„DreamTeam“ - eine synchrone CSCW-Plattform für heterogene Umgebungen

Jörg Roth
Fernuniversität Gesamthochschule in Hagen
Fachbereich Informatik
Praktische Informatik II
Joerg.Roth@Fernuni-Hagen.de

Zusammenfassung: Dieses Paper stellt eine Plattform zur synchronen Gruppenarbeit vor. Neben der Einordnung in das Gebiet CSCW (Computer Supported Cooperative Work) wird diese Plattform gegenüber bestehenden Plattformen abgegrenzt. Ein Merkmal dieser Plattform ist, daß die Belange von Studenten der Fernuniversität berücksichtigt werden. Studenten wählen sich in der Regel von einem PC über eine Modemverbindung in das Internet ein. Gegenüber den CSCW-Systemen, bei denen eine schnelle Verbindung über lokale Netzwerke sowie eine ständige Erreichbarkeit im Netz vorausgesetzt wird, ergeben sich damit zusätzliche Problemfelder.

1 Einleitung

Aktuelle Realisierungen zur computerunterstützten Gruppenarbeit lassen sich in die Bereiche *asynchrone* und *synchrone* Systeme einteilen. Asynchrone Systeme unterstützen Gruppen, deren Teilnehmer sich zeitlich nicht abstimmen müssen. Einige asynchrone System haben in den letzten Jahren teilweise eine beachtliche Verbreitung erreicht. Beispiele für solche Systeme sind *Lotus Notes* und *BSCW*. Einen Überblick über asynchrone Gruppensysteme vermittelt [HHW+96].
Bei *synchronen* Gruppensystemen wird vorgesetzt, daß sich Teilnehmer zeitlich abstimmen, d.h. eine Gruppe muß sich für einen bestimmten Zeitpunkt zusammenfinden, um eine Sitzung abzuhalten. Im Gegensatz zu den asynchronen Systemen stehen hier die Problembereiche *Echtzeitkommunikation* (teilweise mit Multimediadaten), *Synchronisation* und *Konsistenzerhaltung* im Vordergrund. Synchrone Systeme kann man in die Gebiete *computerunterstützte Kommunikation* und *verteilte Applikationen* einteilen, wobei es auch Systeme gibt, die beides abdecken. Bei der computerunterstützten Kommunikation werden Rechner und Netzwerk als Datenkanal benutzt. Der Rechner ist dabei nur zur Datenaufbereitung, z.B. zur Datenkomprimierung und -dekomprimierung, notwendig. Beispiele sind Audio- und Videokonferenz-Systeme. Die Hauptfunktion, nämlich die Übertragung von Video- und Audiodaten, könnte prinzipiell auch über andere Kanäle durchgeführt werden (Audio über Telefon, Video über Analogkanäle). Der Computer wird hier als Ein- und Ausgabemedium verwendet, da von dort aus in bequemer Weise auf eine vorhandene Netzinfrastruktur zugegriffen werden kann.
Der Bereich der verteilten Anwendungen setzt dagegen den Arbeitsplatzrechner als Front-End zwingend voraus, da die Anwendungen seine Rechenfähigkeit benötigen. Während bei der

computerunterstützten Kommunikation der Echzeittransport von Multimediadaten im Vordergrund steht, sind bei verteilten Anwendungen Probleme der Datenkonsistenz und der verteilten Benutzerführung zu lösen.
Verteilte Anwendungen teilt man in die Klassen *kollaborationstransparent* und *kollaborationsbewußt* ein. Anwendungen der ersten Klasse sind ursprüngliche Einzelplatzanwendungen, die durch Modifikation bzw. durch ein geeignetes Hilfsmittel einer verteilten Benutzergruppe zugänglich gemacht werden. Bei der Erstellung solcher Anwendungen wurde der Tatsache keine Beachtung geschenkt, daß sie in verteilten Umgebungen laufen. Somit existieren keine gruppenspezifischen Funktionen in den Anwendungen.
Kollaborationsbewußte verteilte Anwendungen werden dagegen explizit für den verteilten Betrieb konzipiert. Sie besitzen u.a. eigene Dialogelemente für den Mehrbenutzerbetrieb, und es wird zwischen privaten und öffentlichen Fenstern unterschieden.
Eine Übersicht über die vielfältigen Ausprägungen von CSCW ist in [Scho96] zu finden.

2 Eigenschaften einer CSCW-Plattform

Im folgenden wird der Bereich kollaborationsbewußter, verteilter, synchroner Anwendungen näher betrachtet. Synchrone CSCW-Systeme sollten neben verteilten Anwendungen geeignete Kommunikationskanäle bereitstellen. In der Regel wird ein gemeinsamer Audiokanal als wichtig empfunden, während man oft auf Video verzichtet bzw. aus Gründen der Bandbreite nicht sinnvoll nutzen kann. Im folgenden wird davon ausgegangen, daß ein entsprechender Kanal vorliegt und das Augenmerk mehr auf die Problematik verteilter Anwendungen gelenkt.

2.1 Allgemeine Eigenschaften

Nach [RG96] gibt es vier Bereiche, die eine CSCW-Plattform für synchrone Anwendungen abdecken sollte:

- *runtime-architectures*:
  Hierunter werden die Gebiete *verteilte Kontrolle*, *Synchronisation*, *Kommunikation* und *Fehlerbehandlung* verstanden.
- *groupware programming abstraction*:
  Bestimmte Dienste sollten dem Anwendungsentwickler von verteilten Anwendungen zur Verfügung stehen. Darunter fallen u.a. die Funktionen zur Kommunikation (RPC, Multicast, Eventverteilung, verteilte Objekte etc.).
- *groupware widgets*:
  Zur Dialoggestaltung verteilter Anwendungen sind spezielle Dialogelemente notwendig. Die Plattform sollte die wesentlichen zur Verfügung stellen, damit der Anwendungsentwickler von dieser Arbeit entlastet wird. Beispiele für gruppenspezifische Dialogelemente sind Multi-User-Scrollbars, Telepointer und Fenster für den Benutzerstatus.
- *session managers*:
  Für die Verwaltung von Gruppensitzungen gibt es verschiedene Modelle. Der Entwickler von verteilten Anwendungen sollte sich nicht mehr mit der Aufgabe der Sitzungsverwaltung beschäftigen. Vielmehr sollten sitzungsrelevante Dienstleistungen als Building Blocks zur Verfügung stehen.

Die Laufzeitumgebung eines entsprechenden CSCW-Systems sollte die folgenden Grunddienste abdecken:

- Profilverwaltung:
  Alle relevanten Profile (Benutzerprofile, Rechnerprofile, Kanalprofile und Sitzungsprofile)

müssen von einem Laufzeitsystem verwaltet werden. Wenn notwendig, müssen Profile persistent gehalten werden.

- Kommunikationskanäle:
  Alle Kommunikationskanäle sollten innerhalb des Laufzeitsystems bereitgestellt und verwaltet werden. Es muß verhindert werden, daß verteilte Anwendungen außerhalb der Plattform eigene Kanäle öffnen. Neben der reinen Bereitstellung der Kanäle sorgt eine Serviceabstraktion dafür, daß die verteilten Anwendungen auf einem hohen Niveau miteinander kommunizieren können. Diese Abstraktion wird durch die Bereitstellung entsprechender Protokollfamilien erreicht.
- Sitzungsverwaltung:
  Die Plattform muß die Sitzungen und deren Zustände verwalten. Über eine geeignete Schnittstelle werden Sitzungsinformation den verteilten Anwendungen zugänglich gemacht.

2.2 Zentrale vs. dezentrale Topologie

Es ist eine grundlegende Frage, die bei der Konzeption von CSCW-Systemen zu klären ist, ob eine zentrale oder dezentrale Topologie gewählt werden soll. Eine zentrale oder servergestützte Topologie besitzt einen ausgezeichneten Rechner, den Server, der die Daten und Ereignisse verwaltet. Ein herausragender Vorteil dieser Architektur ist, daß die Synchronisation der verschiedenen Benutzer sehr einfach ist. Da alle Nachrichten über den Server transportiert werden, findet ein Serialisierung statt, und alle Klienten erhalten sie in derselben Reihenfolge. Datenkonsistenz kann erzielt werden, indem alle relevanten Datenobjekte auf dem Server liegen und der Zugriff von außen über definierte Transaktionen erfolgt.
Bei dezentralen oder replikativen Systemen gibt es keinen ausgezeichneten Server. Alle Teilnehmer sind untereinander verbunden. Da in einem Netzwerk der Datentransport unterschiedlich lange dauern kann, erreichen Ereignisse die Teilnehmer zu unterschiedlichen Zeitpunkten. Für die Erhaltung der zeitlichen Reihenfolge kann damit keine Gewähr übernommen werden. Solche Probleme müssen von überlagerten Protokollen abgefangen werden.
Der Vorteil von dezentralen Systemen liegt darin, daß sie ohne einen Server funktionieren. Ein Server stellt meist sowohl von der Verfügbarkeit, als auch von den Kommunikationsbandbreiten einen Flaschenhals dar, so daß u.U. der Vorteil einer einfachen Systemstruktur aufgehoben wird. Eine ausführliche Diskussion über die Frage zentrale vs. dezentrale Architektur findet man in [GR96].

2.3 Striktes vs. relaxiertes WYSIWIS

Teilnehmer an einer Gruppensitzung bearbeiten denselben Informationsraum. Dies hat unmittelbar Auswirkungen darauf, wie Benutzer die gemeinsamen Daten wahrnehmen. Es gibt verschiedene Modelle. Ein häufiges ist *WYSIWIS* (What You See Is What I See). Hiermit wird ausgedrückt, daß jeder Benutzer genau dieselbe Sicht auf die gemeinsamen Daten hat. Aus der Perspektive eines einzelnen Benutzer heißt das, daß jedes Ereignis oder jedes Objekt, auf das er sich in einer bestimmten Aktion bezieht, von allen anderen Teilnehmern auch exakt so wahrgenommen wird. Eine Form, in der das Aussehen auf allen Rechnern exakt gleich aussieht, wird *striktes WYSIWIS* genannt.
In der Regel hat jedoch jeder Benutzer unterschiedliche Vorlieben, seinen Arbeitsbereich einzurichten. Hieraus resultieren geringfügige Unterschiede in der Darstellung. Werden neben den öffentlichen Bereichen auch noch private Arbeitsbereiche benutzt, kann prinzipiell nicht mehr auf allen Rechnern exakt dieselbe Sicht vorausgesetzt werden. Hierbei spricht man dann von *relaxiertem WYSIWIS.* Bei dieser Form ist zwar der grobe Inhalt der verteilten Arbeitsbe-

reiche auf allen Rechnern gleich, es können aber Unterschiede bzgl. der Fensterpositionen, der Skalierungen und der eingestellten Sichtbarkeitsbereiche vorliegen.
In der Regel wird diese Form von WYSIWIS bevorzugt, da eine exakt gleiche Darstellung auf allen Systemen einerseits nur schwer zu erreichen ist, andererseits nicht die Wünsche nach einer individuellen Arbeitsplatzgestaltung beachtet.
Eine Gegenüberstellung der einzelnen Formen verteilter Arbeitsumgebungen ist in [AG94] zu finden.

2.4 Eigenschaften für die Belange der Fernuniversität

Soll ein CSCW-System im Rahmen der Fernuniversität eingesetzt werden, so müssen spezielle Belange berücksichtigt werden. Fernstudenten, die sich zu Gruppen zusammenfinden, verfügen über uneinheitliche Betriebssystemplattformen. Da es unzumutbar ist, für die Arbeit ein bestimmtes Betriebssystem vorzuschreiben, ist eine Lösung zu favorisieren, die für verschiedene Betriebssysteme geeignet ist.
In der Regel sind die Arbeitsplatzrechner der Fernstudenten PCs. Der Zugang zum Netzwerk erfolgt in den meisten Fällen über Modems. Hiermit genügen die Umgebungen nicht den Stabilitätskriterien, die bei lokalen Netzwerken mit Workstations vorliegen. Verbindungsabbrüche oder Systemabstürze erfordern, daß sich ein Student erneut einwählt. Die Umgebung muß für solche Fälle ausgelegt sein und darf keine komplizierten Verfahren für das Wiederaufsetzen verlangen. Zusätzlich muß beachtet werden, daß eine gegenüber lokalen Netzen deutlich verringerte Kommunikationsbandbreite vorliegt.
Ein weiteres wesentliches Merkmal betrifft die Verfügbarkeit im Netzwerk. Fernstudenten sind die meiste Zeit offline und nur die kurze Zeit, in denen sie eine Modemverbindung aufgebaut haben, sind sie im Netzwerk ansprechbar. In der Regel kann also ein potentieller Kommunikationspartner nicht aktiv angesprochen werden, sondern es muß gewartet werden, bis dieser online ist.
Ein weiteres Problem in diesem Zusammenhang sind die ständig wechselnden Netzwerkadressen. Teilnehmer, die über einen Service-Provider an das Netzwerk angeschlossen sind, erhalten jeweils für eine Sitzung eine Netzwerkadresse aus einem Pool zugewiesen. Verfahren, die also einen bestimmten Teilnehmerarbeitsplatz anhand einer festen Netzwerkadresse identifizieren, sind hiermit zum Scheitern verurteilt.

2.5 Die Rendezvous-Problematik

Unter einem *Gruppenrendezvous* (im folgenden kurz *Rendezvous*, [Scho96]) soll das Zusammenfinden von Teilnehmern zu einer Sitzung verstanden werden. Ein Rendezvous findet normalerweise in zwei Phasen statt. Zuerst einigen sich die potentiellen Teilnehmer auf eine Zeitspanne, in der die Sitzung aktiv sein soll. Diese Rendezvous-Phase beginnt u.U. einige Tage vor dem Sitzungsbeginn und kann auf langsame Medien der Informationsübermittlung (z.B. Email) zurückgreifen.
Unmittelbar vor oder während des Sitzungsbeginns findet eine zweite Phase statt. Jeder Teilnehmer muß darüber informiert werden, wer an der Sitzung teilnimmt. Im dezentralen Fall muß darüber hinaus ermittelt werden, wie jeder Teilnehmer über das Netzwerk erreicht werden kann.
Wie oben angesprochen, unterscheiden sich Teilnehmer, die über Einwählpunkte mit dem Netzwerk verbunden sind, von Teilnehmern, die in einem lokalen Netz arbeiten, in zwei wesentlichen Punkten. Zum einen sind sie nicht dauerhaft im Netzwerk erreichbar, zum anderen haben sie keine feste Netzwerkadresse. Diese Punkte haben einen Einfluß darauf, wie sich Teilnehmer zu Gruppen zusammenfinden. Eine einfache Lösung stellt ein *Well-Known-Server*

oder *Registrar* [RG96] dar. Dieser ist für alle potentiellen Teilnehmer fest eingestellt und muß immer erreichbar sein. Alle Sitzungsstarts müssen über ihn abgewickelt werden, und jeder Teilnehmer muß seine An- und Abmeldung an ihn richten. Hiermit kann jeder neue Benutzer auf einfache Weise ermitteln, welche Sitzungen gerade aktiv sind und welche Benutzer an den Sitzungen teilnehmen.
Möchte man jedoch keine zentrale Architektur benutzen, so sind die Lösungsansätze komplizierter. Einen Ausweg bietet eine Hybridstruktur, in der in der Rendezvous-Phase eine zentrale Architektur vorliegt und während der Sitzung eine replikative. Ein derartiger Ansatz ist in [RG96] beschrieben.
Die Entwicklung eines auf die speziellen Bedürfnisse der Fernuniversität ausgelegten Konzeptes ist Gegenstand der laufenden Forschungsaktivitäten.

2.6 Beispiele

2.6.1 Habanero

Habanero [NCSA] ist eine vollständig in Java geschriebene CSCW-Umgebung. Anwendungen, die bereits in Java vorliegen, können durch Hinzunahme von Bibliotheksaufrufen sowie geringfügigen Änderungen für diese Umgebung präpariert werden und stehen damit einer Gruppe zur Verfügung.
Habanero besitzt eine zentrale Architektur. Ein bestimmter Rechner muß den Habanero-Server starten. Die Information, welcher Rechner dies ist, muß allen Teilnehmern bekannt sein. Das Sitzungsmanagement ist auf das Wesentliche reduziert. Ein Teilnehmer erzeugt eine Sitzung, indem er einen Namen vergibt und Sitzungseinstellungen vornimmt. Jeder weitere Teilnehmer kann durch Direkteingabe des Sitzungsnamens dieser Sitzung beitreten.
Nach dem Sitzungsstart stehen den Benutzern eine Reihe vordefinierter Anwendungen zur Verfügung. Als Demonstrationsanwendungen stehen u.a. ein Chattool, ein Zeichentool, ein Votingtool und ein verteilter Taschenrechner zur Verfügung. Gestartete Anwendungen werden immer global für alle geöffnet. Schließt einer der Teilnehmer eine Anwendung, so wird diese für alle geschlossen.
Die Menge der Anwendungen kann durch Eigenentwicklungen erweitert werden. Das Hauptkonzept, durch das eine Verteilung erreicht wird, ist eine verteilte Ereignisverwaltung. Wird ein Benutzerereignis ausgelöst, so wird nicht nur eine lokale Behandlungsroutine aufgerufen, sondern, über den Habanero-Server, die Behandlungsroutine aller Teilnehmer. Im einfachsten Fall bedeutet dies, daß eine vorhandene Anwendung ohne große Änderung für alle Benutzer zur Verfügung steht, indem einfach alle Benutzereignisse über diesen Verteilungsmechanismus laufen. In diesem Fall würde es sich um eine kollaborationstransparente Anwendung handeln, bei der jedes Ereignis, egal von welchem Teilnehmer ausgehend, global wirkt.
Für Anwendungen, bei denen nicht jeder Benutzer zu jeder Zeit die Anwendung bedienen darf, stellt Habanero einen Mechanismus zur Zugriffskontrolle bereit. Ein Beispiel sind Zwei-Personen-Spiele, bei denen immer nur ein Spieler ziehen darf. Hier wechseln sich zwei Teilnehmer mit der Bedienung des Systems ab. Habanero bietet die Möglichkeit, beliebige Mechanismen selbst zu programmieren und in das System zu integrieren.

2.6.2 DOLPHIN

DOLPHIN ([SGH+94], [Dol96] und [GMD]) ist ein kooperatives Hypermedia-System. Der Schwerpunkt liegt mehr auf der Bearbeitung von Hypermedia-Dokumenten als auf der Verteilung in der Gruppe. Das System ist in VisualWorks Smalltalk realisiert.
Das System benötigt keine Server-Anwendung. Ein Klient, der ein Dokument zum ersten Mal öffnet, wird automatisch zum *Master* für dieses Dokument. Jeder weitere Klient, der dasselbe

Dokument bearbeiten möchte, muß die Netzwerkadresse des Masters kennen. Der Master übernimmt die Serverfunktionalität für alle weiteren Klienten.
Mit einem verteilter Dokumenteneditor können nun Hypermedia-Dokumente bearbeitet werden. Neben Texten und Freihand-Grafiken können Bitmap-Bilder beliebig auf einer Seite positioniert werden. Weitere Seiten können geöffnet und über Links miteinander verbunden werden. Neben einem öffentlichen Bereich, der für alle Teilnehmer denselben Inhalt zeigt, können beliebig viele private Fenster geöffnet werden.
Mit DOLPHIN erstellte Dokumente können gespeichert und zu beliebigen Zeitpunkten erneut geladen und bearbeitet werden.

2.6.3 Share-Kit

Share-Kit ([TUB], [Jah95] und [Edl93]) ist eine auf UNIX basierende Plattform, um in C geschriebene Programme in einer verteilten Umgebung zu starten. Share-Kit ist zentral organisiert, d.h. wie bei Habanero muß ein ausgezeichneter Rechner eine Server-Anwendung starten. Ein potentieller Teilnehmer stellt zuerst den entsprechenden Server ein und startet dann eine Anwendung. Ist auf dem Server eine entsprechende Anwendung von einem anderen Teilnehmer gemeldet, so wird der aktuelle Status an das neue Gruppenmitglied übertragen. Alle folgenden Operationen sind dann für alle Teilnehmer sichtbar.
Share-Kit beinhaltet kein Sitzungsmanagement im strengen Sinn. Eine Sitzung existiert implizit, wenn ein beliebiger Teilnehmer eine Anwendung startet, ohne daß diese explizit angemeldet oder eingerichtet werden muß. Es gibt daher auch kein Sitzungsprofil, das bestimmte Eigenschaften von Sitzungen spezifiziert.
Das Verteilungskonzept basiert auf der Möglichkeit, ausgewiesene Prozeduren so aufzurufen, daß sie für jeden Teilnehmer ausgeführt werden (Multicast-RPC). Da diese Eigenschaft nicht in der C-Syntax verankert ist, wurde eigens dafür ein Präprozessor entwickelt, der entsprechende Prozedurrümpfe anlegt.

2.6.4 Rendezvous

Rendezvous [HBP+93] ist ein streng zentral aufgebautes System. Auf der Basis von X-Windows wurde ein Verteilungsmechanismus konzipiert. Ein zentraler Rechner stellt alle Fensterdaten einer Sitzung mit dem X-Protokoll für die Klienten bereit. Die Klientenrechner bauen ausgehend von diesen Informationen ihre Fenster auf und senden die Benutzeraktionen an den Server.
Den Teilnehmern ist es erlaubt, unterschiedliche Darstellungen auf denselben Daten einzustellen. So kann z.B. eine statistische Verteilung als Histogramm oder als Kuchendiagramm aufgerufen werden, je nach Wunsch des Teilnehmers. Der Server stellt für jeden Klienten die gewünschte Darstellungsmethode bereit.
Zur Programmierung der Darstellungsmethoden wurde eine eigene Sprache entworfen, die *Rendezvous-Sprache*. Sie basiert auf Common Lisp und ist objektorientiert. Zwei wesentliche Merkmale müssen bei einer verteilten Anwendung programmiert werden: *constraints* und *event handlers*. Mit constraints werden Programmteile spezifiziert, die die Konsistenz der gemeinsamen Daten erhalten. Mit event handlers werden Behandlungsroutinen für Benutzerereignisse realisiert.
Anwendungsentwickler werden durch eine Klassenbibliothek unterstützt. Bestandteile der Bibliothek sind grafische Primitive, vordefinierte Darstellungsmethoden, sowie Dialog- und Mehrbenutzerelemente für die grafische Benutzungsoberfläche.

2.6.5 Groupkit

Groupkit ([GK], [RG96] und [GR96]) ist ein umfangreiches Paket zur Entwicklung verteilter Anwendungen. Es basiert auf TCL-TK. Eine Bibliothek entlastet den Anwendungsentwickler

von Standardaufgaben. Die Dienstleistungsbereiche *Sitzungsmanagement, Kommunikation* und *verteilte Benutzerführung* werden abgedeckt. Den Anwendungen stehen Funktionen zur Verfügung, um den Gruppenstatus bzw. Informationen zu den Teilnehmern zu erfragen.
Groupkit ist größtenteils dezentral aufgebaut. Nur während des Sitzungsrendezvous' wird ein zentraler Server, der *Registrar*, benötigt. Ist das Rendezvous abgeschlossen, erfolgt die Kommunikation dezentral.
Als gruppenspezifische Dialogelemente stehen verteilte Mauszeiger und Spezialelemente für Gruppenanwendungen zur Verfügung. Ein Spezialelement, der Multi-User-Scrollbar, wird in diversen Demo-Applikationen verwendet. Es handelt sich dabei um einen Scrollbar, bei der jeder Benutzer die Balken der anderen sieht und somit auf deren Bildausschnitte schließen kann. Es gibt Funktionen, um Bildausschnitte zu synchronisieren. Z.B. kann ein Teilnehmer immer den Bildausschnitt einstellen, den ein spezieller anderer Teilnehmer ausgewählt hat.
Die Groupkit-Umgebung wird mit einer Vielzahl von Demonstrationsanwendungen ausgeliefert. So ist neben einigen verteilten Texteditoren und Zeichentools auch ein verteilter Webbrowser zu finden.
Ein herausragendes Merkmal von Groupkit ist, daß die Sitzungsverwaltung angepaßt werden kann. Das Sitzungsmanagement ist ein Building Block des Groupkit-Systems und kann von Anwendungsentwicklern erweitert oder neu realisiert werden.

2.6.6 Zusammenfassung der Beispiele

Aus den oben vorgestellten Plattformen nimmt DOLPHIN eine gesonderte Stellung ein. DOLPHIN ist ein System mit einer einzigen festen Gruppenanwendung, einem kooperativer Hypermedia-Editor. Dieser kann zwar für unterschiedliche Aufgaben eingesetzt werden, ein Anwendungsentwickler wird aber nicht unterstützt, beliebige Anwendungen für diese Plattform zu realisieren.
Alle vorgestellten Ansätze zeichnen sich alle durch eine große Geschlossenheit und Vollständigkeit aus. Doch durch diese Geschlossenheit ist es problematisch, konzeptionelle Änderungen einzubringen. Habanero, Share-Kit und Rendezvous besitzen eine zentrale Architektur. Groupkit und DOLPHIN sind dezentral organisiert, benötigen aber für bestimmte Dienste zentrale Instanzen bzw. Rechner mit festen Netzwerkadressen.
Die Sitzungsverwaltung ist bei den meisten Plattformen auf das Wesentliche reduziert. Groupkit bietet die Möglichkeit, die Sitzungsverwaltung anzupassen oder zu erweitern.
Die Unterstützung der Rendezvous-Phasen ist bei allen Systemen rudimentär entwickelt.

3 Die DreamTeam-Umgebung

DreamTeam ist eine CSCW-Umgebung für synchrone Gruppenarbeit. Sie soll vorerst als Forschungs- und Testplattform dienen, ein späterer Einsatz in einer „realen" Umgebung (z.B. zur Unterstützung eines Softwarepraktikums) soll aber vorbereitet werden. Gesichtspunkte wie die Verwendung von PCs über Modems werden bei der Realisierung beachtet.

Bei der Konzeption von DreamTeam sollen folgende Merkmale erfüllt werden:
- einfache Bedienbarkeit
- größtmögliche Plattformunabhängigkeit
- einfaches Wiederaufsetzen bei Verbindungsabbrüchen
- Unterstützung mehrerer Dialogsprachen
- möglichst dezentraler Betrieb

Der erste Punkt sollte für Software immer zutreffen, die von einer Benutzergruppe ohne besondere Vorkenntnisse verwendet wird. Die Plattformunabhängigkeit resultiert aus der Tatsache, daß die typische Benutzergruppe über PCs mit unterschiedlichen Betriebssystemen verfügt. Zusätzlich zu PCs soll das System aber auch auf UNIX-Workstations lauffähig sein, da es auch universitätsintern eingesetzt wird. Ein Wiederaufsetzen im Fehlerfall sollte ohne besondere Verfahrensweisen möglich sein, da dies in der zu erwartenden Einsatzumgebung häufiger der Fall sein kann.
Die Verwendung mehrere Dialogsprachen ist teilweise schon ein Standardmerkmal von Softwaresystemen. Oft muß sich der Benutzer aber vor oder während der Installation eines Pakets für eine Landessprache entscheiden. In DreamTeam kann diese Entscheidung zur Laufzeit getroffen und beliebig revidiert werden.
Die Wahl für eine dezentrale Architektur wird später ausführlich begründet.

Folgende Funktionsbereiche werden von der DreamTeam-Umgebung abgedeckt:
- Verwalten von Benutzer-, Rechner- und Sitzungsprofilen
- Sitzungsabgleich mit anderen Rechnern
- Durchführung des Sitzungsrendezvous'
- verteilte Benutzerkontrolle
- Verteilung von Ereignissen
- Behandlung von Späteinsteigern
- weitere Built-In-Dienste

Entwickler von verteilten Anwendungen werden durch eine Funktionsbibliothek und eine Laufzeitumgebung unterstützt.

3.1 Architektur

Die DreamTeam-Umgebung ist dezentral aufgebaut, d.h. es gibt keine Server-Anwendung. Folgende Gründe waren dafür ausschlaggebend:
- Der Verwaltungsaufwand zum Betreiben eines Servers sollte vermieden werden. Server müssen besondere Kriterien bzgl. der Ausfallsicherheit genügen. Zusätzlich müssen Server im Rahmen einer Versionsumstellung ständig aktualisiert werden. Benutzergruppen müssen eingerichtet und die zugehörigen Zugriffsberechtigungen vergeben werden. Dieser Pflegeaufwand fällt bei einer dezentralen Architektur weg.
- Eine zentrale Architektur erfordert für jedes Ereignis in der Gruppe einen Nachrichtentransport zu dem Server und wieder zurück zu den Klienten. Dieser Weg kann in bestimmten Situationen beträchtlich länger sein als die Verbindung der Teilnehmer direkt untereinander. Zusätzlich müssen bei servergestützten Systemen auch lokale Ereignisse in der Regel über den Server laufen, während dezentrale Systeme diese intern behandeln können. Im Rahmen einer Echtzeitkommunikation ist eine replikative Struktur effizienter.
- Letztendlich stehen einem bei einer dezentralen Architektur andere Kommunikationswege offen. So ist z.B. eine Direktschaltung mehrerer Teilnehmer über ISDN ohne Server denkbar.

Bei der Entscheidung für eine dezentrale Architektur wurde ein komplexeres Laufzeitsystem in Kauf genommen.

3.2 Laufzeitsystem

Eine der Grunddienste eines CSCW-Systems ist die Verwaltung der verschiedenen Profile. DreamTeam verwaltet folgende:

Benutzerprofil:
Das Benutzerprofil beinhaltet alle Daten, die den Benutzer in einer Sitzung ausweisen, also Name, Adresse, Emailadresse und sein Bild. Ein Benutzer kann sein Profil beliebig ändern. Anhand von Zeitstempeln wird bei einer Sitzung festgestellt, ob sich ein Profil geändert hat und ggf. aktualisiert. Mit Hilfe von Übersichtsfunktionen können alle Teilnehmer die Profile der anderen Benutzer einsehen.

Rechnerprofil:
Im Rechnerprofil sind rechnerspezifische Daten festgehalten. Darunter fällt der Rechnertyp, das Betriebssystem und die Netzwerkadresse, wenn diese fest eingestellt ist. Als weiteres kann die Anbindungsart (Anbindung lokal oder über Einwählpunkt) angegeben werden. Das Rechnerprofil wird vom System automatisch generiert. Einige Einträge können vom Benutzer editiert werden.

Sitzungsprofil:
Im Sitzungsprofil ist festgehalten, welche Anwendungen in einer Sitzung verwendet werden sollen und ob es Beschränkungen für Benutzer geben soll. Beispielsweise kann es sinnvoll sein, nur eine maximale Anzahl von Benutzern zu einer Sitzung zuzulassen oder den Moderator bei einem Zutrittswunsch vorher zu fragen. Das Sitzungsprofil kann von dem Benutzer geändert werden, der eine Sitzung erzeugt hat. Andere Teilnehmer können das Sitzungsprofil nur einsehen.

Beim Start der DreamTeam-Anwendung werden vorliegende Profile von dem lokalen Dateisystem eingelesen. Danach wird eine Laufzeitumgebung mit folgenden Tasks gestartet:

Logmanager:
Der Logmanager verwaltet Meldungen über Ereignisse, die im System anfallen. Er gibt sie an den Dialog weiter und schreibt sie, wenn vom Benutzer gewünscht, in eine Datei. Letzteres dient zur Auswertung von Fehlersituationen während der Entwicklungsphase oder der nachträglichen Analyse von Sitzungen. Über die Benutzungsoberfläche sind Systemmeldungen in Form von Übersichtslisten abrufbar. Besondere Ereignisse (z.B. Fehler) führen zu Nachrichtenfenstern, die der Benutzer quittieren muß.

Verbindungsmanager:
Hier werden die Verbindungen verwaltet, die von den verteilten Anwendungen verwendet werden. Jede verteilte Anwendung ist mit der entsprechenden Anwendung aller Gruppenteilnehmer verbunden. Der Auf- und Abbau dieser Verbindungen wird vom Verbindungsmanager automatisch vorgenommen.

Sitzungsmanager:
Der Sitzungsmanager verwaltet die Sitzungen des lokalen Systems. Dazu gehört z.B. der Abgleich mit anderen Rechnern. Der Sitzungsmanager führt auch die Sitzungsstarts durch und aktiviert damit die verteilten Anwendungen.

Transfermanager:
Neben spontanen Ereignissen, die in Echtzeit eine Beantwortung erfordern, gibt es Anforderungen, deren Ausführungen längere Zeit in Anspruch nehmen dürfen. Dazu gehören Dateiübertragungen zwischen Rechnern. Der Transfermanager bietet Dienstleistungen zur Dateiübertragung an. Beispielsweise wird das Bild eines Benutzers, der einer Sitzung neu beitritt, über diesen Kanal versendet. Der Transfermanager kann auch von verteilten Anwendungen genutzt werden.

3.3 Sitzungsverwaltung

Ein wichtiger Grunddienst eines CSCW-Systems ist die Sitzungsverwaltung. Da DreamTeam keine zentrale Instanz besitzt, muß die Sitzungsverwaltung dezentral ablaufen. Für eine bestimmte Sitzung existiert ein ausgezeichneter Teilnehmer, der *Urheber* der Sitzung. Der Urheber ist derjenige Teilnehmer, der die Sitzung generiert hat, der also das Sitzungsprofil erstellt hat. Der Urheber stellt für alle Sitzungsteilnehmer eine Anlaufstelle für den Sitzungsbeitritt und -austritt dar. Erst wenn der Urheber eine Sitzung gestartet hat, können andere Teilnehmer der Sitzung beitreten. Tritt der Urheber aus der Sitzung aus, so können die anderen Teilnehmer zwar noch weiterarbeiten, es können aber keine weiteren Teilnehmer beitreten. Abbildung 1 zeigt die Übergänge eines Sitzungszyklus. Hierbei sind die einzelnen Zustände jeweils unterteilt in den Zustand des Urhebers und den Zustand eines beliebigen anderen Teilnehmers.

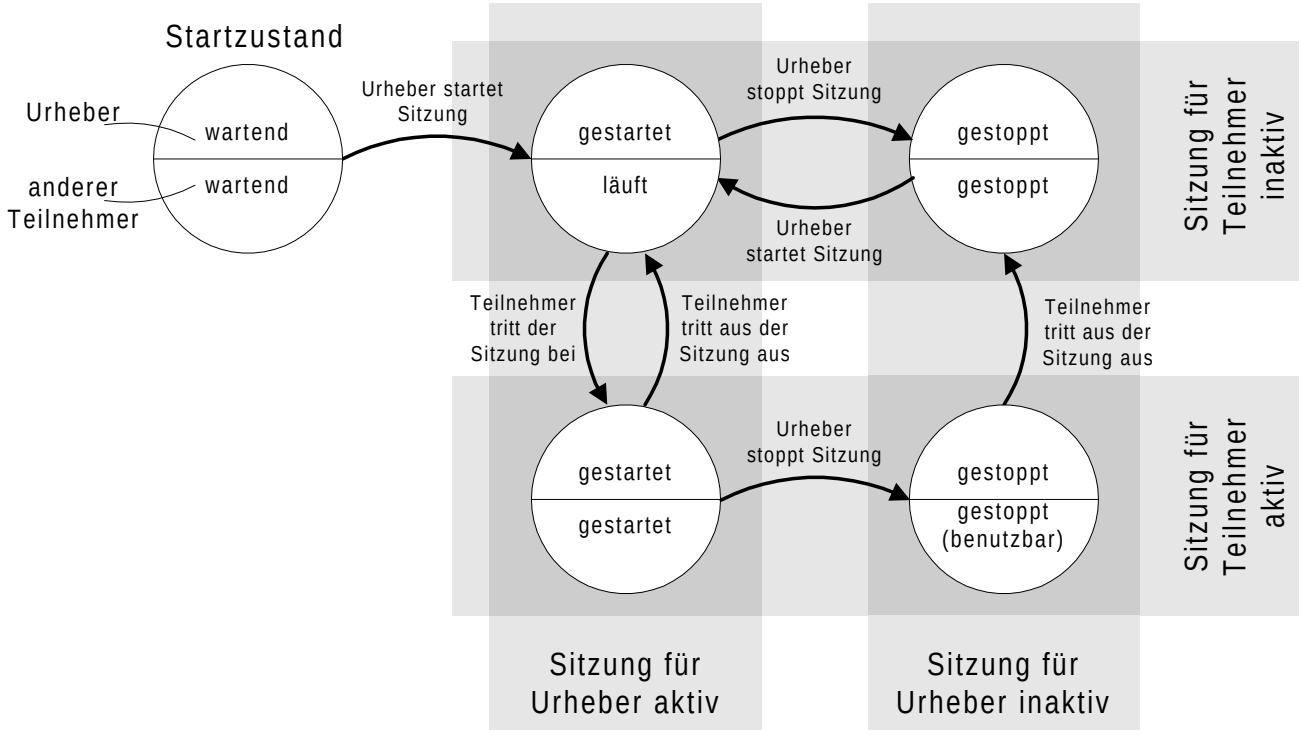

Abbildung 1: Lebenszyklus einer Sitzung

Die Zustände *wartend* und *gestoppt* geben an, daß diese Sitzung zur Zeit inaktiv ist. *Gestoppte* Sitzungen sind mindestens einmal gelaufen, während *wartende* Sitzungen seit der Generierung durch den Urheber noch nie aktiv waren.
Die Zustände *gestartet* und *läuft* zeichnen eine aktive Sitzung aus. *Gestartet* ist eine Sitzung, wenn die verteilten Anwendungen auf den lokalen Rechnern geöffnet sind. Eine Sitzung mit

dem Status *läuft* gibt an, daß eine Sitzung schon vom Urheber gestartet ist, ein potentieller Teilnehmer ihr aber noch nicht beigetreten ist.

3.4 Verteilungskonzept

Es gibt viele Möglichkeiten der Informationsverteilung in einem Gruppensystem. Eine häufig benutzte Form ist Multicast-RPC. Dabei führt ein spezieller Aufruf eines lokalen Systems dazu, daß auf allen Systemen eine Prozedur aufgerufen wird. Hierzu werden die Parameter in geeigneter Form versendet und dem Aufruf mitgegeben. Mit dieser Methode können verschiedene Aufgabenbereiche abgedeckt werden.
DreamTeam verwendet eine spezielle Form von Multicast-RPC. Ein bestimmter Aufruf mit einer vordefinierten Parameterliste führt zu einem verteilten Aufruf auf allen Teilnehmersystemen. Die feste Parameterzahl könnte im ersten Ansatz als Einschränkung gesehen werden. Die DreamTeam-Umgebung ist jedoch objektorientiert aufgebaut und an einer Parameterposition können beliebige Objekte stehen. Hiermit ist es möglich, auch komplexe Strukturen, als Objekt verpackt, an alle Teilnehmer zu versenden. Als einzige Einschränkung muß für das Objekt gelten, daß es *serialisierbar* ist. Serialisierbare Objekte haben die Fähigkeit, ihre Datenstrukturen über einen Datenkanal zu versenden und am Zielort wieder aufzubauen.
Mit dem Konzept serialisierbarer Objekte wird auch das Problem der Späteinsteiger gelöst. Späteinsteiger sind Sitzungsteilnehmer, die erst der Sitzung beitreten, nachdem schon Aktionen von anderen Teilnehmern ausgeführt wurden. Sie finden also einen nichtinitialen Sitzungszustand vor, der erst übertragen werden muß. Die Objekte, die eine Sitzung repräsentieren, sind in DreamTeam serialisierbare Objekte, so daß sie komplett an die Späteinsteiger übertragen werden können. Nachdem diese die Datenstrukturen komplett geladen und aufgebaut haben, verfügen sie über den aktuellen Sitzungszustand und können von nun an über die Ereignisverteilung schritthaltend den internen Sitzungszustand fortschreiben.

3.5 Kommunikationskanäle

DreamTeam verfügt über eine ganzen Satz von Kommunikationskanälen, die nach Bedarf aufgebaut werden. Die Aufteilung in verschiedene Kanalarten ist begründet in unterschiedlichen Nutzungsprofilen für unterschiedliche Aufgaben. Folgende Kommunikationskanäle stehen zur Verfügung:

Verbindungskanäle:
Über diese Kanäle werden die Aktivitäten der verteilten Anwendungen während der Sitzungen übertragen. Es handelt sich um schnelle Verbindungen, da Ereignisse der Anwendungen in Echtzeit ohne nennenswerte Verzögerungen verteilt werden müssen. Sonst wären Irritationen der Benutzer die Folge. Als Übertragungsmedium wurden permanent geöffnete Socketverbindung gewählt. Das schnellere, jedoch unsichere Medium der Datagramme wurde bewußt vermieden, da dann Konsistenzprobleme zu behandeln wären. Sendeaufrufe werden blockierend ausgeführt. Es findet lediglich eine Pufferung statt, um Sender und Empfänger zumindest minimal zu entkoppeln. Der Verwaltungsaufwand für Threads, die eine nicht blokkierende Sendeoperation bereitstellen, wurde dadurch eingespart. Zusätzlich bekommt der Sender sofort eine Erfolgskontrolle über seinen Sendebefehl. Die Verbindungskanäle werden beim Start einer Sitzung bzw. beim Beitritt eines Benutzers aufgebaut und bleiben für die Dauer der Teilnahme bestehen. Hiermit wird die Zeit für Auf- und Abbau gespart.

Sitzungskanäle:
Über diese Kanäle werden Abgleichoperationen der teilnehmenden Rechner durchgeführt, also

- Verteilung der Sitzungsprofile
- Verteilung der Statuswechsel
- Durchführung und Verteilung der An- und Abmeldungen der Benutzer

Vom Wesen her sind diese Operationen weniger zeitkritisch als die Verbindungsereignisse. Sie werden daher von Tasks niederer Priorität behandelt. Der Sendevorgang wird im Hintergrund durchgeführt, ist also nicht blockierend. Es wird ein Thread generiert, der die Sendung durchführt. Die Nachricht über Erfolg oder Mißerfolg der Sendung übergibt der Thread dem System über eine definierte Schnittstelle.
Sitzungskanäle werden nicht permanent gehalten, sondern nur bei Bedarf aufgebaut und nach Beendigung einer Transaktion wieder terminiert.

Transferkanäle:
Sie stellen die langsamste Verbindungsart bereit. Übertragungen (i.d.R. Dateitransfers) können minutenlang dauern und sollten das System wenig belasten. Deshalb bekommen die behandelnden Prozesse die niedrigste Priorität zugeordnet. Analog zu Sitzungsverbindungen findet ein Sendevorgang nicht blockierend statt. Auch hier wird die Verbindung nur bei Bedarf aufgebaut.

Eine Übersicht über die Kanäle ist im folgenden Bild wiedergegeben.

zeitkritisch

zeitliche Relevanz

unkritisch

Kanäle	verwaltet durch	Senden	Priorität	Auf-/Abbau
Verbindungs-kanäle	Verbindungs-manager	im Vordergrund	hoch	einmal pro Sitzung
Sitzungs-kanäle	Sitzungs-manager	im Hintergrund	mittel	bei jeder Transaktion
Transfer-kanäle	Transfer-manager	im Hintergrund	niedrig	bei jeder Transaktion

Abbildung 2: Kommunikationskanäle

In der weiteren Entwicklung könnte sich die Notwendigkeit ergeben, einen noch schnelleren Kanal einzuführen. Sollen z.B. Audio- und Videofunktionen in die Umgebung integriert werden, sind völlig neue Anforderungen zu erfüllen. Hierzu reichen die oben beschriebenen Kanalarten nicht aus.

3.6 Struktur der DreamTeam-Klassenbibliothek

Die DreamTeam-Umgebung wurde mit Hilfe einer hierarchischen Klassenbibliothek realisiert. Sie umfaßt folgende Ebenen:

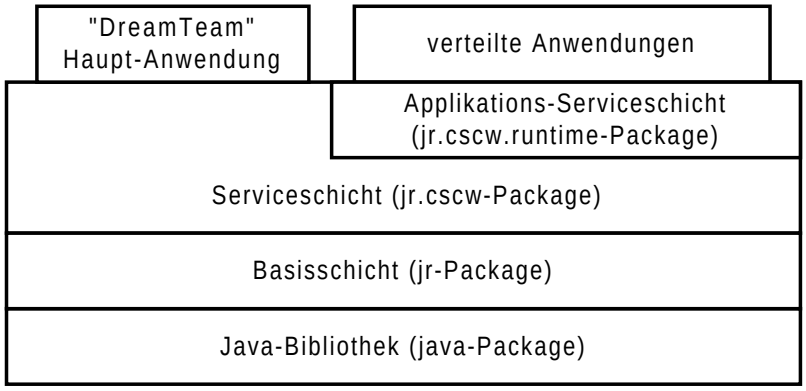

Abbildung 3: Ebenen der Klassenbibliothek

Die Realisierung wurde vollständig in Java (JDK1.02) [Sun] vorgenommen. Ein Basispaket (jr-Package) realisiert grundlegende Funktionen, die in der Java-Bibliothek z.Zt. fehlen. Das Basispaket beinhaltet noch keine CSCW-spezifischen Klassen und könnte somit auch für andere Zwecke eingesetzt werden.
Die Serviceschicht (jr.cscw-Package) ist dagegen auf eine CSCW-Umgebung ausgerichtet. Hier befinden sich die Datenstrukturen der oben beschriebenen Profile, die Programmcodes der Tasks sowie der Dialoge zur Konfiguration und Bedienung des Systems.
Neben der allgemeinen Serviceschicht wurde eine applikationsspezifische Serviceschicht (jr.cscw.runtime-Package) realisiert, die auf die allgemeinen Belange der verteilten Anwendungen zugeschnitten ist. Die oberste Schicht wird letztendlich durch das Hauptprogramm der DreamTeam-Applikation und die jeweiligen verteilten Anwendungen dargestellt.
Zur Zeit umfaßt die gesamte Klassenbibliothek 175 Klassen mit über 2000 Methoden. Sie ist komplett in Eigenentwicklung entstanden und wird ständig erweitert.

3.7 Benutzungsoberfläche

3.7.1 Plattformunabhängigkeit

Da die Umgebung plattformunabhängig ausgeführt ist, mußte bei der Realisierung der Oberfläche der kleinste gemeinsame Nenner der Dialogelemente aller Plattformen verwendet werden. Im wesentlichen wurden die bekannten Elemente wie Menüs, Listboxen, Eingabefelder, Checkboxen und Knöpfe verwendet. Ein wesentliches Element, von dem in dieser Umgebung verstärkt Gebrauch gemacht wird, fehlt jedoch als Standardelement: das Ikonen-Feld. Es wird in Systemen wie OS/2 oder Windows95 vielfach benutzt, ist jedoch nicht standardisiert. Deshalb wurde ein Ikonen-Feld-Element mit den Standard-Ausdrucksmitteln der graphischen Oberflächen nachimplementiert.
Bei der Realisierung der Umgebung wurde die Plattformunabhängigkeit dadurch gewährleistet, daß auf drei Plattformen gleichzeitig getestet wurde. Diese waren *Solaris*, *Windows95* und *OS/2*. Prinzipiell dürften zwar keine Unterschiede zwischen den Plattformen auftreten, es hat sich jedoch gezeigt, daß manche Details unterschiedlich wiedergegeben werden. Unvermeidlich können Unterschiede sein, wenn auf einem System z.B. bestimmte Schriftarten nicht vorhanden sind oder graphische Ausprägungen von Dialogelementen in einem bestimmten Format vorliegen müssen. Solche Unterschiede können zu Formatierungsfehlern in Dialogfenstern führen, wenn die Implementierung zu stark auf eine einzige Plattform ausgerichtet ist.
Andere Unterschiede liegen in den speziellen Eigenschaften der Betriebssysteme begründet. So werden beispielsweise Pfad- und Dateinamen unterschiedlich gebildet. Einige Systeme verwalten verschiedene Laufwerke im Dateisystem, andere hängen alle Verzeichnisse unter

ein einziges Stammverzeichnis. Solche Unterschiede müssen von der Laufzeitumgebung flexibel abgefangen werden.
Durch das parallele Entwickeln auf drei Plattformen ist gewährleistet, daß Probleme durch plattformspezifische Besonderheiten früh erkannt werden.

3.7.2 Dialogsprache

Die Dialogsprache soll vom Benutzer gewählt werden können. Hiermit können alle systemgenerierten Meldungen in der gewünschten Landessprache ausgegeben werden. Zusätzlich werden alle Dialogelemente mit Texten, also Menüs, Knöpfe, Labels etc., landesspezifisch ausgelegt. Zur Zeit sind die Sprachen Englisch und Deutsch implementiert. Prinzipiell können beliebige Sprachen hinzugefügt werden, wenn sie durch Unicode-Zeichen darstellbar sind.
Der Unicode-Zeichensatz [Uni] gewährleistet eine plattformübergreifende Darstellung länderspezifischer Sonderzeichen. Zur Zeit existiert eine Unterstützung der lateinischen Schriftzeichen mit den europäischen Sonderzeichen (deutsche Umlaute, französische Akzents etc.). Dies entspricht dem Zeichensatz nach ISO-8859-1. Eine Ausweitung auch auf nichtlateinische Zeichen (z.B. chinesische) ist jedoch für die Zukunft geplant.

3.7.3 Beschreibung der wichtigsten Fenster

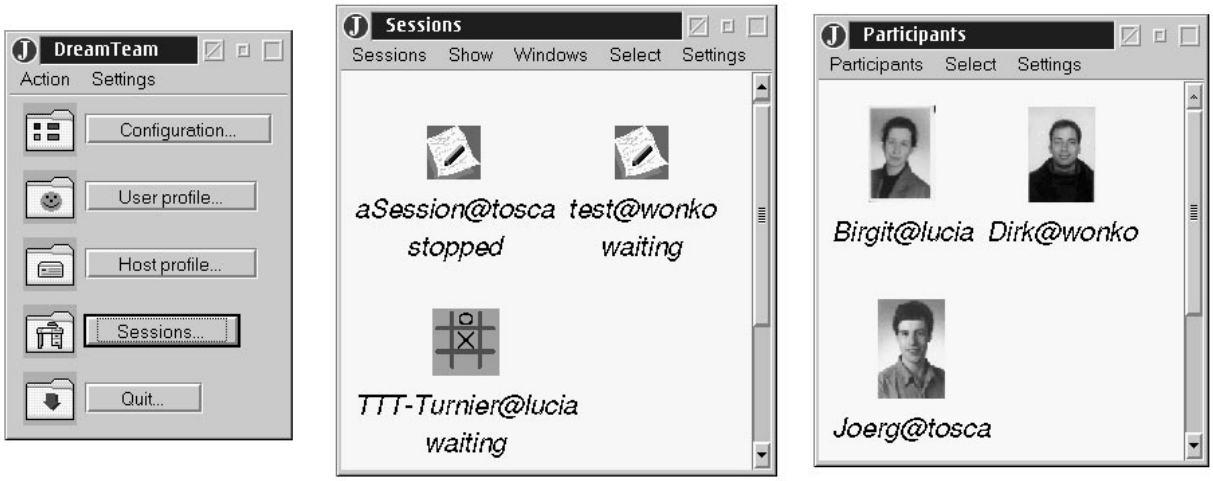

Abbildung 4: Die wichtigsten Fenster der DreamTeam-Anwendung

Nachdem alle Tasks der Anwendung erfolgreich gestartet sind, gelangt man in das Hauptfenster der Anwendung (Abbildung 4 links). Dieses Fenster ist während der Laufzeit der Anwendung immer offen.
Folgende Funktionen können hier ausgeführt werden:

- Konfigurieren des Systems
- Erstellen des Benutzerprofils des lokalen Benutzers
- Erstellen des Rechnerprofils des lokalen Hosts
- Öffnen des Sitzungsfensters
- Beenden des Systems

Die Konfigurationsarbeiten werden in der Regel nur beim initialen Einrichten des Systems durchgeführt. Die Funktion *Sitzungen* ist die wichtigste Funktion für die laufende Arbeit.
Nach dem Anwählen wird das Sitzungsfenster (Abbildung 4 Mitte) geöffnet. In dem Sitzungsfenster sind alle Sitzungen als Ikonen dargestellt, die dem System bekannt sind. Wird ein fremdes System im Rahmen eines Rendezvous' kontaktiert, so werden die lokal vorliegenden Sitzungslisten ausgetauscht, so daß nachher jedes System die Sitzungsprofile des anderen

kennt. Im Sitzungsfenster können wahlweise alle Ikonen oder nur die Ikonen der lokal definierten Sitzungen angezeigt werden.
Der Benutzer kann Sitzungsprofile erstellen, ändern oder löschen. Sitzungen können immer nur vom Urheber modifiziert werden, Sitzungen anderer Rechner werden nur angezeigt. Der Urheber einer Sitzung kann diese starten und öffnet damit die verteilten Anwendungen. Benutzer anderer Rechner können nun der gestarteten Sitzung beitreten. Nach dem Start oder Beitritt öffnet sich neben den Anwendungsfenstern ein weiteres Fenster, das alle zur Zeit teilnehmenden Benutzer anzeigt (Abbildung 4 rechts).
Verläßt ein Benutzer die Sitzung, so wird auch seine Ikone gelöscht. Liegt dem lokalen System ein Bild des Benutzers vor, so wird als Ikone dieses Bild benutzt. Liegt kein Bild vor, so wird versucht, von dem System des Benutzers ein Bild zu laden. Schlägt das fehl, so wird eine Standard-Ikone verwendet.

3.8 Built-In-Dienste

Zusätzlich zu dem oben beschriebenen Grundumfang an Diensten wurden weitere Funktionen in die Plattform integriert. Diese Dienste können von Anwendungsentwicklern genutzt werden. Einige Dienste wurden in die DreamTeam-Fenster integriert und können damit von jedem Teilnehmer genutzt werden, ohne daß sie in verteilte Anwendungen integriert sein müssen.

Nachrichtendienst:
Ein einfacher Nachrichtendienst ist in das Teilnehmerfenster integriert. Selektiert man eine oder mehrere Teilnehmerikonen, kann an die entsprechenden Teilnehmer eine Textnachricht versendet werden. Diese Funktion ist unabhängig von einer Chat-Anwendung, die eventuell in der Sitzung gestartet wurde, verfügbar. Erhält ein Teilnehmer eine Nachricht, so öffnet sich automatisch ein entsprechendes Fenster. In diesem kann direkt eine Antwort eingegeben werden, die der ursprünglichen Sender erhält.

Dateiübertragungen:
Eine weitere Funktion des Teilnehmerfensters ist die Möglichkeit, Dateien zu versenden. Ein Teilnehmer selektiert einen anderen Benutzer und wählt die Funktion zur Dateiübertragung aus. Danach öffnet sich eine Dialogbox, mit der die zu versendende Datei ausgewählt werden kann. Nach der Bestätigung wird der andere Teilnehmer über eine Dateiübertragung informiert. Er kann nun entweder diese Übertragung verweigern oder ein Verzeichnis auswählen, in das die Datei geschrieben werden soll. Es ist einem fernen Teilnehmer somit nicht möglich, lesend auf ein lokales Dateisystem zuzugreifen. Die Übertragung erfolgt grundsätzlich nur in der oben beschriebenen Richtung.

Verteilte Mauszeiger:
Die Möglichkeit, seinen eigenen Mauszeiger für andere Benutzer sichtbar zu machen bzw. andere Mauszeiger in den lokalen Fenster darzustellen, ist in die Umgebung integriert. Der Anwendungsentwickler muß lediglich bei der Realisierung eines Fensters eine bestimmte Fensterklasse benutzen, die diese Eigenschaften zur Verfügung stellt. Über spezielle Aufrufe kann gesteuert werden, ob und in welcher Form die Verteilung erfolgen soll. Die Übertragung der Mauspositionen und der Mausereignisse erfolgt mit einem eigenen Protokoll über den Verbindungskanal.

3.9 Beispielanwendungen

Derzeit sind folgende Beispielanwendungen mit Hilfe der DreamTeam-Umgebung realisiert worden:

- ein „Hello World“-Programm
- ein Tick-Tack-Toe-Spiel
- eine Chat-Anwendung
- ein Zeichentool

Das „Hello World“-Programm ist keine verteilte Anwendung im eigentlichen Sinne. Es dient nur als Demonstration bzw. kann als Grundgerüst für die Implementierung eigener Anwendungen benutzt werden. Über einen Knopf kann ein Anwender seine eigene Kennung an die Gruppe verteilen.

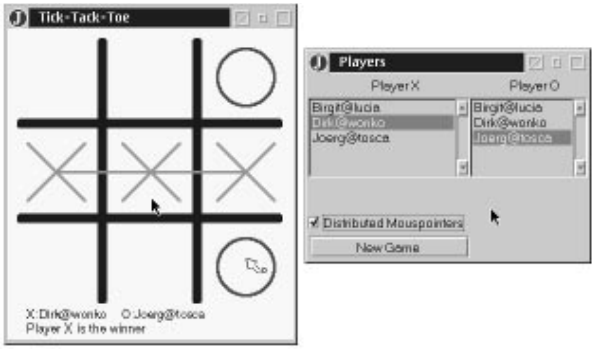

Abbildung 5: Die Tick-Tack-Toe-Anwendung

Die Tick-Tack-Toe-Anwendung (Abbildung 5) war ursprünglich eine Ein-Benutzer-Anwendung, bei der ein einzelner Benutzer gegen den Computer spielt. Durch einige Modifikationen konnte diese Anwendung einer Gruppe zugänglich gemacht werden, wobei jetzt beliebige Teilnehmer gegeneinander spielen können. Der Urheber der Sitzung übernimmt eine besondere Rolle. Er bekommt neben dem Spielfeld ein zusätzliches Fenster (Abbildung 5 rechts). In diesem kann er die Spieler auswählen, die gegeneinander spielen sollen. Alle Teilnehmer, die nicht für das Spiel ausgesucht wurden, können nur zusehen. Hat der Urheber „verteilte Mauszeiger“ gewählt, so werden neben dem eigenen Mauszeiger auch die Mauszeiger der Spieler angezeigt.

Eine Chat-Anwendung ist Bestandteil von fast jeder CSCW-Umgebung. Meistens ist das Aussehen und die Funktionalität ähnlich. Über ein Eingabefeld kann ein Text verfaßt werden, der über eine Schaltfläche an alle Teilnehmer verteilt wird.

Abbildung 6: Zeichentool

Als letzte Beispielanwendung wird ein Zeichentool angeführt (Abbildung 6). Mit der gehaltener Maustaste können Linien gezeichnet werden. Über eine Löschfunktion ist partielles Löschen möglich. Zeichenoperationen werden sofort an die anderen Teilnehmer verteilt. Jeder Teilnehmer kann entscheiden, ob der lokale Mauszeiger an die Gruppe verteilt werden soll oder nicht. Zusätzlich kann entschieden werden, ob auf dem lokalen Zeichenblatt andere veröffentlichte Mauszeiger dargestellt werden sollen. Hiermit hat jeder Benutzer die maximale Flexibilität für die Darstellung verteilter Mauszeiger.

3.10 Vergleich mit anderen Plattformen

Ein direkter Vergleich mit anderen Plattformen ist in der Regel schwierig, da mit jeder Realisierung eine eigene Zielsetzung verfolgt wurde. Trotzdem soll hier der Versuch unternommen werden, DreamTeam gegenüber den anderen vorgestellten Plattformen abzugrenzen.

Die Plattformen Habanero, Share-Kit und Rendezvous sind zentral organisiert. Sie setzen voraus, daß auf einem bekannten Server eine zentrale Anwendung gestartet wurde, bevor der erste Teilnehmer aktiv werden kann. DreamTeam hingegen ist mit seiner streng dezentralen Organisation für den typischen Einsatz im Rahmen der Fernuniversität besser geeignet. Wie diskutiert, ergeben sich bei einer dezentralen Struktur eine Reihe von Vereinfachungen. Ein Vorteil ist, daß die geringen Bandbreiten bei der beschriebenen Zugangsart besser ausgenutzt werden. Zusätzlich fällt der Verwaltungs- und Wartungsaufwand eines zentralen Servers weg, der besondere Eigenschaften bzgl. Ausfallsicherheit besitzen muß. Die Vorteile einer dezentralen Struktur werden allerdings mit einem erhöhten Aufwand bei der Sitzungskontrolle erkauft.
Der Wegfall einer zentralen Instanz erfordert zusätzliche Dienste. So ist in DreamTeam die Möglichkeit des Dateitransfers direkt in die Plattform eingebaut. Hiermit ist es den Teilnehmern möglich, auch ohne einen FTP-Server Dokumente auszutauschen.
Mit den verschiedenen Kanalarten in der DreamTeam-Umgebung wurde zusätzlich versucht, den Erfordernissen des typischen Einsatzes gerecht zu werden. So wurden aufgabenspezifisch verschiedene Datenkanäle bereitgestellt. Dadurch daß die Priorität der Kanäle und die Dauer der Verbindungen konkret für die jeweilige Aufgabe ausgerichtet sind, können die vorhandenen Ressourcen besser ausgenutzt werden. Die Plattform kann leichter um neue Protokollfamilien erweitert werden, da durch die verschiedenen Kanalarten eine Kapselung der unterschiedlichen Dienstleistungsbereiche vorgenommen wurde.
Die Implementierung von DreamTeam wurde mit einer objektorientierten Klassenbibliothek auf der Basis von Java durchgeführt. Java besitzt eine große Verbreitung und existiert schon für viele Plattformen, insbesondere für PC-Plattformen. In einige Betriebssysteme ist Java integriert, so daß keine zusätzliche Installation erforderlich ist. Java ist somit für den Einsatz im Rahmen der Fernuniversität besonders geeignet. Bei der Implementierung wurden keinerlei Voraussetzungen bzgl. der Umgebung gemacht. So wird z.B. kein X-Windows-System oder UNIX-Netzwerk vorausgesetzt. Hiermit unterscheidet sich DreamTeam wesentlich von anderen Plattformen. Rendezvous beispielsweise baut auf dem X-Protokoll auf. Share-Kit ist nur für UNIX-Umgebungen vorgesehen. Groupkit benötigt eine Plattform, auf der TCL-TK läuft. Z.Zt. liegt der Schwerpunkt von TCL-TK auf UNIX-Systemen.
Die Entwicklung auf der Basis einer objektorientierten Klassenbibliothek begünstigt die Wartbarkeit und die Weiterentwicklung des Systems. Das Prinzip der Vererbung wird an vielen Stellen der Plattform ausgenutzt. Insbesondere Entwickler verteilter Anwendungen profitieren davon. Ein Anwendungsentwickler kann durch die Verwendung von Standardklassen bei der Realisierung einer neuen Anwendung schon viele Aufgabenbereiche abdecken. Es müssen nur die anwendungsspezifischen Teile realisiert werden. Hiermit ergeben sich schlanke Anwendungen, was wieder der optimalen Ausnutzung der Ressourcen zugute kommt. Im Vergleich dazu baut Share-Kit auf der Sprache C auf. C hat diverse Nachteile bezüglich der Sicherheit des resultierenden Codes und der Wartbarkeit der Bibliotheken. Zusätzlich unterstützt C keine Vererbung. Anwendungen für Groupkit werden in TCL-TK geschrieben. TCL-TK hat zwar eine umfangreiche Bibliothek, aber auch hier wird keine Vererbung unterstützt.
Die Benutzungsoberfläche von DreamTeam ist voll graphisch und ikonenbasiert. Das Element des Ikonen-Felds wurde eigens nachimplementiert. Die Entscheidung, den Aufwand für die Nachimplementierung in Kauf zu nehmen, war begründet in der Erkenntnis, daß Ikonen für ungeübte Benutzer eine intuitive Möglichkeit bieten mit einem Rechner zu interagieren. Zusätzlich ist eine Hypertext-Hilfefunktion in die Umgebung integriert, die den unerfahrenen Bediener bei der Arbeit unterstützen soll. Das Hilfesystem kann auch von verteilten Anwendungen genutzt werden. Bei den Plattformen Dolphin, Share-Kit, Groupkit und Habanero ist keine integrierte Hilfefunktion vorhanden.

Die Möglichkeit, verschiedene Landessprachen in der Umgebung einzustellen, wird zur Zeit nur von DreamTeam bereitgestellt. In die Umgebung integriert ist ein Mechanismus, der für alle systeminternen Texte (auch Beschriftungen von Menüs und Knöpfen) die landesspezifische Variante benutzt. Darüber hinaus kann der Landes-Eintrag im Benutzerprofil auch von anderen Benutzern eingesehen, bzw. von verteilten Anwendungen automatisch ausgewertet werden. So kann z.B. eine Chat-Anwendung automatisch eine Landessprache vorschlagen bzw. Englisch empfehlen, wenn die Teilnehmer nicht alle dieselbe Landessprache sprechen.
Dem Gruppenrendezvous fällt im Rahmen der Fernuniversität eine besondere Rolle zu. Die vorgestellten Plattformen bieten keine speziellen Dienstleistungen für die Einbindung von Teilnehmern an, die über einen Einwählpunkt mit dem Netzwerk verbunden sind. Die Integration solcher Benutzer stellt jedoch besondere Anforderungen an die Umgebung, so daß die Realisierung eines tragfähigen Konzeptes notwendig ist.

4 Zukünftige Entwicklungen

Die nächsten Entwicklungen umfassen schwerpunktmäßig folgende Bereiche:

- Erforschung und Systematisierung der Rendezvous-Phase
- Konkretisierung der Applikations-Serviceschicht
- Entwicklung von weiteren verteilten Anwendungen
- Integration von computerunterstützter Kommunikation

Zur Zeit befindet sich im Prototypen noch kein hinreichend tragfähiges Konzept zur Abwicklung eines Gruppenrendezvous'. In der Literatur beschriebene Verfahren über einen Well-Known-Server, der die Beitrittsanforderungen sammelt und den Sitzungsstart abwickelt, sind im Zusammenhang mit der Fernuniversität weniger geeignet. Wünschenswert wäre ein vollständig dezentrales System. Es wird versucht, ein solches Konzept zu finden und zu realisieren.
Parallel dazu sollen weitere verteilte Anwendungen realisiert werden. Anhand derer soll eine Anforderungsliste erstellt werden, so daß das Paket für die Applikations-Serviceschicht weiter konkretisiert werden kann. Dieses Paket soll für eine maximale Bandbreite denkbarer CSCW-Anwendungen eine Basis darstellen. Besonderes Augenmerk gilt dabei der Entwicklung von gruppenspezifischen Dialogelementen. Diese bestimmen in großem Maße die Mensch-Computer-Interaktion in verteilten Sitzungen und entscheiden damit über die Akzeptanz von verteilten Anwendungen.
Eine erste verteilte Anwendung wird ein Anzeigeprogramm für HTML-Dokumente sein. Hiermit soll es möglich sein, über den Inhalt von Hypertext-Dokumenten in einer Gruppe zu diskutieren. Neben der Möglichkeit, Links in der Gruppe zu verfolgen, soll jeder Teilnehmer Bemerkungen direkt in ein Dokument eintragen zu können.
Als letzter Punkt wird eine Integration von Diensten der computerunterstützten Kommunikation angestrebt. Als wichtigster Dienst ist hierbei eine Audio-Verbindung zu nennen, die über die Umgebung bereitgestellt und verwaltet werden soll.

5 Zusammenfassung

DreamTeam ist eine synchrone, gruppenbewußte CSCW-Umgebung, die auf die Belange von Benutzern abgestimmt ist, die über ein temporäres, störanfälliges Medium (Modemverbindungen) in einem Netzwerk miteinander verbunden sind. Dabei verfügen die Teilnehmer über heterogene Betriebssystemumgebungen, größtenteils auf der Basis von PCs. Während die Dien-

ste für verteilte Sitzungen schon existieren, müssen die Rendezvousphase noch systematisiert und die entsprechenden Leistungen realisiert werden.
Ein wesentliches Merkmal von DreamTeam ist die dezentrale Architektur. Hiermit wurde versucht, bestimmten Problemen zentraler Lösungsansätze entgegenzuwirken.
DreamTeam dient zur Zeit als Forschungs- und Testplattform. Als Schwerpunkt soll hiermit ein generelles Anforderungsprofil verteilter synchroner Anwendungen erstellt werden. Hierbei gilt den gruppenspezifischen Dialogelementen eine besondere Beachtung, da sie in starkem Maße auf die Bedienbarkeit verteilter Anwendungen einwirken.

Literatur

[AG94] Antunes P., Guimaraes N.,
Multiuser Interface Design in CSCW Systems,
BROADCAST Technical Report Series No. 71, ESPRIT Basic Research Project 6360, 1994

[Dol96] *DOLPHIN 2 Quick Reference Manual*
ftp://ftp.darmstadt.gmd.de/pub/ocean/software/dolphin/manual.ps, 1996

[Edl93] Edlich S.,
Software Cooperation with the Share-Kit: Influences of Semantic Levels on the Working Efficiency,
Vienna Conference on Human Computer Interaction VCHCI '93, Vienna, Austria, Sept. 20-22, 1993, 225-234

[GK] *Groupkit Home Page*
http://www.cpsc.ucalgary.ca/projects/grouplab/groupkit/

[GMD] *DOLPHIN Home Page*
http://www.darmstadt.gmd.de/publish/ocean/activities/internal/dolphin.html

[GR96] Greenberg S., Roseman M.
Groupware Toolkits for Synchronous Work,
Department of Computer Science, University of Calgary, Canada, Nov. 1996,
http://www.cpsc.ucalgary.ca/projects/grouplab/papers/Trends97/report.ps.gz

[HBP+93] Hill R. D., Brinck T., Patterson J. F., Rohall S. L., Wilner W. T.,
Rendezvous Language,
Communications of the ACM, Vol. 36, No. 1, Jan. 1993, 62-67

[HHW+96] Hazemi R., Hailes S., Wilbur Steve, Pitsika M., Wilbur Sylvia,
ACOL Project - Deliverable 1,
http://www.cs.ucl.ac.uk/staff/R.Hazemi/del1/del1.ps, Feb. 1996

[Jah95] Jahn P.,
Getting started with Share-Kit,
ftp://ftp.cs.tu-berlin.de/pub/local/kbs/share-kit/share-kit.tar.gz, 1995

[NCSA] *NCSA Habanero Homepage*
http://www.ncsa.uiuc.edu/SDG/Software/Habanero/HabaneroHome.html

[RG96] Roseman M., Greenberg S.
Building Real-Time Groupware with GroupKit, A Groupware Toolkit,
ACM Transactions on Computer-Human Interaction, Vol. 3, No. 1, Mar. 1996, 66-106

[Scho96] Schooler E. M.,
Conferencing and collaborative computing,
Multimedia Systems, Vol. 4, 1996, 210-225

[SGH+94] Streitz N. A., Geißler J., Haake J. M., Hol J.,
DOLPHIN: Integrated Meeting Support across LiveBoards, Local and Remote Desktop Environments,
Proceedings of the ACM '94 Conference on Computer Supported Cooperative Work (CSCW '94), Chapel Hill, North Carolina, Oct. 22-26, 1994, 345-358

[Sun] *JavaSoft Home Page*
http://java.sun.com

[TUB] *Share-Kit Home Page*
http://www.cs.tu-berlin.de/cs/research/english/projects/share-kit.html

[Uni] *Unicode Home Page*
http://www.stonehand.com/unicode.html