



Fakultät für Elektrotechnik, Informatik und Mathematik
Institut für Informatik

Mobilität in der kooperativen Wissensarbeit

*Entwicklung einer Musterarchitektur für
mobil-verteilte Wissensräume*

Dissertation

Schriftliche Arbeit zur Erlangung
des akademischen Grades
Doktor der Naturwissenschaften

Bernd Eßmann

Paderborn
März 2006

Inhaltsverzeichnis

1 Einleitung	1
2 Szenarien	11
2.1 Präsenzkooperation	14
2.1.1 Lerngruppe	15
2.1.2 Spontane Kooperation	26
2.2 Kooperation mit räumlich verteilten Teilnehmern	34
2.2.1 Räumlich getrennte Kooperation	36
2.2.2 Distant Learning	41
2.3 Herausforderungen mobil-verteilter Kooperation	50
3 Anforderungen	53
3.1 Phasen mobil-spontaner Kooperation	55
3.2 Medienfunktionen und Mobilität	57
3.3 Neue Qualitäten mobiler Kollaboration	60
3.3.1 Vernetzung und Sichtbarkeit	61
3.3.2 Kontextualisierung	63
3.3.3 Konsistenz und Reversibilität	66
3.4 Neue Rahmenbedingungen	73
3.4.1 Kommunikation	75
3.4.2 Koordination	80
3.4.3 Kooperation	83
3.5 Zusammenfassung	85
4 Lösungsansätze für eine technische Umsetzung	87
4.1 Knotenaufbau – Microkernel	88
4.2 Vernetzung und Sichtbarkeit	90
4.2.1 Protokolle zur Vernetzung mobiler Knoten	92
4.2.2 Vernetzung verteilter Dienste	98
4.2.3 Netzwerktopologie	99
4.3 Persistenz	100
4.3.1 Verteilte Persistenzsysteme	101
4.3.2 Verteilte Persistenz als gemeinsamer Speicher	116
4.3.3 Konsistenz in verteilten Persistenzsystemen	118
4.4 Ereigniskontrolle	120

5 Musterarchitektur mobil-verteilter Wissensräume	129
5.1 Replikationsstrategien für die mobil-spontane Kooperation	130
5.2 Gruppen-Tuple Spaces	137
5.3 Einbettung externer Dienste	143
5.4 Gesamtarchitektur mobil-verteilter Wissensräume	145
6 Schluss und Ausblick	155
Literatur	177
Stichwortverzeichnis	179

Abbildungsverzeichnis

2.1	Der Prozess einer mobil-spontanen Kooperation	13
3.1	Anforderungen mobilitätsunterstützender Kooperationsumgebungen	55
3.2	Der Kollaborationsprozess als Regelkreis	56
3.3	Schlüsselfaktoren mobil-verteilter Wissensräume	61
3.4	Verknüpfung von realem Raum und virtuellem Wissensraum	65
3.5	Kollaboration – Kommunikation, Koordinierung und Kooperation .	75
4.1	In die Schaffung mobil-verteilter Wissensräume involvierte Forschungsbereiche	88
4.2	Mobil-verteilter Wissensraum: Handlungs- und Umsetzungsebene . .	89
4.3	Auf DHTs basierende Overlay-Netzwerk-Protokolle und deren Anwendungen	105
4.4	Abbildung eines physikalischen Netzwerkes in einen logischen Adressraum	106
4.5	Verwaltungstabellen und Routinginformationen eines Pastry-Knotens	108
4.6	Auswirkung der Lokalität auf das Routing in DHTs	109
4.7	Initialisierung der Verwaltungstabellen beim Anmelden eines neuen Knotens in ein Pastry-Netzwerk	110
4.8	Pastrys Selbstorganisation im Fall einer Netzwerksegmentierung . .	110
4.9	Das Replikationsverfahren von PAST	113
4.10	Trade-off verteilter Persistenzkonzepte	115
4.11	Abonnieren eines bereits veröffentlichten Themas in Scribe	123
4.12	Abbestellen eines abonnierten Themas in Scribe	124
4.13	Gegenüberstellung von Single-Multicast und Multiple-Unicast	125
5.1	Offline-Verfügbarkeit in einer verteilten Persistenzschicht	130
5.2	Schematische Übersicht des Versionierungsmechanismus	132
5.3	Der Hypercube als Distributionstopologie	135
5.4	Distributionstopologie basierend auf einem Sparse Graphen	137
5.5	Replikation des Gruppen-Tuple Space zwischen Independent Nodes .	141
5.6	Virtuelle Wissenräume zweier Kooperationsgruppen mit den zugeordneten Gruppen-Tuple Spaces	142
5.7	Exemplarische Architektur einer mobil-spontanen Kooperationsumgebung	147

Mobilität in der kooperativen Wissensarbeit

1 Einleitung

Mobilität bestimmt in der heutigen Welt wesentliche Faktoren im Leben der Menschen. Mobile Geräte und Infrastrukturen sind zu einem festen Bestandteil des Alltagslebens geworden. Der naheliegende Wunsch, die neuen technischen Möglichkeiten für eine kooperative Wissensstrukturierung in mobilen Alltagssituationen zu nutzen, bleibt dennoch bis heute unerfüllt. Bei der Suche nach den Gründen für das Fehlen einer durchgängigen Unterstützung mobiler Kooperationsprozesse hilft eine Betrachtung der Entwicklungsstränge kooperativer Wissensstrukturierung und mobilen Arbeitens.

Ausgehend von einer zunächst isolierten Dokumentenverarbeitung und späteren Workflow-Systemen haben sich in der heutigen Arbeitswelt *Groupware*- und *Computer Supported Cooperative Work (CSCW)-Systeme* etabliert, die ein kooperatives Bearbeiten von Dokumenten erlauben. Aus einer technischen Perspektive spiegelt sich diese Entwicklung in fest vernetzten Kommunikations- und Kooperationsstrukturen wider. In diesen Strukturen nehmen dedizierte Server eine zentrale Rolle für die Koordination der Kooperationsprozesse und die persistente Speicherung der gemeinsam genutzten Dokumente ein.

Parallel zu diesen zentralisierten Strukturen begleiten mobile Computer ihre Besitzer in allen Lebenslagen und führen so zu neuen Formen der Mobilität in computergestützten Arbeitsprozessen. Der Arbeitsplatz wird nicht länger von dem Standort des Computers bestimmt, sondern kann an jedem Aufenthaltsort des Benutzers sein. Die Integration funkbasierter Netzwerkschnittstellen in mobile Computer erhöht zudem die Unabhängigkeit der Benutzer von Netzwerk- und Telefonanschlüssen und ermöglicht – zumindest theoretisch – einen flexiblen Zugang zu etablierten Netzwerkstrukturen und eine direkte Vernetzung mobiler Geräte untereinander.

Die Möglichkeiten der spontanen Vernetzung der Mobilgeräte über *mobile Ad-Hoc-Netzwerke* enthält ein hohes Potential für eine direkte und unabhängige computergestützte Interaktion der mobilen Benutzer. Dennoch bleiben die Nutzer in ihrer Mobilität oft isoliert und arbeiten auf einem persönlichen Dokumentenfundus. Diese Isolation wird oft lediglich temporär durch einen sporadischen Austausch von Dokumenten unterbrochen. Es fehlt die langfristige Integration einer kooperativen Dokumentenbearbeitung. Daher findet auch eine computergestützte Kooperation im mobilen Umfeld kaum statt.

Lediglich in Reichweite von Zugangspunkten zu klassischen Netzwerkinfrastrukturen verbinden sich die Benutzer mit zentralen Kooperationsdiensten und nutzen diese wie bei einem stationären Rechner. Trotz der Möglichkeit einer Ad-Hoc-Vernetzung ist eine spontane Kooperation zwischen Benutzern in einer mobilen Umgebung ohne feste Kooperationsinfrastrukturen äußerst selten. Das Brachliegen die-

ser nahe liegenden Verwendung mobiler Computer für spontane Kooperationssitzungen wirft Fragen nach den Gründen auf.

Die Realisierung einer mobilitätsfreundlichen Kooperationsumgebung ist aber ohne ein Wissen um die Nutzungsszenarien und die strukturellen Grenzen einer technischen Realisierung nicht möglich. Diese Arbeit untersucht daher anhand des Wechselspiels zwischen den Nutzungsszenarien die Probleme, die sich aus einer Zusammenführung der Techniken aus dem Feld der mobilen Vernetzung und des CSCW ergeben. Mangels einer durchgehenden Persistenz in mobilen Netzwerken ist dabei unklar wie eine längerfristige Kooperationen in einem mobilen Umfeld zustande kommen soll.

Um diese fehlende Konvergenz beider Entwicklungen zu beschreiben werden im Rahmen dieser Arbeit Nutzungsszenarien entwickelt, die eine Verknüpfung von mobilen und spontanen Netzwerken mit kooperationsunterstützenden Infrastrukturen betrachten. Derzeit bekannte Szenarien bewegen sich zumeist isoliert in einer der beschriebenen Welten. Klassische CSCW-Szenarien beschränken sich häufig auf die Zusammenarbeit in etablierten Infrastrukturen, während sich Szenarien der mobilen Vernetzung auf die Aspekte mobilitätsfreundlicher Netzwerke konzentrieren. Nutzungsszenarien für eine Verknüpfung von computergestützter Kooperation und spontaner Vernetzung sind hingegen bis heute relativ unerforscht.

Ein wesentliches Ziel der entwickelten Nutzungsszenarien ist die Ermittlung der spezifischen *funktionalen Elemente*, die essentiell für eine Unterstützung mobiler Kooperationsprozesse sind. Diese funktionalen Anforderungen hängen von den technischen Elementen der angestrebten Kooperationsumgebung ab. Da sich die technischen Elemente ihrerseits an den funktionalen Anforderungen orientieren, bedingen sich beide Ebenen der Anforderungsbestimmung gegenseitig. Im Rahmen dieser Arbeit werden zur Unterscheidung dieser Anforderungsebenen die Begriffe *Handlungsebene* für die funktionalen Anforderungen und *Umsetzungsebene* für die technischen Anforderungen geprägt.

Die hier entwickelten Nutzungsszenarien helfen zunächst die Anforderungen auf der Handlungsebene zu ermittelten. Ein wichtiges Resultat ist der freie und spontane Charakter der Kooperation infolge der Unabhängigkeit von festen Infrastrukturen. Die sporadischen und spontanen Verbindungen zu den Kooperationspartnern ähneln dabei stark einer natürlichen Zusammenarbeit in Situationen des täglichen Lebens. In diesen treffen sich Personen bewusst oder auch zufällig und beginnen eine kooperative Handlung aus dem momentanen Kontext heraus.

Die Spontanität der kooperativen Handlung und die hohe Bedeutung des aktuellen Kontextes der Kooperationspartner werden in dieser Arbeit als eine der wesentlichen Herausforderungen an die technischen wie funktionalen Komponenten der zukünftigen mobilen Kooperationsumgebung identifiziert. Insbesondere die funktionalen Elemente der Kooperationsunterstützung müssen dem freien und flexiblen Charakter der Zusammenarbeit gerecht werden. Daher setzt der hier entworfene Ansatz auf das langjährig erprobte Konzept der *virtuellen Wissensräume*¹ mit den zugehörigen

¹Vgl. [Hampel, 2001]

primären Medienfunktionen als eine Verkörperung freier dokumentenzentrierter Kooperation. Die virtuellen Wissensräume bieten die wesentlichen Mechanismen für ein freies Arbeiten und stellen über persistente Objekte auch eine längerfristige Gruppenarbeit sicher. Diese Nachhaltigkeit können die häufig verfolgten kommunikativen Lösungsansätze nicht leisten. Daher wird in dieser Arbeit der Ansatz einer Überführung des Konzeptes der virtuellen Wissensräume in eine mobile Netzwerkumgebung verfolgt und unter dem Begriff der *mobil-verteilten Wissensräume* zusammengefasst.

Im Feld derzeit verfügbarer Lösungsansätze aus Forschung und Industrie zeigen sich zwei unterschiedliche Herangehensweisen an die Vernetzung mobiler Computer: Erstere erweitert existierende feste Netzwerkinfrastrukturen mit flächendeckenden Zugangsmöglichkeiten für Mobilgeräte; Zweitere schafft neue Protokolle für eine spontane Vernetzung mobiler Computer in selbstorganisierenden Netzwerken. Zukünftig dürfte sich eine heterogene Netzwerkinfrastruktur aus von Dienstanbietern bereitgestellten festen Infrastrukturen und untereinander spontan vernetzten Mobilgeräten ergeben. Übergänge zwischen diesen Strukturen besitzen eine hohe Dynamik und machen eine Vorhersage über deren Verfügbarkeit schwierig. Die Überführung der virtuellen Wissensräume in die Welt der mobilen Vernetzung führt zu bisher nicht betrachteten Problemstellungen, die im Rahmen dieser Arbeit herausgearbeitet werden.

Betrachtet man die Ad-Hoc-Vernetzung der beteiligten Geräte als Basis einer mobil-spontanen Kooperation, lässt diese, im Gegensatz zu den herkömmlichen Ansätzen des CSCW, keine Server-zentrierten Architekturen mehr zu. Die Speicherung der Inhalte und die Verwaltung der Kooperationsumgebung kann nicht länger ein dedizierter Server übernehmen, da von dessen Verfügbarkeit die Funktionsfähigkeit der gesamten Kooperationsumgebung abhängt. Als alternative Architektur verfolgt diese Arbeit eine *verteilte Verwaltung und Speicherung der Kooperationsinhalte* auf den beteiligten Knoten. Diese Vorgehensweise ermöglicht eine Gewähr der Verfügbarkeit der Kooperationsumgebung auch in mobilen Nutzungsszenarien. Bei Ausfall eines Knotens können dessen Aufgaben automatisch auf die verbliebenen Knoten transferiert werden. Jedoch gestalten sich innerhalb dynamischer Netzwerkstrukturen die persistente Speicherung von Kooperationsobjekten und der nahtlose Zugriff auf diese durch alle Kooperationspartner äußerst schwierig.

Eine wesentliche Problemstellungen ist daher die *persistente Speicherung von Kooperationsobjekten*, die gerade in dokumentenzentrierten Kooperationskonzepten, wie dem der virtuellen Wissensräume, von zentraler Bedeutung ist. Daher orientieren sich die in dieser Arbeit entwickelten Ansätze an einer Bereitstellung persistenter virtueller Wissensräume in mobil-spontanen Kooperationsszenarien. Die Frage einer *verteilten und persistenten Speicherung des gemeinsamen Wissensraums* beinhaltet auch die *Vermeidung von Medienbrüchen² infolge von Verbindungsabbrüchen zur eigenen Kooperationsgruppe* und deren Ressourcen. Daher stellt diese Arbeit die Forderung auf, Kooperationsobjekte auch dann erreichen zu können, wenn gerade

²Vgl. [Keil-Slawik und Selke, 1998]

keine Verbindung zu den Kooperationspartnern und Kooperationsinfrastrukturen besteht. Diese Forderung wird im Folgenden als *Offline-Verfügbarkeit* definiert.

Um die Verfügbarkeit der für die Kooperation benötigten Dokumente auch bei Abwesenheit einiger Knoten des Netzwerkes garantieren zu können, wird im Rahmen des entwickelten Lösungsansatzes im Wesentlichen die Strategie einer *Distribution* und *Replikation* der Kooperationsobjekte verfolgt. Während die Distribution Objekte auf mehreren Knoten verteilt speichert und dadurch verhindert, dass der gesamte Datenraum von lokalen Ausfällen betroffen ist, speichert die Replikation Objekte redundant auf mehrere Knoten und erhöht so deren generelle Verfügbarkeit. Die Persistenz und die Verfügbarkeit der Kooperationsobjekte hängen somit stark von den Distributions- und Replikationsstrategien der verteilten Kooperationsumgebung ab. Sie bestimmen auch die Eignung der Kooperationsumgebung für einen Einsatz im mobilen Nutzungsumfeld.

Derzeitige Ansätze für eine Persistenz in verteilten Systemen, wie z. B. die *Distributed Hashtables (DHT)*, verteilen und replizieren Datenobjekte scheinbar zufällig auf die Knoten des Netzwerkes. Eine nähere Betrachtung im Rahmen dieser Arbeit ergibt, dass sie daher eine gesteuerte Verfügbarkeit von Objekten auf einzelnen Knoten nicht ohne weiteres gewährleisten können. Insbesondere fehlt oftmals die Möglichkeit, Objekte zwischen den replizierenden Knoten zuverlässig abzulegen, wenn beteiligte Knoten zeitweise nicht im Netzwerk verfügbar sind.

Ziel dieser Arbeit ist eine Musterarchitektur, die geeignete Komponenten der Forschungsfelder der computergestützten Kooperation und der spontanen Vernetzung verknüpft und um fehlende Elemente ergänzt, um so eine Unterstützung der entwickelten Nutzungsszenarien zu ermöglichen. Dies geschieht sowohl ausgehend von existierenden CSCW-Strukturen als auch von verfügbaren Techniken spontaner Vernetzung. Der Einsatz verfügbarer und erprobter Technologien ermöglicht schon in frühen Phasen des Entwurfs eine alltagspraktische Lösung zu erlangen. Daher werden klassische Konzepte der Gruppenarbeit mit neuen Möglichkeiten der mobilen Ad-Hoc-Netzwerke so zusammengebracht, dass eine durchgängige Nutzung über diesen Bereich entsteht und Medienbrüche verringert werden, die einer langfristigen Nutzung im Wege stehen.

Dazu müssen im Rahmen dieser Arbeit Probleme der verteilten Speicherung und Verwaltung der Kooperationsumgebung gelöst werden. Diese Fragestellungen werden ausgehend von den entwickelten Nutzungsszenarien angegangen und bestimmen so den Aufbau der Arbeit. Zuerst werden daher die angesprochenen Nutzungsszenarien erarbeitet, ohne die eine Eingrenzung der funktionellen und technischen Anforderungen nicht möglich ist. Um die Lücke zwischen CSCW-Systemen und mobilen Netzwerken zu schließen, werden anhand der dort ermittelten Anforderungen die benötigten Komponenten für eine Musterarchitektur mobil-verteilter Wissensräume identifiziert. Diese Komponenten werden in einem nächsten Schritt durch eigens entwickelte Strategien ergänzt. Erst diese Strategien erlauben eine Verknüpfung der Komponenten zu der gesuchten Musterarchitektur. Zu diesem Zweck wird im Rahmen der vorliegenden Arbeit eine an die Gegebenheiten der mobil-verteilten Koope-

ration angepasste mobilitätsfreundliche Distributions- und Replikationsstrategie für eine verteilten Persistenz entworfen.

Mangels vergleichbarer Infrastrukturen werden nach der Einleitung in Kapitel 2 Nutzungsszenarien der mobilen Kooperation in verteilten Softwareumgebungen entworfen und für eine Bestimmung der zentralen Anforderung an mobil-verteilte Kooperationssysteme genutzt. Die dort entworfenen Szenarien orientieren sich sowohl an der scheinbar „natürlichen“ computergestützten Zusammenarbeit in mobilen Lebenssituationen, als auch an der räumlich getrennten Kooperation in klassischen CSCW-Systemen. Die Szenarien erlauben dabei eine Beschreibung von Kooperationskonstellationen, die auf technischer Ebene zurzeit noch nicht definierbar sind. Diese sind Mischungen aus Präsenzszenarien und Szenarien räumlich getrennter Kooperation. Besonders für solche Mischszenarien existieren bis dato keine adäquaten Vorschläge für deren Gestaltung. Ihnen wird in dieser Arbeit daher ein besonderer Stellenwert eingeräumt.

Grundlage der Nutzungsszenarien ist stets eine Kooperation auf Basis eines gemeinsamen Fundus von Dokumenten. Die in den Nutzungsszenarien für eine computergestützten Kooperation verwendeten Werkzeuge entstammen heutigen CSCW-Systemen und werden in einen mobilen Kontext transferiert. Da die Arbeitswelt im Sinne klassischer Workflows stärker formalisiert ist und die Lernwelt einen eher konstruktivistischen Ansatz mit offener Zusammenarbeit darstellt, entstammen die entwickelten Nutzungsszenarien zu je einer Hälfte der Welt des Lernens und des Arbeitens. Innerhalb der entwickelten Szenarien wird von der ständigen Verfügbarkeit der gemeinsam genutzten Dokumente für alle Teilnehmer ausgegangen, auch wenn diese nicht über ein Netzwerk in Verbindung stehen.

Bei der Betrachtung der Szenarien stellt sich heraus, dass diese sich über den gesamten Kooperationsprozess in drei Phasen gliedern, die jeweils sehr unterschiedliche Aktivitäten enthalten. Die erste Phase ist die der *Gründung*, in der die Teilnehmer die Kooperation bewusst oder unbewusst beginnen. In der Phase der *Kooperation* erfolgt die eigentliche Zusammenarbeit basierend auf den gemeinsam genutzten Dokumenten. Die letzte Phase ist die der *Auflösung*. Sie wird in klassischen CSCW-Szenarien kaum betrachtet. Ihr kommt im mobilen Nutzungsumfeld jedoch eine besondere Bedeutung zu.

Die Phase der *Auflösung* entspringt der im Rahmen dieser Arbeit vorgenommenen Betrachtung der Dynamik in spontanen Kooperationsprozessen. Aufgrund der Mobilität der Teilnehmer darf eine starke Fluktuation innerhalb der Gruppenstruktur angenommen werden. Jedes Verlassen der Kooperation kommt für Benutzer im mobilen Umfeld einer Auflösung derselben gleich, da ein Wiederkehren in die Kooperation mit der gleichen Teilnehmerkonstellation aufgrund der Mobilität aller Teilnehmer nicht vorausgesagt werden kann. Oftmals wird die Kooperation in einer anderen Nutzungskonstellation weitergeführt, so dass die Auflösung auch als eine Art Übergang begriffen werden kann. Analog dazu wird das Wiedereintreten in die Kooperation als *Gründung* mit einer anderen Teilnehmerkonstellation betrachtet.

Trotz der Mobilität der Nutzer tragen die entwickelten Szenarien dem Umstand Rechnung, dass auch eine mobile Kooperation an bestimmte Orte gebunden sein

kann oder aus einem bestimmten Ortskontext heraus entsteht. Dies macht den Nutzungskontext zu einem wichtigen Aspekt dieser Szenarien. Mit steigender Mobilität gewinnt dieser an Bedeutung. Die Nutzbarmachung des Nutzungskontextes ist daher eine wesentliche Fragestellung für die im Rahmen dieser Arbeit entworfene Musterarchitektur einer mobilitätsunterstützenden Kooperationsumgebung.

Weiterhin werden die Szenarien in zwei Gruppen unterteilt. Die erste Gruppe bilden die Szenarien der *Präsenzkooperation*, die der „natürlichen“ Kooperation bei einem persönlichen Treffen der Teilnehmer entspricht. Die zweite Gruppe bildet die *Kooperation mit räumlich verteilten Teilnehmern* und orientiert sich eher an klassischen CSCW-Szenarien mit örtlich getrennten Kooperationspartnern, überträgt diese aber auf eine mobil-verteilte Kooperationsumgebung. Die Szenarien beider Gruppen werden grundsätzlich auch hinsichtlich der Möglichkeit des *nahtlosen Übergangs zwischen Präsenzkooperation und räumlich getrennter Kooperation* betrachtet. Sie tragen zudem der Option Rechnung, existierende CSCW-Systeme in die Kooperation integrieren zu wollen.

Die entworfenen Szenarien sind die Grundlage für die Beschreibung der technischen Rahmenbedingungen der Umsetzungsebene. Die in Kapitel 3 untersuchten Anforderungen an die Musterarchitektur mobil-verteilter Wissensräume resultieren direkt aus den Handlungsmustern der einzelnen Szenarien. In dem aufgestellten Anforderungskatalog an die zukünftige mobil-verteilte Kooperationsumgebung identifiziert diese Arbeit als wesentliche Problembereiche die „*Vernetzung und Sichtbarkeit*“ der Kooperationspartner, die „*Kontextualisierung*“ der Kooperationsumgebung und die „*Konsistenz*“ der verteilt und redundant gespeicherten Kooperationsobjekte.

Die Anforderungen im Bereich „*Vernetzung und Sichtbarkeit*“ der Kooperationspartner werden als direkte Folge der Mobilität der Benutzer in den Szenarien identifiziert. Sie lassen sich zumeist der Phase der *Gründung* zuordnen. Erster Schritt der Kooperation ist stets die Etablierung einer Kommunikation zwischen den Kooperationspartnern. Mangels eines zentralen Verzeichnisses verfügbarer Dienste und Benutzer müssen die Informationen in der verteilten Umgebung gesammelt und aufbereitet werden. Erst anhand dieser Informationen kann eine Kooperationsgruppe etabliert werden.

Speziell im Umfeld großer Kooperationsstrukturen und verschiedenster angebotener Dienste sind Mechanismen der *Kontextualisierung* der Kooperationsumgebung notwendig, um eine Auswahl potentieller Kooperationspartner und -dienste treffen zu können. Diese Forderung verlangt insbesondere eine Unterstützung der Benutzer durch eine Bereitstellung von Kontextinformationen. Diese betreffen sowohl potentielle Kooperationspartner als auch verfügbare Dienste im aktuellen Kooperationsumfeld. Daher können Kontextinformationen in jeder Phase der Kooperation genutzt werden, um die Benutzer im Kooperationsprozess zu unterstützen.

Die ebenfalls ermittelte Anforderung der *Konsistenz* der Kooperationsobjekte entstammt der in der Mobilität der Nutzer begründeten Forderung nach einer *Verteilung* und *Replikation* der gemeinsamen Kooperationsobjekte auf die mobilen Geräte aller beteiligter Kooperationspartner, um eine ständigen Verfügbarkeit der gemeinsam genutzten Dokumente zu gewährleisten (*Offline-Verfügbarkeit*). Bei einer Zu-

sammenarbeit mittels derart stark replizierten Datenräumen müssen alle Repliken stets synchronisiert werden. Im Falle eines Konflikts müssen zudem die abweichen den Änderungen beliebig rückgängig gemacht oder verändert werden können. Diese Anforderungen werden in dieser Arbeit unter den Begriffen „*Synchronizität*“ und „*Reversibilität*“ zusammengefasst.

Neben den Anforderungen, die aus der neuen Qualität der mobilen Nutzung verteilter Kooperationsumgebungen entstehen, werden des Weiteren auch die Funktionen herkömmlicher zentralisierter Kooperationssysteme speziell im mobilen Nutzungskontext betrachtet. Dabei offenbart sich ein Anpassungsbedarf herkömmlicher Funktionen der Kooperationsunterstützung für deren Nutzung in mobil-verteilten Systemen. Diese Funktionen entstammen den Basisdiensten der Bereiche *Kommunikation*, *Koordination* und *Kooperation*, welche die Basis der Kooperation in mobilen Nutzungsszenarien darstellen. Aufgrund ihres neuartigen mobil-spontanen Charakters wird diese Form der Kooperation im Rahmen dieser Arbeit *Kollaboration* getauft.

Die in Kapitel 3 ermittelten Anforderungen an eine Basisarchitektur für die Bereitstellung virtueller Wissensräume in einem mobil-verteilten Nutzungsumfeld werden in Kapitel 4 auf bereits existierender Lösungsansätze aus den Forschungsfeldern der mobilitätsunterstützenden Netzwerktechnologien, der verteilten Kommunikations- wie Persistenzsysteme und der Kooperationssysteme untersucht.

Es zeigt sich, dass die benötigten Protokolle zur Vernetzung und Kommunikation in mobilen und weiträumig verteilten Systemen vorhanden, aber noch nicht in alltägliche Arbeitsumgebungen integriert sind. Mittels einer Kombination von existierenden und spontan etablierten Netzwerken lässt sich eine weiträumige und flexible Kommunikationsstruktur aufbauen. Für die noch fehlende automatische Vernetzung verteilt bereitgestellter Dienste in derartigen Netzwerken existieren ebenfalls Lösungen, die bereits in heutigen Netzwerkstrukturen zum Einsatz kommen.

Anders verhält es sich bei Systemen zur Realisierung verteilter Persistenz. Zwar findet sich eine große Anzahl von Lösungsansätzen, insbesondere im Bereich der verteilten Hash-Tabellen (DHT), sie erweisen sich aber als zunächst wenig geeignet für mobile Kooperationsszenarien. Durch ihre Konzentration auf die Aufrechterhaltung des Gesamtsystems bei Knotenausfällen erlauben sie keine *Offline-Verfügbarkeit* des unverbundenen Knotens. Wie die im Rahmen dieser Arbeit entwickelten Nutzungsszenarien zeigen, spielt diese jedoch in mobilen Kooperationsszenarien eine wichtige Rolle, da eine Aufrechterhaltung der Grundfunktionalität der Kooperationsumgebung und somit eine lokale Replikation der Kooperationsobjekte auf dem unverbundenen Knoten benötigt wird. Dennoch liefern die DHTs dank eines von ihnen aufgespannten so genannten *Overlay-Netzwerks* interessante Ansätze für die Adressierung von Knoten in verteilten Systemen. Daher wird in der vorliegenden Arbeit der Ansatz der DHTs weiter verfolgt. Die beschriebenen Beschränkungen bezüglich der Gewährleistung einer Offline-Verfügbarkeit werden dazu mit Hilfe der in Kapitel 5 entworfenen Distributions- und Replikationsstrategie umgangen.

Weiterhin wird in dieser Arbeit mit den bewährten Tuple Spaces ein Konzept aufgegriffen, das einen einfachen und räumlich wie zeitlich entkoppelten Zugriff auf die

Kooperationsobjekte erlaubt. Bereits existierende Lösung für spontan vernetzte Tuple Spaces fehlt aber ein Konzept zur Gewährleistung einer persistenten Speicherung gemeinsam genutzter Kooperationsobjekte unter Wahrung der Offline-Verfügbarkeit. Diese Lücke wird mit dem in Kapitel 5 entworfenen Konzept der *Gruppen-Tuple Spaces* geschlossen.

In Kapitel 4 werden die wesentlichen Komponenten für eine Musterarchitektur mobil-verteilter Wissensräume identifiziert. Im darauf folgenden Kapitel 5 wird die Integration der gewählten Lösungsansätze betrachtet. Insbesondere werden die ermittelten Defizite der betrachteten Lösungsansätze hinsichtlich einer Unterstützung mobil-verteilter Kooperation systematisch behoben. Dazu werden sie mit eigens auf die Bedürfnisse einer mobil-verteilten Kooperation zugeschnittenen konzeptionellen und technischen Bausteine zu der gesuchten Musterarchitektur ergänzt.

Als Grundlage für die technische Realisierung einer verteilten Persistenz dienen die innovativen Konzepte der DHTs. Die vorliegende Arbeit entwirft zusätzlich, basierend auf den Anforderungen der betrachteten Szenarien, eine gezielte Distributions- und Replikationsstrategie inklusive einer Versionierung der verteilten Objekte. Die Versionierung dient dabei der Überwachung der Konsistenz des verteilten Datenraumes und unterstützt die kooperative Behebung von Konsistenzkonflikten. Eventuelle Konfliktsituationen, die nicht von der Kooperationsumgebung automatisch behoben werden können, werden mit einem an verteilte Dateisysteme angelehnten Mechanismus gelöst. Er erlaubt den Benutzern, von einander abweichende Versionen kooperativ zusammenzuführen.

Mit dieser Distributions- und Replikationsstrategie ist es möglich, Objekte gezielt an bestimmte Knoten des Kooperationsnetzwerkes zu senden und zwischen diesen zu replizieren. Was fehlt, ist eine logische Struktur, die hilft, die Replikation der Objekte zu steuern, und ein Mechanismus, um zwischen zeitweise unverbundenen Knoten zu kommunizieren. Zu diesem Zweck wird im Rahmen dieser Arbeit das Konzept der *Gruppen-Tuple Spaces* entworfen, das eine Bündelung replizierender Knoten in „Gruppen“ erlaubt. Die Knoten einer Gruppe teilen sich einen gemeinsamen und leicht zugreifbaren replizierten Speicher. Gleichzeitig wird eine *zeitliche und räumlich entkoppelte Kommunikation zwischen den Knoten* erreicht, da die Kommunikation in dem Tuple Space über eine zeitweise Ablage von Nachrichtenobjekten im gemeinsamen Tuple Space realisiert wird. Die Nachrichtenobjekte werden automatisch an den Empfänger ausgeliefert, wenn dieser sich das nächste Mal mit dem Tuple Space verbindet. Dieser Kommunikationsmechanismus wird des Weiteren für die Ereigniskontrolle und Koordination der verteilten Umgebung genutzt.

Auf diese Abstraktionschicht setzen die mobil-verteilten Wissensräume auf. Jeder Kooperationsgruppe wird mit ihrem zugehörigen Arbeitsbereich ein *Gruppen-Tuple Space* zugeordnet. In diesem werden sowohl die Kooperationsobjekte als auch die für die Benutzer normalerweise nicht sichtbaren Verwaltungsobjekte abgelegt. Durch diese persistente Speicherung der gesamten Kooperationssitzung finden die Gruppenteilnehmer ihren Kooperationsbereich beim Betreten stets so vor, wie sie ihn zum Ende der letzten Kooperationssitzung verlassen haben. Dabei kann die Kooperation in der vom virtuellen Wissensraum gewohnten Weise mittels Anwendung

der *primären Medienfunktionen* auf die Kooperationsobjekte erfolgen. Mit dieser neuartigen Verknüpfung verteilter und stark replizierter Speicherkonzepte mit virtuellen Wissensräumen etabliert diese Arbeit ein Konzept für eine durchgängige Verfügbarkeit der gemeinsamen Kooperationsdaten.

Die hier entworfene Musterarchitektur gliedert sich somit in vier Schichten. Die *Vernetzungsschicht* erlaubt eine generelle Kommunikation zwischen den Knoten des Kooperationsnetzwerkes und ein Auffinden potentieller Kooperationspartner. Die Schicht zur *Objektverwaltung und -speicherung* setzt auf diese Kommunikationsschicht auf und bildet den technischen Kern der Basisarchitektur. Sie enthält neben der Adressierung der Knoten Mechanismen für die gezielten Replikation der Kooperationsobjekte und deren Versionierung. Die Schicht der *Gruppen-Tuple Spaces* abstrahiert diese technische Sicht in ein schlüssiges Modell zur Bündelung von replizierenden Knoten. Sie bietet diesen außerdem Mechanismen für eine zeitlich und räumlich entkoppelte Kommunikation. Als oberste Schicht ermöglichen die *mobil-verteilten Wissensräume* den Einsatz bewährter Kooperationskonzepte auch in mobil-spontanen Nutzungsszenarien.

War eingangs dieser Arbeit die Realisierung einer mobilitätsfreundlichen Kooperationsumgebung in Bezug auf die funktionalen und technischen Anforderungen unklar, so helfen die entwickelten Nutzungsszenarien und aufgezeigten Lösungsansätze die wesentlichen technischen Problemstellungen zu identifizieren. Ausgehend von den im Rahmen der vorliegenden Arbeit entwickelten Nutzungsszenarien erlaubt diese Musterarchitektur mobil-verteilter Wissensräume die Umsetzung einer mobilen Kooperationsumgebung. Diese ermöglicht mobilen Benutzern eine freie Kooperation ohne Nutzungshemmisse durch äußere Gegebenheiten. Dabei ist jederzeit eine Einbettung externer Dienste in die Kooperationsumgebung sowie ein nahtloser Übergang zwischen Präsenzkooperation und räumlich getrennter Kooperation möglich. Trotz des komplexen Umfeldes einer mobil-spontanen Kooperation wird der Benutzer durch Nutzung verfügbarer Kontextinformationen und einer automatischen Konfiguration der Kooperationsumgebung auch im heterogenen Umfeld nicht mit technischen Fragestellungen belastet.

Die Arbeit schließt mit einer kurzen Zusammenfassung der einzelnen Kapitel und gibt einem Ausblick auf zukünftige Forschungsfragen.

Mobilität in der kooperativen Wissensarbeit

2 Spontane Wissensorganisation im mobilen Nutzungsumfeld – Szenarien

Die Unterstützung der Kooperation in einem mobilen Nutzungsumfeld ist an viele Rahmenbedingungen gebunden. Diese bestehen zum einen aus der technischen Systemumgebung und zum anderen aus den möglichen Nutzungsszenarien. Die technische Systemumgebung ist dabei durch portable Hardwarearchitekturen, wie z. B. Notebook, PDA oder Mobiltelefon, und die mögliche mobile Netzwerkanbindung, wie z. B. WLAN, Bluetooth, GRPS, UMTS, etc., geprägt. Die Nutzungsszenarien sind im Gegensatz zu den technischen Rahmenbedingungen nur schwer zu identifizieren und zu benennen. Um eine Grundlage zur Ermittlung der Anforderungen an eine Basisinfrastruktur für die spontane und auch geplante Kooperation in einem mobilen Nutzungsumfeld zu schaffen, werden in diesem Kapitel wesentliche Szenarien einer mobilen Kooperation beschrieben.

In allen hier beschriebenen Szenarien werden die Benutzer durch ihnen persönlich zugeordnete mobile Geräte in ihrer Mobilität begleitet. Die Szenarien ordnen sich somit in dem unter dem Begriff des „*Normadic Computing*“ [Kleinrock, 1995] bekannten Paradigma hochmobiler Benutzer in einem ständig wechselnden technischen Umfeld ein. Im Gegensatz zu der von Mark Weiser geprägten Sichtweise des „*Ubiquitous Computing*“ [Weiser, 1991] besteht das Ziel also nicht darin, die Kooperationsumgebung in Alltagsgegenständen einzubetten, sondern in einer Nutzung der vorhandenen persönlichen mobilen Computer für die Kooperation mit anderen Menschen. Dennoch laufen die folgenden Szenarien nicht quer zu einem durch Computer angereicherten Lebensumfeld, sondern beziehen in der Umgebung verfügbare Dienste mit ein, ohne von diesen abhängig zu werden. Die Benutzer sollen Zeitpunkt und Ort der Kooperation frei und abhängig vom aktuellen Umfeld wählen können und dabei ihre bevorzugten und auf sie konfigurierten Geräte benutzen können.

Die Realisierung einer derart gestalteten mobilitätsfreundlichen Kooperationsumgebung ist ohne ein Wissen um die Nutzungsszenarien und die strukturellen Grenzen einer technischen Realisierung nicht möglich. Klassische CSCW-Szenarien beschränken sich jedoch häufig auf die Zusammenarbeit in etablierten Infrastrukturen, während sich Szenarien der mobilen Vernetzung auf die Aspekte mobilitätsfreundlicher Netzwerke konzentrieren. Um die fehlende Konvergenz kooperativer Wissensstrukturierung und spontaner Vernetzung zu beschreiben, werden in diesem Kapitel Nutzungsszenarien entwickelt, die eine Verknüpfung von mobilen und spontanen Netzwerken mit kooperationsunterstützenden Infrastrukturen betrachten. Anhand

des Wechselspiels zwischen diesen Nutzungsszenarien werden die Probleme untersucht, die sich aus einer Zusammenführung der Techniken aus dem Feld der mobilen Vernetzung und des CSCW ergeben.

Für eine grobe Einordnung der benötigten Kooperationsunterstützung lassen sich die möglichen Kooperationsformen in unterschiedliche Kategorien unterteilen. Hier hat sich die Unterteilung in *Präsenzkooperation* und *entfernte Kooperation*, sowie *synchroner* und *asynchroner Kooperation* etabliert. Während Präsenzkooperation und entfernte Kooperation etwas über die räumliche Relation der Kooperationspartner aussagt, beschreiben die Kategorien der synchronen und asynchronen Kooperation, ob die Kooperation zum gleichen Zeitpunkt (*synchron*) oder zu versetzten Zeitpunkten (*asynchron*) stattfindet.

Ein Online-Chat ist demnach ein synchroner Kooperationsdienst während E-Mail zumeist asynchron genutzt wird. Des Weiteren ist ein Smartboard ein Hilfsmittel für die Präsenzkooperation, bei der alle Teilnehmer anwesend sind, und E-Mail wiederum ein Dienst zur entfernten Kooperation, bei der die Teilnehmer meist räumlich von einander getrennt sind.

Wie man an dem Beispiel des E-Mail-Dienstes sieht, ist die Zuordnung eines Dienstes zu einer bestimmten Kooperationskategorie nicht immer eindeutig möglich. Dennoch lässt sich mit Hilfe dieser Kategorien leicht erkennen, ob ein Dienst für das geplante Kooperationsszenario geeignet ist. Durch eine Einordnung der in diesem Kapitel beschriebenen Szenarien in eine oder mehrere Kategorien lassen sich daher bereits erste Rückschlüsse auf die benötigten Kooperationsdienste ziehen.

Der hier verfolgte Ansatz ähnelt den in [Booch et al., 1998] beschriebenen *Usecases*¹. Die Grundidee der Verwendung von Usecases ist, die Entwickler eines Softwaredirektives davor zu bewahren, während der technischen Realisierung die benötigte Funktionalität aus den Augen zu verlieren und sich stattdessen durch technische Aspekte leiten zu lassen. Eine ähnliche Funktion übernehmen die Szenarien in diesem Kapitel. Eine Formalisierung dieser Szenarien geht dabei jedoch nur so weit, wie es eine Präzisierung der einzelnen Handlungen der Kooperationspartner erfordert, um die allgemeine Verständlichkeit der Szenarien nicht zu gefährden.

Die Szenarien dieser Arbeit fußen auf dem Konzept der *virtuellen Wissensräume* [Hampel, 2001], die bereits in der serverzentrierten Architektur des *open-sTeam*-Systems² implementiert sind. Ein Ziel der vorliegenden Arbeit ist es, dieses flexible Konzept der dokumentenzentrierten Kooperation in mobile Nutzungsszenarien zu überführen und eine entsprechende technische Dienste-Infrastruktur zu entwerfen. Daher werden in den folgenden Szenarien insbesondere die mobile Kooperation in virtuellen Wissensräumen betrachtet und bezüglich einer technischen Unterstützung der Zusammenarbeit durch eine mögliche Kooperationsinfrastruktur analysiert.

Dieses Kapitel gliedert sich in zwei Szenariengruppen entsprechend der räumlichen Beziehung der Kooperationspartner zueinander. Die erste Szenariengruppe „*Prä-*

¹Die *Usecases* werden in der Beschreibungssprache *Unified Modeling Language (UML)* weiter formalisiert und verfeinert.

²Im Folgenden wird *open-sTeam* der besseren Lesbarkeit wegen *sTeam* genannt.

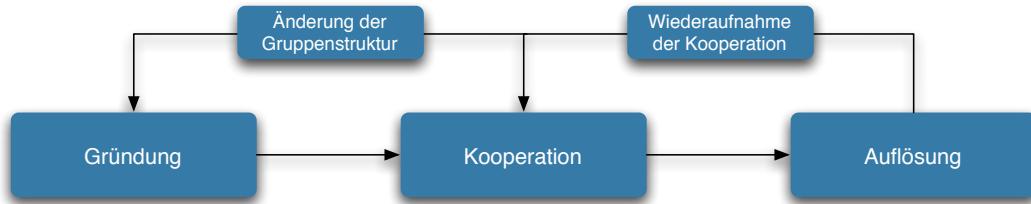


Abbildung 2.1: Der Prozess einer mobil-spontanen Kooperation gliedert sich in drei Phasen: Die *Gründung* steht am Anfang und besteht aus der Gruppengründung und der Erstellung des gemeinsamen Arbeitsbereiches. Die eigentliche *Kooperation* findet in der gleichnamigen Phase statt. Das *Auflösen* der Kooperation erfolgt aus Sicht der einzelnen Teilnehmer immer am Ende ihrer Beteiligung an der Kooperationssitzung. Sie können dieser aber jederzeit in einer anderen Nutzungskonstellation wieder beitreten.

„*Präsenzkooperation*“ betrachtet hauptsächlich Kooperationsformen bei der alle Teilnehmer räumlich zugegen sind. Wohingegen die zweite Szenariengruppe „*Kooperation mit räumlich verteilten Teilnehmern*“ hauptsächlich Kooperationsformen mit räumlich entfernten Kooperationspartnern betrachtet. In beiden Gruppen wird dabei auch auf Mischszenarien beider Kooperationsformen Bezug genommen.

Jede der zwei Gruppen enthält wiederum zwei Hauptscenarien. Bei der Gruppe „*Präsenzkooperation*“ sind dies die Hauptscenarien „*Lerngruppe*“ und „*Spontane Kooperation*“. Die Gruppe „*Kooperation mit räumlich verteilten Teilnehmern*“ enthält die Hauptscenarien „*Räumlich getrennte Kooperation*“ und „*Distant Learning*“.

Die Hauptscenarien sind in drei Kooperationsphasen unterteilt, die den Ablauf eines Kooperationsszenarios in *Gründung*, *Kooperation* und *Auflösung* gliedern. Die Phase der *Gründung* enthält die Prozesse, in denen die Teilnehmer eine Kooperation bewusst oder unbewusst beginnen. Die Phase der *Kooperation* betrachtet die eigentliche Zusammenarbeit zwischen den Teilnehmern. Die Phase der *Auflösung* ist besonders im kooperativen Nutzungskontext wichtig, da sie die Archivierung der Kooperationsergebnisse einschließt und in mobilen Szenarien oft einen Übergang zu einer neuen Nutzungskonstellation in der Kooperationsumgebung darstellt. Die *Auflösung* geht somit oft direkt in die *Gründung* einer Kooperation mit anderer Konstellation über. So entsteht eine Art Regelkreis, in dem die Kooperationskonfiguration ständig dem Nutzungskontext angepasst wird (Abbildung 2.1).

Die insgesamt vier Hauptscenarien enthalten jeweils mehrere detaillierte Fallbeschreibungen, die eine Identifizierung der benötigten Unterstützungsfunktionen erlauben. Beispiele dieser detaillierten Fallbeschreibungen sind die Szenarien „*Gründung einer Gruppe aus dem Ortskontext der Teilnehmer*“ (Szenario 1.1), „*Kooperatives Arbeiten in einem gemeinsamen Gruppenbereich*“ (Szenario 1.6) und „*Überführen einer zentral verwalteten Kooperationsgruppe in eine mobil-verteilte Kooperationsumgebung*“ (Szenario 3.1). Es empfiehlt sich im Folgenden für einen Überblick

über das Hauptszenario auf den Fließtext und für einen tieferen Einblick in die Einzelszenarien auf die grauen Boxen mit den detaillierten Fallbeschreibungen zurückzugreifen.

2.1 Präsenzkooperation

In die Gruppe der Szenarien der *Präsenzkooperationen* oder auch *Face-to-Face Kooperationen*, ordnen sich alle Kooperationssituationen ein, in denen die Kooperationspartner direkt und ohne räumliche Trennung miteinander interagieren. Aufgrund dieser Umstände haben die Benutzergruppen daher eine begrenzte Größe und verständigen sich zumeist ohne technische Hilfsmittel über direkte Sprache und Gestik. Bei der Kommunikation sind alle Kooperationspartner gleichberechtigt und erfolgt an alle Partner zugleich. Es handelt sich demnach um eine ungerichtete Kommunikation zwischen allen Teilnehmern. In einigen Fällen mag jedoch ein Moderator die Kooperation steuern, um den Kooperationsprozess zu koordinieren. Dies macht dann Sinn, wenn die Kooperationspartner sich ansonsten durch ihre Handlungen behindern würden, weil sie gemeinsam auf einige wenige Kooperationsressourcen zugreifen.

Die Kooperationsressourcen bilden hier oftmals technische Hilfsmittel wie Wissensquellen in Form von Literatur und Notizen, die während der Kooperationssitzung mit Beschriftungen, Bleistiftskizzen oder Tafelbildern ergänzt werden. Durch das verstärkte Aufkommen von mobilen Computern und digitalen Medien werden viele der Hilfsmittel und Ressourcen inzwischen durch ihre digitalen Pendants ergänzt oder ersetzt. Es kommt zu einer Vermischung von herkömmlichen und computerisierten Hilfsmitteln.

In einem solchen Kooperationsumfeld gilt es, diese technischen Hilfsmittel nahtlos und ohne *Medienbrüche* einzubinden. Dies bedeutet, die natürliche Kooperation durch die technischen Hilfsmittel zu ergänzen und die herkömmlichen Kooperationsmaterialien möglichst einfach in die computergestützte Kooperation zu integrieren. Dieses Unterfangen ist aufgrund der vielen nicht digitalisierten Materialien wie Schmierzettel, Bücher, Mitschriften, etc. schwierig. Der Trend zeigt jedoch, dass die digitalen Hilfsmittel zunehmend eine wichtige Stellung im Alltagsleben der Benutzer einnehmen³. Zusätzlich kann ein verstärkter Einsatz dieser digitalen Hilfsmittel helfen, die unvermeidlichen Medienbrüche durch einen Wechsel zwischen analoger und digitaler Welt zu reduzieren.

Um Medienbrüche, die durch den Wechsel zwischen analogen zu digitalen Hilfsmitteln entstehen, zu vermeiden, wird in den Szenarien – falls verfügbar – direkt die digitale Variante gewählt. Dabei wird davon ausgegangen, dass die digitale Lösung zumindest gleichwertig zu ihrem analogen Vorbild ist. Eventuelle Nachteile der aktuellen digitalen Umsetzungen analoger Hilfsmittel werden der Einfachheit halber bei dieser Betrachtung ignoriert und beide Varianten als zumindest gleichwertig ange-

³ Als Beispiele seien hier *eBooks* und *Digital Ink* als Ersatz für Literatur in Papierform, *TabletPC* als Ersatz für Notizzettel und *Smart Whiteboards* als Ersatz für Tafeln genannt.

nommen. Es scheint eine realistische Annahme zu sein, dass die digitalen Hilfsmittel sich mehr und mehr den Analogen angleichen und zusätzlich einen funktionalen Mehrwert bieten werden. Die Nutzung der neuen digitalen Hilfsmittel entwickelt besonders in Verbindung mit mobilitätsfreundlichen Kommunikationsschnittstellen und -infrastrukturen spannende Möglichkeiten für eine nahtlose Kooperation im mobilen Umfeld.

2.1.1 Lerngruppe

Das erste Szenario der Präsenzkooperation mobiler Benutzer bildet eine studentische Lerngruppe. Lerngruppen haben üblicherweise wenige Mitglieder, die sich nach losen Verabredungen an einem geeigneten Ort treffen. Die Gruppenstärke variiert von Treffen zu Treffen durch die Abwesenheit einiger Teilnehmer oder das Hinzukommen von Gästen. Ziel einer solchen Lerngruppe ist das gemeinsame Aufarbeiten eines bestimmten Wissensgebietes bzw. eines vorgegebenen Lernstoffes. In den meisten Fällen schließt sich an die Lernphase eine zu absolvierende Prüfung an einem festen Termin an. Alle Kooperierenden arbeiten daher auf ein gemeinsames Ziel hin, welches jedoch nicht nur durch die gemeinsam geschaffenen Artefakte, sondern auch durch das individuell erlangte Wissen geprägt ist.

Gründung, Anbahnungsphase Der organisatorische Rahmen einer Lerngruppe ist durch eine lose und gleichberechtigte Kooperation aller Teilnehmer geprägt. Die Gründung der Gruppe erfolgt durch Absprache während eines spontanen Treffens im Vorlesungsumfeld oder auch geplant durch einen Aushang.

Die schnelle Identifizierung potentieller Kooperationspartner ist also eine erste wichtige Komponente der Kooperationsumgebung. Potentielle Kooperationspartner können z. B. Benutzer in räumlicher Nähe oder mit ähnlichem Interessenprofil sein. Die Initiatoren lassen den potentiellen Partnern eine Kooperationsanfrage zukommen. Wird die Anfrage positiv beantwortet, fügen sie die Teilnehmer der Lerngruppe hinzu – unter Umständen ist aber auch eine Bewerberliste sinnvoll, welche die Initiatoren nutzen können um die passenden Kandidaten auszuwählen.

Szenario 1.1: Gründung einer Gruppe aus dem Ortskontext der Teilnehmer

- Gründung einer Gruppe mittels Auswahl anwesender Kooperationspartner

Die Benutzerin T_1 einer mobilen Kooperationsumgebung möchte anhand von potentiellen Teilnehmern (T_1, \dots, T_n) eine Lerngruppe mit zunächst höchstens 6 Mitgliedern gründen, um für eine Prüfung zu lernen. Mit zwei der gewünschten Teilnehmer (T_2, T_3) hat sie sich am Rande einer Vorlesung getroffen. Um die Gruppe anhand der anwesenden Teilnehmer zu gründen, lässt sie sich alle Benutzer der Kooperationsumgebung in ihrer Nähe anzeigen. Dies sind die Benutzer T_2, T_3 und T_x . Da auch nicht gewünschte Benutzer als mögliche Gruppenmitglieder in der Umgebung

vorhanden sind, wählt sie die Benutzer, die zur Gruppe gehören sollen (T_2 und T_3), aus der Menge der Anwesenden aus. Aus dieser Selektion heraus gründet sie eine neue Gruppe G innerhalb der Kooperationsumgebung. Die nicht anwesenden aber bekannten Benutzer (T_4 , T_5 und T_6) lassen sich vorläufig hinzufügen und bekommen so eine Einladung nach deren Bestätigung sie automatisch Mitglieder der Gruppe G werden.

Neben der Gruppengründung mit Anwesenden bei einem spontanen Treffen, kann eine solche Gründung auch durch eine Art *Aushang* oder *Laufzettel* angekündigt werden. In diesem Fall wird eine Nachricht an alle potentiellen Teilnehmer gesandt. Um auch Teilnehmer zu erreichen, die derzeit nicht im System verfügbar sind, wird die Anfrage *epidemisch* – also bei Kontakt von Benutzer zu Benutzer – weitergegeben. Diese Anfrage trägt ein *Gültigkeitsdatum*, welches festlegt, wie lange der Aushang weitergereicht werden soll. Läuft dieses Datum ab, verfällt die Anfrage und wird nicht mehr an weitere Benutzer weitergereicht.

Interessierte Benutzer, die an der Lerngruppe teilnehmen wollen, schicken eine Antwort auf die Anfrage in Richtung der initiiierenden Benutzer. Diese können noch entscheiden, wer endgültig in die Lerngruppe aufgenommen wird. Sie sind somit Verwalter der Lerngruppe. Die Verwaltungsrechte können später an andere Teilnehmer weitergegeben werden, um die Lerngruppe zu erweitern oder Gäste einzuladen zu können.

Soll die Lerngruppe nicht öffentlich ausgehangen werden oder wird die Lerngruppe durch ein persönliches Treffen initiiert, kann der Prozess der Ankündigung für einen begrenzten Benutzerkreis verwendet werden. Dazu können die gründenden Teilnehmer eine Auswahl der Zielgruppe durch Benennen der gewünschten Teilnehmer oder gewünschter Attribute potentieller Teilnehmer vornehmen. Mögliche Attribute wären hier z.B. *Teilnehmer der Vorlesung „Architekturen des CSCW“* oder – im Fall des persönlichen Treffens – *alle anwesenden Personen*.

Szenario 1.2: Gruppengründung mittels Aushang und Eingrenzung des Nutzerkreises

- Gründen einer Gruppe mit unbekanntem Benutzerkreis
- Definieren gewünschter Benutzerprofile für die Gruppengründung

Der Benutzer T_1 einer mobilen Kooperationsumgebung möchte eine Lerngruppe gründen, ohne alle der möglichen Teilnehmer bereits zu kennen. Er macht deshalb eine systemweite Ankündigung, die alle in Frage kommenden Benutzer erreichen soll. Da es sich um eine Lerngruppe zur Vorlesung „*Architekturen des CSCW (ACSCW)*“ handelt, beschließt er, nur Benutzer zu adressieren, die an dieser Vorlesung teilnehmen. Außerdem sollen alle Teilnehmer in der selben Stadt (hier: Paderborn) wohnen, um Präsenztreffen leichter organisieren zu können. Dazu

gibt er beide Kriterien als Filter über die Attribute *Gruppe=ACSCW* und *Adresse.Stadt=Paderborn* ein. Nur Benutzer, die diesen Kriterien entsprechen, bekommen die Ankündigung angezeigt.

Um eine Verbreitung an einen möglichst großen Benutzerkreis zu gewährleisten, wird der Aushang innerhalb der computergestützten Kooperationsumgebung an *alle Benutzer* weitergereicht sobald diese kontaktiert werden. Die Benutzer reichen den virtuellen Aushang wiederum an andere Benutzer weiter, die sie treffen, auch wenn sie selbst nicht zum gewünschten Personenkreis gehören. Nach einer zuvor definierten Zeit wird die Ankündigung wieder entfernt und von allen Geräten gelöscht.

Benutzer, die an der Lerngruppe teilnehmen wollen, beantworten die Ankündigung positiv. Der Initiator bekommt diese Benutzer daraufhin in einer Liste potentieller Teilnehmer angezeigt, aus der er die endgültige Gruppe zusammenstellen kann. Als Hilfsmittel für die Entscheidung über mögliche Mitglieder nutzt er Chat-Mechanismen und Benutzerprofile.

Der Aushang von T_1 erreicht die Benutzerin T_2 als sich ihr mobiles Gerät über ein Funknetzwerk mit dem Gerät von Benutzer T_1 verbindet. Allerdings wird Benutzerin T_2 der übergebene Aushang nicht angezeigt. Das Attribut *Adresse.Stadt=Paderborn* trifft bei ihr zwar zu, aber sie besucht nicht die Vorlesung „Architekturen des CSCW (ACSCW)“ und ist somit auch nicht Mitglied in der geforderten Gruppe ACSCW. Benutzerin T_2 lebt mit Benutzerin T_3 , deren Profil den gewünschten Vorgaben entspricht, in einer Wohngemeinschaft. Als T_2 in der gemeinsamen Wohnung eintrifft, wird die Ankündigung automatisch an die ebenfalls anwesende T_3 übermittelt und dieser aufgrund der Übereinstimmung des Profils angezeigt.

Teilnehmerin T_3 entschließt sich, der Lerngruppe beizutreten und schickt eine positive Antwort an den Initiator T_1 . Die Antwort wird ebenfalls epidemisch übertragen. Daher erhält T_2 eine Kopie der Antwort, ohne diese je zu Gesicht zu bekommen. Diesmal kommt T_3 während der besagten Vorlesung noch vor T_2 in Reichweite von T_1 und beantwortet die Anfrage somit direkt. Das Antwortobjekt von T_2 verfällt entweder nach einer gewissen Lebenszeit, wird beim nächsten Kontakt mit T_3 zurückgezogen oder erreicht T_1 und wird dann von dem Gerät von T_2 entfernt.

Nach der Gründung der Lerngruppe muss die Kooperationsumgebung die Benutzer zu einer Gruppe innerhalb des Systems zusammenschließen und einen gemeinsamen Arbeitsbereich zur Verfügung stellen. Gleichzeitig sind die Berechtigungen der Teilnehmer einer gleichberechtigten Kooperation entsprechend zu setzen. Dies bedeutet im Detail, dass der Gruppenbereich und alle enthaltenen Dokumente, bis auf einige gezielte Ausnahmen, für alle Teilnehmer voll zugreifbar sind. Zunächst sind also alle Dokumente im Besitz aller Gruppenmitglieder. Bei Bedarf werden die Ausnahmen nachträglich festgelegt und mit entsprechenden Berechtigungen versehen. Die Verwaltung der Gruppe selbst kann entweder durch die Gründungsmitglieder, die designierten Gruppenadministratoren oder alle Gruppenmitglieder erfolgen. In einer

gleichberechtigten Kooperation erscheint in der Regel eine Verwaltung seitens aller Gruppenmitglieder sinnvoll.

Szenario 1.3: Gemeinsamer Arbeits- und Lernbereich für Dokumente und aktive Objekte

- Gründen einer Gruppe mit einem gemeinsamen Bereich
- Berechtigungen im Gruppenbereich
- Entfernter Zugriff auf gemeinsam genutzte Objekte

Begleitend zur Vorlesung „Architekturen des CSCW (ACSCW)“ hat sich eine Lerngruppe mit vier Mitgliedern zusammengeschlossen. Die Gruppe nutzt einen gemeinsamen Arbeitsbereich für alle gemeinsam genutzten Objekte. Im Sinne einer offenen und gleichberechtigten Kooperation haben alle Teilnehmer die gleichen Zugriffsrechte auf die in dem Bereich enthaltenen Objekte. Wenn Benutzer neue Objekte in dem Gruppenbereich ablegen, haben alle Angehörigen der Gruppe das Recht, das Objekt gemäß seiner Relevanz zu bewerten, anzuordnen, zu kommentieren, zu bearbeiten oder sogar zu löschen, wenn sie es für irrelevant halten.

Die Abstimmung der Handlungen geschieht in der Regel mittels direkter Kommunikation zwischen den anwesenden Teilnehmern. Kommunikation mit abwesenden Teilnehmern wird über ein Nachrichtensystem koordiniert. Die Kommunikation bezüglich eines Objektes wird an das Objekt selbst gekoppelt und so ein intuitiver Bezug für die Kommunikation hergestellt. Damit abwesende Teilnehmer nicht ständig den Gruppenbereich aufsuchen müssen, um den Fortschritt zu verfolgen, kann die Kommunikation oder auch ein Änderungsprotokoll als persönliche Nachricht zugestellt werden.

Die Lerngruppe G besitzt vier Teilnehmer ($G = \{T_1, \dots, T_4\}$). Diese befinden sich derzeit in einer Kooperationsssituation und wollen gemeinsam das Dokument d_1 bearbeiten. Dieses Dokument liegt in einem der Gruppe zugeordneten Bereich der Kooperationsumgebung. Nur die Gruppenmitglieder haben auf diesen Bereich Zugriff. Innerhalb dieses Bereichs dürfen alle Gruppenmitglieder alle Objekte und Dokumente ändern, löschen und neu anlegen. In dem leeren Bereich legt T_1 das Dokument d_1 an, das von nun an alle Gruppenmitglieder lesen und ändern (schreiben) können. Des Weiteren legt T_1 ein Dokument d_2 an, in dem er eigene Notizen ablegen will. T_1 möchte diese Notizen auch dem Rest der Gruppe zugänglich machen, aber den anderen Gruppenmitgliedern nicht erlauben, den Inhalt zu ändern. Daher legt er das Dokument im gemeinsamen Gruppenbereich ab, entzieht aber allen Gruppenmitgliedern die Schreibberechtigung.

Für eine Kooperation mittels dieser Dokumente treffen sich T_1 , T_2 und T_3 in einem Raum und setzen sich um einen Tisch. Sie verfügen alle über mobile Geräte, die miteinander vernetzt sind. Teilnehmer T_4 befindet sich an einem anderen Ort und greift über eine Netzwerkverbindung auf dieselbe Kooperationsumgebung wie

die Gruppenmitglieder vor Ort zu. Während T_1 , T_2 und T_3 eine Gliederung für das Dokument d_1 erstellen, wird T_4 über jede Änderungen im Dokument informiert. Nachdem eine grobe Gliederung steht, fügt er einen neuen Gliederungspunkt in d_1 ein, was wiederum den anderen Gruppenmitgliedern sofort angezeigt wird.

Kooperationsphase Nach der Gründung der Gruppe verabredet die Gruppe sich für Präsenztreffen. Die Treffen können dabei für einen bestimmten wiederkehrenden Termin verabredet oder von Mal zu Mal neu bestimmt werden. Nicht anwesende Teilnehmer müssen über diese Termine informiert und bei Bedarf in die Terminfindung eingebunden werden. Neben *Benachrichtigungsmechanismen* erlauben *Abstimmungsfunktionen* und *Gruppenkalender* eine Unterstützung der Prozesse in der Terminfindung.

Szenario 1.4: Terminkoordination für die Präsenzsitzungen in Kleingruppen

- Unterstützung des Terminfindungsprozesses

Eine Gruppe von gleichberechtigten Benutzern möchte einen einmaligen Termin oder sogar wiederkehrende Termine für regelmäßige Treffen bestimmen. Hierzu ist es wichtig, dass möglichst alle Teilnehmer zu diesem Termin verfügbar sind. Üblicherweise wird eine erste Terminfindung bei dem ersten Treffen zusammen mit der Gruppengründung erfolgen. Generell gibt es zwei mögliche Vorgehensweisen:

- a) **Iterative Abstimmung:** Die Initiatoren der Lerngruppe machen einige Terminvorschläge, über die in der Gruppe abgestimmt wird. Der Termin mit der meisten Zustimmung wird gewählt. Sollte ein Teilnehmer an diesem Termin nicht können, muss die Wahl eventuell mit neuen Vorschlägen wiederholt werden. Dieses Verfahren bietet sich an, wenn alle Teilnehmer ähnliche Termine haben (z. B. einheitlicher Stundenplan von Teilnehmern in der gleichen Studienphase), ansonsten kann das Verfahren aus b) schneller zum Erfolg führen.
- b) **Terminüberdeckung:** Jeder der Teilnehmer nennt die eigenen, bereits existierenden wiederkehrenden Termine (Stundenplan). Der neue Termin wird in der Zeit geplant an dem keine Überdeckungen der existierenden Termine auftreten. Gibt es eine solche Übereinstimmung nicht, werden die existierenden Termine priorisiert oder bereits existierende Termine verschoben. Dies ist in der Regel ein interaktiver Prozess, bei dem die Teilnehmer die Wichtigkeit ihrer Termine argumentieren. Im Gegensatz zum Verfahren in a) muss jeder Teilnehmer alle seine Termine offen legen und Rechenschaft über diese führen.

Eine Präsenzsitzung beschleunigt die Terminfindung aufgrund deren stark interaktiven Charakters. Aber auch eine räumlich entfernte und zeitlich versetzte Form

der Terminbestimmung ist möglich. Das Ergebnis der Terminabstimmung für regelmäßige Termine muss aber im Gegensatz zu einmaligen Terminen den Konsens aller Teilnehmer finden.

Im Folgenden plant eine Gruppe von gleichberechtigten Benutzern einen einzelnen Termin für ein Präsenztreffen. Dies kann z. B. der erste Termin einer interaktiven Terminbestimmung für regelmäßige Treffen sein oder ein außerordentliches Treffen für eine zusätzliche Lernsitzung. Dieser einmalige Termin bedarf keiner strikten Verfügbarkeit aller Teilnehmer, Ziel ist aber dennoch ein möglichst hoher Konsens. Eine Bestimmung von regelmäßigen Terminen erfolgt analog.

In einer *iterativen Abstimmung* wird nach der Nennung möglicher Termine durch T_1 mehrheitlich einer der Termine gewählt. Die Mehrheit der Stimmen fällt auf einen Termin, an dem zwei Mitglieder nicht verfügbar sind. Deshalb beschließt die Gruppe die Abstimmung unter Ausschluss dieses Termins zu wiederholen. Die Teilnehmerin T_2 ist bei der Terminfindung nicht persönlich zugegen und soll später von dem mehrheitlichen Ergebnis in Kenntnis gesetzt werden. Weil die Abstimmung aber für einen wichtigen Termin erfolgt, an dem sie unbedingt teilnehmen möchte, überträgt sie ihr Stimmrecht auf das anwesende Mitglied T_3 . T_3 weiß aber bei dem gewählten Termin nicht ob T_2 diesen wahrnehmen kann und T_2 wird kurzfristig über technische Hilfsmittel wie Telefon oder Chat hinzugezogen. Stimmt T_2 zu, ist der Termin gefunden.

Die Teilnehmer T_1 , T_2 , T_3 und T_4 wollen nach abgeschlossener Gründung der Gruppe G mittels des Verfahrens der *Terminüberdeckung* einen Termin für das nächste Treffen verabreden. T_1 , T_2 und T_3 treffen sich zufällig in der Kantine. T_4 ist nicht anwesend. Sie ermitteln mit Hilfe eines Gruppenkalenders in ihrer Kooperationsumgebung einen möglichen Termin. Dazu legen sie Ihre persönliche Kalender übereinander und wählen einen bei allen Teilnehmern freien Termin. Der persönliche Kalender von T_4 kann nicht berücksichtigt werden, da dieser nicht vorliegt und zudem die Berechtigung zur Einsicht des persönlichen Kalenders von T_4 fehlt. Daher verzichten die drei Anwesenden zunächst auf eine Einbeziehung und bestimmen einen Termin auf die Gefahr hin, mit dem persönlichen Kalender von T_4 zu kollidieren. Der Termin wird in den Gruppenkalender eingetragen und somit auch T_4 bekannt gegeben. Da dieser Termin in dem persönlichen Kalender von T_4 bereits belegt ist, tritt er direkt über die Kooperationsumgebung mit den anderen drei Teilnehmern in Kontakt. Unter zusätzlicher Einbeziehung des Terminkalenders von T_4 wird daraufhin ein endgültiger Termin bestimmt.

Während der Präsenzsitzungen arbeiten die Lernpartner mit einem gemeinsamen Fundus von Dokumenten. Dieser wird größtenteils kooperativ erweitert und kommentiert, soll aber auch individuellen Bedürfnissen angepasst werden. Es müssen daher sowohl individuelles als auch kooperatives Arbeiten innerhalb derselben Arbeitsumgebung unterstützt werden, um diese nicht ständig wechseln zu müssen. Damit bei einer wachsenden Anzahl von Dokumenten die Übersicht nicht verloren geht, ist es wichtig, dass der Kooperationsbereich strukturiert werden kann – ähnlich zu

den in [Hampel, 2001] beschriebenen Szenarien. Da nicht alle Teilnehmer bei jeder Präsenzszitung anwesend sein können, müssen darüber hinaus Änderungen in dem gemeinsamen Kooperationsbereich seit der letzten Teilnahme für die abwesenden Benutzer kenntlich gemacht werden.

Szenario 1.5: *Individuelles und kooperatives Arbeiten an Dokumenten*

- Zusammenspiel von Gruppenbereich und persönlichem Bereich
- Verknüpfung von Dokumentversionen

Die Lerngruppe beschließt, den Lernstoff anhand des vorhandenen Vorlesungsskriptes aufzuarbeiten. Sie gehen zu diesem Zweck sequentiell durch den Stoff der Vorlesung und somit durch die Kapitel des Skripts. Während ihrer Lernsitzungen versehnen sie gemeinsam das Skript mit ihren Kommentaren oder fügen Zusatzmaterialien hinzu. Dabei werden zum Teil auch Änderungen eines anderen Teilnehmers nochmals bearbeitet, kommentiert oder entfernt, weil sie z. B. falsch oder ungenau sind.

Neben dem gemeinsam bearbeiteten Dokument besitzen alle Teilnehmer eine eigene Kopie des Skripts, in das persönliche und für die Kooperation irrelevante Kommentare einfügt werden. Auf diese privaten Daten darf kein anderer Teilnehmer zugreifen oder sie ändern. Optional können die Teilnehmer eine Kopie ihrer persönlichen Version wieder in die kooperative Fassung einfließen lassen oder umgekehrt, kooperative Anteile in die persönliche Version übernehmen.

Die Benutzer T_1 , T_2 und T_3 sind Mitglieder der Gruppe G . Sie bearbeiten in ihrem gemeinsamen Gruppenbereich eine elektronische Version des Vorlesungsskripts zur Vorlesung „*Architekturen des CSCW (ACSCW)*“. Das Skript wird als Dokument d_1 analog zu Szenario 1.3 im gemeinsamen Gruppenbereich abgelegt. Zusätzlich speichert jeder Teilnehmer eine Kopie d'_1 des Skript in seinen persönlichen Arbeitsbereich. Teilnehmerin T_1 macht in ihrer persönlichen Version einige Anmerkungen zu einem Abschnitt im Skript. Sie beschließt, dass diese Anmerkungen nützlich für ihre Kooperationspartner sein können und markiert diese für eine Übernahme in das gemeinsame Dokument d_1 . Daraufhin sind diese für alle Gruppenmitglieder sichtbar. Teilnehmer T_2 entdeckt einen Fehler in den Anmerkungen von T_1 und korrigiert diesen. Die Änderung wird in dem gemeinsamen Dokument wiederum sofort für alle sichtbar. In der persönlichen Version von T_1 bleibt die fehlerhafte Anmerkung zunächst bestehen, wird aber bezüglich der neu existierenden Version markiert. T_1 erkennt ihren Fehler und übernimmt die Änderung in ihr persönliches Skript – somit sind beide Versionen der Anmerkungen wieder synchron.

Obwohl der Ablauf des kooperativen Lernens in der Lerngruppe bedingt durch den Lernstoff, persönliche Vorlieben der Lernenden und die bewusst angewendeten Lernstrategien variiert, wird im Rahmen der hier beschriebenen Szenarien folgender Ablauf als typisch angenommen:

1. **Sammeln von Materialien:** Zu Beginn werden alle möglichen Quellen für das Erlernen des gewünschten Stoffes zusammengetragen. Dies geschieht in der Regel nicht arbeitsteilig. Die Teilnehmer suchen getrennt nach allen Materialien, die sie für sich relevant halten.
2. **Sichten der Materialien:** Die gesammelten Materialien werden gemeinsam von den Lernpartnern gesichtet und sortiert. Dabei werden die irrelevanten Materialien beiseite gelegt und zentrale Quellen in den Mittelpunkt des gemeinsamen Interesses gestellt.
3. **Aufarbeiten des Lernstoffes:** Die zentralen Quellen werden gemeinsam durchgearbeitet, aus anderen Quellen ergänzt sowie mit Kommentaren versehen.
4. **Verdichten des Wissens:** Das Verdichten des Wissens meint den Prozess, relevantes Wissen für das Erreichen des Lernziels zu ermitteln und aufzuarbeiten. Dazu strukturieren es die Lernenden häufig in eine Form die ihrem mentalen Modell entspricht. Dies kann z. B. mittels semantischer Karten erfolgen, die sich z. B. durch eine räumliche Anordnung im virtuellen Wissensraum verwirklichen lassen [Hampel und Eßmann, 2004]. Es sind aber natürlich auch andere Strukturierungsmittel möglich. Diese Struktur der Dokumente kann als ein Artefakt oder Ergebnis des Kooperationsprozesses betrachtet werden, das durch ein sukzessives neues Durchlaufen des beschriebenen Prozesses ständig verfeinert wird.

Szenario 1.6: Kooperatives Arbeiten in einem gemeinsamen Gruppenbereich

- Zentral gespeicherter Gruppenbereich
- Gruppenvernetzung bei einem Treffen
- Verhindern von zeitgleichen Änderungen
- Behandeln von Versionskonflikten
- Mitführen einer persönlichen Kopie

Nach der Gründung einer Lerngruppe G mit den Teilnehmern und Teilnehmerinnen T_1, \dots, T_6 innerhalb des Kooperationssystem (siehe Szenario 1.1 und Szenario 1.2) treffen sich deren Mitglieder erstmalig zu einem vereinbarten Termin (vgl. Szenario 1.4), um den aufzuarbeitenden Lernstoff zu sichten.

Teilnehmerin T_1 legt dazu zunächst die vom Lehrstuhl als PDF-Dokumente veröffentlichten Foliensätze (d_1), Übungszettel (d_2, d_3) und das Vorlesungsskript (d_4) in ihrem Gruppenbereich innerhalb der Kooperationsumgebung ab. Hinzu kommen

noch auf Vorschlag des Teilnehmers T_2 , ein Skript von einer verwandten Veranstaltung in die Sammlung aufzunehmen (d_5). Auch einige andere Dokumente aus den Beständen der einzelnen Teilnehmer finden Einzug in den gemeinsamen Gruppenbereich. So entsteht eine weit gefasste Sammlung von Materialien, die mit dem Thema der Vorlesung verwandt sind.

Die Treffen finden stets an einem verabredeten Ort in Anwesenheit aller Teilnehmer statt. Daher werden die gemeinsamen Dokumente in dem Gruppenbereich der Kooperationsumgebung auf dem Notebook von T_1 gespeichert. Dazu schafft dieser Teilnehmer einen für alle Teilnehmer zugreifbaren Bereich. Das Notebook wird zu jeder Sitzung mitgebracht. Zu Beginn der Lernsitzung verbinden sich alle Teilnehmer über ein bereitgestelltes Netzwerk mit dem Notebook und greifen von nun an über ihren persönlichen Rechner auf den zentralen Dienst zu. Teilnehmer T_2 verbindet zunächst seinen Rechner mit dem Netzwerk und über dieses mit der Kooperationsumgebung auf dem Notebook von T_1 . Die Kooperationsumgebung stellt einen Gruppenbereich für die Gruppe G zur Verfügung. Mit diesem Gruppenbereich verbindet sich T_2 und kann somit auf die Dokumente der Lerngruppe zugreifen. Er öffnet den Übungszettel d_3 und erarbeitet eine Lösung für die Aufgabenstellung. Gleichzeitig hat T_1 dasselbe Dokument d_3 geöffnet und ebenfalls eine Teillösung skizziert. T_2 speichert nun seine Lösung in die Datei d_3 ab. Wenig später speichert T_1 ihre Skizze ebenfalls in das gemeinsame Dokument d_3 und überschreibt somit die Lösung von T_2 , die daraufhin verloren ist. Glücklicherweise hat T_2 noch eine lokale Kopie des Dokuments d_3 mit seiner Lösung. Er überschreibt wiederum (diesmal gewollt) die Teillösung von T_1 im gemeinsamen Gruppenbereich. T_1 und T_2 nehmen noch einige Korrekturen an der Lösung vor und beenden die Lernsitzung.

Am Ende der Sitzung müssen die Teilnehmer ähnlich wie in Szenario 1.5 die persönlichen Versionen der gemeinsam genutzten Dokumente mit den kooperativen Änderungen in Einklang bringen, um diese jenseits der Kooperationssitzungen verfügbar zu haben – auf die kooperativen Versionen haben sie zwischen den Treffen keinen Zugriff. Alternativ könnte ein zentraler Server verwendet werden, der die Dokumente bereit stellt und von gewissen Orten aus erreicht werden kann.

T_2 möchte die Lösung gerne mit nach Hause nehmen und fertigt eine lokale Kopie an. Bis zur nächsten Sitzung nehmen sowohl T_1 als auch T_2 wiederum Änderungen an der Lösung von d_3 vor. Teilnehmerin T_1 ändert das Dokument d_1 direkt im Gruppenbereich auf ihrem Notebook. T_2 ändert seine lokale Kopie d'_3 , da er außerhalb der Kooperationssitzungen keinen Zugriff auf den gemeinsamen Gruppenbereich hat. Infolge dieser asynchronen Änderungen existieren wieder zwei Versionen des Dokuments d_3 , die beim nächsten Treffen zusammengeführt werden müssen. Die Kooperationsumgebung erkennt die abweichenden Versionen des Dokuments d_3 , indem sie sich den letzten Stand eines erfolgreichen Abgleichs merkt und bei Änderungen in beiden Versionen (Original d_3 und Kopie d'_3) die Benutzer auf ein mögliches Problem aufmerksam macht. Von dem Kooperationssystem gewarnt, vergleicht T_2 zunächst den Inhalt seiner lokalen Kopie d'_3 mit dem gemeinsamen Dokument d_3 . Er stellt fest, dass sich die gemeinsame Version geändert hat und muss nun sei-

ne Änderungen in das gemeinsame Dokument einpflegen. Da er das Dokument in einem Editor auf seinem Notebook ändert, muss er es wiederum mit der entfernten gemeinsamen Version in dem gemeinsamen Bereich vergleichen. Sind seit dem Öffnen des Dokuments keine Änderungen erfolgt, darf er die konsolidierte Version zurück in das gemeinsame Gruppenareal schreiben.

Eine Besonderheit von Lerngruppen ist, dass aufgrund des losen Kooperationsverbundes die Gruppenstruktur recht dynamisch sein kann. Eventuell scheiden Gruppenmitglieder aus, da sie das Lernziel (z. B. das Bestehen einer Prüfung) verworfen haben oder neue Teilnehmer stoßen zur Gruppe hinzu. In beiden Fällen müssen die Berechtigungen auf die Kooperationsobjekte entsprechend gesetzt werden. Im Falle eines Ausscheidens sind die in der Abschlussphase beschriebenen Schritte für den ausscheidenden Teilnehmer zu vollziehen.

Weiterhin führen mögliche Gäste zu einer kurzfristigen Veränderung der Struktur einer Lerngruppe. Sie nehmen meist einmalig an der Lerngruppe teil, um zusammen mit den Mitgliedern zu lernen oder auch um als Spezialist Wissen an die Gruppe weiterzugeben. Für einmalige Gäste erscheinen die in dem Abschnitt *Gründung, Anbahnungsphase* beschriebenen Mechanismen zur Aufnahme von Mitgliedern überdimensioniert, da Gäste nicht voll in die Gruppe integriert werden müssen und eventuell auch nicht gleichberechtigt zu den Gruppenmitgliedern sind – hier werden flexible Alternativmechanismen benötigt.

Szenario 1.7: Hinzukommen und Ausscheiden von Gruppenmitgliedern

- Ausscheiden und Nachrücken von Gruppenmitgliedern

Aus der Lerngruppe G mit den Teilnehmern T_1 , T_2 und T_3 scheidet das Mitglied T_3 aus, da es an der Prüfung nicht teilnehmen kann. Dennoch würde T_3 gerne für die nächste Prüfungsphase auf die Lernergebnisse zurückgreifen. Da T_3 nicht länger Mitglied in der Gruppe G ist, wird T_3 in Zukunft von den Kommunikationsprozessen ausgeschlossen und darf auch nicht mehr auf den gemeinsamen Gruppenbereich zugreifen, Materialien in diesem ändern oder diese kommentieren. Die Teilnehmer T_1 und T_2 gewähren T_3 jedoch explizit weiterhin Leserechte auf den gemeinsamen Bereich.

Teilnehmer T_3 möchte aber lieber eine Momentaufnahme des Lernfortschritts zum Zeitpunkt des Ausscheidens haben. Er fertigt dazu eine Kopie des Gruppenbereichs im persönlichen Bereich an.

Die möglichen Lernpartner T_4 und T_5 , die in der Gründungsphase (siehe Szenario 1.1 und Szenario 1.2) ebenfalls Interesse an einer Teilnahme an der Lerngruppe G zeigten, wurden in eine Warteliste eingefügt, aus der nun ein neues Gruppenmitglied nachrücken kann. T_1 und T_2 entscheiden sich für die Kandidatin T_4 . T_4 kann nun ihrerseits Materialien in den gemeinsamen Bereich einbringen und wie zuvor T_3 als gleichberechtigtes Mitglied am Kooperationsprozess teilnehmen.

Szenario 1.8: Treffen mit Gastteilnehmer

- Gleichberechtigter Gastbenutzer mit vollem Zugriffsrecht auf den Gruppenbereich

Die Lerngruppe G mit den Teilnehmern T_1, T_2 und T_3 trifft sich zu einem gemeinsam ausgehandelten Termin (siehe Szenario 1.4). Zu ihnen gesellt sich noch T_4 als Gastteilnehmer hinzu. T_4 hat bereits im Vorjahr erfolgreich eine Prüfung über den Lernstoff abgelegt und will der Gruppe beim Lernen helfen. Dazu bekommt T_4 einmalig Zugriffsrechte auf den gemeinsamen Arbeitsbereich der Lerngruppe. Somit ist T_4 gleichberechtigt zu den anderen Teilnehmern. – Es wäre alternativ auch nur ein lesender Zugriff auf den Gruppenbereich möglich. – T_4 kann somit Materialien aus seinem persönlichen Bestand zu dem Gruppenfundus hinzufügen oder die Materialien der Gruppe ändern. Nach Beendigung der Lernsitzung werden diese Rechte automatisch wieder entfernt.

Auflösung, Abschlussphase Lerngruppen existieren meist nur über einen festgelegten Zeitraum. Mit dem Erreichen des Lernziels kann die Lerngruppe oftmals aufgelöst werden. Die Lernenden haben jedoch häufig den Wunsch, die Ergebnisse für einen späteren Zeitpunkt zu archivieren. Eine Weitergabe der Lernergebnisse an andere Lerngruppen ist ebenfalls übliche Praxis. Die Archivierung der Lernergebnisse kann zum einen durch ein Aufrechterhalten der Gruppe geschehen, um später in den Gruppenbereich zurückkehren zu können, oder zum anderen durch das Speichern einer Momentaufnahme des Gruppenbereichs in den persönlichen Arbeitsbereich der Teilnehmer gewährleistet werden.

Die Alternative des Aufrechterhaltens aller je existierenden Gruppen ließe die Gruppenstruktur unübersichtlich werden, da diese kontinuierlich mit jeder neuen Lerngruppe wächst. Auf diese Art geht schnell die Übersicht verloren. Wird der Bereich der Gruppen auf die Geräte ihrer Teilnehmer repliziert, ergibt sich hieraus zusätzlich ein stetig wachsender Speicherplatzbedarf auf den betroffenen Geräten.

Da die Teilnehmer mobile Geräte mit begrenzter (und teurer) Speicherkapazität nutzen, ist darauf zu achten, dass das Archiv platzeffizient angelegt wird. Eine weitere Möglichkeit besteht darin, das Archiv auf einen externen Dienst auszulagern, um später wieder darauf zugreifen zu können. Natürlich muss der Dienst in diesem Fall stets erreichbar sein, sonst steht das Archiv in mancher mobilen Situation nicht zur Verfügung.

Szenario 1.9: Auflösen der Lerngruppe

- Auflösen einer Gruppe
- Archivieren von Gruppenbereich und Gruppenstruktur

- Überführen der mobilen Gruppe in ein zentralisiertes CSCW-System
- Reaktivierung der Gruppe für eine Wiederaufnahme der mobilen Kooperation

Nach der Prüfung wird die Lerngruppe G mit den Mitgliedern T_1 , T_2 und T_4 aufgelöst. Dies bedeutet, dass die Organisationsstruktur der Gruppe und die gemeinsamen Kommunikationskanäle nicht mehr benutzt werden sollen. Die Gruppe an sich kann somit gelöscht werden. Da T_1 , T_2 und T_3 aber beim Lernen Freundschaft geschlossen haben, beschließen sie, die Gruppe für eine nicht zweckgebundene Kommunikation zu nutzen, um weiterhin über das Kooperationssystem miteinander in Kontakt treten zu können.

Die Teilnehmer T_1 und T_2 übernehmen darüber hinaus die Daten aus dem Gruppenbereich als Archiv in ihren persönlichen Fundus auf und speichern diese auf ihrem persönlichen mobilen Computer. So können sie auch in Zukunft auf die Lernergebnisse zurückgreifen. T_1 leitet sein Archiv an den Benutzer T_4 weiter, der nicht Mitglied der Lerngruppe war, aber in Zukunft einen ähnlichen Lernstoff zu bewältigen hat.

Da ein solches Archiv wertvollen Speicherplatz auf dem mobilen Gerät von Teilnehmerin T_2 belegt, speichert diese es auf einem zentralen CSCW-Server. Dort können auch T_1 und T_2 in Zukunft auf dieses Archiv zugreifen, da die Gruppenstruktur samt Berechtigungen auf Wunsch ebenso auf den Server übertragen werden. Für den Fall einer Folgeveranstaltung überträgt T_2 also gleich die gesamte Gruppenstruktur mit auf den Server, um die Gruppe bei Bedarf zu reaktivieren. Als im darauf folgenden Jahr eine Folgeveranstaltung angeboten wird, übertragen T_1 , T_2 und T_3 die Gruppe zurück in ihre mobile Kooperationsumgebung und können so die Lerngruppe an dem Punkt fortzusetzen, an dem sie die letzte Lerngruppe beendet hatten.

Das hier behandelte Szenario der Lerngruppe mit mobilen Geräten hat einen klaren Fokus auf Lernszenarien. Dennoch lässt es sich auf andere alltägliche Kooperationssituationen übertragen. Besonders im Bereich der angesprochenen spontanen Lernsitzungen kann dieses Szenario um Aspekte des nachfolgenden Szenarios der spontanen Kooperation ergänzt werden.

2.1.2 Spontane Kooperation

Eine von klassischen CSCW Umgebungen nicht unterstützte Variante der Kooperation in Kleingruppen ist die *spontane Kooperation*, der keine Planungsprozesse vorausgehen. Häufig ist diese Form der Kooperation zeitlich begrenzt und dient einem kurzfristigen Ziel. Als Beispiel dient hier die Kooperation zweier Studierender, die sich im Bus treffen, dort über eine Übungsaufgabe unterhalten und daraufhin versuchen, gemeinsam einen Lösungsweg zu finden. Eine solche spontane Kooperation kann auch in eine längerfristige Kooperation übergehen. Entsprechend dem Beispiel könnten die Studierenden z. B. beschließen, eine längerfristige Lerngruppe (vgl. Abschnitt 2.1.1) zu gründen.

Rahmenbedingungen Wie das einführende Beispiel schon andeutet, findet eine spontane Kooperation oft unter unkontrollierten Rahmenbedingungen statt. Voraussetzung ist lediglich, dass sich mindestens zwei Kooperationswillige treffen. Über das technische Umfeld kann somit nichts ausgesagt werden. Für die hier vorgestellten Szenarien wird aber vom Vorhandensein eines persönlichen mobilen Computers mit integrierter Netzwerkschnittstelle ausgegangen. Da in einem spontanen Kooperationszenario nicht von einer vorhandenen Netzwerkinfrastruktur ausgegangen werden kann, muss die Kooperationsumgebung über ein *Ad-Hoc-Netzwerk* kommunizieren können.

Szenario 2.1: *Grundszenario spontaner Kooperation*

- Spontanes Treffen im öffentlichen Raum
- Fehlende Kooperationsinfrastruktur

Das der spontanen Kooperation zugrunde liegende Szenario besteht aus den drei Teilnehmern und Teilnehmerinnen T_1 , T_2 und T_3 , die sich im öffentlichen Raum ohne bestehende Infrastrukturen für eine Kooperationsunterstützung treffen. Dieses Treffen ist zufällig und nicht geplant. Alle Beteiligten verfügen über mobile Computer mit zueinander kompatiblen drahtlosen Netzwerkschnittstellen, die es erlauben, ein Ad-Hoc-Netzwerk zu etablieren. Dabei entsteht ein gemeinsames Netzwerk, das den mobilen Geräten der Teilnehmer eine Kommunikation untereinander erlaubt. Die Kooperationsumgebung nutzt diese Kommunikationsstruktur für die Koordination der Kooperation und den Datenaustausch.

Gründung, Anbahnungsphase Spontane Kooperation integriert sich üblicherweise nahtlos in den Alltag der Kooperierenden, ohne dass sie einer der Beteiligten in voraus geplant hat. Oft wird den Teilnehmern der eigentliche Kooperationsakt zunächst nicht einmal bewusst. Aus diesem Grund kann auch die entsprechende formale Kooperationsgruppe nicht bewusst gegründet werden. Sie entsteht vielmehr aus dem Kooperationskontext. Jeder der Anwesenden bei einem Treffen ist potentieller Kooperationspartner und kann während des Verlaufs der Kooperationssitzung in die Kooperationsgruppe ein- oder wieder austreten.

Die Kooperationspartner treffen sich also in alltäglichen Situationen und starten den Kooperationsprozess spontan. Die Kooperation selbst erfolgt entweder aufgrund des günstigen Moments oder weil sich während des Treffens unvorhergesehen ein gemeinsames Kooperationsziel ergibt. Da die Kooperation nicht geplant ist, stehen nur in der Umgebung vorhandene Hilfsmittel zur Verfügung. Dies sind insbesondere die mitgeführten mobilen Computer, die zu diesem Zweck genutzt werden, um eine entsprechende Kooperationsumgebung bereitzustellen.

Die Kooperationsunterstützung muss sich nahtlos in den Kooperationsprozess einbetten und ohne eine aufwändige Konfiguration seitens der Kooperierenden bereit-

stehen. Im Idealfall aktivieren die Teilnehmer der Kooperation ihre mobilen Geräte und gründen in der ad hoc bereitstehenden computergestützten Kooperationsumgebung beiläufig eine Kooperationsgruppe mit dem dazugehörigen gemeinsamen virtuellen Wissensraum.

Szenario 2.2: Erstellen einer vertraulichen Projektskizze während einer Zugreise – Gründung

- Gruppengründung mit Anwesenden ohne bestehende Netzwerkinfrastruktur
- spontanes Erstellen einer temporären Kooperationsgruppe

Drei Arbeitskollegen T_1 , T_2 und T_3 , die der gleichen Organisation angehören, unternehmen gemeinsam eine längere Zugreise. Sie sitzen zusammen in einem Abteil und können sich so vis-a-vis unterhalten. Während des Gesprächs entsteht die Idee für ein neues gemeinsames Projekt. Aufgrund des günstigen Zeitpunktes und der verfügbaren gemeinsamen Zeit schlägt Teilnehmer T_1 vor, das neue Projekt direkt zu skizzieren und in die Wege zu leiten. Zu diesem Zweck soll spontan eine gemeinsamer Projektplan erarbeitet werden.

Die beiden Kollegen stimmen zu und aktivieren ihre mitgeführten Notebooks, die mittels eines Ad-Hoc-Netzwerkes vernetzt werden. Anschließend starten sie die virtuelle Kooperationsumgebung die ihnen einen gemeinsamen Arbeitsbereich und die benötigten Unterstützungsfunktionen zur Verfügung stellt. Da die Projektskizze einen vertraulichen Status besitzt, begrenzt Teilnehmer T_1 die Gruppe in der Kooperationsumgebung mittels Selektion über die verfügbaren Benutzer auf sich und die zwei anwesenden Kollegen T_2 und T_3 . So entsteht eine temporäre Gruppe G^* . Andere Benutzer, die ebenfalls im Zug sitzen und mit der Kooperationsumgebung verbunden sind, können nicht auf den gemeinsamen Arbeitsbereich der drei Kollegen zugreifen.

Da Teilnehmer T_2 bereits eine erste Rohskizze in der Datei d_1 vorliegen hat, stellt er diese in dem gemeinsamen Arbeitsbereich zur Verfügung, indem er sie per *Drag & Drop* dort hinzieht. Der Arbeitsbereich wird dabei standardmäßig auf allen beteiligten Geräten identisch dargestellt. Hinzu kommen von Teilnehmer T_1 noch ein Strategiepapier (d_2) der Organisationsleitung, in die das Projekt eingeordnet werden muss. Alle drei Teilnehmer der Gruppe G^* haben nun gleichberechtigten Zugriff auf die gemeinsamen Dokumente und dürfen sie sowohl lesen als auch verändern.

■ Fortsetzung in Szenario 2.4

Szenario 2.3: Bearbeiten einer Übungsaufgabe in einer Vorlesungspause – Gründung

- Gründung einer spontanen Kooperationsgruppe in einer existierenden Netzwerkinfrastruktur
- Nutzung des Ortsbezugs für die Gruppengründung

Drei Studierende der Informatik T_1 , T_2 und T_3 befinden sich in einem Kooperationsszenario ähnlich Szenario 2.1. Sie treffen sich zwischen zwei Vorlesungen in einem Bistro der Universität. Teilnehmerin T_1 fragt die beiden anderen Studierenden T_2 und T_3 , ob diese bereits eine ihnen aufgetragene Matheaufgabe gelöst haben. Die Studierenden verneinen dieses, auch wenn Teilnehmer T_2 bereits eine erste Idee für einen Lösungsansatz auf einigen Zetteln notiert hat.

Die technische Ausstattung der Teilnehmer ist wie folgt: Teilnehmerin T_1 besitzt einen PDA, Teilnehmer T_2 ein Notebook und Teilnehmer T_3 einen PenPC. Neben diesen technischen Geräten existieren noch ein Fachbuch und besagte handschriftliche Notizen aus dem Besitz von Teilnehmer T_2 . In den persönlichen Beständen befinden sich bei Teilnehmerin T_1 eine digitale Version einer Aufgabenstellung mit Dateinamen d_1 und bei Teilnehmer T_2 zusätzlich die Anleitung zur Lösung ähnlicher Aufgabenstellungen mit Dateinamen d_2 – Teilnehmer T_3 besitzt zu Beginn des Treffens keine eigenen Unterlagen.

Sie etablieren mittels ihrer mobilen Computer eine spontan vernetzte Kooperationsumgebung. Dazu nutzen sie das verfügbare Funknetzwerk der Universität. Sofort werden alle verfügbaren Kooperationspartner angezeigt. Da sie sich im Funknetz der Universität befinden, kontaktiert die Kooperationsumgebung alle anderen verfügbaren Kooperationsanwendungen im Netz der Universität und findet so eine große Anzahl potentieller Kooperationspartner. Teilnehmerin T_1 beschränkt die möglichen Teilnehmer auf Personen in ihrer Nähe. Aus dieser übersichtlichen Auswahl bildet sie nun mit ihren zwei Kommilitonen eine temporäre Kooperationsgruppe G^* . Somit verfügen alle drei über einen abgeschlossenen gemeinsamen Arbeitsbereich, in den sie ihre elektronisch verfügbaren Materialien einbringen, indem sie diese aus ihrem privaten Fundus dort ablegen.

Das Fachbuch und die Skizze des Lösungsansatzes von Teilnehmer T_1 liegen auf dem gemeinsam genutzten Tisch neben den Computern.

■ Fortsetzung in Szenario 2.5

Kooperationsphase Sobald die Infrastruktur für die Zusammenarbeit etabliert ist, nutzen die Teilnehmer deren kooperationsunterstützenden Funktionen. Bei einer Kooperation basierend auf gemeinsam genutzten Dokumenten, dienen diese Funktionen dem Austauschen, kooperativen Bearbeiten und Kommentieren der gemeinsamen Dokumente. Die Dokumente kommen dabei aus dem privaten Fundus der Kooperationspartner, wo sie oftmals bereits in digitalisierter Form vorliegen. Nachdem die Dokumente in dem gemeinsamen Gruppenbereich abgelegt wurden, stehen sie allen Kooperationspartnern zur Verfügung. Wie in dem Szenario der Lerngruppe (vgl. Abschnitt 2.1.1) sind alle Kooperationspartner gleichberechtigt bezüglich der

Nutzung der gemeinsamen Dokumente. Eine Sperrung eines Dokuments in dem gemeinsamen Arbeitsbereich für den Zugriff durch die Kooperationspartner ist nur in Ausnahmen sinnvoll.

Eine Besonderheit der spontanen Kooperation ist die häufige Beschränkung ihrer Dauer auf eine einzige Sitzung. Es sind nur wenige gemeinsame Objekte involviert und der Arbeitsaufwand, diese zu strukturieren, ist für eine so kurzfristige Kooperation zu hoch. Daher wird seitens der Nutzer wenig Zeit für die Strukturierung des Wissensraums aufgebracht, und der Wissensraum muss sich gegebenenfalls automatisch strukturieren. Verschachtelte Strukturen, wie Ordner, sind in diesem Zusammenhang z. B. weniger effizient als semantische Karten, die nebenläufig beim Ablegen der Objekte entstehen. Sollen die Ergebnisse einer spontanen und kurzfristigen Kooperation weiterverwendet werden, müssen diese in persistente Kooperationssstrukturen überführt werden.

Die Kooperationspartner stehen bei dieser Kooperationsform im direkten Blickkontakt zueinander und benötigen somit keine virtuellen Kommunikationskanäle wie synchrone oder asynchrone Nachrichten. Die Kommunikation geschieht intuitiv über direkte Sprache und Gesten. Dies ermöglicht eine sehr natürliche Interaktion zwischen den Kooperationspartnern. Üblicherweise werden in solch ein Kooperationsszenario auch analoge Hilfsmittel, wie Notizzettel und Bücher, integriert. Mit diesen externen Dokumenten muss die computergestützte Kooperationsumgebung harmonieren. Dies bedeutet nicht, dass sie in der Lage sein muss, diese Dokumente in ihre virtuelle Welt zu überführen, sondern vielmehr, dass sie parallel zu den klassischen Hilfsmitteln die Kooperation um computergestützte Methoden ergänzt, ohne die aus der Nutzung klassischer Hilfsmittel entstehenden Medienbrüche weiter zu verstärken. Das bedeutet auch, die Nutzungskonzepte an die der klassischen Hilfsmittel anzupassen. Zur Verdeutlichung sei hier als negatives Beispiel der Wechsel zwischen dem Schreiben mit dem Stift auf den Notizzetteln und dem Schreiben mit einer Tastatur auf einem daneben platzierten Notebook angeführt.

Ein mehr oder weniger bewusster Wechsel von den klassischen Kooperationshilfsmitteln in Richtung einer CSCW-Umgebung wird nur geschehen, wenn für die Benutzer ein Mehrwert entsteht und alle Kooperationsdokumente ebenfalls in der computergestützten Umgebung verfügbar sind. Natürlich sind auch Umgebungen mit einer erweiterten Realität (*Augmented Reality*) denkbar, wenngleich diese in alltäglichen Situationen aufgrund der hohen technischen Anforderungen noch nicht realisierbar erscheinen. In den behandelten Szenarien wird diese Möglichkeit daher außer Acht gelassen.

Da die Kooperation in einem gemeinsamen Gruppenbereich spontan erfolgt, beinhaltet diese Szenarien entsprechend wenig Annahmen über die konkreten Rahmenbedingungen oder den Ablauf der Kooperationsprozesse. Die Kooperation erfolgt frei und freiwillig, dementsprechend wenig restriktiv darf die Kooperationsumgebung in ihren Unterstützungsfunktionen sein.

Szenario 2.4: Erstellen einer vertraulichen Projektskizze während einer Zugreise – Kooperationsphase

■ Fortsetzung von Szenario 2.2

- Bearbeiten von Dokumenten in der spontan errichteten Kooperationsumgebung
- Wechsel zwischen kooperativer und individueller Arbeit

Die drei Kollegen ($G^* = \{T_1, T_2, T_3\}$) sichten zunächst die vorhandenen Materialien (d_1, d_2) und diskutieren den Entwurf in d_1 von Teilnehmer T_2 . In einem ersten Schritt ordnen sie den Entwurf in das Strategiepapier d_2 der Organisationsleitung ein und Teilnehmer T_3 versieht die entsprechende Stelle in d_2 mit einer Annotation. T_2 hat noch eine Anmerkung dazu und kommentiert seinerseits die Annotation.

Da die Struktur des Entwurfs d_1 bei der Kollegin T_1 und dem Kollegen T_3 nicht auf Zustimmung stößt, entschließen sie sich, den Entwurf in seine einzelnen Punkte zu zerlegen, um einige Punkte zu erweitern und anschliessend neu zu gliedern. Dazu zerlegen sie das Dokumente in viele kleinere Dokumente ($d_{1,1}, \dots, d_{1,n}$), die die einzelnen Gliederungspunkte der Skizze enthalten. Sie bewerten jeder für sich die einzelnen Punkte und entscheiden, welche Punkte entfernt oder hinzugefügt werden müssen. Danach strukturieren sie kooperativ die einzelnen Dokumente als semantische Karte, in die jedes Dokument eingeordnet wird. Um eine bereits erfolgte Strukturierung durch die anderen Kollegen nicht unbewusst zu verändern, werden die Sichten aller Beteiligten auf die semantische Karte synchronisiert.

Am Ende der Sitzung hat sich eine Gliederung der Punkte etabliert, deren Hauptabschnitte paritätisch an je einen Kollegen weitergegeben wird. Dieser Abschnitt wird von jedem der drei Teilnehmer zunächst individuell weiterbearbeitet.

■ Fortsetzung in Szenario 2.6

Szenario 2.5: Bearbeiten einer Übungsaufgabe in einer Vorlesungspause – Kooperationsphase

■ Fortsetzung von Szenario 2.3

- Bearbeiten und Zusammenführen persönlicher Versionen eines Dokumentes

Die drei Studierenden ($G^* = \{T_1, T_2, T_3\}$) entscheiden sich, die Kooperation exklusiv in der computergestützten Kooperationsumgebung fortzusetzen, um Medienbrüche durch die Vermischung von herkömmlichen und digitalen Dokumenten zu vermeiden. Dazu müssen sie zunächst die auf den Notizzetteln niedergeschriebenen Lösungsansätze in ein digitales Dokument übertragen. Ein einfaches Abbild der Notizen reicht ihnen allerdings nicht aus, da sie das Dokument im Nachhinein bearbeiten möchten und Teile ihrer Lösung zur Weiterverarbeitung an

ein Algebrasystem übergeben wollen, das auf dem Notebook von Teilnehmerin T_3 installiert ist.

Daher übertragen sie die Notizen von Hand und mit Hilfe einer speziellen Software auf Zeichenebene in das Zieldokument d_3 . Innerhalb des gemeinsamen Arbeitsbereiches ist das Dokument für alle drei Teilnehmer sichtbar, und sie können es dort gemeinsam bearbeiten. Um ein möglichst breites Spektrum von Lösungswegen zu explorieren, beschließen sie, die Lösung der Aufgabe zunächst getrennt, jeder für sich, fortzuführen. Sie erstellen dazu persönliche Versionen d_3^1 , d_3^2 und d_3^3 des Dokumentes d_3 . Teilnehmerin T_1 bearbeitet Dokument d_3^1 , Teilnehmer T_2 Dokument d_3^2 und Teilnehmerin T_3 Dokument d_3^3 .

Nach der individuellen Arbeitsphase führen sie die persönlichen Versionen wieder zusammen zu einem Dokument. Zunächst bewerten sie die Teildokumente und verwerfen Dokument d_3^1 , da der enthaltene Lösungsweg nicht zum gewünschten Ergebnis führt. Der Ansatz von Dokument d_3^2 erscheint am vielversprechendsten und soll um eine Teillösung aus d_3^3 ergänzt werden. Das so entstehende Dokument ersetzt die alte Lösungsskizze d_3 .

Dieser Vorgang der individuellen Lösungsüberarbeitung und des Zusammenführens der resultierenden Lösungsdokumente wird im Laufe der Sitzung iterativ wiederholt bis ein zufrieden stellendes Lösungsdokument entstanden ist – oder die Kooperationssitzung aus einem anderen Grund endet.

■ *Fortsetzung in Szenario 2.7*

Auflösung, Abschlussphase Da die Gruppengründung in spontanen Kooperationszenarien nicht geplant erfolgt und eher eine nebenläufige Handlung ist, erfolgt auch die Beendigung der Kooperation eher unbewusst. Der entstandene virtuelle Wissensraum und die zugehörige Kooperationsgruppe bestehen exakt so lange wie die Kooperationssitzung andauert. Sie werden zum Ende der Kooperationssitzung ebenso beiläufig aufgelöst, wie sie entstanden sind. Ähnlich verhält es sich mit dem gemeinsamen Datenbestand.

Aufgrund der Begrenzung der spontanen Kooperation auf eine einzige Sitzung, existiert zu Beginn kein langfristiges Kooperationsziel. Oftmals möchten die Kooperierenden dennoch am Ende der Sitzung die Kooperation fortführen. Die zunächst spontane und kurzfristige Kooperation kann dann in eine Kooperationsform überführt werden, die über einen längeren Zeitraum bis zum Erreichen des Kooperationsziels fortgesetzt wird. Dazu kann die temporäre Kooperationsgruppe in eine langfristige mobile Kooperationsstruktur mit einer etablierten Benutzer- und Rechtestruktur überführt werden oder sogar in ein klassisches CSCW-System transportiert werden.

Selbst wenn eine Fortsetzung der Kooperation nicht gewünscht ist, liegt es häufig im Interesse der Teilnehmer, zumindest Teilergebnisse der Kooperation in ihren persönlichen Bestand zu übernehmen. Aufgrund des gleichberechtigten Charakters spontaner Kooperation sind hierzu alle beteiligten Kooperationspartner berechtigt.

Die Kooperationsergebnisse können dazu innerhalb der Kooperationsumgebung oder auf einem anderen Speichermedium für den persönlichen Gebrauch abgelegt werden. Alternativ können sie auch direkt in eine andere Kooperation eingebracht werden. Natürlich können die Kooperationsergebnisse auch die Grundlage für eine spätere noch nicht geplante spontane Kooperation mit denselben Teilnehmern dienen. Da eine solche später spontane Kooperationssitzung aber nicht voraussehbar ist, scheint es oft nicht sinnvoll, die Gruppenstruktur und den Datenbestand aufrecht zu erhalten, ohne zu wissen, ob eine solche Konstellation je wieder eintritt.

Szenario 2.6: Erstellen einer vertraulichen Projektskizze während einer Zugreise – Abschlussphase

■ Fortsetzung von Szenario 2.4

- Beenden der Kooperationssitzung
- Archivieren der Kooperationsstruktur

Die Teilnehmerin T_3 der Gruppe ($G^* = \{T_1, T_2, T_3\}$) verlässt den Zug einige Stationen vor den Kollegen und beendet daher die Verbindung zur Kooperationsumgebung. Weil die temporäre Kooperationsgruppe G^* mit den verbleibenden zwei Mitgliedern zunächst fortbesteht überführt sie eine Kopie der Arbeitsergebnisse zu dem Zeitpunkt des Verlassens der Kooperationssitzung in ihren persönlichen Arbeitsbereich, der sich auf ihrem Notebook befindet, und nimmt diese für eine spätere Verwendung mit.

Die Teilnehmer T_1 und T_2 setzen die Kooperation nach dem Ausscheiden der Teilnehmerin T_3 fort. Nach Erreichen des Zielbahnhofs beenden sie die Kooperationssitzung. Teilnehmer T_2 verlässt ebenfalls die Kooperationssitzung und hat wie Teilnehmerin T_3 zuvor die Möglichkeit, eine Kopie der Arbeitsergebnisse mitzunehmen. Teilnehmerin T_1 hat nun beim Beenden der Kooperationssitzung folgende Optionen:

- Verwerfen der Gruppenstruktur und der Kooperationsergebnisse.
- Archivieren der Kooperationsergebnisse im persönlichen Arbeitsbereich.
- Überführen der Kooperationsgruppe inklusive Arbeitsbereich in eine permanente Kooperationsgruppe innerhalb der Kooperationsumgebung.
- Übertragen der Kooperationsgruppe inklusive Arbeitsbereich auf einen CSCW-Server, um eine höhere Verfügbarkeit für alle Beteiligten zu erreichen.

Um die Kooperation zu einem späteren Zeitpunkt fortführen zu können, bewahrt Teilnehmerin T_1 die Gruppenstruktur und Kooperationsergebnisse, indem sie die temporäre Kooperationsgruppe und deren Arbeitsbereich in eine permanente Gruppe überführt. Zu diesem Zweck ordnet sie als letzte Handlung der Kooperationsitzung die Gruppe in die bereits existierende Gruppenstruktur der Kooperationsumgebung ein.

Szenario 2.7: Bearbeiten einer Übungsaufgabe in einer Vorlesungspause – Abschlussphase

■ Fortsetzung von Szenario 2.5

- Überführen einer spontanen Kooperationsgruppe in eine persistente Gruppe

Die temporäre Gruppe von Studierenden ($G^* = \{T_1, T_2, T_3\}$) hat gemeinsam einen Lösungsansatz d_3 für die Aufgabe in Dokument d_1 erarbeitet und beendet nun die Kooperationssitzung. Da eine gemeinsame Abgabe der Lösung für Übungsaufgaben bei der entsprechenden Vorlesung nicht vorgesehen ist, übernehmen die Studierenden die Teillösungen in ihre persönlichen Lösungsdokumente (d_3^1, d_3^2, d_3^3), die später elektronisch oder in Papierform an den Dozenten der Veranstaltung übermittelt werden müssen, um sich Zusatzpunkte für die ausstehende Klausur zu verdienen.

Die Studierenden beschließen aufgrund ihrer positiven Kooperationserfahrungen aus der einmaligen spontanen Kooperationssitzung heraus, eine permanente Lerngruppe (vgl. Szenario in Abschnitt 2.1.1) zu gründen. Zu diesem Zweck überführen sie die temporäre Gruppe $G^* = \{T_1, T_2, T_3\}$ in eine permanente Gruppe $G = \{T_1, T_2, T_3\}$ auf dem CSCW-Server des Instituts und ordnen diese in dessen Gruppenstruktur ein. Der zugehörige gemeinsame Arbeitsbereich wird ebenfalls dauerhaft auf dem CSCW-Server gespeichert und somit den Studierenden T_1 , T_2 und T_3 langfristig verfügbar gemacht.

Die in diesem Abschnitt behandelten Szenarien der „*Lerngruppe*“ (2.1.1) und der „*Spontane Kooperation*“ (2.1.2) setzen stets die Anwesenheit der beteiligten Benutzer voraus. In dem nächsten Abschnitt werden in Kontrast dazu räumlich verteilte Benutzer in eine Kooperation einbezogen.

2.2 Kooperation mit räumlich verteilten Teilnehmern

Eine der Hauptaufgaben klassischer CSCW-Systeme ist die Unterstützung der Kooperation räumlich getrennter Personen. Diese Systeme haben die Zielsetzung, die räumliche Trennung zu überbrücken, indem sie geeignete *Kommunikationsmechanismen* bereitstellen, spezielle *Kooperationswerkzeuge* zur Verfügung stellen und einen *gemeinsamen Arbeitsbereich* (engl. *shared workspace*) bereitstellen. Die Erforschung und Entwicklung von Kooperationswerkzeugen und -konzepten in den letzten Jahrzehnten hat zu einigen ausgereiften Lösungen für verschiedenste Szenarien räumlich getrennter Kooperation geführt. Die meisten Lösungen wurden gemäß dem vorherrschenden Client-Server-Paradigma als zentral verwaltete Kooperationsumgebungen umgesetzt. Da viele dieser Systeme inzwischen weit über einen prototypischen Charakter hinaus sind, werden sie in Bildungseinrichtungen und der Industrie produktiv eingesetzt. Im privaten Umfeld sind sie hingegen selten zu finden, was an den Tatsachen liegen mag, dass aufgrund der zentralisierten Architektur eine *Abhängigkeit von ständigen Kommunikationsverbindungen* besteht, diese Systeme nur auf *speziellen*

Servern laufen, die ständig in Betrieb sein müssen, und außerdem durch Spezialisten konfiguriert und gewartet werden müssen.

Die in Kapitel 2.1 angesprochenen Szenarien der Präsenzkooperation sind eine wichtige Komponente um computergestützte Kooperationen nahtlos in den Alltag der Menschen zu integrieren. Hier besteht seitens der klassischen CSCW-Systeme eine Lücke, die es zu schließen gilt. Wie die Szenarien der Präsenzkooperation zeigen, steht der Nutzung klassischer CSCW-Systeme für derartige Szenarien zumindest deren zentralisierte Architektur im Wege. Neuere Ansätze versuchen daher die Kooperationsumgebung auf alle beteiligten Computer zu verteilen.

Dieses neue Architekturprinzip im Bereich des CSCW bedingt aber zugleich wesentliche Veränderungen für die Bereitstellung von Mechanismen zur Kooperationsunterstützung. Insbesondere die technische Umsetzung wird stark von den Mechanismen abweichen, die derzeit in zentralisierten CSCW-Systemen zu finden sind. An diesen technischen Möglichkeiten müssen sich auch die Konzepte der mobil-verteilten Kooperation orientieren. Insbesondere erschließt die Berücksichtigung der Mobilität der Nutzer in einer verteilten Architektur völlig neue Perspektiven für die Kooperationsunterstützung in CSCW-Umgebungen. Diese neuen Perspektiven deuteten sich zum Teil bereits in den Szenarien der Präsenzkooperation an. Die folgenden Szenarien dienen daher der Betrachtung der neuen Aspekte einer räumlich getrennten Zusammenarbeit in mobil-verteilten Kooperationsumgebungen.

Bei einer kombinierten Unterstützung von Präsenzkooperation und räumlich verteilter Kooperation ist es wichtig, die Benutzer nicht zu nötigen, die Kooperationsumgebung zu wechseln, weil sie von einer Form der Kooperation in die andere übergehen. Der optimale Architekturansatz muss daher ein möglichst breites Spektrum von Kooperationszenarien unterstützen. Sollte ein Wechsel zwischen unterschiedlichen Kooperationsumgebungen sinnvoll oder gar unvermeidlich erscheinen, benötigen die Kooperationsumgebungen nahtlose Übergänge, um den Benutzer nicht mit vermeidbaren technischen Hürden zu konfrontieren.

In den folgenden Szenarien wird eine Kooperationskultur beschrieben, die nur entstehen kann, wenn Kooperation immer und überall möglich und die Nutzung der Kooperationsumgebung unabhängig des aktuellen Aufenthaltsorts der Benutzer ist. Diesbezüglich wäre eine allgegenwärtige Netzwerkverbindung zum Internet wünschenswert. Da diese aber auch in Zukunft nicht immer und vor allem oft nicht kostengünstig möglich ist, wird die Kooperationsumgebung robust gegen Verbindungsabbrüche zum Netzwerk zu gestalten sein. Dies betrifft sowohl die Unterstützung von individuellem Arbeiten im gemeinsamen virtuellen Wissensraum, wie auch die Möglichkeit zur Präsenzkooperation ohne Zugang zum Internet. Sollte eine Kooperationsgemeinschaft durch einen Verbindungsabbruch im Netzwerk voneinander getrennt werden, ist zudem bei wiederkehrender Netzwerkverbindung der zeitweise getrennte Wissensraum wieder zu vereinigen.

Die Szenarien räumlich getrennter Kooperation sind eine klassische Domäne der gängigen Client-Server-basierten CSCW-Systeme. Um aber einen nahtlosen Übergang zwischen den Szenarien aus Abschnitt 2.1 (*Präsenzkooperation*) und 2.2 (*Kooperation mit räumlich verteilten Teilnehmern*) zu erreichen, ist es wichtig, diese Szena-

rien in der neuartigen verteilten CSCW-Umgebung ebenfalls zu unterstützen. Ergänzend zu einer Verschmelzung von Präsenzkooperation und räumlich getrennter Kooperation in einer mobil-verteilten Kooperationsumgebung wird in Abschnitt 2.2.2 (*Distant Learning*) eine stärkere Verknüpfung von Client-Server-basierten und mobil-verteilten Kooperationssystemen diskutiert. Der folgende Abschnitt betrachtet aber zunächst die Unterstützung räumlich getrennter Benutzer in einer mobil-verteilten CSCW-Umgebung.

2.2.1 Räumlich getrennte Kooperation

Räumlich getrennte Kooperation teilt sich einige wesentliche Merkmale mit der Präsenzkooperation aus Kapitel 2.1. Sie zeichnet sich aber dadurch aus, dass zumindest ein Teil der Mitglieder einer Kooperationsgruppe sich nie oder nur selten gemeinsam an einem Ort aufhalten. In dieser Tatsache liegen einige besondere Anforderungen an die Mechanismen zur Kooperationsunterstützung begründet. Während bei der Präsenzkooperation Mechanismen zur nahtlosen Einbettung der computergestützten Kooperationsumgebung in „natürliche“ Kooperationsszenarien im Vordergrund standen, sind es bei der räumlich getrennten Kooperation Mechanismen, die es erlauben, Teilnehmer in die Kooperation einzubinden, die während der Kooperationssitzung nicht körperlich zugegen sind.

Gründung, Anbahnungsphase In einer verteilten Kooperationsumgebung kann die Gründung einer Kooperationsgruppe über Mechanismen der Gruppenfindung für räumlich getrennte Teilnehmer geschehen (vgl. auch Szenario 1.2) oder die Gruppe aus einer zuvor etablierten Kooperationsform übernommen werden. Dabei spielt es keine Rolle ob die Kooperation zuvor in einem klassischen CSCW-System oder einem mobil-verteilten CSCW-System stattgefunden hat.

Im Fall eines mobil-verteilten CSCW-Systems erfolgt die Gruppengründung analog zu der Anbahnungsphase in Abschnitt 2.1.1 und 2.2.1. In diesem Fall können die Gruppenstrukturen beibehalten werden und auch die Kooperationsbereiche müssen nicht weiter angepasst werden. Erst in der Kooperationsphase selber werden speziell angepasste Mechanismen für eine örtlich getrennte Kooperation benötigt.

Soll eine Gruppe von einem klassischen serverbasierten CSCW-System auf ein mobil-verteiltes CSCW-System überführt werden, so muss die Gruppe mit allen enthaltenen Benutzern und zugehörigen Arbeitsbereichen in das mobil-verteilte CSCW-System übertragen werden. Der Gruppenarbeitsraum muss gemäß der Szenarien in Abschnitt 2.1 auf die Geräte aller Teilnehmer verteilt werden und der persönliche Arbeitsbereich der Benutzer muss auf deren Geräte übertragen werden. Soll der Server auch weiterhin genutzt werden können, so verbleiben die Daten zunächst auf diesem gespeichert. Bezuglich des persönlichen Arbeitsbereiches bedeutet dieses Vorgehen, dass von diesem nun mindestens zwei Versionen existieren (eine auf dem Server und eine in der mobil-verteilten Umgebung auf dem mobilen Gerät des Benutzers). Bei dem Gruppenbereich sind sogar deutlich mehr Versionen betroffen (mindestens je

eine Version auf jedem Rechner der Gruppenmitglieder und zusätzlich eine Version auf dem Server).

Szenario 3.1: Überführen einer zentral verwalteten Kooperationsgruppe in eine mobil-verteilte Kooperationsumgebung

- Überführen einer Kooperationsgruppe von einem CSCW-Server in ein mobil-verteiltes CSCW-System
- Verteilen des gemeinsamen Arbeitsbereiches der Gruppe auf die beteiligten Geräte

Eine in einem CSCW-Server existierende Gruppe auf mit drei Teilnehmern ($G = \{T_1, T_2, T_3\}$) soll für eine Kooperation mittels mobiler Geräte in eine mobil-verteilte Kooperationsumgebung überführt werden. Die Gruppe G besitzt auf dem Server einen gemeinsamen Arbeitsbereich mit diversen Dokumenten ($A = \{d_1, \dots, d_n\}$).

Da nicht alle Gruppenmitglieder während der Kooperationssitzungen stets zugänglich sind, kann der Arbeitsbereich A nicht zentral auf einem mobilen Gerät der Gruppenteilnehmer gespeichert werden. Daher wird auf jedem beteiligten Mobilcomputer eine Kopie des gemeinsamen Arbeitsbereichs abgelegt. Ausgehend von einem Mobilcomputer je Teilnehmer entstehen drei Kopien des Arbeitsbereichs und der enthaltenen Dokumente ($A^{T_1} = \{d_1^{T_1}, \dots, d_n^{T_1}\}$, $A^{T_2} = \{d_1^{T_2}, \dots, d_n^{T_2}\}$, $A^{T_3} = \{d_1^{T_3}, \dots, d_n^{T_3}\}$). Diese Kopien werden bei jedem Kontakt der Teilnehmer untereinander abgeglichen.

Soll die Gruppe nicht aus einem anderen System oder einer anderen Kooperationsform übernommen werden, so bedarf es einer Möglichkeit, diese Gruppe ohne persönliches Treffen zu gründen. Dies geschieht analog zu dem Vorgehen in zentralisierten CSCW-Systemen: Ein zukünftiges Gruppenmitglied muss zunächst eine neue Gruppe erstellen. Damit existieren eine Gruppe mit einem Mitglied (dem Benutzer, der die Gruppe erstellt hat) und der dazugehörige Gruppenbereich. Zu dieser Gruppe kann der gründende Benutzer (und später auch alle weiteren berechtigten Gruppenmitglieder) Einladungen an ihm bekannte Benutzer verschicken. Beantworten diese die Einladung positiv, gehören sie von nun an zu der Gruppe. Alternativ kann die Gruppe auch über ein Verteilungsverfahren oder entsprechende Verzeichnisse öffentlich bekannt gemacht werden (vgl. auch Szenario 1.2). Interessierte Benutzer können dann eine Mitgliedschaft beantragen, die daraufhin von den berechtigten Gruppenmitgliedern entschieden wird.

Kooperationsphase Da die Teilnehmer einer räumlich getrennten Kooperationsgruppe über unterschiedliche Arten einer Netzwerkanbindung verfügen, kann es notwendig sein, die Verteilung der Kooperationsobjekte ebenfalls unterschiedlich vorzunehmen. Der Abgleich verteilt gespeicherter Objekte kann je nach Verbindungsgüte

lange dauern oder die genutzten Verbindungen können zu instabil sein, um einen entfernten Objektzugriff sinnvoll durchzuführen.

Szenario 3.2: Entfernte Teilnahme an einer verteilten Kooperation über eine sporadische Funkverbindung

- Verteiltes Kooperieren über unzuverlässige Netzwerkverbindungen

Eine Gruppe mit drei örtlich getrennten Mitgliedern ($G = \{T_1, T_2, T_3\}$) arbeitet in einer gemeinsamen Arbeitsumgebung A . Teilnehmerin T_1 reist mit der Bahn über Land und nimmt über eine GSM-Verbindung an der Kooperationssitzung teil. Die Übertragung des gesamten Inhalts der Arbeitsumgebung A dauert über die Verbindung mindestens 5 Minuten. In unregelmäßigen Abständen bricht die Verbindung zu dem GSM-Netz ab. In dem Fall eines Verbindungsabbruchs ist keine Kommunikation mit den anderen Teilnehmern mehr möglich. Bis eine neue GSM Verbindung aufgebaut ist können zwischenzeitlich mehrere Minuten vergehen.

Obwohl die Verbindung bei einem bevorstehenden Abbruch schlechter wird und dieser somit in etwa vorausgesagt werden kann, reicht die Zeit nicht, um die noch offenen Änderungen in dem verteilten Arbeitsbereich komplett zurückzuschreiben und die geänderte Kopie des gesamten Arbeitsbereiches A^{T_1} an alle Teilnehmer zu verteilen. Während der bestehenden Verbindung werden daher stetig die Veränderungen ΔA^{T_1} innerhalb des gemeinsamen Arbeitsbereiches A übertragen und in die Kopien A^{T_2} und A^{T_3} der Teilnehmer T_2 und T_3 eingepflegt. Bei einem Verbindungsabbruch werden die Änderungen gesammelt und im Nachhinein an T_2 und T_3 verteilt. Die Änderungen seitens T_2 und T_3 werden auf dem gleichen Weg bei der Kopie A^{T_1} von T_1 eingepflegt. Ziel des ständigen Abgleichs ist es alle verteilten Kopien des gemeinsamen Arbeitsbereiches A identisch zu halten ($A = A^{T_1} = A^{T_2} = A^{T_3}$).

Besteht eine Verbindung, sehen alle Teilnehmer mittels einer gemeinsamen synchronisierten Sicht auf den Arbeitsbereich A , an welchem Teil der enthaltenen Dokumente die anderen Teilnehmer derzeit arbeiten.

Während einer Verbindungsunterbrechung arbeiten nun Teilnehmerin T_1 und Teilnehmer T_2 an demselben Dokument $d \in A$ ohne voneinander zu wissen. Bei dem Versuch der Umgebung, die Änderungen ΔA_1^T von T_1 beim nächsten Verbindungsauftakt bei dem synchronisierten Arbeitsbereich A einzupflegen, würde die Umgebung die Änderungen von T_2 in dem zwischen T_2 und T_3 synchronisierten Dokument d überschreiben. Im Gegenzug würden die Änderungen von Teilnehmer T_2 die Änderung von T_1 in deren Kopie von d überschreiben. Die Folge wären zwei Versionen (d' und d'') des Dokumentes d , die nicht länger synchron sind ($d' = d^{T_1} \neq d^{T_2} = d^{T_3} = d''$). Da nur jeweilige Änderungen seitens der Teilnehmer übertragen werden, würde auch kein weiterer Abgleich vorgenommen. Es bliebe bei den asynchronen Kopien, deren Inhalt mit weiteren Änderungen noch weiter divergieren würde.

Um dieses Dilemma zu vermeiden, überprüft die Kooperationsumgebung vor dem Abgleich zweier Kopien eines verteilten Objektes erst, ob diese zwischen dem letzten

Abgleich von mehr als einem Teilnehmer geändert wurden. Sollte dies der Fall sein, werden die beteiligten Teilnehmer (T_1 und T_2) aufgefordert, den Konflikt vor der endgültigen Synchronisierung zu lösen. Erst dieses konsistente Objekt wird dann an alle Beteiligten verteilt. So bleibt der gemeinsame Arbeitsbereich konsistent.

Bei einer Kooperation über große Entfernungen hinweg ergeben sich häufig stark asynchrone Kooperationsszenarien. Da die Teilnehmer bei asynchronen Kooperationsformen die Veränderung im gemeinsamen Arbeitsbereich nicht zeitgleich mitverfolgen können, müssen Informationen über Änderungen dementsprechend aufbereitet werden. Dies kann z. B. durch Zusammenfassungen oder automatisch generierte Annotationen geschehen.

Auch die Kommunikation erfolgt in diesen Kooperationsszenarien asynchron. Dabei wird die asynchrone Kommunikation, auch wegen ihres häufig hierarchischen und weniger flüchtigen Charakters, der synchronen Kommunikation vorgezogen. Diskussionsforen und ausfeilte Annotationsmechanismen, die mit einzelnen Kooperationsobjekten verknüpft werden, helfen darüber hinaus, eine dokumentenzentrierte Kommunikation zu etablieren, ohne jedes Mal explizit auf das betroffene Objekt Bezug nehmen zu müssen. Die Diskussion wird ähnlich wie ein digitaler Notizzettel an das Dokument geheftet, so dass alle Kooperationsmitglieder direkt den Zusammenhang zwischen Notiz und Objekt erkennen können.

Szenario 3.3: Zeitversetzte Kooperation in verteilten CSCW-Anwendungen

- Zeitlich stark versetzte Kommunikation
- Asynchroner Datenaustausch über Mitteldienste

Zwei über mehrere Zeitzonen hinweg entfernte Personen, T_1 und T_2 , wollen mittels eines mobil-verteilten Kooperationssystems eine Softwarearchitektur für eine Warenverwaltung entwerfen. Anders als bei einem Client-Server-System läuft das verteil arbeitende CSCW-System gleichberechtigt auf den Computern beider Teilnehmer. Innerhalb der Kooperationsumgebung gründen sie eine Gruppe $G = \{T_1, T_2\}$ mit zugehörigem gemeinsamen Arbeitsbereich A . Dieser wird auf den beteiligten Computern synchronisiert gespeichert, so dass jeder Teilnehmer stets die aktuellen der Kooperation zu Grunde liegenden Daten im Zugriff hat. Wird also von Teilnehmerin T_1 die Datei d in den gemeinsamen Arbeitsbereich A eingebracht, wird diese zunächst in der lokalen Instanz der Kooperationsumgebung gespeichert und dann zum nächstmöglichen Zeitpunkt auf die entfernte Instanz der Kooperationsumgebung von Benutzer T_2 übertragen.

Da beide Teilnehmer aufgrund der Zeitverschiebung nur selten zur selben Zeit Arbeiten, besteht die Gefahr, dass die CSCW-Anwendungen ebenfalls nur selten zur selben Zeit laufen. Dies stünde dem regelmäßigen Abgleich des gemeinsamen Arbeitsbereiches im Wege. Die Anwendung muss daher in der Lage sein, die Gegenseite

bei Bedarf kontaktieren zu können. Startet Teilnehmerin T_1 für eine Arbeitssitzung die CSCW-Anwendung, so kontaktiert diese die ihr bekannte Gegenstelle von Teilnehmer T_2 und „weckt“ diese bei Bedarf, um eventuell anstehende Änderungen zu übertragen. Danach kann die Anwendung auf dem Computer von Teilnehmer T_2 wieder beendet werden. Sind beide Anwendungen zur gleichen Zeit aktiv, schicken sie sich ständig die Änderungen innerhalb des gemeinsamen Arbeitsbereichs.

Teilnehmer T_2 muss seinen Computer jedoch häufiger vom Netzwerk trennen, da er ihn mit auf Dienstreisen nimmt. Daher versendet er die Änderungen, die er vorgenommen hat, per asynchroner Kommunikationsdienste (z. B. E-Mail) an die Anwendung von Teilnehmerin T_1 oder hinterlegt sie bei einer weiteren im Netzwerk verfügbaren Anwendungsinstanz. Sobald die Anwendung auf dem Computer von T_1 aktiv wird, empfängt diese die Nachricht mit den Änderungen und integriert diese in die lokale Version der gemeinsamen Arbeitsumgebung und beginnt mit der Sitzung.

Auflösung, Abschlussphase Bei der Beendigung einer räumlich getrennten Kooperationsgemeinschaft sind die Vorgänge ähnlich zu denen der *Präsenzkooperation* (vgl. Abschnitt 2.1). Je nach Wunsch der Teilnehmer geht es darum, die Kooperationsergebnisse zu archivieren und für eine weitere Verwendung aufzuarbeiten, oder die gesamte Kooperationsgemeinschaft in eine andere Kooperationsform zu überführen.

Bei einer zeitlich versetzten Kooperation muss darauf geachtet werden, dass bei Abschluss der Kooperation alle Instanzen der verteilten Kooperationsumgebung synchronisiert sind. Alle Teilnehmer müssen also über die Beendigung der Kooperation benachrichtigt werden und ihre Daten ein letztes Mal abgleichen. Danach ist eine konsistente Veränderung des *gemeinsamen* Arbeitsbereiches nicht mehr möglich. Anschließende Veränderungen resultieren in einer persönlichen Version des Arbeitsbereiches, die von dem gemeinsamen Arbeitsbereich entkoppelt ist. Bei einer Überführung in eine andere Kooperationsform gilt das Gleiche. Erst in einem konsistenten Zustand darf die Kooperationsumgebung transformiert werden.

Szenario 3.4: Beenden einer verteilten und räumlich getrennten Kooperation

- Beenden einer zeitlich und räumlich getrennten Kooperationsgruppe
- Gründen einer neuen Kooperationsgruppe aus einer beendeten Gruppe

Eine Kooperationsgruppe mit drei Mitgliedern ($G = \{T_1, T_2, T_3\}$) beendet eine Kooperation mittels einer mobil-verteilten Kooperationsumgebung nach einer längeren Phase der Zusammenarbeit, weil das gemeinsam bearbeitete Projekt beendet ist.

Die Projektleiterin T_1 „friert“ zu diesem Zweck den gemeinsamen Arbeitsbereich A mit den enthaltenen Dokumenten d_1 und d_2 in seinem jetzigen Zustand ein und archiviert ihn auf ihrem lokalen Rechner, um später den Projektbericht erstellen zu können. Das „Einfrieren“ des Arbeitsbereiches wird T_2 und T_3 beim nächsten Kontakt zu T_1 über die Synchronisation des Arbeitsbereichs bekannt gegeben.

T_2 hatte aber zwischenzeitlich noch Änderungen in den Dokumenten d_1 und d_2 im Arbeitsbereich vorgenommen. Die Änderungen an d_1 fallen zeitlich vor den Entzug des Schreibrechts auf den gemeinsamen Arbeitsbereich und die an d_2 zeitlich danach. Daher werden die Änderungen an d_1 bei der nächsten Synchronisation mit T_1 und T_3 noch vor dem Durchführen der Zugriffsänderungen von T_1 ausgeführt. Die Modifikationen Δd_2 , die nach der Zugriffsänderung liegen, werden jedoch abgelehnt, da die entsprechenden Rechte fehlen.

Die Kooperationsumgebung bietet daher an, ein neues Dokument $d'_2 = d_2 + \Delta d_2$ zu erzeugen, das die Daten des alten Dokuments plus die Änderungen enthält. Da das neue Dokument d'_2 nicht innerhalb des gesperrten gemeinsamen Arbeitsbereiches abgelegt werden kann, hat T_2 die Optionen, das Dokument entweder in seinem persönlichen Bereich abzulegen oder den Arbeitsbereich zu kopieren und auf der Kopie weiterzuarbeiten. T_2 entscheidet sich für die letztere Option und erstellt eine Kopie A' von A . Diese enthält anstatt des alten Dokuments d_2 das geänderte d'_2 ($A' = \{d_1, d'_2\}$). Auf diesem Areal beschließt T_2 seine Kooperation mit T_3 in einer neuen Kooperationsgruppe $G' = \{T_2, T_3\}$ fortzuführen, um ein Folgeprojekt zu initiieren und weist daher G' den Gruppenarbeitsbereich A' zu.

2.2.2 Distant Learning

Eine populäre Variante der räumlich getrennten Kooperation ist das *Distant Learning* aus dem Bildungsbereich. Dieses ermöglicht Lernenden an einem Kurs teilzunehmen, ohne körperlich an der Bildungseinrichtung anwesend zu sein. Für eine Teilnahme müssen die Lernenden unter Umständen nicht einmal gleichzeitig mit der Lernumgebung verbunden sein. Die genaue Gestaltung des Kurses hängt beim Distant Learning von den zugrunde liegenden didaktischen Konzepten ab. Diese sollen im Folgenden aber keine Rolle spielen.

In einfachen Formen des Distant Learning werden lediglich Lernmaterialien und Übungsaufgaben über ein E-Learning-System bereitgestellt. Oft müssen Übungsaufgaben bis zu einem bestimmten Zeitpunkt gelöst und zur Kontrolle des Lernfortschritts an den Lehrenden gesandt werden. Diese Kontrolle kann auch durch automatisierte Mechanismen übernommen werden. Für Rückfragen zum Lernstoff bestehen eventuell Kommunikationskanäle, um die Betreuer zu kontaktieren. Oft bieten die E-Learning-Systeme einfache Formen der Kooperationsunterstützung wie Mailinglisten oder Foren an. Um den Zugang zu den Materialien und den Diensten zu erleichtern, werden diese meist in einer Webseite gebündelt.

Trotz bidirektionaler Kommunikationsmechanismen wie Forum und Mailingliste ist ein solches Lernen vom Konsumieren der Lerninhalte und von einer oft einsei-

tigen Kommunikation vom Lehrenden zum Lernenden geprägt. Eine Kooperation zwischen den Lernenden erfolgt oft nur außerhalb des E-Learning-Systems, da sie nicht ausreichend unterstützt wird. Aber auch in Lernumgebungen, die Kooperationen zwischen den Lernenden ausdrücklich unterstützen, werden häufig nicht in dem möglichen Maße für eine Kooperation genutzt.

Dies mag zum einen an einer mangelnden kulturellen Einbettung der computergestützten Kooperation bei den Lernenden liegen. Darum wird diese oft beim Aufbau der Kurse gefördert [vgl. Hampel et al., 2003; Keil-Slawik und Hampel, 2003]. Zum anderen mag aber auch ein Grund darin bestehen, dass die computergestützten Kooperationsumgebungen nicht überall verfügbar und nicht in den Alltag der Lernenden eingebettet sind. Diese Lücke können mobil-verteilte Kooperationssysteme schließen. In diesem Abschnitt soll daher der Einsatz von mobil-verteilten Kooperationssystemen in dem Bereich des kooperativen Distant Learning im Mix mit der Präsenzkooperation betrachtet werden.

Gründung, Anbahnungsphase Die Kurse, die einem Distant Learning Szenario zu Grunde liegen, werden stets von Benutzern in der Rolle des Lehrenden angeboten. Diese treten somit als Experten und Dienstleister auf, die eine Wissensvermittlung in Form eines aufbereiteten Kurses an eine Gruppe Lernender anbieten. Im Folgenden wird ein mögliches Lernszenario skizziert, ohne dabei ein didaktisches Konzept präsentieren zu wollen.

Im Rahmen einer Lehrveranstaltung richten die Lehrenden einen gemeinsamen Wissensraum mit den aufbereiteten Kursunterlagen ein, der in mehrere Lerneinheiten aufgeteilt ist. Zu jeder Lerneinheit gehören vertiefende Übungen. Um ein vorgreifendes Lernen zu verhindern, können diese Lerneinheiten nach und nach in den gemeinsamen Wissensraum eingebracht werden.

Auf die Kursmaterialien dürfen die Kursteilnehmer nur lesend zugreifen, während die Betreuer auch Schreibrechte haben. Für die freie Kooperation der Kursteilnehmer besteht ein Bereich der keinerlei Beschränkungen unterlegen ist. Die Betreuer besitzen neben dem gemeinsamen Wissensraum mit den Lernenden einen eigenen gemeinsamen Bereich, in dem die Koordination der Lehre erfolgen kann.

Der Lehrbereich besitzt zudem eine Außendarstellung des Kurses in Form einer Webseite, die auch durch Nichtmitglieder des Kurses betrachtet werden kann. Hier sind z. B. das Kursprofil und die Zugangsbedingungen beschrieben. Außerdem kann hier eine Anmeldung für den Kurs erfolgen. Die Anzahl der möglichen Anmeldungen ist oft beschränkt, so dass ab einer bestimmten Anzahl von Teilnehmern an dem Kurs keine Anmeldungen mehr möglich sind.

Nach der Anmeldung befinden sich alle Teilnehmer des Kurses in einer gemeinsamen Gruppe. Bei einer großen Anzahl von Teilnehmern werden die einzelnen Teilnehmer eventuell zusätzlich in eine weitere Struktur von Kleingruppen eingeordnet. Diese Unterteilung kann u. a. notwendig werden, weil eine thematische Zuordnung vorgenommen werden soll oder nur begrenzte Platzressourcen bei Präsenzübungen vorhanden sind.

Szenario 4.1: Kurs mit zeitweise örtlich verteilten Teilnehmern – Gründung

- Seminar mit regelmäßigen Präsenzphasen
- Gruppenstruktur mit Untergruppen und gemeinsamen wie getrennten Arbeitsbereichen

Für ein Präsenzseminar mit neun Teilnehmern (T_1, \dots, T_9) sollen deren mobile Computer genutzt werden, um eine *Computer Supported Cooperative Learning (CSCL)*-Umgebung zur Verfügung zu stellen. Es sind regelmäßige Präsenzphasen vorgesehen. Der Dozent D stellt die Lehrunterlagen in einem auf seinem mobilen Rechner erstellten virtuellen gemeinsamen Arbeitsbereich A bereit. Für die Zugriffskontrolle erstellt er eine Gruppe G , die auf den gemeinsamen Arbeitsbereich zugreifen darf. In diese Gruppe nimmt er alle beim ersten Seminartermin anwesenden Teilnehmer auf.

Die Gruppenmitglieder erhalten dadurch lesenden Zugriff auf die zur Verfügung gestellten Lehrmaterialien d_1, \dots, d_n . Außerdem existiert ein Bereich, in dem die Gruppenmitglieder frei kooperieren können, also Schreib- und Leserechte besitzen.

Der gemeinsame Arbeitsbereich des Seminars wird auf alle mobilen Geräte verteilt. Geplant sind Gruppenarbeiten in Gruppen zu je drei Teilnehmern. Hierzu erzeugt der Dozent drei Untergruppen in der Gruppe G ($G = G_1 \cup G_2 \cup G_3$). Die einzelnen Gruppen dürfen nicht auf die Gruppenbereiche der jeweils anderen Gruppen zugreifen. Zum Austausch zwischen den Gruppen soll der gemeinsame Kooperationsbereich A des Seminars genutzt werden. Die Gruppenarbeit erfolgt zwischen den Präsenzphasen.

■ *Fortsetzung in Szenario 4.3*

Szenario 4.2: Zentral verwalteter Kurs mit örtlich getrennten Teilnehmern – Gründung

- Mobile Arbeitsbereiche in Verbindung mit einem CSCW-Server

Ein Dozent D an einer deutschen Hochschule gibt einen Kurs für deutsche und chinesische Studierende des Maschinenbaus. Die Veranstaltung soll den chinesischen Studierenden eine Kursteilnahme ermöglichen, ohne dass sich diese in Deutschland aufhalten müssen. Zu diesem Zweck stellte D sowohl sein Vorlesungsskript d_1 als auch die Präsentationsfolien d_2 über einen gemeinsamen Arbeitsbereich A_{Lehre} zur Verfügung. Dieser ist nur für Mitglieder der Gruppe G zugänglich.

Ein externer Besucher bekommt statt des gemeinsamen Arbeitsbereichs eine Webseite mit Veranstaltungsbeschreibung und Anmeldeinformationen zu sehen.

Der gemeinsame Arbeitsbereich wird über ein CSCW-System der Universität bereit gestellt. Studierende, die an dem Kurs teilnehmen möchten, melden sich über die Webseite an und werden daraufhin der Gruppe G hinzugefügt. Von nun an haben diese Studierenden Zugriff auf die Lehrmaterialien.

Um die Kooperation der Studierenden zu fördern, sind sie zusätzlich Mitglieder der Gruppe G_{Stud} und erhalten Zugriff auf den zugehörigen gemeinsamen Arbeitsbereich A_{Stud} , in dem sie sich auch außerhalb des Lehrbetriebs virtuell treffen und austauschen können. Dieser Bereich bildet zusammen mit dem Bereich A_{Lehre} mit den Lehrmaterialien den gemeinsamen Arbeitsbereich A der Vorlesung. Dieser dient den Studierenden als zentrale Anlaufstelle für die Vorlesung.

Um die mobile Kooperation zu unterstützen, können die Lernenden den gemeinsamen Arbeitsbereich A des Kurses auf ihrem mobilen Computer „mitnehmen“ und auch ohne Verbindung zum CSCW-Server der Universität nutzen, um mit anderen Studierenden des Kurses zu kooperieren.

■ *Fortsetzung in Szenario 4.4*

Kooperationsphase Nach der Gründung eines Kurses und der Anmeldung aller Mitglieder bleibt die Gruppenstruktur in der Regel stabil. Die Lehrenden stellen die Materialien in Lerneinheiten geordnet nach und nach in den gemeinsamen Arbeitsbereich. Dieser Vorgang wird in der Präsenzlehre meist von Vorlesungen begleitet, in denen die Lehrenden durch den Lehrstoff führen und bei Bedarf Anmerkungen und Ergänzungen anbringen. Beim reinen Distant Learning entfällt die Vorlesung, die Teilnehmer arbeiten den Stoff selbstständig auf. Ergänzungen sind oft gebündelt mit dem Skript für die Lerneinheiten. Manchmal wird auch versucht, die fehlende Präsenzvorlesung durch eine Video- oder Audioübertragung derselben zu kompensieren.

Ergänzt wird jede Lerneinheit durch Übungen, in denen die Lernenden das Gelernte anwenden und vertiefen. Diese Übungen erfolgen häufig in Kleingruppen. Den Lernenden steht zur Bewältigung der Übungsaufgaben ein Tutor zur Seite, der die Übungen betreut und für Rückfragen zur Verfügung steht. In einer räumlich getrennten Variante der Übung muss es dementsprechende Kommunikations- und Koordinationsmittel geben.

Die Übungsaufgaben lassen sich auf diverse Arten beschreiben. Die einfachste Form ist eine informelle Beschreibung der Aufgabe in Schriftform. Diese kann einfach als Dokument in dem gemeinsamen Wissensraum abgelegt werden. Die Lernenden senden ihre Lösungen der Aufgaben an die Betreuer oder legen sie in einem speziellen Bereich des Wissensraums ab, auf den nur die Betreuer Zugriff haben. Soll die erfolgreiche Teilnahme an dem Kurs von diesen Lösungen abhängen, sind die Berechtigungen für die Lösungsdokumente so zu vergeben, dass ein Abschreiben oder nachträgliches Ändern von Lösungen verhindert wird.

Übungsaufgaben können aber auch so gestellt werden, dass die Korrektheit der Lösungen durch die Lernumgebung ausgewertet werden kann (z.B. Multiple Choice

Aufgaben). Dies entlastet die Betreuer und bietet zusätzlich die Möglichkeit einer direkten Rückmeldung an die Lernenden. Eine weitere Möglichkeit ist, die Übungsaufgaben so zu beschreiben, dass sie personalisiert werden können. Dann werden sie für jeden Lernenden individuell erstellt. Dies geschieht über Generatoren oder Templates. Diese verwenden entweder einen Zufallswert oder einmaliges Attribut der Benutzer (z.B. die Identifikationsnummer), um die Aufgaben speziell für sie zu erstellen. Ein beispielhaftes Szenario einer automatischen Erzeugung und Bewertung von personalisierten Übungsaufgaben mit Hilfe eines *Computer Algebra Systems (CAS)* beschreibt [Bleckmann et al., 2005].

Bei Rückfragen zu einer Aufgabe oder zu einem Dokument beziehen sich die Lernenden stets auf diese Objekte, daher wird die Kommunikation an das betreffende Objekt geknüpft. Die Kommunikation referenziert so stets implizit das betroffene Objekt. Bei einer Kooperation mit herkömmlichen Schriftdokumenten würde dazu das betroffene Dokument in den Mittelpunkt gerückt. In einer computergestützten Kooperationsumgebung verhält es sich ähnlich. Kommunikation kann entweder in Form eines an ein Dokument geknüpften Diskussionsstranges oder, in anderer Richtung, in Form eines Verweises auf das Dokument aus der Kommunikation mit diesem verknüpft werden.

Szenario 4.3: Kurs mit zeitweise örtlich verteilten Teilnehmern – Kooperationsphase

■ Fortsetzung von Szenario 4.1

- Synchronisation von gemeinsamen Arbeitsbereichen einer Gruppe
- Synchronisation gemäß der Zugangsrechte zu einem Bereich
- Lösung von Versionskonflikten nach Trennungsphase

Die Teilnehmer des Seminars aus Szenario 4.1 treffen sich einmal in der Woche, um unter Anleitung des Dozenten D die Ergebnisse der Gruppenarbeiten zu diskutieren und den Gegenstand des Seminars thematisch zu vertiefen. Zwischen diesen Präsenzsitzenungen treffen sich die drei Gruppen G_1 , G_2 und G_3 jeweils untereinander und bearbeiten die ihnen gestellten Aufgaben. Zu diesem Zweck verbinden sie ihre mobilen Computer und arbeiten auf dem gemeinsamen Gruppenbereich (A_{G_1}, \dots, A_{G_2}) im Arbeitsbereich A des Seminars.

Um selbstständig zu Hause arbeiten zu können, befindet sich auf jedem der mobilen Geräte eine identische Kopie des Seminarbereiches. Haben in der Zwischenzeit ein oder mehrere Gruppenmitglieder in dem Bereich Änderungen vorgenommen, so muss der Gruppenbereich zu Beginn der Gruppensitzung abgeglichen werden. Bei einander widersprechenden Änderungen seitens zweier oder mehr Mitgliedern muss dieser Konflikt aufgelöst werden.

Teilnehmer T_1 und T_2 aus Gruppe G_1 haben jeweils zu Hause das Dokument d_1 bearbeitet. Es existieren somit die zwei Versionen $d_1^{T_1}$ und $d_1^{T_2}$, die miteinander im Konflikt stehen. Sie werden zu Beginn der Kooperationssitzung auf diesen

Konflikt aufmerksam gemacht und aufgefordert, ihn zu beheben. Als Hilfsmittel bekommen sie beide Versionen nebeneinander dargestellt. Nachdem sie den Konflikt aufgelöst haben, wird mit der konfliktbereinigten Datei das alte Dokument d_1 im gemeinsamen Gruppenbereich A_{G_1} überschrieben.

Während und am Ende der Kooperationssitzung ist der gemeinsam genutzte Bereich auf allen mobilen Computern der Gruppe identisch, da die Instanzen auf den mobilen Computern der Teilnehmer ständig synchronisiert werden.

Da das Gruppenmitglied T_3 bei der gemeinsamen Sitzung von Gruppe G_1 fehlte, ist dessen Version des gemeinsamen Arbeitsbereiches A_{G_1} veraltet. Als Teilnehmer T_3 Teilnehmer T_2 nach der Kooperationssitzung in der Universität trifft, kann T_3 diese jedoch mittels der aktuellen Instanz von T_2 aktualisieren.

Die Kooperation erfolgt meist über direkte Kommunikation zwischen den Gruppenmitgliedern während gemeinsamer Treffen. Gleichzeitig werden die Objekte im gemeinsamen Arbeitsbereich in der virtuellen Kooperationsumgebung manipuliert. In einem allen Seminarteilnehmern zugänglichen Bereich A_{public} werden die Arbeitsergebnisse für das nächste Seminartreffen aufbereitet und zugänglich gemacht. Jede Gruppe legt ihre Arbeitsergebnisse in diesem öffentlich zugänglichen Bereich ab. Gruppe G_1 erstellt eine Kopie d'_2 der Datei d_2 aus ihrem Gruppenbereich A_{G_1} und legt diese in dem Bereich A_{public} ab. Daraufhin wird die Kopie d'_2 für alle Seminarteilnehmer zugänglich und in deren lokalen Instanzen des gemeinsamen Arbeitsbereiches synchronisiert. Dies trifft nicht für die Originaldatei im gemeinsamen Arbeitsbereich A_{G_1} von Gruppe G_1 zu, welche nur zwischen deren Mitglieder synchronisiert wird.

Trifft sich das gesamte Seminar, werden alle gemeinsam einsehbaren Bereiche in Einklang gebracht. Dies geschieht wiederum analog zu dem Abgleichen zu Beginn eines Gruppentreffens. Jeder mit einem anderen Benutzer gemeinsam einsehbare Bereich muss dabei berücksichtigt werden. Für jeden Teilnehmer heißt dies folgende Bereiche abzulegen: den eigenen Gruppenbereich (A_{G_x}) mit den anderen Gruppenmitgliedern (G_x); den öffentlichen Gruppenbereich A_{public} mit allen Teilnehmern (T_1, \dots, T_n) und dem Dozenten D ; den Kooperationsbereich der Studierenden mit den anderen Teilnehmern (T_1, \dots, T_n); und den Seminarbereich mit dem Dozenten D (da nur dieser ihn verändern darf). Dies bedeutet, dass alle Bereiche, die ein Benutzer einsehen kann, mit gleichberechtigten Benutzern abgeglichen werden müssen. Während des Treffens bleiben diese Bereiche wiederum im Einklang.

■ *Fortsetzung in Szenario 4.5*

Szenario 4.4: Zentral verwalteter Kurs mit örtlich getrennten Teilnehmern – Kooperationsphase

■ *Fortsetzung von Szenario 4.2*

- Mischung von zentralen und mobil-verteilten Kooperationssystemen

- Workflow-Elemente in Kooperationsprozessen
- Integration externer Dienste in die Kooperationsumgebung

Da ein gemeinsames Treffen der Studierenden und des Dozenten aus Szenario 4.2 aufgrund der großen Distanz nicht möglich ist und wegen der Zeitverschiebung auch eine zeitliche Übereinstimmung nur schwer zu finden ist, wird die Veranstaltung nicht wie eine Vorlesung im klassischen Sinne geführt. Die Studierenden erhalten wöchentlich eine neue Lerneinheit in Form eines kommentierten Vorlesungsskriptes. Des Weiteren gibt es zu jeder Lerneinheit eine Art Videovorlesung des Dozenten, in der er die Vorlesung vor einer laufenden Kamera hält. Auch diese Videoaufzeichnung wird den Lernenden zur Verfügung gestellt. Die Studierenden erarbeiten sich anhand dieser Materialien die Lerninhalte selbstständig.

Zusätzlich zu den Lerneinheiten werden stets Übungsaufgaben ausgegeben, die in Kleingruppen von zwei bis vier Studierenden zu bearbeiten sind. Dazu nutzen die Gruppen die mobil-verteilte Lernumgebung. Sie bilden selbstständig eine Gruppe G_x mit zugehörigen Gruppenbereich A_{G_x} . Diese Gruppen existieren nun sowohl auf dem CSCW-Server als auch in der mobil-verteilten Lernumgebung, da die Gruppen zwischen ihr und dem CSCW-Server synchronisiert werden.

Um eine Übungsaufgabe zu bearbeiten, erzeugen die Teilnehmer T_1 , T_2 und T_3 aus Gruppe G_1 mittels eines Generatorobjektes im Arbeitsbereich der Vorlesung eine personalisierte Übungsaufgabe $a_{(T_1, T_2, T_3)}$, die mittels der Benutzerkennungen auf sie zugeschnitten ist. Sie verschieben die Übungsaufgabe in ihren Arbeitsbereich A_{G_1} ihrer Gruppe G_1 . In einer oder mehreren Gruppensitzung erstellen sie daraufhin eine Lösung $l_{(T_1, T_2, T_3)}$ und übermitteln diese in einen virtuellen Abgabekasten im Arbeitsbereich A der Vorlesung. Nach der Bewertung wird ihnen das Ergebnis zusammen mit einer korrigierten Lösung in ihren Gruppenbereich zurückkopiert.

Zu Beginn der Gruppensitzung von Gruppe G_1 muss der gemeinsame Bereich A_{G_1} zwischen den Gruppenmitgliedern ($G_1 = \{ T_1, T_2, T_3 \}$) abgeglichen werden. Bei einer Verbindung zum CSCW-Server wird dieser in den Abgleich einbezogen, ansonsten geschieht dieser Abgleich erst bei der nächsten Verbindung eines Gruppenmitglieds mit dem Server. Der Server ist derzeit über eine schlechte Funkverbindung an die Computer der Gruppe angebunden, so dass der Abgleich nicht zügig vollzogen werden kann. Da aber nach einer ersten Überprüfung keine Konflikte zwischen den Objekten auf dem Server und denen in dem verteilten gemeinsamen Arbeitsbereich A_{G_1} der Gruppe G_1 zu bestehen scheint, werden die geänderten Objekte nach und nach im Hintergrund übertragen.

Für die Kooperation nutzen die Gruppenmitglieder das schnellere Ad-Hoc-Netzwerk zwischen ihren Mobilcomputern. Die Kooperation wird so nicht von der langsamen Server-Anbindung beeinträchtigt. Auch als nach einiger Zeit die Verbindung zum Server abbricht, hat dies keine direkte Auswirkung auf die Kooperation. Die Gruppenmitglieder nutzen die Lehrunterlagen, die die Teilnehmerin T_1 bei einer vorherigen Verbindung mit dem Server bereits abgeglichen hatte. Diese aktuellste

verfügbare Version besorgen sich die Mitglieder (T_2, T_3) somit von T_1 statt direkt vom Server. Jetzt verfügen auch sie über den neuesten verfügbaren Stand der Lehrunterlagen.

Da der Abgleich mit dem Server nicht abgeschlossen war, muss die Synchronisation bei der nächsten Verbindung wieder aufgenommen werden. Bis dahin bleiben einige Objekte im gemeinsamen Arbeitsbereich nicht benutzbar. Eine Verknüpfung, z. B. die auf ein noch nicht geladenes Dokument in den Lehrunterlagen weist, zeigt derzeit noch ins Leere. Das Zielobjekt war auf dem Server neu in den Arbeitsbereich eingefügt worden (*Semantische Inkonsistenz*).

Am Ende der Gruppensitzung übergeben die Lernenden die Lösung $l_{(T_1, T_2, T_3)}$ in den Abgabekasten, der ebenfalls als lokale Kopie auf ihren mobilen Geräten liegt. Aufgrund der fehlenden Verbindung zum CSCW-Server ist diese Lösung jedoch zunächst nur lokal verfügbar. Schon seit dem Einfügen der Lösung in den Abgabekasten haben die Teilnehmer keinen Zugriff mehr auf die Lösung, um sie eventuell nachträglich zu ändern (*Zugriffsschutz in verteilten Systemen*).

Beim nächsten Kontakt eines Gruppenmitglieds zum Server wird die lokale Kopie des Abgabekastens mit diesem synchronisiert und somit die Lösung $l_{(T_1, T_2, T_3)}$ auf den Server übertragen. Diese kann nun von den Tuto ren korrigiert werden. Eine Aufgabe, die zu bearbeiten war, kann automatisch durch den Computer korrigiert werden. Zu diesem Zweck liegt ein aktives Objekt im Abgaberaum, das die Lösung an ein externes Expertensystem übermittelt. Dieses betrachtet die Lösung, überprüft und bewertet sie und übermittelt das Ergebnis zurück an das aktive Objekt im Abgaberaum. Das Ergebnis wird daraufhin direkt mittels Ablage eines Bewertungsobjektes $b_{(T_1, T_2, T_3)}$ in den Arbeitsbereich A_{G_1} an die Gruppe übermittelt. Eine Kopie der Lösung und der Bewertung verbleiben zusätzlich im Bewertungsraum, um die Bewertung im Nachhinein nachvollziehen zu können.

■ Fortsetzung in Szenario 4.6

Auflösung, Abschlussphase Am Ende eines Kurses steht oftmals eine Erfolgskontrolle in Form einer Prüfung. Diese kann ähnlich wie die Übungsaufgaben gehandhabt werden. Wurden die Übungsaufgaben bewertet, können sie eventuell sogar die Abschlussprüfung ersetzen. Manche qualifizierenden Einrichtungen verlangen für diese Prüfungen eine persönliche Anwesenheit der Prüflinge, um das Prüfungsumfeld besser kontrollieren und eine Manipulation der Prüfungsergebnisse verhindern zu können.

Neuere Ansätze betrachten die Kooperationsergebnisse der Lernenden selbst als eine Art Erfolgskontrolle. Anhand der Artefakte der Kooperation werden das Wissen, die Wissensstruktur und der Grad der Beteiligung seitens der Lernenden bewertet. In dem *Jour Fixe* Konzept [Hampel et al., 2003; Keil-Slawik und Hampel, 2003] wird diese Bewertung mit einem persönlichen Fachgespräch ergänzt. Eventuell muss bei einer Bewertung der Kooperationsergebnisse aus verwaltungsrechtlichen Gründen der gemeinsame Arbeitsbereich der Prüflinge archiviert werden, da er in das

Prüfungsergebnis einfließt und für dessen Nachvollziehung wichtig ist. Er darf somit nicht alleine auf den Geräten der Lernenden verbleiben, sondern muss in einem zum Prüfungstermin „eingefrorenen“ Zustand an einem zentralen und geschützten Ort archiviert werden.

Doch auch die Lernenden haben häufig das Bedürfnis, die Kooperationsergebnisse zu bewahren und für eine spätere Nutzung im Zugriff zu haben. Anders verhält es sich bei den organisatorischen Strukturen wie der Gruppenzugehörigkeit und den Kooperationsbereichen, die nach Ende einer Kooperation meist nicht mehr genutzt werden. Die Lernergebnisse werden in diesem Fall in die persönlichen Bereiche der Lernenden transferiert und die Kooperationsstrukturen anschließend gelöscht. Allerdings bilden sich auch häufig Langzeitkooperationen, bei denen die Gruppenstruktur langlebiger ist als die wechselnden Inhalte der Kooperation.

Szenario 4.5: Kurs mit zeitweise örtlich verteilten Teilnehmern – Auflösung

■ Fortsetzung von Szenario 4.3

- Synchronisierte Präsenzprüfung anhand der Kooperationsergebnisse
- Archivierung der Prüfungsrelevanten Kooperationsergebnisse auf einem externen Dienst
- Archivierung von Objekten inklusive Metadaten für spätere Verwendung

Am Ende des Seminars führt der Dozent D mit jeder der drei Gruppen (G_1, G_2, G_3) ein Fachgespräch, in dem insbesondere auf die veröffentlichten Inhalte der Gruppe eingegangen wird. Jeder der Teilnehmer hat in der Prüfung seinen mobilen Computer vor sich. Zu Beginn gleichen die Geräte der Teilnehmer und des Dozenten den Arbeitsbereich des Seminars untereinander ab. Jeder der Anwesenden hat somit die gleiche Version zur Verfügung. Da das Fachgespräch entlang der Inhalte des Arbeitsbereichs strukturiert wird, werden Objekte, die aktuell besprochen werden, markiert und kommentiert. Die Gesten der Zeigegeräte werden auf allen Geräten zugleich angezeigt.

Der Dozent stellt den Studierenden die Aufgabe, Bilder im Arbeitsbereich, die wichtige Ereignisse in der Geschichte der Entstehung des Computers repräsentieren, auf einem Zeitstrahl anzuhängen. Er gibt dazu ein Ereignis an und die Prüflinge müssen einer nach dem anderen ein Bild anordnen. Dazu verschiebt jeder auf seinem Computer das Bild an eine passende Stelle des Zeitstrahls. Da die Instanzen des Wissensraums während der Sitzung abgeglichen bleiben, erhält jeder Teilnehmer der Sitzung dieselbe Ansicht auf den Wissensraum. Dies erlaubt dem Dozenten, wenn notwendig, unterstützend einzutreten.

Am Ende der Prüfung ist für die Studierenden der Gruppe G_1 das Seminar abgeschlossen. Der öffentliche Bereich wird eingefroren und auf einem Server im Prüfungsamt archiviert. Auch der Dozent nimmt keine Änderung an den Lehrmaterialien mehr vor. Die Studierenden archivieren, jeder für sich, den für sie sichtbaren

Inhalt des Arbeitsbereichs. Dabei werden alle Metadaten der enthaltenen Objekte derart gespeichert, dass es möglich ist, die Kooperationsobjekte in anderen Kooperationssituationen wieder zu verwerten.

Szenario 4.6: Zentral verwalteter Kurs mit örtlich getrennten Teilnehmern – Auflösung

■ Fortsetzung von Szenario 4.4

- Archivieren von Lehrmaterialien für einen zukünftigen Zugriff
- Wechsel zwischen etablierten und spontanen Kooperationsszenarien

Am Ende der Veranstaltung wird der gemeinsame Arbeitsbereich A der Vorlesung noch bis zum Abschluss aller Prüfungen aktiv von den Tuto ren betreut. Die Teilnehmer der Vorlesung können die Inhalte und Kommunikationsmechanismen nutzen, um für die Prüfungen zu lernen und Kontakt zu den Tuto ren aufzunehmen. Nach Beendigung aller Prüfungen endet die aktive Betreuung seitens der Tuto ren. Doch der gemeinsame Arbeitsbereich wird so archiviert, dass alle Studierenden auch weiterhin auf alle Inhalte inklusive ihrer eigenen Kooperationsbereiche zugreifen können. Der Veranstalter garantiert zu diesem Zweck eine 10jährige Verfügbarkeit des Arbeitsbereiches über den CSCW-Server.

Die studentische Gruppe G_1 steht kurz vor ihrer Prüfung und gründet für ein gemeinsames Lernen eine Lerngruppe (siehe Abschnitt 2.1.1). Sie übertragen zu diesem Zweck einige Objekte aus dem Arbeitsbereich A der Vorlesung in ihren spontan erstellten Wissensraum A_{G_1} . Nach der Prüfung löschen die Mitglieder der Lerngruppe ihre lokale Kopie des Arbeitsbereichs der Vorlesung, da dieser auf dem CSCW-Server verfügbar bleibt. Den verteilten Wissensraum archiviert jeder Teilnehmer für sich.

Nach Ende der Veranstaltung sperren die Betreuer den Arbeitsbereich A für einen schreibenden Zugriff, so dass dieser nicht mehr verändert werden kann. Teilnehmer der Veranstaltung dürfen aber weiterhin auf alle Bereiche, die ihnen zugänglich waren, lesend zugreifen. Die für die Studierenden freien Bereiche bleiben für eine Aufrechterhaltung von Kooperationsgemeinschaften kooperativ nutzbar.

2.3 Herausforderungen mobil-verteilter Kooperation

Betrachtet man die in diesem Kapitel vorgestellten Szenarien, zeigt sich ein breites Einsatzspektrum für computergestützte Kooperation im Alltagsleben. Bei einer Durchtrennung der Abhängigkeit von festen und unflexiblen Infrastrukturen ergibt sich automatisch eine neue Qualität der Nutzung von CSCW-Systemen. Dabei scheint ein evolutionäres Vorgehen bei der Transformation zentralisierter Kooperationsumgebungen in eine mobil-verteilte Architektur angeraten, da beide Systeme

mausprägungen, wie aus den Szenarien ersichtlich, ihre Berechtigung haben. Der Verlässlichkeit und generellen Verfügbarkeit eines zentralisierten CSCW-Dienstes steht die mobile und spontane Nutzung einer zwischen gleichberechtigten Benutzern verteilten CSCW-Umgebung entgegen.

Um die Vorteile beider Ausprägungen der Kooperation für die Benutzer zugänglich zu machen, muss es Ziel sein, eine Systemkonvergenz von zentralisierten und mobil-verteilten CSCW-Ansätzen zu erreichen. Nur so kann ein nahtlos verknüpftes Kooperationsumfeld geschaffen werden.

Während viele der technischen Fragestellungen auf Seite der zentralisierten CSCW-Dienste bereits beantwortet scheinen, ergeben sich in der Unterstützung mobil-verteilter Systeme völlig neue Problemstellungen aufgrund der geänderten technischen Perspektive. In der Tat haben sich die benötigten Mechanismen für eine Kooperationsunterstützung kaum verändert – das Konzept des virtuellen Wissensraums mit den zugehörigen primären Medienfunktionen als Beispiel ist aufgrund seiner Flexibilität auch der mobil-spontanen Kooperation gewachsen – aber die Bereitstellung dieser Mechanismen wirft in dem neuen technischen Umfeld völlig neue Fragen für deren Bereitstellung auf.

Die grundlegenden Fragenstellungen scheinen in einer Unterstützung spontaner Gruppengründungen im strukturlosen Umfeld, dem Bereitstellen einer verteilten und dennoch hochverfügbaren persistenten Arbeitsumgebung und der Einbindung verfügbarer externer Dienste in der Umgebung der Benutzer zu sein. Anhand der gebotenen Szenarien werden daher in dem folgenden Kapitel die genauen Anforderungen an eine derartige technische Infrastruktur zur spontanen Vernetzung von Kooperationsgruppen in mobil-verteilten Wissensräumen bestimmt.

Mobilität in der kooperativen Wissensarbeit

3 Anforderungen für eine technische Unterstützung mobiler Wissensorganisation

Architekturkonzepte heutiger Kooperationsumgebungen hängen stark von bestehenden Netzwerkinfrastrukturen ab und verlangen in jeder Phase des Kooperationsprozesses nach einer Verbindung zu einem zentralen Server des CSCW-Systems. Der Server übernimmt hier die Rolle des Koordinators der Kooperation und sorgt außerdem für eine zentrale Speicherung der gemeinsam genutzten Dokumente. Aus den in Kapitel 2 aufgezeigten Szenarien wird jedoch ersichtlich, dass zentralisierte Infrastrukturen aufgrund der technischen Rahmenbedingungen nur bedingt für eine computergestützte Kooperation im mobilen Nutzungsumfeld geeignet sind.

Oftmals sind zu Beginn einer mobilen Kooperation überhaupt keine Netzwerkinfrastrukturen verfügbar. Die Etablierung eines unabhängigen Kommunikationsnetzwerkes ist daher der erste Schritt in Richtung einer mobilen Kooperationsumgebung. Die so neu entstandene Spontanität in der Vernetzung der Kooperationsumgebung lässt eine Abhängigkeit von zentralisierten Kommunikations- und Kooperationssstrukturen nicht länger zu. Zudem entwickelt der gesamte Kooperationsprozess aufgrund ständig wechselnder Kontexte und Nutzungskonstellationen eine hohe Dynamik. Will ein Softwaresystem mobile Kooperationsszenarien unterstützen, muss es daher deren neuen Qualitäten und Anforderungen Rechnung tragen.

Mobile Formen der virtuellen Wissensorganisation stellen somit neue Herausforderungen an das Forschungsfeld des CSCW. Die Dienste dedizierter CSCW-Server werden in den neuen mobilitätsfreundlichen Architekturen von verschiedenen Knoten eines Peer-to-Peer-Netzwerks aller Teilnehmer bereitgestellt und bilden dadurch eine verteilte Kooperationsumgebung. Kleine und große Gruppen mobiler Knoten bilden so spontan Netzwerke, treten existierenden Kooperationsnetzwerken bei und verlassen sie kurz darauf wieder. Die klassische Trennung zwischen Dienstanbieter (Server) und Dienstnehmer (Client) wird aufgelöst und durch eine Peer-to-Peer-Infrastruktur ersetzt. Der technische Terminus *Peer* erhält in mobilen Kooperationsszenarien eine zusätzliche Bedeutung für eine *spezielle Form der gleichberechtigten Zusammenarbeit* durch die gegenseitige Bereitstellung von Dienstleistungen und Ressourcen.

Eng verbunden mit derartigen flexiblen Mechanismen für das Anbieten und Nutzen von Diensten ist eine neue Art der Datenhaltung. In klassischen CSCW-Umgebungen mit einem dedizierten Server werden die Daten und Objekte in einer zentralen Persistenzschicht gespeichert. In mobil-verteilten und spontan verbundenen Formen der Kooperation lässt sich der Zugriff auf eine solche zentrale Persistenzschicht

nicht länger durchgängig realisieren. Aus diesem Grund werden Wissensobjekte und Kooperationsdaten über die mobilen Geräte der Benutzer verteilt gespeichert und führen zu einer verteilten Datenbasis ohne zentrale Verwaltungs- und Kontrollmechanismen.

Für die Kommunikation innerhalb der Kooperationsumgebung können die mobilen Geräte der Benutzer Ad-hoc-Netzwerke oder bereits existierende verwaltete Netzwerkinfrastrukturen wie das Internet nutzen. Nicht zuletzt verschwinden die Grenzen zwischen individueller und kooperativer Wissensorganisation in Nutzungszenarien mobiler Kooperation. Die Benutzer wechseln nahtlos von der Gruppenarbeit zur individuellen Arbeit und umgekehrt. Es entstehen neue Konstellationen des Gebrauchs von kooperationsstützenden Systemen, die sich stark an dem gewünschten Nutzungskontext orientieren. Kontext meint in diesem Zusammenhang die Etablierung von virtuellen Kollaborationsgruppen in Abhängigkeit von der sozialen Struktur der Gruppe und/oder dem Aufenthaltsort der Teilnehmer.

Diese komplexen Kooperationskonstellationen führen zu hochdynamischen Netzwerkstrukturen, die eine verteilte und redundante Speicherung des gemeinsamen Wissensraums über die beteiligten Geräte unverzichtbar machen. Diese verteilte und redundante Speicherung der gemeinsamen Kooperationsobjekte erlaubt es den Benutzern mit diesen zu arbeiten, auch wenn sie zeitweise von ihren Kollaborationspartnern getrennt sind. Dieses Vorgehen kann aber in abweichenden Versionen redundant gespeicherter und überschneidend bearbeiteter Dokumente resultieren. Ein Konflikt bei dem Versuch die Versionen des redundant gespeicherten Dokuments wieder zusammenzuführen ist hier nahezu unvermeidbar. Darum müssen derartige Konflikte kooperativ mittels geeigneter Werkzeuge behoben werden können.

Die Konsistenz der Kollaborationsdaten rückt somit in den Mittelpunkt des Interesses für den Entwurf von mobil-verteilten Kooperationsumgebungen. Hier sind Nutzer nicht länger an die reine Online-Arbeit (verbunden mit einem Server oder den Kollaborationspartnern) gebunden, sondern arbeiten online wie offline mit verschiedenen teils replizierten Materialien. Im Unterschied zu zentral gesteuerten CSCW-Systemen, die ebenfalls eine Offline-Arbeit und Replikationsmechanismen bieten, fehlt in der mobil-verteilten Kooperationsumgebung zunächst eine zentrale Kontrollinstanz, welche die Lösung von Versionskonflikten koordiniert. Ohne solche zentralen Kontrollstrukturen müssen auch die bewährten Gruppen- und Zugriffskonzepte auf die Anforderungen einer dezentralen verteilten Verwaltung zugeschnitten werden.

Die Vielfalt der Aspekte einer Zusammenarbeit in einem mobilen Kooperationskontext, welche die zukünftigen Architekturen mobil-verteilter Kooperationsumgebungen bestimmt, macht eine genaue Anforderungsbestimmung anhand der in Kapitel 2 vorgestellten mobil-verteilten Kooperationsszenarien notwendig. In diesem Kapitel werden daher die technischen Anforderungen an eine mobil-verteilte Kooperationsumgebung mit Blick auf das bewährte Konzept der virtuellen Wissensräume (vgl. Abschnitt 3.2) beschrieben und entlang der Handlungs- wie der Umsetzungsperspektive der Kooperationsunterstützung eingeordnet. Dies sind zum einen, die den neuen Qualitäten der mobilen Zusammenarbeit geschuldeten Bereiche der „*Vernetzung und Sichtbarkeit*“, „*Kontextualisierung*“ und „*Konsistenz und Reversibilität*“



Abbildung 3.1: Die Anforderungen mobilitätsunterstützender Kooperationsumgebungen beinhalten neben den bekannten Unterstützungsfunktionen zur *Kommunikation*, *Koordination*, *Kooperation* und *Kollaboration* zusätzlich die Faktoren *Vernetzung* und *Sichtbarkeit*, *Kontextualisierung* und *Konsistenz* und *Reversibilität*

und zum anderen, die auf klassischen Kooperationsunterstützung aufbauenden und an mobil-verteilten Strukturen anzupassenden Bereiche der *Kommunikation*, *Koordination* und *Kooperation* (vgl. Abbildung 3.1). Als erster Schritt werden nun die drei typischen Phasen mobil-spontaner Kooperation skizziert.

3.1 Phasen mobil-spontaner Kooperation

Szenarien des Arbeitens in mobilen und spontanen Arbeitsgruppen können in drei wesentliche Phasen aufgeteilt werden: *Gründung*, *Kooperation* und *Abschluss* [vgl. Eßmann und Hampel, 2005b]. *Phase I (Gründung)* beinhaltet die Etablierung der Kooperationsgruppe und das Sammeln der Dokumentenbasis. *Phase II (Kooperation)* wird durch die Dynamik der mobilen Zusammenarbeit mit wechselnden Benutzerzusammensetzungen charakterisiert (Benutzer betreten und verlassen die Kollaboration in einer spontanen Weise). Die abschließende *Phase III (Abschluss)* beendet den Kooperationsprozess (zumindest zeitweise) und bewahrt die Kooperationsergebnisse für spätere Kollaborationssitzungen. Von Phase III aus kann die Kooperation fortgesetzt, zu einem späteren Zeitpunkt wieder aufgenommen werden, in eine zentral verwaltete Infrastruktur überführt werden oder als Basis für die individuelle Bearbeitung der Kooperationsergebnisse dienen (vgl. Abbildung 3.2).

Keine der drei Phasen ist zu ihren benachbarten Phasen abgeschlossen, der Übergang zwischen ihnen erfolgt nahtlos. Jede Änderung in der Gruppenstruktur der Kollaboration bewirkt einen Wechsel zu einer benachbarten Phase. Im Falle eines Beitritts von Benutzern in die Kooperation ist stets die Phase I (Gründung) betroffen und im Fall des Ausscheidens von Benutzern aus der Kooperation kommt stets

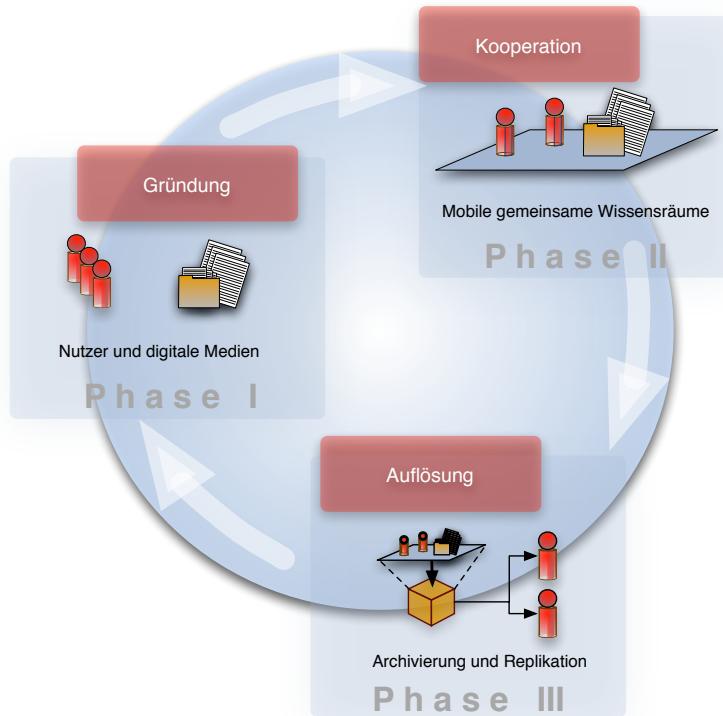


Abbildung 3.2: Der Kollaborationsprozess als Regelkreis. In Phase I (Gründung) gründen Benutzer Kollaborationsgruppen, in Phase II (Kooperation) erfolgt die eigentliche Kollaboration in virtuellen Wissensräumen und in Phase III (Auflösung) werden die Ergebnisse für zukünftige Kooperationssitzungen archiviert.

Phase III (Auflösung) ins Spiel, um die bisherigen Kollaborationsergebnisse für die scheidenden Benutzer zu archivieren.

Die Szenarien aus Kapitel 2 zeigen, dass das Auffinden und Auswählen potentieller Partner für die Kooperation aus den anwesenden und erreichbaren Benutzer eine erste wichtige Aufgabe der *Gründungsphase* einer Kooperationsgruppe ist. Durch diese Auswahl werden diejenigen Benutzer bestimmt, die Zugang zu dem gemeinsamen mobilen Wissensraum haben. Die Auswahl geeigneter Kollaborationspartner kann beispielsweise, wie in einigen der vorgestellten Szenarien, durch explizite Auswahl aus einem vorhandenen Fundus von Benutzern erfolgen.

Alternativ können Benutzer auch die Mitgliedschaft bei der Gruppe beantragen und bei deren Zustimmung in diese aufgenommen werden. Ein wichtiger Faktor, der hilft, die Benutzer bei der Gruppengründung zu unterstützen, kann hierbei der Kontext der potentiellen Partner sein. Als Beispiel sei hier der gemeinschaftliche Aufenthaltsort der Gruppenmitglieder als Grundlage für die Gruppengründung ge-

nannt. Natürlich können zusätzlich auch räumlich entfernte Benutzer eingebunden werden, sofern die nötigen technischen Infrastrukturen existieren.

Die *Kooperationsphase* der so gegründeten Kooperationen, ist durch den Austausch und das Strukturieren von Materialien innerhalb der mobilen Arbeitsbereiche gekennzeichnet. Zu diesem Zweck fügen die Teilnehmer Dokumente in die Struktur ein, um sie gemeinsam mit den Kooperationspartnern zu nutzen. In einer mobil-spontanen Kooperation haben alle Teilnehmer zunächst dieselben Zugriffsrechte bezüglich des gemeinsamen Arbeitsbereiches. Zudem findet hier die Kooperation zumeist in Präsenzsituationen statt, kann aber auch räumlich entfernte Benutzer einschließen. Somit ist die Gruppenstruktur durch eine hohe Fluktuation der verfügbaren Gruppenmitglieder gekennzeichnet – neue Mitglieder treten spontan der Gruppe bei und vorhandene Mitglieder verlassen diese bewusst oder aufgrund ihrer Mobilität.

In der *Auflösungsphase* einer spontan vernetzten Kooperationsgruppe übertragen die Beteiligten die Kooperationsergebnisse für eine spätere Weiterverwendung als persönliche Kopie auf ihre mobilen Geräte oder Archivieren sie auf einen verfügbaren dedizierten Server. Scheiden einzelne Teilnehmer aus, speichern sie die aktuellen Ergebnisse lokal und aktualisieren diese bei einem späteren Wiedereintritt in die Kooperationssitzung. Dabei kann es sein, dass unverbundene Teilnehmer auf den Kooperationsmaterialien weiterarbeiten und der Kooperation in einer geänderten Medienkonstellation wieder betreten.

So führen aus einer medienperspektivischen Sicht technische Änderungen wie der Abbruch von Netzwerkverbindungen zu Änderungen in höheren Ebenen der Kooperationsstruktur. Während auf der technischen Ebene neue Kommunikationsverbindungen initiiert werden müssen, muss den nun unverbundenen Nutzern ein weiterer Zugriff auf die Kooperationsmaterialien gewährt werden. Diese Offline-Verfügbarkeit der Medien in einer mobil-verteilten Kooperationsumgebung führt zu besonderen Anforderungen aus Handlungs- und Umsetzungsperspektive. Der folgende Abschnitt betrachtet daher zunächst die Folgen der Mobilität für das medienzentrierte Konzept der virtuellen Wissensräume.

3.2 Medienfunktionen und Mobilität

Die Bewertung existierender und zukünftiger Architekturen im Bereich der Kooperationsunterstützung für mobile Nutzungsszenarien bedarf der Identifizierung der benötigten Basisfunktionalität. Diese bildet das Mindestmaß an Unterstützungs-funktionen für eine reibungslose Zusammenarbeit mittels einer computergestützten Kooperationsumgebung. Aus der Perspektive einer medienzentrierten Kooperation in gemeinsamen Arbeitsbereichen hat eine mangelnde Unterstützung dieser Basisfunktionalität automatisch einen so genannten *Medienbruch* zur Folge. Die Benutzer können bestimmte Aufgaben nicht mehr innerhalb des Kooperationssystems bewältigen und müssen auf externe Werkzeuge ausweichen. Medienbrüche stellen somit eine

technikbedingte Unterbrechung der Arbeitsvorgänge dar. Keil-Slawik und Hampel [vgl. Hampel, 2001, S.3] beschreiben diese Medienbrüche wie folgt:

„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.“

Um Medienbrüche in Hinsicht auf eine medienzentrierte Kooperation zu vermeiden, wird im Rahmen dieser Arbeit auf das erprobte Konzept der *virtuellen Wissensräume* und *Medienfunktionen* [Hampel, 2001] zurückgegriffen und bezüglich mobil-spontaner Kooperationsszenarien betrachtet. Die virtuellen Wissensräume stellen ein raumbasiertes Konzept für die Strukturierung gemeinsam genutzter Medien- und Dokumentenbestände dar, das mittels so genannter *primärer Medienfunktionen* die Basisfunktionalität für die gemeinsame Manipulation der Struktur des virtuellen Wissensraumes und der enthaltenen Medien bereitstellt. Die primären Medienfunktionen können durch weitergehende Medienfunktionen ergänzt werden, die höhere Ebenen der Kooperationsunterstützung implementieren. Insgesamt werden drei Ebenen der Medienfunktionen unterschieden:

- *Primäre Medienfunktionen* bilden die Gruppe der *Basisfunktionen*, ohne die eine medienbruchfreie Zusammenarbeit auf dem Fundament einer gemeinsamen Mediennutzung nicht möglich ist. Die primären Medienfunktionen werden zur weiteren Differenzierung noch einmal in *individuelle* und *kooperative* Medienfunktionen unterteilt. Erstere ermöglichen die *individuelle* Mediennutzung im eigenen Wahrnehmungsraum. Sie bestehen aus den Funktionen *Erzeugen*, *Löschen*, *Verknüpfen* und *Arrangieren*. Zweitere sind für den *kooperativen* Mediengebrauch unabdingbar und werden aus den Funktionen *Übertragen*, *Zugreifen* und *Synchronisieren* gebildet.
- *Sekundäre Medienfunktionen* bilden die Gruppe komplexerer Funktionen der Mediennutzung. Sie enthalten bereits eine Annahme über die Art der Mediennutzung durch die Anwender. Sie sind nicht universell einsetzbar und erscheinen nicht in jedem Nutzungskontext sinnvoll.
- *Tertiäre Medienfunktionen* passen sich der Mediennutzung der Benutzer an und versuchen den virtuellen Wissensraum nach deren Bedürfnissen zu strukturieren.

Während die *primären Medienfunktionen* eine minimale aber für die Kooperation ausreichende Basisfunktionalität zur computergestützten und medienzentrierten Zusammenarbeit bieten, stellen die höheren Ebenen der Medienfunktionen eine an bestimmte Nutzungskonzepte gebundene Funktionalität dar. Diese Arbeit fokussiert beim Mediengebrauch daher die technische Unterstützung der primären Medienfunktionen, um die Basis für eine kooperative Medienstrukturierung in mobilen Nutzungsszenarien zu schaffen.

Die technische Unterstützung der Medienfunktionen kann, je nach Umgebung und Tiefe der technischen Betrachtung, unterschiedlich komplex ausfallen und setzt sich dabei oftmals aus weiteren Funktionen zusammen. Als Beispiel sei hier das Versenden einer Datei an einen anderen Nutzer für eine gemeinsame Bearbeitung genannt. Diese Funktion des Versendens kann als eine Sequenz von Einzelschritten betrachtet werden: Die Bestimmung der Adresse des Empfängers, die Ermittlung eines Pfades von der Adresse des Versenders zur Adresse des Empfängers, und die Übermittlung der Datei zum nächsten benachbarten Knoten auf diesem Pfad. Je nach Komplexität des Umfeldes und Detailtiefe der Betrachtung sind einige dieser Schritte ihrerseits Grundfunktionen oder können wiederum durch Sequenzen weiterer, grundlegenderer Funktionen beschrieben werden.

Die technische Komplexität der Umgebung überträgt sich so auch auf die medienperspektivische Betrachtung der *Medienfunktionen*. Eine Beobachtung der *primären Medienfunktionen* in ihrer zeitlichen Entwicklung zeigt, dass die *kooperativen primären Medienfunktionen* im Gegensatz zu den *individuellen primären Medienfunktionen* einem stetigen Wandel unterworfen sind. Wurde anfangs keine Unterscheidung zwischen individuellen und kooperativen primären Medienfunktionen getroffen, betrachten die kooperativen primären Medienfunktionen die Besonderheiten einer vernetzten und gemeinsamen Mediennutzung. Insbesondere die *kooperativen primären Medienfunktionen* (im Folgenden einfacheitshalber *kooperative Medienfunktionen* genannt) hängen stark von dem technischen Umfeld ab, in dem sie implementiert werden.

Die Diskussion um die Abhängigkeit der kooperativen Medienfunktionen von der Betrachtung des technischen Umfeldes kann z. B. an der Frage der Funktion *Adressieren* als *kooperative Medienfunktion* festgemacht werden. Ursprünglich war sie keine kooperative Medienfunktion, obwohl das *Adressieren* eines Medienobjektes eine Basisfunktion jeder kooperativen Mediennutzung ist. Um ein Medienobjekt bearbeiten zu können, muss es zunächst benannt (adressiert) werden können. Der Aufwand für das Adressieren eines Medienobjektes steigt mit der Komplexität der Vernetzung der Kooperationsumgebung. Diese Komplexität könnte für die Benutzer bedeuten, dass sie in dem System ein Medienobjekt nicht adressieren können, was zu einem Medienbruch führt. Folgt man dieser Logik, ist Adressieren möglicherweise eine Medienfunktion, da es eine essentielle Grundfunktion für einen durchgängigen Mediengebrauch ist.

Der Grund, dass die Funktion *Adressieren* noch nicht bei den Medienfunktionen erscheint, mag darin liegen, dass diese Funktion in einem geschlossenen und zentral verwalteten System mit einem ebenso geschlossenen und zentral verwalteten Namensraum als selbstverständlich erscheint. Dies ändert sich jedoch bei der Bereitstellung von Medienobjekten in offenen und verteilten Systemen. Da die Objekte auf nahezu allen Knoten des Systems gespeichert sein können, entsteht hier die Frage nach deren genauen Speicherorten für einen Zugriff auf die Medienobjekte.

Analog zu dem Vorgehen der Ermittlung *primärer Medienfunktionen* für eine medienzentrierte Basisfunktionalität, werden in den folgenden Abschnitten technische Grundfunktionen identifiziert, die ein Kooperationssystem bieten muss, um eine me-

dienzentrierte Kooperation in mobilen Nutzungsszenarien zu unterstützen. Dabei werden Anknüpfungspunkte an die primären Medienfunktionen geschaffen und eine Überführung dieser von zentral verwalteten Systemen auf verteilte und offene Architekturen betrachtet. Der besondere Fokus liegt dabei auf einer spontanen Etablierung von Kooperationsumgebungen in der zuvor trivial erscheinende Grundfunktionen zu komplexen und zentralen Fragestellungen werden, die bei weitem nicht jedes vorhandene System selbstverständlich bereitstellt. Grundlage für diese Betrachtung bilden die in Kapitel 2 entwickelten Szenarien.

Die folgenden Abschnitte sind daher wie folgt gegliedert: Der Abschnitt „*Neue Qualitäten mobiler Kollaboration*“ identifiziert die benötigten Grundfunktionen für die neuartige mobile Nutzung von virtuellen Wissensräumen in einer mobilitätsunterstützenden Kooperationsumgebung. Die in diesem Abschnitt vorgestellten Anforderungen ergeben sich daher aus dem neuartigen mobilen Gebrauch und weniger aus den technischen Anforderungen im mobilen Nutzungsumfeld an sich. Diese neuen technischen Anforderungen an klassische Unterstützungsfunktionen des CSCW werden im Anschluss daran im Abschnitt „*Neue Rahmenbedingungen*“ vorgestellt.

3.3 Neue Qualitäten mobiler Kollaboration

Aus den in Kapitel 2 entwickelten Szenarien mobiler Kooperation ergeben sich spezielle Anforderungen an den Funktionsumfang und die Mechanismen der neuartigen Kooperationsumgebung. Einige dieser Anforderungen liegen abseits der klassischen Kooperationsszenarien und leiten sich aus dem spontanen Charakter der Zusammenarbeit und der Mobilität der Benutzer her.

Grundsätzlich orientieren sich die Anforderungen an der Kollaboration in mobilen Wissensräumen an den in Kapitel 2 entwickelten Nutzungsszenarien. Die neuen Qualitäten der Ad-Hoc-Kooperation mit Hilfe mobiler Computer werden im Folgenden aus diesen Szenarien ermittelt und analysiert. Vorweggenommen lassen sich die innovativen Aspekte mobil-spontaner Kollaboration entlang der Begriffe *Vernetzung*, *Sichtbarkeit*, *Kontextualisierung*, *Konsistenz* und *Reversibilität* einordnen.

Die Komplexität der semantischen Verknüpfung von mobilen Wissensräumen und der spontane Charakter der mobilen Kollaborationsszenarien zwingen die Kooperationsumgebung, die Strukturen des mobilen Wissensraums so weit wie möglich automatisch zu erschließen, um die Aufmerksamkeit der Benutzer nicht von der eigentlichen Zusammenarbeit abzulenken.

Besonders in der Gründungsphase spielen *Vernetzung*, *Sichtbarkeit* und *Kontext* eine wichtige Rolle. Ist der *mobile Wissensraum* erst errichtet, muss die Kooperationsumgebung zudem dessen *Konsistenz* sicherstellen. Da sich die Strukturen des mobilen Wissensraums aufgrund der Mobilität seiner Nutzer schnell und dynamisch ändern können, ist es wichtig, die Konsistenz so lange wie möglich aufrecht zu erhalten. Sollte dies nicht gelingen, muss die Kooperationsumgebung Mechanismen bereitstellen, die im Falle einer Inkonsistenz in der Struktur der mobil-verteilten

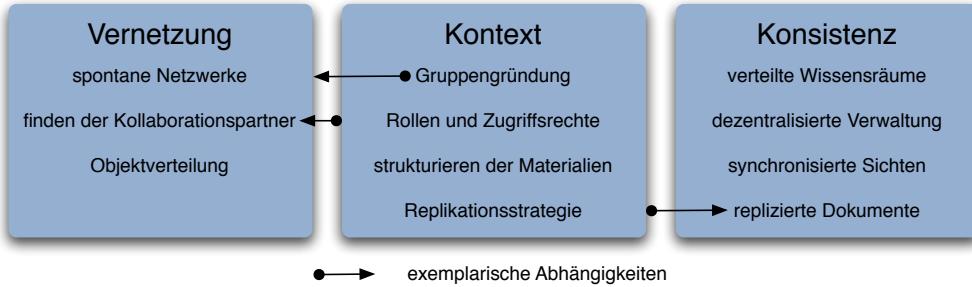


Abbildung 3.3: Schlüsselfaktoren mobil-verteilter Wissensräume: *Vernetzung und Sichtbarkeit, Kontextualisierung und Konsistenz und Reversibilität*.

Wissensräume deren Konsistenz wieder herstellen helfen. Das Zusammenspiel dieser Schlüsselanforderungen verdeutlicht Abbildung 3.3.

Die folgenden drei Abschnitte befassen sich folglich mit den Anforderungen der „*Vernetzung und Sichtbarkeit*“, „*Kontextualisierung*“ und „*Konsistenz und Reversibilität*“ als Schlüsselfaktoren einer mobilitätsunterstützenden Kooperationsumgebung auf Basis mobil-verteilter Wissensräume.

3.3.1 Vernetzung und Sichtbarkeit



Der Faktor, der eine Kooperation erst ermöglicht, ist die Wahrnehmung potentieller Kooperationspartner und die Kontaktaufnahme zu diesen. Der Trend zu allgegenwärtigen Netzwerkinfrastrukturen an nahezu allen öffentlichen Plätzen lässt den Eindruck entstehen, Benutzer könnten jederzeit auf eine solche Netzwerkinfrastruktur zurückgreifen, um über diese eine Kommunikation mit ihren Kooperationspartnern zu etablieren. Trotz der Anstrengungen kommerzieller Dienstleister, Netzwerke allgegenwärtig zugänglich zu machen, bleiben viele Benutzer von der Nutzung dieser Infrastrukturen ausgeschlossen.

Gründe hierfür sind häufig proprietäre Protokolle, Zugangsbegrenzungen auf geschlossene Personenkreise, geblockte Dienste oder die Verbindungskosten. Da der Zugang zu bestehenden Netzwerkinfrastrukturen nicht überall gewährleistet wer-

den kann oder zu hohe Kosten verursachen würde, müssen mobilitätsunterstützende Kooperationsumgebungen als ersten Schritt der Kollaboration ein Kommunikationsnetzwerk zwischen den mobilen Geräten der Benutzer einrichten. Als technologische Basis für diese Kommunikationsstrukturen eignen sich insbesondere die in Abschnitt 4.2.1 vorgestellten Ad-Hoc-Netzwerk-Protokolle, die in der Lage sind, mit einem minimalen Konfigurationsaufwand seitens der Benutzer, ein Netzwerk zwischen den mobilen Computern zu errichten.

Eine solche *Ad-Hoc-Vernetzung der mobilen Kooperationspartner* ist zentrale Voraussetzung für jede weitere computergestützte Kooperationshandlung. Dieser Aspekt der Vernetzung geht über die in Abschnitt 4.2 behandelte technische Vernetzung hinaus. Zwar stellen diese innovativen Netzwerkprotokolle die Basis spontaner Computerkommunikation bereit, doch bleibt die Kooperationsumgebung ohne eine Wahrnehmung potentieller Kooperationspartner und kooperationsunterstützender Dienste für den Benutzer schwer zugänglich.

Für eine Kooperationsumgebung werden zahlreiche miteinander verknüpfte Dienste benötigt, die helfen, Kollaborationspartner zu finden, mit ihnen zu kommunizieren und mit gemeinsamen Dokumenten zu arbeiten. Diese Dienste müssen für jeden Teilnehmer der Kooperationsumgebung verfügbar sein. Sowohl die Basisdienste als auch die Zusatzdienste sollten automatisch und ohne Benutzereingriff konfiguriert werden. Dies gilt insbesondere, wenn Dienste an Dritte im Netzwerk angeboten werden, da hier die Motivation der lokalen Benutzer zur Konfiguration des Dienstes am niedrigsten ist. Generell ist die manuelle Konfiguration eines Dienstes stets ein mögliches Nutzungshemmnis, das es zu vermeiden gilt.

Der zu Beginn der Kooperation wichtigste Dienst ist die Identifizierung potentieller Kollaborationspartner. Diese Identifizierung kann durch die Auswertung des Kontextes verfügbarer Benutzer oder über deren Nutzerprofil erfolgen. Als Kontext der Benutzer kann deren Aufenthaltsort sowohl in der virtuellen Welt der Wissensräume als auch in der realen Welt dienen oder aber deren Beziehungen zu anderen Benutzern. Besonders in der Präsenzkooperation, die eine intuitive Form der spontanen Kollaboration darstellt, ist der Aufenthaltsort der Benutzer in ihrem realen Umfeld nützlich für die Gruppengründung. Nutzerprofile wiederum zeigen explizit Interessen und Kompetenzen der potentiellen Kollaborationspartner auf. Alle diese Hilfestellungen unterstützen somit die gezielte Auswahl der Kollaborationspartner für die Bildung einer Gruppe innerhalb der Kooperationsumgebung.

Die Struktur der Gruppe, die von den Kollaborationspartnern formell oder informell gebildet wird, definiert den Charakter der Kollaboration. Sie ist Indiz für die benötigte Vernetzung der Teilnehmer. Während sich im natürlichen Kooperationsumfeld derartige Gruppen auf rein sozialer und räumlicher Basis finden, ist es schwierig, derartige Gruppenstrukturen in mobilen Kooperationsumgebungen nebenläufig zum Kooperationsprozess zu bilden. Eine formalisierte Gruppenstruktur wie in klassischen CSCW-Systemen kann sich kaum an die Dynamik mobil-verteilter Kooperationsszenarien anpassen. Die formale Gründung einer Gruppe ist unter diesen Gesichtspunkten zu umständlich und genügt daher nicht dem Verhältnis zwis-

schen Aufwand und Nutzen in den oft kurzfristigen Sitzungen einer mobilen Ad-Hoc-Kooperation.

Der spontane Charakter der Kooperationssitzungen bedingt außerdem eine hohe Dynamik in der Zusammensetzung der Kooperationsgruppen. Neu hinzukommende Benutzer nehmen an der Kooperationssitzung teil, während andere diese wieder verlassen, um sich anderen Tätigkeiten zuzuwenden. Diese Dynamik wird im Szenario 1.7 verdeutlicht. Die dynamische Kooperation in mobil-verteilten Kooperationsszenarien benötigt also entsprechend *dynamische Gruppenstrukturen*. Das Hauptaugenmerk liegt hierbei auf der Flexibilität bezüglich der Fluktuation der Mitglieder und der beiläufigen Gründung einer solcher Kooperationsgruppe anhand des Arbeitskontextes wie in Szenario 2.2 beschrieben.

Da aber klassische und formelle Kooperationsformen weiterhin unterstützt werden sollen, müssen diese dynamischen Gruppenstrukturen auch gemeinsam mit langfristig genutzten Gruppenstrukturen existieren können. Trotz ihrer Dynamik sollte eine *Gruppe gegenüber fremden Benutzern abgeschlossen* sein (vgl. Szenario 2.2), da die Gruppe nicht nur die Vernetzung der Kooperationspartner definiert, sondern oft auch die Berechtigung auf gruppeneigene Dokumente vorgibt.

Die Dynamik mobiler Kooperation muss mit den formalen Anforderungen an die Gruppenstruktur einer CSCW-Umgebungen vereint werden. Diese kann nur erreicht werden, indem sich die Kooperationsumgebung selbsttätig vernetzt und Prozesse der Gruppenkonstruktion und -verwaltung mit Hilfe des Kontextes der Kooperationsteilnehmer automatisiert.

3.3.2 Kontextualisierung



Kontext spielt in einer mobilitätsunterstützenden Kooperationsumgebung eine wichtige Rolle für die Vernetzung der Benutzer. Dieser Kontext kann z. B. aus dem derzeitigen Aufenthaltsort oder den zurzeit benutzten Dokumenten bestehen. Die soziale Vernetzung mittels Gruppenbildung wird dabei ebenso wie die technische Vernetzung von diesem Kontext bestimmt.

In klassischen CSCW-Umgebungen bilden Informationen über den Kontext der Benutzer, die sich nicht direkt aus dem Kooperationssystem selbst ergeben, eine häufig vernachlässigte Möglichkeit, den Nutzern komfortable Hilfestellungen für die

Kooperation zu geben. In einer mobilen Kooperationsumgebung jedoch wird die Integration verfügbarer Kontextinformationen zu einem maßgebenden Faktor für deren Effizienz. Bei oft nur kurzfristigen spontanen Kooperationssitzungen steht der Aufwand für eine manuelle Einrichtung der computergestützten Kooperationsumgebung in keinem Verhältnis zum Nutzen der Kooperationsumgebung. Kontext ist folglich ein grundlegendes Werkzeug, um den Kollaborationsprozess in mobilen Nutzungszenarien zu konfigurieren.

Bei einer intensiven Einbeziehung des Nutzerkontextes in die Kooperationsumgebung kann dieser gleich auf mehreren Ebenen für die Strukturierung von Kooperationsprozessen genutzt werden. Auf Medienebene können Dokumente und andere Materialien anhand ihres Nutzungskontextes strukturiert werden. So kann bereits zu Beginn des Kollaborationsprozesses ein vorstrukturierter Wissensraum zur Verfügung gestellt werden, anstatt die Nutzer mit einer unsortierten Anhäufung von Materialien zu konfrontieren.

Aus einer organisatorischen Perspektive können z. B. Kooperationsgruppen gegründet werden, bei denen alle um einen Tisch versammelten Personen oder alle Studierenden eines Seminars automatisch in diese aufgenommen werden. Als eine wichtige Quelle für den Kontext eines Benutzers – speziell in mobilen Nutzungsszenarien – kann der *Ortsbezug* (engl. Location Awareness) gelten [vgl. Dourish und Bellotti, 1992]. Präsenzkooperation ist die natürliche Form spontan-mobiler Zusammenarbeit, in der die Nähe zu anderen Benutzern auch die Kooperationsbeziehung zu diesen definiert. Diesen Kooperationskontext gilt es in die virtuelle Kooperationsumgebung zu übertragen.

In mobilen Nutzungsszenarien wie Szenario 1.1 wird eine *Wahrnehmung und Identifizierung potentieller Kooperationspartner*, die für eine Zusammenarbeit zur Verfügung stehen, benötigt. Für Präsenzkooperationen muss man *potentielle Kooperationspartner anhand ihres örtlichen Kontextes filtern* können (vgl. Szenario 1.1, Szenario 2.1). Zu diesem Zweck reicht häufig das Wissen um die relative Nähe der Benutzer zu einem Bezugspunkt aus. Dieser Bezugspunkt kann ein anderer Benutzer in der Kooperationsumgebung oder ein kooperationsrelevanter Gegenstand sein. Beispiele mögen hier sein, eine Gruppe mit allen potentiellen Kooperationspartnern in zwei Metern Umkreis von sich selbst zu gründen – oder mit allen um einen Tisch sitzenden Benutzern. Auch ohne Kenntnis der absoluten Position der Teilnehmer und mit nur ungenauer relativer Positionsbestimmung kann bereits eine sinnvolle Unterstützung der Kooperationsprozesse bewerkstelligt werden.

Mittels des Ortsbezugs entsteht so eine soziale Vernetzung zwischen den Teilnehmern der Kooperationsumgebung, in der auch Rollen und Rechte vom aktuellen Kontext der Benutzer abhängen. Der aktuelle Aufenthaltsort kann z. B. einem Benutzer einen Gastzugang zu den Ressourcen der Kooperationsgruppe eröffnen oder einer Person am Rednerpult automatisch die Rolle des Moderators zuweisen. Eine manuelle Festlegung der Rollen und Rechte in einem Kooperationsszenario ist oftmals eine komplexe Aufgabe für die Verwalter einer Kooperationsumgebung, die durch die Nutzung von Kontextinformationen minimiert werden kann.

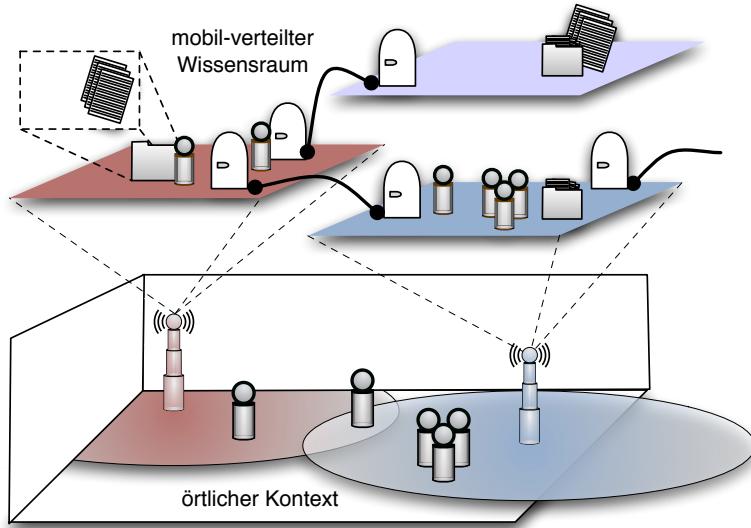


Abbildung 3.4: Bei der Verknüpfung von realem Raum und virtuellem Wissensraum werden örtlichen Gegebenheiten (z. B. einem Seminarraum) virtuellen Wissensräumen zugeordnet. Über die Verbindungen zwischen den Wissensräumen können so auch die realen Räume miteinander verknüpft werden.

Über ihren Kontext können so die Gruppenmitglieder einer Kollaboration bestimmt und zugleich ihre Rolle in der Gruppe definiert werden. Wenn die Benutzer ihren Kontext verändern, passen sich die Zugriffsrechte und Rollenzuweisungen automatisch an den neuen Kontext an. Somit wird durch eine Änderung des Kontextes innerhalb der virtuellen Kooperationsumgebung entlang des realweltlichen Kontextes nebenläufig eine Verknüpfung von virtuellem und realem Wissensraum erreicht [Eßmann und Hampel, 2004]. Abbildung 3.4 verdeutlicht diese Verknüpfung durch die Bildung von Gruppenkontexten in Abhängigkeit des Ortsbezugs der Teilnehmer.

Aus einer technischen Betrachtungsweise kann der Kontext auch für vom Benutzer unbemerkte und transparente Vorgänge in der Kooperationsumgebung genutzt werden. Die in mobilen Nutzungsszenarien übliche verteilte Speicherung der Kooperationsobjekte kann durch den Kontext der beteiligten Knoten gesteuert werden. Der Kontext der Benutzer kann z. B. bei der Entscheidung helfen, auf welchen Knoten der Kooperationsumgebung die Objekte repliziert werden sollen. Während viele CSCW-Systeme die genutzten Objekte bei einem Zugriff lediglich zwischenspeichern (*Caching*), kann der Kontext genutzt werden, um eine gezielte Replikationsstrategie zu entwickeln. Beispielsweise sind Dokumente, die einer Gruppe gehören, meist für alle Mitglieder von Interesse. Eine sinnvolle Replikationsstrategie könnte also den Gruppenkontext nutzen, um die Dokumente auf die Geräte aller Gruppenmitglieder zu replizieren. Dadurch sind die Dokumente der Gruppe für alle Mitglieder

auch im Falle einer Trennung von der Gruppe jederzeit zugänglich. Eine eingehende Betrachtung der Replikation anhand des Gruppenkontextes findet sich in Kapitel 4.

Zusammenfassend betrachtet kann der Kontext der Benutzer somit auf organisatorischer Ebene, Medienebene und technischer Ebene genutzt werden, um mobil-spontane Kooperationsszenarien zu unterstützen. Konkret lässt sich der Kontext in der Kooperation selbst für Gruppengründung, Festlegung von Rollen und Rechten und die Strukturierung des virtuellen Wissensraums verwenden. Aus einer eher technischen Sichtweise kann die Konfiguration der Dienste der Kooperationsumgebung und insbesondere die Replikation der genutzten Objekte über eine Nutzung des Kooperationskontextes gesteuert werden. Die Kontextualisierung der Kooperationsumgebung ist somit eine wesentliche Anforderung für die Unterstützung mobil-verteilter Kooperationsszenarien.

3.3.3 Konsistenz und Reversibilität



In mobil-verteilten Wissensräumen sind die Kooperationsobjekte verteilt auf mobilen Computern der Benutzer gespeichert. Um die Verfügbarkeit der gemeinsam genutzten Objekte sicherzustellen, müssen Teile des Wissensraums oder gar der gesamte Wissensraum zwischen den mobilen Geräten repliziert werden. Die Replikation erlaubt den Kooperationspartnern den Zugriff auf die gemeinsam genutzten Materialien, auch wenn sie von dem Rest der Kooperationsgruppe getrennt sind. Wenn die Replikation alle Objekte beinhaltet, die für eine Aufgabe benötigt werden, können auch isolierte Benutzer den Kollaborationsprozess aufrechterhalten. Diese Möglichkeit zum isolierten Arbeiten an gemeinsamen Objekten in den unverbundenen Phasen erfordert die Unabhängigkeit der mobil-verteilten Kooperationsumgebungen von Diensten entfernter Dienstleister (Server).

In verteilten Architekturen sind zu diesem Zweck Daten sowie Dienste über alle Knoten des zu Grunde liegenden Netzwerkes verteilt. Essentielle Daten und Dienste können dabei auch redundant auf einer Gruppe mehrerer Knoten beheimatet sein. Dieses Vorgehen erlaubt es einem Knoten, die Dienste eines anderen Knotens zu übernehmen, wenn dieser ausfallen sollte. Die Verteilung der Daten und Dienste ist oftmals eine Strategie zur Lastverteilung zwischen den einzelnen Knoten und zur Erhöhung der Ausfallsicherheit des Gesamtsystems.

Der Ausfall eines einzelnen Knotens führt dadurch nicht mehr zum Ausfall des gesamten Systems. Allerdings müssen replizierte Dienste und Daten miteinander abgeglichen werden, um die Konsistenz des Systems zu wahren. Der Fokus der Architektur vieler verteilter Systeme liegt folglich in der Aufrechterhaltung des Gesamtsystems und nicht in der Unterstützung mobiler und zeitweise isolierter Knoten. Ein Verbindungsabbruch zu einem einzelnen Knoten wird stets nur aus der Perspektive des Gesamtsystems betrachtet und dessen Dienste an andere Knoten im Verbund übertragen. Dadurch bleibt das Gesamtsystem bei Knotenausfällen intakt – der isolierte Knoten aber ist solange unbrauchbar bis er wieder Verbindung zum Gesamtsystem aufnehmen kann [Eßmann et al., 2006].

Die neue Qualität mobil-verteilter Kooperationsumgebungen liegt in der Mobilität der Benutzer und der Dynamik der technischen Umgebung. Der Verlust der Anbindung zum Verbund der Kooperationspartner ist eine häufige Begleiterscheinung dieser Mobilität. In verteilten Kooperationssystemen, die nur die Funktionsfähigkeit des Gesamtsystems gewährleisten, sind isolierte Benutzer gezwungen, ihre Arbeitsvorgänge innerhalb des gemeinsamen Arbeitsbereichs ruhen lassen, bis sie wieder Kontakt zu dem Verbund aufnehmen können. Da dies einen Bruch in der durchgehenden Nutzung der Kooperationsumgebung darstellen würde, ist es wichtig, zwischen verteilten Systemen zu unterscheiden, in denen alle funktionierenden Knoten stets mit dem Verbund in Kontakt stehen müssen und solchen, in denen mobile Knoten auch im unverbundenen Zustand (eventuell eingeschränkt) weiterfunktionieren können.

Beide Ausprägungen verteilter Systeme haben unterschiedliche Anforderungen an die Verteilung der Ressourcen, um ihre Funktion aufrechterhalten zu können. Für eine Unterscheidung dieser Systeme werden an dieser Stelle zwei Klassen verteilter Systeme mit redundanten Diensten definiert: die *online-redundanten* verteilten Systeme und die *offline-redundanten* verteilten Systeme.

Während die Klasse der *online-redundanten verteilten Systeme* ihre Priorität in die Aufrechterhaltung der Gesamtfunktionalität legt, liegt diese bei der Klasse der *offline-redundanten verteilten Systemen* in der Aufrechterhaltung der Grundfunktionalität für die einzelnen Knoten. Letztere erlaubt ihren Benutzern auch im unverbundenen Fall ihre Arbeit fortzusetzen.

Der wichtigste Dienst in einem Kollaborationssystem auf Basis der virtuellen Wissensräume ist die Bereitstellung der von den Kooperationspartnern benötigten Materialien und Dokumente (genereller: Objekte). Alle benötigten Objekte müssen zwischen den beteiligten Partnern repliziert werden, damit auf diese bei Bedarf zugegriffen werden kann. Bei der Änderung eines replizierten Objektes müssen zudem alle Repliken aktualisiert werden, um dem aktuellen Stand zu entsprechen und um den Datenbestand der Kooperationsumgebung konsistent zu halten.

Auf Grund der Mobilität der Benutzer und der daraus resultierenden hohen Wahrscheinlichkeit von Netzwerkunterbrechungen, bei denen auch ganze Teilnetzwerke getrennt werden können, ist eine *Online-Synchronisation* der Repliken nicht immer möglich. Daher wird die Konsistenz der Kollaborationsdaten ein zentraler Punkt für die Designfragen eines mobil-verteilten Kollaborationssystem sein.

Eine gezielte Replikationsstrategie erlaubt den nahtlosen Wechsel zwischen *verbundenen (online)* und *unverbundenen (offline)* Arbeiten. Dies ist gerade in mobilen Nutzungszenarien aufgrund der Dynamik in der Vernetzung von hoher Bedeutung. In unverbundenen Phasen des Kollaborationsprozesses fehlt den Benutzern jegliche Wahrnehmung über die Handlungen ihrer Kollaborationspartner. Daher kann es zu einer parallelen Bearbeitung mehrerer Repliken desselben Dokuments kommen, ohne dass die beteiligten Benutzer von den konkurrierenden Änderungen wissen.

Im Fall einer parallelen Änderung der Repliken desselben Objekts, ohne gegenseitige Wahrnehmung der Änderungen anderer, werden die replizierten Instanzen zwangsläufig inkonsistent zueinander. Diese *Gefährdung der Konsistenz* kann sowohl auf Medienebene als auch auf semantischer Ebene liegen. Die Konsistenz auf Medienebene ist z. B. gefährdet, wenn auf redundant gespeicherten Daten zeitgleich gearbeitet ohne diese ständig (online) zu synchronisieren. Die semantische Konsistenz ist in Gefahr, wenn verteilt gespeicherte Daten semantisch gekoppelt sind aber aufgrund gleichzeitiger Änderung der verteilten Datenbestände dieser Zusammenhang verloren geht. Beide Formen der Inkonsistenz können die Konsistenz des gesamten Wissensraums gefährden.

Die Erhaltung der *Konsistenz* der Kooperationsdaten bildet einen der zentralen Faktoren für eine mobile Kooperationsumgebung. Bei der Arbeit in virtuellen Wissensräumen besitzen die Beteiligten eine gemeinsame Sicht auf die Kooperationsdaten. Änderungen an einzelnen Objekten in dieser Umgebung sind somit für alle Benutzer gleichzeitig wahrnehmbar. In Client-Server-Umgebungen werden Änderungen an einem Objekt durch einen Client an den Server angewiesen, der diese ausführt und im Anschluss alle anderen beteiligten Clients über die Änderungen informiert. Die Clients erneuern daraufhin ihre Darstellung der gemeinsamen Sicht, um den aktuellen Stand der Kooperationsobjekte zu repräsentieren. Die zentrale Position des Servers in dieser Architektur befähigt diesen, die Konsistenz der Änderungen an den manipulierten Objekten zu überprüfen und so Konflikte bei zeitgleichen Änderungsanforderungen zu lösen – also nur eine Änderung nach der anderen zuzulassen. In einer in mobil-verteilten Kollaborationsszenarien vorherrschenden Peer-to-Peer-Architektur ist ein ständiger Abgleich der Datenbestände zwischen den einzelnen Peers unvermeidlich.

Synchronizität Eine Gefährdung der Konsistenz muss demnach durch ein Aufrechterhalten der *Synchronizität der Datenbestände* innerhalb des verteilten Kollaborationssystems vermieden werden. Die *Synchronizität* ist wesentliches Indiz für die Benutzbarkeit des Kooperationssystems. Um diese zu Gewährleisten, muss der Abgleich verteilter Datenbestände automatisch und im Hintergrund geschehen. Diese nebenläufige Synchronisierung entlastet den Benutzer von derartigen Aufgaben. Dennoch muss der Benutzer im Konfliktfall eingreifen und die Inkonsistenzen komfortabel beseitigen können. Im Fall einer beschädigten Konsistenz des Datenbestandes ist es daher wichtig, auf die letzte korrekte Version zugreifen zu können. Diesem Aspekt wird mit der Forderung nach *Reversibilität* Rechnung getragen. Dazu gehört

auch die Möglichkeit, die Änderungen, die die Konsistenz verursacht haben, in den konsistenten Datenbestand einzuarbeiten.

Die in Szenario 1.3 und Szenario 1.5 geforderte Synchronisierung von Objekten in der verteilten Kooperationsumgebung entspricht einem wesentlichen Teil der Anforderung der *Synchronizität* im Bereich mobiler Kooperationsumgebungen. Um den Benutzer möglichst wenig zu belasten, muss die *Synchronizität*, wie in Szenario 3.3 ersichtlich, möglichst als Hintergrundprozess erfolgen. Der Abgleich muss dabei nicht nur die physische Konsistenz der Kooperationsdaten überwachen und aufzeigen (Szenario 4.3), sondern auch soweit möglich die *semantische Konsistenz* aufrechterhalten helfen (Szenario 4.4).

Wo nicht anders möglich, kann der Abgleich zwischen den Datenbeständen auch asynchron (z. B. per E-Mail) erfolgen, falls ein direkter und zeitnauer Abgleich nicht möglich ist (Szenario 3.3). Wichtig ist jedoch, dass die Synchronisierung der Kooperationsobjekte die Gruppenstruktur unterhalb der Kooperationspartner berücksichtigt (Szenario 3.3) und den Benutzern erlaubt, sowohl eine eigene, als auch eine verteilte Kopie desselben Objektes zu verwalten (Szenario 1.6).

Die Strategie eines ständigen Abgleichs der verteilten Datenbestände zur Erfüllung der Anforderung der *Synchronizität* ist ein probates Mittel zur Aufrechterhaltung der *Konsistenz* innerhalb des Kollaborationssystems. Sie setzt allerdings voraus, dass alle beteiligten Knoten des Verbundes ständig miteinander in Kontakt stehen – eine Voraussetzung die insbesondere in mobilen Nutzungsszenarien oft nicht erfüllt werden kann. Jenseits der technisch bedingten Lücken innerhalb der Netzwerkommunikation verlassen Benutzer häufig für einen längeren Zeitraum den Verbund und Arbeiten isoliert oder in einer Teilgruppe weiter auf den Kooperationsdaten. Diese Tatsache macht eine ständige Koordinierung von konkurrierenden Änderungen unmöglich. Ein Sperren von Objekten für die Bearbeitung, wie es beim *Pessimistic Locking* der Fall ist, ist aus mehreren Gründen in einer solchen Umgebung nicht empfehlenswert:

- Die Zeiträume der Sperrung können aufgrund von Verbindungsabbrüchen (bis zu nächsten Synchronisierung) sehr lange dauern und machen die gesperrten Objekte für alle anderen Benutzer unbrauchbar.
- Wie die Änderung selbst, muss die Sperranforderung an alle Repliken übermittelt werden.
- Die Sperrung ist oft nicht vorhersagbar und erfolgt zuweilen erst wenn die Verbindung zu den anderen Repliken bereits unterbrochen ist.

Da ein vorgreifendes Sperren der gemeinsam genutzten Objekte aus diesen Gründen nicht möglich ist, empfiehlt sich der Einsatz von so genannten *optimistischen Sperren* (*Optimistic Locking*) [Kung und Robinson, 1981]. Optimistische Sperren beruhen auf der Annahme, dass konkurrierende Änderungen an einem Objekt eher selten vorkommen. Daher wird ein Konflikt nicht bereits im Vorfeld durch ein Sperren für den Fall einer Änderung verhindert, sondern erst dann eingegriffen, wenn ein

Konflikt zwischen zwei unterschiedlichen Versionen desselben Objektes entstanden ist. Im Fall eines solchen Konflikts wird das Objekt für weitere Änderungen gesperrt und die beteiligten Systeme (bzw. deren Benutzer) aufgefordert, den Konflikt zu lösen.

An diesem Punkt des Kooperationsprozesses können aufgrund langfristig unverbundenen Arbeitens bereits große Teile des Inhalts eines Kooperationsobjektes von einander abweichen. Anders als bei Versionskontrollsystmen für die Softwareentwicklung können diese Abweichungen bei natürlichsprachlichen Texten nicht automatisch gelöst werden. Die Benutzer werden daher im Falle eines Versionskonfliktes vom Kooperationssystem bei dessen Lösung nicht unterstützt.

Weiterhin kann ein automatisches Zusammenführen der Dateien zu semantischen Konflikten führen oder das Ergebnis unverträglich zu weiteren Änderungen Dritter sein, was weitere noch größere Versionskonflikte zur Folge hätte. Daher ist es wichtig, jeden Schritt der Bearbeitung reversibel zu machen, um spätere Konflikte besser lösen zu können.

Reversibilität Die Synchronizität der Kooperationsobjekte ist die zentrale Anforderung für die Konsistenz einer verteilten Kooperationsumgebung. Da eine ständige Verbindung zwischen den Kooperationspartnern nicht garantiert ist, kann ein von der Gruppe isoliertes Arbeiten der Teilnehmer die Regel werden. Daher bedarf es der Möglichkeit bei Verletzung der Konsistenz der Kooperationsobjekte den Schaden rückgängig machen zu können. Falls möglich können die vorgenommenen Änderungen nach und nach in Abstimmung mit den Kooperationspartnern zu einer konsistenten Version des Kooperationsobjektes zusammengeführt werden.

Zu diesem Zweck gilt es, die *Reversibilität* der Manipulationen an den Kooperationsobjekten sicherzustellen. Dies betrifft zum einen die *Versionierung*, die beiläufig alle Änderungen an einem Objekt protokolliert, Konflikte zwischen redundant gespeicherten und zeitgleich bearbeiteten Objekten erkennt und erlaubt, diese Änderungen bei Bedarf rückgängig zu machen. Zum anderen betrifft dies aber auch die bewusste *Archivierung* von Kooperationsobjekten für spätere Kooperationssitzungen.

Szenario 1.5, Szenario 1.6 und Szenario 3.2 verdeutlichen die zentrale Rolle, die der *Versionierung* der Kooperationsobjekte im Fall des unverbundenen Arbeitens mit gemeinsamen Objekten zukommt. Die Versionierung hilft hier die Konflikte bei konkurrierender Bearbeitung eines redundant verteilten Objektes aufzudecken und zu lösen. Manchmal ist das Erstellen konkurrierender Versionen aber auch, wie in Szenario 2.5 beschrieben, ein gewollter Prozess, der ebenfalls mittels *Versionierung* unterstützt werden kann.

Die Funktionen zum kooperativen Bewerten und Zusammenführen konkurrierender Dokumentversionen sind daher ebenfalls Bestandteil der Anforderung *Versionierung*. Wie wichtig diese Funktionen für den verteilten kooperativen Prozess sind, heben Szenario 2.5, Szenario 3.2 und Szenario 4.3 hervor.

Reversibilität beinhaltet aber nicht nur den Einsatz einer Versionskontrolle im laufenden Kooperationsprozess, um die Konsistenz der Kooperationsobjekte bei konkurrierender Bearbeitung zu wahren, sondern auch *Archivierung* der Kooperationsergebnisse für einen späteren Zugriff auf diese. Während Client-Server-Systeme zu diesem Zweck die Objekte persistent in einer Datenbank oder Ähnlichem ablegen, ist dies in einer mobilen Kooperationsumgebung nicht unbedingt möglich, da die Ressourcen der mobilen Rechner weit hinter denen von zentralen Servern liegen. Da mehrere Kooperationspartner Interesse an einem solchen Archiv haben könnten, muss es wiederum redundant gehalten werden.

Archivieren bedeutet in diesem Zusammenhang, dass ein Archiv angelegt wird, *nachdem* die eigentliche Kooperation abgeschlossen ist. Daher müssen Archive nicht synchronisiert, sondern lediglich für jeden interessierten Teilnehmer separat gespeichert werden. Demnach erscheint es nicht immer sinnvoll, diese Archive in dem persistenten und stark replizierten Datenspeicher der mobilen Kooperationsumgebung zu halten. Zum einen würden dadurch unnötig Ressourcen der Kooperationsumgebung für die Synchronisation der Archive verbraucht. Zum anderen würden nachträgliche, nicht länger im Rahmen einer Kooperation getätigten Änderungen weiterhin an alle Teilnehmer weitergereicht, obwohl diese eventuell keinerlei Interesse mehr an den Änderungen haben und lieber das ursprüngliche Kooperationsergebnis für eine spätere Weiterverwendung nutzen wollen.

Damit das Archiv als Basis einer neuerlichen Kooperationssitzung dienen kann muss die vollständige Konfiguration als Momentaufnahme der Kooperationsumgebung inklusive der Informationen über Dokumenten- und Gruppenstruktur enthalten (Szenario 1.7, Szenario 1.9). Dazu gehören auch die Metadaten aller Kooperationsobjekte als wichtiger Bestandteil der Kooperationsergebnisse (Szenario 4.5). Die archivierten Momentaufnahmen einer Konfiguration können an Dritte (Szenario 1.9) übermittelt oder in eine zentralisierte CSCW-Umgebung eingefügt werden, um eine langzeitige Verfügbarkeit sicherzustellen (Szenario 1.9, Szenario 4.6). Für den Schutz sensibler Daten besteht bei einer Weitergabe des Archivs an Dritte die Notwendigkeit, gewisse Informationen aus einem Archiv zu entfernen (z. B. Gruppenstruktur).

Soll eine Wiederaufnahme einer Kooperation anhand eines Archivs gewährleistet werden, muss das Kooperationssystem in der Lage sein, den letzten Zustand der archivierten Kooperation vollständig wieder herzustellen. Da die äußeren Umstände (verfügbare Kooperationspartner, technische Infrastruktur) sich bis zur Wiederaufnahme ändern können, muss zudem bei der Zustandswiederherstellung flexibel auf externe Fehlerquellen und Änderungen der Konfiguration reagiert werden können. (Szenario 4.5).

In einem offenen Netzwerkumfeld, wie es in mobil-verteilten Kooperationsszenarien Anwendung findet, der Einbindung externer Dienste und fremder Kooperationsumgebungen eine zentrale Bedeutung zu. Die daraus resultierenden Anforderungen an die mobil-verteilte Kooperationsumgebung werden unter dem Begriff der *Offenen Architektur* zusammengefasst.

Offene Architektur Eine offene Architektur und frei zugängliche Schnittstellen bilden die Voraussetzungen für eine flexible Einbettung eines Softwaresystems in eine heterogene Infrastruktur, wie man sie in mobil-verteilten Nutzungsszenarien vorfindet. Der Austausch mit Drittanwendungen gewährleistet, dass die Kooperationsumgebung nicht isoliert von der Netzwerkumgebung betrieben wird und Medienbrüche beim Austausch von Daten mit externen Anwendungen vermieden werden. Darüber hinaus erlaubt eine offene Architektur den Benutzern, das System an ihre Bedürfnisse anzupassen, indem sie dieses mit eigenen Komponenten anreichern und funktional erweitern. Dies schließt die Einbindung externer Dienste mit ein (Szenario 4.4).

Eine solche offene Architektur darf aufgrund der geforderten Flexibilität nicht monolithisch aufgebaut sein. Sie muss modular aus austauschbaren Komponenten bestehen, die über wohldefinierte Schnittstellen kommunizieren. Nur so kann die Funktionalität des Systems verändert werden, ohne tief in die Gesamtarchitektur einzugreifen. Besonders in Hinsicht auf die Nachhaltigkeit von Kooperationssystemen ist die Modularität und eine offene Architektur eine unverzichtbare Eigenschaft. Diese erlaubt es auch Dritten, die Systeme kontinuierlich an ihre Anforderungen anzupassen und so eine fortwährende Nutzung zu gewährleisten.

Diese Forderungen werden durch die Unfähigkeit monolithischer und zentralisierter CSCW-Umgebungen, sich an die Mobilität der Benutzer anzupassen, noch unterstrichen. Aufgrund einer oft monolithischen Architektur sind sie schwer an die Gegebenheiten eines mobilen Einsatzgebietes anzupassen. Dieses Problem wird durch die Abhängigkeit von einer zentralisierten Persistenz zusätzlich verstärkt. Da die Eignung der bestehenden CSCW-Architekturen für mobile Szenarien somit nicht ohne weiteres hergestellt werden kann, ist für einen nahtlosen Übergang zwischen mobilen und zentralisierten CSCW-Systemen die Bereitstellung entsprechender Schnittstellen essenziell. In Szenario 4.2 wird die Anforderung einer Kombination von zentralisierter und mobiler CSCW-Topologie verdeutlicht.

Möchten Benutzer einer zentral verwalteten Kooperationsumgebung ihre Kooperationsobjekte mitnehmen, um diese mobil weiterverarbeiten zu können, ist eine Übernahme der Kooperationsdaten von einem CSCW-Server in die mobile CSCW-Umgebung äußerst nützlich. Auf dem umgekehrten Weg bietet der Import von Kooperationsobjekten aus einer mobilen CSCW-Umgebung in ein serverbasiertes CSCW-System eine zentrale wie verlässliche Anlaufstelle für die Kooperationspartner und garantiert eine langfristige Verfügbarkeit der Kollaborationsdaten. Auch die zusätzlichen Möglichkeiten eines in eine Diensteinfrastruktur eingebundenen CSCW-Servers machen einen nahtlosen Übergang von verlässlichen zentralisierten Architekturen zu flexiblen mobil-verteilten Architekturen attraktiv (Szenario 2.6, Szenario 2.7).

Die Forderung nach *Offenheit* in mobil-verteilten Kooperationssystemen liegt somit in mehreren Faktoren:

- Der *evolutionäre Übergang von monolithischen, zentralisierten Kooperationsystemen zu modularen, mobil-verteilten Kooperationssystemen* zur besseren

Unterstützung der Mobilität der Benutzer ohne einen Bruch in deren Kooperationsumgebungen,

- die *nahtlose Integration externer Dienste* in die Kooperationsumgebung und
- der nahtlose *Transport von Objekten zwischen den Kooperationssystemen*.

Das Erfüllen der Forderung nach einer *Offenen Architektur* erlaubt dem Kollaborationssystem somit eine externe Konsistenz, die dem Benutzer die nahtlose Nutzung externer Dienste ohne Medienbrüche ermöglicht. Die vorgestellten Bereiche der *Vernetzung und Sichtbarkeit*, der *Kontextualisierung* und der *Konsistenz und Reversibilität* bilden die zentralen Anforderungen für ein mobil-verteiltes Kooperationssystem.

Der folgende Abschnitt behandelt darauf aufbauend die grundlegenden technischen Anforderungen an ein Kooperationssystem und betrachtet diese im Spannungsfeld der Mobilität. Die mobil-spontane Zusammenarbeit baut im besonderen Maße auf Funktionen aus den klassischen Bereichen der Kommunikation, Koordination und Kooperation auf, die allerdings einer Anpassung an die Gegebenheiten der mobilen Nutzung bedürfen.

3.4 Neue Rahmenbedingungen computergestützter Kooperationsdienste



Die Zusammenarbeit im mobilen Nutzungsumfeld bedarf einer ähnlichen technischen Unterstützung der Kommunikations- und Koordinationsprozesse wie die klassische Kooperation in zentralisierten CSCW-Architekturen. Im Spannungsfeld mobil-spontaner Kooperation werden jedoch angepasste und weitergehende Mechanismen benötigt, um eine nahtlose Nutzung der Kooperationsumgebung zu gewährleisten.

Für eine Unterscheidung zur klassischen Kooperation in zentralisierten CSCW-Systemen wird an dieser Stelle der Begriff der *Kollaboration* eingeführt. Zwar ist der Begriff *Kollaboration* im Deutschen mit einer militärischen Bedeutung belegt¹, wird aber im englischen Sprachraum synonym für den Begriff der Kooperation verwendet.

¹ „Kollaboration [...]: aktive Unterstützung einer feindlichen Besatzungsmacht gegen die eigenen Landsleute“, aus: DUDEN-Fremdwörterbuch, Bd. 5, Mannheim 1997.

Im Folgenden wird der Begriff der *Kollaboration* verwendet, um die spontanen und dynamischen Qualitäten der Zusammenarbeit in einem mobil-verteilten Nutzungs-umfeld zu betonen und von denen klassischer Kooperationssysteme abzugrenzen. Er wird daher mit folgender Bedeutung belegt:

Kollaboration ist die spontane oder geplante Zusammenarbeit zwischen zwei oder mehr Personen, die aufbauend auf technischen Unterstützungsfunctionen zur Kommunikation, Koordination und Kooperation geschieht. Diese Interaktion findet zwischen vernetzten Partnern in virtuellen und realen Räumen basierend auf Kooperationsobjekten (Dokumenten) statt. Die Kollaboration ist dabei durch einen stark spontanen und dynamischen Charakter geprägt.

Kollaboration wird in diesem Sinne als interaktiver Prozess betrachtet, den es durch die Kooperationsumgebung zu unterstützen gilt. Dabei muss Kollaboration nicht zielgerichtet sein oder einem bestimmten Zweck dienen – der spontane Charakter der Zusammenarbeit in mobilen Nutzungsszenarien kann die Zusammenarbeit zwischen den Benutzern sogar zu einem nebenläufigen Prozess machen, der den eigentlichen Zweck eines Treffens flankiert.

Dabei bilden *Kommunikation*, *Koordination* und *Kooperation* die Grundlage des Kollaborationsprozesses, die durch entsprechende Werkzeuge und Protokolle gewährleistet werden muss. Anders als bei dem 3K-Modell [Teufel et al., 1995] werden Kommunikation, Koordinierung und Kooperation als ein hierarchisches Technikmodell betrachtet, in dem jede Ebene die Grundlage für die ihr jeweils übergeordnete bildet (siehe Abbildung 3.5). Die Kommunikation ist somit die Basis für die Koordinierungsprozesse, auf die wiederum viele Kooperationsfunktionen aufsetzen. Mit Hilfe der Kooperationsfunktionen findet in der obersten Ebene der Kollaboration die Unterstützung einer spontanen und dynamischen Zusammenarbeit in einer mobil-verteilten Kooperationsumgebung statt. Jede Ebene greift dabei auf alle untergeordneten Ebenen zu.

Viele der aus Kapitel 2 hervorgehenden Anforderungen an die mobil-verteilte Kooperationsumgebung lassen sich entlang der technischen Unterstützung von Kommunikation, Koordination und Kooperation anordnen. Diese gilt es somit an die neuen Anforderungen einer mobil-verteilten Architektur anzupassen. Darauf aufbauend wird dem beiläufigen und spontan-dynamischen Charakter der Kollaboration in der Ebene der Kollaborationsunterstützung gezielt Rechnung getragen.

Die im Folgenden betrachteten technischen Grundfunktionalitäten für die Kollaboration im mobilen Nutzungsumfeld legen somit die Basis für eine Nutzung der mobil-verteilten Kooperationsumgebung. Während die in Abschnitt 3.3 vorgestellten Anforderungen aus der neuartigen Nutzung von Kooperationsumgebungen im mobilen Umfeld abgeleitet sind (*Handlungsebene*), handelt es sich in diesem Abschnitt um Mechanismen, die den Benutzern zumeist verborgen bleiben, aber dennoch die Basistechnologie für die Nutzung einer Kooperationsumgebung in mobilen Szenarien bilden (*Umsetzungsebene*). Eine scharfe Trennung zwischen Anforderungen auf Handlungsebene und Umsetzungsebene gelingt nicht in allen Aspekten der Anforderungsbestimmung, sie bildet aber dennoch einen geeigneten Ansatz, um technikbe-

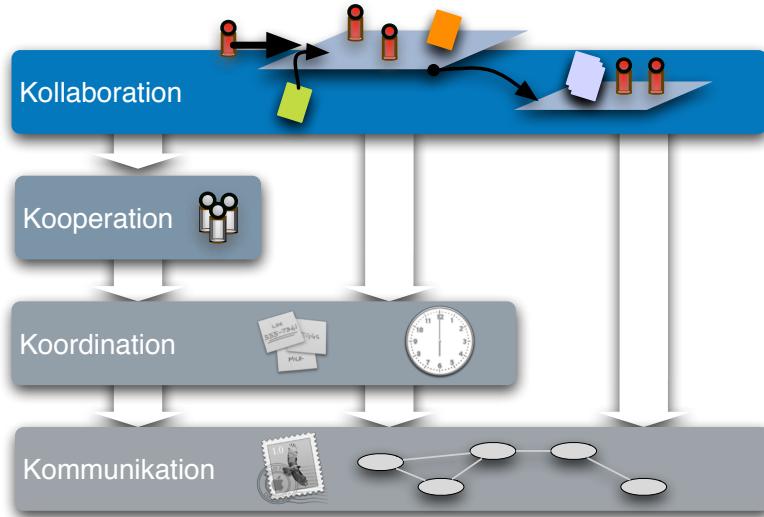


Abbildung 3.5: Kollaboration baut auf die Ebenen der Kommunikation, Koordinierung und Kooperation auf [vgl. Eßmann und Hampel, 2005a].

dingte Anforderungen von denen zu trennen, die der mobilen Nutzung von Kooperationsystemen geschuldet sind.

Als Beispiel wird mit der Forderung der *Versionierung* der Kooperationsobjekte aus technischer Sicht eine wichtige Anforderung an das System gestellt, um Konflikte zwischen redundant gespeicherten Objekten aufzudecken. Gleichzeitig stellt sie aber aus Sicht der mobilen Nutzung einen wichtigen Mechanismus für die kooperative Bearbeitung von Dokumenten ohne Wahrnehmung der Handlungen der Kooperationspartner dar. Im Gegensatz dazu ist die *Technologie zur spontanen Vernetzung* z. B. eine rein technische Anforderung, die aufgrund ihres spontanen und dynamischen Charakters die Anforderungen an die Kommunikation innerhalb der Kooperationsumgebung mitprägt.

Die technischen Rahmenbedingungen eines mobilen Nutzungsumfelds stellen somit gewisse Anforderungen an eine mobil-verteilte Kooperationsumgebung. Im Folgenden werden die beschriebenen Szenarien bezüglich ihrer besonderen Bedürfnisse an diese technische Infrastruktur analysiert.

3.4.1 Kommunikation

Kommunikation bildet die Grundlage jeder Zusammenarbeit. Erst durch eine stabile Kommunikationsinfrastruktur zwischen den Kooperationspartnern, bzw. deren Geräten, ist eine Kooperationsunterstützung überhaupt möglich. In synchronen Kooperationsformen muss diese Kommunikation nicht nur verlässlich sondern auch hinreichend schnell erfolgen. Zusätzlich besteht bei einer Zusammenarbeit in spontanen

Zusammenkünften häufig der Wunsch nach einem sofortigen Einstieg in die Kooperation, ohne zuvor die genutzten Hilfsmittel konfigurieren zu müssen. Der Konfigurationsaufwand würde hier den Nutzen einer computergestützten Kooperationsumgebung übersteigen und somit unter Umständen zu einem Nutzungsverzicht seitens der mobilen Benutzer führen. Ein Ausweichen auf alternative Hilfsmittel (z. B. Zettel und Stift) hätte bei einer späteren Rückkehr in die computergestützte Kooperationsumgebung einen Medienbruch zur Folge, da die Kooperationsergebnisse erst wieder in diese übertragen werden müssten.

Die grundlegenden Techniken für die Kommunikation in der Kooperationsumgebung müssen daher mobil-spontanen Kooperationsszenarien gewachsen sein und ohne Eingreifen der Benutzer die mobilen Geräte ad hoc vernetzen. Diese automatische Etablierung von Kommunikationsstrukturen schließt dabei auch eine Verknüpfung der bereitgestellten Dienste ein, die die höheren Ebenen der Kooperationsunterstützung bereitstellen.

Vernetzung

Der Vernetzung kommt als Basistechnologie für die Kommunikation zwischen den beteiligten Computern eine zentrale Rolle bei der Etablierung einer verteilten Kooperationsumgebung zu. Aufbauend auf die durch das Netzwerk zur Verfügung gestellte Infrastruktur verbinden sich die von den Geräten der Teilnehmer bereitgestellten Dienste zu der verteilten Kooperationsumgebung. Verteilte Anwendungen nutzen für die Kommunikation zwischen den Diensten oftmals Protokolle in der Anwendungsschicht, welche ihrerseits die tiefer gelegenen Netzwerkprotokolle als Transportschicht nutzen. Wichtig ist, dass die Protokolle in jeder Schicht dem besonderen Charakter des mobil-verteilten Nutzungsumfeldes gerecht werden.

Spontane Vernetzung Für einen hürdenlosen Einstieg in die Kooperationsumgebung muss die zugrunde liegende Netzwerk-Infrastruktur schnell und ohne Benutzereingriff etabliert werden können. Diese Anforderung wird unter dem Begriff der „*Spontanen Vernetzung*“ zusammengefasst. Szenario 2.1 verdeutlicht zentrale Aspekte der spontanen Vernetzung im Rahmen einer mobil-verteilten Kooperation.

Primär beinhalten die Anforderungen der *spontanen Vernetzung* die Fragestellung nach einer transparenten Vernetzung und sofortigen Verfügbarkeit der Kooperationsumgebung aus Sicht der Benutzer (Szenario 1.6, Szenario 2.1). Daher muss diese Infrastruktur bei einem Zusammentreffen von Kooperationspartnern ad hoc und ohne deren Eingreifen gebildet werden. In Umgebungen, in denen nicht auf bestehende Netzwerkstrukturen zurückgegriffen werden kann, bedeutet dies, diese selbstständig zu errichten und die Instanzen der verteilten Anwendung mittels der anwendungsspezifischen Protokollsicht zu koppeln.

Sollten bereits existierende Netzwerkinfrastrukturen verfügbar sein, hat die Kooperationsumgebung die Möglichkeit, diese für die Vernetzung der verteilten Anwendung zu nutzen. Allerdings sind etablierte Netzwerkinfrastrukturen oft nur für beschränkte Nutzerkreise verfügbar (z. B. Kunden eines Netzbetreibers) und schlie-

ßen somit einige potentielle Partner von der Teilnahme an der Kooperation aus. Ein weiteres Problem der Nutzung solcher zentral administrierten Netzwerkinfrastrukturen für eine flexible Zusammenarbeit ist die Reglementierung der in dem Netzwerk nutzbaren Dienste über das Blockieren der seitens des Betreibers nicht erwünschten Kommunikationsprotokolle.

Derartige Einschränkungen lassen auch hier autarke Netzwerkinfrastrukturen zwischen den Kooperationsteilnehmern sinnvoll erscheinen. Diese können bei bestehender Möglichkeit existierende Netzwerkstrukturen wie in Szenario 2.3 einbinden, um in diesen verortete Dienste zu nutzen. In einem solchen Szenario treten Benutzer mit Zugang zu dem restriktiven Netz als eine Art Mittelsmänner auf, um den anderen Teilnehmern Zugang zu diesem Netz zu gewähren.

Die spontane Etablierung der Kommunikation in einer Kooperationsumgebung ist aber nicht auf Netzwerkstruktur und die Kopplung der verteilten Anwendung beschränkt. Gleichzeitig müssen die benötigten Kooperationsfunktionen eingerichtet und zur Verfügung gestellt werden (Szenario 2.1). Dies ist insbesondere in verteilten Umgebungen mit einem Aushandeln der Verteilung der benötigten Dienste verbunden.

Sind alle benötigten Dienste verfügbar, kann die Zusammenarbeit über diese technische Infrastruktur erfolgen. Da die Benutzer sich nicht in einem starren Szenario bewegen und keine festen Strukturen existieren, müssen auch die Strukturen der Kooperationsumgebung spontan etabliert werden (Szenario 2.3). Erst dann können die Kollaborationspartner miteinander kommunizieren und eine gemeinsame Sitzung starten.

Robustheit Da weder Netzwerkstruktur, Diensteinfrastruktur noch die Kooperationsstruktur in mobilen Nutzungsszenarien als stabil betrachtet werden können, muss die Kooperationsumgebung flexibel und robust auf Änderungen auf allen Ebenen der Kooperation reagieren können. Dies liegt insbesondere an der Instabilität der zugrunde liegenden Netzwerkverbindungen. Durch Bewegung der Benutzer kommt es häufig und meist ohne Vorwarnung zu Verbindungsabbrüchen zwischen den Kollaborationspartnern. Zudem geht in mobilen Szenarien kooperatives und individuelles Arbeiten nahtlos ineinander über. Um die Kollaborationsdaten auch in unverbundenen Situationen verfügbar zu haben und die Kooperationsumgebung nicht in bestimmten Nutzungskontexten unbrauchbar zu machen, muss ein Weiterarbeiten auch für von der Gruppe isolierte Kollaborationspartner möglich sein (Szenario 3.2).

Aus den instabilen Netzwerkstrukturen in einer mobilen Umgebung leitet sich auch die Forderung nach einer flexiblen Wahl der Übermittlungsmethoden für Daten der Kooperationsumgebung ab (Szenario 3.2). Anders als in fest strukturierten und zentral administrierten Netzwerken ist die Verbindungsqualität selten gleich bleibend sondern variiert in der Übermittlungskapazität. Dies hat zur Folge, dass eventuell nicht alle Daten über eine Netzwerkverbindung gesandt werden können.

Da die Erreichbarkeit einzelner Knoten in der Netzwerkumgebung nicht garantiert werden kann, dürfen essenzielle Dienste nicht nur von einer zentralen Stelle

angeboten werden. Daraus ergibt sich die Forderung nach *Unabhängigkeit von zentralisierten Diensten* (Szenario 4.4). Als eine Daumenregel kann gesagt werden, dass mit der Bedeutung eines Dienstes für die Funktionsfähigkeit der Kooperationsumgebung dessen Verteilung im Netzwerk steigen muss. Auch unter dem Aspekt der Robustheit müssen daher Dienste, ohne welche die Kooperationsumgebung nicht mehr betriebsfähig ist, auf jedem Knoten vorhanden und aktivierbar sein. Nur so kann der Knoten im unverbundenen Fall autark weiter betrieben werden.

Nachrichtensystem Über die Protokolle der Netzwerkschicht tauschen die Instanzen der mobil-verteilten Kooperationsumgebung Nachrichten über lokale und netzweite Ereignisse aus, um sich untereinander abzugleichen und so die Zusammenarbeit der Benutzer zu ermöglichen. Auch die Benutzer senden sich gegenseitig Nachrichten, um miteinander zu kommunizieren. Diese Nachrichtenübermittlung muss an den dynamischen Charakter der mobil-spontanen Nutzungsszenarien angepasst sein. Nachrichten können z. B. nicht wie in einem Client-Server-System zentral hinterlegt werden. Eine direkte Übermittlung an den Empfänger über eine Punkt-zu-Punkt-Verbindung ist wegen einer weitläufigen und instabilen Vernetzung oder einer zeitversetzten Zusammenarbeit auch nicht immer möglich. Dies gilt insbesondere für Systemnachrichten über Ereignisse in der Kooperationsumgebung. Daher muss sowohl das allgemeine Nachrichtensystem wie auch die Ereigniskontrolle innerhalb der mobil-verteilten Kooperationsumgebungen dezentral und asynchron realisiert werden, wie aus Szenario 1.2 und Szenario 1.3 ersichtlich.

Da aufgrund der Mobilität der Benutzer und der daraus resultierenden Dynamik der Struktur in der Kooperationsumgebung, der Kreis der Adressaten für die Nachrichten anders als in statischen Kooperationsumgebungen meist nicht fest vorgegeben und durch den Kontext der Anwender innerhalb der Kooperationsumgebung definiert ist (vgl. Abschnitt 3.3.2), müssen die Empfänger während des Nachrichtenversandes ermittelt werden. Dies gilt insbesondere für Nachrichten an Benutzergruppen, die in ad hoc gegründeten Kollaborationssitzungen nicht durch explizite administrative Handlungen festgelegt werden. Daher ist es wichtig, einen flexiblen Adressatenkreis für sämtliche Nachrichten in der Kooperationsumgebung zu unterstützen (Szenario 1.2).

Objekte/Medien

Kommunikation in virtuellen Wissensräumen findet üblicherweise über die enthaltenen Medien statt [vgl. Keil-Slawik et al., 2005]. Textdokumente sind nur eine mögliche Ausprägung der mannigfaltigen Medientypen. Eine objektorientierte Betrachtungsweise lässt eine einheitliche Behandlung aller Elemente innerhalb der Kooperationsumgebung zu und erlaubt einen einheitlichen Umgang mit allen zur Zusammenarbeit benötigten Daten.

Dieser Abschnitt beschäftigt sich mit der Verwaltung der Objekte innerhalb der Kooperationsumgebung. Da die Objekte Grundlage des Kollaborationsprozesses sind, ist es wie bei allen kritischen Diensten wichtig, die Verwaltung der Objekte über die

beteiligten Knoten zu verteilen. Bis auf einige wenige Ausnahmen wird dieser grundlegende Dienst auf nahezu allen Knoten laufen müssen, um die Offline-Verfügbarkeit der Kooperationsumgebung zu garantieren. Dabei gilt es, sowohl den Dienst als auch die von ihm verwalteten Objekte nach einem Muster zu verteilen, so dass eine optimale Verfügbarkeit der Kooperationsobjekte unter Berücksichtigung der Netzwerk- und Speicherressourcen der beteiligten Knoten gewährleistet ist.

Distribution Um eine Verteilung der Objekte in der Kooperationsumgebung zu ermöglichen, müssen Objekte zwischen den Geräten der Kooperationspartner migrieren können (Szenario 1.2). Damit das Objekt nach der Migration auf dem Zielknoten denselben Zustand aufweist, den es auf dem Ursprungsknoten hatte, müssen alle Eigenschaften des Objekts erhalten bleiben. Im Fall einer redundanten Speicherung der Objekte innerhalb der Kooperationsumgebung, kann die Replikation der Objekte als eine Migration ohne ein Löschen des Objektes vom Ursprungsknoten betrachtet werden. Wichtig bei einer Replikation ist zusätzlich, dass die Repliken des Objektes miteinander gekoppelt bleiben und Änderungen an dem Objekt zwischen diesen synchronisiert werden (siehe Abschnitt 3.3.3).

Für die effiziente Verbreitung von Objekten in verteilten Kooperationsumgebungen, kann eine epidemische Distributionsstrategie gewählt werden (Szenario 1.2). Ein solches Distributionsmuster bezieht oft auch unbeteiligte Knoten ein, die nicht zu den eigentlichen Zielknoten gehören. Sie speichern das Objekt zwischen und liefern es bei Gelegenheit an die eigentlichen Zielknoten aus. Erreicht ein Objekt die Empfänger nicht, muss es nach festen Kriterien verworfen werden können, da es sonst die zwischenspeichernden Knoten langfristig belastet.

Bei einer dokumentenzentrierten Zusammenarbeit in virtuellen Wissensräumen nutzt eine Kollaborationsgruppe stets einen festen gemeinsamen Arbeitsbereich für die Strukturierung ihrer Kooperationsobjekte. Die verteilte Speicherung dieses gemeinsamen Arbeitsbereiches in mobil-spontanen Kooperationsszenarien führt schnell zu Problemen bei der Persistenz der Kollaborationsergebnisse. Trotz der Kurzlebigkeit der Gruppenstrukturen in spontanen Kollaborationen darf sich der gemeinsame Arbeitsbereich mit den Kooperationsergebnissen nicht einfach auflösen sobald sich die Gruppe auflöst. Die in einer spontanen Kollaboration entstandenen Dokumente müssen also persistent bleiben, damit auf die Kooperationsergebnisse später noch einmal zurückgegriffen werden kann (Szenario 2.7).

Da sich die spontan entstandenen Kollaborationsgruppen nicht zwangsläufig in der selben Konstellation wieder zusammenfinden oder die an der Kollaborationssitzung beteiligten Benutzer die Ergebnisse eventuell individuell weiterbearbeiten wollen, müssen die betroffenen Objekte auf die persönlichen Geräte aller Beteiligten separat gespeichert werden (Szenario 2.6).

Eine weitere Möglichkeit zur Erhöhung der Verfügbarkeit der verteil bearbeiteten Kooperationsobjekte ist die zusätzliche Speicherung auf einem zentralen CSCW-Server. Dieser bildet eine verlässliche Instanz in der verteilten Umgebung, die eine höhere Verfügbarkeit verspricht als mobile Knoten. Allerdings darf dieses Vorge-

hen nicht zu einer erneuten Abhängigkeit von zentralen Diensten führen. Die Daten sind daher während des Kooperationsprozesses weiterhin redundant auf den mobilen Geräten vorzuhalten. Die Möglichkeit einer derartigen Integration von zentralen CSCW-Servern erlaubt zudem eine nahtlose Verknüpfung verteilter und zentralisierter Kooperationsumgebungen (Szenario 4.2).

Konsistenz Da die Kooperationsobjekte in dem Kooperationsnetzwerk unter Umständen auf mehrere Knoten repliziert werden, ist ein Überwachen konkurrierender Zugriffe auf die Objekte unerlässlich, um deren Konsistenz zu wahren (Szenario 1.6). Aufgrund der Tatsache, dass die Instanzen der Kooperationsumgebung in mobil-verteilten Szenarien nicht ständig vernetzt sind, können konkurrierende Zugriffe auf ein Objekt eventuell erst zeitlich verzögert festgestellt werden. Bei der nächsten Kontaktaufnahme der beteiligten Knoten müssen demnach die replizierten Instanzen eines Objektes auf einen Konflikt überprüft werden. Diese Problematik des konkurrierenden Zugriffs auf schwach gekoppelten Kooperationsobjekte aufgrund der nur sporadisch verfügbaren Vernetzung, widmet sich eingehend der Abschnitt 3.3.3.

3.4.2 Koordination

Neben der technischen Unterstützung der Kommunikation ist die der Koordination der Teilnehmer in einem mobilen Nutzungsumfeld wichtiges Kriterium für eine gezielte Zusammenarbeit. Zu den Koordinationsmechanismen gehören insbesondere Werkzeuge zur Abstimmung und Organisation der Kollaboration. Diese sind insofern im mobilen Nutzungsumfeld wichtig als das den Benutzern, die eine mobil-verteilte Kooperationsumgebung selbst verwalten müssen, eine Hilfestellung durch Dritte oft nicht zur Verfügung steht.

Die Nutzung der mobil-verteilten Kooperationsumgebung findet im Gegensatz zu klassischen CSCW-Systemen zumeist am selben Ort und zur gleichen Zeit statt. Die Unterstützung einer einfachen Methode für die Terminbestimmung von Präsenztreffen, wie in Szenario 1.4 dargestellt, gehört daher zu den zentralen Anforderungen an mobilitätsunterstützende Kooperationssysteme. Eine Einbettung dieser Abstimmungsmechanismen in die Kooperationsumgebung erlaubt dabei einen Bezug auf bereits existierende Nutzerstrukturen und hilft Medienbrüche durch ein Verlassen der Arbeitsumgebung zu vermeiden. Dazu müssen die Koordinationsmechanismen entlang des konzeptionellen Designs der Kooperationsumgebung implementiert werden. Beispiele sind diesbezüglich *Gruppenkalender* für die Koordinierung der Zusammenarbeit in der Gruppe wie auch *Persönliche Kalender* für individuelle Planungsprozesse in derselben Umgebung (Szenario 1.4). Der Zugriff auf die Gruppenkalender muss an die Rechteverwaltung gekoppelt und auch im unverbundenen Fall gewährleistet sein. Wegen der sporadischen Vernetzung der Kooperationsteilnehmer müssen die Termine zwischen den Teilnehmern als Gruppennachrichten zwischen den diesen epidemisch verteilt werden (Szenario 1.4). Dies garantiert eine schnelle Verteilung der Termine an alle Teilnehmer (vgl. Abschnitt 3.4.1).

Neben der Terminabsprache für geplante Kooperationssitzungen muss die Verwaltung der Gruppenstruktur an die Besonderheiten mobil-spontaner Kollaboration angepasst sein. Im Folgenden werden gezielt Mechanismen besprochen, die für die Mobilität der Nutzer von essentieller Bedeutung sind. Generelle Aspekte der Gruppenverwaltung, wie sie auch in zentralisierten CSCW-Systemen zu finden sind, sollen in diesen Zusammenhang dabei nicht behandelt werden.

Koordination – Rollen und Rechte

Gruppenstrukturen bilden stets die Berechtigungen der Mitglieder an die der Gruppe gehörenden Ressourcen ab. Daher ist die Gruppenstruktur eng mit den Berechtigungen der Benutzer innerhalb der Kooperationsumgebung verknüpft. Während in den meisten Fällen eine gruppenbasierte Verwaltung von Berechtigungen bezüglich der Kooperationsobjekte ausreichen mag, müssen die Berechtigungsmechanismen auch komplizierte Berechtigungsstrukturen abbilden können. Dies ermöglicht den Benutzern die Gestaltung einer an ihre Bedürfnisse angepassten Arbeitsumgebung. Eine granulare Rechteverwaltung (Szenario 1.3, Szenario 4.1) ist daher unabdingbar für eine flexible und universelle Nutzung der Kooperationsumgebung, die sich den Bedürfnissen der Benutzer anpasst.

Da die Kollaboration in mobil-verteilten Kooperationsumgebungen zuweilen in einzelne Präsenzsitzungen aufgeteilt abläuft, sollten auch Berechtigungen zeitlich an die Dauer einer Sitzung gebunden werden können. Als Beispiel sei hier die Gastrolle genannt (vgl. Abschnitt 3.3.2). Die Gastrolle ist gerade in spontanen Kooperations-szenarien eine wichtige Voraussetzung für eine einfache Einbindung externer Kooperationspartner, die nicht Teil der normalen Kooperationsgruppe sind (Szenario 1.7, Szenario 1.8). Aus diesem Grund müssen die Mechanismen für eine einfache Handhabung der Gastrolle bereits in die Grundkonzeption des Rechtesystems eingebunden sein. Eine Berücksichtigung der Rollen der Kooperationsmitglieder (von denen ein Kooperationsmitglied mehrere gleichzeitig haben kann – einige davon eventuell zeitlich begrenzt) sollte neben einem benutzer- und gruppenzentrierten Rechtesystem möglich sein (Szenario 4.2).

Die Zugriffskontrolle stellt eine verteilte Architektur vor besondere Herausforderungen. Die Basismechanismen der mobil-verteilten Kooperationsumgebung müssen einen Zugriffsschutz auch ohne zentrale Sicherheitsinstanz gewährleisten können (Szenario 4.4). Besonders bei einer Verteilung der Objekte über die Geräte aller beteiligten Benutzer ist sicherzustellen, dass die Objekte vor dem Zugriff durch nicht autorisierte Benutzer geschützt sind, auch wenn sie zeitweise – z. B. in Folge einer epidemischen Verbreitung (vgl. Abschnitt 3.4.1) – auf deren Geräten gespeichert werden (Szenario 1.2). Weil die Benutzer die Software der Kollaborationsanwendung auf ihrem eigenen Gerät manipulieren könnten, muss der Zugriffsschutz an dem Objekt selber verankert sein und darf nicht von Zugriffskontrollmechanismen der Anwendung abhängen. Dies kann nur durch eine geeignete Verschlüsselung der Kooperationsobjekte geschehen.

Objekte können innerhalb der Kooperationsumgebung auch Geräte von fremden Benutzern durchlaufen, daher ist die Sichtbarkeit der Objekte für die Benutzer abhängig von dem Adressatenkreis zu gestalten. Dabei ist es möglich, dass der Kreis der Adressaten zum Zeitpunkt der Versendung des Objektes noch nicht bekannt ist und nur anhand eines Benutzerprofils benannt werden kann. Dies trifft z. B. bei Rundschreiben wie Einladungen zu einer Kooperationssitzung zu. Die Sichtbarkeit von Kooperationsobjekten muss deshalb auch in Abhängigkeit von Benutzerprofilen definiert werden können (Szenario 1.2).

Da sich Gruppenstrukturen in mobilen Kooperationssystemen durch eine hohe Dynamik auszeichnen, benötigen die Initiatoren einer Kooperation effiziente Werkzeuge für die Gruppenverwaltung. Um ein unkontrolliertes Wachstum der Gruppen zu verhindern und daraus resultierende unübersichtliche Gruppenstrukturen zu vermeiden, müssen Kooperationsgruppen in ihrer Größe beschränkt werden können. Wartelisten ermöglichen hier ein Nachrücken von neuen Kooperationspartnern, wenn ein Gruppenmitglied wie in Szenario 1.7 die Kollaboration verlässt. Die Gruppenadministratoren müssen die Möglichkeit besitzen, diese Wartelisten zu verwalten, um den Prozess der Aufnahme neuer Benutzer in die Gruppe beeinflussen zu können.

Für eine flexible Erweiterung der Kooperationsgruppe ist in mobil-spontanen Kollaborationsszenarien die *Gastbenutzer-Rolle* ein unverzichtbares Hilfsmittel. Sie erlaubt Kooperationspartnern eine temporäre Teilnahme an der Kollaborationssitzung, ohne diese gleich als vollwertiges Mitglied der Gruppe verwalten zu müssen (Szenario 1.7, Szenario 1.8). Die Rolle des Gastbenutzers ist für den Zeitraum der aktuellen Kollaborationssitzung gültig und mit Einschränkungen bezüglich der Zugriffsrechte auf die Objekte der Gruppe verbunden. In manchen Kooperationsszenarien ist es sinnvoll unterschiedliche Gastrollen vergeben zu können, wie z. B. *Berater* oder *Beobachter*. Dazu muss die Gastrolle durch die Gruppenadministratoren frei konfigurierbar sein und diese Konfiguration als Voreinstellung für eine bestimmte Gastrolle hinterlegt werden können.

Für die Verwaltung von Kooperationsgruppen in einer dynamisch vernetzten Kooperationsumgebung ist die netzweite Bekanntgabe einer Gruppe nach deren Gründung eine wichtige Funktion, um potentielle Kooperationspartner auf die Gruppe aufmerksam zu machen (Szenario 1.2). In mobil-verteilten Kooperationsumgebungen fehlt zumeist ein zentrales Verzeichnis der verfügbaren Kooperationsgruppen, in dem Kooperationswillige nachschlagen könnten. Einträge in einem solchen zentralen Verzeichnis sind aufgrund der Dynamik der Gruppenstrukturen oft veraltet – eine spontan gegründete Kooperationsgruppe existiert bereits nicht mehr oder es kann keine Netzwerkverbindung mehr zu den Gruppenmitgliedern aufgebaut werden. Alle Teilnehmer der Kooperationsumgebung müssen daher mittels geeigneter dezentraler Dienste fortwährend über die verfügbaren Kollaborationsgruppen in ihrer Umgebung informiert werden.

Mobil-spontane Kooperationen werden oftmals aus dem Kontext eines informellen Zusammentreffens heraus ins Leben gerufen. Erst im Anschluss werden sie bei Bedarf in längerfristige und formelle Kooperationsformen überführt. Somit ist die Möglichkeit zur Integration spontan etablierter Gruppen in langfristige und organisierte

Gruppenstrukturen unverzichtbar (Szenario 2.7). Diese nachträgliche Integration erlaubt es den Benutzern aus der spontanen Kollaboration heraus, erst nachträglich eine Entscheidung über deren Organisationsform (*kurzfristig, spontan* oder *langfristig, organisiert*) zu treffen. Die im Nachhinein gewählte Gruppenstruktur muss somit nicht der Struktur entsprechen, welche die Gruppe zu Beginn der Kollaboration aufwies.

Soll die Verfügbarkeit der Ressourcen einer Gruppe erhöht werden, die in der mobil-spontanen Kooperationsumgebung gegründet wurde, ist es sinnvoll, Teile der mobil-verteilten Gruppenstruktur mit der eines zentralisierten CSCW-Systems zu synchronisieren (Szenario 4.4). Dieses Vorgehen erlaubt den Nutzern der zentralisierten CSCW-Umgebung außerdem, ihre zentral verwalteten Kooperationsgruppen auch im mobilen Umfeld benutzen zu können. Eine solche Möglichkeit verhindert zudem einen Bruch bei der Nutzung einer mobil-verteilten und einer institutionellen zentralisierten CSCW-Umgebung (vgl. Abschnitt 3.3.3).

3.4.3 Kooperation

Um kollaborative Prozesse in allen Facetten unterstützen zu können, muss die mobil-verteilte Kooperationsumgebung *technische Mechanismen für die Gestaltung der Kooperation* bereitstellen. Kooperationsmechanismen bieten eine Unterstützung der organisierten Zusammenarbeit, die über die reine Koordination der Gruppe hinausgeht. Sie regeln die Nutzung der gemeinsamen Ressourcen und helfen bei deren Strukturierung. Da die Benutzer in einer mobil-verteilten Kollaborationssituation oft autonom von bestehenden Infrastrukturen agieren, muss die Kollaboration *selbstorganisiert* ablaufen. Neben der Verwaltung der Kooperation stellen sich aber aufgrund des spontanen und verteilten Charakters der Kooperationsumgebung zusätzlich Anforderungen an das zugrunde liegende Objektmodell. Im Folgenden werden zunächst die Anforderungen an die Zugriffsmechanismen *Gruppenverwaltung, Berechtigungen* und *Zugriffsschutz* vorgestellt. Daran anknüpfend, werden die Anforderungen an das Objektmodell mit *Personalisierung* und *Objektbehandlung* betrachtet.

Gruppen Schon im Abschnitt *Koordination* (3.4.2) wurde die Gruppenverwaltung als Mittel zur Koordinierung der Kollaborationspartner behandelt. In diesem Abschnitt wird darüber hinausgehend die Bedeutung der Gruppenstruktur für die gezielte dokumentenzentrierte Arbeit betrachtet. Grundlage der Organisation einer Kooperation ist die Schaffung einer Gruppenstruktur, über die potentielle Partner in die Kooperation eingebunden oder von ihr ausgeschlossen werden können. Diese Gruppenstruktur muss den Anforderungen seitens der Mobilität der Kollaborationspartner genügen und auch komplexe Kooperationsstrukturen abbilden können (vgl. Szenario 4.1). Das Gruppenkonzept muss sich dabei durch alle Bereiche der mobil-verteilten Kooperationsumgebung ziehen und wie die Kooperationsobjekte auch auf allen beteiligten Knoten durchgehend und konsistent verfügbar sein (vgl. Abschnitt 3.3.3).

Die Gruppen in einer mobil-verteilten Kooperationsumgebung müssen Mechanismen zum Schutz ihrer Ressourcen vor dem Zugriff Dritter besitzen. Diese Anforderung ist umso wichtiger, da es keine übergeordnete Instanz gibt (z. B. ein abgeschottetes Serversystem), welche die Ressourcen einer Gruppe nach außen schützt. Diese Abgeschlossenheit der Gruppe nach außen ist ein wichtiger Aspekt für das Vertrauen der Benutzer in die Kooperationsumgebung (Szenario 2.2). Die zugrunde liegenden Zugriffsmechanismen sind dabei an die Bedürfnisse der gruppenbasierten Arbeit anzupassen und sollten eine Kooperation der Gruppenmitglieder untereinander nicht behindern.

Die Unterstützung der Kooperationsprozesse in der Gruppe muss auch in die strukturelle Konzeption der Kooperationsumgebung einfließen. Den Mitgliedern einer Gruppe muss ein gemeinsamer Arbeitsbereich zur Verfügung stehen, in dem die Gruppe ihre gemeinsamen Objekte verwalten kann (Szenario 1.3). Die Zugriffsbe rechtigungen für diesen Arbeitsbereich sind an die Struktur der Gruppe zu koppeln. Um jedem Gruppenmitglied auch parallel ein individuelles Arbeiten zu ermöglichen, benötigt jeder Benutzer zusätzlich einen eigenen *persönlichen Arbeitsbereich*. Dieser muss sich in derselben Kooperationsumgebung wie der Arbeitsbereich der Gruppe befinden, um einen nahtlosen Wechsel vom individuellen zum kooperativen Arbeiten und um einen einfachen Transfer von Objekten von dem einen Arbeitsbereich in den anderen zu erlauben (vgl. Szenario 1.5).

Objektbehandlung Ähnlich wie beim Zugriffsschutz muss die Verwaltung der Kooperationsobjekte auf die verteilte Objektspeicherung ausgelegt werden. Besonders die hohe Wahrscheinlichkeit kollidierender Objektversionen, bei deren verteilten und redundanten Speicherung, prägt den Umgang mit Objekten innerhalb der Kooperationsumgebung. So müssen die über die Kooperationsumgebung verteilten Repliken eines Objektes miteinander verknüpft bleiben, um konkurrierende Änderungen anderer Benutzer nachvollziehen zu können. Aus zwei Kopien eines Objektes, die auf unterschiedlichen Knoten des Kooperationsnetzwerkes gespeichert sind und dort konkurrierend bearbeitet werden, dürfen nicht aufgrund der unterschiedlichen Änderungen zwei separater Objekte resultieren – es sei denn, dies ist ausdrücklicher Wunsch der Benutzer.

Um die Konsistenz von abweichenden Objektkopien wiederherstellen zu können, müssen die Benutzer von den Differenzen zwischen abweichenden Versionen sobald wie möglich in Kenntnis gesetzt werden. Im Idealfall geschieht dies durch eine synchronisierte Sicht auf die gemeinsam genutzten Objekte. In diesem Fall werden die Kopien bei jeder Änderung untereinander abgeglichen. Die Benutzer erfahren so zeitnah von Änderungen seitens ihrer Kooperationspartner (vgl. Abschnitt 3.3.3). Leider ist diese starke Synchronisation der Objektkopien im Fall hoher Latenzen bei der Netzwerkkommunikation oder bei Verbindungsabbruch nicht länger möglich. Die technische Unterstützung der Kooperation muss daher zusätzliche Mechanismen zur Verfügung stellen, welche die *Verknüpfung der verteilten Objektkopien* auch bei schwacher Synchronisation gewährleisten kann und *Differenzen zwischen den Repli-*

ken zeitnah aufdecken kann (Szenario 1.5). Werden derartige Differenzen zwischen Objektkopien aufgedeckt, ist ein *kooperativer Bewertungsprozess als Kernmechanismus der Synchronisation* durch die Objektverwaltung zu unterstützen (Szenario 2.4, Szenario 2.5).

Um die Flexibilität der Kooperationsumgebung für unterschiedlichste Einsatzgebiete zu erhöhen, sollte die Ausgestaltung der Objekte noch weitere Eigenschaften erfüllen. Eine Erweiterung der Objekte um aktive Elemente erlaubt z. B. bei Bedarf Arbeitsprozesse (engl. Workflows) abzubilden (Szenario 1.2, Szenario 4.4). Man spricht in diesem Zusammenhang auch von *aktiven Objekten*.

Ähnlich wie die aktiven Objekte erhöhen auch *personalisierte Objekte* die Flexibilität der Kooperationsumgebung bzgl. der möglichen Nutzungsszenarien. Bei der Personalisierung von Objekten werden Eigenschaften des Objekts an einen bestimmten Benutzer oder eine Benutzergruppe angepasst. Dies kann aufgrund der Vorlieben der Benutzer oder aufgrund der Realisierung einer bestimmten Form der Kooperation geschehen (Szenario 4.4).

3.5 Zusammenfassung

Die Nutzung von Kooperationsumgebungen in mobilen Nutzungsszenarien ist gekennzeichnet von dem spontanen und nebenläufigen Charakter der Kooperation. Anders als in klassischen Kooperationsszenarien sind deren mobil-spontanen Ausprägungen selten geplant und in ihrer Konstellation von Sitzung zu Sitzung variabel (z. B. Örtlichkeit, technisches Umfeld, personelle Zusammensetzung). Dieser spontan-dynamische und nebenläufige Charakter ist es, der die Kooperation zu einer in den Tagesablauf integrierten Kollaboration transformiert. Diese Kollaboration findet in unterschiedlichen Alltagskontexten statt und bedarf spezieller Hilfsmittel, um sie in eine stärker koordinierte Form zu bringen.

Betrachtet man die Anforderungen an eine mobil-verteilte Kooperationsumgebung, die auf eine Unterstützung derartig spontan-dynamischer Zusammenarbeit zielt, zeigen sich neben den speziellen Anforderungen an die grundlegende Unterstützung von *Kommunikation, Koordination* und *Kooperation*, die Anforderungen der *Vernetzung und Sichtbarkeit*, der *Kontextualisierung* und der *Konsistenz und Reversibilität*, die den neuen Qualitäten mobil-spontaner Kooperationsformen geschuldet sind. Während erstere auch in klassischen Kooperationsumgebungen von zentraler Bedeutung sind und den Verhältnissen einer mobil-verteilten Umgebung angepasst seien müssen, folgen letztere aus dem mobil-spontanen Charakter des neuartigen Nutzungsumfeldes.

Da die Objekte der Kooperationsumgebung verteilt und unter Umständen redundant auf den Knoten des Netzwerkverbundes gespeichert sind und die Knoten den alltäglichen Bewegungen der Benutzer folgen, ist die Wahrung der Konsistenz die zentrale Herausforderung an die neuartigen mobil-verteilten Kollaborationssysteme. Weiterhin sind Kollaborationssitzungen ohne zentrale Dienste, wie z. B. Verzeichnisserver über die verfügbaren Sitzungen und Gruppen, zu etablieren. Dies stellt

Herausforderungen sowohl an die technische Vernetzung als auch an die soziale Verknüpfung der verfügbaren Kollaborationspartner. Stets gilt es dabei, die Komplexität der Konfiguration der Kooperationsumgebung vor dem Benutzer zu „verstecken“ und diese soweit wie möglich automatisch im Hintergrund erfolgen zu lassen. Der Kontext der Benutzer spielt in dieser Hinsicht eine wichtige Rolle und wird in mobilen Nutzungsszenarien zu einer unverzichtbaren Komponente.

Im folgenden Kapitel werden Systeme und Lösungen betrachtet, die wesentliche Komponenten für die Bewältigung dieser Herausforderungen darstellen. Dabei handelt es sich sowohl um Basistechnologien als auch um kooperationsstützende Systeme mit Blick auf mobile Teilnehmer. Am Ende des Kapitels wird basierend auf diesen Lösungen ein Entwurfsmuster für mobil-verteilte Kooperationsumgebungen skizziert.

4 Lösungsansätze für eine technische Umsetzung

Die Schaffung mobiler Kooperationsumgebungen beinhaltet Aspekte verschiedener Forschungsfelder der Informatik (vgl. Abbildung 4.1). Wie in [Eßmann und Hampel, 2003b] vorgestellt, sind diese im Wesentlichen in vier Bereiche zu unterteilen: Hardwarelösungen, Netzwerkprotokolle, Architekturkonzepte und Benutzerschnittstellen.

Als Grundlage für eine Kooperation in mobilen Nutzungsszenarien bedarf es zunächst portabler Geräte, die Benutzer in ihren Alltagssituationen begleiten, ohne sie in ihrer Mobilität zu behindern oder zu stören. Die immer steigende Verbreitung von so genannten Smartphones – einer Mischung aus Mobiltelefon und PDA – zeigt, dass inzwischen Hardwarelösungen existieren, die diese Bedingungen einer hohen Portabilität bei ausreichender Rechenleistung erfüllen.

Um dem Benutzer ein intuitiven Umgang mit der Kooperationsumgebung zu bieten, müssen Benutzungsschnittstellen geschaffen werden, die auf derartig kleine tragbare Geräte zugeschnitten sind oder gut für diese skalieren. Wichtiger Aspekt ist somit eine gute Anpassung der Darstellung sowie des Funktionsumfangs an die Fähigkeiten des benutzten Gerätes. Um die Eignung derartig kleiner Displays für die Darstellung virtueller Wissensräume zu ermitteln, wurden eigens Prototypen entwickelt, die die Möglichkeiten eines dokumentenzentrierten Arbeiten auf stiftbasierten Geräten mit kleinem Display ausloten [Eßmann und Hampel, 2003a, b]. Die positiven Ergebnisse aus der Nutzung der Prototypen lassen den Schluss zu, dass derartige Kleingeräte durchaus für eine Kooperation innerhalb virtueller Wissensräume geeignet sind. Es zeigte sich aber, dass Problemstellungen der mobilen Kooperation aus dem Spannungsfeld der Technologien zur Bereitstellung einer nahtlosen Verfügbarkeit des Kooperationsdienstes verwurzelt sind.

Zunächst einmal müssen mobile Geräte Netzwerkschnittstellen bieten, die eine Vernetzung mit den Geräten der Kooperationspartner erlauben. Im Fokus der Betrachtung entsprechender Technologien steht die Unabhängigkeit von zentral bereitgestellten Netzwerkinfrastrukturen, da bereits etablierte Infrastrukturen in vielen Nutzungssituationen nicht verfügbar und zudem häufig zu kostenintensiv sind.

Diese Faktoren stellen zweifelsohne Hemmnisse für eine durchgängige Nutzung mobiler Kooperationsumgebungen dar. Soll eine Kooperationsumgebung vom Benutzer angenommen werden, muss deren Benutzung kostenneutral zu alternativen Kooperationswerkzeugen sein, da die Kooperationspartner sonst auf diese Alternativen ausweichen. Dieses Ausweichen kann zu Medienbrüchen führen, die eine Rückkehr in die Kooperationsumgebung bei erneuter Verfügbarkeit erschweren. Um derartige Nutzungshemmisse zu vermeiden, ist es wichtig, die genutzten Netzwerkprotokolle



Abbildung 4.1: In die Schaffung mobil-verteilter Wissensräume involvierte Forschungsbereiche

so zu wählen, dass zum einen eine Unabhängigkeit von existierenden Netzwerkinfrastrukturen besteht und zum anderen deren Vorteile einer weiträumigen und zuverlässiger Vernetzung durch Übergänge genutzt werden kann. Die Netzwerkprotokolle bilden somit eine Schlüsseltechnologie für die mobile Kooperationsumgebung und werden in Abschnitt 4.2.1 genauer untersucht.

Wie bereits angedeutet, setzt die Kooperationsumgebung auf diese Netzwerkstrukturen auf. Sie muss dabei den Besonderheiten der Netzwerkinfrastruktur angepasst sein. Dies betrifft sowohl die Toleranz gegen Verbindungsabbrüche in einem sehr dynamischen Netzwerk als auch die Nutzung bereits existierender Strukturen in sporadisch verfügbaren klassischen Netzwerken. Es bedarf eines Entwurfsmusters für eine Architektur, die derartiger Rahmenbedingungen gewachsen ist und die nötige Flexibilität für das heterogene Nutzungsumfeld aufweist. Die Entwicklung eines solchen Entwurfsmusters bildet den Kern dieses Kapitels. Ausgehend von mobilitätsunterstützenden Protokollen werden die technischen Lösungsansätze anhand der in Kapitel 3 ermittelten Anforderungen bewertet.

Die drei Kernfelder *Vernetzung*, *Kontext* und *Konsistenz* werden in diesem Kapitel gemäß den Anforderungen der Handlungsebene in technische Rahmenbedingungen der Umsetzungsebene betrachtet (vgl. Abbildung 4.2). In einem ersten Schritt wird im Folgenden der Aufbau des einzelnen Knoten im mobilitätsfreundlichen Kooperationsnetzwerk betrachtet.

4.1 Knotenaufbau – Microkernel

Der dynamische technische Kontext, in dem sich mobil-verteilte Kooperationsanwendungen einbetten verlangt, ein hohes Maß an Anpassungsvermögen. Der Zwang zur Anpassung und Weiterentwicklung entsteht sowohl aus kurzfristigen Änderungen der Netzwerk- und Dienstekonstellationen infolge der hohen Mobilität der Koope-

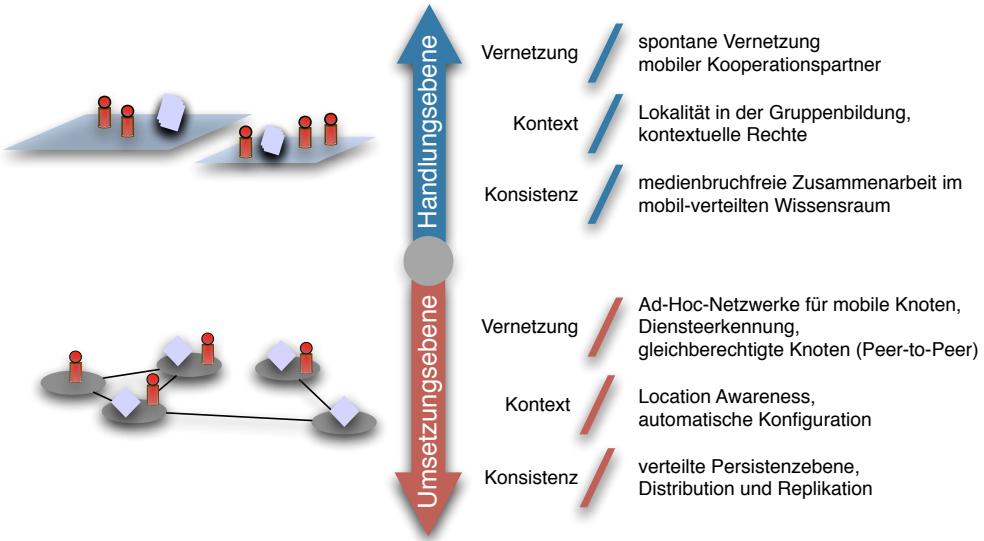


Abbildung 4.2: Der mobil-verteilte Wissensraum aus Perspektive von Handlungsebene und Umsetzungsebene

rationsknoten und der Nutzungsvorlieben der Benutzer als auch aus langfristigen Trends im Bereich der gewählten technischen Lösungsansätze.

Die Architektur der Anwendung muss dieser Dynamik mit einem hohen Maß an Flexibilität gewachsen sein. Im Fall der Mobilität der Netzwerknoten bedeutet dies, auf Änderungen im Netzwerkumfeld flexibel zu reagieren, verfügbare Dienste zur Laufzeit einzubinden und fehlende Dienste bei Bedarf zu ersetzen. Bei den wechselnden Nutzungsvorlieben der Benutzer muss sich die Anwendung durch einen hohen Grad an Konfigurierbarkeit an diese anpassen lassen. Des Weiteren müssen technische Neuerungen leicht in die Anwendung eingepflegt und veraltete funktionale Bestandteile ersetzt werden können.

Die im Bereich der klassischen zentralisierten CSCW-Systeme eingesetzten monolithischen Systemarchitekturen erschweren eine Anpassung an neue Nutzungskonstellationen durch starke Abhängigkeiten zwischen den funktionalen Bestandteilen. Oftmals ziehen sich die von den Systemen bereitgestellten Funktionen quer durch die gesamte Architektur des monolithischen Systems. Änderungen an Bestandteilen des Systems ziehen so aufgrund der Abhängigkeiten oft weitere Änderungen quer durch die Architektur des Systems nach sich. Auch Anpassungen der Konfiguration können häufig erst nach einem Neustart des Systems wirksam werden, da die funktionalen Abhängigkeiten eine Veränderung der Funktionsweise einzelner Bestandteile nicht zulässt, ohne die innere Konsistenz des Systems zu gefährden.

Um derartig miteinander verwobene Architekturen zu verhindern, zerlegt das Entwurfsmuster des *Microkernel* [Buschmann et al., 1996] ein Gesamtsystem in vonein-

ander getrennte und in sich gekapselte Funktionseinheiten, die über wohldefinierte Schnittstellen miteinander kommunizieren. Zentrales und einzig unverzichtbares Element einer solchen Architektur ist der Microkernel selbst. Dabei verfügt dieser über eine minimale Funktionalität und zeichnet sich lediglich für das Laden und Entfernen der Module verantwortlich. Abhängigkeiten zwischen Modulen erkennt der Microkernel in der Regel automatisch und lädt die entsprechenden Module bei Bedarf nach. Sämtliches Verhalten des Microkernel-Systems entspringt den Funktionen der verknüpften Module.

Das Entwurfsmuster des Microkernel wurde ursprünglich im Bereich der Betriebssysteme eingesetzt [Rashid et al., 1989], findet sich inzwischen aber auch in Rahmenarchitekturen für die Softwareentwicklung wieder. Prominente Vertreter sind hier das Plugin-Konzept des *Eclipse-Framework* [Gamma und Beck, 2003], *JBOSS* mit dem *JBossMicrokernel*¹ und *Hivemind*² aus dem *Jarkarta Projekt* der *Apache Foundation*.

Microkernel werden im besonderen Maße in komplexen und verteilten Systemen wie *Cooperative Virtual Environments (CVE)* eingesetzt. Beispiele sind hier *JADE* [Oliveira et al., 1999], *GNU/Maverik* [Hubbold et al., 1999] und *Bamboo* [Watsen und Zyda, 1998]. Aber auch im Bereich der Multi-Agenten-Systeme findet sich mit *MadKit* [Gutknecht et al., 2001] eine microkernelbasierte Architektur.

Besonders interessant im Hinblick auf die Ziele dieser Arbeit ist der Einsatz von microkernelbasierten Architekturen im Bereich der Peer-to-Peer-Netzwerke, *Ocean-Store* [Kubiatowicz et al., 2000] und verteilten CSCW-Cluster [Bopp und Hampel, 2005].

Monolithische und zentralisierte Ansätze klassischer CSCW-Umgebungen können aufgrund ihres starren Designs eine Anpassung an stark skalierende mobile Kooperationsumfelder kaum leisten. Das modulare und flexible Konzept des Microkernel verspricht hingegen ein passendes Entwurfsmuster für mobil-verteilte Kooperationsumgebungen, deren Dynamik und Flexibilität sich insbesondere an der spontanen Vernetzung der Benutzer in mobil-verteilten Kooperationsszenarien messen lassen muss. Einen ersten Architekturansatz in dieser Hinsicht wird in [Eßmann et al., 2004b] vorgestellt.

4.2 Vernetzung und Sichtbarkeit

Die Anforderungen an die Netzwerktechnologien, die mobil-verteilten Kooperationsumgebungen zu Grunde liegen, werden durch die neue Mobilität der Kooperationspartner und der daraus resultierenden Dezentralisierung bestimmt. Die Integration von funkbasierten Netzwerkschnittstellen wie *Wireless Local Area Network (WLAN)*³, *Worldwide Interoperability for Microwave Access (WiMAX)*⁴, *bluetooth*⁵,

¹<http://www.jboss.org/wiki/Wiki.jsp?page=JBossMicrokernel>

²<http://jakarta.apache.org/hivemind/>

³IEEE 802.11

⁴IEEE 802.16

etc. erlaubt in vielen Mobilcomputern inzwischen die kabellose Vernetzung zweier Gegenstellen in der physikalischen Schicht. Aber erst die darauf aufsetzenden Netzwerkprotokolle ermöglichen die eigentliche Kommunikation und den Aufbau komplexerer Netzwerktopologien. Diese sind somit als Kommunikationsgrundlage eine der Schlüsseltechnologien für die mobil-verteilte Kooperationsumgebung und werden im Folgenden genauer betrachtet.

Auf der technischen Ebene müssen die mobilen Geräte miteinander verbunden und die verteilte angebotenen Dienste verknüpft werden, um Nachrichten austauschen und den Kooperationsprozess koordinieren zu können. Doch die in diesem Abschnitt betrachteten Technologien gehen über eine rein technische Vernetzung von mobilen Geräten hinaus. Erst eine soziale Verknüpfung der Teilnehmer über Mechanismen der gegenseitigen Wahrnehmung innerhalb der Kooperationsumgebung erlauben eine computergestützte Zusammenarbeit. Daher werden in diesem Abschnitt auch die Aspekte der Bereitstellung einer Wahrnehmung der potentiellen Kooperationspartner betrachtet.

Protokolle zur technischen Vernetzung der Kooperationsumgebung müssen an den Anforderungen aus Kapitel 3 gemessen werden. Aus diesen ergeben sich eine Anzahl von Punkten als wesentliche Merkmale einer mobilitätsfreundlichen Vernetzung von Kooperationsumgebungen:

Spontan Die zentrale Anforderung, die aus der Mobilität der Benutzer des Netzwerkes resultiert, ist die Fähigkeit, Netzwerke ohne bestehende Infrastruktur (spontan, ad hoc) zu bilden.

Robust Die Funktion des Netzwerks muss auch bei Ausfall einzelner Knoten und bei geringer Bandbreite gewährleistet sein.

Skalierbar Die Netzwerkinfrastruktur darf nicht durch das Hinzukommen von zusätzlichen Knoten in seiner Leistungsfähigkeit beeinträchtigt werden.

Netzübergreifend Dienste aus etablierten Netzwerkstrukturen muss das spontan etablierte Netzwerk bei Verfügbarkeit von Zugangspunkten zu bestehenden Netzinfrastrukturen (z. B. Internet) einbinden können. Bei einem direkten Wechsel der Zugangspunkte sollte es nicht zu einer Unterbrechung der genutzten Dienste kommen.

Kompatibel Für die Einbindung existierender Netzwerkanwendungen muss die mobile Netzwerkinfrastruktur kompatibel zu den Standardprotokollen TCP/IP und UDP/IP sein.

Automatische Konfiguration Die Konfiguration der Netzwerkinfrastruktur muss transparent, automatisch und ohne Benutzereingriff geschehen, um möglichst wenig Nutzungshemmnisse zu generieren.

⁵IEEE 802.15.1

An diesen Merkmalen müssen sich die Protokolle zur Vernetzung mobil-verteilter Kooperationsumgebungen messen lassen. Das Forschungsfeld Netzwerkprotokolle für mobile Knoten hat bereits eine Vielzahl möglicher Lösungen hervorgebracht, die zum Teil völlig verschiedene Lösungsansätze verfolgen und von ebenso verschiedenen Verständnissen von Mobilität zeugen. Der kommende Abschnitt beleuchtet diese unterschiedlichen Ansätze anhand ihrer Eignung für mobile Kooperationsszenarien.

4.2.1 Protokolle zur Vernetzung mobiler Knoten

Im Bereich der Netzwerkprotokolle gibt es zwei von Grund auf unterschiedliche Betrachtungsweisen von Mobilität. Die Erste betrachtet die so genannte *Macro Mobility* oder *Inter Domain Mobility*. In dieser Form der Mobilität wird der mobile Knoten als ein Teil eines relativ stabilen Netzwerkes betrachtet, der über Zugangspunkte mit diesem verbunden ist und in seiner Mobilität zwischen den Zugangspunkten wechselt. In diesem Bereich hat sich das *Mobile IP-Protokoll* zu einer Art Standard etabliert.

Die andere Betrachtungsweise sieht alle Knoten des Netzwerkes als mobil und lose über wechselnde Netzwerkverbindungen verbunden. In diesem Zusammenhang wird von *Micro Mobility* oder *Intra Domain Mobility* gesprochen. Protokolle dieser Mobilitätsklasse müssen zum einen eine dynamische Vernetzung der Knoten bieten und zum anderen versuchen, stabile Netzwerkpfade in dem dynamischen Netzwerk zu etablieren.

Mobile IP – Netzwerkprotokolle für mobile Endgeräte

Mobilität innerhalb etablierter Netzwerke wie dem Internet meint zumeist das Wechseln zwischen verschiedenen Netzwerkdomains. In diesem Fall wird von *Macro Mobility* oder *Inter Domain Mobility* gesprochen. Diese haben zum Ziel, mobilen Geräten den nahtlosen Wechsel zwischen Domains ohne Verbindungsabbrüche zu ermöglichen. Bei Lösungen, die einen Neustart der Netzwerkverbindungen benötigen und damit einen Abbruch bestehender Anwendungen in Kauf nehmen, unterstützen in diesem Zusammenhang lediglich so genannte *Portability* aber keine Mobilität (engl. *Mobility*) [vgl. Valko, 1999]. Eine vielversprechende Lösung, die eine dementsprechende Unterstützung der Mobilität bietet, ist die von der *Internet Engineering Task Force (IETF)* entwickelte IP Protokollerweiterung *Mobile IP*.

Bei *Mobile IP* [Perkins, 1997] handelt es sich um eine Erweiterung des im Internet gebräuchlichen IP-Protokolls, die eine Einbindung mobiler Endgeräte in etablierte Netzwerkinfrastrukturen bietet.

Damit ein Endgerät reibungslos zwischen verschiedenen Netzwerken wechseln kann, muss stets für die Erreichbarkeit des mobilen Knotens gesorgt werden und einen unterbrechungsfreien Wechsel zwischen Zugangspunkten verschiedener Subnetze erlauben, ohne bestehende Netzwerkverbindungen unterbrechen zu müssen. In IPv4-basierten Netzwerken wird jedem Knoten für dessen Adressierung eine eindeutige IP-Adresse aus dem assoziierten Subnetz zugewiesen. Wechselt der Knoten zwischen

zwei Subnetzen, erhält dieser eine IP-Adresse aus dem Subnetz dem er neu beitritt. Auf diese Art wird ein korrektes *Routing* der IP-Pakete gewährleistet.

Aufgrund der neuen IP-Adresse erreichen Pakete die an die ursprüngliche Adresse gesendet wurden den Knoten nicht mehr. Daher müssen bestehende Verbindungen zunächst beendet und mit der neuen Adresse wieder aufgenommen werden. Dazu muss der mobile Knoten aber erst den Gegenstellen seine neue Adresse mitteilen. Somit ist ein temporärer Verbindungsabbruch unvermeidbar. Neue Verbindungen zu dem mobilen Knoten hingegen sind aufgrund fehlender Addressinformationen kaum möglich.

Ein nahtloser Übergang zwischen zwei Subnetzen ist in Standard IPv4 Netzwerken somit kaum möglich. Um dieses Problem bei Beachtung der Kompatibilität zum IPv4-Protokoll umgehen zu können, entwarf die IETF die IPv4-kompatible Mobile IP-Variante *Mobile IPv4 (MIPv4)* [Perkins, 2002]. Der Grundgedanke ist jedem Endgerät im Netzwerk eine feste eindeutige Adresse zuzuweisen und so Verbindungsabbrüche aufgrund von Addresswechseln zu vermeiden.

Jedes mobile Endgerät wird in MIPv4 einem Heimatnetz zugeordnet, in dem ein so genannter *Home Agent (HA)* die Erreichbarkeit des mobilen Knotens gewährleistet. Als Gegenstücke befinden sich in fremden Subnetzen *Foreign Agents (FA)*, an die sich der mobile Knoten beim Betreten der Subnetze anmeldet. Der FA meldet daraufhin dem HA die *Care-of-Adresse* des mobilen Knotens. Will nun ein Teilnehmer im Netz den mobilen Knoten kontaktieren, wendet er sich nicht direkt an diesen, sondern sendet seine Anfrage an die feste Adresse des mobilen Knotens im Heimatnetz. Der HA nimmt diese Anfrage stellvertretend entgegen und leitet sie an den FA des Netzes, in dem sich der mobile Knoten befindet, weiter. Der FA übergibt die Anfrage seinerseits an den mobilen Knoten. Die Anfragen werden somit über HA und FA umgeleitet.

Die Kommunikation des mobilen Knotens zu seinen Kommunikationspartner kann hingegen direkt erfolgen (*Dreiecksrouting*). Bei einem Wechsel des Subnetzes ändert sich jedesmal die care-of Adresse. Da die Kommunikationspartner aber über die feste Adresse des mobilen Knoten in dessen Heimatnetz mit diesem kommunizieren, wird der Datenverkehr lediglich an die neue care-of Adresse umgeleitet. Die Verbindungen müssen nicht zurückgesetzt werden.

Die Hauptnachteile der Mobile IP Implementierung für IPv4 entstehen aus dem Dreiecksrouting und der daraus entstehenden Abhängigkeit von Home Agent und Foreign Agent: Datenpakete sind aufgrund des Dreiecksroutings über HA und FA länger unterwegs (*Verzögerung, engl.: Delay*), das gesamte Routing hängt von der Funktionsfähigkeit von HA und FA ab (*Robustheit, engl.: Robustness*) und der HA muss die Last der Kommunikation aller im zugeordneten mobilen Knoten tragen (*Skalierbarkeit, engl.: Scalability*).

Als Alternative bietet sich hier *Mobile IPv6 (MIPv6)* an, das anders als MIPv4 auf IPv6 basiert und von dessen Vorteilen profitiert [Johnson et al., 2004]. In IPv6 ist es möglich, die Routen zwischen zwei Knoten automatisch zu optimieren, was den FA überflüssig macht. Der mobile Knoten meldet seine care-of Adresse nur noch an den HA, der Ansprechpartner für potentielle Kommunikationspartner in der In-

italisierungsphase der Kommunikation ist. Sobald die Verbindung zwischen mobilen Knoten und Kommunikationspartner hergestellt ist, können die Datenpakete dank der automatischen Routenoptimierung direkt gesandt werden, ohne weiterhin vom HA abhängig zu sein.

Mit Hilfe des Mobile IP Protokolls ist es somit möglich, mit mobilen Geräten von einer Domain zu einer anderen zu wechseln, ohne die benötigte Netzwerkverbindung zu verlieren. Für die Anwendung dieser Kommunikationslösung wird allerdings stets eine ständige Verbindung zum Internet benötigt und jede Organisation benötigt einen Agenten, der die Bekanntmachung ihrer mobilen Knoten übernimmt. Diese Lösung hängt somit nach wie vor stark von einer bestehenden Netzwerkinfrastruktur ab. Da diese Abhängigkeit in mobilen Szenarien problematisch sein kann, versprechen die so genannten *Ad-Hoc-Netzwerk-Protokolle* die unabhängige Etablierung von Netzwerken von Grund auf.

Ad-Hoc-Netzwerke

Protokolle, die in mobilen Nutzungsszenarien spontan eine Netzwerkinfrastruktur etablieren, ordnet man der Protokollfamilie der *Mobile Ad-Hoc-Networks (Manets)* zu. Die einzelnen Netzwerknoten werden als *Hops* bezeichnet. Ist ein Hop durch einen anderen Hop nur über eine Route über weitere Hops erreichbar, spricht man von *Multihop-Verbindungen*. Hops in mobilen Netzwerkstrukturen sind in der Regel beweglich und können nicht über längere Zeiträume einer festen Position zugeordnet werden. Daher bedürfen Manets spezieller dynamischer Routingverfahren und Handoff-Mechanismen. Eine Auswahl entsprechender Verfahren findet sich in [Perkins, 2001].

Bei der Mobilität der Hops innerhalb eines Manets spricht man von *Micro Mobility* oder *Intra Domain Mobility*. Diese zeichnet sich durch eine sehr viel höhere Dynamik als die zuvor behandelte *Inter Domain* oder *Macro Mobility* aus. Des Weiteren kann nicht von der Erreichbarkeit fester Dienste ausgegangen werden, wie es z.B. bei den Home Agents des Mobile IP-Protokolls der Fall ist.

Die einfachste Version von Routingverfahren für mobile Netzwerke unterstützt nur direkte Verbindungen zwischen Hops. Ein Vertreter dieser Protokolle ist bereits in vielen Betriebssystemen integriert und wird *Link-Local* genannt. Im Prinzip ist Link-Local ein Protokoll zur automatischen Vergabe von IP-Adressen, das Routing bleibt identisch zu dem in klassischen IP-Netzen. Für die Aushandlung der IP-Adresse wählt jeder Hop zufällig eine Adresse aus einem für die Link-Local-Vernetzung freigehaltenen IP-Addressbereich $129.169.0.0/255.255.0.0$ und sendet diese per Broadcast ins Netzwerk. Wird die Adresse nicht bereits durch einen anderen Hop zur Kommunikation benutzt, darf er diese nutzen, um über sie zu kommunizieren. Ist aber die gewählte Adresse bereits durch einen anderen Hop für die Kommunikation gebunden, legt deren derzeitiger Besitzer, ebenfalls per Broadcast, Einspruch ein, und die Prozedur beginnt von vorne. Eine bereits vergebene aber nicht an eine aktive Kommunikation gebundene Adresse muss hingegen wieder freigegeben werden.

Die Kommunikation bei diesen Verfahren ist stets auf die direkte Kommunikation zwischen den Kommunikationspartnern beschränkt und durch die vielen Broadcasts mit einem hohen Protokoll-Overhead versehen. Des Weiteren ist die Anzahl der möglichen Teilnehmer auf 65534 beschränkt⁶ – wenngleich bei einer Link-Local-Vernetzung eine derartige Netzwerkgröße unwahrscheinlich erscheint. Für Face-to-Face-Kooperationen ist diese Vernetzung trotz ihrer Einschränkungen geeignet. Eine Multihop-Vernetzung ist in diesen Situationen aufgrund des direkten Kontakts der Kommunikationspartner nicht notwendig. Die geforderte Kompatibilität zu klassischen IP-Netzwerken ist hingegen ohne weiteres gegeben.

Link-Local unterstützt keine weiträumige Vernetzung von örtlich verteilten Kommunikationspartnern, da es keine indirekten Multihop-Verbindungen kennt. Im Gegensatz zu den meisten aufwändigeren Multi-Hop-Routing-Protokollen ist es aber bereits für die mobilen Nutzer verfügbar. Die Mehrzahl der aufwändigeren Manet-Protokolle sind bisher nur als Spezifikation und Algorithmen für Netzwerksimulatoren wie z. B. dem *Network Simulator 2 (ns2)*⁷ verfügbar. Die Integration der unterschiedlichen Lösungen in die Protokollsichten der Betriebssysteme ist ein langwieriger Prozess und die Protokolle an sich sind oftmals inkompatibel zueinander.

Im Wesentlichen unterscheiden sich die Protokolle zur Realisierung eines Manet in der Art und Weise der Verwaltung der Routingpfade für das Netzwerk. Die eine Gruppe der Protokolle berechnet die Routingpfade zu den Kommunikationspartnern stets im voraus (*proaktiv*), so dass bei Kontaktanfragen zu einem entfernten Knoten der Versendepfad für die Nachricht bereits bekannt ist. Die andere Gruppe berechnet den Pfad zu entfernten Knoten erst, wenn die Kommunikation initiiert wird (*reakтив*). Durch die reaktive Ermittlung des Routingpfades vom Sender zum Empfänger wird der Verbindungsaufbau im Vergleich zum proaktiven Verfahren verzögert, aber die ständige Kommunikation zur Aktualisierung der Routinginformationen vermieden.

Zu der proaktiven Protokollfamilie gehören beispielsweise das *Destination-Sequence Distance-Vector (DSDV)* Protokoll [Perkins und Bhagwat, 1994, 2000] und das *Optimized Link State Routing (OLSR)* Protokoll [OLSR, 2003], das unter anderem im *Freifunk Projekt*⁸ für die Etablierung eines stadtübergreifenden WLAN in Berlin Verwendung findet.

Zu den reaktiven Protokollen gehören das *Dynamic Source Routing (DSR)* Protokoll [Johnson, 1994], das *Temporally-Ordered Routing Algorithm (TORA)* Protokoll [Park und Corson, 1997, 2001] und das *Ad hoc On Demand Distance Vector (AODV)* Protokoll [Perkins und Royer, 1999; Perkins et al., 2003].

Aufgrund der unterschiedlichen Charakteristika der Routingmechanismen scheinen die proaktiven Netzwerke mit ihren vorberechneten Routingtabellen intuitiv besser für autonome aber relativ statische Netzwerke geeignet, wohingegen die re-

⁶Die ersten und die letzten 256 Adressen sind von der *Internet Assigned Numbers Authority (IANA)* für zukünftige Anwendungen reserviert.

⁷<http://www.isi.edu/nsnam/ns>

⁸<http://freifunk.net/>

aktiven Protokolle eher in dynamischen Konstellationen mit mobilen Knoten interessant erscheinen.

Vergleiche von Netzwerkprotokollen finden aufgrund fehlender Implementierungen für Betriebssysteme und dem Wunsch nach kontrollierten Rahmenbedingungen zumeist nur im Simulator und unter theoretischen Annahmen statt [Broch et al., 1998; Dyer und Boppana, 2001; Campell et al., 2002]. Hier werden die Protokolle häufig unter extremen Bedingungen in Worst-Case-Szenarien betrachtet oder nur spezielle Aspekte der Protokolle aufgegriffen, die eine generelle Aussage über deren Alltagstauglichkeit erschweren.

Um eine stärkere Betrachtung der Alltagstauglichkeit der Manet Protokolle zu erlauben, wurde an der *Uppsala University* das standardisierte *Ad hoc Protocol Evaluation (APE)* Testumfeld entworfen [Lundgren et al., 2002], das die verfügbaren Protokollimplementierungen mittels realer Hard- und Softwareumgebungen erproben hilft. Bei Betrachtung der gängigen Manet-Protokolle zeigte sich jedoch nur eine bedingte Eignung für hochmobile und skalierende Netzwerkumgebungen [Tschudin et al., 2005].

Nichtsdestotrotz scheint z. B. das AODV-Protokoll für Szenarien mobiler Vernetzung grundsätzlich gut geeignet und es existieren einige nutzbare Implementierungen. Mit *KERNEL-AODV NIST*⁹ und *AODV-UU*¹⁰ sind zwei freie Implementierungen für das *Linux* Betriebssystem verfügbar. Und darüber hinaus existiert mit *AODV for Microsoft Windows*¹¹ auch eine Protokollerweiterung für *WindowsXP*.

Nach der Etablierung eines funktionsfähigen Manets fehlt zum Betrieb von Internet-Anwendungen noch die IP-Schicht, die das Adressieren der Rechner erlaubt. Hier existiert, wie bei Mobile IP, die grundsätzliche Möglichkeit, entweder IPv4 oder IPv6 zu benutzen. IPv6 hat den klaren Vorteil, aufgrund des 128-bit Adressraumes und der Einbeziehung der HardwareAdresse des Netzwerkdevices, mit hoher Wahrscheinlichkeit eine eindeutige Adresse zuzuweisen, ohne Adresskonflikte im Netzwerk zu erzeugen. Leider sind viele Netzwerkanwendungen noch nicht in IPv6 Netzwerken lauffähig.

IPv4 hat mit 16 Bit einen sehr viel kleineren Adressraum, der Adresskonflikte sehr viel wahrscheinlicher macht. Außerdem muss jedes Gerät bei einer automatischen Konfiguration eine eigene Adresse raten und dann über ein Protokoll auf Konflikte überprüfen. Das Link-Local Protokoll lässt sich in Multihop-Netzwerken nicht verwenden, da es Konflikte nur mit direkten Nachbarn überprüfen kann. Ähnlich dem Link-Local-Protokoll verfährt das *Duplicate Adress Detection* Protokoll [Perkins et al., 2001; Jeong et al., 2005]. Anstatt von ARP-Requests verwendet es allerdings modifizierte ICMP-Pakete und ist daher in der Lage, eine grössere Netzwerkumgebung auf Konflikte zu überprüfen. Das *Prophet Adress Allocation-Protokoll* [Zhou et al., 2003] verwendet ein stochastisches Verfahren, um Adresskonflikte vorhersagen und somit vermeiden zu können.

⁹http://w3.antd.nist.gov/wctg/aodv_kernel/

¹⁰<http://www.docs.uu.se/~henrik1/aodv/>

¹¹<http://moment.cs.ucsb.edu/AODV/aodv-windows.html>

Mit den hier vorgestellten Lösungen lässt sich eine spontane Vernetzung realisieren, die zur Etablierung der mobil-verteilten Kooperationsumgebung dienen kann. Bei einer Einbindung entfernter Teilnehmer wäre jedoch eine Einbindung von eventuell vorhandenen Zugängen zu weiträumigen Netzwerkstrukturen wie dem Internet wünschenswert.

Mischarchitekturen

Mischarchitekturen kombinieren inter- und intra-domain Vernetzung. Ursprüngliches Ziel dieses Ansatzes ist die Verminderung der bei Mobile IP entstehenden Netzlast durch die Pakete zur Aktualisierung der Care-of-Adresse des mobilen Clients zum Home Agent. Für die *Inter Domain Mobility* oder auch *Macro Mobility* setzt man dabei auf IETF's Mobile IP Protokoll und erweitert es für *Intra Domain Mobility (Micro Mobility)* um eigene Handoff- und Routingstrategien. So werden bei Positionswechseln innerhalb einer Domain keine Pakete an den HA versandt. Vergleiche einiger Verfahren finden sich in [Reinbold und Bonaventure, 2001] (Hierarchical Mobile IP, Proactive Handoff, Fast Handoff, TeleMIP, Cellular IP, HAWAII und EMA) und [Campell et al., 2002] (Ethernet Switch, Cellular IP, Hawaii, Hierarchical Mobile IP). Im Folgenden seien die wichtigsten Verfahren kurz vorgestellt.

Das *Hierarchical Mobile IP* ist eine Erweiterung des Mobile IP Protokolls um hierarchische Registrierung an Agents [Gustafsson et al., 2000]. So können unterhalb der Agents eigene Netzwerkstrukturen entstehen, die vom Home Agent selbst verwaltet werden. Auch die *Handoff-Aware Wireless Access Internet Infrastructure (HAWAII)* [Ramjee et al., 1999] nutzt Mobile IP für *Inter Domain Mobility* und gewährleistet die *Intra Domain Mobility* über einen *Foreign Domain Root Router*, der die lokalen Netzwerkpfade verwaltet. Diese Pfade werden periodisch erneuert und anhand *Forwarding Schemes* ermittelt. *CellularIP* [Valko, 1999] ist ein ähnliches Verfahren, das stets optimale Pfade zum Gateway ermittelt.

Neben der Entlastung der Kommunikation zur Pflege des Mobile IP Netzwerkes kann die Kombination von Manets und Mobile IP helfen, die Reichweite der Ad-Hoc-Netzwerke zu erhöhen. Die erste wirkliche Mischform zwischen Mobile IP und selbstorganisierten Ad-Hoc-Netzwerken realisiert das Verfahren in [Tseng et al., 2003]. Dieser Ansatz nutzt für *Macro Mobility* Mobile IPv4 und realisiert *Micro Mobility* über das Ad-Hoc-Netzwerk-Protokoll DSDV. Damit ist das lokale Netzwerk komplett selbstverwaltet und dennoch über den Foreign Agent an das Internet angebunden. Die Knoten sind somit stets über ihren Home Agent erreichbar.

Derartige Mischarchitekturen für die spontane Vernetzung von mobilen Knoten mit einer optionalen Erreichbarkeit aus dem Internet erlauben es, autonome Netzwerke zwischen den mobilen Knoten zu etablieren und bei Verfügbarkeit zusätzlich Dienste aus dem Internet sowie entfernte Kooperationspartner einzubinden. Die Dynamik und Skalierung derartiger Lösungen macht die zentralisierten Architekturen klassischer CSCW-Systeme ungeeignet für einen Einsatz in vielen der möglichen Vernetzungskonstellationen.

4.2.2 Vernetzung verteilter Dienste

In mobilen Netzwerkumgebungen sind die Benutzer mit einer ständig wechselnden Dienstinfrastruktur konfrontiert. Einige der verfügbaren Dienste laufen dabei eventuell auf entlegenen Knoten im Internet und andere direkt in der unmittelbaren Umgebung des Benutzers. Ohne eine Wahrnehmung der verfügbaren Dienste können aber weder die einen noch die anderen genutzt werden. Es bedarf daher passender *Service Discovery-Verfahren*, um die verfügbaren Dienste allen Benutzern zugänglich zu machen.

Ähnlich wie mit der Sichtbarkeit der Dienste in mobil-spontanen Netzwerken verhält es sich mit der Wahrnehmung potentieller Kooperationspartner, die in der Netzwerkumgebung verfügbar sind. Aus einer technischen Sichtweise können sie wie ein Dienst betrachtet werden, dessen Verfügbarkeit durch die lokale Instanz der mobil-verteilten Kooperationsumgebung auf den persönlichen Geräten gewährleistet und bekanntgegeben wird.

Als Besonderheit gegenüber virtualisierten Diensten im Netz besteht bei den Kooperationspartnern eventuell ein Interesse an Informationen über deren aktuellen Aufenthaltsort, um einen räumlichen Bezug zu sich herzustellen (vgl. Abschnitt 3.3.2). Dieser Bezug kann entweder aus bereitgestellten Informationen von *Location Awareness-Diensten* [vgl. Hightower und Borriello, 2001] oder über die Informationen aus der Netzwerkumgebung gewonnen werden. Letzteres wäre z. B. über eine Auswertung der Güte der Funkverbindung oder der Anzahl der Wegpunkte auf dem Routingpfad möglich.

Da in mobil-spontanen Kooperationsszenarien zentrale Verzeichnisse über verfügbare Dienste – wie jeder andere Dienst auch – nicht direkt sichtbar sind und zudem als unzuverlässig gelten müssen, muss jeder Dienst sich selber im Netz bekannt machen.

In der *lokalen Netzwerkumgebung* kann die Service Discovery per *IETF Zeroconf Workinggroup* spezifizierte *Multicast DNS (mDNS)* [Guttman, 2001; Huck et al., 2002; Stirling und Al-Ali, 2003] bewerkstelligt werden. Ein mDNS konformer Dienst kündigt sich dazu mittels einer Multicast-Nachricht bei allen anderen Knoten innerhalb desselben Subnetzes an und teilt in dieser Nachricht mit, wie er zu erreichen ist. Mit diesen Informationen können sich dann die Clients mit dem Dienst verbinden.

Sollen Dienste jedoch in *weiträumig verteilten Systemen* bekannt gemacht werden, bedarf es einer Infrastruktur, die Dienstankündigungen an interessierte Clients weiterreichen. Zwar kennt mDNS so genannte *Proxies*, die die Dienstankündigungen in ein anderes Subnetz weiterleiten können, diese eignen sich aber lediglich dafür Subnetze zu bündeln, nicht aber um autonom weiträumige Diensteinfrastrukturen zu etablieren [vgl. Eßmann et al., 2004a].

Derartig weiträumige Infrastrukturen mit gleichberechtigten Dienstanbieter wie Dienstnehmern hat sich das *JXTA Framework*¹² zum Ziel gesetzt [Gong, 2001; Brookshier et al., 2002]. Zu diesem Zweck etabliert JXTA ein weiträumiges Peer-to-Peer-Netzwerk, bei dem Dienste zunächst lokal bekannt gegeben werden und dann

¹²<http://www.jxta.org>

nach und nach mittels so genannter *Rendezvous-Peers* in anderen Netzwerken propagiert werden. So können auch entfernte Peers die beworbenen Dienste eines Peers in Anspruch nehmen.

Beide Lösungen eignen sich für eine Service Discovery in mobil-verteilten Kooperationsszenarien. Während mDNS eher auf die lokale Netzwerkumgebung zielt, stellt JXTA ein komplexes Dienste-Netzwerk mit entsprechendem Kommunikations-Overhead dar. Beide Ansätze schliessen sich nicht gegenseitig aus und ihre Nutzung kann von der gewollten Reichweite des bekanntzugebenden Dienstes abhängig gemacht werden.

4.2.3 Netzwerktopologie

Mobile IP und Manets bilden zusammen mit den geeigneten Service Discovery-Verfahren mobilitätsfreundliche Netzwerkinfrastrukturen für mobil-verteilte Kooperationsumgebungen. Besonders Manets zeichnen sich durch eine hohe Dynamik der Verbindungen im Netzwerk aus; aber auch die Mobile IP Vernetzung enthält eine Abkehr von statischen und zentral verwalteten Infrastrukturen. Klassische Kooperationssysteme hingegen orientieren sich an Client-Server-Architekturen mit für die Funktion kritischen zentralisierten Diensten.

Beispiel einer solchen zentralisierten CSCW-Architektur ist das im *Heinz Nixdorf Institut (HNI)* an der *Universität Paderborn* entwickelte *open-sTeam*. Es implementiert das in Abschnitt 3.2 besprochene Konzept der *kooperativen virtuellen Wissensräume* [Hampel, 2001]. Diese bilden einen raumbasierten Ansatz zur kooperativen Wissensorganisation. Der Wissensraum ist in virtuelle Räume unterteilt, in denen sich die Kooperationspartner, repräsentiert durch virtuelle Avatare, treffen und ihr Wissen anhand von Dokumenten und Kommunikationsmedien kooperativ strukturieren. Für diesen dokumentenzentrierten Ansatz der Kooperation bilden die *Medienfunktionen* die benötigte Grundfunktionalität ab [Hampel und Keil-Slawik, 2002].

Die zentrale Komponente des Systems bildet ein dedizierter *open-sTeam*-Server, der den gesamten Wissensraum speichert. Er ist der zentrale Anlaufpunkt für die Kooperationspartner und alleinverantwortlich für die Verwaltung der Kooperationsumgebung und die Persistenz des Wissensraumes. Daher hängt die Funktionsfähigkeit der Kooperationsumgebung stark von der Erreichbarkeit des Servers ab.

Das Gros der gängigen Kooperationssysteme basiert ebenfalls auf zentral verwalteten Arbeitsbereichen mit einer ebenfalls zentral gespeicherten gemeinsamen Dokumentenbasis. Prominente Beispiele sind hier *BSCW* [Bentley et al., 1997] und *Lotus Notes* [Kawell et al., 1988]. Lotus Notes ist eine Groupware-Umgebung mit mächtigen Replikationsmechanismen für mobiles Arbeiten. Die Replikation ist aber von dem zentralen Notes-Server abhängig, so dass kooperative Interaktion nur im verbundenen Zustand möglich ist.

Um diese zentrale Speicherung auf einem isolierten Server aufzubrechen, lässt sich mittels Verknüpfung mehrerer isolierter Server ein *Server-Verbund (Cluster)* mit einem gemeinsamen *verteilten* Datenraum schaffen. Dieser Ansatz basiert aber

weiterhin auf der Annahme, dass Server im Netzwerk fest verfügbar sind und der Wissensraum auf mehrere dedizierte Server verteilt gespeichert wird.

Einen Schritt weiter geht der *Peer-to-Peer-Ansatz*, der auch Clients in die Verwaltung und Speicherung der verteilten Softwareumgebung einbezieht [vgl. Eßmann und Hampel, 2005; Eßmann und Funke, 2005]. Prinzipiell treten alle teilnehmenden Knoten (*Peers*) in einem solchen Verbund gleichberechtigt auf und können notfalls die Funktion von ausgefallenen Peers übernehmen.

Das *Groove Framework*¹³ erlaubt es, eine Peer-to-Peer-Kooperationsumgebung aufzubauen und mittels Kommunikations- und Dateiaustauschmechanismen zu vernetzen. Für einige Funktionalitäten, wie Initialisierung der Kooperationssitzung und Synchronisation der verteilten Arbeitsbereiche, benötigt Groove allerdings einen zentralen Server, der die Kontaktaufnahme initiiert. Groove kann durch externe Werkzeuge erweitert werden, die in die Groove Arbeitsumgebung eingebettet werden.

Einen anderen Ansatz bezüglich der Implementierung einer Peer-to-Peer-Umgebung verfolgt *Speakeasy* [Edwards et al., 2002b]. Speakeeasy ist zunächst ein Peer-to-Peer Framework, um kooperative mobile Anwendungen zu implementieren. Mit *Casca* [Edwards et al., 2002a] existiert aber bereits eine Beispielimplementierung für eine Peer-to-Peer Kooperationsumgebung, die jedoch sitzungsbasiert ist und keine persistenten Arbeitsbereiche zu Verfügung stellt.

Eine eingeschränkte Persistenz der Kooperationsmaterialien bietet *Swifff* [Eßmann et al., 2004c; Slawik et al., 2004]. Swifff ist eine verteilte Kooperationsumgebung, die auf das *JXTA* Peer-to-Peer Framework aufsetzt (vgl. Abschnitt 4.2.2). Es bildet einen lose vernetzten gemeinsamen Wissensraum, in dem Objekte verteilt bearbeitet werden können. Die Bearbeitung eines entfernten Objektes erzeugt aber stets eine lokale Kopie. Diese ist sodann als eigenständiges Objekt im Netzwerk verfügbar. Eine Synchronisation der Kopien der verteilten und gemeinsam genutzten Objekte findet nicht statt.

Es existieren also erste Ansätze für die mobile und verteilte Nutzung von gemeinsamen Arbeitsumgebungen. Insbesondere die Peer-to-Peer-Topologie verspricht hier den Anforderungen einer dynamischen mobilen Kooperationsumgebung gewachsen zu sein. Durch ihre verteilte Architektur wirft sie aber Fragen bezüglich der konsistenten und persistenten Speicherung der gemeinsam genutzten Objekte auf. Es gilt daher, in einem nächsten Schritt, verteilte Persistenzkonzepte auf ihre Eignung für eine Kooperationsumgebung mit gleichberechtigten mobilen Knoten zu betrachten. Dabei liegt der besonderere Fokus auf den Anforderungen, die aus der Dynamik der spontanen Vernetzung und der Wahrung der Konsistenz des gemeinsamen und verteilten gespeicherten Arbeitsbereiches erwachsen.

4.3 Persistenz

Die Netzwerkumgebungen in denen sich mobile Kollaborationsumgebungen bewegen, zeichnen sich durch eine hohe Heterogenität aus und stecken so die technischen

¹³<http://www.groove.net>

Rahmenbedingungen für die mobile Kooperation ab. Die hohe Dynamik lässt eine exklusive Speicherung von Objekten auf zentralen Servern nicht zu. In Netzwerkumgebungen mit Zugang zum Internet bietet ein Server als zentrale Anlaufstelle eine Reihe von Vorteilen und vereinfacht die Verwaltung der Kooperationsumgebung. Fehlt ein solcher Zugangspunkt jedoch, sind die Client-Anwendungen aufgrund der Abhängigkeit von zentralen Servern nicht länger nutzbar. Um eine Kooperationsumgebung in einem Ad-Hoc-Netzwerk ohne Verbindung zu einem dedizierten Server nutzen zu können, müssen deren Dienste verteilt auf den erreichbaren Knoten laufen.

Diese Forderung betrifft auch die Verfügbarkeit der zur Kollaboration benötigten Objekte. Auch diese müssen verteilt über die Geräte aller beteiligten Benutzer vorliegen. Bei der Bereitstellung eines verteilten Objektspeichers ist es wichtig, Strategien und Konzepte zu entwickeln, die festlegen, welche Objekte auf welchen Geräten gespeichert werden sollen. Beispielsweise kann die Strategie einer besitzorientierten Speicherung verfolgt werden, bei der jeder Benutzer seine eigenen Dateien lokal speichert und anderen Benutzern zur Verfügung stellt. Gerade in kooperativen Szenarien fällt eine solche Besitzzuordnung allerdings schwer.

Der Entwurf von Konzepten und Mechanismen für die persistente und aus Benutzersicht transparente Speicherung der Kooperationsobjekte bildet daher eine zentrale Forschungsfrage dieser Arbeit. Die Persistenzkonzepte müssen den Anforderungen der *Offline-Verfügbarkeit* entsprechen, also eine Objektverfügbarkeit gewährleisten, auch wenn der Benutzer vom Rest der Kooperationsumgebung getrennt ist. In den folgenden Abschnitten werden Ansätze für verteilte Persistenzsysteme in dynamischen Peer-to-Peer-Netzwerken insbesondere unter diesem Gesichtspunkt betrachtet.

4.3.1 Verteilte Persistenzsysteme

Eine verteilte Speicherung von Daten wird in Peer-to-Peer-Netzwerken oft mit Filesharing-Anwendungen gleichgesetzt. Tatsächlich kamen viele Impulse zu den derzeitigen Peer-to-Peer-Speicherlösungen aus diesem Bereich. Das wesentliche Problem von Peer-to-Peer-basierten Filesharing-Anwendungen ist das Auffinden der im Netzwerk verfügbaren Daten. Diese Fragestellung ist mit dem Auffinden der verfügbaren Dienste in verteilten Umgebungen verwandt (vgl. Abschnitt 4.2.2).

Einer der frühen Filesharing-Dienste, *Napster*¹⁴, benutzte zu diesem Zweck Indexserver, die ein Verzeichnis der Speicherorte der verfügbaren Dateien bereitstellen. Zu diesem Zweck schicken die Peers Listen ihrer lokal gespeicherten Dateien an den Indexserver, der diese in sein Verzeichnis aufnimmt. Sucht ein Peer eine bestimmte Datei, fragt er diese beim Indexserver an, der daraufhin mögliche Speicherorte zurückliefert. Die eigentliche Datenübertragung erfolgt direkt zwischen den Peers sobald ein Peer mit der Datei gefunden ist. Der Schwachpunkt eines solchen Vorgehens ist die Abhängigkeit des Gesamtsystems von eben diesen Indexservern. Ohne sie können die Daten im Peer-to-Peer-Netzwerk nicht lokalisiert werden.

¹⁴ Napster ist inzwischen ein kommerzieller Musikdienst, der außer dem Namen kaum noch Gemeinsamkeiten mit dem ursprünglichen Filesharing-Netzwerk besitzt. <http://www.napster.com>

*Gnutella*¹⁵ [Karbhari et al., 2004] versucht dieses Manko zu beheben, indem es Anfragen als Broadcast an verbundene Peers schickt. Sollten diese die gesuchte Datei nicht selber bereitstellen, leiten sie die Anfrage ihrerseits an ihre Nachbar-Peers weiter. Um das hohe Kommunikationsaufkommen aufgrund einer derartigen Suchstrategie zu reduzieren, bricht die Suche nach einer vorbestimmten Zahl von Weiterleitungen der Suchanfrage ab und vermeidet auch eine Ringsuche, bei der bereits befragte Knoten erneut angefragt werden. Trotz dieser Maßnahmen erzeugt eine derartige Suchstrategie noch immer einen immensen Kommunikations-Overhead, ohne garantieren zu können, dass eine derartige Suche bei Existenz der gewünschten Daten erfolgreich ist.

Distributed Hashtable (DHT)

Eine Lösung für das Dilemma des Auffindens von Daten in Peer-to-Peer-Netzwerken bietet das Verteilen der Daten auf die einzelnen Peers nach einem eindeutigen und reproduzierbaren Schema. Diesen Ansatz verfolgen das Konzept der *Distributed Hashtables*, das sich an dem Funktionsprinzip der klassischen Hash-Tabellen orientiert. Von den zu speichernden Daten wird mittels einer *Hash-Funktion*¹⁶ ein *Hash-Wert*¹⁷ ermittelt, der dann als *Index*¹⁸ darüber entscheidet, an welcher Position in der *Hash-Tabelle* die entsprechenden Daten hinterlegt werden.

Das grundlegende Vorgehen ist hierbei die Indizierung und Zuweisung von Werten auf feste Positionen einer Tabelle. Dabei hat die Tabelle eine festgelegte Größe, aus der sich ein ebenso fester Adressraum ablesen lässt. Ein solches Verfahren ermöglicht einen nahezu direkten Zugriff auf gesuchte Daten. Allerdings bietet dieser Ansatz als Suchmöglichkeit nur die Form des so genannten „*Exact-Matching*“ an. Anfragen führen nur dann zum Erfolg, wenn der Suchbegriff exakt mit dem Wert übereinstimmt über den der Hashwert erzeugt wurde.

Eine DHT ist eine algorithmische Umsetzung der Abbildung von Indizes aus einer Hash-Tabelle auf die Rechnerknoten in einem Netzwerk. Möglichkeiten und Eigenschaften einer Hash-Tabelle werden so auf Rechnernetze übertragen und können zur Speicherung von Daten, Routing- und Verteilungsinformationen genutzt werden. Die Hash-Tabelle wird dazu auf die Knoten eines Peer-to-Peer-Netzes verteilt gespeichert. Jeder Knoten eines solchen Netzwerkes ist ein Index in der verteilten Hash-Tabelle und verwaltet ein Segment des Hash-Bereiches der Daten. Die Segmente werden von den Nachbarknoten im Hash-Bereich begrenzt.

Daten, die in der DHT gespeichert werden sollen, werden über eine eindeutige Hash-Funktion in denselben Wertebereich abgebildet wie die Netzwerknoten. Sie werden auf dem Knoten gespeichert, der ihrem Hash-Wert am „nächsten“ ist. Die Berechnung der „Nähe“ hängt dabei von deren Definition in der verwendeten DHT

¹⁵<http://www.the-gdf.org/wiki>

¹⁶Eine Abbildung $h : K \rightarrow S$ heißt *Hash-Funktion*, wenn $|K| \geq |S|$ gilt.

¹⁷Ein *Hash-Wert* ist ein skalarer Wert, der aus einer komplexeren Datenstruktur (Zeichenketten, Objekte, ...) mittels einer Hash-Funktion berechnet wird.

¹⁸Der *Index* ist eine Positionsangabe innerhalb strukturierter Speicherbereiche.

Umsetzung ab. Über Verfahren der DHTs kann somit effizient herausgefunden werden, welcher Knoten eines Netzwerks für ein bestimmtes Datum verantwortlich ist.

In Peer-to-Peer-Netzwerken kennt ein Peer meist nur seine direkten Nachbarn. Somit können viele Daten nicht direkt an den Zielknoten gesandt werden. Darum setzen DHTs ein *Key-Based Routing (KBR)* ein, in dem die Daten gemäß ihres *Hash-Wertes (key)* durch das Peer-to-Peer-Netzwerk an ihr Ziel gesandt werden. Dadurch entsteht ein Routingverfahren in der Anwendungsschicht, das das Routing der Netzwerkschicht überlagert – man spricht daher in Bezug auf DHTs auch von *Overlay-Netzwerken*.

Der Weg im physikalischen Netzwerk wird durch eine schrittweise Vorgabe von bekannten Zwischenzielen im logischen Netzwerk aus der Anwendungsebene vorgegeben. Einen Überblick über DHT-Systeme inklusive eines Vergleichs zu „klassischen“ Peer-to-Peer Filesharing-Anwendungen liefert [Balakrishnan et al., 2003].

Inzwischen existieren neben den DHTs weitere Verfahren, die auf das KBR zurückgreifen. Da DHTs keine gezielte Speicherung von Daten auf bestimmten Knoten zulassen und somit eine Lokalität der Daten vermissen lassen, bieten *Decentralized Object Location and Routing (DOLR)* Verfahren einen verteilten Verzeichnisdienst, der die Speicherposition der Daten statt der Daten selbst speichert.

Besonders im kooperativen Anwendungsfeld wird für die Kommunikation über das DHT Overlay-Netzwerk neben der Punkt-zu-Punkt-Kommunikation zwischen zwei Peers auch eine effiziente Gruppenkommunikation benötigt. Lässt sich Erstere gut über DHTs bewerkstelligen, indem eine Nachricht mit der nodeID des Zielknotens versendet wird, ist eine derart gestaltete Punkt-zu-Punkt Kommunikation mit allen Knoten einer Gruppe oft ineffizient. Die Nachrichten müssen dabei auf dem Weg zu den Zielknoten manche Knoten mehrmals passieren, wenn diese auf dem Pfad zu mehr als einem der Zielknoten liegen. Für die Gruppenkommunikation existieren daher die so genannten *CAST* Verfahren. Sie implementieren eine effiziente und verteilte *Anycast-* und *Multicast-Kommunikation* in einem KBR-Netzwerk.

Die hier vorgestellten Verfahren auf Basis von DHTs werden von den Peer-to-Peer-Anwendungen genutzt, um eine fehlerresistente und skalierbare Netzwerkkommunikation und Objektspeicherung zu erreichen. [Dabek et al., 2003] ordnet die einzelnen Lösungen folglich in drei Schichten (*engl. tiers*) ein. Die unterste Schicht (*tier 0*) bildet das *Key-Based-Routing (KBR)*. Auf diese Schicht setzen die höheren *Protokollabstraktionen* in *tier 1* auf. Dazu gehören neben dem eigentlichen *Distributed HashTables (DHT)* auch das *Decentralized Object Location and Routing (DOLR)* und *Any- und Multicast (CAST)*. In der obersten Schicht (*tier 2*) sind alle Anwendungen eingeordnet, die die Protokollabstraktionen nutzen oder gar direkt auf das KBR zurückgreifen.

Zu den ersten vier Verfahren, die ein KBR für Peer-to-Peer-Netzwerke bereitstellen gehören *CAN* [Ratnasamy et al., 2001], *Chord* [Stoica et al., 2001a], *Pastry* [Rowstron und Druschel, 2001a] und *Tapestry* [Zhao et al., 2004]. Sie stellen neben der Routing-Schicht (*tier 0*) auch einen Teil der Protokollabstraktionen aus *tier 1* bereit. Inzwischen wurden einige der Verfahren stark weiterentwickelt und sind Vorbild für neue verbesserte Implementierungen geworden. Als Beispiel ist nach dem Vorbild

von *Pastry Bamboo* [Rhea et al., 2005b] entstanden. Dieses kann besser auf Änderungen im Peer-to-Peer-Verbund reagieren. Aus *Chord* ist *Kademlia* [Maymounkov und Mazières, 2002] hervorgegangen, welches u. a. in den Filesharing-Anwendungen *eMule*¹⁹ und *Bittorrent*²⁰ eingesetzt wird.

Um eine Kompatibilität zwischen den unterschiedlichen Implementierungen herzustellen, schlägt [Dabek et al., 2003] vor, an den Schnittstellen zwischen den Schichten eine einheitliche Schnittstelle (*Application Programming Interface (API)*) zu schaffen. Diese würde den Entwicklern von Peer-to-Peer-Anwendungen erlauben, die unterschiedlichen DHT-Verfahren gegeneinander auszutauschen, bzw. mit verschiedenen Overlay-Netzwerken zusammenzuarbeiten. Die meisten der derzeitigen Anwendungen sind aber noch klar einem bestimmten DHT-Verfahren zugeordnet.

Für *Pastry* existieren mit *PAST* [Rowstron et al., 2001] ein verteilter Objektspeicher, mit *Scribe* [Castro et al., 2002] eine Multicasting Lösung und mit *Splitstream* [Castro et al., 2003] sogar eine Streaming-Anwendung. Auf Chord setzt das verteilte Dateisystem *CFS* [Dabek et al., 2001] auf. *Tapestry* nutzen *OceanStore* [Kubiatowicz et al., 2000] als ewigen Objektspeicher und *Bayeux* [Zhuang et al., 2001] als Multicast-Anwendung.

Während die meisten der genannten Anwendungen Protokollabstraktionen in *tier 1* nutzen, gibt es auch Ausnahmen, die wie z. B. die *Internet Indirection Infrastructure (I3)* [Stoica et al., 2002] lediglich das Key-based Routing in *tier 0* nutzen. I3 erlaubt eine asynchrone Datenkommunikation in Peer-to-Peer Netzwerken und setzt dazu auf *Chords KBR* auf. Diese Unterstützung asynchroner Kommunikation zwischen den Peers wird zusätzlich in *ROAM* genutzt, um in einem Peer-to-Peer-Netzwerk der Mobilität der Peers stärker Rechnung tragen zu können [Zhuang et al., 2003]. Einen Überblick über die Zusammenhänge der vorgestellten DHT-basierten Protokolle und Peer-to-Peer-Anwendungen gibt Abbildung 4.3.

Eine Aussage über die generelle Eignung DHT-basierter Persistenzsysteme für die mobil-verteilte Kooperationsumgebungen kann nur schwer getroffen werden. Zum einen spannen DHT-Netzwerke selbstverwaltete und fehlerresistente Anwendungs-umgebungen auf, zum anderen verfügen sie derzeit kaum über Lösungsansätze für eine Offline-Verfügbarkeit der gemeinsam genutzten Datenbestände. Um das Potential der DHTs für die gewünschte Kooperationsumgebung zu durchleuchten wird daher zunächst ein vielversprechender Vertreter bezüglich der Etablierung einer mobilitätsfreundlichen verteilten Persistenz durchleuchtet.

Realisierung einer DHT-basierten Persistenzschicht am Beispiel von *Pastry* und *PAST*

Das effiziente Auffinden von Objekten und das Übertragen von Nachrichten sind Schlüsselfaktoren einer verteilt verwalteten und gespeicherten Kooperationsumgebung. Die Eigenschaften des Routingsverfahrens und die Strategien zur Replikation der Objekte innerhalb des verteilten Speichers bestimmen wesentlich die Eignung der

¹⁹<http://www.emule-project.net>

²⁰<http://www.bittorrent.com/>

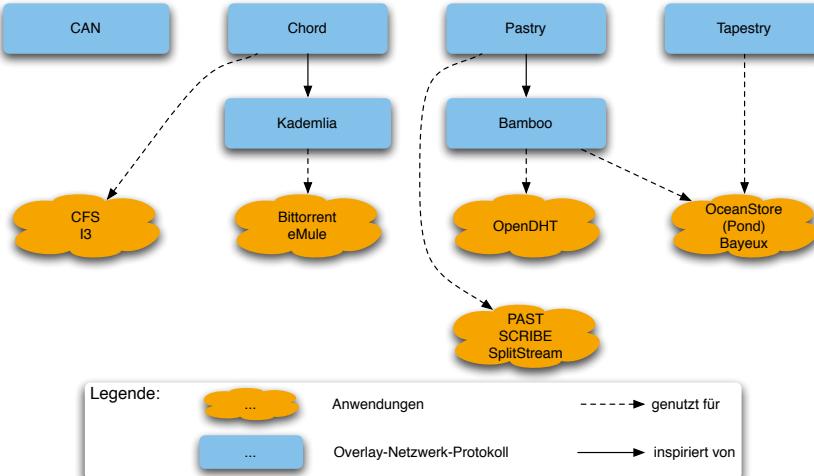


Abbildung 4.3: Auf DHTs basierende Overlay-Netzwerk-Protokolle und deren Anwendungen

Persistenzschicht für eine mobil-verteilte Kooperationsumgebung gemäß den Anforderungen in Kapitel 3. Daher sollen nun stellvertretend für die Gruppe der DHT Systeme die grundlegenden Mechanismen von *Pastry* inklusive der Persistenzschicht *PAST* und der Any- und Multicast Lösung *SCRIBE* betrachtet werden (Die Betrachtung von *SCRIBE* erfolgt im Abschnitt *Ereigniskontrolle (4.4)*). Die Wahl fiel aus zwei Gründen auf *Pastry* und seine Anwendungen: Zum einen handelt es sich um etablierte Vertreter der Gruppe der DHT-Lösungen und zum anderen ergänzen sich die zur Verfügung gestellten Mechanismen zu den benötigten Grundfunktionalitäten einer Kooperationsumgebung. Die Diskussion eines mobil-verteilten Kooperationsystems auf Basis dieser Verfahren findet sich Kapitel 5.

Das Design von *Pastry* beinhaltet Mechanismen für ein selbst organisierendes, virtuelles Netzwerk. In solch einem Netzwerk hat jeder Netzwerkknoten die Aufgabe, eingehende Nachrichten zu verarbeiten und eigenständig weiterzuleiten. Es erfolgt keinerlei Steuerung der Nachrichtenvermittlung von außen.

Der Adressraum von *Pastry* hat eine Größe von 128-Bit. Innerhalb dieses Adressraumes verfügt jeder Knoten über eine eindeutige *Pastry-Adresse (nodeID)*²¹. Mittels einer zufälligen Vergabe der nodeIDs wird eine gleichmäßige Verteilung der Knoten in dem Adressraum sichergestellt²². Diese zufällige Verteilung verhindert mit großer Wahrscheinlichkeit, dass örtlich benachbarte Knoten auch im logischen *Pastry-Adressraum* direkt nebeneinander liegen. Ein Ausfall in einem physikalischen Netzwerksegment hat so nicht gleich den Ausfall eines kompletten *Pastry Segments*

²¹In einem *Pastry* Netzwerk sind somit bis zu $2^{128} - 1$ Knoten möglich.

²²Eine alternative Möglichkeit ist es, die nodeIDs aufgrund eindeutiger Daten, wie zum Beispiel der Macadresse oder dem Namen des Rechners, automatisch erzeugen zu lassen.

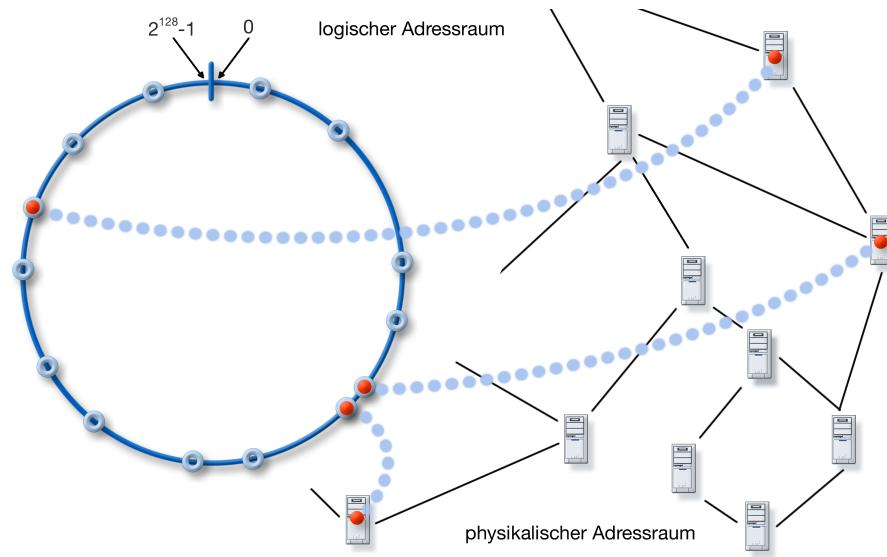


Abbildung 4.4: Ein physikalisches Netzwerk von Knoten wird in einen logischen Adressraum abgebildet. Hierbei impliziert die örtliche Nähe zweier Knoten keineswegs auch eine Nähe der logischen Adressen und umgekehrt

zur Folge – verteilt auftretende Lücken in der logischen Netzwerkstruktur lassen sich oft ohne eine komplette Neustrukturierung des Netzwerks schließen. Die Entkopplung der physikalischen Position der Netzwerkknoten vom logischen Adressraum wird in Abbildung 4.4 noch einmal verdeutlicht.

Nachrichten²³, die innerhalb des Pastry Netzwerkes versandt werden sollen, bekommen einen *Schlüssel (key)* aus dem gleichen Adressraum wie die nodeIDs zugeordnet. Eine Nachricht wird dabei stets an den Knoten gesandt, dessen nodeID dem Nachrichtenschlüssel am nächsten liegt.

[Rowstron und Druschel, 2001a] beschreiben die prinzipielle Idee des Routingverfahren wie folgt: Nachrichtenschlüssel und nodeID werden als Ziffernfolgen zur Basis 2^b (b ist ein beliebiger aber fester Wert) angenommen. In jedem Routingschritt wird die Nachricht vom aktuellen Knoten an einen Knoten übergeben, dessen nodeID eine Ziffer (oder b Bits) mehr Übereinstimmung mit dem Nachrichtenschlüssel hat als der aktuelle Knoten. Die übereinstimmenden Ziffern werden als *Präfix (engl. prefix)* bezeichnet. Schritt für Schritt stimmen so immer mehr Stellen des Nachrichtenschlüssels mit der nodeID des aktuellen Knotens überein. Kann bei einem Routingschritt kein Knoten mit längerem Präfix gefunden werden, wird der Knoten ermittelt, der den numerisch geringsten Abstand zu dem Nachrichtenschlüssel hat. Dieser ist der Zielknoten der Nachricht.

²³Nachrichten können u. a. Anfragen, Anweisungen, zu speichernde Daten oder auch Verwaltungsinformationen beinhalten.

Aufgrund des Designs des Routings in Pastry erreicht ein Schlüssel in einem Pastry-Netzwerk mit N Knoten und aktuellen Routingtabellen sein Ziel in $\lceil \log_2 N \rceil$ Schritten. Dabei ist b ein beliebig gewählter Konfigurationsparameter, der die Größe der Routingtabellen bestimmt. Pastry initialisiert normalerweise $b = 4$ [Rowstron und Druschel, 2001a].

Verwaltungsinformationen Für das Routing sind Angaben anderer Knoten und weitere Informationen nötig. Diese werden in jedem Knoten in drei Tabellen festgehalten. Hierbei wird zwischen *Routing Table*, *Neighborhood Set* und *Leaf Set* unterschieden:

Die *Routing Table* (R) enthält $\lceil \log_2 N \rceil$ Zeilen. Jede Zeile enthält $2^b - 1$ Einträge, welche aus einer nodeID und der dazugehörigen Netzwerkadresse bestehen. Dabei enthält die Zeile k (beginnend bei 0) immer nodeIDs mit gleichem Präfix der Länge k . Dieses Präfix stimmt mit den ersten k Stellen der eigenen nodeID überein. Ab der Stelle $k + 1$ differieren die nodeIDs von der eigenen nodeID. Die Einträge jeder Zeile sind von links nach rechts nummerisch aufsteigend sortiert. Die Wahl des Konfigurationsparameters b bestimmt also den Trade-off zwischen Grösse der Routing Tabelle ($\lceil \log_2 N \rceil \times (2^b - 1)$) und der Anzahl der maximalen Routing-Schritte ($\lceil \log_2 N \rceil$), die benötigt werden, um eine Nachricht zwischen zwei beliebigen Knoten des Netzwerkes zu versenden.

Das *Neighborhood Set* (M) enthält eine Liste der – gemäß eines festgelegten *Maßes der Nähe* (engl. *Proximity Metric*)²⁴ – nächsten $|M|$ Nachbarn. Sie dient nicht direkt dem Routingverfahren, sondern wird als Entscheidungshilfe bei der Erstellung der Routing-Einträge eingesetzt.

Das *Leaf Set* (L) enthält $\lfloor \frac{|L|}{2} \rfloor$ Einträge, die kleiner als die eigene nodeID sind und $\lfloor \frac{|L|}{2} \rfloor$ Einträge, die größer als die eigene nodeID sind. Die Einträge im Leaf Set haben die besondere Eigenschaft, dass ihre *nodeID numerisch* am nächsten bei der eigenen nodeID liegen. Dadurch befinden sich diese Knoten im Adressraum in logischer Nähe, was allerdings keinen Rückschluss auf ihre physikalische Nähe zulässt, da die Vergabe der *nodeID* zufällig geschieht. Das Leaf Set ist direkter Bestandteil des Routingverfahrens und bestimmt zu einem wesentlichen Teil auch die Ausfallsicherheit des Netzwerkes. Nachrichten im Pastry-Netzwerk erreichen nur dann garantiert ihr Ziel, wenn nicht mehr als $\lfloor \frac{|L|}{2} \rfloor$ benachbarte Knoten gleichzeitig ausfallen. $|L|$ ist ein Konfigurationsparameter und hat üblicherweise einen Wert von 16 oder 32. Abbildung 4.5 veranschaulicht den Zusammenhang von Routingverfahren und Tabellen anhand eines Beispiels.

Routing Jede Nachricht im Pastry-Netzwerk wird über einen Schlüssel (key) identifiziert, der aus demselben Wertebereich wie die nodeIDs stammt. Während dies bei Nachrichten die nodeID des Zielknotens sein kann, ist es bei zu speichernden Objekten ein mittels einer fest definierten Hash-Funktion über eine bestimmte Objek-

²⁴Das *Maß der Nähe* definiert, wann ein Knoten einem anderen *nahe* ist. Nähe kann z. B. eine besonders gute physikalische Netzwerkverbindung zwischen den Knoten sein.

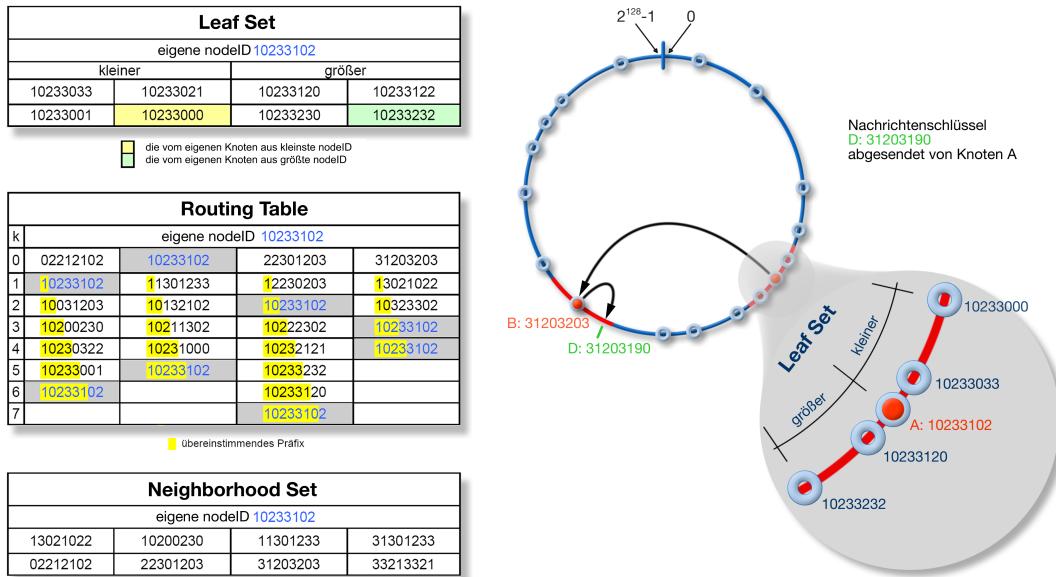


Abbildung 4.5: Verwaltungstabellen und grafische Darstellung der Routinginformationen eines Pastry-Knotens mit der nodeID 10233102, $b = 2$ und $l = 8$ (Alle Ziffern haben die Basis 4)

teigenschaft, wie den Objektnamen oder den Inhalt, ermittelter Wert. Das Pastry-Routingverfahren stellt dabei sicher, dass Nachrichten wie Objekte auch zu den errechneten Zielknoten gelangen.

Sobald eine Nachricht mit dem Schlüssel D einen Knoten A erreicht, wird zunächst geprüft, ob dieser Nachrichtenschlüssel im Wertebereich der im *Leaf Set* gespeicherten nodeIDs liegt und somit von seiner logischen Adresse her in unmittelbarer Nähe des aktuellen Knotens liegt. In diesem Fall wird die Nachricht direkt an den Knoten aus dem *Leaf Set* gesandt, dessen *nodeID* dem Nachrichtenschlüssel D numerisch am nächsten liegt, wobei auch die eigene nodeID berücksichtigt wird. Sollte jedoch der Schlüssel D nicht innerhalb des Wertebereichs des *Leaf Set* liegen, wird über die *Routing-Tabelle* ein Knoten ermittelt, der eine um eine Stelle längeres Präfix als der eigene Knoten besitzt. Dazu wird als Erstes die Anzahl der übereinstimmenden Stellen im *Präfix* zwischen der eigenen nodeID A und dem Schlüssel D ermittelt. Da die Zeile k die nodeIDs enthält, die auf k Stellen mit der nodeID des aktuellen Knotens übereinstimmen, schlägt das Verfahren in der Zeile $k+1$ der Routingtabelle nach. Sollte dort kein passender Eintrag vorhanden sein oder der korrespondierende Knoten nicht erreichbar sein, wird die Nachricht unter Berücksichtigung der Verwaltungstabellen zu jenem Knoten weitergeleitet, der über die gleiche Präfixlänge wie der aktuelle Knoten A zum Nachrichtenschlüssel D verfügt, jedoch numerisch näher an dem Nachrichtenschlüssel D liegt.

Als Abwandlung dieses Routingverfahrens, das lediglich die Anzahl der Stellen der Präfixübereinstimmung einbezieht, erlaubt Pastry, die Nähe der Knoten zueinander

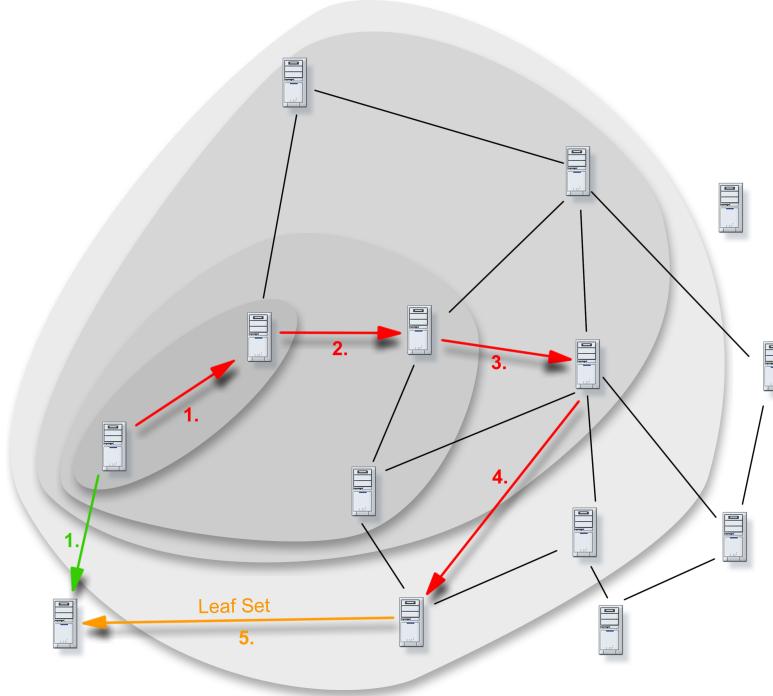


Abbildung 4.6: Bei Routing-Entscheidungen, die das Lokalitätsprinzip nicht berücksichtigen, werden zwar gültige Routingschritte in den Knoten durchgeführt, diese führen aber nicht notwendigerweise zu einer optimalen Gesamtroute

gemäß des vorgegebenen *Maßes der Nähe* zu betrachten. Dieses Vorgehen erlaubt, die *Lokalität* der Knoten für den Aufbau der Verwaltungstabellen zu berücksichtigen und so ungünstige Routing-Pfade innerhalb des Pastry-Netzwerkes zu vermeiden. Abbildung 4.6 zeigt je einen Routing-Pfad mit und ohne Berücksichtigung des Lokalitätprinzips. [Rowstron und Druschel, 2001a] weisen darauf hin, dass Pastry bei Berücksichtigung des *Maßes der Nähe* beim Aufbau der Verwaltungstabellen und Durchführen des Routings einen stets „guten“ Routing-Pfad wählt.

Selbstorganisation Aufgrund der Selbstorganisation im Pastry-Netzwerk, muss ein neuer Knoten mit nodeID X zunächst einen Einstiegsknoten ausfindig machen²⁵, dessen nodeID als A bezeichnet sei. X schickt zunächst eine *join*-Nachricht an den Knoten A . Dabei ist der Nachrichtenschlüssel gleich der nodeID X des neuen Knotens. Von Einstiegsknoten wird die *join*-Nachricht über das Pastry-Routingverfahren zu dem Knoten Z gesandt, dessen nodeID dem Nachrichtenschlüssel X numerisch am nächsten ist. Alle Knoten auf dem Pfad vom Einstiegsknoten A zum Zielknoten Z

²⁵Das Aufspüren von Einstiegsknoten in das Pastry-Netzwerk kann z. B. durch einen so genannten *expanding ring multicast* geschehen.

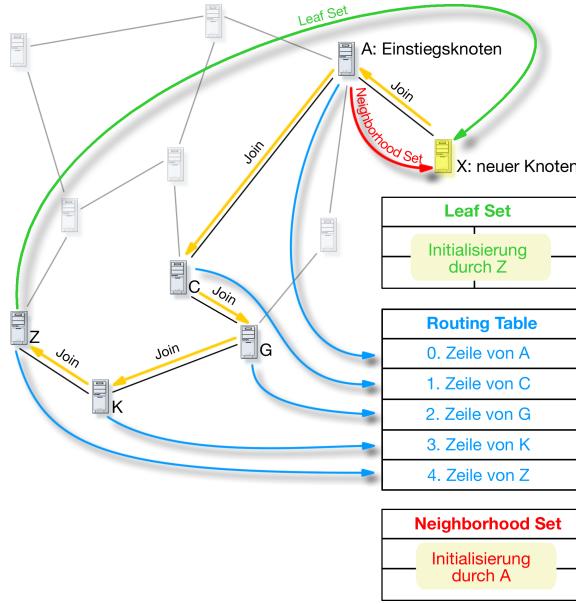


Abbildung 4.7: Initialisierung der Verwaltungstabellen beim Anmelden eines neuen Knotens in ein Pastry-Netzwerk

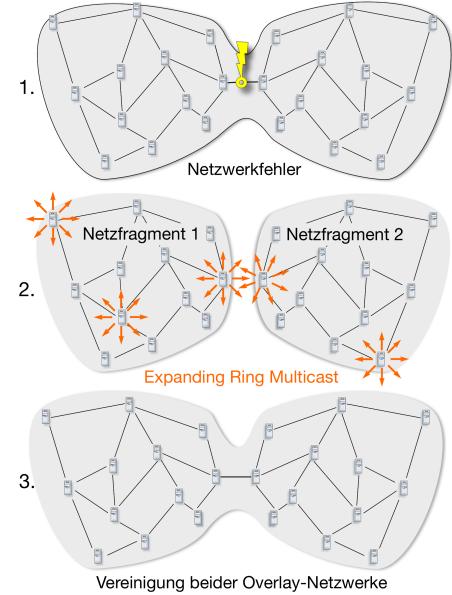


Abbildung 4.8: Pastrys Selbstorganisation im Fall einer Netzwerksegmentierung

senden zugleich die eigenen Verwaltungstabellen an den neuen Knoten X, der aus diesen Informationen seine eigenen Verwaltungstabellen initialisiert:

Da der Einstiegsknoten A in der Nähe vom neuen Knoten X liegt, übernimmt der neue Knoten dessen *Neighborhood Set*. Der Knoten Z wiederum ist der Knoten mit dem geringsten numerischen Distanz zur nodeID X des neuen Knotens und liefert daher sein *Leaf Set* als initiales *Leaf Set* für den neuen Knoten X.

Der Inhalt der *Routing Table* wird in mehreren Schritten ermittelt. Die anfangs leere *Routing Table* wird von Zeile 0 an erzeugt. Im Normalfall haben die nodeIDs A und X kein gemeinsames Präfix. Daher kann die Zeile 0 der *Routing Table* von A in die Zeile 0 der *Routing Table* von X übernommen werden. Nacheinander steuert jeder Knoten auf dem Nachrichtenpfad die fehlenden Zeilen bis hin zur Zeile i bei, wobei i die Länge des mit dem Nachrichtenschlüssel X übereinstimmenden Präfixes ist (vgl. Abbildung 4.7).

Bei einem übereinstimmenden Präfix mit i Stellen des Einstiegsknotens A und des neuen Knotens X werden die ersten i Zeilen der *Routing Table* von A in die von X übernommen. Abschließend informiert der neue Knoten alle in seinen Verwaltungstabellen eingetragenen Knoten durch das Senden der eigenen Verwaltungstabellen [Rowstron und Druschel, 2001a].

Sollte ein Knoten im Pastry-Netzwerk ohne Warnung ausfallen, kann dies unter anderem dadurch festgestellt werden, dass er nicht mehr über seine topologischen Nachbarn erreichbar ist. In einem solchen Fall müssen die Verwaltungstabellen *Leaf*

Set, *Routing Table* und *Neighborhood Set* reorganisiert werden, um ein korrektes Routing zu gewährleisten.

Um einen ausgefallenen Knoten im *Leaf Set* eines Knotens zu ersetzen wird zuerst der Knoten mit der größten erreichbaren nodeID in der entsprechenden Hälfte des *Leaf Sets* kontaktiert, dessen *Leaf Set* angefordert und mit dem eigenen verglichen. Nach dem Prüfen der Erreichbarkeit wird der Knoten in das eigene *Leaf Set* übernommen, der numerisch am nächsten zum ausgefallenen Knoten liegt. Diese neue nodeID ersetzt nun die nodeID des ausgefallenen Knotens. So hat jeder Knoten die Möglichkeit, sich selbst zu aktualisieren, solange nicht mehr als $\left\lfloor \frac{|L|}{2} \right\rfloor$ Knoten des *Leaf Sets* gleichzeitig ausfallen.

Bei einem Ausfall eines Knotens in der *Routing Table* werden alle anderen nodeIDs aus der Zeile der ausgefallenen nodeID der Reihe nach kontaktiert. Dabei wird geprüft, ob diese in der *Routing Table* an der Stelle des ausgefallen Knotens einen Eintrag haben, der erreichbar ist. Ist dies nicht der Fall, wird in der nächsten Zeile dasselbe Verfahren angewandt. So wird sichergestellt, dass ein im Pastry-Netzwerk vorhandener Zielknoten auf jeden Fall gefunden wird.

Bei Ausfall eines Knotens im *Neighborhood Set* werden die noch verfügbaren Knoten aus dem *Neighborhood Set* kontaktiert und deren *Neighborhood Sets* angefordert. Die neuen nodeIDs werden gemäß des *Maßes der Nähe* auf ihre Nähe zum eigenen Knoten geprüft. Die nächsten Knoten werden dann in das eigene *Neighborhood Set* übernommen.

Pastry ist also in der Lage, selbstständig auf Knotenausfälle zu reagieren und die Verwaltungstabellen zu korrigieren. Dabei nutzen die Knoten stets die Informationen der verbliebenen Knoten im Netzwerk, um sich zu reorganisieren. [Rowstron und Druschel, 2001a] zeigt darüber hinaus Möglichkeiten auf, wie auf sporadische Knotenausfälle und bösartige Knoten im Netzwerk reagiert werden kann.

Durch die selbstständige Reorganisation des Pastry-Netzwerkes kann es bei dem Ausfall von Knoten vorkommen, dass ein Pastry-Netzwerk partitioniert und sich über eine Reorganisation der Verwaltungstabellen voneinander unabhängige Netzwerke bilden. Um derart getrennte Pastry-Netzwerke wieder zusammenführen zu können, versuchen die Knoten eines Pastry-Netzwerkes über periodische *Expanding Multicast-Nachrichten* andere Pastry-Netzwerke zu finden und sich mit diesen über eine Reorganisation zu vereinen. Dieser Vorgang wird in Abbildung 4.8 noch einmal veranschaulicht.

Verteilte Speicherung von Objekten in PAST

Das Pastry Routing-Verfahren implementiert ein selbstorganisiertes und gegen Knotenausfälle resistentes Overlay-Netzwerk, in dem Nachrichten effizient und sicher zu den teilnehmenden Knoten gesandt werden können. Für die verteilte Speicherung von Objekten in dem Pastry-Netzwerk existiert mit *PAST* [Rowstron und Druschel, 2001b] eine Persistenzschicht für Pastry, die neben der verteilten Speicherung und Suche von Objekten auch eine Replikation und Verschlüsselung der gespeicherten Daten erlaubt. Ziel von PAST ist die persistente Speicherung von Daten in einem

DHT-basierten Peer-to-Peer-Netzwerk unter den Gesichtspunkten der Verfügbarkeit, Skalierbarkeit und Sicherheit. PAST stellt zu diesem Zweck einen globalen Speicher ohne zentrale Verwaltung zur Verfügung. Bei der Entwicklung der PAST-Architektur wurden folgende Designziele berücksichtigt:

- *Effizienz*: Für eine effiziente Verwaltung der Daten in einem Peer-to-Peer-Netzwerk wird das Pastry-Routingverfahren zum Verteilen und Anfordern von Daten eingesetzt. Dies erlaubt Antworten auf Suchanfragen in durchschnittlich weniger als $\lceil \log_2 N \rceil$ Suchschritten.
- *Replikation*: Die Objekte werden in einen engen Bereich im logischen Adressraum des Pastry-Netzwerkes repliziert. Durch die zufällige Natur des Pastry-Adressraums bedeutet dies eine weit gestreute Replikation in das physikalische Netzwerk. Die Replikation kommt ebenfalls ohne zentrale Steuerung aus.
- *Dezentralisierte Datenverwaltung*: Dank des Pastry-Netzwerkes besitzt jeder Knoten die Informationen, die er benötigt, um auf ein Objekt zuzugreifen. Die zufällige Natur des Adressraums stellt dabei eine balancierte Speicherung der Objekte im Netzwerk sicher. Durch Zwischenspeichern häufig gestellter Anfragen (Caching) werden kürzere Antwortzeiten erreicht [Druschel und Rowstron, 2001].
- *Quotaverwaltung*: Damit der im Pastry-Netz zur Verfügung stehende Speicher nicht unkontrolliert verbraucht wird, kann jedem Benutzer eine feste Speichergrenze zugewiesen werden [Rowstron et al., 2001].

PAST erlaubt jedem Knoten, Objekte in den verteilten Speicher abzulegen oder selbst Speicher anzubieten. Die Knoten verwaltet PAST mit der von Pastry erzeugten 128 Bit langen nodeID. Eine Datei besitzt im verteilten Speicher eine 160 Bit lange fileID die beim Aufruf der `insert()`-Funktion erzeugt wird. Die fileID ist ein Hash-Wert, der aus dem gegebenen Dateinamen, dem öffentlichen Schlüssel des Besitzers und einer zufälligen Komponente berechnet wird. Die zufällige Komponente sorgt dafür, dass die Datei eine mit hoher Wahrscheinlichkeit eindeutige fileID erhält. Zusätzlich wird zu jeder Datei ein Zertifikat mit den Einträgen *File-ID*, *Replikationsfaktor (k)*, *Streuungsfaktor*, *Einfügedatum* und *Hash-Wert des Dateiinhaltes* erzeugt. Das Zertifikat wird außerdem durch den Besitzer der Datei signiert und danach zusammen mit der verschlüsselten²⁶ Datei im PAST-Netzwerk gespeichert.

Zur Speicherung einer Datei im PAST-Netzwerk wird zunächst mittels des Pastry-Routingverfahrens ein passender Zielknoten ermittelt, dessen nodeID den numerisch geringsten Abstand zur fileID hat. Dabei werden nur die 128 höchstwertigen Bits der 160 Bit langen fileID für den Vergleich herangezogen. Gemäß des *Replikationsfaktors (k)* werden aus dem *Leaf Set* des Zielknotens k im logischen Adressraum direkt benachbarte Knoten ausgewählt, deren nodeIDs numerisch am nächsten zur fileID liegen. Auf diesen Knoten wird eine Kopie der Datei abgelegt (siehe Abbildung 4.9).

²⁶Weiterführende Informationen zum Zugriffsschutz in PAST finden sich in [Rowstron et al., 2001].

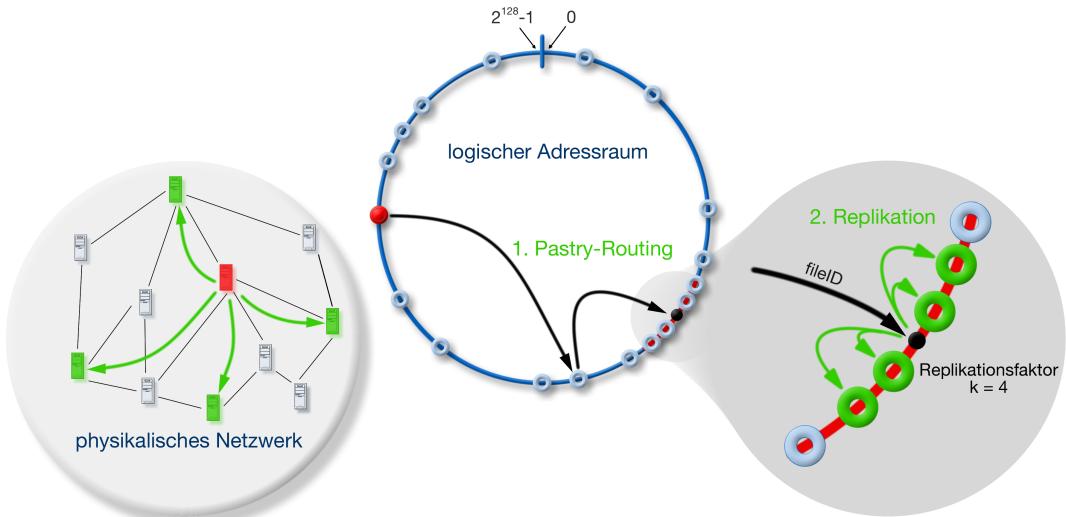


Abbildung 4.9: Das Replikationsverfahren von PAST mit dem Replikationsfaktor $k = 4$

Die Nähe der gespeicherten Replikate im logischen Adressraum hat eine breite Streuung der Replikate im physikalischen Netzwerk zur Folge. Andererseits sind die Replikate im logischen Adressraum leicht aufzufinden. Der *Replikationsfaktor* (k) bestimmt somit die Verfügbarkeit der Datei und kann je nach Bedarf frei gewählt werden²⁷. PAST bietet kein Verfahren um Daten verlässlich zu löschen. Der Besitzer einer Datei kann diese jedoch vom speichernden Knoten zurückfordern. Replikate auf Knoten, die das Netzwerk verlassen haben, sind von dieser Rückforderung allerdings nicht betroffen. Des Weiteren erfolgt auf den Knoten des PAST Netzwerkes eine Zwischenspeicherung von Dateien (Caching), um häufig angefragte Daten schnell verfügbar zu machen. Auch diese zwischengespeicherten Dateien werden eventuell durch die Rückforderung nicht erreicht.

Soll eine zuvor in PAST-Netzwerk gespeicherte Datei angefordert werden, wird die Anfrage zu dem erreichbaren Knoten geleitet, dessen nodeID numerisch am nächsten zu den 128 höchstwertigen Bits der gesuchten fileID ist. Wenn einer der k replizierenden Knoten im Netzwerk verfügbar ist, wird dieser Knoten aufgrund der Eigenschaften des Pastry-Routingverfahrens gefunden, so kann die Datei an die nodeID des anfragenden Knotens gesandt werden. Aufgrund der zufälligen Verteilung der Replikate im physikalischen Netzwerk, der dezentralen selbstorganisierenden Netzwerkverwaltung und einer unabhängigen Netzwerkanbindung der k replizierenden Knoten wird mit hoher Wahrscheinlichkeit einer der k Knoten verfügbar sein.

²⁷Auf Grund der Nutzung des *Leaf Sets* für die Auswahl der Replikationsknoten muss gelten $k < |L|$.

Diskussion DHT-basierter Persistenzkonzepte

PAST schafft durch die verwendete Speicherstrategie einen globalen und selbstorganisierenden Speicher basierend auf Peer-to-Peer-Technologie. Durch die Replikationsstrategie, die eine breite Verteilung der Replikate im physikalischen Netzwerk verfolgt, wird zudem eine hohe Verfügbarkeit der gespeicherten Daten unabhängig von der Zuverlässigkeit der einzelnen Netzwerkknoten erreicht. Allerdings lässt sich wegen der synchronen Befehlskommunikation in *PAST* eine replizierte Datei nicht zurückfordern (löschen), wenn replizierende Knoten zeitweise nicht verfügbar sind.

Ähnlich wie in *PAST* sind auch die weiteren auf DHT-Verfahren basierenden Persistenzkonzepte für ein möglichst zuverlässiges Gesamtsystem ausgelegt und betrachten zugreifende Knoten nur im Online-Fall. Vom Netzwerk getrennt haben sie keinen Zugriff auf die Persistenz. Das *OceanStore* Projekt verfolgt wie *PAST* ebenfalls die Bereitstellung eines globalen verteilten Speichers, in dem Dateien nach Aussagen der Entwickler nahezu *ewig* verbleiben sollen [Kubiatowicz et al., 2000]. Um die Speicherung der Daten so zuverlässig als möglich zu gestalten, versucht *OceanStore* Daten in Blöcke zerteilt auf zuverlässigen Knoten zu speichern. Zusätzlich bietet *OceanStore* neben einer Replikation auch die Versionierung der gespeicherten Daten an. Es kennt keine Methoden, um Daten (oder deren Versionen) wieder aus dem globalen Speicher zu entfernen. *OceanStore* ist für den kommerziellen von professionellen Dienstleistern verwalteten Einsatz ausgelegt. Mit *Pond* [Rhea et al., 2003] existiert auch ein erster Prototyp und ist, obwohl ursprünglich für *Tapestry* [Zhao et al., 2004] entworfen, inzwischen auch für den Pastry-Ableger *Bamboo* [Rhea et al., 2005b] verfügbar.

Bamboo bietet auch die Grundlage für *OpenDHT* [Rhea et al., 2005a], das ebenfalls einen global verfügbaren Speicher implementiert. Die Besonderheit von *OpenDHT* besteht in einer frei verfügbaren Infrastruktur, die für alle Interessenten zugänglich ist. Allerdings sind die Daten in *OpenDHT* nur für einen gewissen Zeitraum gespeichert und werden nach einer begrenzten Lebenszeit wieder aus dem verteilten Speicher entfernt.

Das *Cooperative FileSystem (CFS)* [Dabek et al., 2001] ist ein auf dem *Chord* Protokoll [Stoica et al., 2001b] aufsetzendes Dateisystem, das ähnlich wie *OceanStore* Dateien in Blöcke zerteilt und verteilt auf dem speichernden Knoten ablegt. Außerdem werden die Blöcke auf mehrere Knoten repliziert. CFS versucht dabei aber die Komplexität von *OceanStore* zu vermeiden und bietet daher keine aufwendigen Versionierungsmechanismen und Knotenauswahlverfahren.

Allen Konzepten ist gemeinsam, dass sie ihren Fokus auf die Aufrechterhaltung des Gesamtsystems legen. Sie kennen keine Mechanismen, um bestimmte Daten auf einem Knoten verfügbar zu halten, wenn dieser offline ist. Eine gezielte Replikation auf eine fest definierte Gruppe von Knoten ist ebenfalls nicht vorgesehen. Des Weiteren kennt nur *OceanStore* eine Versionierung der gespeicherten Daten, die es erlauben würde, Versionskonflikte bei gemeinschaftlich bearbeiteten Daten zu identifizieren und zu lösen.



Abbildung 4.10: Trade-off der Strategien zur Bereitstellung von persistenten Objekten auf mobilen und verteilten Knoten

Diese Designentscheidungen lassen die betrachteten Persistenzsysteme in ihrer derzeitigen Form für eine mobile-verteilte Kooperationsumgebung ungeeignet erscheinen. Offensichtlich existiert ein Trade-off zwischen der Skalierbarkeit des Persistenzsystems und der Verfügbarkeit der gespeicherten Daten (vgl. Abbildung 4.10). Interessant bleibt aber die Möglichkeit der Bereitstellung einer selbstorganisierenden, zuverlässigen und effizienten Peer-to-Peer-Persistenzschicht. Durch eine Anpassung der Replikationsstrategien und die Ergänzung um eine verteilte Versionskontrolle können DHT-basierte Persistenzsysteme die gewünschten Eigenschaften prinzipiell erfüllen.

Die Konzepte der Distributed Hashtables bieten somit einen geeigneten Ausgangspunkt für die gewünschte Persistenzschicht mobil-verteilter Wissensräume. Der Entwurf eines Konzeptes für eine auf DHT-Protokollen basierenden Persistenzschicht, das sich gezielt an den Bedürfnissen von mobil-verteilten Kollaborationsszenarien, wie in Kapitel 2 vorgestellt, orientiert, ist somit eine der zentralen Forschungsfragen dieser Arbeit. Um zusätzlich den Zugang zu solch einer Persistenzschicht zu erleichtern, ist es wichtig, die Komplexität der DHT-Protokolle für die Entwickler und Anwender bei der Nutzung des verteilten Speichers zu verbergen. Um diese Möglichkeiten zum Verbergen dieser Komplexität auszuloten, sind Ansätze für einen transparenten Zugriff auf einen verteilten Speicher zu betrachten.

4.3.2 Verteilte Persistenz als gemeinsamer Speicher

Der selbstorganisierende Charakter von verteilten Speicherkonzepten und die Heterogenität der zugrunde liegenden Netzwerkumgebungen verlangen Entwicklern wie Benutzern für den Zugriff auf derartige Systeme ein Verständnis der grundlegenden Konzepte und Mechanismen ab. Die Basistechnologien sind darüber hinaus nicht gegeneinander austauschbar oder parallel nutzbar, ohne die Komplexität weiter zu erhöhen. Die Entwickler von Kooperationswerkzeugen müssen sich daher schon zu Beginn auf eine Technologie festlegen oder im Nachhinein die Anwendungen mühsam auf alternative Persistenzsysteme portieren.

Dabek et al. schlagen daher für die DHT-Protokolle in [Dabek et al., 2003] eine Abstraktion der Programmierschnittstelle vor, die es erlaubt, Anwendungen für DHT-Netzwerke unabhängig vom genutzten DHT-Verfahren zu implementieren. Der Vorschlag bezieht sich aber ausschließlich auf DHT-Protokolle zur Vernetzung der Knoten – eine ähnliche Abstraktion für die auf diese aufsetzenden verteilten Persistenzsysteme ist nicht vorgesehen.

Aufgrund der Vielfalt und Komplexität der Schnittstellen zu verteilten Persistenzsystemen, wird die Aufmerksamkeit der Entwickler von der Kooperationsunterstützung weg zur technischen Lösung der Probleme der Objektspeicherung und -verwaltung sowie Koordinierung der teilnehmenden Knoten gelenkt. Da die Knoten einer mobilen Kollaboration oft offline sind, befinden sich die einzelnen Teile der mobil-verteilten Kooperationsumgebung mit einer gewissen Wahrscheinlichkeit nicht gleichzeitig im Kooperationsnetzwerk und müssen sich dennoch untereinander koordinieren. Der Entwickler benötigt daher Werkzeuge zur Lösung der Probleme der zeitlichen und räumlichen Entkopplung der Peers mit Hilfe der zu Grunde liegenden Netzwerk- und Persistenztechnik.

Das Problem eines gemeinsamen Datenraums für mehrere lose gekoppelte Prozesse existierte bereits vor dem Aufkommen von verteilten und über Peer-to-Peer-Netzwerke gekoppelten Systemen. Die Multiprozesskommunikation in Mehrprozessorsystemen warf ähnliche Fragestellungen auf. Obwohl auf derselben Hardwareplattform laufend, benötigen parallele Prozesse einen gemeinsamen Datenraum, der es ihnen erlaubt, miteinander zu kommunizieren, ohne direkt miteinander in Verbindung treten zu müssen.

Zu diesem Zweck entwarf Gelernter *Linda* [Gelernter, 1985; Gelernter und Carriero, 1989], ein Framework, das Prozessen ermöglicht, *Tuple* in einem *Tuple Space* genannten gemeinsamen Datenraum abzulegen und über diese zu kommunizieren. Die Tuple verbleiben bei Bedarf über die Lebenszeit des zugehörigen Prozesses hin aus im Tuple Space. Prozesse legen dazu ihre Daten mittels der Funktion `in()` als Tuple im Tuple Space ab, ohne sich um deren weitere Verarbeitung zu kümmern.

Ein Prozess, der an diesen Daten interessiert ist, greift auf die Tuple mittels der Funktionen `read()` (Tuple lesen und im Tuple Space belassen) oder `out()` (Tuple dem Tuple Space entnehmen) zu. Der Zugriff auf die Tuple findet dabei nicht direkt sondern über *Templates* statt. Mittels des Templates wird ein so genanntes *Matching* durchgeführt, bei dem die Tuple im Tuple Space mit dem Template verglichen wer-

den. Das erste übereinstimmende Tuple wird als Ergebnis zurückgeliefert. Sollte kein Tuple zu dem Template passen, verbleibt das Template so lange im Tuple Space, bis ein passendes Tuple in den Tuple Space eingefügt wird.

Tuple Spaces sorgen so für eine zeitliche Entkopplung der Kommunikation von parallelen Prozessen und verhindern, dass sich Prozesse gegenseitig in ihrer Kommunikation blockieren. Linda implementiert diesbezüglich einen flüchtigen Datenraum, der ursprünglich für die parallele Programmierung von Mainframes ausgelegt und nicht für verteilte Systeme vorgesehen ist. Der Tuple Space in Linda ist nicht persistent und skaliert aufgrund einer fehlenden inneren Struktur schlecht für große Datenräume.

Wegen einer fehlenden Persistenz der Tuple in Linda erweitert *Persistent Linda (PLinda)* [Anderson und Shasha, 1991] die Tuple Spaces um eine Datenbankfunktionalität für die Speicherung der Tuple. Ein ähnliches Konzept verfolgt auch der von *IBM Research* gewählte Ansatz der *TSpaces* [Wykoff et al., 1998; Lehman et al., 2001], der ebenfalls eine Datenbankanbindung für einen persistenten Tuple Space zur Verfügung stellt. Zusätzlich bietet der Ansatz eine Zugriffskontrolle auf Benutzer- und Gruppenebene.

TSpaces ist auf eine Verwendung mit der Java-Programmiersprache zugeschnitten und erlaubt eigene Objekttypen innerhalb des Tuple Space zu registrieren. Wie *TSpaces* ist auch die in *SUN Microsystems Jini Framework* integrierte Tuple Space-Implementierung namens *JavaSpaces* [Freeman et al., 1999] speziell auf eine Nutzung in Java ausgelegt, sieht aber in ihrer Spezifikation keine Persistenz vor. Der serverzentrierte Ansatz von *TSpace* verhindert eine generelle Eignung für den Einsatz in mobilen verteilten Szenarien. Auch verteilte und fehlertolerante Ansätze für Linda [Xu und Liskov, 1989] und *TSpaces* [Lehman et al., 2001] berücksichtigen stets nur eine vorübergehende Unterbrechung der Verbindung zum Gesamtsystem, bieten aber keine Offline-Verfügbarkeit für einzelne Knoten.

Auf den Einsatz in einer verteilten Umgebung zielen Bestrebungen, mit *JXTA-Spaces*²⁸ Tuple Spaces auch für das JXTA Peer-to-Peer-Framework zur Verfügung zu stellen. Der aktuelle Stand der Implementierung erlaubt aber noch keine verteilte Speicherung des Tuple Space.

Durch eine Ausrichtung auf ad hoc vernetzte Netzwerknoten ist *Linda in a Mobile Environment (LIME)* [Picco et al., 1999] eine für die mobile Nutzung konzipierte Tuple Space-Implementierung, die auf eine zentrale Speicherung verzichtet. Stattdessen wird der Tuple Space verteilt auf den beteiligten *Hosts* abgelegt. Für die Organisation des verteilten Datenraumes bietet LIME Tuple Spaces und Eventmechanismen auf *Host- und Netzwerkebene* an und erlaubt eine Aufteilung des Tuple Space in mehrere logische Partitionen.

LIME unterstützt neben der *physischen Mobilität* auch eine *logische Mobilität*, die es Programmen erlaubt, zwischen den mobilen Geräten zu migrieren. In diesem Zusammenhang wird in LIME bezüglich der Programme von *Agenten* (engl. *agents*) gesprochen. Jeder Agent hat Zugriff auf einen persistenten *Interface Tuple Space*

²⁸<http://jxtaspaces.jxta.org>

(ITS), der mittels Vernetzung mit anderen ITS einen *flüchtigen gemeinsamen Tuple Space* bildet. Tuple sind im gemeinsamen Tuple Space so lange verfügbar, wie der Knoten mit dem speichernden ITS im Netzwerk verfügbar ist.

In LIME können mehrere ITS auf einem Knoten gleichzeitig existieren. Sie bilden gegeneinander abgeschlossene Tuple Spaces und vernetzen sich jeweils separat zu flüchtigen gemeinsamen Tuple Spaces. Die ITS werden dabei über ihren Namen identifiziert und schließen sich auf einem Knoten zu einem *Host-Level Tuple Space* zusammen. Um zu garantieren, dass ein Agent im Netzwerk ein bestimmtes Tuple erhält, ist es möglich, ein Tuple gezielt in die ITS eines entfernten Agenten zu speichern, an den es bei der nächsten Netzwerkverbindung übergeben wird.

LIME bietet mit seinem Ansatz für räumlich und zeitlich entkoppelte Kommunikation ein wirksames Werkzeug, um die Koordinierung verteilter Systeme im mobilen Nutzungsumfeld zu ermöglichen, ohne dass eine ständige direkte Verbindung zwischen den teilnehmenden Knoten bestehen muss. Der Zugriff auf den gemeinsamen Datenraum über `out()`, `in()` und `read()` Kommandos ermöglicht Entwicklern, sich auf die Funktionalität der verteilten Anwendungen die auf dieses Konzept aufsetzen zu konzentrieren, ohne sich um die Spezifika der Vernetzung der verteilten Programminstanzen kümmern zu müssen.

Die Frage der netzweiten Persistenz von Objekten bleibt bei LIME aber zum Teil ungelöst. Tuple in lokalen ITS sind zwar persistent und für den Knoten jederzeit erreichbar, um aber auf Tuple in entfernten ITS zugreifen zu können, müssen die entsprechenden Knoten im Netzwerk verfügbar sein. Wenngleich die Möglichkeit Tuple direkt in entfernte ITS zu speichern eine gezielte Verteilung der gemeinsam genutzten Daten erlaubt, fehlt hier doch ein durchgängiges Replikationskonzept.

Auch bleibt die Entscheidung, in welche entfernten ITS welches Tuple einzufügen ist, den Entwicklern der Anwendungssoftware überlassen. Weiterhin gilt es für sie, die Vernetzung der beteiligten Knoten zu beachten, um die Objekte der verteilten Umgebung verfügbar zu halten. Auch die Frage nach der Konsistenz von redundant gehaltenen Objekten in einem derartigen System bleibt von den vorgestellten Systemen ungelöst.

4.3.3 Konsistenz in verteilten Persistenzsystemen

Die Verfügbarkeit einer mobil-verteilten Kooperationsumgebung ist stark von der Verfügbarkeit der gemeinsam genutzten Objekte abhängig. In einem mobilen Nutzungszenario können Netzwerktrennungen nicht nur im Fehlerfall, sondern auch auf ausdrücklichen Wunsch der Benutzer auftreten, wenn diese z. B. Batteriereserven sparen wollen. Daher ist eine Replikationsstrategie, die den Bedürfnissen mobiler Benutzer angepasst ist, ein wichtiges Kriterium für den Erfolg einer solchen mobil-verteilten Kooperationsumgebung. Die Replikationsstrategie muss die Verfügbarkeit der Objekte für den Fall berücksichtigen, dass Benutzer offline vom restlichen Kooperationsnetz sind und isoliert an den gemeinsamen Objekten weiterarbeiten möchten (*Offline-Verfügbarkeit*). Daten, die für eine Kooperation benötigt werden, müssen somit auf allen beteiligten Geräten lokal gespeichert sein.

Werden Objekte derart im Netzwerk repliziert und gemeinsam von mehreren Benutzern verändert, müssen sie zudem ständig synchronisiert werden, um nicht von einander abweichende lokale Kopien entstehen zu lassen. Dies hätte eine Inkonsistenz des replizierten Objektes zur Folge. Aufgrund der räumlich und zeitlich entkoppelten Kooperation im mobilen Nutzungsumfeld ist eine zeitnahe Synchronisation oft nicht möglich.

In einem Netzwerk, in dem mehrere Knoten einen Objektspeicher replizieren, entsteht bei einem überlappenden Schreibzugriff auf zwei Repliken ein Konflikt bei deren nächster Synchronisation. Der einfachste Schutz gegen derartige Schreibkonflikte ist es, Schreibzugriffe nur auf einem Knoten (*Master*) zuzulassen und allen anderen replizierenden Knoten (*Slaves*) nur Lesezugriffe zu erlauben. Die Änderungen werden dann vom Master zu den Slaves weitergegeben.

Sollen alle Knoten die replizierten Objekte lokal schreiben dürfen (*Multiple Write Replication*) und dennoch die Konsistenz der Repliken bewahrt werden, müssen die Repliken vor einem Schreibzugriff gesperrt werden. Nach dem Schreiben der Änderung und der Synchronisation der Repliken kann die Sperre wieder aufgehoben werden.

Diese beiden *konservativen Verfahren* erlauben einen schreibenden Zugriff auf die Daten nur, wenn eine Sperre auf (fast) alle betroffenen Knoten gesetzt werden kann. Sie sind daher für mobile Nutzungsszenarien eher ungeeignet.

Besser geeignet für mobile Nutzungsszenarien sind *optimistische Verfahren*, die Schreibvorgänge erlauben, ohne von vornherein Konflikte zu vermeiden (*Optimistic Concurrency Control*, [vgl. Kung und Robinson, 1981]). Die Knoten synchronisieren ihre Repliken nach einem Schreibvorgang, ohne den Verwaltungsaufwand eines Sperrmechanismus betreiben zu müssen. Sollte dabei ein Konflikt bemerkt werden, müssen Mechanismen für die Beseitigung dieses Konflikts bereitstehen. Das Ergebnis ist ein schwach konsistenter Speicher, in dem die ACID-Eigenschaften²⁹ [Gray, 1988] nicht mehr zu jeder Zeit gegeben sind.

Es existieren einige Implementierungen verteilter Speicher für mobile Einsatzfelder. Neben einigen verteilten Datenbankkonzepten [vgl. Holliday et al., 2002] sind die größte Zahl der Lösungen Dateisysteme. Letztere werden aufgrund ihrer Mechanismen zur Aufrechterhaltung oder Wiederherstellung der Konsistenz im Folgenden eingehender betrachtet.

Das *Andrew File System (AFS)* [Campbell, 1998; Howard et al., 1988] liefert über einen Server Dateien an die Clients aus und merkt sich, welche Clients eine Datei geöffnet haben. Bei einer Änderung der Datei durch einen der Clients werden die anderen Clients vom Server informiert und aktualisiert. Dieses Vorgehen hilft, den Netzwerkverkehr beim Zugriff auf entfernte Dateien zu reduzieren und kann auch für eine unverbundene Arbeit genutzt werden [Huston und Honeyman, 1993].

Während in AFS die Dateien von einem einzelnen Server bereitgestellt werden, hat *Ficus* [Popek et al., 1990] das Ziel, Dateisysteme für weiträumige Netzwerke („Na-

²⁹Gewünschte Eigenschaften bei Transaktionen in verteilten Systemen, bestehend aus *Atomicity*, *Consistency*, *Isolation* und *Durability*.

tionwide Networks“) [Popek et al., 1990] zu etablieren und zu dem Zweck gespiegelte Datenträger im Netzwerk zur Verfügung zu stellen. Diese sollen trotz möglicher Verbindungsabbrüche konsistent gehalten werden können.

Coda [Kistler und Satyanarayanan, 1992; Satyanarayanan, 2002] besitzt eine ähnliche Client-Server-Topologie, wurde aber direkt auf ein unverbundenes und mobilitätsfreundliches Arbeiten zugeschnitten [Satyanarayanan et al., 1993]. Zu diesem Zweck werden für die Benutzer interessante Dateien lokal repliziert (*Hoarding*). Im Fall eines Verbindungsabbruchs zu dem Server können die Benutzer auf den lokalen Repliken weiterarbeiten und diese später wieder mit dem Server synchronisieren. Tritt ein Konflikt wegen zeitgleicher Änderungen an einer Datei auf, wird diese gesperrt und die verschiedenen Versionen in einem Verzeichnis mit dem Namen der Datei abgelegt. Dort kann sie manuell von den beteiligten Teilnehmern zusammengeführt werden.

Das *PAYER File System (PFS)* [Dwyer und Bharghavan, 1997] bietet ebenfalls eine mobilitätsfreundliche unverbundene Bearbeitung der lokal replizierten Objekte. Es gestattet den auf die Daten zugreifenden Anwendungen aber zusätzlich, selbst eine Konsistenzstrategie für diese zu bestimmen. Auf diese Art soll die Lösung des in [Fox und Brewer, 1999] beschriebenen Dilemmas eines Trade-offs zwischen starker Konsistenz und hoher Verfügbarkeit der Daten den Entwicklern der bearbeitenden Anwendung überlassen werden.

Diesen Ansatz verfolgen auch *TACT* [Yu und Vahdat, 2002] und *GLOMAR* [Cuce und Zaslavsky, 2003]. Beide unterstützen Anwendungsentwickler darin, eine passende Replikationsstrategie gemäß ihren Anforderungen an die Stärke der Konsistenz der replizierten Anwendungsdaten zu wählen.

Während Systeme wie *Coda* und *PFS* den Ansatz mobilitätsfreundlicher Dateisysteme verfolgen, betrachtet *Bayou* [Terry et al., 1995] direkt die Objektverteilung in Peer-to-Peer-Systemen. In [Edwards et al., 1997] wird dieses Konzept auf asynchron vernetzte Kooperationssysteme übertragen, zielt aber nicht direkt auf mobil-spontane Kooperationslösungen.

Die aufgeführten Lösungsansätze lassen die Vermutung zu, dass ein genereller Speicheransatz wie ein Dateisystem oft nicht in der Lage ist, die bestmögliche Abwägung zwischen Verfügbarkeit und Konsistenz zu bieten. Dieses Problem äußert sich in Systemen, die versuchen, die Entwickler einer Anwendung entscheiden zu lassen, wie hoch die jeweiligen Anforderungen für Verfügbarkeit und Konsistenz der Anwendungsdaten sein sollen. Keiner der Ansätze verfügt über eine Replikationsstrategie, die auf eine mobil-spontane Kooperation in virtuellen Wissensräumen zugeschnitten ist.

4.4 Ereigniskontrolle

Da in kooperativen Umgebungen eine hohe Interaktivität der Benutzer charakteristisch für synchronisierte Arbeitsabläufe ist, müssen Ereignisse innerhalb der gemeinsam bearbeiteten Bereiche so schnell wie möglich in den Wahrnehmungsraum der

Kooperationspartner gebracht werden. Verteilte Systeme müssen sich dazu ständig über Ereignisse in der gemeinsam bereitgestellten Umgebung austauschen können.

Um Ereignisse zwischen Systemen zu kommunizieren, existieren zwei wesentliche Paradigmen: Das *Request&Reply*- bzw. *Pull-Paradigma* und das *Publish/Subscribe*- bzw. *Observer-Paradigma*. Beim ersten ist der Empfänger der Ereignisse für das Abrufen der Zustandsänderungen zuständig. Eine Synchronisation kann hier über regelmäßiges Abfragen der aktuellen Ereignisse in geringen Zeitabständen geschehen (*Polling*). Beim Observer-Paradigma [Gamma et al., 1995, S. 193ff] hingegen meldet sich der Empfänger einmalig beim Sender an und abonniert alle Ereignisse bezüglich der Zustandsänderungen bestimmter Objekte. Tritt ein Ereignis auf, benachrichtigt der Sender direkt alle Abonnenten (*Pushing*).

Das Observer-Paradigma hat den besonderen Vorteil, dass es eine lose Koppelung von Sender und Empfänger erlaubt, da nur eine Kommunikation in Richtung des Empfängers notwendig ist und auch *Multicast-Kommunikation* unterstützt wird. Mehrere Empfänger können so gleichzeitig über Zustandsänderungen informiert werden. Beide Aspekte machen das Observer-Paradigma zu einem idealen Werkzeug für die Ereignis-Behandlung in mobil-verteilten Kooperationssystemen. Im Folgenden wird daher mit *Scribe* eine Implementierung dieses Paradigmas betrachtet, die auf das DHT-basierte Pastry aufsetzt.

Scribe

Scribe [Rowstron et al., 2001; Castro et al., 2002] ist die Implementierung einer Any- und Multicast-Lösung gemäß dem Observer-Paradigma für das Pastry Overlay-Netzwerk (vgl. Abschnitt 4.3.1). Bei Scribe liegen alle teilnehmenden Knoten im logischen Adressraum von Pastry. Die Eigenschaften von Pastry gelten somit auch für Scribe. Als Publisher/Subscriber-System unterstützt es zwei Arten von Teilnehmern. Zum einen gibt es Teilnehmer, die Themen veröffentlichen und zum anderen welche, die veröffentlichte Themen abonnieren, an denen sie interessiert sind. Diese Themen können auch Ereignisse über Statusänderungen von Objekten sein.

Grundsätzlich kann jeder Knoten eines Pastry-Netzwerks mit Hilfe von Scribe Themen veröffentlichen. Alle anderen Knoten können diese Themen abonnieren. Der Herausgeber eines Themas kann nun die Abonnenten über Veränderungen in Kenntnis setzen. Dies wird über Ereignisse realisiert, welche durch Scribe automatisch versandt werden, ohne eine festgelegte Reihenfolge einzuhalten. Das System ist so konzipiert, dass Knoten zugleich Abonnementen als auch Herausgeber sein können. Bestehende Abonnements können in gleicher Weise gekündigt wie bestellt werden. Zu jedem Thema im Netzwerk erstellt Scribe einen eigenen Multicast-Baum, in dem sich alle Abonnenten des Themas wiederfinden.

Da Scribe eine reine Funktionserweiterung von Pastry ist, ist es ebenfalls völlig dezentralisiert aufgebaut. Jeder Knoten kann dabei gleichzeitig mehrere Funktionen einnehmen, wie z. B. Herausgeber, Wurzel eines Multicast-Baums, Knoten innerhalb des Multicast-Baumes und Abonnement.

Jedes Thema wird eindeutig über eine *topicID* identifiziert. Diese wird als Hash-Wert aus dem Namen des Themas und der Identität des Herausgebers errechnet. Diese topicID liegt wieder im logischen Adressraum des Pastry-Netzwerkes. Ein Knoten, dessen nodeID den numerisch kleinsten Abstand zur topicID hat, wird als so genannter *Rendezvous-Knoten* bestimmt. Jedes Thema besitzt genau einen Rendezvous-Knoten, aber ein Knoten kann zugleich der Rendezvous-Knoten mehrerer Themen sein. Dieser Knoten ist dann auch die Wurzel des Multicast-Baumes [Rowstron et al., 2001].

Um ein Thema in einem Scribe-Netzwerk anzulegen, muss Scribe zunächst die topicID des Themas erzeugen und mit Hilfe von Pastry an den zuständigen Knoten übergeben. Dieser Knoten nimmt das neue Thema in seine Themenliste auf. An diesem Punkt kann zusätzlich eine Sicherheitsprüfung durchgeführt werden, welche die Berechtigung des Verfassers dahingend prüft, ob dieser ein Thema anlegen darf. Von nun an ist dieser Knoten der Rendezvous-Knoten des neuen Themas.

Abonenntenverwaltung Scribe erstellt für jedes Thema einen Multicast-Baum, dessen Wurzel der Rendezvous-Knoten bildet. Der Multicast-Baum für das Versenden der Nachricht entsteht aus den Routen der Abonnenten zu dem Rendezvous-Knoten. Die Abonnements werden dabei dezentral verwaltet, damit die Funktionalität auch bei einer hohen Anzahl von Abonnenten sichergestellt werden kann, ohne den Rendezvous-Knoten zu überlasten. Teilnehmer eines Multicast-Baumes müssen nicht selbst Abonennten des Themas sein sondern können auch lediglich Nachrichten weiterleiten. Jeder Knoten besitzt für jedes Thema eine eigene Nachfolgertabelle, in der für die eigenen Nachfolgeknoten im Multicast-Baum jeweils nodeID und IP-Adresse gespeichert sind.

Wenn ein Knoten ein Thema abonnieren möchte, sendet er eine entsprechende Anfrage über Pastry an den zugehörigen Rendezvous-Knoten. Die Anfrage enthält entsprechend als ID für den Nachrichtenschlüssel die des Rendezvous-Knoten. Auf der Route zu dem Rendezvous-Knoten wird auf jedem Zwischenknoten, über den die Nachricht gesandt wird, die Nachfolgertabelle aktualisiert, indem geprüft wird, ob zu diesem Thema bereits eine Nachfolger-Tabelle existiert. Muss die Tabelle neu angelegt werden, wird sie mit der nodeID des übermittelnden Knoten initialisiert. Der übermittelnde Knoten ist somit neuer Nachfolger des aktuellen Knotens im Multicast-Baum. Die ursprüngliche Anfrage des Abonnenten wird verworfen und der aktuelle Knoten sendet seinerseits eine Abonnementanfrage zu dem Thema. Dieser Vorgang wiederholt sich bis ein Knoten gefunden wird, bei dem die Tabelle schon vorhanden ist. Dieser prüft, ob der anfragende Knoten bereits in der Nachfolger-Tabelle enthalten ist. Ist dies der Fall, wird die Nachricht lediglich Richtung Rendezvous-Knoten weitergeleitet, ansonsten wird zusätzlich der Knoten als Nachfolger ergänzt.

Auf diese Weise baut sich der spätere Multicast-Baum von den Blättern (Abonnenten) zur Wurzel (Rendezvous-Knoten) selbstständig auf. An Stellen, wo bereits ein Weg zur Wurzel besteht, endet die Vervollständigung der Route zur Wurzel und der existierende Weg wird ab dieser Stelle für die Verteilung der Nachrichten mit-

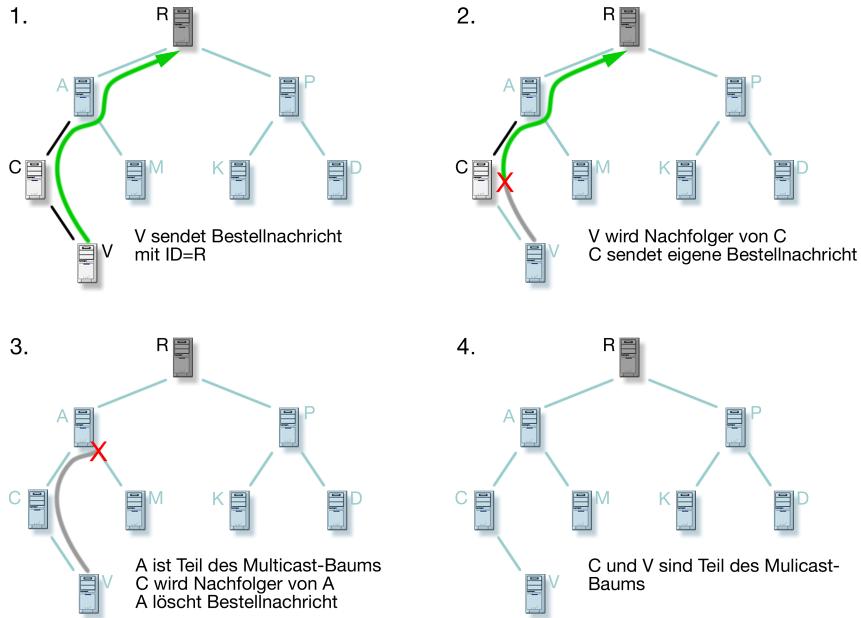


Abbildung 4.11: Abonnieren eines bereits veröffentlichten Themas über einen noch nicht im Multicast-Baum enthaltenen Knoten

benutzt. Betrachtet man die ermittelten Wege nun von der Wurzel zu den Blättern, ergibt sich aufgrund der Pastry-Eigenschaften ein optimaler Pfad. Außerdem wird keine Nachricht im Multicast-Baum doppelt über eine Verbindung zweier Knoten gesandt. In Abbildung 4.11 wird das Abbonieren eines Themas schematisch dargestellt.

Möchte ein Knoten nun ein zuvor abonniertes Thema abbestellen, so kennzeichnet dieser Knoten das Thema als nicht mehr benötigt. Wenn keine Einträge in der Nachfolger-Tabelle vorhanden sind, wird eine Abbestell-Nachricht an den Vorgänger im Multicast-Baum gesandt, der den Knoten daraufhin aus der Nachfolger-Tabelle des Themas löscht. Die Nachricht wandert sukzessiv durch den Multicast-Baum in Richtung Wurzel, bis sie einen Knoten erreicht, dessen Nachfolger-Tabelle nach Löschung des abbestellenden Knotens noch weitere Einträge enthält. So werden auch die Knoten aus dem Multicast-Baum entfernt, die nur als Verteilerknoten für die Weiterleitung der Nachrichten an den abbestellenden Knoten enthalten waren (vgl. Abbildung 4.12).

Das in Scribe enthaltene Verfahren zur Verwaltung der Abonnenten ist auch bei einer hohen Anzahl von Abonnenten sehr leistungsfähig. Die Abonnentenliste wird auf den Knoten der Multicast-Bäume gespeichert und ist durch die zufällige Zuordnungsstrategie gleichmäßig über das Pastry-Netzwerk verteilt. Dies ist eines der wesentlichen Merkmale von Scribe, das die Leistungsfähigkeit des Nachrichtenverands auch bei großen Mengen von Themen und Abonnenten bewahrt.

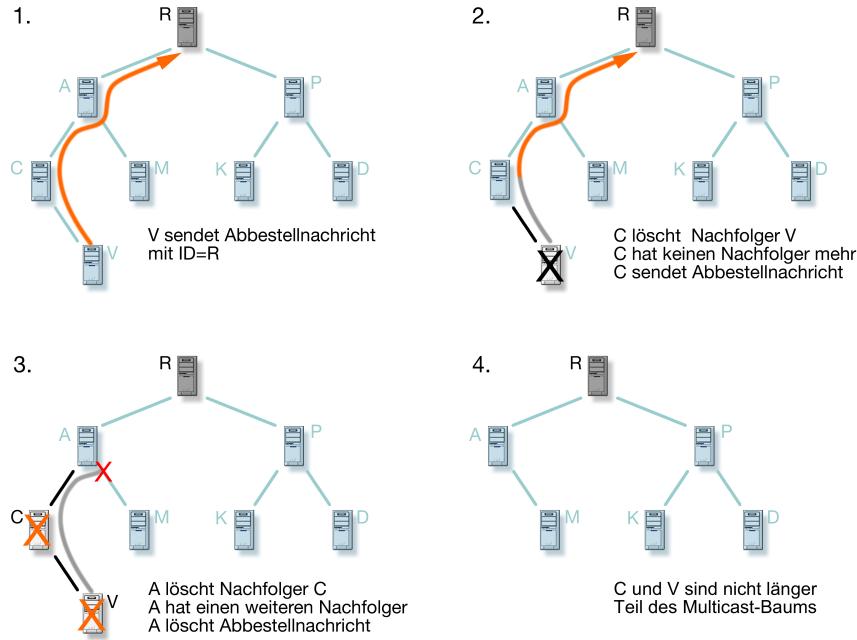


Abbildung 4.12: Abbestellen eines abonnierten Themas: Knoten, die nur Weiterleitungsfunktionen haben, werden wieder aus dem Multicast-Baum entfernt

Die Abonnement-Anfragen werden lokal auf Knoten der bereits vorhandenen Multicast-Bäume bearbeitet und bilden somit eine dezentrale Verwaltungsstruktur. Zusätzlich wird eine Erweiterung des Multicast-Baums um einen neuen Knoten lokal vorgenommen, was die Notwendigkeit einer zentralen Bearbeitung der Anfragen vermeidet. Der durch das Pastry-Routing optimale Multicast-Baum führt zu einer Nachrichtenweitergabe, die Mehrfachversendungen und die damit einhergehende erhöhte Netzwerkkommunikation vermeidet. Abbildung 4.13 verdeutlicht den Unterschied zwischen einem Nachrichtenversand in Scribe (Single-Mulicast) und dem herkömmlichen Multiple-Unicast. Grundsätzlich lässt sich bei einem Vergleich der beiden Ansätze folgendes feststellen:

- *Single-Multicast:* Bei dieser Variante des Nachrichten-Versandes versendet der Rendezvous-Knoten Nachrichten nur an seine direkten Nachfolger im Multicast-Baum. Dies löst auf diesen Knoten wiederum einen Versand zu deren Nachfolgern aus. Diese Kette setzt sich solange fort, bis alle Knoten im Multicast-Baum erreicht worden sind. Die gesendete Nachricht durchläuft dabei den Multicast-Baum und wird lediglich einmal über jede Kante zwischen den Knoten übertragen. Vorteilhaft ist auch, dass der Rendezvous-Knoten nur wenige Knoten für den Versand zu einem Thema direkt kennen muss.
- *Multiple-Unicast:* Der Rendezvous-Knoten muss bei diesem Verfahren alle Zielknoten selbst kennen. Er versendet eine Nachricht der Reihe nach an alle Ziel-

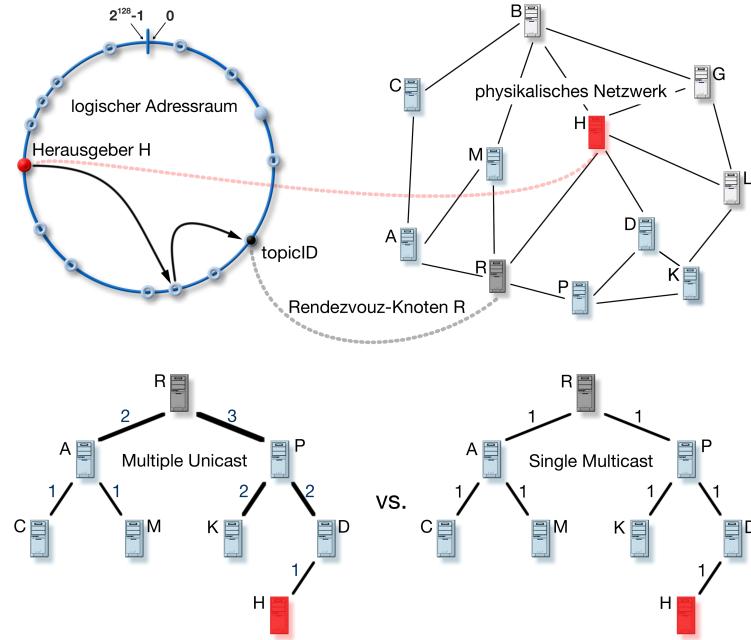


Abbildung 4.13: Gegenüberstellung des in Scribe verwendeten performanten Single-Multicast-Verfahrens und des üblichen Multiple-Unicast-Verfahrens

knoten. Die Nachricht wird somit mehrfach über die gleichen Nachrichtenwege versandt. Dieses Verfahren benötigt einen performanten Sender und eine Netzwerktopologie mit einer entsprechend hohen Sende-Bandbreite.

Durch die positive Auswirkung des Pastry-Routings auf die Entstehung der Multicast-Bäume werden sowohl die physikalischen Verbindungseigenschaften im realen Netzwerk als auch kurze Routen über wenige Knoten berücksichtigt.

Verbreitung von Nachrichten Soll eine Nachricht an die interessierten Knoten versendet werden, so hat der Herausgeber der Nachricht zwei Möglichkeiten, den Rendezvous-Knoten zu erreichen. Bei der ersten Variante kennt der Herausgeber bereits die IP-Adresse des Rendezvous-Knoten und nimmt direkt Kontakt mit diesem auf.

Ist die IP-Adresse des Rendezvous-Knotens dem Herausgeber jedoch nicht bekannt, kann die Nachricht über Pastry versandt werden, indem sie mit der *topicID* als Nachrichtenschlüssel an den Rendezvous-Knoten übermittelt wird. Der Rendezvous-Knoten teilt in diesem Fall seine IP-Adresse aufgrund einer in der Nachricht enthaltenen Anforderung dem Herausgeber mit. Über die IP-Adresse kann dieser von nun an seine Nachrichten direkt zum Rendezvous-Knoten senden.

Vom Rendezvous-Knoten aus wird die Nachricht im Multicast-Baum an die Abonnenten verteilt. Sollte ein Rendezvous-Knoten durch einen neu ins Pastry-Netzwerk

hinzugefügten Knoten ersetzt werden, leitet er die Nachrichten zunächst an den neuen Knoten weiter und teilt dem Herausgeber der Nachricht die neue IP-Adresse mit. Auch die Variante des Sendens einer Nachricht mit der topicID als Nachrichtenschlüssel führt zu der Behebung des Defekts durch eine solche Veränderung. Da der Versand stets über einen Rendezvous-Knoten läuft, können an dieser Stelle auch die Berechtigungen für den Nachrichtenversand verwaltet werden und Angriffe auf das Nachrichtensystem abgeblockt werden.

Zuverlässigkeit und Ausfallsicherheit Um eine Zuverlässigkeit und Ausfallsicherheit von Scribe zu gewährleisten, müssen die Rahmenbedingungen relativ stabil sein, da Scribe eine Zustellung der Nachrichten nicht garantiert. Besonders bei der Verteilung von Ereignissen in verteilten Systemen muss aber eine gewisse Verlässlichkeit der Zustellung gewahrt bleiben. Scribe ist beim Nachrichtenversand zunächst einmal auf eine möglichst hohe Effizienz ausgerichtet. Durch eine Erweiterung, der in Scribe genutzten Kommunikationsmechanismen kann die Zuverlässigkeit allerdings erhöht werden.

Scribe benutzt für eine verlässliche Nachrichtenzustellung das verbindungsorientierte TCP-Protokoll für die Kommunikation zwischen den Knoten des Multicast-Baumes [Rowstron et al., 2001]. Um die Zuverlässigkeit dieses Verfahrens weiter zu erhöhen, wird für jedes Ereignis eine eindeutige Nummer vergeben. Die nummerierte Nachricht verbleibt über eine fest definierte Zeit auf den Knoten im Multicast-Baum. Sollte die Zustellung einer Nachricht fehlgeschlagen, kann so der Versand der Nachricht wiederholt werden.

Durch diese temporäre Zwischenspeicherung der Nachrichten und ihre Nummerierung können sowohl bei Ausfällen des Rendezvous-Knoten, als auch bei Ausfällen von Knoten im Multicast-Baum, Fehler kompensiert werden. Dies geschieht durch Vergleichen der aktuellen und der gespeicherten Ereignis-Nummer in den Knoten. Wurde ein Ereignis noch nicht empfangen, wird es weitergeleitet und ansonsten verworfen. So wird eine höhere Zuverlässigkeit bei der Nachrichtenzustellung erreicht.

Um die Funktionsfähigkeit des Multicast-Baumes zu jedem Zeitpunkt zu gewährleisten, stehen Anwendungen in Scribe verschiedene Mechanismen zur Verfügung, um eine Reparatur des Multicast-Baumes im Fehlerfall zu bewerkstelligen. Um den Ausfall eines Knotens im Multicast-Baum frühzeitig zu erkennen, senden die Multicast-Knoten Kontrollnachrichten zu ihren Nachfolgern. Den Nachfolgern ist so der Status ihrer Vorgänger im Multicast-Baum bekannt. Zur Verringerung des Kommunikationsaufkommens werden auch die normalen Nachrichten als Lebenszeichen genutzt.

Bleiben die Lebenszeichen eines Knotens für eine bestimmte Zeit aus, wird eine Prozedur zur Korrektur des Defekts gestartet. Der Knoten, der den Defekt entdeckt hat, sendet dazu über Pastry eine Abonnement-Anfrage mit der topicID als Nachrichtenschlüssel. Pastry findet aufgrund seiner Fehlerresistenz einen neuen Weg in dem Multicast-Baum und umgeht den vermeintlich ausgefallenen Knoten. Über diese neue Wegsuche werden eventuell auch neue Knoten zum Multicast-Baum hinzugefügt.

Mittels dieser Methode kann Scribe auch den Ausfall eines Rendezvous-Knoten im Multicast-Baum selbstständig beheben. Dazu werden im Betrieb die für den Nachrichtenversand benötigten Informationen des Rendezvous-Knotens analog zu dem Replikationsverfahren in PAST (vgl. Abschnitt 4.3.1) auf k Knoten des *Leaf Sets* repliziert. Da diese k Knoten aus dem *Leaf Set* des Rendezvous-Knoten stammen, sind sie die direkten Nachbarn innerhalb des logischen Adressraumes.

Sollte ein Rendezvous-Knoten ausfallen, wird dies von seinen Nachfolgern erkannt und über Pastry eine neue Abonnement-Anfrage gesandt. Die Nachricht wird zu dem Knoten gesandt, der numerisch am nächsten zur topicID liegt. Dies war zuvor der ausgefallene Rendezvous-Knoten. Da dieser nicht mehr erreichbar ist, wird die Nachricht zu der nächstgelegenen nodeID im logischen Adressraum geleitet. Der zugehörige Knoten stammt aufgrund der Eigenschaften des Pastry-Routings mit hoher Wahrscheinlichkeit aus dem replizierenden Leaf Set des alten Rendezvous-Knotens. Da auf dem Nachbarknoten eine Kopie der für den Rendezvous-Knoten benötigten Informationen hinterlegt ist, kann dieser nun die Rolle des Rendezvous-Knotens nahtlos übernehmen.

Die Herausgeber bemerken den Ausfall des alten Rendezvous-Knoten, da dieser nicht mehr direkt über seine IP-Adresse zu erreichen ist, und lassen über Pastry eine neue Route berechnen. So gelangen sie ebenfalls zu dem neuen Rendezvous-Knoten.

Die Einträge in den Nachfolger-Tabellen werden solange aufrechterhalten, wie regelmäßig Nachrichten von den Nachfolgern das bestehende Interesse an dem abonnierten Thema bestätigen, sonst werden sie automatisch entfernt, um den Multicast-Baum nicht degenerieren zu lassen. Die Fehlerbehandlung ist unabhängig von der Größe des Multicast-Baumes, da diese Nachrichten nur an eine kleine Auswahl von Knoten gesandt werden müssen. Von einer Wiederherstellung eines defekten Multicast-Baumes ist nur das direkte Umfeld des Fehlers betroffen. Somit können Fehler an verschiedenen Stellen gleichzeitig behoben werden.

Analog zu dem hier vorgestellten Verfahren eines ereignisbasierten Nachrichtensystems kann mit Hilfe von Scribe auch ein Gruppenverwaltung realisiert werden. Dazu werden Gruppen wie Themen betrachtet, in denen nicht Abonennten, sondern Gruppenmitglieder enthalten sind. Das Addressierungssystem für die Gruppenkommunikation ist ähnlich dem des Nachrichtenversands, wobei die topicID durch die groupID ersetzt wird. Der Rendezvous-Knoten wird wiederum durch einen Knoten vertreten, dessen nodeID numerisch am nächsten zur groupID ist. Alle Funktionen, wie das Hinzufügen oder Entfernen von Mitgliedern in den Gruppen, werden analog zum Abonnieren und Abbestellen von Themen bewerkstelligt [Castro et al., 2002].

Bewertung von DHTs in mobil-verteilten Anwendungsszenarien

Scribe gibt den Entwicklern von verteilten Anwendungen ein mächtiges Werkzeug zur Multicast-Kommunikation und insbesondere zur Gruppenkommunikation in die Hand. Zusammen mit der Möglichkeit der Bereitstellung einer selbstorganisierten Netzwerkstruktur (Pastry) inklusive eines replizierten verteilten Speichers (PAST)

erscheinen die DHT-basierten Lösungsansätze als eine ideale Plattform für die Implementierung einer verteilten Kooperationsumgebung.

Für eine mobilitätsfreundliche Infrastruktur fehlt noch die Unterstützung der *Offline-Verfügbarkeit* verteilt gespeicherter Daten. Unverbundene Knoten werden in allen Lösungsansätzen stets als defekt betrachtet und es existiert zudem keine Möglichkeit, Kooperationsdaten gezielt auf eine bestimmte Knotengruppe zu replizieren. Auch werden Nachrichten (mit Ausnahme von I3) stets nur synchron versandt, so dass unverbundene Knoten auch beim Wiedereintritt von einem Teil der Ereigniskontrolle abgeschnitten bleiben.

Trotz dieser Defizite derzeitiger DHT-Implementierungen bleiben viele der angebotenen Lösungsansätze vielversprechend für die Umsetzung einer mobil-verteilten Kooperationsumgebung. In dem folgenden Kapitel 5 sollen daher die fehlenden Bausteine ergänzt werden und so zu einem Entwurfsmuster für mobil-verteilte Kooperationsumgebungen basierend auf dem Konzept der virtuellen Wissensräume kombiniert werden.

5 Musterarchitektur mobil-verteilter Wissensräume

Bei Betrachtung der Szenarien mobil-verteilter Kooperation und der verfügbaren Lösungsansätze ergibt sich ein zwiespältiges Bild. So scheinen für viele der in Kapitel 3 ermittelten Anforderungen bereits geeignete Lösungen zu existieren, während zur Erfüllung anderer kritischer Anforderungen keiner der betrachteten Lösungsansätze geeignet scheint.

Die Defizite zeichnen sich insbesondere mit Blick auf eine mobilitätsfreundliche verteilte Persistenz ab. Die zentrale Forderung nach *Offline-Verfügbarkeit der Kooperationsdaten* und *zeitlich entkoppelter Ereigniskommunikation*, die zusammengekommen ein Fortsetzen der Arbeit in unverbundenen Kooperationssituationen erlauben würden, wird von keinem der betrachteten Ansätze erfüllt. Die Schaffung einer geeigneten Persistenzschicht wird so zu einer zentralen Forschungsfrage für die Unterstützung mobil-spontaner Kooperationsszenarien.

Auf der anderen Seite existieren mit den vorhandenen Lösungsansätzen einige wichtige Bausteine für die zukünftigen mobil-verteilten Kooperationsumgebungen. So erlauben die Protokolle für mobil-spontane Vernetzung das Verbinden von Mobilgeräten und Diensten sowie das Einbinden existierender Netzwerkinfrastrukturen. Mittels dezentraler Persistenzsysteme auf DHT-Basis lässt sich ein verteilter globaler Speicher für die Kooperationsumgebungen realisieren, es fehlt aber an Mechanismen für eine gesteuerte Replikation und einer Synchronisation der Repliken. Mit anonymen und zeitlich entkoppelten Speicherkonzepten in Form von Tuple Spaces besteht zudem die Möglichkeit, die technischen Strukturen einer mobil-verteilten Persistenz in eine konsistente Metapher für den Zugriff auf einen gemeinsamen und strukturierten Speicher einzubinden.

In den folgenden Abschnitte werden die vorhandenen Komponenten um wichtige Bausteine zu einer Gesamtarchitektur ergänzt. Diese enthält u. a. eine Persistenzschicht für die Bereitstellung mobil-verteilter Wissensräume in einer spontan vernetzten Kooperationsumgebung. Zu diesem Zweck wird eine speziell auf die Bedürfnisse mobil-spontaner Kooperation zugeschnittene Replikations- und Versionsierungsstrategie und ein für die Gruppenarbeit angepasstes Tuple Space-Konzept vorgestellt. Zudem wird die Einbettung externer Kontext- und Persistenzdienste in diese Strukturen behandelt. Die so gewonnenen Bausteine und Konzepte ergänzen sich schließlich zu einem neuartigen Entwurfsmuster einer Architektur zur Bereitstellung mobil-verteilter Wissensräume.

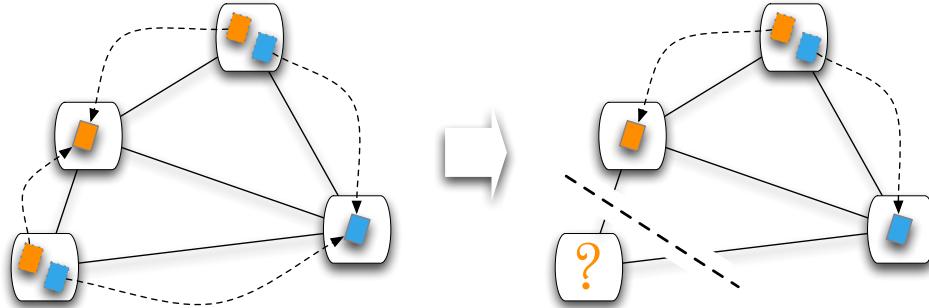


Abbildung 5.1: Offline-Verfügbarkeit in einer verteilten Persistenzschicht

5.1 Replikationsstrategien für die mobil-spontane Kooperation

Der Zugriff auf einen über ein Ad-Hoc-Netzwerk errichteten verteilten gemeinsamen Speicher ist in mobil-spontanen Nutzungsszenarien von Verfügbarkeit einer Netzwerkverbindung zum speichernden Knoten abhängig¹. Wenn die in so einem mobilen Netzwerk verteilt gespeicherten Objekte nicht in intelligenter Weise über die beteiligten Knoten des Netzwerkes repliziert werden, sind sie wie in Abbildung 5.1 dargestellt bei einem entfernten Zugriff eventuell nicht verfügbar [vgl. Eßmann et al., 2006].

Ein gängiges Konzept zur Erhöhung der Zuverlässigkeit von Diensten in einer unzuverlässigen Umgebung ist *Redundanz* (vgl. Kapitel 3). Diese führt aber gleichzeitig zu einem erhöhten Verwaltungsaufwand und Speicherbedarf. Basiert der Dienst zudem auf einem gemeinsamen und veränderlichen Datenbestand, müssen die Kopien der redundant gespeicherten Daten stets mit dem Original synchronisiert werden. Erfolgt die Bearbeitung der replizierten Objekte auf allen redundant gehaltenen Kopien, muss die Synchronisation in jede Richtung erfolgen. Zusätzlich können in diesem Fall Konflikte bei einer gleichzeitigen Bearbeitung eines replizierten Objektes auf zwei unterschiedlichen Knoten auftreten.

Eine Erreichbarkeit aller Repliken eines Objektes zum Zeitpunkt einer Änderung und der nachfolgenden Synchronisation kann in mobilen Netzwerken nicht garantiert werden. Daher wird die Synchronisation eventuell zeitversetzt und *epidemisch* zwischen den Knoten erfolgen. Epidemisch bedeutet in diesem Zusammenhang, dass sich die Knoten untereinander synchronisieren, sobald sie in Reichweite kommen. Alle synchronisierten Knoten gleichen ihrerseits die Änderungen mit weiteren nicht aktualisierten Knoten ab. Diese Strategie verhindert, dass der ändernde Knoten sich mit allen replizierenden Knoten direkt synchronisieren muss und dementsprechend Bandbreite benötigt². Obwohl das epidemische Verfahren eine dynamischere Struk-

¹[Satyanarayanan, 1996] nennt als wesentliche Probleme mobiler Netzwerke deren hohe Schwankungen in ihrer Leistungsfähigkeit und Verfügbarkeit.

tur besitzt, ähnelt es stark dem Versand von Nachrichten im Multicast-Baum von Scribe (vgl. Abschnitt 4.4).

Mögliche Konflikte durch zeitgleiche Änderungen innerhalb der verteilten Umgebung erkennt die Synchronisation über einen *Concurrency Control-Mechanismus* und macht sie daraufhin kenntlich. Eine Sperrung der replizierten Objekte während der Bearbeitung würde Konflikte verhindern, kommt aber schon deshalb nicht in Frage, weil die Sperre unverbundene Knoten nicht erreichen würde und wie die Änderung ebenfalls epidemisch übertragen werden müsste. Die einzige mögliche Vorgehensweise in sporadisch verbundenen Systemen ist ein so genanntes *optimistisches Sperren* (engl. *optimistic locking*) der replizierten Objekte³. Dies bedeutet, dass ein Objekt erst gesperrt wird, wenn ein Konflikt aufgetreten ist. Um die Sperre wieder aufzuheben, muss infolgedessen der Konflikt zuerst behoben werden. Diese Strategie basiert auf der Annahme, dass Konflikte die Ausnahme bleiben und selten auftreten.

Mit größeren zeitlichen Abständen zwischen den Synchronisationen der replizierten Objekte erhöht sich gleichzeitig die Wahrscheinlichkeit eines Konflikts. Daher wird eine Synchronisation im optimalen Fall sofort nach der Änderung eines replizierten Objektes erfolgen. Ist die Synchronisation aufgrund fehlender Verbindung zu den replizierenden Knoten nicht möglich, erfolgt sie spätestens bei dem nächsten Verbindungsauflaufbau. Auch hier hilft eine epidemische Synchronisation im verteilten Netzwerk die Abstände zwischen den Aktualisierungen zu verkürzen.

Die im Folgenden vorgestellte Replikationsstrategie ist insbesondere für die mobil-verteilte Kooperation in virtuellen Wissensräumen ausgelegt, kann aber auf alle kooperativen Operationen in einem gemeinsamen Datenraum übertragen werden. Die Replikationsstrategie ist in der Lage, die replizierenden Knoten selbstständig zu wählen und zugleich die Offline-Fähigkeit der gemeinsam bearbeiteten Objekte für jeden Kooperationspartner zu gewährleisten. Zusätzlich wird ein Mechanismus bereitgestellt, um Konflikte in der Synchronisation aufzudecken und zu beheben.

Um nicht beliebige Daten auf alle erreichbaren Knoten zu replizieren, stellt die Replikationsstrategie Mechanismen bereit, die ermitteln, welche Daten auf welchen Knoten repliziert werden sollen. Als *Strukturierungsmittel für die Replikation* bietet sich bei dem Konzept der virtuellen Wissensräume die Kooperationsgruppe an. Da im virtuellen Wissensraum jeder Gruppe genau ein gemeinsamer Arbeitsbereich zugeordnet ist, wird dieser mit allen enthaltenen Objekten zwischen den Gruppenmitgliedern repliziert. Im Idealfall hat so jedes Gruppenmitglied alle die Kooperation betreffenden Daten auch im Fall einer gestörten Verbindung verfügbar.

Sollen auch mobile Geräte mit wenig lokalem Speicherplatz in die Kooperation integriert werden können, kann die Replikationsstrategie in mehreren Stufen abgeschwächt werden. Dazu werden die teilnehmenden Knoten in drei Klassen unterteilt. Die erste Klasse sind die *Independent Nodes*, die alle Objekte der Kooperationsgruppe replizieren. Die zweite Klasse der *Selective Nodes* kennt den Inhalt des gemeinsamen Arbeitsbereiches und wählt die Objekte für die lokale Replikation gezielt aus.

²Vgl. [Terry et al., 1995]

³Vgl. [Kung und Robinson, 1981]

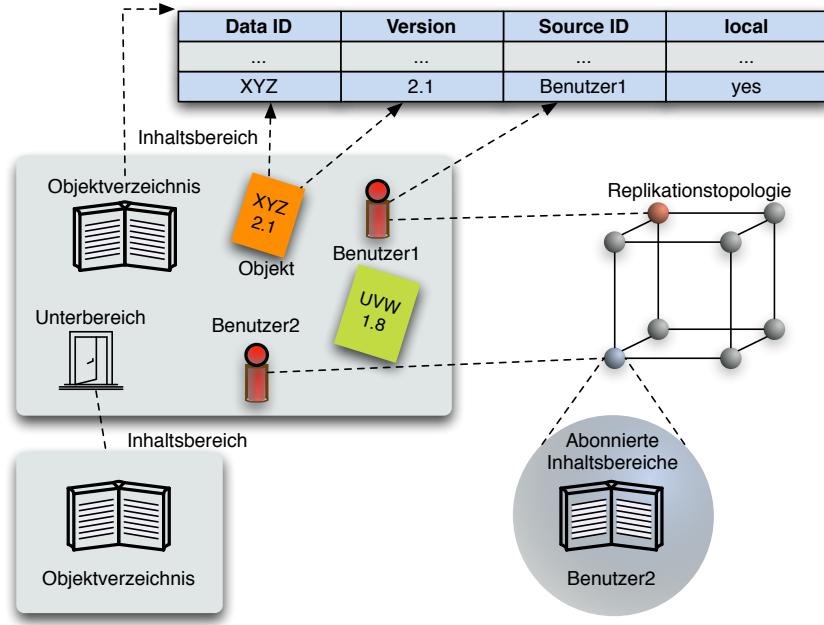


Abbildung 5.2: Schematische Übersicht des Versionierungsmechanismus

Die dritte Klasse repräsentiert Geräte, die über quasi keinen lokalen Speicher verfügen und die Objekte erst bei Bedarf über das Netzwerk auf einem anderen Knoten bearbeiten. Diese Klasse nennt sich *On Demand Nodes*.

Geräte der Klasse der *On Demand Nodes* sind nicht in der Lage ohne eine Netzwerkverbindung zu mindestens einem Gruppenknoten der Klasse der *Independent Nodes*, an der Kooperation teilzunehmen. Tatsächlich ist es möglich, mit einem *Independent Node* und beliebig vielen *On Demand Nodes*, eine Art Client-Server-System nachzubilden. Diese Konfiguration mit einem einzigen *Independent Node* ist gleichzeitig die Untergrenze für eine kooperationsfähige Infrastruktur mit der hier vorgestellten Replikationsstrategie. Eine Aggregation des Datenraums aus den nur zum Teil replizierten Daten auf den *Selective Nodes* ist nicht vorgesehen. *Selective Nodes* können lediglich auf beiden Seiten vorhandene Repliken untereinander synchronisieren.

Im Folgenden wird die technische Umsetzung der Replikationsstrategie auf Basis eines *Overlay-Netzwerkes* wie den *Distributed Hash Tables (DHTs)* aus Abschnitt 4.3.1 diskutiert. Das Overlay-Netzwerk zeichnet sich für die Zustellung der Objekte und deren Aktualisierungen an die einzelnen Knoten verantwortlich. Die Replikation bündelt die beteiligten Knoten in Replikationsgruppen, sendet die Aktualisierungen an deren Mitglieder und verwaltet die benötigten Daten für die genutzte Strategie. Die Konfliktaufdeckung und Behebung lehnt sich an den in Abschnitt 4.3.3 präsentierten Verfahren aus *Bayou* und *Coda* an. Sie geschieht im Wesentlichen über eine verteilte Versionierung der Objekte.

Repliziert werden in der Regel komplett Inhaltsbereiche mit den enthaltenen gemeinsamen Objekten. Um Konflikte mit geringen Datenaufkommen frühzeitig erkennen zu können, existiert zu jedem Inhaltsbereich ein Verzeichnis der enthaltenen Objekte. In dem Verzeichnis ist für jedes im Inhaltsbereich enthaltene Objekt die eindeutige Identifikationsnummer, die Version des Objektes und die Adresse des Knotens, der die Version erstellt hat, gespeichert. Außerdem befindet sich dort ein Vermerk, ob das Objekt schon lokal vorhanden ist oder noch empfangen werden muss. Das Verzeichnis ist ein Objekt des Inhaltsbereiches wie jedes andere auch und wird somit über die gleichen Mechanismen verteilt und synchronisiert. Einen Überblick über diese Struktur gibt Abbildung 5.2.

Zusätzlich verwaltet jeder Knoten ein lokales Verzeichnis über die abonnierten Inhaltsbereiche inklusive deren lokalen Versionsnummern. Wird ein Knoten aktiv und meldet sich an dem Persistenznetzwerk an, kann er anhand dieses Verzeichnisses vergleichen, welche Inhaltsbereiche noch aktuell sind und welche er aufgrund einer neueren Version im Netz abgleichen muss. Stimmt die lokale Version mit der entfernten Version eines replizierten Inhaltsbereiches überein, ist dieser auf dem aktuellen Stand. Bei abweichender Version liegt eine Diskrepanz im betroffenen Inhaltsbereich vor und der Knoten führt einen Abgleich durch. Dabei kann er anhand der verfügbaren Verzeichnisinformationen entscheiden, wie er mit den *lokalen Repliken* der Inhaltsbereiche verfahren muss:

- *Neue lokale Objekte* werden in das Netzwerk verteilt. Dazu werden die Daten über das Overlay-Netzwerk epidemisch mit den replizierenden Nachbarknoten abgeglichen.
- *Aktuelle Objekte* können ignoriert werden, da ihre Version mit der im Netzwerk übereinstimmt.
- *Veraltete Objekte* werden aus dem Netzwerk aktualisiert. Dazu kontaktiert der Knoten die replizierenden Nachbarknoten und bittet um das Objekt. Diese senden das Objekt über das Overlay-Netzwerk an den anfragenden Knoten. Sollte das Objekt noch nicht auf den replizierenden Nachbarknoten verfügbar sein, ist das Objekt noch sehr neu und befindet sich in der epidemischen Verteilung. Es wird den Knoten über den Verteilungsprozess automatisch erreichen.
- *Objekte mit Versionskonflikt*, bei denen sowohl die lokale als auch die im Netzwerk verfügbare Version eines Objektes aktueller sind als zum Zeitpunkt der letzten Synchronisation, werden gesperrt und als inkonsistent markiert, bis der Konflikt behoben ist. Um kaskadierende Konflikte zu vermeiden, werden außerdem anhand der Verzeichniseinträge die replizierenden Knoten ermittelt und über ein Ereignis über den Konflikt in Kenntnis gesetzt. Nun ist es die Aufgabe der Anwendungsschicht oder der Benutzer selbst, den Konflikt zu beheben. Ist der Konflikt gelöst, wird das konfliktfreie Objekt als neue Version über das Netzwerk verbreitet und wieder freigegeben.

Mittels dieser Replikationsstrategie können alle Objekte so aktuell als nur möglich gehalten werden. Im verbundenen Zustand werden die Knoten automatisch mit Aktualisierungen versorgt (*Pushing*). Nach einer Verbindungsunterbrechung erfolgt der Abgleich direkt bei der Anmeldung an die verteilte Persistenz durch den Knoten selbst (*Polling*). Der Abgleich der Änderungen mit der verteilten Persistenz wirft die Frage nach der Vorgehensweise bei der Distribution der Aktualisierungen an die Mitgliedsknoten der Replikationsgruppe auf. Als Replikationsgruppe wird diesbezüglich die Gruppe aller teilnehmenden *Independent Nodes* der Kooperationsgruppe verstanden.

Eine mögliche Distributionsstrategie ist ein sternförmiger Punkt-zu-Punkt-Abgleich mit allen replizierenden Knoten im Netz. In diesem Fall wären bei n replizierenden Knoten alle n Knoten in einem Schritt aktualisiert. Dies würde bedeuten, dass sich der ändernde Knoten mit $n - 1$ Nachbarn abgleichen muss. Dies könnte z. B. über ein DHT oder ein anderes Adressierungsverfahren geschehen. Betrachtet man die Verteilungsstrategie als einen Graphen, kommt dies einem Knotengrad⁴ von $n - 1$ gleich. Die Änderung muss über jede Kante einmal verschickt werden. Dies kann schnell zu einem Flaschenhals am ändernden Knoten führen, da alle Änderungen über eine einzige Netzwerkverbindung gesandt werden. Auch wenn der Knoten über mehrere Netzwerkschnittstellen verfügen würde, ist die Bandbreite, die dem Knoten zur Verfügung steht, begrenzt. Da sich die für die Übertragung einer Änderung benötigte *Gesamtbandbreite* aus der Formel „*Größe der Änderung * Knotengrad*“ ergibt, und die Größe der Änderung nicht beeinflussbar ist, ergibt sich die einzige Optimierungsmöglichkeit im Bereich des Knotengrades.

Der Knotengrad lässt sich verringern, indem die Änderungen in einer Reihentopologie von Knoten zu Knoten weitergereicht werden. Dies hat allerdings zur Folge, dass die Änderung den letzten der n Knoten erst nach $n - 1$ Schritten erreicht. Ziel ist aber eine Übertragung der Änderungen an alle Knoten mit möglichst wenig Bandbreitenverbrauch je Knoten und in möglichst wenig Schritten. Gesucht wird also eine Distributionsstrategie, die einen geringen Knotengrad hat und deren Graph einen geringen Durchmesser besitzt.

So eignen sich die in Abschnitt 4.4 vorgestellten Verfahren mittels *Multicast-Bäumen*, wie sie z. B. in *Scribe* zu finden sind, durch ihre Baumstruktur gut für eine epidemische Verbreitung von Objekten im Graphen. Die Höhe des Knotengrads im (balancierten) Baum bestimmt dabei automatisch die Entfernung zum letzten Knoten. Allerdings muss der Distributionsbaum bei einer hohen Knotendynamik, wie sie in mobilen Nutzungszenarien vorkommt, häufig reorganisiert werden, um nicht zu degenerieren. Des Weiteren ist ein solcher Graph nicht symmetrisch, da alle Nachrichten zunächst an den Wurzelknoten gesandt werden müssen, der diese dann nach unten weiterverteilt. Somit können Knoten in der Nähe des Wurzelknotens ihre Nachrichten schneller verbreiten als die Knoten in tieferen Schichten des Distributionsbaums.

⁴Knotengrad ist die Anzahl der Kanten, die einen Knoten verlassen.

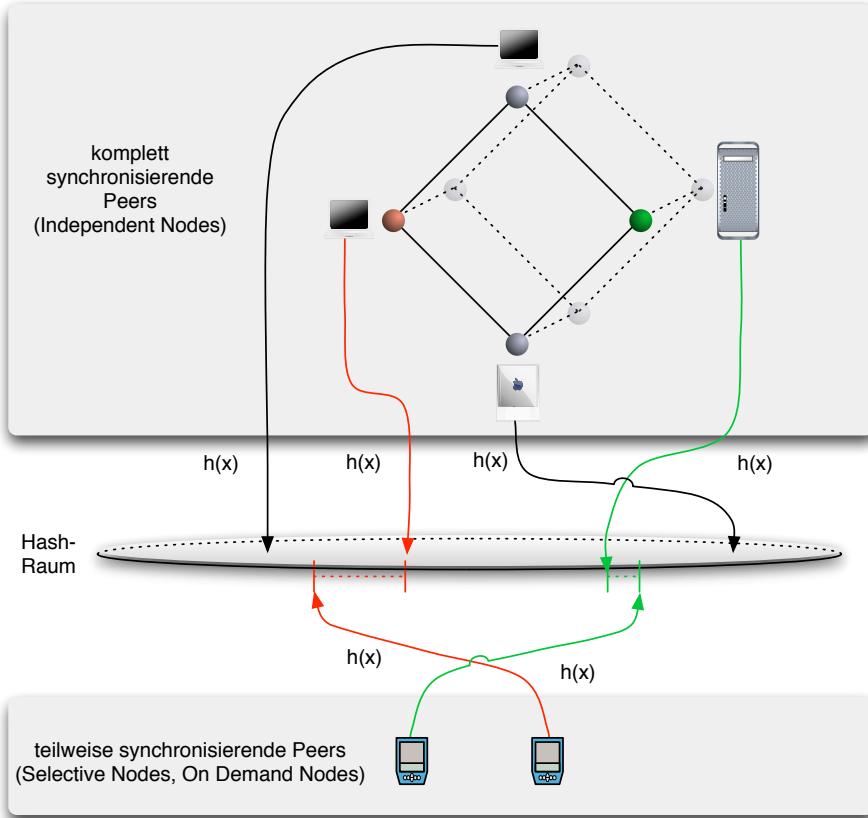


Abbildung 5.3: Ein $HC_{(3)}$ als Distributions- und Replikationstopologie, bestehend aus *Independent Nodes* mit den über den Hashraum verbundenen *Selective/On Demand Nodes*

Die Symmetrieeigenschaft, von allen Knoten aus alle anderen gleich gut erreichen zu können, ist im *Hypercube* gegeben. Hypercubes sind für einen guten Daten durchsatz und eine geringe Ausdehnung bekannt: Ein Hypercube mir d Dimensionen ($HC_{(d)}$) verbindet $n = 2^d$ Knoten bei einer Ausdehnung von lediglich $\log(n) = d$ und einem Knotengrad der ebenfalls bei $\log(n)$ liegt. Dies macht den Hypercube zu einer interessanten Topologie für die Verteilung der Aktualisierungen innerhalb der replizierenden Gruppe.

In dem hier vorgestellten Verfahren wird bei der Änderung eines replizierten Objektes auf einem Knoten die Aktualisierung an die direkten Nachbarknoten im Hypercube gesandt⁵. Diese verteilen die Aktualisierungen wiederum an ihre Nachbarn

⁵Nutzt man eine Hypercube-Topologie für die Synchronisation der Objekte in der replizierenden Gruppe, ist jeder Mitgliedsknoten ein Eckpunkt des Hypercubes. In diesem Hypercube kann man die Daten nun z. B. per *hypercube multicast* [Jang, 1990] oder epidemisch [vgl. Terry et al., 1995] verteilen.

bis die Aktualisierung alle Knoten erreicht hat. Dies geschieht aufgrund der Eigenschaften des Hypercubes in $\log(n)$ Schritten. Die Adressierung der Nachbarknoten im Netzwerk geschieht dabei wie gewohnt über das Overlay-Netzwerk. Jeder Knoten muss daher lediglich die Adressen seiner Nachbarknoten in dem Hypercube kennen.

In dem Hypercube sind alle *Independent Nodes* der Kooperationsgruppe enthalten. Die *Selective Nodes* und die *On Demand Nodes* werden mit den *Independent Nodes* assoziiert. Mit diesen gleichen sie sich Punkt-zu-Punkt ab. Der *Independent Node* verbreitet die Änderungen von seinen assoziierten *On Demand* und *Selective Nodes* innerhalb der Replikationsgruppe. Die Zuordnung der *Selective Nodes* und der *On Demand Nodes* zu den *Independent Nodes* erfolgt über deren Abstand zueinander.

Beim einem *konsistenten Hashverfahren*⁶ wird der *Independent Node* mit der kleinsten Differenz zum Hashwert des *Selective Nodes* oder *On Demand Nodes* gewählt. Bei *Pastry* (vgl. Abschnitt 4.3.1) könnte auch das *Neighborhood Set* genutzt werden, um eine gute Netzwerkverbindung zwischen den Knoten als Maß für Nähe zu nutzen. Eventuell kann über ein gewichtetes Hashing noch eine Lastverteilung über die *Independent Nodes* erreicht werden. Die Synchronisation zwischen den beteiligten Knoten über den Hypercube und das Overlay-Netzwerk wird in Abbildung 5.3 dargestellt.

In [Eßmann et al., 2005] wird das hier vorgestellte Replikationsverfahren mit einer Abwandlung des Hypercubes vorgestellt. Das abgewandelte Verfahren basiert auf *Sparse Graphen*⁷ deren Hauptidee die Aufteilung der Eckpunkte des Hypercubes in mehrere Knoten ist. Dies erlaubt eine höhere Anzahl von Knoten pro Dimension (vgl. Abbildung 5.4) und führt zu einer besseren Lastverteilung zwischen den Knoten im Graphen, da der Knotengrad bei leicht höherer Pfadlänge geringer ist. Der weitaus größere Vorteil der Sparse Graphen gegenüber dem Hypercube ist die Flexibilität in Bezug auf die Knotenzahl. Ein Sparse Graph muss nicht auf die nächst höhere Dimension erweitert werden, wenn die Knotenzahl des Hypercubes gefüllt ist. Aus Gründen der Optimierung kann dies jedoch bei Bedarf zu einem günstigen Zeitpunkt nachgeholt werden. Knotenausfällen kann oft mit lokalen Rekonfigurierungen begegnet werden. Erst eine ungünstige Partitionierung des Netzwerkes würde zu einer sofortigen Neuinitialisierung führen.

Das hier betrachtete Verfahren für die Replikation und Distribution der gemeinsamen Objekte einer Kooperationsgruppe in einem mobilen Nutzungsumfeld ergänzt die verfügbaren Technologien für eine dynamische und verteilte Kommunikation in Form von Overlay-Netzwerken um die wichtige Komponente der *Offline-Fähigkeit der Kooperationsobjekte* bei möglichst allen Kooperationsteilnehmern. Durch die Klassifikation der potentiell genutzten Geräte gemäß ihrer Fähigkeit zur Replikation des gemeinsamen Arbeitsbereichs wird eine Teilnahme auch für leistungsschwarze Hardwarekomponenten ermöglicht. Des Weiteren sorgen die Mechanismen zur Konfliktbehandlung für eine frühzeitige Aufdeckung von Konflikten und die Mögliche-

⁶Vgl. [Karger et al., 1997]

⁷Vgl. [Elsässer et al., 2001]

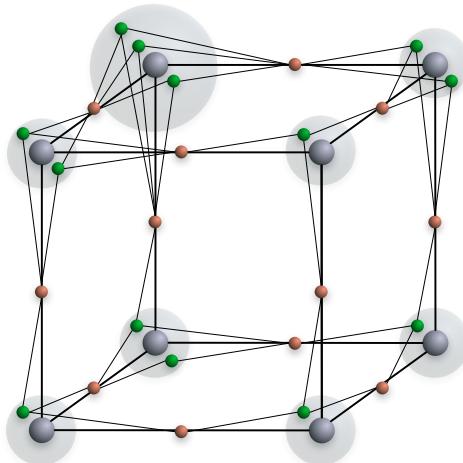


Abbildung 5.4: Distributionstopologie basierend auf einem *Sparse Graphen* der Dimension $d = 3$ mit 33 Knoten ($G_{(33)}$)

keit zur kooperativen Behebung derselben. Bei allen einbezogenen Verfahren wurde ein starker Fokus auf Effizienz und Transparenz für die Benutzer gelegt.

Auch wenn mit den hier vorgestellten Bausteinen bereits eine mobilitätsfreundliche verteilte Persistenz in eine Anwendung eingebunden werden kann, würde die Integration der Persistenzschicht doch einen Großteil der Entwicklungsarbeit für den Anwendungsentwickler einnehmen. Ziel ist es, ein Konzept zu entwickeln, das die komplexe Struktur der hier vorgestellten Persistenzschicht in einen gekapselten und universellen verteilten Speicher verwandelt. Der folgende Abschnitt beschäftigt sich mit dem flexiblen Konzept der Tuple Spaces, das diese Aufgabe leisten kann.

5.2 Gruppen-Tuple Spaces als Paradigma für eine Persistenzschicht mobil-verteilter Wissensräume

Die Kombination von Technologien zur spontanen Vernetzung, Kommunikation und Speicherung in verteilten Systemen mit der im vorigen Abschnitt vorgestellten Replikationsstrategie, ermöglicht einen strukturierten und verteilten Speicher, der die gewünschten Eigenschaften der mobil-spontanen Vernetzung und der Offline-Verfügbarkeit erfüllt. Der Zugriff auf einen derart verteilten und stark replizierten Speicher setzt aber von Benutzern wie Entwicklern eine tiefgehende Kenntnis der Konzepte und Strategien der verteilten Persistenz voraus. Die Komplexität des mobil-verteilten Speichers erschwert die Erstellung und Benutzung der darauf aufsetzenden Kooperationsanwendungen erheblich. Idealerweise sollte sich der Zugriff auf die Objekte so gestalten, als würden diese lokal vorliegen.

Auch fehlen in dieser Architektur noch geeignete Kommunikationsmechanismen für die mobile Kooperation mittels verteilter Anwendungen. Die Knoten, die die verteilte Anwendung aufspannen, sind mitunter nicht zeitgleich im Netzwerk erreichbar und können somit Nachrichten nicht direkt miteinander austauschen. Daher wird die Kommunikation in der hier vorgestellten Architektur nicht nur räumlich sondern auch zeitlich entkoppelt. Die in Abschnitt 4.4 beschriebenen Mechanismen leisten diese zeitliche Entkoppelung nicht, sondern bieten nur eine synchrone Kommunikation in einer räumlich verteilten Topologie.

Die in Abschnitt 4.3.2 vorgestellten *Tuple Spaces* bieten genau die Eigenschaften eines einfachen Objektzugriffs und einer räumlich und zeitlich entkoppelten Kommunikation. Der transparente Zugriff auf die gespeicherten Objekte wird mittels des *Tuple/Template-Konzepts* bewerkstelligt. Dieses Konzept wird im Folgenden noch einmal kurz skizziert.

Eine Anwendung, die auf den Tuple Space zugreift, speichert ein Datum, indem es dieses in Form eines Vektors von Attributen in ein *Tuple* verpackt und im *Tuple Space* ablegt (`in(tuple)`). Um ein Objekt aus dem *Tuple Space* zu lesen oder zu entnehmen wird über ein *Template* das gesuchte *Tuple* mittels der gewünschten Attributwerte definiert und bei einer Übereinstimmung an die Anwendung übergeben (`read(template)` und `out(template)`).

LIME zeigt, dass über dieselben Mechanismen des Tuple Space auch eine zeitlich und räumlich entkoppelte Kommunikation etabliert werden kann. Nachrichten werden analog zu den Datenobjekten ebenfalls als Tuple im Tuple Space abgelegt, können aber zusätzlich mit einer Empfängeradresse versehen werden, damit sie nur die gewünschten Knoten erreichen. Sind die Zielknoten mit dem netzweiten Tuple Space verbunden, liefert ein spezielles Template alle vorliegenden Nachrichten an den Knoten aus.

Zwar basieren viele Implementierungen von Tuple Spaces, wie z. B. *Linda*, *Java-Spaces* und *TSpaces*, auf einem lokalen Speicher für die Multiprozess-Kommunikation oder auf Client-Server-Konzepten mit zentralen Diensten im Netzwerk, aber es existieren auch Ansätze zu spontan vernetzten und verteilten Tuple Spaces, wie z. B. *LIME* (vgl. Abschnitt 4.3.2). Keiner der vorhandenen Ansätze bietet jedoch eine wirksame Replikationsstrategie für die Offline-Fähigkeit. Außerdem sind die existierenden Systeme nicht auf die Bedürfnisse gruppenbasierter Kooperationskonzepte zugeschnitten. Im Folgenden wird daher ein für mobil-verteilte Kooperationsumgebungen geeigneter Tuple Space entworfen.

In weiträumig verteilten und dynamisch vernetzten Tuple Spaces existiert wie in allen verteilten Speicherkonzepten das Problem der Distribution und Ortung der Datenobjekte (Tuple). Prinzipiell existieren zwei mögliche Methoden für die Verteilung und Ortung der Tuple in einem über das Netzwerk verteilten Tuple Space⁸. Die erste Methode ist die „*Hash-based Distribution*“, die eine Hashfunktion benutzt, um die Tuple auf die verfügbaren Knoten zu verteilen und anschließend wieder aufzufinden. Dieser Ansatz ähnelt stark den in Abschnitt 4.3.1 beschriebenen *Distributed Hash*-

⁸Vgl. [Fenwick und Pollock, 1997]

tables (DHT). Die zweite Methode wird „*Operator-based Distribution*“ genannt. Sie nutzt eine Replikation der Daten über die Knoten des verteilten Tuple Space.

Bei der „*Operator-based Distribution*“ lassen sich entweder die Tuple oder die Templates an alle beteiligten Knoten replizieren. Bei einem „*Positive Broadcast*“ werden die Tuple auf alle beteiligten Knoten repliziert und in deren lokalen Tuple Space gespeichert. Bei einem „*Negative Broadcast*“ werden hingegen die Templates an alle Knoten gesandt und dort gegen die lokal vorhandenen Tuple verglichen. Bei einer Übereinstimmung wird das lokale Tuple mit einer Übereinstimmung an den anfragenden Knoten gesandt. Darüber hinaus existiert mit dem „*Hybrid Broadcast*“ eine dritte Möglichkeit, die Tuple und/oder Templates jeweils an eine Teilmenge der Knoten versendet. Somit existieren im Wesentlichen die folgenden Vorgehensweisen für einen verteilten Tuple Space:

- *Hash-basierte Tuple Verteilung*: Tuples werden nach einem Schlüssel auf die Knoten verteilt.
- *Operator-basierte Verteilung*: Tuple oder Templates werden an alle Knoten oder eine Teilmenge der Knoten des Netzwerkes gesandt. Hier ergeben sich wieder drei mögliche Varianten:
 - *Negativer Broadcast*: Die Tuple werden lokal auf den Knoten gespeichert, die Templates werden an alle Knoten verteilt (Broadcast) und liefern die passenden Tuple zurück.
 - *Positiver Broadcast*: Die Tuple werden an alle beteiligten Knoten verteilt (Broadcast) und die Templates werden lokal auf die Tuple angewandt, die einen Knoten erreichen.
 - *Hybrid Broadcast*: Sowohl Tuple als auch Templates werden auf Untergruppen der Knoten des Netzwerkes verteilt.

Jedes der Verfahren hat je nach betrachtetem Anwendungskontext Vor- und Nachteile. Die Verteilung der Tuple anhand einer Hash-Funktion hat wie bei den DHTs eine gute Lastverteilung auf Kosten der ständigen Verfügbarkeit zur Folge. Fällt der speichernde Knoten aus, ist das Tuple nicht länger erreichbar. Bei Operatorbasierten Verfahren entsteht ein höherer Speicherbedarf und es muss die Wahrung der Kohärenz beachtet werden. Wenn z.B. ein *Negative Broadcast* mehrere Tuple zurückliefert, wird nur ein Tuple als Treffer gewählt. Die anderen ebenfalls passenden Tuple werden wieder in die Tuple Spaces eingefügt aus denen sie empfangen wurden. Des Weiteren zieht ein *Positive Broadcast* einen hohen Speicherbedarf bei allen beteiligten Knoten nach sich und der replizierte Tuple Space muss zur Wahrung der Konsistenz synchronisiert werden.

Mit den genannten Verfahren lässt sich die im vorigen Abschnitt 5.1 beschriebene Replikationsstrategie in einen verteilten Tuple Space einbetten. Die Benutzer erhalten so einfach und transparent Zugriff auf die hochverfügbare Persistenzschicht

inklusive eines Mechanismus für die zeitlich und räumlich entkoppelte Kommunikation zwischen den beteiligten Knoten.

Um eine flexible Nutzung des verteilten Speichers zu erlauben und dennoch gezielt ein gruppenbasiertes Arbeiten in gemeinsamen Arbeitbereichen zu unterstützen, wird der Tuple Space in dem hier vorgestellten Ansatz in drei Ebenen gegliedert. Einem *Network Tuple Space*, einem *Host Tuple Space* und einem *Gruppen-Tuple Space*.

Erstere zwei sind den Konzepten von *LIME* entlehnt und bilden die netzwerk- und geräteweite Speicherung und Kommunikation ab. Der *Network Tuple Space* spannt sich über die gesamte verteilte Umgebung und erlaubt die globale Speicherung von Daten, die nicht ständig verfügbar sein müssen. Um den Speicherbedarf möglichst gering zu halten, wird der globale *Network Tuple Space* über die beteiligten Knoten gleichmäßig verteilt (z. B. mittels Hashing). Der *Host Tuple Space* hingegen steht nur lokal auf dem jeweiligen Gerät zur Verfügung und wird auch nur lokal gespeichert. Er ist für den Austausch und die Kommunikation von lokalen Prozessen zuständig.

Die dritte Ebene bilden die so genannten *Gruppen-Tuple Spaces*, die im Rahmen dieser Arbeit speziell für die Gruppenarbeit konzipiert wurden. Von ihnen darf es in der Softwareumgebung beliebig viele geben. Die Knoten des verteilten Tuple Space müssen einen Gruppen-Tuple Space explizit abonnieren, um diesen lokal zu replizieren. Jeder Teilnehmer kann einen solchen Gruppen-Tuple Space spontan gründen und von diesem Moment an über einen eindeutigen Namen addressieren. Der Gruppen-Tuple Space enthält den von einer Knotengruppe replizierten gemeinsamen Inhaltsbereich und liegt somit auf jedem beteiligten Knoten redundant vor. Er wird gemäß der in Abschnitt 5.1 vorgestellten Replikationsstrategie zwischen den Abonnenten verteilt und synchronisiert. Zudem stellt der Gruppen-Tuple Space Mechanismen bereit, die der zugreifenden Anwendung Konflikte aufzeigen und diese lösen helfen.

Die in Abschnitt 5.1 angesprochenen Replikationsklassen „*Independent Nodes*“, „*Selective Nodes*“ und „*On Demand Nodes*“ können auch im Gruppen-Tuple Space genutzt werden, um die Replikationsstrategie anzupassen. Die Gruppenmitglieder eines Gruppen-Tuple Space können bei ihrem Beitritt mitteilen, zu welcher der drei Replikationsklassen sie gehören möchten. Die *Independent Nodes* replizieren den Gruppen-Tuple Space vollständig und jedem Gruppen-Tuple Space muss mindestens ein Independent Node zugeordnet sein. Andere Knoten replizieren lediglich einzelne und explizit gewählte Tuple lokal (*Selective Nodes*) oder greifen nur entfernt über einen Independent Node auf die Tuple des Gruppen-Tuple Space zu (*On Demand Nodes*). Zwischen den Mitgliedern der Klasse der Independent Nodes wird der Gruppen-Tuple Space wie in Abschnitt 5.1 beschrieben repliziert.

Je nach Berechtigung können auch Nichtmitglieder der Gruppe auf Tuple des Gruppen-Tuple Space zugreifen solange ein Independent Node erreichbar ist, der den Gruppen-Tuple Space repliziert. Die Gäste greifen wie Selective Nodes oder On Demand Nodes auf die Tuple zu, besitzen aber nicht dieselben Berechtigungen wie die Gruppenmitglieder. Die Zugriffsrechte hängen von den von der Gruppe vergebenen Berechtigungen für die einzelnen Tuple ab.

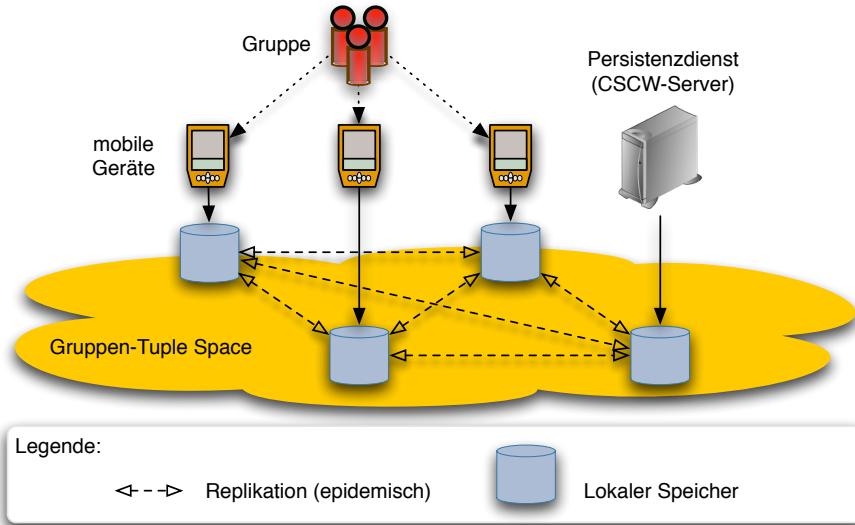


Abbildung 5.5: Replikation des Gruppen-Tuple Space zwischen *Independent Nodes* der Gruppenmitglieder und einen optionalen und nicht sichtbaren Persistenzdienst

Um die Verfügbarkeit des Gruppen-Tuple Space für On Demand Nodes und Selective Nodes zu erhöhen und gleichzeitig auch die Zeitabstände zwischen der Synchronisation der Independent Nodes zu verringern, kann ein zentral positionierter und stets erreichbarer Knoten als eine Art Persistenzdienst in Form eines nicht sichtbaren Mitglieds der Gruppe beitreten und deren Gruppen-Tuple Space replizieren. Dieses Vorgehen ermöglicht auch klassische CSCW-Server in eine solche Struktur einzubinden. Als unsichtbares Mitglied der Gruppe wird der Server automatisch mit allen Veränderungen im Tuple Space der Gruppe abgeglichen und kann so ein Abbild des gemeinsamen Arbeitsbereiches in der eigenen Persistenz vorhalten (vgl. Abschnitt 5.3). Abbildung 5.5 zeigt einen Gruppen-Tuple Space mit voll replizierenden Mitgliedern und einem zusätzlichen Knoten als Persistenzdienst.

Die Struktur des verteilten Tuple Space erinnert durch die Partitionierung in Gruppen-Tuple Spaces stark an die Gruppenstrukturen in virtuellen Wissensräumen. Das Konzept der Gruppen-Tuple Spaces kann daher leicht auf den virtuellen Wissensraum abgebildet und übertragen werden. Im virtuellen Wissensraum ist jeder Gruppe ein *Gruppenbereich* (oder auch Gruppenraum) zugeordnet, in dem die zur Kooperation benötigten Objekte abgelegt werden. Jede Gruppe im virtuellen Wissensraum besitzt somit einen eigenen Gruppen-Tuple Space in der Persistenzschicht und legt die gemeinsamen Objekte als Tuple in diesem ab. Der so gespeicherte gemeinsame Arbeitsbereich ist dank der Replikationsstrategie, zum einen stets so synchron wie möglich und zum anderen stets für die Kooperationsteilnehmer erreich-

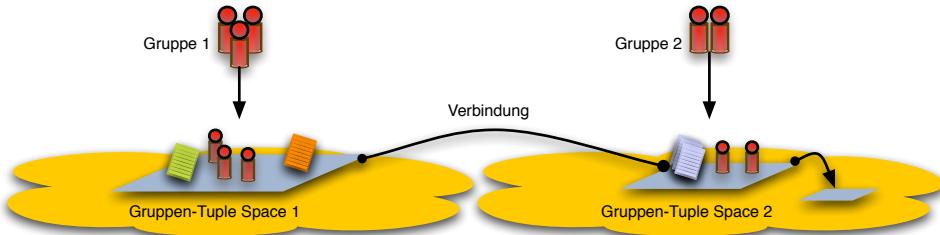


Abbildung 5.6: Virtuelle Wissenräume zweier Kooperationsgruppen mit den zugeordneten *Gruppen-Tuple Spaces*

bar, unabhängig von der Verbindung zu den anderen Gruppenmitgliedern (Offline-Verfügbarkeit).

Die Raumstruktur der Gruppenbereiche innerhalb des Wissensraumes ist üblicherweise zwischen den einzelnen Gruppen über so genannte *Gänge* verknüpft. Diese Möglichkeit bleibt auch bei dessen verteilter Speicherung in Gruppen-Tuple Spaces gegeben. Das Gerät eines Nichtmitglieds, das über einen Gang in einen fremden Gruppenarbeitsraum gelangt, greift als Gast auf den entsprechenden Gruppen-Tuple Space zu. Die Berechtigungen für derartige „externe“ Zugriffe legen die Gruppenmitglieder zuvor explizit fest. Abbildung 5.6 zeigt die über den Gruppen-Tuple Space verteilte Raumstruktur zweier Gruppen.

Die gesamte Gruppenkommunikation und -koordination kann ebenfalls über den Gruppen-Tuple Space abgewickelt werden, indem die Nachrichtenobjekte in diesem abgelegt werden. Die Knoten erhalten die Nachrichten, sobald sie sich mit dem Persistenzsystem verbinden. Über den Network Tuple Space kann zusätzlich eine zeitlich entkoppelte Punkt zu Punkt Kommunikation zwischen beliebigen Knoten etabliert werden. Ein Nachrichtenobjekt wird über das Hashing in Richtung Knoten geroutet und in dessen „Nähe“ gespeichert, bis dieser sich wieder an dem System anmeldet. Eine begrenzte Lebenszeit von Nachrichten und Objekten hilft hier den Speicherplatzbedarf zu kontrollieren und Knoten davor zu bewahren, Nachrichten zwischenspeichern, die nie ausgeliefert werden.

Für eine systemweite Suche nach Objekten kann eine Doppelstrategie angewandt werden. Analog zu den Persistenzdiensten werden *Indexdienste* auf zuverlässigen Knoten etabliert, die alle verfügbaren Objekte verzeichnen. So ist eine schnelle und zuverlässige Suche in dem verteilten Tuple Space möglich, solange sich ein solcher Dienst in Reichweite befindet. Für eine Suche wird ein entsprechendes Template an den Indexdienst gesandt und dieser leitet es an den Knoten weiter, der über das Tuple verfügt. Die zweite Variante nutzt einen *Negative Broadcast*, der das Template an alle Knoten des Tuple Space sendet. So ist eine globale Suche im System auch ohne Indexdienste möglich, wenngleich diese nicht so performant und zuverlässig ist, wie die Suche über einen Indexdienst.

Das Konzept der hier vorgestellten Tuple Spaces offeriert Anwendungen einen transparenten Zugriff auf eine verteilte und stark replizierte Persistenzschicht. Eine Anwendung, die über die hier betrachteten Tuple Spaces auf den gemeinsamen Speicher zugreift, kann dies in der gleichen Weise tun, als wenn alle Objekte lokal vorlägen. Durch die Erweiterung des Konzeptes der Tuple Spaces um die Gruppen-Tuple Spaces wird diesem verteilten Speicher eine logische Struktur gegeben, die eine auf die gruppenbasierte Arbeit zugesetzte Replikationsstrategie erlaubt.

Obwohl diese Konzepte auch für andere Anwendungsszenarien nutzbar sind, eignen sie sich doch im besonderen Maße für die Implementierung eines mobil-verteilten virtuellen Wissensraums. Die enthaltene Replikationsstrategie sorgt für die Offline-Verfügbarkeit der Kooperationsobjekte und orientiert sich intuitiv an der Gruppenstruktur. Mittels der Abstraktionsschicht der Gruppen-Tuple Spaces und der darauf aufsetzenden kooperationsstützenden Schicht eines mobil-verteilten Wissensraumes sind auch die letzten Lücken der Musterarchitektur für eine verteilte Kooperationsumgebung zur Kooperation in mobil-spontanen Nutzungsszenarien geschlossen.

Für eine offene Architektur fehlt noch ein Konzept zur Einbettung externer Dienste in dieses Modell. Daher beschäftigt sich der folgende Abschnitt mit der flexiblen Integration externer Dienste in die Kooperationsumgebung.

5.3 Einbettung externer Dienste in die mobil-verteilte Kooperationsumgebung

Die Einbettung externer Dienste in eine Kooperationsumgebung bietet den Benutzern Zugriff auf Ressourcen außerhalb der Kooperationsumgebung, ohne diese für deren Nutzung verlassen zu müssen. Weiterhin kann die Kooperationsumgebung mit diesen externen Diensten die Kooperationsprozesse der Benutzer unterstützen oder um lokal nicht vorhandene technische Funktionalitäten ergänzen.

Da die externen Dienste nicht überall und in jeder Kooperationssituation verfügbar sind, darf sich die Kooperationsumgebung keinesfalls auf deren Präsenz verlassen oder die Kooperation von diesen externen Diensten abhängig machen. Wichtige Ziele einer Einbettung externer Dienste sind daher eine lose Kopplung an deren Funktionalität und eine stimmige Integration der Dienstleistungen in die Nutzungskonzepte der Kooperationsumgebung.

Unter der Forderung nach einer losen Kopplung wird in diesem Zusammenhang eine Einbettung der externen Dienste bei Verfügbarkeit, ohne Einschnitte in die Grundfunktionalität der Kooperationsumgebung bei deren Abwesenheit, verstanden. Die stimmige Integration in die Nutzungskonzepte bezieht sich auf eine medienbruchfreie Einbettung der externen Dienste in die Kooperationsumgebung, ohne neue Nutzungshemmnisse zu schaffen.

Das Feld möglicher externer Dienste, die sich in eine Kooperationsumgebung einbetten lassen, um Kooperationsprozesse zu unterstützen, ist schwer einzugrenzen. Im Folgenden werden die vier wichtigsten Gruppen externer Dienste kurz erläutert.

Eine wichtige Gruppe externer Dienste, die die Kooperationsumgebung bei der Konfiguration und Darstellung des Wissensraums unterstützen, bilden die *Kontextdienste*. Ein Kontextdienst kann z. B. Positionsinformationen von Kooperationspartnern übermitteln oder den räumlichen Kontext der Kooperationssituation bereitstellen [Eßmann und Hampel, 2004]. Man spricht in diesem Zusammenhang auch von einer *Location Awareness*. Neben dem Ortsbezug können noch viele weitere Kontextinformationen für eine *Awareness* der Handlungen der Kooperationspartner bereitgestellt werden⁹.

Persistenzdienste bilden eine weitere Gruppe externer Dienste, die insbesondere für die Ergänzung mobil-verteilter Persistenzkonzepte interessant ist. Diese können z. B. von dedizierten CSCW-Servern bereitgestellt werden, die Teile des mobil-verteilten Datenraumes spiegeln, so die Verfügbarkeit der enthaltenen Daten erhöhen und zugleich Zugangsmöglichkeiten für Client-Server-basierte Kooperationswerkzeuge bereitstellen. Das Konzept des *Gruppen-Tuple Space* sieht bereits eine Integration solcher Dienste als implizite Mitglieder vor (vgl. Abschnitt 5.2).

Auch Dienste, die nicht direkt in die Funktionalität der eigentlichen Kooperationsumgebung eingreifen, sind oft für eine Kooperationsunterstützung wertvoll. So können z. B. *Compute-Dienste* genutzt werden, die Objekte außerhalb der Kooperationsumgebung transformieren oder auswerten. Diese können Benutzern Hilfestellung bei der Bearbeitung der Kooperationsobjekte geben oder rechenintensive Aufgaben der mobilen Geräte übernehmen. Beispiele sind hier die Einbindung von Visualisierungssystemen zur Darstellung und kooperativen Manipulation komplexer Datensätze [Götz et al., 2006, 2005; Eßmann et al., 2006a, b] und die Integration von *Computer Algebra Systemen (CAS)* für Berechnungen an mathematischen Objekten innerhalb des gemeinsamen Arbeitsbereiches [Bleckmann et al., 2005].

Auch die automatische Generierung von Inhalten für den gemeinsamen Arbeitsbereich ist ein Anwendungsfeld externer Dienste. *Informationsdienste* können z. B. aufbereitete Informationen und Datenobjekte für die Kooperationsgruppe bereitstellen. Möglich ist hier beispielsweise die Einbindung von externen *RSS Feeds* in den gemeinsamen Arbeitsbereich.

Bei der Integration verfügbarer und gewünschter externer Dienste in die mobil-verteilte Umgebung ist deren automatische Konfiguration wichtig. Die Integration kann dabei in unterschiedlichen Schichten der Kooperationsumgebung geschehen.

Das Auffinden und die Konfiguration des Zugriffs auf externe Dienste ist eine typische Aufgabe der Netzwerkschicht. In administrierten Netzwerkinfrastrukturen werden hier häufig Verzeichnisdienste, wie z. B. *Lightweight Directory Access Protocol (LDAP)*-Server, für die Veröffentlichung der verfügbaren Dienste und deren Konfiguration genutzt. Es existieren aber auch Lösungen, die vorhandene Dienste ohne administrativen Eingriff sowohl in der *lokalen Netzwerkumgebung* als auch in *weiträumig verteilten Systemen* bekannt machen (vgl. Abschnitt 4.2.2). Diese Dienstinformationen werden in der Netzwerkschicht gesammelt und an die Kooperations-

⁹Vgl. [Dourish und Bellotti, 1992]

umgebung weitergereicht, die diese externen Dienste einbindet und den Benutzern zugänglich macht.

Dienste, die auf Objekte in der Kooperationsumgebung zugreifen, müssen in die Persistenzkonzepte eingebettet werden, um auf die externen Objekte auch bei Abwesenheit des entsprechenden Dienstes zugreifen zu können. Um diese Integration zu erleichtern bietet sich ein Vorgehen wie in Abschnitt 5.2 an, in dem Persistenzdienste als implizites Gruppenmitglied in die Replikationskonzepte eingebettet werden und so die Daten der Gruppe automatisch replizieren. Die Persistenzdienste werden also nicht in der Schicht der Objektverwaltung und -replikation eingebunden, sondern auf einer höheren Abstraktionsebene integriert. Dies erlaubt eine durchgängige Einbettung externer Persistenzdienste in die mobil-verteilte Kooperationsumgebung als eine Art implizite Gruppenmitglieder.

Eine besondere Klasse externer Dienste bilden jene, die eine direkte Interaktion mit dem Benutzer unterstützen. Sie sind in die Benutzerschnittstelle zu integrieren und als eigenständige Objekte im gemeinsamen Arbeitsbereich zugänglich zu machen. Für die Kooperation innerhalb virtueller Wissensräume bedeutet dies die Integration der Dienste in Form üblicher Kooperationsobjekte mit allen Möglichkeiten zu deren kooperativen Bearbeitung über die bereitgestellten *Medienfunktionen* [vgl. Eßmann et al., 2006a].

Durch das hier beschriebene Vorgehen ergibt sich eine durchgängige Einbettung externer Dienste in eine mobil-verteilte Kooperationsumgebung für virtuelle Wissensräume. Mittels des hier vorgestellten Ansatzes der Integration externer Dienste wird eine Anreicherung der Kooperation mit Kontextinformationen, die Erhöhung der Zuverlässigkeit der verteilten Persistenz und eine Einbettung von Kooperationsobjekten, die der Umgebung z. B. aufgrund beschränkter Ressourcen der mobilen Geräte nicht zur Verfügung stehen würden, erreicht. Besondere Aufmerksamkeit wird dabei auf die Bewahrung der Unabhängigkeit der Grundfunktionalität der mobil-verteilten Kooperationsumgebung von externen Dienstleistern und auf die Vermeidung von Medienbrüchen gelegt.

5.4 Gesamtarchitektur mobil-verteilter Wissensräume

Die in diesem Kapitel vorgestellten Technologien und entworfenen Konzepte fügen sich zu dem gesuchten Architekturmuster für eine Kooperationsumgebung zur Bereitstellung mobil-verteilter Wissensräume zusammen. Wie bereits zu Beginn des Kapitels angedeutet, setzt sich diese Musterarchitektur aus mehreren Forschungsbereichen zusammen. Die unterste Ebene einer solchen Architektur bildet die *Kommunikationsschicht* mit dem physikalischen Netzwerk und den darauf aufsetzenden Protokollen. Sie ermöglicht als Fundament die *Kommunikation zwischen den beteiligten Knoten*.

Auf diese Schicht setzt die Schicht für die *Objektverwaltung und -distribution* auf. Diese Schicht enthält u. a. ein *Overlay-Netzwerk* das die Paket-basierte Kommunikation zwischen den Knoten in der heterogenen Netzwerkschicht maskiert. Zusätzlich

findet sich hier die eigentliche verteilte Persistenz, die auf das Overlay-Netzwerk zurückgreift und sich für *die Speicherung und das Auffinden von Objekten innerhalb des verteilten Systems* verantwortlich zeichnet. Diese Schicht bildet den technischen Kern der mobil-verteilten Kooperationsumgebung.

Die dritte Schicht abstrahiert von der technischen Komplexität der verteilten Objektspeicherung und entkoppelt so die logische Struktur mobil-verteilter Kooperation von der technischen Struktur der Vernetzung. Sie wird daher als *Abstraktionsschicht* bezeichnet. Sie orientiert sich in ihren Konzepten am *Tuple Space*. Mittels dieser Konzepte wird sowohl die *intuitive Strategie zur Objektrepikation* als auch eine *zeitlich wie räumlich entkoppelte Kommunikation* zwischen den beteiligten Anwendungen realisiert. Des Weiteren erlaubt diese Abstraktionsschicht den Entwicklern von mobil-spontanen Kollaborationsumgebungen einen transparenten Zugriff auf die Kooperationsobjekte, ohne sich mit den Fragen der zugrunde liegenden technischen Infrastruktur beschäftigen zu müssen. Dies hilft ihnen sich auf den Entwurf der eigentlichen Kooperationsanwendung zu konzentrieren.

Die oberste Schicht des Architekturkonzeptes bildet der eigentliche *mobil-verteilte Wissensraum*. Er bildet die Konzepte der Kooperation im virtuellen Wissensraum ab. Diese Schicht strukturiert die Kooperationsumgebung über die raumbasierte Metapher der *virtuellen Wissensräume* und stellt die elementaren *primären Medienfunktionen* für die Manipulation der enthaltenen Kooperationsobjekte zur Verfügung. Diese Schicht ist zugleich die höchste Abstraktionsebene der Architektur und bildet über das Konzept des virtuellen Wissensraumes die Schnittstelle zu den Benutzern der mobil-spontanen Kooperationsumgebung. Diese greifen über die genutzten Anwendungen auf den Wissensraum zu.

In den nun folgenden Abschnitten wird dieses *Vier-Schichten-Modell des Architekturkonzeptes einer verteilten Persistenz für virtuelle Wissensräume in mobil-spontanen Nutzungsszenarien* zusammengeführt [Eßmann und Hampel, 2005b] und Schicht für Schicht betrachtet (vgl. Abbildung 5.7). Im Rahmen dieses Überblicks werden noch einmal die Lösungsansätze für die im Rahmen dieser Arbeit identifizierten zentralen Forschungsfragen präsentiert.

Kommunikationsschicht

Die Vernetzung mobiler Knoten ist eine wesentliche Voraussetzung für die Kommunikation innerhalb der mobil-verteilten Kooperationsumgebung. Aufgrund der hohen Mobilität der Benutzer wird eine Kooperationsanwendung auf den mobilen Geräten mit einem *ständig wechselnden Einsatzumfeld* konfrontiert. Bei der Vernetzung mit den lokalen aber auch entfernten Kooperationspartnern ergibt sich eine *stark heterogene Umgebung*. Zugangspunkte zu etablierten Netzwerkinfrastrukturen sind eventuell nicht vorhanden oder stark reglementiert (Funklöcher, Firewalls, etc.). Um dennoch, zumindest mit lokalen Partnern mittels der mitgeführten mobilen Geräte spontan in Kooperation treten zu können, wird eine unabhängige und direkte Vernetzung zwischen den Beteiligten benötigt.

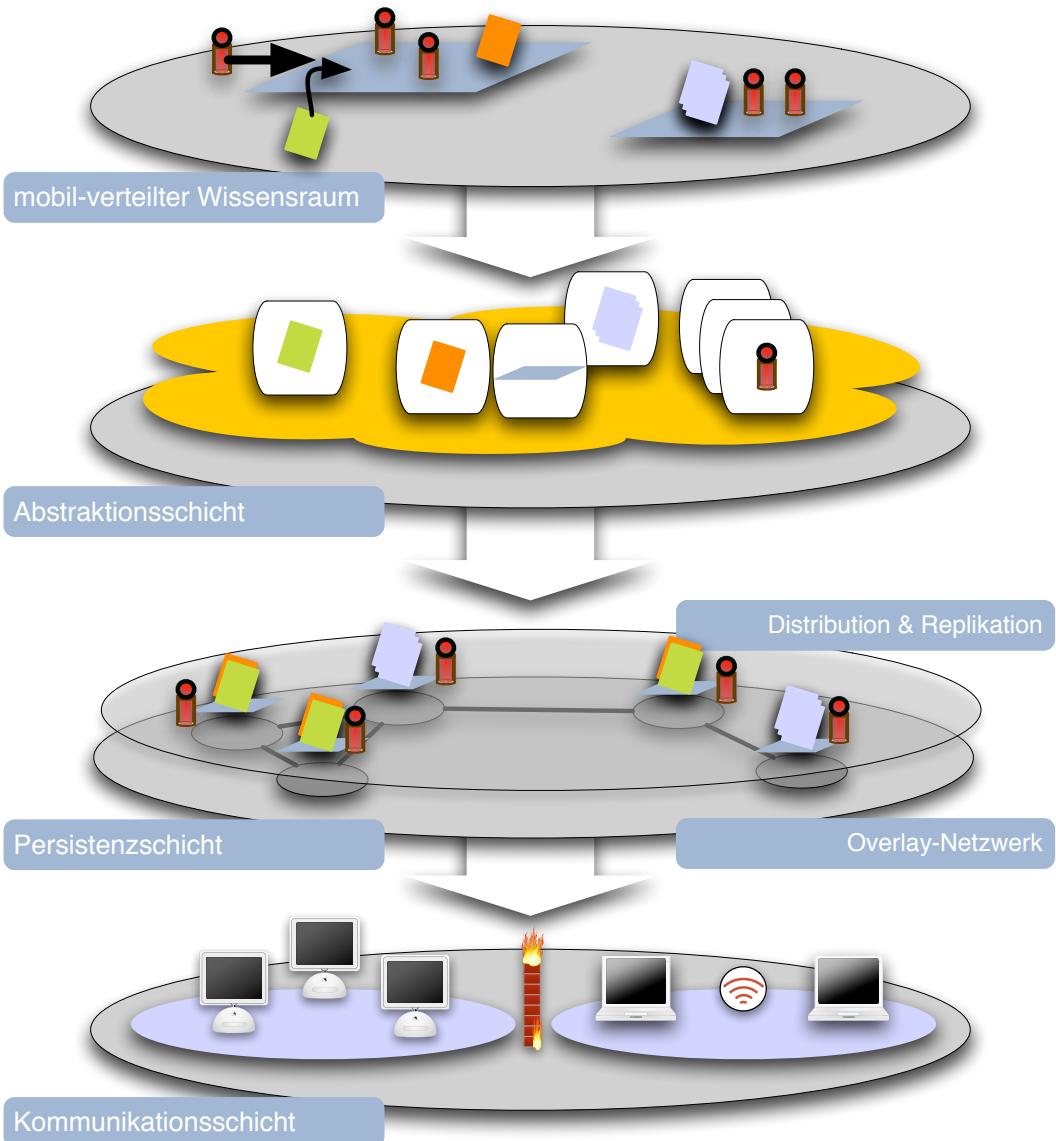


Abbildung 5.7: Die exemplarische Architektur einer mobil-spontanen Kooperationsumgebung gliedert sich in vier Schichten: Die physikalische Netzwerkschicht stellt eine grundlegende Punkt-zu-Punkt Kommunikation zwischen den Knoten bereit. Ein *Overlay-Netzwerk* gewährleistet die Verbreitung von Nachrichten und Objekten. Die Abstraktionsschicht (z. B. in Form eines Tuple Space) bietet einen netzweiten Speicher. Der mobil-verteilte virtuelle Wissensraum ermöglicht eine flexible Kooperation in sowohl mobil-spontanen als auch klassischen Nutzungsszenarien.

Wie Abschnitt 4.2.1 zeigt, ist ein breites Spektrum an möglichen Technologien für eine Vernetzung von mobilen Knoten verfügbar. Besonders geeignet für die so genannte *Face-to-Face Kooperation* sind die *Mobilen Ad-Hoc-Netzwerke*, die unabhängig von den technischen Gegebenheiten der Umgebung in der Lage sind, Kommunikationsinfrastrukturen von Grund auf zu etablieren. Sollen auch räumlich entfernte Partner oder externe Kooperationsdienste in die Kooperation eingebunden werden, ist eine Mischung von spontanen und administrierten Netzwerkinfrastrukturen nahezu unumgänglich. Auch hier existieren bereits einige Lösungen, die im Rahmen dieser Arbeit in das Gefüge einer Musterarchitektur integriert werden.

Die so entstehenden komplexen Strukturen, die in Kombination mit künstlichen Restriktionen seitens der Dienstanbieter von Kommunikationsnetzwerken zu Hürden für die Vernetzung einer verteilten Softwareumgebung bilden können, werfen Fragen bezüglich der Etablierung stabiler Kommunikationsinfrastrukturen zwischen den verteilten Knoten einer Kooperationsumgebung auf. Zum einen ist das Auffinden von potentiellen Kooperationspartnern in solch komplexen und weitreichenden Infrastrukturen schwierig, zum anderen wird die direkte Kommunikation zwischen zwei Knoten durch eventuelle Inkompatibilitäten zwischen den genutzten Protokollen und durch künstliche Hemmnisse zum Schutz institutioneller Interessen erschwert.

Die Lösung bieten hier Peer-to-Peer-Netzwerke, die Mechanismen bereitstellen, um derartige Hemmnisse in der Kommunikationsstruktur zu umgehen und mögliche Kommunikationspartner zu finden. In Kombination mit einer spontanen und direkten Vernetzung von anwesenden potentiellen Kooperationspartnern bietet solch eine Technik die Möglichkeit, auch große Kooperationsnetzwerke zu errichten. Eine derartige Kommunikationsarchitektur wurde in [Eßmann et al., 2004c] umgesetzt.

Es existieren somit geeignete Technologien zur spontanen wie weiträumigen Vernetzung von Kooperationspartnern, wenngleich sie noch nicht für die breite Masse der Benutzer verfügbar ist. Um zukünftig eine flexible Kommunikation der Benutzer gewährleisten zu können, wird es ein wichtiges Ziel sein müssen, im Bereich der Technologien zur spontanen Vernetzung offene Standards zu etablieren und in die physikalische Kommunikationsschicht gängiger Betriebssysteme zu integrieren. In Kombination mit Technologien zur automatischen Dienstekonfiguration und Peer-to-Peer Kommunikation in stark heterogenen Netzwerkinfrastrukturen bildet dies die technische Grundlage für eine allseits verfügbare mobile Kommunikationschicht ohne spürbare technische Barrieren. Eine derart flexibel vernetzte Kommunikationsinfrastruktur bietet ein ideales Umfeld für kooperationsunterstützende Anwendungen im mobilen Nutzungsumfeld.

Objektverwaltung und -distribution

Um in einer sich ständig dynamisch ändernden verteilten Umgebung eine persistente Speicherung von Kooperationsobjekten zu gewährleisten, bedarf es einer ebenfalls verteilten *Objektspeicherung und -verwaltung*. Die hohe Wahrscheinlichkeit einer Trennung der mobilen Kooperationspartner von etablierten Netzwerkinfrastrukturen verbietet ein auf einzelne zentrale Knoten konzentriertes Persistenzkonzept. Um

die Verfügbarkeit der Kooperationsobjekte zu erhöhen, werden diese daher auf den beteiligten Knoten verteilt abgelegt und verwaltet.

Mit Hinblick auf die Unterstützung von Kooperationsprozessen zwischen mobilen Benutzern genügt eine verteilte Objektpersistenz besonderen Anforderungen. So sorgt die im Rahmen dieser Arbeit entworfene Persistenzschicht für die Verfügbarkeit der kooperationsrelevanten Objekte bei allen beteiligten Benutzern (*Offline-Verfügbarkeit*). Die kooperativen Handlungen laufen somit nicht Gefahr in Abhängigkeit von Zeitfenstern und Plätzen mit Zugang zu den benötigten Objekten zu geraten. Gerade hier zeigten sich die Schwächen existierender verteilter Persistenzsysteme.

Eine Kooperation in replizierten Datenräumen mittels ebenfalls replizierter Kooperationsobjekte gelingt nur, wenn die Replikate stets zueinander konsistent sind. Dies ist besonders im Fall einer unverbundenen Bearbeitung von replizierten Objekten nicht garantiert. Daher werden etwaige Inkonsistenzen so früh wie möglich aufgedeckt und behoben. Eine Weiterverwendung inkonsistenter Objekte würde automatisch zu kaskadierenden Inkonsistenzen führen und hätte einen divergierenden und eben nicht mehr gemeinsamen Datenraum zur Folge.

Die verteilte Persistenzschicht der hier vorgestellten Architektur bietet sowohl die ständige Verfügbarkeit der Objekte innerhalb einer Kooperationsgruppe mittels einer *gezielten Replikation der Objekte auf alle an der Kooperation beteiligten Knoten* als auch die Wahrung der Konsistenz des gemeinsamen Wissensraum durch eine *epidemische Synchronisation der Kooperationsobjekte* und deren *Versionierung für die Konfliktkennung und -auflösung* (vgl. Abschnitt 5.1).

Diese positiven Eigenschaften werden durch die Komposition von bewährten DHT-Konzepten und einer engmaschigen epidemischen Replikationsstrategie erreicht. Das resultierende Persistenzkonzept eignet sich insbesondere für die mobile Gruppenarbeit. Im Gegensatz zu gängigen verteilten Persistenzsystemen ist es möglich die Kooperationsobjekte gezielt auf die Knoten der beteiligten Kooperationspartner zu replizieren und synchronisiert zu halten. Für den Fall eines Konflikts zwischen den replizierten Versionen eines Objektes wird dieser aufgrund der Versionierung sofort erkannt und der Kooperationsumgebung angezeigt. Diese versucht draufhin den Konflikt automatisch und im Hintergrund zu lösen oder reicht ihn an die beteiligten Benutzer weiter, die den Konflikt dann kooperativ beheben können. Der Einsatz des DHT-Konzeptes erlaubt es, Objekte effizient über das Netzwerk innerhalb der Gruppe zu verteilen und bei Bedarf zusätzlich global zugänglich zu machen.

Die technische Realisierung der gezielten Objektreplikation geschieht mittels einer logischen Strukturierung von Inhaltsbereichen und im Netzwerk gespeicherten Objektverzeichnissen inklusive Versionsinformationen der enthaltenen Objekte. Die Objekte werden in den Inhaltsbereichen gebündelt. Hier können sie anhand der Informationen über die gespeicherte Version des Objektes mit ihrem jeweiligen lokal replizierten Pendant verglichen werden, um festzustellen, ob eine Synchronisation der Repliken notwendig ist, oder gar ein Versionskonflikt vorliegt. Gleichzeitig wird in diesem Verzeichnis vermerkt, welche Knoten ein Objekt replizieren. So können Änderungen direkt über eine DHT an die betroffenen Knoten weiterverteilt werden. Interessierte Knoten müssen einen Inhaltsbereich lediglich abonnieren, um an der

Replikation der enthaltenen Objekte teilzunehmen. Ab diesem Zeitpunkt werden sie automatisch mit den enthaltenen Objekten synchronisiert.

Zusätzlich sieht die vorgestellte Replikationsstrategie eine Abstufung des Replikationsgrades entsprechend der Ressourcen teilnehmender Knoten vor und erlaubt so auch Geräten mit wenig Speicherplatz den Zugriff auf die stark replizierten verteilten Objekte. Diese müssen aufgrund der mangelnden lokalen Replikation lediglich einige Einschränkungen der Offline-Verfügbarkeit der Objekte in Kauf nehmen.

Übertragen auf den gemeinsamen Wissensraum kann der Arbeitsbereich einer Gruppe einen Inhaltbereich der verteilten Persistenzschicht zugeordnet werden, um so eine verteilte und synchronisierte Speicherung der Kooperationsobjekte zu gewährleisten. Zusätzlich erlaubt das Konzept auch die Integration zentraler Persistenzdienste wie CSCW-Server in die verteilte Persistenzschicht. So lässt sich die Verfügbarkeit der Kooperationsobjekte noch weiter erhöhen. Der Persistenzdienst muss dazu lediglich ebenfalls den Inhaltbereich abonnieren und so den aktuellen Inhalt des Arbeitsbereichs der Kooperationsgruppe spiegeln.

Abstraktionsschicht

Die verteilte Persistenzschicht ist offensichtlich in der Lage, die nötigen Mechanismen zur Verfügung zu stellen, um die Kooperationsobjekte für mobile Benutzer ständig verfügbar zu halten und die Konsistenz des Datenraums über eine weitreichende Synchronisation zu wahren. Prinzipiell wäre schon an diesem Punkt die Implementierung einer mobil-verteilten Kooperationsumgebung möglich. Allerdings müssten in diesem Fall die Kommunikation zur Koordinierung der beteiligten Knoten und die Verwaltung der Inhaltbereiche von jeder Kooperationsanwendung selbst implementiert werden. Dies schließt auch die Mechanismen zum Abgleich der lokalen Objekte mit den abonnierten Inhaltbereichen und deren Objektverzeichnisse mit ein. Zusätzlich erschwert ein solches Vorgehen einen Austausch oder eine Ergänzung der Persistenzschicht durch andere Persistenzkonzepte wegen der starken funktionellen Bindung zwischen Kooperationsanwendung und Objektspeicherung.

Um eine hohe Abstraktion von der eigentlichen Objektverteilung und -replikation für die Entwickler der Kooperationsanwendungen zu wahren und eine nicht an Kooperationsmetaphern gebundene Kapselung der Persistenzschicht zu erreichen, wird der verteilte und replizierte Objektspeicher als ein generischer Speicher abstrahiert. In diesem können Objekte abgelegt und wieder geladen werden, ohne von der genauen Speicherposition des Objekts im Netzwerk zu wissen oder die Replikation explizit festzulegen. Dennoch ist der generische Speicher in der Lage, die *Offline-Verfügbarkeit der Objekte* und die Wahrung der *Konsistenz des gemeinsamen Datenraums* zu garantieren (vgl. Abschnitt 5.2).

Für die Umsetzung dieser wünschenswerten Eigenschaften wird im Rahmen dieser Arbeit das Konzept des *Gruppen-Tuple Spaces* entworfen. Dieser fügt dem bewährten Konzept der Tuple Spaces eine an der Gruppenarbeit orientierte Replikationsstrategie hinzu. In Gruppen-Tuple Spaces werden Knoten, die auf einen gemeinsamen

und replizierten Datenraum zugreifen wollen, in einem gemeinsamen *Tuple Space*¹⁰ gruppiert und können dort mittels einfacher Operationen Objekte (*Tuple*) gespeichert und wieder entnommen werden (vgl. Abschnitt 4.3.2). Dabei kann ein Knoten auf beliebig viele Gruppen-Tuple Spaces gleichzeitig zugreifen und ohne weiteres selber welche generieren.

Auf Wunsch informiert der Tuple Space die Anwendungen wenn ein neues Objekt erscheint oder ein bereits existierendes verändert wurde. Die Anwendung reagiert somit nur auf diese Ereignisse. Ein besonderes Ereignis ist hier der Konflikt von überschneidend geänderten Objekten. Dieser kann insbesondere nach einer Trennung eines Knotens von den anderen Mitgliedern des gemeinsamen und replizierten Tuple Spaces auftreten. Durch die Ereigniskontrolle werden in so einem Fall automatisch alle betroffenen Anwendungen benachrichtigt und das Objekt bis zur Behebung des Konflikts für eine weitere Bearbeitung gesperrt, um ein Kaskadieren der Konflikte zu vermeiden.

Das Konzept des Gruppen-Tuple Spaces stellt zusätzlich zu dem gemeinsamen replizierten Speicher die Mechanismen für eine mobilitätsfreundliche Kommunikation unter den beteiligten Knoten bereit. Zu diesem Zweck werden die Nachrichtenobjekte an eine Knotengruppe oder an einzelne Knoten ebenfalls als Tuple in dem gemeinsamen Tuple Space abgelegt. Dies erlaubt eine *zeitlich und räumlich entkoppelte Kommunikation zwischen den Knoten*, die so Nachrichten auch dann erhalten, wenn sie zeitweise nicht erreichbar sind oder über keine direkte Verbindung zu dem Absenderknoten verfügen. Dieses Nachrichtensystem wird auch für die Verbreitung von Ereignissen zwischen den verteilten Knoten genutzt.

Neben den replizierten und stark gekoppelten Gruppen-Tuple Spaces existieren außerdem ein *Network Tuple Space* und je Gerät ein *Host Tuple Space*, die es erlauben, Objekte und Nachrichten global innerhalb des gesamten Netzwerkes oder lokal auf den Geräten auszutauschen. Diese Tuple Spaces sind allerdings nicht persistent und für die Koordination zwischen fremden Anwendungen konzipiert. Gemein mit den Gruppen-Tuple Spaces ist ihnen die räumliche und zeitliche Entkopplung der Kommunikation und die Fehlerrobustheit gegenüber Verbindungsabbrüchen.

Aufgrund der Integration von automatischer Replikation, Ereigniskontrolle und entkoppelter Kommunikation ist diese Schicht der ideale Ort für die Einbettung externer Dienste in die Kooperationsumgebung. Externe Dienste, die Informationen zur Verfügung stellen oder Daten mit der Kooperationsumgebung austauschen, können ihre Daten über den Tuple Space an alle Teilnehmer verbreiten, ohne deren aktuellen Verbindungsstatus beachten zu müssen. Dies gilt insbesondere für eine Einbindung von Persistenzdiensten, die mit ihrer hohen Erreichbarkeit die Verfügbarkeit des gemeinsamen Datenraums deutlich erhöhen. Der Persistenzdienst muss dazu lediglich dem Tuple Space als implizites Mitglied beitreten, um diesen von da an ohne weiteres Zutun zu spiegeln (vgl. Abschnitt 5.3).

¹⁰Um allerdings nicht die historischen Begrenzungen des *Tuple Space* in *Linda* (vgl. Abschnitt 4.3.2) in Kauf nehmen zu müssen, orientiert sich das vorgestellte Konzept nur grob an seinem Vorbild.

Die Gruppen-Tuple Spaces vereinen in einem schlüssigen Konzept die Mechanismen der Replikation, Ereigniskontrolle und Gruppenkommunikation. Gleichzeitig sind sie ein äußerst geeignetes Werkzeug für die Koordinierung von mobil-verteilten Anwendungen und für die persistente und konsistente Speicherung derer Daten. Dabei erlaubt der Gruppen-Tuple Space einen transparenten Zugriff auf die Persistenzschicht, ohne deren Flexibilität einzuschränken.

Mobil-Verteilter Wissensraum

Für die Bereitstellung eines mobil-verteilten virtuellen Wissensraums, basierend auf der im Rahmen dieser Arbeit entworfenen verteilten Persistenzschicht, wird eine weitere Schicht errichtet, die das *Konzept des virtuellen Wissensraums* auf die *Gruppenbasierten Tuple Spaces* überträgt. Ziel dieser Schicht ist die *Bereitstellung der Strukturen und Mechanismen des virtuellen Wissensraumes auf Basis des Konstrukts der Gruppen-Tuple Spaces*.

Dazu bietet sich die Abbildung je einer Benutzergruppe mit ihrem gemeinsamen Arbeitsbereich innerhalb des Wissensraumes auf je einen Gruppen-Tuple Space an. In diesem werden alle gemeinsamen Objekte einer Kooperationsgruppe gespeichert. Die Struktur des Gruppenbereichs mit den enthaltenen Unterarealen und Dokumenten wird dabei streng objektorientiert als Attribute der korrespondierenden Objekte im Tuple Space abgelegt. Dieses Vorgehen macht die Struktur des Wissensraums in der verteilten Umgebung persistent und sorgt automatisch für eine Verteilung sämtlicher Änderungen in der Struktur des gemeinsamen Wissensraums an alle Gruppenmitglieder über die Synchronisations- und Ereignismechanismen des Gruppen-Tuple Space. So gewährleistet diese zusätzliche Schicht die erprobten kooperationsförderlichen Konzepte des virtuellen Wissensraums und gewährleistet zudem die Synchronizität all seiner verteilten Instanzen.

Alle für die Kooperation relevanten Daten, inklusive Benutzer- und Verwaltungsinformationen, werden in dem gemeinsamen Datenraum abgelegt und so allen Teilnehmern zugänglich gemacht. Da die Schicht des mobil-verteilten Wissensraumes die Objekte aus der Persistenz gemäß der Wissensraum-Metapher aufbereitet, kann eine Anwendung für eine mobil-verteilte Kooperation im virtuellen Wissensraum wie gewohnt auf den gemeinsamen Arbeitsraum der Gruppe zugreifen und sich alle enthaltenen Dokumente, Unterareale und anwesenden Benutzer anzeigen lassen. Zudem kann die Gruppenstruktur des mobil-verteilten Wissensraum dank der Nutzung der Gruppenmetapher in den Speicherstrukturen des Gruppen-Tuple Space intuitiv auf diesen abgebildet werden.

Eine weitere Möglichkeit der Schicht des virtuellen Wissensraums liegt in der automatischen Auflösung von vielen Konflikten in der Struktur des Wissensraums. Da die Semantik dieser Struktur bekannt ist, kann die Kooperationsumgebung entscheiden, ob ein Konflikt zu Fehlern in der Semantik oder der Struktur des virtuellen Wissensraums führen würde und entsprechende Gegenmaßnahmen einleiten. Diese Teilautomatisierung der Konfliktauflösung erlaubt eine Entlastung der Benutzer von

trivialen Konflikten oder solchen, die im technischen Umfeld der Kooperationsumgebung verankert sind.

Die Benutzer erreichen nur noch Konflikte innerhalb des verteilten virtuellen Wissensraums, deren Semantik der Kooperationsschicht nicht bekannt ist oder nicht mit Hilfe dieses Wissens entschieden werden können. Dies ist z. B. bei dem Inhalt von vielen Dokumenten der Fall. Da deren inhaltliche Struktur oft ebenfalls bestimmten Struktureigenschaften entspricht, werden ungelöste Konflikte zunächst an die zugehörigen Anwendungen übergeben. Erst wenn auch diese nicht in der Lage ist, den Konflikt zu lösen wird er an die Benutzer weitergereicht. So ergibt sich eine Lösungskette für die Behebung der Konflikte innerhalb des mobil-verteilten Datenraumes, in die der Benutzer nur noch eingreifen muss, wenn keines der betroffenen Glieder den Konflikt im Hintergrund lösen kann.

Durch die hier vorgestellte voneinander gekapselte und doch ineinander verzahnte Schichtarchitektur wird ein umfassendes Architekturkonzept bereitgestellt, das die Anforderungen einer Kooperation in mobil-spontanen aber auch weiträumig-verteilten Nutzungsszenarien unterstützt. Der virtuelle Wissensraum wird hierbei in all seinen Facetten von einer zentralisierten Architektur in ein mobiles und verteiltes Umfeld transportiert, ohne die Errungenschaften der klassischen Kooperationsumgebungen zu ignorieren. Die hier vorgestellte Architektur erlaubt zusätzlich die Integration zentraler Persistenzdienste und anderer externer Ressourcen in die Kooperationsumgebung. Dabei wird – wo möglich – auf existierende und offene Standards gesetzt, die notfalls um benötigte Eigenschaften erweitert werden. Dieses Vorgehen soll auch zukünftig eine leichte Integration der neuen mobil-verteilten Kooperationsumgebung in existierende Infrastrukturen erlauben.

Mobilität in der kooperativen Wissensarbeit

6 Schluss und Ausblick

Im Anschluss an den Entwurf der Musterarchitektur mobil-verteilter Wissensräume gilt es Fragen bezüglich der erreichten Ziele und der neuen Möglichkeiten einer solchen Kooperationsinfrastruktur zu beantworten. Dies schließt eine Bewertung der technischen und konzeptionellen Lösungen mit ein. Die Perspektiven für die kooperative Wissensstrukturierung, die sich aus einer flexiblen und ortsungebundenen Zusammenarbeit in mobil-verteilten Wissensräumen ergeben, sind ebenso zu betrachten, wie die aus dieser Arbeit resultierenden zukünftigen Arbeitsschritte und Forschungsperspektiven.

Den Ausgangspunkt der Arbeit bildeten die unverbundenen Entwicklungsstränge der kooperativen Wissensstrukturierung und der mobilen Ad-Hoc-Netzwerke. Trotz der wünschenswerten Nutzungskonstellation einer kooperativen Wissensstrukturierung in mobilen Alltagssituationen fehlen geeignete Werkzeuge für eine längerfristige mobile und computergestützte Zusammenarbeit. Es existierten kaum Übergänge zwischen beiden Forschungsfeldern und die jeweiligen Nutzungsszenarien bewegten sich stets streng in einem der beiden Bereiche. Während Nutzungsszenarien der computergestützten Kooperation auf die Zusammenarbeit in festen Infrastrukturen zielten, behandelten Nutzungsszenarien der mobilen Vernetzung zumeist Aspekte mobilitätsfreundlicher Netzwerkstrukturen. Nutzungsszenarien die beide Felder in ihrem Wechselspiel betrachten waren nahezu unerforscht.

Da aber ein Systementwurf für eine mobilitätsfreundliche Kooperationsumgebung ohne eine Kenntnis der Nutzungsszenarien und strukturellen Grenzen kaum möglich ist, war ein erstes Ziel dieser Arbeit die Entwicklung entsprechender Nutzungsszenarien kooperativer Wissensstrukturierung. Erst mit Kenntnis der Nutzungsszenarien ist ein Transfer der Welt der kooperativen Wissensstrukturierung in die der mobilen Ad-Hoc-Netzwerke möglich. Diese Nutzungsszenarien wurden für eine Differenzierung der Anforderungsebenen in funktionale und technische Elemente getrennt, die für eine begriffliche Klärung als „*Handlungsebene*“ und „*Umsetzungsebene*“ bezeichnet wurden.

Als Fundament für die funktionalen Komponenten der Kooperationsumgebung wurde das Konzept der *virtuellen Wissensräume* gewählt. Da mobile Kooperationsszenarien durch eine freie und flexible Zusammenarbeit charakterisiert sind, bieten die virtuellen Wissensräume als Verkörperung freier dokumentenzentrierter Gruppenarbeit ein entsprechend flexibles Kooperationskonzept. Der Transfer der virtuellen Wissensräume in mobile Ad-Hoc-Netzwerke war somit unter dem Begriff der „*mobil-verteilten Wissensräume*“ zentrales Thema dieser Arbeit.

Als Basis für die Betrachtung der benötigten Kooperationsunterstützung in einem mobilen Nutzungsumfeld dienten die in Kapitel 2 aufgestellten Nutzungsszenarien.

Sie orientieren sich sowohl an Formen der *Präsenzkooperation*, die stark der natürlichen Zusammenarbeit im mobilen Umfeld entsprechen, als auch an Kooperationsformen, die eine *Einbeziehung räumlich entfernter Benutzer und Dienste* berücksichtigen. Letztere schließen auch die optionale Integration zentralisierter CSCW-Strukturen in mobil-verteilte Kooperationsumgebungen mit ein. Die entwickelten Nutzungsszenarien entstammen je zur Hälfte der Arbeitswelt, mit stärker formalisierten Kooperationsprozessen, und der Lernwelt, mit eher freien konstruktivistischen Kooperationsansätzen, um ein möglichst breites Spektrum von Kooperationsformen abzudecken.

Für ein besseres Verständnis der Vorgänge in mobilen Kooperationsszenarien wurde in Kapitel 3 ein *Drei-Phasen-Modell* mobil-verteilter Kooperation aufgestellt. Neben den Phasen der *Gründung* und *Kooperation*, die auch in klassischen Kooperationszenarien Beachtung finden, wurde der Phase der *Auflösung* ein hoher Stellenwert eingeräumt. Diese in klassischen Kooperationsszenarien vernachlässigte Phase trägt der Tatsache Rechnung, dass in einem mobilen Umfeld eine Wiederaufnahme der Kooperation in derselben Nutzungskonstellation aufgrund der Mobilität der Nutzer nur selten gelingt. Durch den nahtlosen Übergang zwischen den einzelnen Phasen ist nach dem Ausscheiden aus einer Kooperationssitzung (*Auflösung*) jederzeit ein Wiedereintreten in diese – auch in einer geänderten Nutzungskonstellation – möglich (*Gründung*).

Aufbauend auf die Handlungsmuster der entworfenen Szenarien mobiler Kooperation wurden eine Reihe von technischen Anforderungen an mobil-verteilte Kooperationsumgebungen benannt. In der Arbeit wurden wesentliche Problemstellungen aus den Bereichen der „*Vernetzung und Sichtbarkeit*“ der Kooperationspartner, der „*Kontextualisierung*“ der Kooperationsumgebung und der „*Konsistenz*“ der gemeinsam genutzten und redundant gespeicherten Kooperationsobjekte identifiziert.

Die Forderung aus den Bereichen „*Vernetzung und Sichtbarkeit*“ ist insbesondere in der *Gründungsphase* von zentraler Bedeutung. Eine Wahrnehmung potentieller Kooperationspartner und verfügbarer Dienste ist, ebenso wie eine Kommunikationsinfrastruktur zwischen diesen, die Grundlage für die Bildung einer Kooperationsgruppe. Doch auch für die späteren Phasen im Kooperationsprozess benennt diese Forderung das Mindestmaß technischer Unterstützung für die mobile Zusammenarbeit.

Eng verwandt mit der *Vernetzung und Sichtbarkeit* sind die Forderungen aus dem Bereich der *Kontextualisierung*. Sie erlauben insbesondere in komplexeren Konfigurationen und weiträumigen Strukturen die Auswahl geeigneter Kooperationspartner. Das Wissen um den eigenen technischen wie sozialen Kontext und den der Kooperationspartner unterstützt die Konfiguration der Kooperationsprozesse, die nach festgesetzten Regeln auch automatisch erfolgen kann. Der Nutzung von Kontextinformationen fällt daher besonders in der *Gründungsphase* eine hohe Bedeutung zu.

Unter dem Begriff „*Offline-Verfügbarkeit*“ wurde die Problemstellung einer durchgängigen Verfügbarkeit der Kooperationsdaten zusammengefasst. Für die Unterstützung mobil-verteilter Nutzungskonstellationen wurden die *Distribution* und *Repli-*

kation als zentrale Forderungen für eine durchgängige Verfügbarkeit der gemeinsam genutzten Kooperationsdaten auch in unverbundenen Nutzungssituationen herausgearbeitet.

Die sich aus einer Replikation ergebenen Forschungsfragen bezüglich einer redundanten Speicherung der Kooperationsdaten wurden unter dem Oberbegriff der „*Konsistenz*“ gesammelt. Bei einer derartigen redundanten Speicherung der Kooperationsobjekte kommt der „*Synchronizität*“ der gemeinschaftlich bearbeiteten Daten eine hohe Bedeutung zu, wenn ein Divergieren der gemeinsamen Datenbestände verhindert werden soll. Ist die Synchronizität, z. B. aufgrund längeren unverbundenen Arbeitens, nicht mehr gewahrt, sind zudem Mechanismen zur kooperativen Konfliktlösung und zur Rücknahme und Anpassung der Änderungen bereitzustellen (*Reversibilität*).

Das neue mobile Nutzungsumfeld und die daraus begründete verteilte Infrastruktur macht auch die Anpassung der aus zentralisierten Kooperationssystemen bekannten Mechanismen zur Kooperationsunterstützung notwendig. Die betroffenen Bereiche wurden in *Kommunikation*, *Koordination* und *Kooperation* unterteilt und bilden die Basis für die neuartige Zusammenarbeit in mobilen und spontanen Nutzungsszenarien, die im Rahmen dieser Arbeit mit dem Begriff der „*Kollaboration*“ belegt wurde. Die Einbettung dieser klassischen Kooperationsfunktionen in ein dezentralisiertes und mobiles Kooperationsumfeld wurde als eine der wichtigen Anforderungen identifiziert.

Anhand der in Kapitel 3 ermittelten Anforderungen an die mobil-verteilte Kooperationsumgebung erfolgte in Kapitel 4 eine Bestandsaufnahme verfügbarer Lösungsansätze. Aufgrund der herausgearbeiteten zentralen Stellung der Persistenz für eine Bereitstellung mobil-verteilter Wissensräume lag der Fokus auf den Technologien der mobilitätsfreundlichen Vernetzung und der verteilten Kommunikations- und Persistenzsysteme. Aus ihnen wurde das technische Grundgerüst der Musterarchitektur mobil-verteilter Wissensräume gebildet.

Die durchgeführte Betrachtung der Technologien für eine unabhängige Vernetzung mobiler Geräte jenseits der Zugänge existierender Netzwerkinfrastrukturen ergab die Verfügbarkeit einer Vielzahl von Lösungsansätzen aus dem Forschungsfeld der *mobilen Ad-Hoc-Netzwerke (Manets)*. Es zeigte sich aber, dass diese noch keinen Einzug in den Alltag der Benutzer gefunden haben. Dennoch wurden sie unter der Annahme eines zukünftigen Alltagseinsatzes als Basis mobil-spontaner Vernetzung betrachtet. Protokolle dieses Ursprungs zielen auf eine selbstorganisierende, spontane und robuste Vernetzung mobiler Netzwerknoten und etablieren auch weiträumige Kommunikationsinfrastrukturen. Über eine Hybridlösung mit *Mobile IP* ist darüber hinaus auch eine Einbindung etablierter Netzwerkstrukturen wie das Internet möglich. In Kombination mit ebenfalls verfügbaren Mechanismen zur automatischen Dienstekonfiguration und -verknüpfung kann somit eine technische Infrastruktur zur Initialisierung einer mobil-spontanen Kooperation bereitgestellt werden.

Ein besonderes Bild bot sich bei der Betrachtung verteilter Persistenzsysteme. Zwar sind durch das wachsende Interesse an unabhängigen Peer-to-Peer-Strukturen eine große Anzahl verteilter Persistenzsysteme verfügbar, aber eine nähere Betrach-

tung offenbarte, dass diese für mobil-spontane Kooperationsszenarien wenig geeignet sind. Diese Feststellung begründete sich zum Teil in fehlenden Konzepten für eine *Offline-Verfügbarkeit* gemeinsam genutzter Daten oder einer mangelhaften Wahrung der *Konsistenz* derselben.

Existierende verteilte Persistenzsysteme bewerten einen unverbundenen Knoten stets als defekt und beachten bei der Fehlerbehandlung nur die Funktionalität des Gesamtsystems. Die wichtige Unterstützung des isolierten Knotens in der Phase der unverbundenen Arbeit fehlt zumeist. Gerade hier liegt aber die Besonderheit mobiler Kooperationsszenarien, in denen Verbindungsabbrüche häufig vorkommen. Die Kooperationspartner wollen in der Regel auch bei Isolation von der Gruppe auf den gemeinsamen Kooperationsobjekten weiterarbeiten können.

Jenseits der DHT-basierten Persistenzsysteme bietet das Konzept der *Tuple Spaces* einen einfachen und zeitlich entkoppelten Zugriff auf die gespeicherten Daten. Tuple Spaces erlauben Knoten, die nicht zeitgleich mit dem Netzwerk verbunden sind, dennoch Daten auszutauschen. Diese Mechanismen lassen sich auch für eine Koordination der Kooperationsprozesse in mobilen Nutzungskonstellationen verwenden. Aufgrund dieser interessanten Eigenschaften wurden sie ebenfalls im Rahmen dieser Arbeit betrachtet und zusammen mit den DHTs als mögliche Bausteine einer Architektur für mobil-verteilte Wissensräume herangezogen.

Im Anschluss der eingehenden Analyse der konzeptionellen wie technischen Möglichkeiten und Defizite der verfügbaren verteilten Persistenzsysteme in Kapitel 4 wurden die vielversprechenden Ansätze der DHTs und Tuple Spaces in Kapitel 5 als Bausteine für eine Musterarchitektur mobil-verteilter Wissensräume herangezogen und um fehlende Komponenten ergänzt. Ziel dieser Musterarchitektur war es, die Möglichkeit einer Umsetzung mobil-verteilter Wissensräume mittels verfügbarer Technologien aufzuzeigen und die identifizierten Lücken mit speziell an die Bedürfnisse mobiler Kooperationsszenarien angepassten Konzepten zu schließen.

Die Familie der Distributed Hashtables (DHT) wurde anhand des zuvor ermittelten Anforderungskatalogs um Mechanismen für eine Offline-Verfügbarkeit der Kooperationsdaten zu einer mobilitätsfreundlichen verteilten Persistenzschicht ergänzt. Kern dieser Persistenzschicht ist eine gezielte Distributions- und Replikationsstrategie, die zusammen mit der Versionierung der redundant gespeicherten Objekte eine Offline-Verfügbarkeit der Kooperationsobjekte bei Beibehaltung der geforderten Konsistenz erlaubt. Zusätzlich berücksichtigt das Replikationskonzept die unterschiedlichen Ausbauformen mobiler Geräte und ermöglicht über differenzierte Replikationsklassen auch ressourcenarmen Geräten einen Zugriff auf die verteilte Persistenz.

Konflikte bei einem konkurrierenden Zugriff auf die gemeinsam bearbeiteten Objekte werden durch die integrierte Versionskontrolle zeitnah aufgedeckt und den Benutzern angezeigt. Zusätzlich können über die Versionierung und die Konfliktlösungsmechanismen die betroffenen Objekte kooperativ wieder in Einklang gebracht werden.

Dank der gezielten Distributions- und Replikationsstrategie ist eine Unterstützung mobil-spontaner Kooperation sowohl im dynamischen Verbund mit Koope-

rationspartnern als auch in isolierten Kooperationssituationen gewährleistet. Diese im Rahmen der vorliegenden Arbeit entworfene mobil-verteilte Persistenz wurde durch eine *Abstraktionsschicht* um die Schlüsselkomponenten einer *logischen Replikationsstruktur*, eines *transparenten Objektzugriffs* und einer *zeitlich entkoppelten Kommunikation* zwischen zeitweise unverbundenen Kooperationspartnern ergänzt. War die Nutzung der mobil-verteilten Persistenz bis zu diesem Punkt noch an ein Wissen über die Netzwerkstruktur gebunden, kann nun dank des anonymen Speicherkonzepts des *Tuple Space* auf die verteilten Objekte mit einfachen Operationen zugegriffen werden.

Mit dem Konzept der *Gruppen-Tuple Spaces* wurde der mobil-verteilten Persistenzschicht eine *gruppenbasierte Replikationsmetapher* hinzugefügt. Diese erlaubt eine intuitive Replikation der gemeinsam genutzten Objekte zwischen den Gruppenmitgliedern. Die Versionierung und Synchronisierung der Repliken kann völlig im Hintergrund geschehen und reduziert so die Komplexität des Zugriffs auf den verteilten Speicher in einem erheblichen Maße. Auch die asynchrone Kommunikation und Ereigniskontrolle zwischen den sporadisch verbundenen Knoten wird von den Gruppen-Tuple Spaces übernommen. Über die enthaltene zeitlich entkoppelte Nachrichtenverbreitung werden Nachrichten nach einem Verbindungsabbruch zugestellt, sobald der Zielknoten wieder erreichbar ist. So ist jeder Knoten nach einer verbindungslosen Phase bei Wiedereintritt in den Kooperationsverbund auf dem neuesten Stand.

Die eigentliche Kooperationsumgebung bilden die *mobil-verteilten Wissensräume*, die auf diese technische Abstraktionsschicht aufgesetzt wurden. Jeder Kooperationsgruppe im virtuellen Wissensraum wird ein *eigener Gruppen-Tuple Space* in der Persistenzschicht zugeordnet. In diesem werden alle die Kooperation betreffenden Objekte abgelegt und so automatisch zwischen den Knoten der Gruppenmitglieder repliziert. Jedes Gruppenmitglied verfügt dadurch lokal über den gesamten gemeinsamen Wissensraum der Gruppe. Auf diese Weise stehen die gemeinsamen Kooperationsobjekte auch in Offline-Situationen zur Verfügung. Der gemeinsame Wissensraum wird bei Kontakt zur Gruppe automatisch mit den Repliken der anderen verfügbaren Gruppenmitglieder abgeglichen.

Die so geschaffene Kooperationsumgebung stellt persistente virtuelle Wissensräume mit ihren bewährten Kooperationsmechanismen inklusive der *primären Medienfunktionen* bereit. Durch die durchgängige Verfügbarkeit der Kooperationsobjekte beugt sie zudem Medienbrüchen vor, die aus der Mobilität der Nutzer zu entstehen drohen.

Das so entstandene Vier-Schicht-Modell bildet mit der *Kommunikationsschicht*, der *Objektverwaltung und -distribution*, der *Abstraktionsschicht* und den *mobil-verteilten Wissensräumen* eine mögliche Realisierung technischer Infrastrukturen für die mobile Wissensstrukturierung gemäß den ermittelten Anforderungen. Dabei wird durch die Kapselung der einzelnen Schichten ein modularer Aufbau der Komponenten gewahrt. Dies erlaubt eine einfache Anpassung der Kooperationsumgebung an ein variables Nutzungsumfeld.

Durch die Einbettung von mobilen Ad-Hoc-Netzwerken zur spontanen Vernetzung sowie den Einsatz innovativer Lösungen für eine automatische Dienstekonfiguration, wird eine flexible und mobilitätsfreundliche Kommunikationsinfrastruktur für die Kooperationsumgebung errichtet. Die besondere Berücksichtigung einer geeigneten *Distributions- und Replikationsstrategie* in der darauf aufsetzenden *Persistenzschicht* sorgt für die benötigte *Offline-Verfügbarkeit* und *Konsistenz* der gemeinsam genutzten Kooperationsobjekte. Mittels der in der Abstraktionsschicht eingesetzten *Gruppen-Tuple Spaces* wird zudem ein anonymer Speicherzugriff in Kombination mit einer intuitiven gruppenbasierten Replikation bereitgestellt sowie die Komplexität der Persistenzschicht verbirgt. Zusammen mit der ebenfalls in der Abstraktionsschicht angesiedelten *zeitlich und räumlich entkoppelten Kommunikation* und den *Mechanismen zur Integration externer Dienste* entsteht so der technischer Rahmen für die mobile Kooperationsumgebung. Die oberste Schicht der *mobil-verteilten Wissensräume* verbindet konsequent das bewährte Konzept der virtuellen Wissensräume mit den technischen Lösungen einer mobil-verteilten Persistenz. Die Kooperationspartner erhalten so in allen Situationen ihrer Mobilität Zugriff auf ihre gemeinsam bearbeiteten Materialien, ohne auf die Flexibilität einer dokumentenzentrierten Kooperation in virtuellen Wissensräumen verzichten zu müssen.

Aus dem zunächst unübersichtlichen Problemfeld mobiler Kooperation wurde in der vorliegenden Arbeit die Bereitstellung einer verteilten Persistenz für eine durchgängige Verfügbarkeit der Kooperationsobjekte als Kernproblem identifiziert. Die entworfene Musterarchitektur zeigt entsprechende Wege für die Verknüpfung einer verteilten Persistenz mit weiteren Komponenten zu einer mobil-verteilten Kooperationsumgebung auf. Im Rahmen der Arbeit wurden zudem wesentliche Komponenten der Musterarchitektur aus den Bereich der Vernetzung und Sichtbarkeit, der Kontextualisierung und der Distribution und Replikation prototypisch umgesetzt und erprobt.

Die so entstandene Musterarchitektur offenbart wesentliche Forschungsperspektiven, die sich durch eine Bereitstellung mobilitätsfreundlicher Kooperationsstrukturen ergeben. Die zukünftigen Forschungsfragen bewegen sich dabei in den Bereichen des Einsatzes virtueller Wissensräume in mobilen Kooperationsszenarien, der Sicherheit und Identitätskontrolle in offenen verteilten Kooperationssystemen, der erweiterten Kontextualisierung entlang der Kooperationsprozesse und der Untersuchung alternativer Distributions- und Replikationsstrategien. Im Folgenden sollen diese offenen Forschungsfragen kurz angesprochen werden.

Die in serverzentrierten Umgebungen bewährten Konzepte der virtuellen Wissensräume können mittels neuartiger mobiler Infrastrukturen an einem mobilen Nutzungsumfeld gemessen und verfeinert werden. Insbesondere kann die Bereitstellung mobil-verteilter Wissensräume ausschlaggebend für die Überprüfung existierender und die Identifizierung weitergehender primärer Medienfunktionen sein.

Derart freie Kooperationsumgebungen wie die hier entworfenen mobil-verteilten Wissensräume bedürfen einer Integration dezentral gesteuerter Mechanismen zur Verschlüsselung und Identitätskontrolle. Diese verteilt verwalteten Sicherheitsmechanismen müssen ähnlich intuitiv nutzbar sein wie die vorgestellten Replikations-

strategien. Darum sind neben technischen Erwägungen auch Fragestellungen auf der Handlungsebene zu berücksichtigen.

Die als wichtige Komponente mobiler Kooperation identifizierte *Kontextualisierung der Kooperationsumgebung* wirft neue Fragestellungen im technischen wie konzeptionellen Bereich des Forschungsfeldes der *Context Awareness* auf. Dazu gehören ebenso die besprochenen automatisierten Mechanismen der Gruppengründung anhand eines Ortsbezugs, wie auch die beiläufige Strukturierung des Wissensraums aus dem Kooperationskontext heraus. Des Weiteren müssen Kontextinformationen aufbereitet werden, um die Übersichtlichkeit der Kooperationsumgebung nicht zu gefährden. Besonders in stark mit Kontextinformationen angereicherten Umgebungen wird es notwendig sein, nur bestimmte Kontextinformationen für den Benutzer zugänglich zu machen. Die zuverlässige Differenzierung von „interessanten“ und „uninteressanten“ Kontextinformationen stellt die Entwickler somit vor neue Herausforderungen.

Eine eingehendere Betrachtung verdient auch die Fragestellung nach weiterführenden Distributions- und Replikationsstrategien für spezielle mobile Nutzungsszenarien. Die im Rahmen dieser Arbeit entworfene Replikationsstrategie anhand der Gruppenstruktur ist ein universelles Mittel zur konsistenten Verknüpfung von Kooperationspartnern in der Gruppenarbeit. Alternative Ansätze könnten sich aber als besser an bestimmte Kooperationsszenarien angepasst erweisen. So wäre z. B. eine an die verfügbaren Bandbreiten zwischen den vernetzen Gruppenknoten angepasste Replikationsstrategie denkbar. Auch eine umfassende Einbindung des Kooperationskontextes in die Objektverteilung und -replikation kann sich als probates Mittel für eine Steigerung der Offline-Verfügbarkeit der Kooperationsdaten erweisen.

Die aufgeworfenen zukünftigen Forschungsfragen, die sich durch die hier entworfene Musterarchitektur mobil-verteilter Wissensräume ergeben, heben die Relevanz der Erforschung mobiler Kooperationsumgebungen hervor. Erst eine mobile Kooperationsumgebung, die unabhängig von bestehenden Infrastrukturen funktioniert und sich dennoch in diese integriert, vermag sich im Alltag der Benutzer zu etablieren. Die in dieser Arbeit entworfene Musterarchitektur zeigt einen Weg auf, dieses Ziel durch den Einsatz verfügbarer Technologien und innovativer Konzepte zu erreichen. Die bis dahin fehlenden Bausteine wurden auf die Bedürfnisse mobiler Kooperationszenarien ausgerichtet und fußen auf anerkannte Konzepte und Technologien.

Der im Rahmen dieser Arbeit geleistete Transfer virtueller Wissensräume in das Spannungsfeld mobil-spontaner Netzwerkumgebungen zeigt somit einen Weg auf, mobile Wissensstrukturierung im Einklang mit existierenden Kooperationsinfrastrukturen zu etablieren.

Mobilität in der kooperativen Wissensarbeit

Literaturverzeichnis

Anderson und Shasha 1991

ANDERSON, Brian G. ; SHASHA, Dennis: Persistent Linda: Linda + Transactions + Query Processing. In: BANATRE, J.P. (Hrsg.) ; LE METAYER, D. (Hrsg.): *Research Directions in High-Level Parallel Programming Languages*. Mont-Saint-Michel, France : Springer Verlag, 1991, S. 93–109

Balakrishnan et al. 2003

BALAKRISHNAN, Hari ; KAASHOEK, M. Frans ; KARGER, David ; MORRIS, Robert ; STOICA, Ion: Looking up data in P2P systems. In: *Communications of the ACM* 46 (2003), Nr. 2, S. 43–48

Bentley et al. 1997

BENTLEY, Richard ; APPELT, Wolfgang ; BUSBACH, Uwe ; HINRICHES, Elke ; KERR, David ; SIKKEL, Klaas ; TREVOR, Jonathan ; WOETZEL, Gerd: Basic support for cooperative work on the World Wide Web. In: *International Journal of Human Computer Studies* 46 (1997), Nr. 6, S. 827–846

Bleckmann et al. 2005

BLECKMANN, Peter ; SPROTTE, Renè ; ESSMANN, Bernd ; HAMPEL, Thorsten: Interactive Learning Objects in Mobile E-Learning. In: *Proceedings of the World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education 2005 (E-Learn 2005)*, AACE, 2005, S. 2809–2816

Booch et al. 1998

BOOCH, Grady ; RUMBAUGH, James ; JACOBSON, Ivar: *The Unified Modeling Language User Guide*. Addison-Wesley Professional, 1998

Bopp und Hampel 2005

BOPP, Thomas ; HAMPEL, Thorsten: A Microkernel Architecture for Distributed Mobile Environments. In: CHEN, Chin-Sheng (Hrsg.) ; FILIPE, Joaquim (Hrsg.) ; SERUCA, Isabel (Hrsg.) ; CORDEIRO, José (Hrsg.): *Proceedings of the Seventh International Conference on Enterprise Information Systems (ICEIS 2005)* Bd. 4, 2005, S. 151–156

Broch et al. 1998

BROCH, Josh ; MALTZ, David A. ; JOHNSON, David B. ; HU, Yih-Chun ; JET-CHEVA, Jorjeta: A performance comparison of multi-hop wireless ad hoc network routing protocols. In: *Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking (MobiCom 1998)*, ACM Press, 1998, S. 85–97

Brookshier et al. 2002

BROOKSHIER, Daniel ; GOVONI, Darren ; KRISHNA, Navaneeth: *JXTA: Java P2P Programming*. SAMS, 2002

Buschmann et al. 1996

BUSCHMANN, Frank ; MEUNIER, Regine ; ROHNERT, Hans ; SOMMERLAD, Peter ; STAL, Michael: *Pattern-Oriented Software Architecture – A System of Patterns*. John-Wiley and Sons, 1996

Campbell 1998

CAMPBELL, Richard: *Managing AFS: The Andrew File System*. Prentice Hall, 1998

Campell et al. 2002

CAMPELL, Andrew T. ; GOMEZ, Javier ; KIM, Sanghyo ; WAN, Chieh-Yih ; TURANAYI, Zoltan R. ; VALKO, Andras G.: Comparison of IP micromobility protocols. In: *IEEE Wireless Communications* 9 (2002), S. 72–82

Castro et al. 2003

CASTRO, Miguel ; DRUSCHEL, Peter ; KERMARREC, Anne-Marie ; NANDI, Animesh ; ROWSTRON, Antony ; SINGH, Atul: SplitStream: High-bandwidth multicast in a cooperative environment. In: *Proceedings of the 19th ACM Symposium on Operating Systems Principles (SOSP 2003)*, 2003, S. 298–313

Castro et al. 2002

CASTRO, Miguel ; DRUSCHEL, Peter ; KERMARREC, Anne-Marie ; ROWSTRON, Antony: Scribe: A Large-Scale and Decentralized Application-Level Multicast Infrastructure. In: *IEEE Journal on Selected Areas in Communications (JSAC)* 20 (2002), Nr. 8, S. 1489–1499

Cuce und Zaslavsky 2003

CUCE, Simon ; ZASLAVSKY, Arkady: Supporting multiple consistency models within a mobility enabled file system using a component based framework. In: *Mobile Networks and Applications* 8 (2003), Nr. 4, S. 317–326

Dabek et al. 2001

DABEK, Frank ; KAASHOEK, M. Frans ; KARGER, David ; MORRIS, Robert ; STOICA, Ion: Wide-area cooperative storage with CFS. In: *Proceedings of the 18th ACM Symposium on Operating Systems Principles (SOSP 2001)*, ACM Press, 2001, S. 202–215

Dabek et al. 2003

DABEK, Frank ; ZHAO, Ben Y. ; DRUSCHEL, Peter ; KUBIATOWICZ, John ; STOICA, Ion: Towards a Common API for Structured P2P Overlays. In: KAASHOEK, M. Frans (Hrsg.) ; STOICA, Ion (Hrsg.): *Peer-to-Peer Systems II. Proceedings of the Second International Workshop on Peer-to-Peer Systems (IPTPS 2003)*. Springer Verlag, 2003, S. 33–44

Dourish und Bellotti 1992

DOURISH, Paul ; BELLOTTI, Victoria: Awareness and coordination in shared workspaces. In: *Proceedings of the 1992 ACM Conference on Computer-Supported Cooperative Work (CSCW 1992)*, ACM Press, 1992, S. 107–114

Druschel und Rowstron 2001

DRUSCHEL, Peter ; ROWSTRON, Antony: PAST: A Large-Scale, Persistent Peer-to-Peer Storage Utility. In: *Proceedings of the Eighth Workshop on Hot Topics in Operating Systems (HOTOS 2001)*, IEEE Computer Society, 2001, S. 75–80

Dwyer und Bharghavan 1997

DWYER, Dane ; BHARGHAVAN, Vaduvur: A Mobility-Aware File System for Partially Connected Operation. In: *ACM SIGOPS Operating Systems Review* 31 (1997), Nr. 1, S. 24–30

Dyer und Boppana 2001

DYER, Thomas D. ; BOPPANA, Rajendra V.: A comparison of TCP performance over three routing protocols for mobile ad hoc networks. In: *Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing (MobiHoc 2001)*, ACM Press, 2001, S. 56–66

Edwards et al. 1997

EDWARDS, W. Keith ; MYNATT, Elizabeth D. ; PETERSEN, Karin ; SPREITZER, Mike J. ; TERRY, Douglas B. ; THEIMER, Marvin M.: Designing and Implementing Asynchronous Collaborative Applications With Bayou. In: *Proceedings of the 10th annual ACM symposium on User Interface Software and Technology (UIST 1997)*, ACM Press, 1997, S. 119–128

Edwards et al. 2002a

EDWARDS, W. Keith ; NEWMAN, Mark W. ; SEDIVY, Jana Z. ; SMITH, Trevor F. ; BALFANZ, Dirk ; SMETTERS, D. K. ; WONG, H. C. ; IZADI, Shahram: Using Speakeasy for Ad Hoc Peer-to-Peer Collaboration. In: *Proceedings of the Conference on Computer Supported Cooperative Work (CSCW'02)*, ACM Press, 2002, S. 256–265

Edwards et al. 2002b

EDWARDS, W. Keith ; NEWMAN, Mark W. ; SEDIVY, Jana Z. ; SMITH, Trevor F. ; IZADI, Shahram: Challenge: Recombinant Computing and the Speakeasy Approach. In: *Proceedings of the International Conference on Mobile Computing and Networking (MOBICOM'02)*, ACM Press, 2002, S. 279–286

Elsässer et al. 2001

ELSÄSSER, Robert ; KRALOVI, Rastislav ; MONIEN, Burkhard: Scalable Sparse Topologies with Small Spectrum. In: *Proceedings of the 18th Annual Symposium on Theoretical Aspects of Computer Science (STACS 2001)* Bd. 2010, Springer Verlag, 2001, S. 218–229

Eßmann et al. 2005

ESSMANN, Bernd ; BLECKMANN, Peter ; HAMPEL, Thorsten ; ELSÄSSER, Robert: Distributed Persistence in CSCW Applications / Heinz Nixdorf Institute, Universität Paderborn. 2005 (tr-ri-05-268). – Technical Report

Eßmann und Funke 2005

ESSMANN, Bernd ; FUNKE, Holger: Providing Peer-to-Peer Features to Existing Client-Server CSCW Systems. In: CHEN, Chi-Sheng (Hrsg.) ; FILIPE, Joaquim (Hrsg.) ; SERUCA, Isabel (Hrsg.) ; CORDEIRO, José (Hrsg.): *Proceedings of the Proceedings of the Seventh International Conference On Enterprise Information Systems (ICEIS 2005)* Bd. 4, INSTICC, 2005, S. 271–274

Eßmann und Hampel 2003a

ESSMANN, Bernd ; HAMPEL, Thorsten: Human Computer Interaction and Co-operative Learning in Mobile Environments. In: HARRIS, Don (Hrsg.) ; DUFFY, Vincent (Hrsg.) ; SMITH, Michael (Hrsg.) ; STEPHANIDIS, Constantine (Hrsg.): *Proceedings of the International Conference on Human Computer Interaction (HCII 2003)* Bd. 3, Lawrence Erlbaum Associates, Inc., 2003, S. 694–698

Eßmann und Hampel 2003b

ESSMANN, Bernd ; HAMPEL, Thorsten: Integrating Cooperative Knowledge Spaces into Mobile Environments. In: ROSSETT, Allison (Hrsg.): *Proceedings of the World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education 2003 (E-Learn 2003)*, 2003, S. 225–232

Eßmann und Hampel 2004

ESSMANN, Bernd ; HAMPEL, Thorsten: Collaborative eLearning in Real Places – Deploying Location Awareness for Face-to-Face eLearning Support. In: *Proceedings of the World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education 2004 (E-Learn 2004)* Bd. 1, 2004, S. 2544–2550

Eßmann und Hampel 2005

ESSMANN, Bernd ; HAMPEL, Thorsten: A Framework For Distributed Objects in Peer-To-Peer Cooperation Environments. In: CHEN, Chi-Sheng (Hrsg.) ; FILIPE, Joaquim (Hrsg.) ; SERUCA, Isabel (Hrsg.) ; CORDEIRO, José (Hrsg.): *Proceedings of the Seventh International Conference On Enterprise Information Systems (ICEIS 2005)* Bd. 4, INSTICC, 2005, S. 157–162

Eßmann et al. 2004a

ESSMANN, Bernd ; HAMPEL, Thorsten ; BLECKMANN, Peter ; SPROTTE, René: A Whiteboard at Your Fingertips – Automatic Configuration of e-Learning Services in Heterogeneous Network Environments. In: *Proceedings of the World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education 2004 (E-Learn 2004)* Bd. 1, 2004, S. 2601–2608

Eßmann et al. 2004b

ESSMANN, Bernd ; HAMPEL, Thorsten ; BOPP, Thomas: A Network Component Architecture for Collaboration in Mobile Settings. In: SERUCA, Isabel (Hrsg.) ; FILIPE, Joaquim (Hrsg.) ; HAMMOUDI, Slimane (Hrsg.) ; CORDEIRO, José (Hrsg.): *Proceedings of the Sixth International Conference on Enterprise Information Systems 2004 (ICEIS 2004)* Bd. 4, 2004, S. 337–343

Eßmann et al. 2006a

ESSMANN, Bernd ; HAMPEL, Thorsten ; GÖTZ, Frank: An Open Architecture for Collaborative Visualizations in Rich Media Environments. In: *Proceedings of the Eighth International Conference on Enterprise Information Systems (ICEIS 2006)*, 2006, S. to appear

Eßmann et al. 2006b

ESSMANN, Bernd ; HAMPEL, Thorsten ; GÖTZ, Frank ; ELSNER, Andreas: Embedding Collaborative Visualizations into Virtual Knowledge Spaces. In: *Proceedings of the Seventh International Conference on the Design of Cooperative Systems (COOP 2006)*, 2006, S. to appear

Eßmann et al. 2004c

ESSMANN, Bernd ; HAMPEL, Thorsten ; SLAWIK, Joanna: A JXTA-based Framework for Mobile Cooperation in Distributed Knowledge Spaces. In: KARAGIANIS, Dimitris (Hrsg.) ; REIMER, Ulrich (Hrsg.): *Proceedings of the Fifth International Conference on Practical Aspects of Knowledge Management (PAKM 2004)*, Springer Verlag, 2004, S. 11–22

Eßmann und Hampel 2005a

ESSMANN, Bernd ; HAMPEL, Thorsten: Connectivity – Context – Consistency: Key Factors for Mobility Supporting CSCW/L-Environments. In: *Proceedings of the ConferrenWorld Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education 2005 (E-Learn 2005)*, AACE, 2005, S. 2915–2922

Eßmann und Hampel 2005b

ESSMANN, Bernd ; HAMPEL, Thorsten: A Design Pattern for Mobile-Distributed Knowledge Spaces. In: *Proceedings of the Metainformatics Symposium 2005*, Springer Verlag, 2005, S. to appear

Eßmann et al. 2006

ESSMANN, Bernd ; HAMPEL, Thorsten ; KEIL-SLAWIK, Reinhard: Challenges towards a Distributed Persistence Layer for Next Generation CSCW Applications. In: *Proceedings of the 4th IEEE International Conference on Pervasive Computing and Communications (PerCom 2006). 2nd IEEE International Workshop on PervasivE Learning (PerEL 2006)*, IEEE Computer Society, 2006, S. to appear

Fenwick und Pollock 1997

FENWICK, James B. ; POLLOCK, Lori L.: Issues and Experiences in Implementing a Distributed Tuplespace. In: *Software – Practice and Experience* 27 (1997), Nr. 10, S. 1199–1232

Fox und Brewer 1999

FOX, Armando ; BREWER, Eric A.: Harvest, Yield, and Scalable Tolerant Systems. In: *Proceedings of the The Seventh Workshop on Hot Topics in Operating Systems (HOTOS-VII)*, IEEE Computer Society, 1999, S. 174–178

Freeman et al. 1999

FREEMAN, Eric ; HUPFER, Susanne ; ARNOLD, Ken: *JavaSpaces(TM) Principles, Patterns, and Practice*. Addison-Wesley Professional, 1999

Gamma und Beck 2003

GAMMA, Erich ; BECK, Kent: *Contributing to Eclipse: Principles, Patterns, and Plugins*. Addison-Wesley Professional, 2003

Gamma et al. 1995

GAMMA, Erich ; HELM, Richard ; JOHNSON, Ralph ; VLASSIDES, John: *Design patterns: elements of reusable object-oriented software*. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 1995

Gelernter 1985

GELENTER, David: Generative communication in Linda. In: *ACM Transactions on Programming Languages and Systems (TOPLAS)* 7 (1985), Nr. 1, S. 80–112

Gelernter und Carriero 1989

GELENTER, David ; CARRIERO, Nicholas: Linda in Context. In: *Communications of the ACM* 32 (1989), Nr. 4, S. 444–458

Gong 2001

GONG, Li: JXTA: A network programming environment. In: *IEEE Internet Computing* 5 (2001), Nr. 3, S. 88–95

Götz et al. 2006

GÖTZ, Frank ; DOMIK, Gitta ; ESSMANN, Bernd ; HAMPEL, Thorsten: Collaboration by Illustration: Real-Time Visualization in Web3D. In: *Proceedings of the 11th International Conference on 3D Web Technology (Web3D 2006)*, 2006, S. to appear

Götz et al. 2005

GÖTZ, Frank ; ESSMANN, Bernd ; HAMPEL, Thorsten: Using a Shared Whiteboard for Cooperative Visualization. In: *Proceedings of the HCI International 2005*, 2005

Gray 1988

GRAY, Jim: The transaction concept: virtues and limitations. In: *Readings in database systems*. Morgan Kaufmann Publishers, Inc., 1988, S. 140–150

Gustafsson et al. 2000

GUSTAFSSON, Eva ; JOHNSON, Annika ; PERKINS, Charles E.: Mobile IP Regional Registration / IETF. 2000 (draft-ietf-mobileip-reg-tunnel-03.txt). – Technical Report

Gutknecht et al. 2001

GUTKNECHT, Olivier ; FERBER, Jacques ; MICHEL, Fabien: Integrating tools and infrastructures for generic multi-agent systems. In: *Proceedings of the Fifth international conference on Autonomous agents (AGENTS 2001)*, ACM Press, 2001, S. 441–448

Guttman 2001

GUTTMAN, Erik: Autoconfiguration for IP Networking: Enabling Local Communication. In: *IEEE Internet Computing* 5 (2001), Nr. 3, S. 81–86

Hampel 2001

HAMPEL, Thorsten: *Virtuelle Wissensräume : ein Ansatz für die kooperative Wissensorganisation*. Paderborn, Universität Paderborn, Fachbereich 17 - Informatik, Diss., 2001

Hampel und Eßmann 2004

HAMPEL, Thorsten ; ESSMANN, Bernd: Spatial Structuring of Virtual Knowledge Spaces – a Taxonomy. In: MCKAY, Elspeth (Hrsg.): *Proceedings of the International Conference on Computers in Education (ICCE 2004)*, 2004, S. 577–584

Hampel et al. 2003

HAMPEL, Thorsten ; KEIL-SLAWIK, Reinhard ; ESSMANN, Bernd: Jour Fixe – We Are Structuring Knowledge Collaborative – Structuring of Semantic Spaces as a Didactic Concept and New Form of Cooperative Knowledge Organization Proceedings of E-Learn 2003. In: ROSSETT, Allison (Hrsg.): *Proceedings of the World Conference on E-Learning in Corp., Govt., Health., and Higher Ed. 2004 (E-Learn 2003)*, AACE, 2003, S. 225–232

Hampel und Keil-Slawik 2002

HAMPEL, Thorsten ; KEIL-SLAWIK, Rheinhard: sTeam: Structuring Information in a Team – Distributed Knowledge Management in Cooperative Learning Environments. In: *ACM Journal of Educational Resources in Computing* 1 (2002), Nr. 2, S. 1–27

Hightower und Borriello 2001

HIGHTOWER, Jeffrey ; BORRIELLO, Gaetano: Location systems for ubiquitous computing. In: *IEEE Computer* 34 (2001), Nr. 8, S. 57–66

Holliday et al. 2002

HOLLIDAY, Joanne ; AGRAWAL, Divyakant ; ABBADI, Amr E.: Disconnection Modes for Mobile Databases. In: *Wireless Networks* 8 (2002), Nr. 4, S. 391–402

Howard et al. 1988

HOWARD, John H. ; KAZAR, Michael L. ; MENEES, Sherri G. ; NICHOLS, David A. ; SATYANARAYANAN, M. ; SIDEBOTHAM, Robert N. ; WEST, Michael J.: Scale and performance in a distributed file system. In: *ACM Transactions on Computer Systems (TOCS)* 6 (1988), Nr. 1, S. 51–81

Hubbold et al. 1999

HUBBOLD, Roger ; COOK, Jon ; KEATES, Martin ; GIBSON, Simon ; HOWARD, Toby ; MURTA, Alan ; WEST, Adrian ; PETTIFER, Steve: GNU/MAVERIK: a micro-kernel for large-scale virtual environments. In: *Proceedings of the ACM symposium on Virtual reality software and technology (VRST 1999)*, ACM Press, 1999, S. 66–73

Huck et al. 2002

HUCK, Paul ; BUTLER, Michael ; GUPTA, Amar ; FENG, Michael: A Self-Configuring and Self-Administering Name System with Dynamic Address Assignment. In: *ACM Transactions on Internet technology* 2 (2002), Nr. 1, S. 14–46

Huston und Honeyman 1993

HUSTON, Larry B. ; HONEYMAN, Peter: Disconnected Operation for AFS. In: *Proceedings of the USENIX Mobile and Location-Independent Computing Symposium, 1993*, S. 1–10

Jang 1990

JANG, Jai E.: An optimal fault-tolerant broadcasting algorithm for a hypercube multiprocessor. In: *Proceedings of the 1990 ACM annual conference on Cooperation (CSC '90)*, ACM Press, 1990, S. 96–102

Jeong et al. 2005

JEONG, Jaehoon P. ; PARK, Juangsoo ; KIM, Hyoungjun ; KIM, Dongkyun: Ad Hoc IP Address Autoconfiguration / IETF. 2005 (draft-jeong-adhoc-ip-addr-autoconf-05.txt). – Technical Report

Johnson 1994

JOHNSON, David B.: Routing in Ad Hoc Networks of Mobile Hosts. In: *Proceedings of the Workshop on Mobile Computing Systems and Applications*, IEEE Computer Society, 1994, S. 158–163

Johnson et al. 2004

JOHNSON, David B. ; PERKINS, Charles E. ; ARKKO, Jari: Mobility Support in IPv6 / IETF Network Working Group. 2004 (RFC3775). – Request For Comment

Karbhari et al. 2004

KARBHARI, Pradnya ; AMMAR, Mostafa H. ; DHAMDHERE, Amogh ; RAJ, Himanshu ; RILEY, George F. ; ZEGURA, Ellen W.: Bootstrapping in Gnutella: A Measurement Study. In: BARAKAT, Chadi (Hrsg.) ; PRATT, Ian (Hrsg.): *Proceedings of the Passive and Active Network Measurement, 5th International Workshop (PAM 2004)* Bd. 3015, Springer Verlag, 2004, S. 22–32

Karger et al. 1997

KARGER, David ; LEHMAN, Eric ; LEIGHTON, Tom ; PANIGRAHY, Rina ; LEVINE, Matthew ; LEWIN, Daniel: Consistent hashing and random trees: distributed caching protocols for relieving hot spots on the World Wide Web. In: *Proceedings of the Twenty-ninth annual ACM symposium on Theory of computing (STOC 1997)*, ACM Press, 1997, S. 654–663

Kawell et al. 1988

KAWELL, Leonard J. ; BECKHARDT, Steven ; HALVORSEN, Timothy ; OZZIE, Raymond ; GREIF, Irene: Replicated document management in a group communication system. In: *CSCW '88: Proceedings of the 1988 ACM conference on Computer-supported cooperative work*, ACM Press, 1988

Keil-Slawik und Hampel 2003

KEIL-SLAWIK, Reinhard ; HAMPEL, Thorsten: Neue Wege kooperativen Lernens - Das Paderborner Jour-Fixe-Konzept. In: *DFN Mitteilungen* 63 (2003), Nr. 11, S. 16–20

Keil-Slawik et al. 2005

KEIL-SLAWIK, Reinhard ; HAMPEL, Thorsten ; ESSMANN, Bernd: Re-Conceptualizing Learning Environments: A Framework for Pervasive eLearning. In: *Proceedings of the Third IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW 2005)*, 2005, S. 322–326

Keil-Slawik und Selke 1998

KEIL-SLAWIK, Reinhard ; SELKE, Harald: Forschungsstand und Forschungsperspektiven zum virtuellen Lernen von Erwachsenen. In: *Kompetenzentwicklung '98 – Forschungsstand und Forschungsperspektiven* (1998), S. 165–208

Kistler und Satyanarayanan 1992

KISTLER, James J. ; SATYANARAYANAN, M.: Disconnected Operation in the Coda File System. In: *ACM Transactions on Computer Systems (TOCS)* 10 (1992), Nr. 1, S. 3–25

Kleinrock 1995

KLEINROCK, Leonard: Nomadic computing-an opportunity. In: *CM SIGCOMM Computer Communication Review* 25 (1995), Nr. 1, S. 36–40

Kubiatowicz et al. 2000

KUBIATOWICZ, John D. ; WEIMER, Westley ; WELLS, Chris ; ZHAO, Ben Y. ;

BINDEL, David ; CHEN, Yan ; CZERWINSKI, Steven ; EATON, Patrick ; GEELS, Dennis ; GUMMADI, Ramakrishnan ; RHEA, Sean C. ; WEATHERSPOON, Hakim: OceanStore: An Architecture for Global-Scale Persistent Storage. In: *ACM SIGOPS Operating Systems Review* 34 (2000), Nr. 5, S. 190–201

Kung und Robinson 1981

KUNG, H. T. ; ROBINSON, John T.: On Optimistic Methods for Concurrency Control. In: *ACM Transactions on Database Systems (TODS)* 6 (1981), Nr. 2, S. 213–226

Lehman et al. 2001

LEHMAN, Tobin J. ; COZZI, Alex ; XIONG, Yuhong ; GOTTSCHALK, Jonathan ; VASUDEVAN, Venu ; LANDIS, Sean ; DAVIS, Pace ; KHAVAR, Bruce ; BOWMAN, Paul: Hitting the distributed computing sweet spot with TSpaces. In: *Computer Networks: The International Journal of Computer and Telecommunications Networking* 34 (2001), Nr. 4, S. 457–472

Lundgren et al. 2002

LUNDGREN, Henrik ; LUNDBERG, David ; NIELSEN, Johan ; NORDSTRÖM, Erik ; TSCHUDIN, Christian: A Large-scale Testbed for Reproducible Ad hoc Protocol Evaluations. In: *IEEE Wireless Communications and Networking Conference 2002 (WCNC'02)* 1 (2002), S. 412–418

Maymounkov und Mazières 2002

MAYMOUNKOV, Petar ; MAZIÈRES, David: Kademia: A Peer-to-Peer Information System Based on the XOR Metric. In: DRUSCHEL, Peter (Hrsg.) ; KAASHOEK, M. Frans (Hrsg.) ; ROWSTRON, Antony (Hrsg.): *Peer-to-Peer Systems: First International Workshop, IPTPS 2002. Revised Papers*. Springer Verlag, 2002, S. 53–65

Oliveira et al. 1999

OLIVEIRA, Manuel ; CROWCROFT, Jon ; BRUTZMAN, Don ; SLATER, Mel: Components for distributed virtual environments. In: *Proceedings of the ACM symposium on Virtual reality software and technology (VRST 1999)*, ACM Press, 1999, S. 176–177

OLSR 2003

CLAUSEN, Thomas (Hrsg.) ; JACQUET, Philippe (Hrsg.): Optimized Link State Routing Protocol (OLSR) / IETF Network Working Group. 2003 (RFC3626). – Request For Comment

Park und Corson 1997

PARK, Vincent D. ; CORSON, M. S.: A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks. In: *Proceedings of the 16th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 1997)*, IEEE Computer Society, 1997, S. 1405

Park und Corson 2001

PARK, Vincent D. ; CORSON, Scott M.: Temporally-Ordered Routing Algorithm (TORA) Version 1 Functional Specification / IETF MANET Working Group. 2001 (20 Juli 2001). – Internet Draft

Perkins 1997

PERKINS, Charles E.: *Mobile IP: Design Principles and Practices*. Addison-Wesley, 1997

Perkins 2001

PERKINS, Charles E.: *Ad Hoc Networking*. Boston, USA : Addison-Wesley, 2001

Perkins 2002

PERKINS, Charles E.: IP Mobility Support for IPv4 / IETF Network Working Group. 2002 (RFC3220). – Technical Report

Perkins et al. 2003

PERKINS, Charles E. ; BELDING-ROYER, Elizabeth M. ; DAS, S.: Ad hoc On-Demand Distance Vector (AODV) Routing / IETF Network Working Group. 2003 (RFC3561). – Technical Report

Perkins und Bhagwat 1994

PERKINS, Charles E. ; BHAGWAT, Pravin: Highly Dynamic Destination-Sequenced Distance Vector (DSDV) for Mobile Computers. In: *SIGCOMM 1994 Conference on Communications Architectures, Protocols and Applications* (1994), S. 234–244

Perkins und Bhagwat 2000

PERKINS, Charles E. ; BHAGWAT, Pravin: DSDV – Routing over a Multihop Wireless Network of Mobile Computers. In: *Ad Hoc Networking* (2000), S. 53–74

Perkins et al. 2001

PERKINS, Charles E. ; MALINEN, Jari T. ; WAKIKAWA, Ryuji ; BELDING-ROYER, Elizabeth M. ; SUN, Yuan: IP Address Autoconfiguration for Ad Hoc Networks / IETF Mobile Ad Hoc Networking Working Group. 2001 (draft-ietf-manet-autoconf-01.txt). – Technical Report

Perkins und Royer 1999

PERKINS, Charles E. ; ROYER, Elizabeth M.: Ad-hoc On-Demand Distance Vector Routing. In: *Proceedings of the Second IEEE Workshop on Mobile Computing Systems and Application (WMCSA 1999)*, IEEE Computer Society, 1999, S. 90–100

Picco et al. 1999

PICCO, Gian P. ; MURPHY, Amy L. ; ROMAN, Gruia-Catalin: LIME: Linda Meets Mobility. In: GARLAN, D. (Hrsg.): *Proceedings of the 21st International Conference on Software Engineering (ICSE 1999)*, ACM Press, 1999, S. 368–377

Popek et al. 1990

POPEK, Gerald J. ; GUY, Richard G. ; PAGE, Thomas W. ; HEIDEMANN, John S.: Replication in Ficus Distributed File Systems. In: CABRERA, Luis-Felipe (Hrsg.) ; PÂRIS, Jehan-François (Hrsg.): *Proceedings of the First Workshop on the Management of Replicated Data*, IEEE Computer Society, 1990, S. 5–10

Ramjee et al. 1999

RAMJEE, R. ; VARAFHAN, K. ; SALGARELLI, L. ; THUEL, S. ; WANG, S. Y. ; PORTA, T. L.: HAWAII: A Domain-Based Approach for Supporting Mobility in Wide-Area Wireless Networks. In: *International Conference Network Protocols (ICNP'99)* (1999), S. 283–292

Rashid et al. 1989

RASHID, Richard ; BARON, Robert ; FORIN, Alesandro ; GOLUB, David ; JONES, Michael B. ; JULIN, Daniel ; ORR, Douglas ; SANZI, Richard: Mach: A Foundation for Open Systems. In: *Proceedings of the Second Workshop on Workstation Operating Systems (WWOS2)*, IEEE Computer Society, 1989, S. 176–178

Ratnasamy et al. 2001

RATNASAMY, Sylvia ; FRANCIS, Paul ; KARP, Richard ; SHENKER, Scott: A Scalable Content-Addressable Network. In: *Proceedings of the 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM 2001)*, 2001, S. 161–172

Reinbold und Bonaventure 2001

REINBOLD, Pierre ; BONAVENTURE, Oliver: A Comparison of IP mobility Protocols / Infonet Group, University of Namur, Belgium. 2001 (infonet-TR-13). – Technical Report

Rhea et al. 2005a

RHEA, Sean C. ; BRIGHTEN, Godfrey ; KARP, Brad ; KUBIATOWICZ, John ; RATNASAMY, Sylvia ; SHENKER, Scott ; STOICA, Ion ; YU, Harlan: OpenDHT: a public DHT service and its uses. In: *Proceedings of the 2005 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM 2005)*, ACM Press, 2005, S. 73–84

Rhea et al. 2003

RHEA, Sean C. ; EATON, Patrick ; GEELS, Dennis ; WEATHERSPOON, Hakim ; ZHAO, Ben Y. ; KUBIATOWICZ, John: Pond: the OceanStore Prototype. In: *Proceedings of the Second USENIX Conference on File and Storage Technologies (FAST '03)*, 2003, S. 1–14

Rhea et al. 2005b

RHEA, Sean C. ; GEELS, Dennis ; ROSCOE, Timothy ; KUBIATOWICZ, John: Handling Churn in a DHT. In: *Proceedings of the USENIX Annual Technical Conference (USENIX 2004)*, 2005, S. 127–140

Rowstron und Druschel 2001a

ROWSTRON, Antony ; DRUSCHEL, Peter: Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. In: *Proceedings of the IFIP/ACM International Conference on Distributed Systems Platforms (Middleware)*, 2001, S. 329–350

Rowstron und Druschel 2001b

ROWSTRON, Antony ; DRUSCHEL, Peter: Storage management and caching in PAST, a large-scale, persistent peer-to-peer storage utility. In: *Proceedings of the 18th ACM Symposium on Operating Systems Principles (SOSP 2001)*, ACM Press, 2001, S. 188–201

Rowstron et al. 2001

ROWSTRON, Antony ; KERMARREC, Anne-Marie ; CASTRO, Miguel ; DRUSCHEL, Peter: SCRIBE: The Design of a Large-Scale Event Notification Infrastructure. In: CROWCROFT, Jon (Hrsg.) ; HOFMANN, Markus (Hrsg.): *Proceedings of the Third International COST264 Workshop on Networked Group Communication (NDC 2001)* Bd. 2233, Springer Verlag, 2001, S. 30–43

Satyanarayanan 1996

SATYANARAYANAN, M.: Fundamental challenges in mobile computing. In: *Proceedings of the 15th annual ACM Symposium on Principles of Distributed Computing (PODC'96)*, 1996, S. 1–7

Satyanarayanan 2002

SATYANARAYANAN, M.: The Evolution of Coda. In: *ACM Transactions on Computer Systems* 20 (2002), Nr. 2, S. 85–124

Satyanarayanan et al. 1993

SATYANARAYANAN, M. ; KISTLER, J.J. ; MUMMERT, L.B. ; EBLING, M. R. ; KUMAR, P. ; LU, Q.: Experience with disconnected operation in a mobile computing environment. In: *USENIX Symposium on Mobile and Location-Independent Computing* (1993), S. 11–28

Slawik et al. 2004

SLAWIK, Joanna ; ESSMANN, Bernd ; HAMPEL, Thorsten: Shared Views on Mobile Knowledge - A Concept of a Graphical User Interface. In: KARAGIANIS, Dimitris (Hrsg.) ; REIMER, Ulrich (Hrsg.): *Proceedings of the 5th International Conference on Practical Aspects of Knowledge Management (PAKM 2004)*, Springer Verlag, 2004, S. 82–93

Stirling und Al-Ali 2003

STIRLING, David ; AL-ALI, Firas: Zero Configuration Networking. In: *ACM Crossroads* 9 (2003), Nr. 4, S. 19–23

Stoica et al. 2002

STOICA, Ion ; ADKINS, Daniel ; ZHUANG, Shelley ; SHENKER, Scott ; SONESH,

Surana: Internet Indirection Infrastructure. In: *ACM SIGCOMM Computer Communication Review* 32 (2002), Nr. 4, S. 73–86

Stoica et al. 2001a

STOICA, Ion ; MORRIS, Robert ; KARGER, David ; KAASHOEK, M. Frans ; BALAKRISHNAN, Hari: Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications. In: *Proceedings of the Annual Conference of the ACM Special Interest Group on Data Communication (SIGCOMM 2001)*, ACM Press, 2001, S. 149–160

Stoica et al. 2001b

STOICA, Ion ; MORRIS, Robert ; KARGER, David ; KAASHOEK, M. Frans ; BALAKRISHNAN, Hari: Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications. In: *Proceedings of the Annual Conference of the ACM Special Interest Group on Data Communication (SIGCOMM 2001)*, ACM Press, 2001, S. 149–160

Terry et al. 1995

TERRY, Douglas B. ; THEIMER, M. M. ; PETERSEN, Karin ; DEMERS, A. J. ; SPREITZER, Mike J. ; HAUSER, C. H.: Managing update conflicts in Bayou, a weakly connected replicated storage system. In: *ACM SIGOPS Operating Systems Review* 29 (1995), Nr. 5, S. 172–182

Teufel et al. 1995

TEUFEL, Stefanie ; SAUTER, Christian ; MÜHLHERR, Thomas: *Computerunterstützung für die Gruppenarbeit*. Oldenbourg, 1995

Tschudin et al. 2005

TSCHUDIN, Christian F. ; GUNNINGBERG, Per ; LUNDGREN, Henrik ; NORDSTRÖM, Erik: Lessons from experimental MANET research. In: *Ad Hoc Networks* 3 (2005), Nr. 2, S. 221–233

Tseng et al. 2003

TSENG, Yu-Chee ; SHEN, Chia-Ching ; CHEN, Wen-Tsuen: Integrating Mobile IP with Ad Hoc Networks. In: *IEEE Computer* (2003), S. 48–55

Valko 1999

VALKO, Andreas G.: Cellular IP: A New Approach to Internet Host Mobility. In: *ACM Computer Communication Review* 29 (1999), S. 50–65

Watson und Zyda 1998

WATSEN, Kent ; ZYDA, Michael: Bamboo – A Portable System for Dynamically Extensible, Real-Time, Networked, Virtual Environments. In: *Proceedings of the IEEE 1998 Virtual Reality Annual International Symposium (VRAIS 1998)*, IEEE Computer Society, 1998, S. 252–259

Weiser 1991

WEISER, Marc: The Computer for the Twenty-First Century. In: *Scientific American* (1991), S. 94–104

Wykoff et al. 1998

WYKOFF, Peter ; McLAUGHRY, Stephen ; LEHMAN, Tobin J. ; FORD, Daniel: TSpaces. In: *IBM Systems Journal* 37 (1998), Nr. 3, S. 454–474

Xu und Liskov 1989

XU, A. ; LISKOV, B.: A design for a fault-tolerant, distributed implementation of Linda. In: *Proceedings of the 19th International Symposium on Fault-Tolerant Computing 1989. (FTCS-19)*, 1989, S. 199–206

Yu und Vahdat 2002

YU, Haifeng ; VAHDAT, Amin: Design and evaluation of a conit-based continuous consistency model for replicated services. In: *ACM Transactions on Computer Systems (TOCS)* 20 (2002), Nr. 3, S. 239–282

Zhao et al. 2004

ZHAO, Ben Y. ; HUANG, Ling ; STRIBLING, Jeremy ; RHEA, Sean C. ; JOSEPH, Anthony D. ; KUBIATOWICZ, John D.: Tapestry: A Resilient Global-scale Overlay for Service Deployment. In: *IEEE Journal on Selected Areas in Communications (JSAC)* 22 (2004), Nr. 1, S. 41–53

Zhou et al. 2003

ZHOU, Hongbo ; NI, Lionel M. ; MUTKA, Matt W.: Prophet Adress Allocation for Large Scale MANETs. In: *Ad Hoc Networks Journal* 1 (2003), Nr. 4, S. 423–434

Zhuang et al. 2003

ZHUANG, Shelley Q. ; LAI, Kevin ; STOICA, Ion ; KATZ, Randy ; SHENKER, Scott: Host Mobility Using an Internet Indirection Infrastructure. In: *Proceedings of the First International Conference on Mobile Systems, Applications, and Services (MobiSys 2003)*, 2003, S. 129–144

Zhuang et al. 2001

ZHUANG, Shelley Q. ; ZHAO, Ben Y. ; JOSEPH, Anthony D. ; KATZ, Randy ; KUBIATOWICZ, John: Bayeux: An Architecture for Scalable and Fault-tolerant Wide-area Data Dissemination. In: *Proceedings of the Eleventh International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV 2001)*, ACM Press, 2001, S. 11–20

Mobilität in der kooperativen Wissensarbeit

Stichwortverzeichnis

A

- Abstraktionsschicht 146
- ACID-Eigenschaften 119
- Ad hoc On Demand Distance Vector
 siehe AODV
- Ad hoc Protocol Evaluation (APE) 95
- Ad-Hoc-Kooperation 63
- Ad-Hoc-Netzwerk 27, 62, 76, 91,
 94 – 97, 100
- Adressatenkreis, flexibler 78
- AFS 120
- aktive Objekte 84
- Andrew File System *siehe* AFS
- Anycast 103
- AODV 95
- Apache Foundation 90
- API 103
- Application Programm. Interface *siehe*
 API
- Arbeitsbereich
 - gemeinsamer 17, 22, 28, 34, 79, 83
 - persönlicher 21, 83
- Archivierung 70
- ARP-Requests 96
- asynchrone Kooperation 12
- Auflösung 13
- Automatische Konfiguration 62
- Avatare 99
- Awareness 144

B

- Bamboo 103, 114
- Bayeux 103
- Bayou 120
- Berechtigungen 83

- Bittorrent 103
- Bluetooth 90
- Broadcast 94, 101
- BSCW 99

C

- Caching 65, 112 f
- CAN 103
- Care-of-Adresse 93, 96
- CAS 45, 144
- Casca 99
- CAST 103
- CellularIP 97
- CFS 103, 114
- Chord 103
- Coda 120
- Compute-Dienste 144
- Computer Algebra System *siehe* CAS
- Cooperative FileSystem ... *siehe* CFS
- Cooperative Virt. Environments *siehe*
 CVE

- CSCW 53
 - Cluster 90
- CVE 90
 - Bamboo 90

D

- Decentralized Object Location and Routing *siehe*
 DOLR
- Delay 93
- Destination-Sequence Distance-Vector
 siehe DSDV
- DHT 101 – 116
- Dienst

Verteilung	77
Diensteinfrastruktur	97
Distant Learning	41
Distributed Hash Table (DHTs) ..	132
Distributed Hashtable <i>siehe</i> DHT	
Distributed Hashtable (DHT)	121
Distribution	78
DOLR	102 f
Dreiecksrouting	93
DSDV	95, 97
DSR	95
Duplicate Adress Detection	96
Dynamic Source Routing .. <i>siehe</i> DSR	
E	
E-Learning	41
Eclipse	90
Einstiegsknoten	108
eMule	103
epidemische Distribution	79
Ereigniskontrolle	78, 121 – 128
Expanding Multicast	111
externe Dienste	143
F	
FA	93
Ficus	120
Filesharing	101
Foreign Agent	<i>siehe</i> FA
Foreign Domain Root Router	97
Forwarding Schemes	97
Freifunk Projekt	95
G	
Gültigkeitsdatum	16
Gastrechte und -rollen ...	24 f, 64, 81 f
GLOMAR	120
GNU/Maverik	90
Gnutella	101
Gründung	13
Graphen	134
Groove	99
Groupware	99
Gruppe	
Gründung	56
Kalender	19 f, 80
Kommunikation	128, 142
Koordination	142
Struktur	62 f, 71, 141
Verwaltung	83, 128
Gruppen	83
Gruppen-Tuple Space	140
Gruppenadministrator	17
Gruppenbereich . <i>siehe</i> Arbeitsbereich	
Gruppengründung	16
H	
HA	93
Handoff-Aware Wireless Access Internet Infrastructure	<i>siehe</i> HAWAII
Hash	
Exact-Matching	102
Funktion	102, 138
Index	102
Tabelle	101
Wert	102
Hash-based Distribution	138
HAWAII	97
Hierarchical Mobile IP	97
Hivemind	90
Hoarding	120
Home Agent	i
Hop	94
Host Tuple Space	118, 140
Host-Level Tuple Space ... <i>siehe</i> Host Tuple Space	
Hybrid Broadcast	139
Hypercube	135
I	
I3	103
IANA	94
IBM Research	117
ICMP	96

IEEE 802.11	<i>siehe WLAN</i>
IEEE 802.15.1	<i>siehe Bluetooth</i>
IEEE 802.16	<i>siehe WIMAX</i>
Independent Nodes	131, 140
Indexdienst	101, 142
Informationsdienste	144
Inhaltsbereiche	133
Integration externer Dienste	72
Inter Domain Mobility	<i>siehe Macro Mobility</i>
Interface Tuple Space	<i>siehe ITS</i>
Internet Assigned Numbers Authority	<i>siehe IANA</i>
Internet Indirection Infrastructure	<i>siehe I3</i>
Intra Domain Mobility	<i>siehe Micro Mobility</i>
IP-Adresse	92
IPv4	92, 96
IPv6	93, 96
ITS	118
J	
JADE	90
Jarkarta Projekt	90
Java	117
JavaSpaces	117
JBOSS	90
JBossMicrokernel	90
Jini	117
Jour Fixe	48
JXTA	98, 100, 117
JXTASpaces	117
K	
Kademlia	103
KBR	102 f
Key	<i>siehe Hash-Wert</i>
Key-Based Routing	<i>siehe KBR</i>
Knotenausfall	67
Knotengrad	134
Kollaboration	73
Kommunikation	75 – 80
entkoppelte	118
Kommunikationsschicht	145 f
Konflikt	<i>siehe Versionskonflikt</i>
Konflikterkennung	70
Konsistenz	66 – 70, 79, 84, 118 ff
Gefährdung der	68
semantische	68 f
Kontext	63
Kontextdienste	143
Kooperation	13, 83 f
räumlich verteilt	34
Kooperationsgruppe	56
temporäre	29
Kooperationsphasen	13
Koordination	80
L	
Leaf Set	106 ff, 110
Lerngruppe	15
LIME	117, 138
Linda	117
Linda in a Mobile Environment	i
Link-Local	94
Location Awareness	<i>siehe Ortsbezug</i>
Locking	
Optimistic	69, 119
Pessimistic	69
Lokalität	102, 108
Lotus Notes	99
M	
Maß der Nähe	107 f
Macro Mobility	92, 96
MadKit	90
Manet	94, 97
Matching	117
mDNS	98
Medien	78 ff
Medienbrüche	14, 57 f, 71, 80
Medienfunktionen	58, 99
individuelle	58
kooperative	58
primäre	58

- sekundäre 58
- tertiäre 58
- Micro Mobility 92, 94, 96
- Microkernel 88 ff
- Microkernel 89
- Mobile IP 92 ff, 96 f
- Mobility 92
- Moderator 64
- Modul 90
- Modularität 72
- monolithische Architektur 72, 89
- Multicast 98, 103, 121 f, 134
- Multicast DNS *siehe* mDNS
- Multihop 94
- Multiple Write Replication 119
- Multiple-Unicast 125

- N**
- Nachrichtenschlüssel 105
- Nachrichtensystem 18, 77
- Napster 101
- Negative Broadcast 139, 142
- Neighborhood Set 106, 108, 110
- Network Simulator 2 *siehe* ns2
- Network Tuple Space 140
- Netzwerkprotokolle 92
- Netzwerksimulatoren 95
- ns2 95

- O**
- Objekt
 - aktives *siehe* aktive Objekte
 - personalisiert *siehe* personalisierte Objekte
 - Sichtbarkeit 81
- Objektdistribution 145
- Objekte 78 ff
- Objektmigration 78
- Objektsuche 142
- Objektverwaltung 145
- Observer-Paradigma 121
- OceanStore 90, 103, 114
- Offene Architektur 71 ff

- Offline-Verfügbarkeit 77 f, 101, 104, 117, 119, 128, 131, 141
- OLSR 95
- On Demand Nodes 132, 140
- open-sTeam 12, 99
- OpenDHT 114
- Operator-based Distribution 138
- Optimistic Concurrency Control *siehe* Locking
- Optimized Link State Routing .. *siehe* OLSR
- Ortsbezug 64, 98, 144
- Overlay-Netzwerk 102, 132

- P**
- Partitionierung, Netzwerk- 136
- PAST 103, 105, 111 ff
- Pastry 103 – 116, 121
- Peer 53
- Peer-to-Peer 53, 98 f, 102, 117
- Persistent Linda *siehe* PLinda
- Persistenz 100 – 120
 - verteilte 65, 78, 84, 101 – 120
- Persistenzdienst 79
- Persistenzdienste 144
- personalisierte Objekte 84
- PFS 120
- Phasen der Kooperation
 - Abschluss 55
 - Gründung 55
 - Kooperation 55
- PLinda 117
- Polling 121, 134
- Pond 114
- portabel 87
- Portability 92
- Positive Broadcast 139
- Präfix 105
- Präsenzkooperation 14, 62
- PRAYER File System *siehe* PFS
- proaktiv 95
- Prophet Adress Allocation 96
- Protokollabstraktion 103

Proximity Metric <i>siehe</i> Maß der Nähe	
Proxy	98
Publish/Subscribe-Paradigma	121
Pull-Paradigma	121
Pushing	121, 134
Q	
Quotaverwaltung.....	112
R	
reakтив	95
Rechteverwaltung	81
Redundanz	71, 79, 84
Rendezvous-Knoten	122
Rendezvous-Peers	98
Replikation	66, 79, 112, 131
Replikationsfaktor	112
Replikationsstrategie	130 f
Request&Reply-Paradigma	121
Reversibilität	68, 70 f
ROAM	104
Robustheit	77, 93
Rollen und Rechte	64, 80 ff
Routing	94, 102, 105
Routing Tabelle	106, 110
Routing-Tabelle	108
RSS Feeds	144
S	
Scribe	103, 121 – 128
Abonnentenverwaltung	122
Ausfallsicherheit	125
Nachrichtenverbreitung	125
Thema abbestellen	123
Thema abonnieren	122
Selective Nodes	131, 140
Semantische Inkonsistenz	48
Service Discovery	97 f
shared workspace	34
Single-Multicast	124
Skalierbarkeit	93
Sparse Graph	136
Speakeasy	99
Speicher	
verteilter	<i>siehe</i> Persitenz
Sperren	<i>siehe</i> Locking
Splitstream	103
spontane Kooperation	26
spontane Vernetzung	76
sTeam	<i>siehe</i> open-sTeam
Suche	<i>siehe</i> Objektsuche
SUN Microsystems	117
Symmetrieeigenschaft	135
synchrone Kooperation	12
Synchronisation	68, 119, 130
Online	67
Synchronizität	68 ff
T	
TACT	120
Tapestry	103, 114
Template	117, 138
Temporally-Ordered Routing Algorithm	
<i>siehe</i> TORA	
Terminbestimmung	80
Terminfindung	19
Tier	103
TORA	95
TSpaces	117
Tuple	117, 138
Tuple Space	117
U	
Usecase	12
V	
Vernetzung	76 ff
Verschlüsselung	81, 112
Versionierung	70
Versionskonflikt	68 f, 119, 130, 133
verteilte Hash-Tabelle	<i>siehe</i> DHT
verteilte Systeme	
offline-redundante	67
online-redundante	67

Verteilungsstrategie 134
Verzögerung *siehe* Delay
Vier-Schichten-Modell 146
virtueller Wissensraum . 12, 58, 79, 99

W

Warteliste 24
WIMAX 90
Wireless Local Area Network ... *siehe*
WLAN
WLAN 90
Workflow 47
Workflows 84
Worldwide Interoperability for Micro-
wave Access *siehe*
WIMAX

Z

Zeroconf Workinggroup (IETF)....98
Zugriffskontrolle 18, 48, 65, 81, 83