anonymität von Lernenden und Materialien auf. Ähnlich wie schon im WWW hinterlassen Lernende auch in virtuellen Wissensräumen Datenspuren. Mit den Interaktionsmöglichkeiten und der Flexibilität kooperativer Wissensräume wächst zugleich die Gefahr der Zusammenführung persönlicher Daten eines Lernenden, den Verbindungsdaten der Nutzer und den Kommunikationsinhalten. Damit lassen sich Benutzerprofile von Lernenden herstellen. Beispielsweise wäre (technisch) feststellbar, wer welche Lehrmaterialien gelesen und sich mit welcher Häufigkeit z.B. an Diskussionen beteiligt hat. Die Zusammenführung derartiger Informationen ist

aus Gründen des Datenschutzes unzulässig. Kooperative Wissensräume müssen speziell aus dem Anspruch einer offenen und nutzerzentrierten Ausrichtung heraus Mechanismen entwickeln, um die Erfassung und Zusammenführung unterschiedlichster Protokoll- und Nutzerdaten zu verhindern.

Beispiele sind der mögliche Einsatz von Pseudonymen für Nutzer oder die Bereitstellung von Mechanismen der nachträglichen Anonymisierung von Interaktionen oder Materialien.

Eng verbunden mit letzterem Punkt ist die Gratwanderung zwischen Anonymität der Lernenden und notwendigem Kontext der kooperativen Zusammenarbeit. Verschiedene Stufen und Formen der Awareness-Informationen, speziell der Aktivitäts-Awareness-Informationen, sind dabei sorgsam und für die Nutzer transparent und steuerbar zu gestalten. In der Literatur finden sich zwar eine ganze Reihe möglicher Visualisierungsformen und Strategien zur Ermittlung von Wahrnehmungsinformationen, jedoch kaum Lösungen, die den Aspekt des Datenschutzes ausreichend berücksichtigen.

Perspektiven des technischen Systems

Der Bereich der technischen Entwicklungsperspektiven kooperativer Wissensräume ist eng mit den schon genannten Ausbaumöglichkeiten verzahnt. Primäres Ziel ist hier die Gestaltung einer unter Alltagsbedingungen sich bewährenden, technischen Lösung. Dabei gilt es im Zuge der praktischen Nutzung eine Reihe von Designalternativen zu berücksichtigen und verschiedene Designkonflikte zu lösen. Insbesondere sind technische Standards und bestehende Entwicklungslinien zu berücksichtigen. Hierzu zählen die praxisorientierte Gestaltung von Benutzerschnittstellen, das Schaffen neuer, aktiver Elemente innerhalb kooperativer Wissensräume, Ausbau und Erweiterung von unterstützten Protokollen und Schnittstellen des Servers sowie die Skalierbarkeit des Servers.

Ein erstes wesentliches Feld umfasst die weitere Gestaltung von Benutzerschnittstellen aufbauend auf den Grundstrukturen des entwickelten Kernkonzepts. Schwerpunkte bilden die Einbettung von Funktionen der gegenseitigen Wahrnehmung und der Darstellung und Manipulation von Gruppenstruktur und Benutzerrechten. In beiden Bereichen sind neue Verfahren und Bedienmetaphern zu erproben und in geeignete Benutzerschnittstellen zu überführen. Sowohl das Erscheinungsbild und die Interaktionsmöglichkeiten der angestrebten Schnittstellen (Ergonomie) als auch ihre Funktionalität und ihre notwendige Skalierbarkeit der Funktionalität sind hierbei zu berücksichtigen.

Ein Beispiel ist die Repräsentation von Zugangsberechtigungen durch die Vergabe von virtuellen Schlüsseln. Diese gewähren den Zugang zu bestimmten Räumen einer Lernumgebung. Die Weitergabe eines Schlüssels entspricht der Übertragung der Zugangsberechtigung zu einem virtuellen Raum – ein Verfahren, welches insbesondere die Weitergabe (Delegation) von Berechtigungen auf wenige Grundfunktionen zu reduzieren vermag. Das entwickelte technische Kernkonzept besitzt in seiner Grundstruktur ein hohes Maß an Flexibilität zur Umsetzung derartiger Funktionen. Beispielsweise ließe sich das obige Modell der Vergabe von Rechten durch eine geeignete Benutzungsschnittstelle auch für die Web-Schnittstelle erproben bzw. durch Reduzierung des generischen Rechtemodells auf die Berechtigung zum Betreten eines Raums reduzieren.

Mit Hilfe der definierten Client-Server-Schnittstellen und einer Fortführung des ergebnisbasierten Modells interagierender gemeinsamer Objekte (Shared Objects) bis auf die Ebene des Clients lassen sich mit überschaubarem Aufwand existierende Clients den Anforderungen aus der Praxis anpassen und zugleich neue Lösungen umsetzen und erproben. Hierbei bleibt die Struktur des Servers durch flexible Mechanismen der Registrierung von Attributen an Objekten weitgehend unberührt. In dieser Form lassen sich auf elegante Weise sehr unterschiedliche technische Sichten auf den kooperativen Wis-

sensraum schaffen und diese für spezifische Anforderungen aus dem Nutzungsumfeld anpassen. Ergebnis können für unterschiedliche Nutzerkreise gestaltete Clients sein, z.B. eigene Benutzerschnittstellen/Clients für das schulische Umfeld.

Die Erweiterbarkeit und Flexibilität des Servers liegt zudem in verschiedenen Mechanismen des Schaffens von aktiven Objekten (Protokollwerkzeuge, Abgabekasten für Übungslösungen etc.) begründet. Zukünftig gilt es, für weitere Bereiche kooperativer Lernsituationen geeignete Werkzeuge in kooperative Wissensräume zu integrieren. Auf das obige Beispiel der Vergabe von Berechtigungen durch virtuelle Schlüssel bezogen ließe sich etwa die Vergabe von Schlüsseln über ein aktives Element, das virtuelle „Schlüsselbrett“ realisieren – es besitzt eine festlegbare Anzahl von Zugangsberechtigungen zu einem Raum und kontrolliert die Vergabe und Weitergabe dieser Berechtigungen in Form der virtuellen Schlüssel. Aus technischer Sicht sind zukünftig sowohl die notwendigen Programmierschnittstellen des Servers als auch Verfahren der Entwicklung aktiver Elemente und ihrer Benutzerschnittstellen auszubauen und zu vereinfachen.

Weitere wesentliche, technische Erweiterungsmöglichkeiten bildet der Bereich der Serverentwicklung. Die mögliche Nutzung vorhandener (oder zukünftiger) Standards bezieht sich zunächst auf die Client-Server (COAL) Schnittstelle. COAL wurde als spezielle Ausprägung des RMI-Verfahrens zunächst im Hinblick auf eine übersichtliche und den speziellen Erfordernissen des sTeam-Systems gerecht werdende Lösung konzipiert und umgesetzt. Im Zuge der weiteren Standardisierung netzgestützter Interaktionsformen ist die COAL-Schnittstelle ggf. durch einen geeigneten Standard zu ersetzen, um auch nicht spezifisch für die sTeam-Umgebung entwickelten Werkzeugen einen Zugriff auf den Server zu ermöglichen. Insbesondere im Umfeld der angestrebten weiteren Open-Source Entwicklung des Kernsystems handelt es sich hierbei um einen wichtigen Schritt. In diesem Zusammenhang ist insbesondere der CORBA-Standard auf seine mögliche Eignung zu prüfen. Auch der Einsatz von Multicast-Technologien²⁸⁹ zu einer effizienteren Client-Server-Kommunikation gilt es in Betracht zu ziehen.

Aufseiten des dateiorientierten Zugriffs auf die Materialien des Servers sind weiterhin der WebDAV-Standard und seine Erweiterungen²⁹⁰ zu berücksichtigen und für den Server ggf. verfügbar zu machen. Letzteres bezieht sich sehr stark auf die bessere Integrationsmöglichkeit existierender Werkzeuge der Arbeitsumgebung der Lernenden (beispielsweise von Textverarbeitungs- oder Grafikverarbeitungsprodukten) in die Lernumgebung.

Weitere, in Entwicklung befindliche Standards, wie die Annotation an beliebige Elemente des WWW (Annotea-Standard) oder die freie Verknüpfbarkeit von XML-basierten Dokumenten (XLINK-Standard) tragen ein gutes Stück zu interaktiveren Formen des WWW bei. Die Anpassung und der Ausbau des Servers müssen diesen verschiedenen Strömungen Rechnung tragen und gleichzeitig neue Konzepte forcieren. So definiert der Annotea-Standard an sich lediglich elementare Schnittstellen einer Annotation von Objekten bzw. Protokollerweiterungen. Die wichtige Funktion der Einbettung bleibt der konkreten kooperativen Umgebung überlassen. Erst seine technisch und konzeptionell hinreichend gelungene Integration in kooperative Wissensräume erlaubt die Nutzung von Annotationen als Teil kooperativer Lernprozesse. Hierzu zählen speziell die Einbeziehung von Mechanismen der gegenseitigen Wahrnehmung und ihre sinnvolle Integration in das Benutzer- und Rechtemodell der Lernumgebung.

²⁸⁹ Vgl. [Multicast 2000].

²⁹⁰ Vgl. hierzu beispielsweise die IETF Delta-V Arbeitsgruppe: <http://www.webdav.org/deltav>, Stand 1.12.2001.

Ähnliche Erfordernisse ergeben sich im Umfeld der Integration von Audio-Kommunikationsmechanismen in die beschriebene Architektur. Basierend auf der Technologie von Telekonferenzen lassen sich neben rein textueller Kommunikation auch Audio-Funktionalitäten in virtuellen Räumen nutzen. Einkanalige Audio-Konferenzen, von Teilnehmer zu Teilnehmer, sind in diesem Zusammenhang durch verfügbare Komponenten problemlos in die bestehende Architektur integrierbar. Das technische Rahmenkonzept virtueller Wissensräume bietet bereits die Mechanismen der Adressierung der Teilnehmer innerhalb eines virtuellen Raums; diese können mit einem entsprechenden Werkzeug zur Audio-Kommunikation abgestimmt werden. Um einiges interessanter ist jedoch die Gestaltung von Mehrbenutzer-Audio-Konferenzlösungen.²⁹¹ So kann der virtuelle Raum als natürliche Begrenzung für die Reichweite von Audio-Signalen gelten. Beispielsweise sind Verfahren denkbar, dass sich Diskussionsgruppen an realweltlichen Gegebenheiten des Raums orientieren (z.B. abnehmende Lautstärke bei sich vergrößerndem Abstand einer Person zu einer Diskussionsgruppe).

Ein weiterer Kernbereich der Ausbau- und Erweiterungsmöglichkeiten der Rahmenarchitektur bezieht sich auf die Skalierbarkeit des Servers. Mit einer wachsenden Anzahl von Nutzern steigt das Aufkommen simultaner Zugriffe auf den Server und die Größe der Datenbank zur Ablage entsprechender Materialien. Es lässt sich ein Punkt ausmachen, an dem auch durch steigende verfügbare Ressourcen wie Prozessorleistung, Netzbandbreite und Speicherkapazität dieser Prozess nicht weiter kompensiert werden kann.

Ein erster Ansatzpunkt wäre es, den Server durch eine Anzahl verschiedener Prozesse (Multi-Threaded-Server) auf einem Multi-Prozessor-Rechner skalierbar zu gestalten.²⁹² Schon in der vorliegenden Implementierung sind verschiedene Teile des Servers (beispielsweise das Aktualisieren von Objekten in der Datenbank sowie der Roxen-Webserver und die MySQL-Datenbank) als eigenständige Prozesse ausgeführt. Eine wirkliche Implementierung des Kernservers als Multi-Prozess-Architektur erfordert Erweiterungen in der Ereignisverwaltung sowie Mechanismen des Transaktionsmanagements. Problematisch erweist sich hierbei, dass Pike zum gegebenen Zeitpunkt nur wenig Unterstützung zu einer Multi-Prozess-Architektur bietet.

Eine verteilte Serverarchitektur erscheint um einiges viel versprechender. Ein erster Ansatzpunkt wäre es, die virtuelle Welt durch einen Verbund kommunizierender Server in jeweils von verschiedenen Servern verwaltete Bereiche aufzuteilen. Benutzerdaten und Verknüpfungen zwischen Objekten dürfen von dieser verteilten Architektur nicht beeinflusst sein, sie müssen auch über Servergrenzen hinweg eindeutig gewählt werden.²⁹³ Hinzu kommen eindeutige Objektschlüssel und Mechanismen der Partitionierung einer virtuellen Welt in verschiedene Bereiche. Die Skalierbarkeit des Servers bis auf die Ebene einzelner Objekte (der Möglichkeit, jedes Objekt prinzipiell auf einem anderen Server abzulegen) verursacht noch erheblich umfangreichere Erweiterungen und ggf. eine Neugestaltung der gewählten Architektur. Hier erscheint die Skalierbarkeit eines Servers bis auf die Ebene einer Fachgruppe oder Veranstaltung als sinnvoller Kompromiss.

²⁹¹ Das Fraunhofer Institut für Grafische Datenverarbeitung forscht seit einiger Zeit in diesem Bereich (vgl. [Chodura 2000]).

²⁹² Vgl. hierzu die verschiedenen Architekturen von Multi-Prozess-Servern, beispielsweise die Zuordnung von Zugriffen zu Prozessen wie „Thread-per-request“, „Thread-per-connection“ oder „Thread-per-object“ (vgl. [Schmidt 1998, S. 56ff.]).

²⁹³ Hierzu sind Mechanismen der Replikation der Benutzer- und Gruppenstruktur zwischen den verschiedenen Servern bzw. ein ausgewiesener Server zu deren Verwaltung bereitzustellen.

In ähnlicher Weise sieht die vorgestellte Konzeption zurzeit lediglich eine mögliche Datenbank zur Ablage von Objekten vor. Speziell im Bereich multimedialer Daten wie Audio- oder Videoelemente sind optimierte, an Echtzeitsituationen angepasste (Streaming-) Datenbanken unumgänglich. Dementsprechend sind mittelfristig verschiedene Typen von Datenbanken simultan und unabhängig voneinander zu integrieren.

Der letzte hier genannte Punkt technischer Erweiterungen bezieht sich auf die Sicherheitsaspekte des Systems. Hier sind verschiedene Sicherheitsmodelle der Client-Server-Verbindungen zu nennen. In der derzeitigen Lösung besteht die Möglichkeit, über den Roxen-Webserver eine sichere HTTPS-Verbindung zum Zugriff auf die Webschnittstelle zu nutzen. Verbindungen synchroner Clients über die COAL-Schnittstelle sind unverschlüsselt. Hier ist zu prüfen, inwieweit sich ebenfalls der Secure Socket Layer (SSL)-Standard zur Verschlüsselung derartiger Netzwerkverbindungen eignet. Auch die Übertragung und Speicherung von Passwörtern spielen in diesem Zusammenhang eine wichtige Rolle. Entsprechende Bibliotheken sind sowohl für Java als auch für Pike vorhanden. In jedem Fall ist sicherzustellen, dass es für Unbefugte nicht mit einfachen technischen Mitteln möglich ist, an geschützte Daten innerhalb kooperativer Wissensräume zu gelangen, z.B. durch das Ausspionieren von über das Netzwerk übermittelten Kennworten.

Wie in der vorliegenden Arbeit ausführlich gezeigt, existieren eine Fülle von Settings und Anwendungsbereichen für kooperative Wissensräume. Zugleich ist die Anzahl von Lösungsvorschlägen und technischen Systemen fast unüberschaubar. Damit kann es sicherlich nicht die einzige, optimale Lösung geben.

Das technische Rahmenkonzept kooperativer Wissensräume wählt einen pragmatischen Weg, um einen Ansatz zu schaffen, der unter Alltagsbedingungen angepasst und erweitert werden kann.

Hierbei ist die Umsetzung sicherlich noch mit einer Reihe von Einschränkungen versehen. Hierzu zählt beispielsweise das Problem der technischen Verzahnung synchroner Clients mit Webbrowsern, die Problematik der Integration von Fremdmaterialien in die Lernumgebung oder die Annotation von Dokumenten des WWW. Zu bemerken ist jedoch, dass ein Großteil der genannten Einschränkungen der konzeptuellen Restriktionen des WWW geschuldet sind. Das Konzept kooperativer Wissensräume und insbesondere ihre technische Umsetzung haben sich in vielen Bereichen an den gesetzten technischen Rahmenbedingungen des WWW zu orientieren, bieten aber durch ihre offene und flexible Architektur auch in Zukunft gute Chancen, derartige Defizite durch neue Standards und technologische Entwicklungen zu beseitigen.

Die vorliegende Arbeit trägt zukünftigen Entwicklungen durch zwei wesentliche Strategien Rechnung: Zum einen sind virtuelle Wissensräume eine Plattform und ein Laborrahmen für die Erforschung und Erprobung neuer Entwicklungen und Fragestellungen. Zum anderen sind sie ein praxistaugliches System, welches in verschiedenen Bereichen der Kooperationsunterstützung effektiv genutzt werden kann.

Unterstützt werden beide Entwicklungslinien durch die Entscheidung, das sTeam-System konsequent als Open-Source-Ansatz zu gestalten und damit ein Instrument für die freie Nutzung und Weiterentwicklung zu schaffen.

Abbildungsverzeichnis

Abbildung 2-1: Primäre individuelle und kooperative Medienfunktionen	44
Abbildung 2-2: Automatisierte Übungsabgabe.....	77
Abbildung 2-3: Struktur aus virtuellen Räumen	86
Abbildung 3-1: Raumstruktur	111
Abbildung 3-2: Typische Benutzer- und Gruppenstruktur	127
Abbildung 3-3: Kontextuelle Vererbung	134
Abbildung 3-4: Kontextuelle Vererbung – Verschieben eines Dokuments in den Rucksack eines Benutzers	135
Abbildung 3-5: Objektkapsel, Inhalt und Umgebung	143
Abbildung 3-6: Selbstadministration – Erzeugen eines Raums – Weitergabe von Berechtigungen.....	146
Abbildung 3-7: Annotationen sind Verweise auf Objekte	150
Abbildung 4-1: Gemeinsame Objekte – Shared Objects	165
Abbildung 4-2: Zentralisierte Model View Controller (MVC)-Architektur	171
Abbildung 4-3: Replizierte Architektur	172
Abbildung 5-1: Klassenstruktur	193
Abbildung 5-2: Factories – Instanzieren von Objekten	198
Abbildung 5-3: Registrieren von Attributen an Factories	199
Abbildung 5-4: Nutzer-, Client- und Server-Attribute.....	202
Abbildung 5-5: Abonnieren von Events	205
Abbildung 5-6: Benachrichtigung der Abonent-Objekte	206
Abbildung 5-7: Phasen der Auslösung eines Events.....	207
Abbildung 5-8: Connector-Objekte.....	209
Abbildung 5-9: Client- und serverseitige Proxy-Objekte	210
Abbildung 5-10: Auflösen von Pfaden aus der Raum-/Dokumentenstruktur	214
Abbildung 5-11: Client- Server-Schnittstellen.....	218
Abbildung 5-12: Klassifizierung von Client-Technologie.....	220
Abbildung 5-13: Selbstadministration – Anmeldung.....	236

Literaturverzeichnis

- [Alexander et al. 1977] Alexander, C., Ishikawa, S., Silverstein, M., Jacobson, M., Fiksdahl-King, I., Angel, S.: *A Pattern Language*. New York: Oxford University Press 1977.
- [Ambler 2000a] Ambler, S.W.: The Design of a Robust Persistence Layer For Relational Databases. AmbySoft Inc. White Paper, November 28, 2000. Verfügbar unter: <http://www.ambysoft.com/onlineWritings.html>, Stand 1.12.2001.
- [Ambler 2000b] Ambler, S.W.: Mapping Objects to relational databases. What you need to know and why. *IBM DeveloperWorks*, July 2000. Verfügbar unter: <http://www.ambysoft.com/onlineWritings.html>, Stand 1.12.2001.
- [Ambler 2000c] Ambler, S.W.: Distributed Object Transactions. *Software Development Magazine*. July 2000. Verfügbar unter: <http://www.ambysoft.com/onlineWritings.html>, Stand 1.12.2001.
- [Andrews et al. 1995] Andrews, K., Kappe, F., Maurer, H.: Hyper-G and Harmony: Towards the Next Generation of Networked Information Technology. In: *Proceedings of the ACM conference CHI'95*, Denver, USA, New York: ACM Press 1995, 33–34.
- [Appelt & Mambrey 1999] Appelt, W., Mambrey, P.: Experiences with the BSCW Shared Workspace System as the Backbone of a Virtual Learning Environment for Students. In: Collis, B., Oliver, R. (Hrsg.): *Proceedings of ED-MEDIA'99, World Conference on Educational Multimedia, Hypermedia & Telecommunications*, June 19-24, Seattle, Washington. Charlottesville: Association for the Advancement of Computing Education 1999, 1710–1715.
- [Appelt et al. 1998] Appelt, W., Hinrichs, E., Woetzel, G.: Effectiveness and Efficiency: The Need for Tailorable User Interfaces on the Web. In: *Proceedings of the 7th International World Wide Web Conference*, Brisbane, April 1998. (auf CD-ROM veröffentlicht – ohne Seitenzahlen)
- [Atkinson 1987] Atkinson, W.: *HyperCard*. In: Software for Macintosh Computers, Apple Computer Co, Cupertino, 1987.
- [Bacher & Ottmann 1996] Bacher, Ch., Ottmann, T.: Tools and Services for Authoring on the Fly. In: Carlson, P., Makedon, F. (Hrsg.): *Proceedings of Educational Multimedia and Hypermedia 1996*. Charlottesville: Association for the Advancement of Computing in Education 1996, 7–12.
- [Banavar et al. 1998] Banavar, G., Doddapaneni, S., Miller, K., Mukherjee, B.: Rapidly Building Synchronous Collaborative Applications By Direct Manipulation. In: Poltrock, S., Grudin, J. (Hrsg.): *Proceedings of ACM Conference on Computer Supported Cooperative Work (CSCW'98)*, November 14-18, Seattle, USA. New York: ACM Press 1998, 139–148.
- [Bannon 1992] Bannon, L.J.: Customization and Tailoring of Software Systems: Thinking about the context of tinkering and tailoring. In: *Proceedings of 2nd Oksnøen Symposium, Customizing software systems*, May 23-28, 1992, Oksnøen, Norway, 4–8.
- [Bannon & Bødker 1997] Bannon, L.J., Bødker, S.: Constructing Common Information Spaces. In: Hughes, J.A., Prinz, W., Rodden, T., Schmidt, K. (Hrsg.): *Proceedings of the ECSCW '97, Fifth European Conference on Computer Supported Cooperative Work*, September 7-11, Lancaster, UK. Dordrecht: Kluwer Academic Publishers 1997, 81–96.
- [Bartle 1983] Bartle, R.: A Voice from the Dungeon, *Practical Computing*, December 1983, 126-130. Verfügbar unter: <http://mud.co.uk/richard/avftd.htm>, Stand 1.12.2001.

- [Bartle 1985] Bartle, R.: *MUD Advanced Project Report, an internal memorandum*, CSM-73, published in the Department of Computer Science's series at the University of Essex, January 1985.
- [Bartle 1990a] Bartle, R.: *Early MUD history. Technical report*, Article posted to USENER group rec.games.mud, 1990, Verfügbar unter: <http://www.ludd.luth.se/mud/aber/mud-history.html>, Stand 1.12.2001.
- [Bartle 1990b] Bartle, R.: *Interactive Multi-User Computer Games*, Research Report British Telecom plc., December 1990, 1-114. Verfügbar unter: <http://mud.co.uk/richard/imucg.htm>, Stand 1.12.2001.
- [Beaudouin-Lafon & Karsenty 1992] Beaudouin-Lafon, M., Karsenty, A.: Transparency and awareness in a real-time groupware system. In: *Proceedings of the fifth annual ACM symposium on User interface software and technology*, November 15-18, 1992, Monterrey, USA, 171–180.
- [Beca et al. 1997] Beca, L., Cheng, G., Fox, G.C., Jurga, T., Olszewski, K., Podgorny, M., Sokolowski, P., Stachowiak, T., Walczak, K.: *TANGO - a Collaborative Environment for the World-Wide Web*. (White Paper), Northeast Parallel Architectures Center, Syracuse University, Syracuse, USA, 1–16.
- [Becker 2001] Becker, B.: Virtuelle Gemeinschaften: Ein Mythos? In: Keil-Slawik, R. (Hrsg.): *DIGITALE MEDIEN und gesellschaftliche Entwicklung. – Arbeit, Recht und Gemeinschaft in der Informationsgesellschaft*. Münster: Waxmann 2001, 193–211.
- [Becker & Mark 1998] Becker, B., Mark, G.: Social Conventions in Collaborative Virtual Environments. In: Snowdon, D., Churchill, E. (Hrsg.): *Proceedings of Collaborative Virtual Environments (CVE '98)*, June 17-19, Manchester, UK, 1998, 47–55.
- [Benford & Fahlén 1993] Benford, S., Fahlén, L.: A Spatial Model of Interaction in Large Virtual Environments. In: Michelis, G.D., Simone, C., Schmidt, K.: *Proceedings of the Third European Conference on Computer-Supported Cooperative Work -ECSCW'93*, September 13-17, Milan, Italy. Dordrecht: Kluwer Academic Publishers 1993, 109–124.
- [Bentley & Appelt 1997] Bentley, R., Appelt, W.: Designing a System for Co-operative Work on the World-Wide Web: Experiences with the BSCW System. In: Nunamaker, J.F., Sprague, R. H. (Hrsg.): *Proceedings of the HICCS '97, 30th Annual Hawaii International Conference on System Sciences*, Vol. IV, January 7-10, Wailea, Hawaii. Washington: IEEE Computer Society Press 1997, 297–306.
- [Bentley & Dourish 1995] Bentley, R., Dourish, P.: Medium versus mechanism: Supporting collaboration through customisation. In: Marmolin, H., Sundblad, Y., Schmidt, K. (Hrsg.), *Proceedings of the Fourth European Conference on Computer Supported Cooperative Work - ECSCW '95*, September 10-14, Stockholm, Schweden. Dordrecht: Kluwer Academic Publishers 1995, 133–148.
- [Bentley et al. 1997] Bentley, R., Appelt, W., Busbach, U., Hinrichs, E., Kerr, D., Sikkil, S., Trevor, J., Woetzel, G.: Basic Support for Cooperative Work on the World Wide Web. *International Journal of Human-Computer Studies Special issue on Innovative Applications of the World Wide Web* 46(6) 1997, 827–846.
- [Berners-Lee 1999] Transkript der Rede von Tim Berners-Lee am 14.04.1999: MIT Laboratory for Computer Science (LCS) 35th Anniversary celebrations, Cambridge Massachusetts. Verfügbar unter: <http://www.w3.org/1999/04/13-tbl.html>, Stand 1.12.2001.
- [Berners-Lee & Fischetti 1999] Berners-Lee, T., Fischetti, M.: *Weaving the Web*. New York: Harper Collins Publishers 1999.

- [Bertino et al. 1997] Bertino, E., Samarati, P., Jajodia, S.: An Extended Authorization Model for Relational Databases. *IEEE Transactions on Knowledge and Data Engineering* 9(1) 1997, 85–101.
- [Biemans & Hofte 1999] Biemans, M., Hofte, H.T.: Tailorability: state of the art. A users' perspective on the design, use and evaluation of tailorable systems. Telematica Instituut, internal paper, GigaPort initiative, October 15, 1999, The Netherlands. Verfügbar unter: <http://extranet.telin.nl/dscgi/ds.py/Get/File-1150> Stand 1.12.2000.
- [Bollmeyer 1997] Bollmeyer, J.: HyperMUD. *ACM SIGGROUP Bulletin* 18(1) 1997, 35–36.
- [Booch 1994] Booch, G.: *Object-Oriented Analysis and Design With Applications*. Addison-Wesley Object Technology Series, Boston: Addison-Wesley, 1994.
- [Booch et al. 1998] Booch, G., Jacobson, I., Rumbaugh, J., Rumbaugh, J.: *The Unified Modeling Language User Guide*. The Addison-Wesley Object Technology Series, Boston: Addison-Wesley 1998.
- [Bopp 2000] Bopp, T.: *Konzeption und prototypenhafte Entwicklung eines Servers zur Kooperationsunterstützung in computergestützten Lernumgebungen*. Diplomarbeit vorgelegt am Lehrstuhl für Informatik, Fachgruppe Informatik und Gesellschaft, Prof. Reinhard Keil-Slawik, Universität Paderborn, Mai 2000.
- [Borghoff & Schlichter 1995] Borghoff, U.M., Schlichter, J.H.: *Rechnergestützte Gruppenarbeit – Eine Einführung in verteilte Anwendungen*. Berlin: Springer 1995.
- [Boyd 1993] Boyd, J.: Floor control policies in multi-user applications. *INTERACT '93 and CHI '93 conference companion on Human factors in computing systems*, April 24-29, Amsterdam, The Netherlands. New York: ACM Press 1993, 107–108.
- [Bray 2001] Bray, T.: *What is RDF?* published on XML.com, O'Reilly & Associates Inc., January 24, 2001, Verfügbar unter: <http://www.xml.com/pub/a/2001/01/24/rdf.html>, Stand 1.12.2001.
- [Brennecke & Keil-Slawik 1995a] Brennecke, A., Keil-Slawik, R.: Notes on the "Alltagspraxis" of Hypermedia Design. In: H. Maurer (Hrsg.): *Educational Multimedia and Hypermedia, Proceedings of ED-MEDIA 95*. Charlottesville: Association for the Advancement of Computing in Education 1995, 115–120.
- [Brennecke & Keil-Slawik 1995b] Brennecke, A., Keil-Slawik, R.: Alltagspraxis der Hypermediagegestaltung – Erfahrungen beim Einsatz des World Wide Web und Mosaic in der Lehre. In: Böcker, H.-D. (Hrsg.): *Software-Ergonomie '95 – Mensch-Computer-Interaktion – Anwendungsbereiche lernen voneinander*. Stuttgart: B.G. Teubner 1995, 107–123.
- [Brennecke & Selke 2000] Brennecke, A., Selke, H.: Individuell, Arbeitsteilig und Kooperativ – Ein integrierter Ansatz zur Erstellung, Pflege und Nutzung multimedialer Lehrmaterialien In: Uellner, S., Wulf, V. (Hrsg.): *Vernetztes Lernen mit digitalen Medien. Proceedings der ersten Tagung "Computergestütztes Kooperatives Lernen (D-CSCL 2000)"* 23.- 24. März 2000, Darmstadt. Heidelberg: Physica-Verlag 2000, 129–143.
- [Brennecke & Selke 2001] Brennecke, A., Selke, H.: The HyperSkript Authoring Environment: An Integrated Approach for Producing, Maintaining, and Using Multimedia Lecture Material. In: Montgomerie, C., Viteli, J. (Hrsg.): *Proceedings of ED-MEDIA 2001*, Charlottesville: Association for the Advancement of Computing in Education 2001, 185–190.
- [Brennecke et al. 1997a] Brennecke, A., Engbring, D., Keil-Slawik, R., Selke, H.: Das Lehren mit elektronischen Medien lernen – Erfahrungen, Probleme und Perspektiven bei multimediagestütztem Lehren und Lernen. *Wirtschaftsinformatik* 39(6) 1997, 563–568.

- [Brennecke et al. 1997b] Brennecke, A., Schwolle, U., Selke, H.: The Evolution of an Electronic Teaching and Learning Environment. In: Müldner, T., Reeves, T.C. (Hrsg.): *Proceedings of ED-MEDIA'97, World Conference on Educational Multimedia, Hypermedia & Telecommunications*, Charlottesville (Va.): Association for the Advancement of Computing in Education 1997 (Vol I.) 98–105.
- [Broll et al. 2000] Broll, W., Meier, E., Schardt, T.: The Virtual Round Table – a Collaborative Augmented Multi-User Environment. In: Churchill, E., Reddy, M. (Hrsg.): *Proceedings of the Third International Conference on Collaborative Virtual Environments (CVE2000)*, September 10-12, San Francisco, USA. New York: ACM Press 2000, 39–45.
- [Bruckman 1994] Bruckman, A.: *Programming for Fun: MUDs as a Context for Collaborative Learning*. 16. November 1994. Präsentiert auf der National Educational Computing Conference in Boston, MA, USA, June 1994, Verfügbar unter: <ftp://ftp.media.mit.edu/pub/asb/papers/necc94.txt>, Stand 1.12.2001.
- [Bruckman 1997a] Bruckman, A.: *MOOSE Crossing: Construction, Community, and Learning in a Networked Virtual World for Kids*. PhD Dissertation Amy Bruckman, MIT Media Lab, 1997.
- [Bruckman 1997b] Bruckman, A.: MOOSE Goes to School: A Comparison of Three Classrooms Using a CSCL Environment. In: *Proceedings of CSCL 97*, December 10-14, 1997, Toronto, Canada. (veröffentlicht auf CDROM – ohne Seitenangabe)
- [Bruckman & Resnick 1995] Bruckman, A., Resnick, M.: The MediaMOO Project: Constructionism and Professional Community. *Convergence* 1(1) 1995, 94–109.
- [Bullen & Bennett 1990] Bullen, C.V., Bennett, J.L.: Learning from user experience with groupware. In: *Proceedings of ACM Conference on Computer Supported Cooperative Work (CSCW'90)*, October 1990, Los Angeles, USA, New York: ACM Press 1999, 291–302.
- [Bullock & Benford 1999] Bullock, A., Benford, S.: An access control framework for multi-user collaborative environments. In: Hayne, S.C. (Hrsg.): *Proceedings of the International ACM SIGGROUP Conference on Supporting Group Work (Group99)*, November 14-17, Phoenix, USA. New York: ACM Press 1999, 140–149.
- [Burtis 1997] Burtis, J.: Sociocognitive Design Issues for Interactive Learning Environments Across Diverse Knowledge Building Communities. *Paper presented at the Annual Meeting of the American Educational Association*, Chicago, March 24, 1997. Verfügbar unter: http://csile.oise.utoronto.ca/abstracts/soc-cog_design/, Stand 1.12.2001.
- [Büscher et al. 2000] Büscher, M., Christensen, M., Grønbæk, K., Krogh, P., Mogensen, P., Shapiro, D., Ørbæk, P.: Collaborative Augmented Reality Environments: Integrating VR, Working Materials, and Distributed Work Spaces. In: Churchill, E. Reddy, M. (Hrsg.): *Proceedings of the Third International Conference on Collaborative Virtual Environments (CVE2000)*, September 10-12, 2000, San Francisco, USA. New York: ACM Press 2000, 47–56.
- [Büse 2001] Büse, D.: *Konzeption und prototypische Umsetzung eines Shared Whiteboard für eine kooperationsunterstützende Lernumgebung*. Diplomarbeit vorgelegt am Lehrstuhl für Informatik, Fachgruppe Informatik und Gesellschaft, Prof. Reinhard Keil-Slawik, Universität Paderborn, November 2001.
- [Bush 1945] Bush, V.: As We May Think. *Atlantic Monthly* 176 (1) 1945, 101–108.
- [Cabri et al. 1999] Cabri, G., Leonardi, L., Zambonelli, F.: Supporting Cooperative WWW Browsing: a Proxy-based Approach. In: *7th Euromicro Workshop on Parallel and Distributed Processing*, Februar 1999, Madeira, Portugal, 138–145.

- [Carlstrom 1992] Carlstrom, E.L.: *Better Living Through Language – The Communicative Implications of a Text-Only Virtual Environment – or – Welcome to LambdaMOO!* Grinnell College, 15. Mai 1992. Verfügbar unter: <http://tecfa.unige.ch/~jermann/communicative.txt>, Stand 1.12.2001.
- [Carroll et al. 2000] Carroll, J.M., Rosson, M.B., Isenhour, P.L., Van Metre, C. A., Schafer, W.A., Ganoë, C.H.: MOOsburg: Supplementing a real community with a virtual community. In: *Proceedings of the Second International Network Conference: INC 2000*, Plymouth, UK: University of Plymouth/ Internet Research, 307–316.
- [Chabert et al. 1998] Chabert, A., Grossman, E., Jackson, L.S., Pietrowiz, S.R., Seguin, C.: Java object-sharing in Habanero. *Communications of the ACM* 41(6) 1998, 69–76.
- [Chen & Sun 1999] Chen, D., Sun, Ch.: A Distributed Algorithm for Graphic Objects Replication in Real-time Group Editors. In: Hayne, S.C. (Hrsg.): *Proceedings of the ACM International Conference on Supporting Group Work (GROUP'99)*, November 14-17, Phoenix, USA. New York: ACM Press 1999, 81–90.
- [Chen et al. 1999] Chen, C., Thomas, L., Cole, J., Chennawasin, C.: Representing the semantics of virtual spaces. *IEEE Multimedia* 6(2) 1999, 54–63.
- [Chodura 2000] Chodura, H.: Audiokommunikation für Telekonferenzen. In: Reichwald, R., Schlichter, J. (Hrsg.): *Verteiltes Arbeiten - Arbeit der Zukunft (D-CSCW 2000)*. Stuttgart: B.G. Teubner 2000, 153–163.
- [Churchill & Bly 1999] Churchill, E.F., Bly, S.: It's all in the words: Supporting work activities with lightweight tools. In: Hayne, S.C. (Hrsg.): *Proceedings of the International ACM SIGGROUP Conference on Supporting Group Work (Group99)*, November 14-17, Phoenix, USA. New York: ACM Press 1999, 40–49.
- [Churchill et al. 2001] Churchill, E.F., Snowdown, D.N., Munro, A.J. (Hrsg): *Collaborative Virtual Environments. Digital Places and Spaces for Interaction*. Berlin: Springer 2001.
- [Cooper 2000] Cooper, W.: MUDs, Metaphysics, and Virtual Reality. *The Journal of Virtual Environments* 5(1) 2000. Verfügbar unter: <http://www.brandeis.edu/pubs/jove/HTML/v5/cooper.htm>, Stand 1.12.2001.
- [Coulouris 1998] Coulouris, G.: Securing Groupware for the Internet, Distributed Systems Research Group Technical Report 752. Originally published in *Information Security Bulletin*, October 1998.
- [Coulouris et al. 2001] Coulouris, G., Dollimore, J., Kindberg, T.: *Distributed Systems, Concepts and Design*. Addison Wesley, Harlow: Pearson Education, 2001.
- [Cox & Campbell 1994] Cox, A., Campbell, M.: Multi-User Dungeons. *Interactive Fantasy 2*, Hogshead Publishing 1994, 15–20.
- [Curtis 1992] Curtis, P.: Mudding: Social phenomena in text-based virtual realities. 1992. *Intertek* 3(3), 26-34 (1-21) Verfügbar unter: <ftp://ftp.lambda.moo.mud.org/pub/MOO/papers>, Stand 1.12.2001.
- [Curtis 1997] Curtis, P.: *LambdaMOO Programmer's Manual*, March 1997. Version 1.8.0p6. Verfügbar unter: <ftp://ftp.lambda.moo.mud.org/pub/MOO/ProgrammersManual.ps>, Stand 1.12.2001.
- [Curtis & Nichols 1994] Curtis, P., Nichols, D.A.: Muds grow up: Social virtual reality in the real world. In: *Proceedings of IEEE Computer Conference '94*, IEEE Computer Society Press 1994, 193–200.
- [Curtis et al. 1995] Curtis, P., Dixon, M., Frederick, R., Nichols, D.A.: The Jupiter Audio/ Video Architecture: Secure Multimedia in Network Places. In: *Proceedings of ACM Multimedia '95*, San Francisco, USA 1995.

- [Damer 1998] Damer, B.: *Avatars! – Exploring and Building virtual worlds on the internet*. Berkeley: Peachpit Press 1998.
- [Dave et al. 1992] Dave, A., Sefika, M., Campbell, R.H.: Proxies, application interfaces, and distributed systems. In: *Proceedings of the Second Workshop on Object Orientation in Operating Systems (IWOOS)*, September 24-25, Dourdan, France. IEEE Press, 1992.
- [Davie 1998] Davie, L.: *Universities, Communities, and Site Building: Exploring Three Online Learning Systems Virtual University, WebCSILE & MOOki*. Verfügbar unter: <http://noisey.oise.utoronto.ca/projcool/conferences/ECOO98.html>, Stand 1.12.2001.
- [Davies et al. 1995] Davies, J., Weeks, R., Revett, M.J.: Communicating information agents for the WWW. In: Proceedings of the 4th International World Wide Web Conference, December 1995, Boston, USA, *World Wide Web Journal* 1(1) 1995, Sebastopol CA: O'Reilly, 473–482.
- [Davies et al. 1998] Davies, J., Stewart, S., Weeks, R.: Knowledge Sharing over the World Wide Web, *BT Technology Journal* 16(3) 1998. 104–109.
- [DeSanctis & Gallupe 1987] DeSanctis, G., Gallupe, B.: A foundation for the study of group decision support systems. *Management Science* 33(5) 1987, 589–609.
- [Dewan 1999] Dewan, P.: Architectures for Collaborative Applications. In: Beaudouin-Lafon, M. (Hrsg.): *Computer Supported Co-operative Work, Trends in Software 7*. Chichester Weinheim: John Wiley & Sons 1999, 169–193.
- [Dommel & Aceves 1997] Dommel H. P., Aceves, J. J.: Floor Control for Multimedia Conferencing and Collaboration, *ACM Multimedia Systems* 5(1) 1997, 23–38.
- [Dourish 1996] Dourish, P.: *Open implementation and flexibility in CSCW toolkits*, PhD thesis, University College London, 1996.
- [Dourish & Bellotti 1992] Dourish, P., Bellotti, V. : Awareness and Coordination in Shared Workspaces. In: Turner, J., Kraut, R. (Hrsg.): *Proceedings of the Conference on Computer-Supported Cooperative Work (CSCW'92)*, October 31 - November 4, Toronto, Canada. New York: ACM Press 1992, 107–114.
- [Dourish et al. 1999a] Dourish, P., Lamping, J., Rodden, T.: Building Bridges: Customisation and Mutual Intelligibility in Shared Category Management. In: Hayne, S.C. (Hrsg.): *Proceedings of the International ACM SIGGROUP Conference on Supporting Group Work (Group99)*, November 14-17, Phoenix, USA. New York: ACM Press 1999, 11–20.
- [Dourish et al. 1999b] Dourish, P., Edwards, K., LaMarca, A., Salisbury, M.: Presto: An Experimental Architecture for Fluid Interactive Document Spaces. *ACM Transactions on Computer-Human Interaction* 6(2) 1999, 133–166.
- [Einhoff et al. 2000] Einhoff, M., Schiffner, N., Greb, A., Rühl, C.: Konzept eines Kommunikations-Framework für die Unterstützung von Virtuellen Unternehmen. In: Reichwald, R., Schlichter, J. (Hrsg.): *Verteiltes Arbeiten - Arbeit der Zukunft (D-CSCW 2000)*. Stuttgart: B.G. Teubner 2000, 23–34.
- [Ellis et al. 1989] Ellis, C. A., Gibbs, S. J.: Concurrency control in groupware systems. In: *Proceedings of the 1989 ACM SIGMOD international conference on Management of data*, May 31 - June 2, 1989, Portland, USA, 399–407.
- [Ellis et al. 1991] Ellis, C.A., Gibbs, S.J., Rein, G.: Groupware: some issues and experiences. *Communications of the ACM* 34(1) 1991, 39–58.
- [Engbring 1995] Engbring, D., Keil-Slawik, R., Selke, H.: *Neue Qualitäten in der Hochschulausbildung. Lehren und Lernen mit interaktiven Medien*. Technischer Bericht Nr. 45 des Heinz Nixdorf Institut, Universität Paderborn 1995.

- [Engelbart & English 1968] Engelbart, D.C., English, W.K.: A Research Center for Augmenting Human Intellect. In: *AFIPS Conference Proceedings of the 1968 Fall Joint Computer Conference*, December 1968, San Francisco, USA, Vol. 33, (AUGMENT, 3954), 395–410.
- [Enklaar 2001] Enklaar, C.: *Konzeption und Entwicklung einer Modulstruktur für die Clientseite des kooperativen Systems sTeam*. Diplomarbeit vorgelegt am Lehrstuhl für Informatik, Fachgruppe Informatik und Gesellschaft, Prof. Reinhard Keil-Slawik, Universität Paderborn, Juli 2001.
- [Erickson et al. 1999] Erickson, T., Smith, D.N., Kellogg, W.A., Laff, M., Richards, J.T., Bradner, E.: Socially Translucent Systems: Social Proxies, Persistent Conversation, and the Design of “BABBLE”. In: Williams, M.G., Altom, M.W., Ehrlich, K., Newman, W. (Hrsg.): *CHI'99 Conference Proceedings*, May 15-20, Pittsburgh, USA. New York: ACM Press 1999, 72–79.
- [Evard 1993] Evard, R.: Collaborative Networked Communication: MUDs as Systems Tools. In: *Proceedings of the Seventh Systems Administration Conference (LISA VII)*, November 1993, Monterrey, USA, 1–8.
- [Fanderclai 1995] Fanderclai, T.L.: MUDs in Education: New Environments, New Pedagogies. *Computer-Mediated Communication Magazine* 2(1) 1995, 8.
- [Fischer 2001] Fischer, C.: *Klassifikation von Forschungsansätzen im Bereich Computer Supported Cooperative Work und ihre praktischen Konsequenzen für die Systemgestaltung*. Diplomarbeit vorgelegt am Lehrstuhl für Informatik, Fachgruppe Informatik und Gesellschaft, Prof. Reinhard Keil-Slawik, Universität Paderborn, 2001.
- [Fischer et al. 2001] Fischer, C., Keil-Slawik, R., Richter, A.: Verhaltensprofile im Internet. In: Keil-Slawik, R. (Hrsg.): *Digitale Medien und gesellschaftliche Entwicklung – Arbeit, Recht und Gemeinschaft der Informationsgesellschaft*. Münster: Waxmann 2001, 65–80.
- [Fuchs 1999] Fuchs, L.: AREA: A Cross-Application Notification Service for Groupware. In: Bødker, S., Kyng, M., Schmidt, K.: *Proceedings of the Sixth European Conference on Computer-Supported Cooperative Work - ECSCW99*, September 12-16, Copenhagen, Denmark. Dordrecht: Kluwer Academic Publishers 1999, 61–80.
- [Fuentes & Troya 2000] Fuentes, L., Troya, J.: Towards an Open Multimedia Service Framework. *ACM Computing Surveys* 32(1es) 2000, Artikel 24, 1–6.
- [Gall & Hauck 1997] Gall, U., Hauck, F.J.: Promondia: A Java-Based Framework for Real-Time Group Communication in the Web. In: *Proceedings of the Sixth International World Wide Web Conference*, 1997. Santa Clara (CA), April 1997. Verfügbar unter: <http://www.scope.gmd.de/info/www6/technical/index.html>, Stand 1.12.2001.
- [Gamma et al. 2000] Gamma, E., Helm, R., Johnson, R., Vlissides, J.: *Design Patterns – Elements of Resuable Object-Oriented Software*. Boston: Addison-Wesley 2000.
- [Gavrila & Barkley 1998] Gavrila, S.I., Barkley, J.F.: Formal Specification for Role Based Access Control User/ Role and Role/ Role Relationship Management. In: *Proceedings of the third ACM workshop on Role-based access control*, October 22 - 23, 1998, Fairfax, USA, 81–90.
- [Gibson 1982] Gibson, J.J.: *Wahrnehmung und Umwelt. Der ökologische Ansatz der visuellen Wahrnehmung*. München: Urban und Schwarzenberg 1982.
- [Graham et al. 1996] Graham, T.C., Urnes, T., Nejabi, R.: Efficient Distributed Implementation of Semi-Replicated Synchronous Groupware. In: *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST'96)*, November 1996, Seattle, USA. New York: ACM Press 1996, 1–10.

- [Greenberg 1990] Greenberg, S.: Sharing Views and Interactions with Single-User Applications. In: *ACM Conference On Office Information Systems*, California, USA, April 1990, 227–237.
- [Greenberg 1991] Greenberg, S.: Personalizable Groupware: Accommodating Individual Roles and Group Differences. In: Bannon, L., Robinson, M., Schmidt, K. (Hrsg.): *Proceedings of the Second European Conference on Computer-Supported Cooperative Work - ECSCW'91*, September 25-27, Amsterdam, The Netherlands. Dordrecht: Kluwer Academic Publishers 1991, 17–31.
- [Greenberg & Roseman 1998] Greenberg S., Roseman, M.: *Using a Room Metaphor to Ease Transitions in Groupware*. Research report 98/611/02, Department of Computer Science, University of Calgary, Calgary (Al.), Canada, January 1998.
- [Greenberg & Roseman 1999] Greenberg, S., Roseman, M.: Groupware Toolkits for Synchronous Work. In: Beaudouin-Lafon, M. (Hrsg.): *Computer Supported Co-operative Work, Trends in Software 7*. Chichester Weinheim: John Wiley & Sons 1999, 135–168.
- [Greenhalgh & Benford 1995] Greenhalgh, C., Benford, S.: MASSIVE: A Collaborative Virtual Environment for Teleconferencing. *ACM Transactions on Computer-Human Interaction* 2(3) 1995, 239–261.
- [Greif 1988] Greif, I.: *Computer Supported Cooperative Work: A Book of Readings*. San Mateo: Morgan Kaufmann Publishers, 1988.
- [Greif & Sarin 1987] Greif, I., Sarin, S.: Data Sharing in Group Work. *ACM Transactions on Office Information Systems* 5(2) 1987, 187–211.
- [Griffiths & Wade 1976] Griffiths, P.P., Wade, B.W.: An Authorization Mechanism for a Relational Database System. *ACM TODS* 1(3) 1976, 242–255.
- [Grimm & Hoff-Holtmanns 1999] Grimm, R., Hoff-Holtmanns, M.: Evaluating a Simple Realization of Combining Audio and Textual Data in Educational Material – Making Sense of Nonsense. In: Collis, B., Oliver, R. (Hrsg.): *Proceedings of ED-MEDIA 99 – World Conference on Educational Multimedia, Hypermedia & Telecommunications*. Charlottesville: Association for the Advancement of Computing in Education 1999, 1390–1391.
- [Gross & Prinz 2000] Gross, T., Prinz, W.: Gruppenwahrnehmung im Kontext. In: Reichwald, R., Schlichter, J. (Hrsg.): *Verteiltes Arbeiten - Arbeit der Zukunft (D-CSCW 2000)*. Stuttgart: B.G. Teubner 2000, 115–126.
- [Guernsey 1996] Guernsey, L.: College "MOOs" Foster Creativity and Collaboration Among Users. *The Chronicle of Higher Education*, Section: Information Technology, February 9, 1996, A24. Verfügbar unter: <http://www.bvu.edu/ctown/CHE.html>, Stand 1.12.2001.
- [Gutwin et al. 1996] Gutwin, C., Greenberg, S., Roseman, M.: Supporting Workspace Awareness in Groupware. (CSCW-Video-Program) In: Ackerman, M.S. (Hrsg.): *Proceedings of the ACM 1996 Conference on Computer Supported Cooperative Work (CSCW'96)*, November 16-20, Boston, USA. New York: ACM Press 1996, 8.
- [Haake 1999] Haake, M.: Facilitating Orientation in Shared Hypermedia Workspaces. In: Hayne, S.C. (Hrsg.): *Proceedings of the International ACM SIGGROUP Conference on Supporting Group Work (Group99)*, November 14-17, Phoenix, USA. New York: ACM Press 1999, 365–374.
- [Hagel & Armstrong 1997] Hagel, J., Armstrong, A.G.: *Net Gain: Expanding Markets Through Virtual Communities*. Harvard Business School Press, 1997.

- [Hampel 1999] Hampel, T.: sTEAM – Cooperation and Structuring Information in a Team. In: de Bra, P., Leggett, J. (Hrsg.): *Proceedings of WebNet 99 – World Conference on the WWW and Internet*. Charlottesville: Association for the Advancement of Computing in Education 1999, 469–474.
- [Hampel 2001] Hampel, T.: TRES FACIUNT COLLEGIUM – Paderborn’s Collaboration Centred Approach for New Forms of Learning. In: Price, J., Willis, D., Davis, N., Willis, J. (Hrsg.): *Proceedings of SITE 2001*, March 5-10, 2001, Orlando, USA, 52–57.
- [Hampel & Bopp 2001] Hampel, T., Bopp, T.: Magellan, the Paderborn Approach to Distributed Knowledge Organization. In: Montgomerie, C., Viteli, J. (Hrsg.): *Proceedings of ED-MEDIA 2001*, Charlottesville (Va.): Association for the Advancement of Computing in Education 2001, 649–655.
- [Hampel & Keil-Slawik 1998a] Hampel, T., Keil-Slawik, R.: Vom Holzmodell zur interaktiven Animation und Visualisierung. In: *SUGinfo* (2) 1998, 20–25.
- [Hampel & Keil-Slawik 1998b] Hampel, T., Keil-Slawik, R.: Interaktive Animationen und Visualisierungen, eine neue Qualität und „Spielweise“ für die Mechanik – Ergebnisse der interdisziplinären Projektgruppe ChiLis. In: Cap, C.H. (Hrsg.): *JIT’98 Java Informationstage* (JIT), Berlin: Springer 1998, 155–163.
- [Hampel & Keil-Slawik 2001a] Hampel, T., Keil-Slawik, R.: sTeam – Designing an Integrative Infrastructure for Web-Based Computer Supported Cooperative Learning. In: *Proceedings of the 10th International World Wide Web Conference*, May 1-5, 2001, Hong Kong, 76–85.
- [Hampel & Keil-Slawik 2001b] Hampel, T., Keil-Slawik, R.: Learning in Virtual Communities – The Paderborn Approach to Cooperative Learning. In: Ruokamo, H., Nykänen, O., Pohjolainen, S., Hietala, P. (Hrsg.): *Proceedings of the Tenth International PEG Conference Intelligent Computer and Communication Technology – Learning in Online Communities*, June 23-26, 2001, Tampere, Finland, 177–183.
- [Hampel & Keil-Slawik 2001c] Hampel, T., Keil-Slawik, R.: ^{open}sTeam – Ein Open Source-Projekt zur kooperativen Strukturierung von Informationen im Team. *DFN Mitteilungen* (55) 2001, 4–6.
- [Hampel & Keil-Slawik 2001c] Hampel, T., Keil-Slawik, R.: Novel Client Representations for the Collaborative Virtual Learning Environment sTeam. In: Montgomerie, C., Viteli, J. (Hrsg.): *Proceedings of ED-MEDIA 2001*, Charlottesville (Va.): Association for the Advancement of Computing in Education 2001, 1787–1788.
- [Hampel & Keil-Slawik 2002] Hampel, T., Keil-Slawik, R.: sTeam: Structuring Information in a Team - Distributed Knowledge Management in Cooperative Learning Environments. erscheint in: *ACM Journal of Educational Resources in Computing* 1(2) 2002.
- [Hampel & Nowaczyk 1999] Hampel, T., Nowaczyk, O.: Explorationen für die Mechanikausbildung – eine neue Dimension interaktiver Lehrmaterialien. *44th International Scientific Colloquium, 3. Workshop Multimedia für Bildung und Wirtschaft*. Technische Universität Ilmenau 1999, 101–103.
- [Hampel & Selke 1999] Hampel, T., Selke, H.: Customizing the Web – Two Tools for individual and collaborative use of hypermedia course material. In: Collis, B., Oliver, R.: *Proceedings of ED-MEDIA 99*. Charlottesville: Association for the Advancement of Computing in Education 1999, 634–639.
- [Hampel et al. 1998a] Hampel, T., Keil-Slawik, R., Ferber, F.: mechANIma teach – A New Approach to the Teaching of Mechanics. In: *Proceedings of ED-MEDIA 98*. Charlottesville (Va.), USA. Association for the Advancement of Computing in Education (I) 1998, 458–465.

- [Hampel et al. 1998b] Hampel, T., Keil-Slawik, R., Ferber, F., Müller, W.: Hypermedia-teaching of mechanics- mechANIma. In: Davies, G., O'Eigeartaigh, M.: *3rd Annual Conference on Integrating Technology into Computer Science Education –ITiCSE '98*, Dublin, Ireland, August 18-21, 1998. New York: ACM Press 1998, 112–116.
- [Hampel et al. 1998c] Hampel, T., Ferber, F., Müller, W.: mechANIma teach - A New Approach and Educational Philosophy to the Teaching of Mechanics. In: Davies, G. (Hrsg.): *Proceedings of the XV. IFIP World Computer Congress*, August 31-September 4, 1998, Vienna, Austria and Budapest, Hungary, 393–402.
- [Hampel et al. 1999a] Hampel, T., Keil-Slawik, R., Claassen, B.G., Plohmann, F., Reimann C.: Pragmatic solutions for better integration of the visually impaired in virtual communities. In: Hayne, S.C. (Hrsg.): *Proceedings of the ACM International Conference on Supporting Group Work (GROUP'99)*, November 14-17, Phoenix (Ar.), USA. New York: ACM Press 1999, 258–266.
- [Hampel et al. 1999b] Hampel, T., Keil-Slawik, R., Ferber, F.: Explorations -A New Form of Highly Interactive Learning Materials. In: de Bra, P., Leggett, J. (Hrsg.): *Proceedings of WebNet 99 – World Conference on the WWW and Internet*. Charlottesville: Association for the Advancement of Computing in Education 1999, 463–468.
- [Hampel et al. 1999c] Hampel, T., Magenheimer, J., Schulte, C.: Dekonstruktion von Informatiksystemen als Unterrichtsmethode. Zugang zu objektorientierten Sichtweisen. In: Schwill, A. (Hrsg.): *Informatik und Schule. Fachspezifische und fachübergreifende didaktische Konzepte*. Berlin Heidelberg New York u.a.: Springer 1999, 149–164.
- [Hampel et al. 2001] Hampel, T., Keil-Slawik, R., Nowaczyk, O., Selke, H.: »Ein Schulmeister muss singen können« – Die drei Säulen der Paderborner DISCO. *Wirtschaftsinformatik* 43(1), 2001, 69–76.
- [Harrison & Dourish 1996] Harrison, S., Dourish, P.: Re-Place-ing Space: The Roles of Place and Space in Collaborative Systems. In: Ackerman, M.S. (Hrsg.): *Proceedings of the ACM 1996 Conference on Computer Supported Cooperative Work (CSCW'96)*, November 16-20, Boston, USA. New York: ACM Press 1996, 67–76.
- [Haynes & Holmevik 1998] Haynes, C., Holmevik, J.R.: *High Wired: On the Design, Use, and Theory of Educational Moos*. University of Michigan Press, February 1998.
- [Heinemann 2000] Heinemann, L.: *Analyse, Entwurf und prototypenhafte Umsetzung eines Sicherheitskonzepts für kooperationsunterstützende Systeme*. Diplomarbeit vorgelegt am Lehrstuhl für Informatik, Fachgruppe Informatik und Gesellschaft, Prof. Reinhard Keil-Slawik, Universität Paderborn, Oktober 2000.
- [Henderson & Card 1985] Henderson, A.J., Card, S.A. Rooms: The Use of Multiple Virtual Workspaces to Reduce Space Contention. In: *ACM Transactions on Graphics* 5(3) 1985, 211–243.
- [Henderson & Kyng 1991] Henderson, A., Kyng, M.: There's No Place Like Home: Continuing Design in Use. In: Greenbaum, J., Kyng, M. (Hrsg.): *In Design at Work: Cooperative Design of Computer Systems*. Hillsdale, NJ: Lawrence Erlbaum Associates 1991, 219–240.
- [Herlocker et al. 2000] Herlocker, J., Konstan, J., Riedl, J.: Explaining Collaborative Filtering Recommendations. *ACM Conference on Computer Supported Cooperative Work*, December 2-6. New York: ACM Press 2000, 241–250.
- [Hesse et al. 1997] Hesse, F.W., Garsoffky, B., Hron, A.: Interface-Design für computerunterstütztes kooperatives Lernen. In: Issing, L.J., Klimsa, P. (Hrsg.): *Information und Lernen mit Multimedia*. Weinheim: Psychologie Verlags Union 1997, 253–267.

- [Hill & Slator 2000] Hill, C., Slator, B.M.: Computer Science Instruction in a Virtual World. In: Bourdeau, J., Heller, R. (Hrsg.): *Proceedings of the World Conference on Educational Multimedia, Hypermedia and Telecommunications ED-MEDIA 2000*, June 26 - July 1, 2000, Montreal, Canada, Charlottesville: Association for the Advancement of Computing Education 2000, 406–411.
- [Hill et al. 1994] Hill, R.D., Brinck, T., Rohall, S.L., Patterson, J.F., Wilner, W.: The Rendezvous architecture and language for constructing multiuser applications. *ACM Transactions on Computer-Human Interaction* 1 (2) 1994, 81–125.
- [Ho 2000] Ho, T.: *Analyse und Bewertung synchroner Computer Supported Cooperative Learning (CSCL) – Systeme für die universitäre Lehre*. Diplomarbeit vorgelegt am Lehrstuhl für Informatik, Fachgruppe Informatik und Gesellschaft, Prof. Reinhard Keil-Slawik, Universität Paderborn, Mai 2000.
- [Holmevik & Haynes 1999] Holmevik, J.R., Haynes, C.: *MOOniversity: A Student's Guide to Online Learning Environments*. Boston: Allyn & Bacon 1999.
- [Holst 2000] Holst, S.: Evaluation of Collaborative Virtual Learning Environments: The State of the Art. In: Scheuermann, F. (Hrsg.): *Campus 2000: Lernen in neuen Organisationsformen. Proceedings of GMW 2000. Fachtagung der Gesellschaft für Medien in der Wissenschaft*, September 19-21, 2000, Innsbruck. Medien in der Wissenschaft, Bd. 10. Münster: Waxmann 2000, 199–212.
- [Hübinette 2001] Hübinette, F.: *Programming, using and understanding Pike*. Verfügbar unter: <http://david.hedbor.org/pikeman/docs/tutorial.html>, Stand 1.12.2001.
- [Hunt 1996] Hunt, R.A.: Affordances and Constraints of Electronic Discussions. *13th Inkshed Working Conference*, Hecla Island, Manitoba, May 1996.
- [Huxor 1998] Huxor, A.: An Active Worlds interface to BSCW. In: Snowdon, D., Churchill E. (Hrsg.): *Proceedings of Collaborative Virtual Environments (CVE '98)*, June 17-19, Manchester, UK, 1998, 87–93.
- [Johansen 1988] Johansen, R.: *Groupware: Computer Support for Business Teams*. New York: The Free Press – Macmillan 1988.
- [Johnson 1992] Johnson, Ph.: Supporting Exploratory CSCW with the ENGRET Framework. In: Turner, J., Kraut, R. (Hrsg.): *Proceedings of the ACM Conference on Computer-Supported Cooperative Work (CSCW'92)*, October 31 - November 4, Toronto, Canada. New York: ACM Press 1992, 298–305.
- [Johnson & Johnson 1990] Johnson, D.W., Johnson R.T.: Cooperative Learning and Research. In: Shlomo, S. (Hrsg.): *Cooperative learning theory and research*. New York: Preager 1990, 23–37.
- [Johnson et al. 1990] Johnson, D.W., Johnson, R.T., Stanne, M.B., Garibaldi, A.: Impact of group processing on achievement in cooperative groups. *Journal of Social Psychology* 130, 1990, 507–516.
- [Kawell et al. 1988] Kawell, L., Beckhardt, S., Halvorsen, T., Ozzie, R., Greif, I.: Replicated Document Management in a Group Communication System. *Proceedings of the 2nd International Conference on Computer-Supported Cooperative Work (CSCW'88)*, September 26-28, 1988, Portland (Or.),USA. SIGCHI/SIGOIS ACM 1988, reprinted in: Marca, D., Bock, G. (Hrsg.): *Groupware: Software for Computer-Supported Cooperative Work*. IEEE Computer Society Press 1992, 226–235.
- [Keil-Slawik 1990] Keil-Slawik, R.: *Konstruktives Design. Ein ökologischer Ansatz zur Gestaltung interaktiver Systeme* (Habilitationsschrift). Forschungsbericht des Fachbereichs Informatik, Nr. 90-14, Technische Universität Berlin, 1990.

- [Keil-Slawik 1991] Keil-Slawik, R.: Symbolic Representations and the Creation of Meaning in Systems Development. In: Ivanov, K. (Hrsg.): *Proceedings of the 14th IRIS*. Revised Papers of the 14th Information Systems Research Seminar, 11-14 August, 1991, Umeå-Lövånger, Scandinavia. Research Reports in Information Processing And Computer Science No. 16, University of Umeå 1991.
- [Keil-Slawik 1997] Keil-Slawik, R.: Multimedia und Gesellschaft. In: Ministerium für Wirtschaft und Mittelstand, Technologie und Verkehr des Landes Nordrhein-Westfalen (Hrsg.): *Multimedia-Forschung, Grundlagen der Arbeit und Projekte der TaskForce Multimedia-Forschung im Rahmen der Landesinitiative media NRW*. Schriftenreihe der Landesinitiative media NRW, Bd. 5. Düsseldorf: LV Druck im Landwirtschaftsverlag GmbH 1997, 48–58.
- [Keil-Slawik 1998a] Keil-Slawik, R.: Multimedia als Steinbruch des Lernens. In: Hauff, M. (Hrsg.): *media@uni-multi.media? Entwicklung – Gestaltung – Evaluation neuer Medien*. Münster: Waxmann 1998, 81–99.
- [Keil-Slawik 1998b] Keil-Slawik, R.: Audio – Video – DISCO: Hören, Sehen und Denken in der Digitalen Infrastruktur für computergestütztes kooperatives Lernen. In: Fisch, E., Vollmer, H. (Hrsg.): *Einblicke – Ausblicke: 25 Jahre Universität Gesamthochschule Paderborn*, Paderborn 1998, 94–102.
- [Keil-Slawik 1999] Keil-Slawik, R.: Evaluation als evolutionäre Systemgestaltung. Aufbau und Weiterentwicklung der Paderborner DISCO (Digitale Infrastruktur für computerunterstütztes kooperatives Lernen). In: Kindt, Michael (Hrsg.): *Projektelevaluation in der Lehre – Multimedia an Hochschulen zeigt Profil(e)*. Reihe: Medien in der Wissenschaft, Bd. 7. Münster: Waxmann 1999, 11–36.
- [Keil-Slawik 2000a] Keil-Slawik, R.: Zwischen Vision und Alltagspraxis: Anmerkungen zur Konstruktion und Nutzung typographischer Maschinen. In: Voß, G.G., Holly, W., Boehnke, K. (Hrsg.): *Neue Medien im Alltag: Begriffsbestimmungen eines interdisziplinären Forschungsfeldes*. Opladen: Leske & Budrich 2000, 199–220.
- [Keil-Slawik 2000b] Keil-Slawik, R.: Gestaltung und Nutzung alltagstauglicher Infrastrukturen. In: Beck, U., Sommer, W. (Hrsg.): *LEARNTEC 2000. 8. Europäischer Kongress und Fachmesse für Bildungs- und Informationstechnologie*. Tagungsband, Karlsruhe 2000, 1–8.
- [Keil-Slawik 2001] Keil-Slawik, R.: Neues Lernen und Neue Medien. – Anmerkungen zu einem kulturell unbewältigten Verhältnis. In: *Qualifikations-Entwicklungs-Management (QUEM) report, Arbeiten und Lernen – Lernkultur Kompetenzentwicklung und Innovative Arbeitsgestaltung*, Schriften zur beruflichen Weiterbildung, Heft 67, Arbeitsgemeinschaft Betriebliche Weiterbildungsforschung e. V. Berlin 2001, 109–118.
- [Keil-Slawik & Hampel 2001] Keil-Slawik, R., Hampel, T.: Virtuelle Gemeinschaften: Freizeitspaß oder globaler Marktfaktor? In: Keil-Slawik, R. (Hrsg.): *DIGITALE MEDIEN und gesellschaftliche Entwicklung. – Arbeit, Recht und Gemeinschaft in der Informationsgesellschaft*. Münster: Waxmann Verlag 2001, 245–271.
- [Keil-Slawik & Roth 1999] Keil-Slawik, R., Roth, W.: Designorientierung und Designpraxis – Entwicklung und Einsatz von konstruktiven Gestaltungskriterien In: Arend, U., Eberleh, E., Pitschke, K. (Hrsg.): *Software-Ergonomie '99 – Design von Informationswelten*. Stuttgart Leipzig: B.G. Teubner 1999, 43–52.
- [Keil-Slawik & Selke 1995] Keil-Slawik, R., Selke, H.: Learning Hypertexts: Supporting Active and Collaborative Learning. In: H. Maurer (Hrsg.): *Educational Multimedia and Hypermedia, Proceedings of ED-MEDIA 95*. Charlottesville: Association for the Advancement of Computing in Education 1995, 767.

- [Keil-Slawik & Selke 1998a] Keil-Slawik, R., Selke, H.: Forschungsstand und Forschungsperspektiven zum virtuellen Lernen von Erwachsenen. In: Arbeitsgemeinschaft Qualifikations-Entwicklungs-Management Berlin (Hrsg.): *Kompetenzentwicklung '98 – Forschungsstand und Forschungsperspektiven*. Münster: Waxmann 1998, 165–208.
- [Keil-Slawik & Selke 1998b] Keil-Slawik, R., Selke, H.: Mythen und Alltagspraxis von Technik und Lernen. *Informatik Forum* 12(1) 1998, 9–17.
- [Keil-Slawik et al. 1997] Keil-Slawik, R., Beuschel, W., Gaiser, B., Klemme, M., Pieper, C., Selke H.: Multimedia-Projekte an deutschen Hochschulen. Ein Überblick. In: Hamm, I., Müller-Böling, D. (Hrsg.): *Hochschulentwicklung durch neue Medien: Erfahrungen - Projekte - Perspektiven; mit einer Bestandsaufnahme über Multimedia-Projekte an deutschen Hochschulen*. Gütersloh: Verlag Bertelsmann Stiftung 1997, 199–259.
- [Kelly & Rheingold 1993] Kelly, K., Rheingold, H.: The Dragon Ate My Homework. *Wired* 1(3) 1993, 68–73.
- [Kleinen 1999] Kleinen, B.: Ein Werkzeug zur Moderationsunterstützung. In: Arend, U., Eberleh, E. Pitschke, K. (Hrsg.): *Software-Ergonomie '99*. Stuttgart Leipzig: B.G. Teubner 1999, 195–204.
- [Kleinen et al. 2000] Kleinen, B., Wittstock, M., Herczeg, M.: Inkrementelle Verfeinerung von Kooperationsstrukturen in der Druckindustrie. In: R. Reichwald, J. Schlichter (Hrsg.): *Verteiltes Arbeiten - Arbeit der Zukunft (D-CSCW 2000)*. Stuttgart: B.G. Teubner 2000, 178–191.
- [Klemme et al. 1998] Klemme, M., Kuhnert, R., Selke, H.: Semantic Spaces, In: Höök, K., Munro, A., Benyon, D. (Hrsg.): *Workshop on Personalised and Social Navigation in Information Space*, SICS Technical Report T98:02, Kista, Sweden, 1998, 109–118.
- [Konstan et al. 1997] Konstan, J., Miller, B., Maltz, D., Herlocker, J., Gordon, L., Riedl, J.: GroupLens: Applying Collaborative Filtering to Usenet News. *Communications of the ACM* 40(3) 1997, 77–87.
- [Krasner & Pope 1988] Krasner, G.E., Pope, S.T.: A cookbook for using the Model-View-Controller interface paradigm. *Journal of Object-Oriented Programming* 1(3) 1988, 26–49.
- [Lampson 1974] Lampson, B.: Protection. In: *Proceedings of the 5th Princeton Conference on Information Sciences and Systems*, Princeton, 1971. Reprinted in *ACM Operating Systems* 8(1) 1974, 18–24.
- [Lampson et al. 1992] Lampson, B., Abadi, M., Burrows, M., Wobber, E.: Authentication in Distributed Systems: Theory and Practice. *ACM Transactions in Computer Systems* 10(4) 1992, 265–310.
- [Lantz 1990] Lantz, K.L.: Collaboration awareness in support of collaboration transparency: requirements for the next generation of shared window systems. In: *Proceedings of the CHI'90*, Seattle. New York: ACM Press 1990, 663–671.
- [Lebling et al. 1979] Lebling, P. D., Blank, M. S., Anderson, T. A.: *Zork: A Computerized Fantasy Simulation Game*. IEEE Computers Magazine, April 1979, 51–59.
- [Leland et al. 1988] Leland, M.D.P., Fish, R.S., Kraut, R.E.: Collaborative Document Production Using Quilt. In: *Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW'88)*, September 26-28, Portland, (Or.), USA. New York: ACM Press 1988, 206–215.
- [Leroi-Gourhan 1988] Leroi-Gourhan, A.: *Hand und Wort. Die Evolution von Technik, Sprache und Kunst*. Frankfurt am Main: Suhrkamp 1988.

- [Li & Muntz 1998] Li, D., Muntz, R.: COCA: Collaborative Objects Coordination Architecture. In: Poltrock, S., Grudin, J. (Hrsg.): *Proceedings of ACM Conference on Computer Supported Cooperative Work (CSCW'98)*, November 14-18, Seattle, USA. New York: ACM Press 1998, 179–196.
- [Li et al. 1999] Li, W., Wang, W., Marsic, I.: Collaboration transparency in the DISCIPLE framework, In: Hayne, S.C. (Hrsg.): *Proceedings of the ACM International Conference on Supporting Group Work (GROUP'99)*, November 14-17, Phoenix, USA. New York: ACM Press 1999, 326–335.
- [MacLean et al. 1990] MacLean, A., Carter, K., Lövsstrand, L., Moran, T.: User-tailorable systems: pressing the issues with buttons. In: Chew, J.C., Whiteside, J. (Hrsg.): *Conference on Human Factors and Computing Systems. Conference proceedings on Empowering people: Human factors in computing system: special issue of the SIGCHI Bulletin*, April 1-5, 1990, Seattle, USA. New York: ACM Press 1990, 175–182.
- [Mambrey et al. 2000] Mambrey, P., Pipek, V., Won, M., Wulf, V.: Kommunikation und Kooperation in Knowledge Communities. Abstrakt zum gleichlautenden Workshop im Rahmen der D-CSCW 2000, München. In: Reichwald, R., Schlichter, J. (Hrsg.): *Verteiltes Arbeiten - Arbeit der Zukunft (D-CSCW 2000)*. Stuttgart: B.G. Teubner 2000, 255–256.
- [Mansfield et al. 1999] Mansfield, T., Kaplan, S., Fitzpatrick, G., Phelps, T., Fitzpatrick, M., Taylor R.: Toward Locales: Supporting collaboration with Orbit. *Journal of Information and Software Technology* 41(6) 1999, 367–382.
- [Marais & Bharat 1997] Marais, J., Bharat, K.: Supporting cooperative and personal surfing with a desktop assistant. In: *Proceedings of the 10th Annual ACM Symposium on User Interface Software and Technology*, October 1997, 129–138.
- [Masinter & Ostrom 1993] Masinter, L., Ostrom, E.: Collaborative Information Retrieval: Gopher from MOO. In: *Proceedings of INET 1993*. Verfügbar unter: <http://mirrors.ccs.neu.edu/MOO/papers/MOOGopher.html>, Stand 1.12.2001.
- [McCanne et al. 1997] McCanne, S., Brewer, E., Katz, R., Rowe, L., Amir, E., Chawathe, Y., Coopersmith, A., Mayer-Patel, K., Raman, S., Schuett, A., Simpson, D., Swan, A., Tung, T., Wu, D., Smith, B.: Towards a Common Infrastructure for Multimedia- Networking Middleware. In: *Proceedings of the 7th International Workshop on Network and Operating Systems for Digital Audio and Video (NOSSDAV'97)*, May 1997, 39–49.
- [McKinley et al. 1999] McKinley, P.K., Malenfant, A.M., Arango, J.M.: Pavilion: A Middleware Framework for Collaborative Web-based Applications. In: Hayne, S.C. (Hrsg.): *Proceedings of the ACM International Conference on Supporting Group Work (GROUP'99)*, November 14-17, Phoenix, USA. New York: ACM Press 1999, 179–188.
- [Miao & Haake 2001] Miao, Y., Haake, J.M.: Supporting Problem Based Learning by a Collaborative Virtual Environment: A Cooperative Hypermedia Approach. In: Sprague, R.H. (Hrsg.): *Proceedings of the 34th Annual Hawaii International Conference on System Sciences (HICSS-34)*, January 3-6, 2001, Maui, Hawaii, USA, auf CD-ROM veröffentlicht, 2001, 1–11.
- [Michelis 2000] Michelis, G.D.: Designing for Communities. In: Reichwald, R., Schlichter, J. (Hrsg.): *Verteiltes Arbeiten - Arbeit der Zukunft (D-CSCW 2000)*. Stuttgart: B.G. Teubner 2000, 13–19.
- [Mohan 1999] Mohan, C.: A database perspective on Lotus Domino/ Notes, International Conference on Management of Data and Symposium on Principles of Database Systems. In: *Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data*, May 31-June 3, 1999, Philadelphia, USA, 507.

- [Mørch et al. 1998] Mørch, A., Stiernerlieng, O., Wulf, V.: Tailorable Groupware. In: *Issues, Methods, and Architectures*. Report of a Workshop held at GROUP'97, November 16, 1997, Phoenix, USA. SIGCHI Bulletin 30(2) 1998.
- [Moreau & Gray 1998] Moreau L., Gray N.: A Community of Agents Maintaining Link Integrity in the World Wide Web. In: Nwana, H.S., Ndumu, D.T. (Hrsg.): *PAAM98 - The Third International Conference and Exhibition on The Practical Application of Intelligent Agents and Multi-Agents*, London, UK, 1998, 221–235.
- [Mühlhäuser 1995] Mühlhäuser, M.: *Cooperative Computer-Aided Authoring and Learning. A Systems Approach*. Dordrecht: Kluwer Publishers 1995.
- [Multicast 2000] *mcast2000 white paper – A Survey of the History of Internet Multicast – A Technology Backgrounder*, Stardust Technologies. Inc., 4th Annual Summit on scalable infrastructure and accelerated content delivery, January 2000. Verfügbar unter: <http://www.stardust.com/mcast2000/whitepaperfinal.PDF>, Stand 1.12.2001.
- [Mynatt et al. 1997] Mynatt, E.D., Adler, A., Ito, M., O'Day, V.L.: Design for network communities. In: *Conference Proceedings on Human factors in Computing Systems*, March 22-27, 1997, Atlanta, USA, New York: ACM Press 1997, 210–217.
- [Nardi 1996] Nardi, B.A.: *Context and Consciousness: Activity Theory and Human-Computer Interaction*. Massachusetts Institute of Technology, 1996.
- [Nelson 1974] Nelson, T.H.: *Dream Machines: new freedoms through computer screens - A minority report*. Computer Lib: You can and must understand computers now. Chicago: Hugo's Book Service 1974 (Nachdruck Microsoft Press 1988).
- [Nelson 1980] Nelson, T.H.: Replacing the Printed Word: A Complete Literary System. In: Lavington, S.H. (Hrsg.): *Information Processing 80*. Amsterdam: Publishing Company, 1980, 1013–1023.
- [Neuwirth et al. 1990] Neuwirth, C.M., Kaufer, D.S., Chandhok, R., Morris, J.H.: Issues in the Design of Computer Support for Co-authoring and Commenting. In: *Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW'90)*, October 1990, Los Angeles, USA. New York: ACM Press 1990, 183–195.
- [Nodder et al. 1999] Nodder, Ch., Williams, G., Dubrow, D.: Evaluating the usability of an evolving collaborative product – changes in user type, tasks and evaluation methods over time. In: Hayne, S.C. (Hrsg.): *Proceedings of the ACM International Conference on Supporting Group Work (GROUP'99)*, November 14-17, Phoenix, USA. New York: ACM Press 1999, 150–159.
- [Nomura et al. 1998] Nomura, T., Hayashi, K., Hazama, T., Gudmundson, S.: Interlocus: workspace configuration mechanisms for activity awareness. In: Poltrock, S., Grudin, J. (Hrsg.): *Proceedings of ACM Conference on Computer Supported Cooperative Work (CSCW'98)*, November 14-18, Seattle, USA. New York: ACM Press 1998, 19–28.
- [O'Day et al. 1996] O'Day, V.L., Bobrow, D.G., Shirley, M.: The Social-Technical Design Circle. In: Ackerman, M.S. (Hrsg.): *Proceedings of the ACM 1996 Conference on Computer Supported Cooperative Work (CSCW'96)*, November 16-20, Boston, USA. New York: ACM Press 1996, 160–169.
- [Patterson et al. 1990] Patterson, J.F., Hill, R.D., Rohall, S.L., Meeks, S.W.: Rendezvous: an architecture for synchronous multiuser applications. In: *Proceedings of the Conference on Computer-Supported Cooperative work*, October 7-10, 1990, Los Angeles, USA, New York: ACM Press 1990, 317–328.
- [Pfister & Wessner 2000a] Pfister, H.R., Wessner, M.: Evaluation von CSCL-Umgebungen. In: Krahn, H., Wedekind, J. (Hrsg.): *Virtueller Campus '99. Heute Experiment - morgen Alltag?*, Medien in der Wissenschaft, Bd. 9. Münster: Waxmann 2000, 139–149.

- [Pfister & Wessner 2000b] Pfister, H.R., Wessner, M.: Verteilt und doch gemeinsam: Computergestütztes Lernen im Team. Sankt Augustin: *DER GMD-SPIEGEL* 1/2 2000.
- [Pfister et al. 1998] Pfister, H.R., Wessner, M., Beck-Wilson, J., Miao, Y., Steinmetz, R.: Rooms, protocols, and nets: Metaphors for computer supported cooperative learning of distributed groups. In: Bruckman, A.S., Guzdial, M., Kolodner, J.L., Ram, A. (Hrsg.): *Proceedings of ICLS 98, International Conference of the Learning*, December 16-19, Georgia Institute of Technology, Atlanta, USA. Charlottesville: Association for the Advancement of Computing in Education 1998, 242–248.
- [Plötzner et al. 2000] Plötzner, R., Bodemer, D., Hoppe, H.U., Tewissen, F.: Kooperatives Problemlösen auf der Grundlage gemeinsamer Visualisierungen. In: Reichwald, R., Schlichter, J. (Hrsg.): *Verteiltes Arbeiten - Arbeit der Zukunft (D-CSCW 2000)*. Stuttgart: B.G. Teubner 2000, 91–102 .
- [Powers 1997] Powers, M.: *How to Program a Virtual Community – Attract new web visitors and get them to stay!* New York: Ziff-Davis Press 1997.
- [Prakash & Shim 1994] Prakash, A., Shim, H.S.: DistView: Support for Building Efficient Collaborative Applications using Replicated Objects. In: Furuta, R., Neuwirth, Ch. (Hrsg.): *Proceedings of the ACM Conference on Computer-Supported Cooperative Work (CSCW'94)*, October 22 - 26, Chapel Hill, USA. New York: ACM Press 1994, 153–164.
- [Preiß 2000] Preiß, M.: *Integration synchroner Kommunikationsmechanismen in computergestützten Lernumgebungen (Entwicklung in der Programmiersprache JAVA)*. Diplomarbeit vorgelegt am Lehrstuhl für Informatik, Fachgruppe Informatik und Gesellschaft, Prof. Reinhard Keil-Slawik, Universität Paderborn, Juli 2000.
- [Prinz 1999] Prinz, W.: NESSIE: An Awareness Environment for Cooperative Settings. In: Bødker, S., Kyng, M., Schmidt, K.: *Proceedings of the Sixth European Conference on Computer-Supported Cooperative Work - ECSCW99*, September 12-16, Copenhagen, Denmark. Dordrecht: Kluwer Academic Publishers 1999, 391–410.
- [Prinz & Gräther 2000] Prinz, W., Gräther, W.: Das Social Web Cockpit: Ein Assistent für virtuelle Gemeinschaften. In: Reichwald, R., Schlichter, J. (Hrsg.): *Verteiltes Arbeiten - Arbeit der Zukunft (D-CSCW 2000)*. Stuttgart: B.G. Teubner 2000, 127–138.
- [Probst et al. 1999] Probst, G., Raub, S., Romhardt, K.: *Wissensmanagement*. Wiesbaden: Gabler 1999.
- [Raymond 2001] Raymond, E.S.: *The Cathedral and the Bazaar*. Verfügbar unter <http://www.tuxedo.org/~esr/writings/cathedral-bazaar/>, Stand 1.12.2001.
- [Reid 1994] Reid, E.: *Cultural Formations in Text-Based Virtual Realities*. PhD thesis, University of Melbourne, January 1994.
- [REP-FIT-1999-64] *BSCW Basic Support for Cooperative Work Version 3.3 Handbuch*. GMD Report Nr. 64, Oktober 1999.
- [Rheingold 1993] Rheingold, H.: *The Virtual Community: Homesteading on the Electronic Frontier*. Reading: Addison-Wesley 1993.
- [Robinson 1992] Robinson, M.: Its not a bloody typewriter you know. In: *Proceedings of 2nd Oksnøen Symposium, Customizing software systems*, May 23-28, Oksnøen, Norway, 1992.
- [Roseman & Greenberg 1992] Roseman, M., Greenberg, S.: GroupKit: A groupware toolkit for building real-time conferencing applications. In: Turner, J., Kraut, R. (Hrsg.): *Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW'92)*, November 1-4, Toronto, Canada. New York: ACM Press 1992, 43–50.

- [Roseman & Greenberg 1993] Roseman, M., Greenberg, S.: Building flexible groupware through open protocols. In: *ACM Conference On Office Information Systems*, October 1993, California, USA. New York: ACM Press 1993, 279–288.
- [Roseman & Greenberg 1996a] Roseman, M., Greenberg, S. TeamRooms: Network Places for Collaboration. *Proceedings of the ACM 1996 Conference on Computer Supported Cooperative Work*, November 16-20, 1996, Boston, USA, New York: ACM Press 1996, 325–333.
- [Roseman & Greenberg 1996b] Roseman, M. and Greenberg, S.: Building Real Time Groupware with GroupKit, A Groupware Toolkit. *ACM Transactions on Computer Human Interaction* 3(1) 1996, 66–106.
- [Roth & Unger 1998] Roth, J., Unger, C.: “Dream Team” – a platform for synchronous collaborative applications. In: Herrmann, T., Just-Hahn, K. (Hrsg.): *Groupware und organisatorische Innovation (D-CSCL '98)*. Stuttgart: B.G. Teubner 1998, 153–165.
- [Roth & Unger 2000] Roth, J., Unger C.: Developing synchronous collaborative applications with TeamComponents. In: Dieng, R. et al. (Hrsg.): *Designing Cooperative Systems, Proceedings of the 5th International Conference on the Design of Cooperative Systems (COOP'2000)*. Amsterdam: IOS Press 2000, 353–368.
- [Sandor et al. 1997] Sandor, O., Bogdan, C., Bowers, J.: Aether: An Awareness Engine for CSCW. In: Hughes, J.A., Prinz, W., Rodden, T., Schmid, K. (Hrsg.): *Proceedings of the Fifth European Conference on Computer-Supported Cooperative Work - ECSCW97*, September 7-11, Lancaster, UK. Dordrecht: Kluwer Academic Publishers 1997, 221–236.
- [Satyanarayanan 1989] Satyanarayanan, M.: Integrating Security in a Large Distributed System. *ACM Transactions on Computer Systems* 7(3) 1989, 247–280.
- [Schenk & Schwabe 2000] Schenk, B., Schwabe, G.: Auf dem Weg zu einer Groupware-Didaktik – Erfahrungen und Konzepte aus den Schulungen in Cuparla. In: Reichwald, R., Schlichter, J. (Hrsg.): *Verteiltes Arbeiten - Arbeit der Zukunft (D-CSCW 2000)*. Stuttgart: B.G. Teubner 2000, 63–76.
- [Schiffner & Chodura 2000] Schiffner, N., Chodura, H.: Mediazine – A Combination of Television, Radio, WWW, Telecommunication and 3D Computer sound and graphics. In: Churchill, E., Reddy, M. (Hrsg.): *Proceedings of the Third International Conference on Collaborative Virtual Environments (CVE2000)*. September 10-12, San Francisco, USA. New York: ACM Press 2000, 147–154.
- [Schmidt 1998] Schmidt, D.C.: Evaluating Architectures for Multithreaded Object Request Brokers. *Communications of the ACM* 44(10) 1998, 54–60.
- [Schuckmann et al. 1996] Schuckmann, C., Kirchner, L., Schümmer, J., Haake, J.M.: Designing Object-Oriented synchronous groupware with COAST. In: Ackerman, M.S. (Hrsg.): *Proceedings of the ACM 1996 Conference on Computer Supported Cooperative Work (CSCW'96)*, November 16-20, Boston, USA. New York: ACM Press 1996, 30–38.
- [Schulmeister 2001] Schulmeister, R.: *Virtuelle Universität Virtuelles Lernen*. München Wien: Oldenbourg Wissenschaftsverlag 2001.
- [Schulte et al. 1999] Schulte, C., Selke, H., Huth, C.: Kooperative Arbeitsplattformen – CSCW-Systeme (BSCW, Hyperwave und Lotus Notes) in Lehr- und Lernkontexten. *LOG IN* 19(3/4) 1999, 42–55.
- [Shen & Dewan 1992] Shen, H., Dewan, P.: Access Control for Collaborative Environments. In: Turner, J., Kraut, R. (Hrsg.): *Proceedings of the Conference on Computer-Supported Cooperative Work – CSCW92*, October 31 - November 4, Toronto, Canada. New York: ACM Press 1992, 51–58.

- [Shiozawa et al. 1999] Shiozawa, H., Okada, K., Matsushita, Y.: Perspective Layered Visualization of Collaborative Workspaces. In: Hayne, S.C. (Hrsg.): *Proceedings of the ACM International Conference on Supporting Group Work (GROUP'99)*, November 14-17, Phoenix, USA. New York: ACM Press 1999, 71–80.
- [Shirmohammadi et al. 1998] Shirmohammadi, S., Oliveira, J.C., Georganas, N.D.: Applet-Based Multimedia Telecollaboration: A Network-Centric Approach. *IEEE Multimedia Magazine* 5(2) 1998, 64–73.
- [Sikkel 1997] Sikkel, K.: *A Group-based Authorization Model for Computer-Supported Cooperative Work*. GMD – Forschungszentrum, Arbeitspapier der GMD 1055, März 1997.
- [Simone & Bandini 1997] Simone, C., Bandini, S.: Compositional Features for Promoting Awareness within and across Cooperative Applications. In: Hayne, S.C., Prinz, W.: *Proceedings of the International ACM SIGGROUP Conference on Supporting Group Work: The Integration Challenge - Group97*, November 16-19, Phoenix, USA. New York: ACM Press 1997, 358–367.
- [Sinemus 2000] Sinemus, M.: *Konzeption und prototypische Entwicklung eines Clients zur Kooperationsunterstützung in computergestützten Lernumgebungen*, Diplomarbeit vorgelegt am Lehrstuhl für Informatik, Fachgruppe Informatik und Gesellschaft, Prof. Reinhard Keil-Slawik, Universität Paderborn, März 2000.
- [Slator & Hill 1999] Slator, B.M., Hill, C.: Mixing Media For Distance Learning: Using Ivn And Moo In Comp372. In: Collis, B., Oliver, R.: *Proceedings of ED-MEDIA 99*. Charlottesville: Association for the Advancement of Computing in Education 1999, 881–886.
- [Sohlenkamp & Chwelos 1994] Sohlenkamp, M., Chwelos, G.: Integrating Communication, Cooperation, and Awareness: The DIVA Virtual Office Environment. In: Furuta, R., Neuwirth, Ch. (Hrsg.): *Proceedings of the ACM Conference on Computer-Supported Cooperative Work (CSCW'94)*, October 22 - 26, Chapel Hill, USA. New York: ACM Press 1994, 51–58.
- [Sohlenkamp et al. 1998] Sohlenkamp, M., Prinz, W., Fuchs, L.: PoliAwaC – Design und Evaluation des PoliTeam Awareness-Client. In: *Tagungsband der D-CSCW'98*, Fachtagung des German Chapter of the ACM und der Gesellschaft für Informatik und dem Fachgebiet „Informatik und Gesellschaft“, Fachbereich Informatik, Universität Dortmund, 28.- 30. September 1998, Dortmund, 181–194.
- [Spellman et al. 1997] Spellman, P.J., Mosier, J.N., Deus, L.M., Carlson, J.A.: Collaborative Virtual Workspace. In: Hayne, S.C., Prinz, W. (Hrsg.): *Proceedings of the International ACM SIGGROUP Conference on Supporting Group Work (Group97)*, November 16-19. Phoenix, USA. New York: ACM Press 1997, 197–203.
- [Spence 2001] Spence, R.: *Information Visualization*. Harlow London New York: ACM Press Books, Addison-Wesley 2001.
- [Stefik et al. 1987] Stefik, M., Foster, G., Bobrow, D.G., Kahn, K., Lanning, S., Suchman, S.: Beyond the chalkboard: computer support for collaboration and problem solving in meetings. *Communications of the ACM* 30(1) 1987, 32–47.
- [Steinfeld et al. 1999] Steinfeld, Ch., Jang, Ch.-Y., Pfaff, B.: Supporting Virtual Team Collaboration: The TeamSCOPE System. In: Hayne, S.C. (Hrsg.): *Proceedings of the ACM International Conference on Supporting Group Work (GROUP'99)*, November 14-17, Phoenix, USA. New York: ACM Press 1999, 81–90.
- [Stiemerling 1999] Stiemerling, O., Hinken, R., Cremers, A.B.: The EVOLVE Tailoring Platform: Supporting the Evolution of Component-Based Groupware. In: *Proceedings of EDOC'99*, September 27.-30. Mannheim: IEEE Press 1999, 106–115.

- [Stiemerling & Wulf 1998] Stiemerling, O., Wulf, V.: Beyond “Yes and No”- Extending Access Control in Groupware with Awareness and Negotiation. In: Darses, F., Zaraté, P.P. (Hrsg.): *Proceedings of the Third International Conference on the Design of Cooperative Systems (COOP'98)*, Vol. I, 26.-29.5.1998, Cannes, France, INRIA, 1998, 111–120.
- [Tappe 2000] Tappe, T.: *Konzeption und Entwicklung von Annotationsmechanismen zur Kooperationsunterstützung im Internet und Intranet auf der Basis von XML und Java*. Diplomarbeit vorgelegt am Lehrstuhl für Informatik, Fachgruppe Informatik und Gesellschaft, Prof. Reinhard Keil-Slawik, Universität Paderborn, November 2000.
- [ter Hofte 1998] Hofte, G.H.: *Working Apart Together – Foundations for Component Groupware*, Telematica Instituut Fundamental Research Series, Vol. 1, Enschede, the Netherlands, 1998.
- [Teufel et al. 1995] Teufel, S., Sauter, C., Mühlherr, T., Bauknecht, K.: *Computerunterstützung für die Gruppenarbeit*. Bonn: Addison-Wesley 1995.
- [Theng et al. 1996] Theng, Y.L., Thimbleby, H., Jones, M.: 'Lost in hyperspace': Psychological problem or bad design? *AP-CHI'96*, Singapore 1996, New York: ACM-Press 1996, 387–396.
- [Thomas & Peasley 1998] Thomas, S.L., Peasley, A.E.: *Lotus Notes Certification : Application Development and System Administration*. New York: Computing McGraw-Hill 1998.
- [Tietze & Steinmetz 2000] Tietze, D.A., Steinmetz, R.: Ein Framework zur Entwicklung komponentenbasierter Groupware. In: Reichwald, R., Schlichter, J. (Hrsg.): *Verteiltes Arbeiten - Arbeit der Zukunft (D-CSCW 2000)*. Stuttgart: B.G. Teubner 2000, 49–62.
- [Tollmar et al. 1996] Tollmar, K., Sandor, O., Schömer, A.: Supporting Social Awareness @ Work Design and Experience. In: Ackerman, M.S. (Hrsg.): *Proceedings of the ACM 1996 Conference on Computer Supported Cooperative Work (CSCW'96)*, November 16-20, Boston, USA. New York: ACM Press 1996. 298–307.
- [Trevor & Koch 1997] Trevor, J.; Koch, T.: MetaWeb: Bringing synchronous Groupware to the World Wide Web. In: Hughes, J. A.; Prinz, W.; Rodden, T.; Schmidt, K. (Hrsg.): *Proceedings of the ECSCW '97, Fifth European Conference on Computer Supported Cooperative Work*, September 7-11, Lancaster, UK. Dordrecht: Kluwer Academic Publishers 1997, 65–69.
- [Trevor et al. 1994] Trevor, J., Rodden, T., Mariani, J.: The Use of Adapters to Support Cooperative Sharing. In: Furuta, R., Neuwirth, Ch. (Hrsg.): *Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW'94)*, October 22 – 26, Chapel Hill, USA. New York: ACM Press 1994, 219–230.
- [Turkle 1995] Turkle, S.: *Life on the Screen: Identity in the Age of the Internet*. New York: Simon & Schuster 1995.
- [Turkle 1998] Turkle, S.: All MOOs are Educational—the Experience of “Walking through the Self”. In: Haynes, C., Holmevik, J.R.: *High Wired: On the Design, Use, and Theory of Educational Moos*. University of Michigan Press 1998, ix–xix.
- [Voss et al. 1999] Voss, A., Nakata, K., Juhnke, M.: Concept Indexing. In: Hayne, S.C. (Hrsg.): *Proceedings of the International ACM SIGGROUP Conference on Supporting Group Work (Group99)*, November 14-17, Phoenix, USA. New York: ACM Press 1999, 1–10.
- [Wan 1993] Wan, D.: *CLARE: A Computer-Supported Collaborative Learning Environment Based on the Thematic Structure of Research and Learning Artefacts*. Department of Information and Computer Sciences, University of Hawaii, Honolulu, Hawaii 96822, PhD Thesis Proposal, Number ICS-TR-93-01, January 1993.

- [Wenger 1998] Wenger, E.: Communities of Practice: Learning as a Social System. *The Systems Thinker* 9(5) 1998.
- [Wessner & Pfister 2001] Wessner, M., Pfister, H.R.: Kooperatives Lehren und Lernen. In: Schwabe, Unland, Streitz (Hrsg.): *CSCW Kompendium – Lehr- und Handbuch für das computergestützte kooperative Arbeiten*. Berlin: Springer 2001, 251–263.
- [Wessner et al. 1998] Wessner, M., Beck-Wilson, J., Pfister, H.R.: Clear - A Cooperative Distributed Learning Environment. In: Ottmann, T., Tomek, I.: *Proceedings of ED-MEDIA / ED-TELECOM 98*. Freiburg, Germany, Charlottesville: Association for the Advancement of Computing in Education 1998 (Vol.II), 1876–1877.
- [Wessner et al. 1999a] Wessner, M., Pfister, H.R., Miao, Y.: Using Learning Protocols to Structure Computer-Supported Cooperative Learning. In: Collis, B., Oliver, R.: *Proceedings of ED-MEDIA 99*, World Conference on Educational Multimedia, Hypermedia & Telecommunications, June 19-24, 1999, Seattle, Washington, Charlottesville: Association for the Advancement of Computing in Education 1999, 471–476.
- [Wessner et al. 1999b] Wessner, M., Pfister, H.R., Miao, Y.: Umgebungen für computerunterstütztes kooperatives Lernen in der Schule. In: Schwill, A. (Hrsg.): *Informatik und Schule. Fachspezifische und fachübergreifende didaktische Konzepte*. 8. GI-Fachtagung Informatik und Schule INFOS99, 22.-25. September, Potsdam. Reihe: Informatik aktuell, Berlin: Springer 1999, 86–93.
- [Wexelblat 1999] Wexelblat, A.: History-Based Tools for Navigation, *IEEE's 32nd Hawaii' International Conference on System Sciences (HICSS'99)*, January 5-8, Hawaii, USA. IEEE Computer Society Press 1999.
- [Wiesner 2001] Wiesner, T.: *Entwicklung eines Prototypen zum Erstellen und Erschließen von Pfaden (Trails) im WWW*. Diplomarbeit vorgelegt am Lehrstuhl für Informatik, Fachgruppe Informatik und Gesellschaft, Prof. Reinhard Keil-Slawik, Universität Paderborn, Februar 2001.
- [Williams 1987] Williams G.: HyperCard. In: *Byte* 12(14) 1987, 109–117.
- [Wulf 1999] Wulf, V.: Let's see your Search-Tool! – Collaborative Use of Tailored Artefacts in Groupware. In: Hayne, S.C. (Hrsg.): *Proceedings of the International ACM SIGGROUP Conference on Supporting Group Work (Group99)*, November 14-17, Phoenix, USA. New York: ACM Press 1999, 50–59.
- [Yamazaki & Herder 2000] Yamazaki, Y., Herder, J.: Exploring Spatial Audio Conferencing Functionality in Multiuser Virtual Environments. In: Churchill, E., Reddy, M. (Hrsg.): *Proceedings of the Third International Conference on Collaborative Virtual Environments (CVE2000)*, September 10-12, San Francisco, USA. New York: ACM Press 2000, 207–208.
- [Zhang et al. 1999] Zhang, Z., Haffner, E., Heuer, A., Engel, T., Meinel, Ch.: Role-based Access Control in Online Authoring and Publishing Systems vs. Document Hierarchy. In: *Proceedings on the Seventeenth ACM Annual International Conference on Computer Documentation*, September 12-14, 1999, New Orleans, USA, 193–198.