Verteilte interaktive Medien

Martin Mauve
Universität Mannheim
Lehrstuhl für Praktische Informatik IV
http://www.informatik.uni-mannheim.de/~mauve/

Abstract: In der vorliegenden Arbeit wird gezeigt, dass verteilte interaktive Medien eine homogene Medien-Klasse bilden. Insbesondere ermöglichen es Vertreter dieser Klasse, dass eine verteilte Benutzergruppe mit dem Medium selbst interagiert. Typische Beispiele für verteilte interaktive Medien sind Shared-Whiteboard-Systeme, Netzwerk-basierte Computerspiele und verteilte Virtual-Reality-Umgebungen. Um nachzuweisen, dass die Idee einer homogenen Medienklasse gültig ist, wird ein gemeinsames Medienmodell präsentiert. Weiterhin wird ein Anwendungsprotokoll mit dem Namen RTP/I vorgestellt, welches die Entwicklung von wiederverwendbarer Funktionalität in Form von generischen Diensten ermöglicht. Diese sind medienunabhängig und so für die gesamte Medienklasse einsetzbar. Im Rahmen der Arbeit wurden drei generische Dienste entwickelt: Konsistenzerhaltung, Sitzungsaufzeichnung und die Unterstützung von Nachzüglern. Um die Funktionsfähigkeit der einzelnen Bestandteile der Arbeit zu demonstrieren, entstand der Prototyp einer 3D-Telekooperationsanwendung, die zur Zeit von Siemens für den Einsatz in einem Call-Center-Produkt erweitert wird.

1 Einleitung

Bislang ging man im Bereich der Rechnernetze davon aus, dass es zwei große Medienklassen gibt, die über ein Computer-Netzwerk übertragen werden: diskrete Medien (Text und Bilder) sowie kontinuierliche Medienströme (Audio und Video). Für diese Medienklassen wurden Protokolle, Mechanismen und Dienste entwickelt, welche deren Transport über ein Rechnernetz unterstützen. Der Kernbeitrag der vorliegenden Arbeit ist die Identifikation einer weiteren Netzwerk-basierten Medienklasse: die Klasse der verteilten interaktiven Medien. Vertreter dieser Klasse erlauben einer räumlich verteilten Benutzergruppe die direkte Interaktion mit dem Medium selbst. Typische Beispiele für verteilte interaktive Medien sind Shared-Whiteboard-Systeme, Netzwerk-basierte Computerspiele und verteilte Virtual-Reality-Umgebungen.
Mit der Existenz einer dritten Medienklasse stellt sich die Frage, wie diese in geeigneter Weise durch Protokolle, Mechanismen und Dienste unterstützt werden kann. Um diese Frage zu beantworten wurde ein gemeinsames Medienmodell für verteilte interaktive Medien entwickelt. Dieses Modell ermöglicht die Diskussion von gemeinsamen Eigenschaften und Problemstellungen mit einem geeigneten Vokabular. Das gemeinsame Medienmodell wird in Abschnitt 2 vorgestellt.
Ein Medienmodell reicht jedoch noch nicht aus für die Entwicklung von wiederverwendbarer Funktionalität. Daher wird das “Real Time Application Level Protocol for Distributed Interactive Media” (RTP/I) in Abschnitt 3 eingeführt. Diese erlaubt die Entwicklung generischer Dienste. In Abschnitt 4 werden drei generische Dienste beschrieben: Konsistenzerhaltung, Sitzungsaufzeichnung und die Unterstützung von Nachzüglern.
Um nachzuweisen, dass die vorgestellten Konzepte tatsächlich in der Praxis einsetzbar

sind, wurde eine 3D-Telekooperationsanwendung mit dem Namen TeCo3D entwickelt. Diese folgt dem Medienmodell und verwendet RTP/I sowie die genannten generischen Dienste. TeCo3D wird in Abschnitt 5 präsentiert.

2 Medienmodell

Um die Eigenschaften einer Medienklasse zu verstehen und um in der Lage zu sein, über sie zu sprechen, ohne auf das Vokabular eines speziellen Mediums zurückgreifen zu müssen, ist ein geeignetes Medienmodell erforderlich. Ein solches Modell für die Klasse der verteilten interaktiven Medien wird im Folgenden dargestellt. Die hier präsentierten Elemente sind die wichtigsten Bestandteile des Medienmodells; eine ausführlichere Beschreibung findet sich in [Mau00b].

2.1 Zustände und Ereignisse

Ein verteiltes interaktives Medium besitzt einen *Zustand*. So ist beispielsweise der Zustand einer Shared-Whiteboard-Präsentation definiert über den Inhalt aller Seiten, die in der Präsentation vorhanden sind. Um diesen Zustand wahrnehmen zu können, benötigt ein Benutzer eine *Anwendung*. Die Anwendung hält eine lokale Kopie der relevanten Teile des Medienzustandes. Aus diesem Grund bezeichnet man die Architektur von Anwendungen für verteilte interaktive Medien auch als repliziert.

Für alle Anwendungen, die an einer Sitzung teilnehmen, sollten die lokalen Zustandskopien konsistent sein, d.h. alle Anwender sollten möglichst denselben Medienzustand wahrnehmen. Aus diesem Grund ist es notwendig, die lokalen Zustandskopien mit geeigneten Mechanismen zu synchronisieren.

Der Zustand eines verteilten interaktiven Mediums kann sich auf zwei verschiedene Arten ändern: entweder in Abhängigkeit von der Zeit oder durch Ereignisse. Zwischen zwei zeitlich aufeinanderfolgenden Ereignissen ist der Zustand eine deterministische Funktion in Abhängigkeit von der Zeit. Daher erfordern Zustandsänderungen, die durch das Verstreichen der Zeit hervorgerufen werden, in der Regel keine Kommunikation. Jede Anwendung kann diese Zustandsänderungen selbst berechnen. Ein typisches Beispiel für eine solche Zustandsveränderung ist die Animation eines Objektes.

Eine Zustandsveränderung, die keine deterministische Funktion in Abhängigkeit von der verstrichenen Zeit ist, wird als *Ereignis* bezeichnet. Typischerweise sind Ereignisse Interaktionen der Benutzer mit dem Medium. Eine Lenkbewegung in einem verteilten Autorennspiel ist ein Beispiel für ein Ereignis. Immer, wenn ein Ereignis auftritt, besteht die Gefahr, dass die lokalen Zustandskopien der verschiedenen Benutzer nicht mehr identisch sind. Daher erfordern Ereignisse sowohl die Übertragung von Informationen, als auch einen geeigneten Konsistenzerhaltungsmechanismus.

Die in verteilten interaktiven Medien verwendeten Mechanismen zur Konsistenzerhaltung müssen grundsätzlich anderen Ansprüchen genügen, als solche, die im Bereich der verteil-

ten Datenbanken verwendet werden. Hauptsächlich liegt dies darin begründet, dass sich der Zustand in Abhängigkeit von der Zeit verändern kann, und dass Benutzer in der Lage sind, den Zustand in kooperativer Weise zu beeinflussen. So könnten gewisse Objekte in einer verteilten Virtual-Reality-Umgebung beispielsweise nur dann bewegt werden, wenn zwei oder mehr Benutzer helfen, da diese Objekte "zu schwer" für einen einzelnen Benutzer sind. Beide Eigenschaften können in klassischen verteilten Datenbanken nicht ohne weiteres abgebildet werden. Weitere Unterschiede sind die hohe Anzahl der Repliken (eine für jeden Benutzer) sowie die kurze Verzögerungszeit, die von verteilten interaktiven Medien erwartet werden. Insgesamt erfordern diese Unterschiede Ansätze, die den Eigenschaften dieser Medienklasse angepaßt sind.

2.2 Sub-Komponenten

Um eine flexible und skalierbare Handhabung des Medienzustandes zu gewährleisten, ist es häufig notwendig, den Zustand in kleinere Teile zu zerlegen. Diese Teile werden im Folgenden als *Sub-Komponenten* bezeichnet. Die Zerlegung des Zustandes in Sub-Komponenten erlaubt es Anwendungen, nur diejenigen Teile des Zustandsraumes zu verfolgen, die für den lokalen Benutzer relevant sind. Beispiele für Sub-Komponenten in einer verteilten Virtual-Reality-Umgebung sind 3D-Objekte (ein Avatar, ein Auto, ein Raum). In einem Shared-Whiteboard-System können dies die einzelnen Seiten der Präsentation sein.

3 RTP/I

Das Real-Time Application Level Protocol for Distributed Interactive Media (RTP/I) setzt das Medienmodell in ein Anwendungsprotokoll um[MHKE01]. Es ermöglicht die Entwicklung von wieder verwendbarer Funktionalität in Form von *generischen* Diensten. RTP/I ist verwandt mit dem Real-Time Transport Protocol (RTP) [SCFJ00], welches hauptsächlich für die Übertragung von Audio und Video benutzt wird. Während RTP/I viele Eigenschaften von RTP wiederverwendet, wurde es doch sehr stark verändert, um für verteilte interaktive Medien optimal geeignet zu sein.

Aus zwei Gründen wurde für die Konkretisierung des Medienmodells ein Protokoll und nicht eine Programmierschnittstelle gewählt: (1) ein Protokoll ist vollständig unabhängig von der darüberliegenden Technologie, wie Programmiersprachen oder Betriebssysteme. (2) Der Engpass für verteilte interaktive Medien ist zumindest heutzutage das Netzwerk. Daher ist es unbedingt erforderlich, das verwendete Anwendungsprotokoll für die Medienklasse zu optimieren. Dieses Vorgehen schließt nicht aus, dass später auch die Programmierschnittstellen standardisiert werden. So ist es beispielsweise vorstellbar, für verteilte interaktive Medien CORBA über RTP/I anstelle des zur Zeit für CORBA verwendeten Internet Inter Orb Protocols (IIOP) einzusetzten.

RTP/I besteht aus zwei Teilen: dem Daten-Protokoll für die Übertragung von Ereignissen

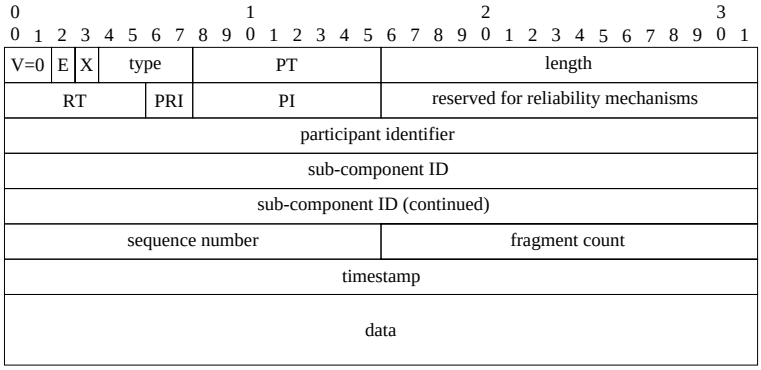

Abbildung 1: RTP/I-Datenpaket

und Zuständen, sowie dem Kontroll-Protokoll für die Übertragung von Meta-Informationen über die Teilnehmer einer Sitzung und die in dieser Sitzung enthaltenen Sub-Komponenten.

3.1 RTP/I-Daten-Protokoll

Das Daten-Protokoll von RTP/I stellt einen standardisierten Rahmen für die Übertragung von Zuständen und Ereignissen zur Verfügung. In dem standardisierten Rahmen sind Informationen enthalten, die für alle verteilten interaktiven Medien benötigt werden. Abbildung 1 zeigt die Datenfelder des Rahmens. Von besonderem Interesse sind die folgenden Informationen: *type* identifiziert den Nachrichtentyp (Ereignis oder Zustand), *sub-component ID* ist ein eindeutiger Bezeichner für die betroffene Sub-Komponente, *participant identifier* ist ein eindeutiger Bezeichner für den Teilnehmer, der diese Nachricht gesendet hat, *timestamp* gibt den Zeitpunkt an, zu dem die Nachricht gesendet wurde, und *data* enthält die eigentlichen Daten, welche in medienspezifischer Weise kodiert sind.
Mit diesen Informationen ist es möglich, die Semantik eines Medienstroms zu interpretieren, ohne die medienspezifische Kodierung von Ereignissen und Zuständen verstehen zu müssen. Weiterhin erlaubt es das Daten-Protokoll, den Zustand einer bestimmten Sub-Komponente in standardisierter Weise nachzufragen. Dies bildet die Grundlage für die Entwicklung generischer Dienste.

3.2 RTP/I-Kontroll-Protokoll

Das RTP/I-Kontroll-Protokoll (RTCP/I) vermittelt Informationen über die Teilnehmer einer Sitzung. Dies beinhaltet beispielsweise die Namen der Teilnehmer und deren E-Mail Adressen. Diese Informationen können für eine einfache Sitzungs-Kontrolle verwendet werden. Weiterhin stellt RTCP/I Informationen über die Sub-Komponenten bereit, die in einer Sitzung enthalten sind. Diese Informationen werden regelmäßig angekündigt. Sie

beinhalten drei verschiedene Informationen für jede Sub-Komponente: (1) den eindeutigen Bezeichner der Sub-Komponente, wie er auch im Daten-Protokoll verwendet wird, (2) einen Namen, welcher einer Anwendung oder einem Anwender erlaubt festzustellen, ob diese Sub-Komponente relevant ist, (3) einen Marker, der anzeigt, ob diese Sub-Komponente von wenigstens einer Anwendungsinstanz verwendet wird, um das Medium dem Benutzer darzustellen. Solche Sub-Komponenten werden als aktiv bezeichnet. Die sichtbare(n) Seite(n) einer Shared-Whiteboard-Präsentation wären beispielsweise aktive Sub-Komponenten. Diese Meta-Informationen können von generischen Diensten verwendet werden um die Semantik des Medienstromes besser zu verstehen.

4 Generische Dienste

Basierend auf dem Medienmodell und auf RTP/I wurden bereits drei generische Dienste entwickelt: ein Dienst für die Konsistenzerhaltung, ein Dienst für die Aufzeichnung von Sitzungen, sowie ein Dienst für die skalierbare Unterstützung von Nachzüglern.

4.1 Konsistenzerhaltung

Konsistenzerhaltung ist ein bekanntes Forschungsgebiet für verteilte interaktive Medien, die ihren Zustand nur aufgrund von Ereignissen und nicht durch das Verstreichen der Zeit verändern. Insbesondere für verteilte Texteditoren wurden sehr interessante Ansätze zur Konsistenzerhaltung entwickelt. Prinzipiell sorgen die bekannten Ansätze dafür, dass auf allen Ereignissen eine Reihenfolge definiert wird. Weiterhin erzwingen sie, dass nach Ausführung aller Ereignisse, jede lokale Zustandskopie so aussieht, als wären die Ereignisse in dieser Reihenfolge ausgeführt worden [SJZ+98].

Als Konsistenzdienst für die gesamte Klasse der verteilten interaktiven Medien ist ein solcher Ansatz ungeeignet, da im Allgemeinen der Zustand von Elementen dieser Medienklasse auch vom Ablauf der Zeit abhängt. Daher ist es von entscheidender Bedeutung, dass nicht nur auf die Reihenfolge der Ausführung von Ereignissen geachtet wird, sondern auch auf den Zeitpunkt der Ausführung. Für die betrachtete Medienklasse lässt sich das folgende *Konsistenzkriterium* definieren:

Ein verteiltes interaktives Medium ist genau dann konsistent, wenn zu einem Zeitpunkt, nachdem alle Ereignisse auf allen lokalen Zustandskopien ausgeführt wurden, alle lokalen Zustandskopien identisch mit dem Zustand sind, der erreicht worden wäre, wenn alle Ereignisse in der richtigen Reihenfolge und zum richtigen Zeitpunkt ausgeführt worden wären.

Im Rahmen der Dissertation entstand ein generischer Dienst, der dieses Konsistenzkriterium sicherstellt. Die fundamentale Idee ist hierbei diejenigen Ereignisse, die vom lokalen Benutzer erzeugt werden, solange zu verzögern, bis es wahrscheinlich ist, dass alle entfernten Anwendungen das Ereignis empfangen haben. Diese Methode wird als *lokale Verzögerung* (oder auf englisch *local lag*) bezeichnet [Mau00a]. Wenn die lokale Verzöge-

rung hoch genug gewählt wird, so können die Ereignisse gleichzeitig von allen Anwendungsinstanzen ausgeführt werden. Dies geschieht auf Kosten der Reaktionszeit der Anwendung, da die Benutzereingaben nicht sofort für den lokalen Benutzer sichtbar werden. Erste wahrnehmungspsychologische Experimente haben gezeigt, dass Verzögerungswerte von bis zu 80 ms von den Benutzern generell nicht wahrgenommen werden können. Weiterhin zeigte sich, dass Benutzer sich sehr schnell an höhere Verzögerungen zwischen 200 ms und 300 ms anpassen können. Außerdem ist bekannt, dass die Netzwerkverzögerung bei interkontinentalen Sitzungen typischerweise im Bereich von 100 ms bis 150 ms liegt. Der Einsatz einer geeigneten lokalen Verzögerung kann somit in vielen Fällen die Konsistenz in verteilten interaktiven Medien herstellen.

Obwohl die Einführung von lokaler Verzögerung das Einhalten des Konsistenzkriteriums wahrscheinlich macht, kann sie es nicht garantieren. So können beispielsweise Pakete in einem Netzwerk verloren gehen und bis zu deren erneuter Übertragung kann sehr viel Zeit vergehen. Es ist daher notwendig, einen geeigneten Reparaturmechanismus zu verwenden, der in diesen Ausnahmefällen für die Einhaltung des Konsistenzkriteriums sorgt. Prinzipiell gibt es verschiedene Ansätze, die diese Aufgabe erfüllen können [Mau00a] [Mau00c]. Der vorliegende generische Dienst sorgt mit Hilfe einer sogenannten Floor-Control dafür, dass höchstens ein Benutzer zu einem bestimmten Zeitpunkt Ereignisse für eine Sub-Komponente erzeugen darf [Mau00b]. Die Anwendung dieses Benutzers hält also notwendigerweise immer den Zustand der Sub-Komponente, welcher dem Konsistenzkriterium entspricht. Wird bei einer anderen Anwendungsinstanz die Verletzung des Konsistenzkriteriums festgestellt, so fordert diese den Zustand von der Anwendung an, die den korrekten Zustand hält.

Alle Bestandteile des Dienstes für die Konsistenzerhaltung wurden auf der Basis des Medienmodells unter Verwendung von RTP/I entwickelt und sind somit für beliebige verteilte interaktive Medien einsetzbar.

4.2 Skalierbare Unterstützung von Nachzüglern

Wenn eine Anwendung einer bereits bestehenden Sitzung mit einem verteilten interaktiven Medium betritt, dann ist es notwendig, dass die Anwendung mit dem Zustand der für sie relevanten Sub-Komponenten initialisiert wird. Bei einer Shared-Whiteboard-Präsentation muß eine Anwendung beispielsweise zumindest die Zustände der sichtbaren Seiten erhalten, bevor der Benutzer an der Sitzung teilnehmen kann.

Da verteilte interaktive Medien häufig in Sitzungen mit sehr vielen Teilnehmern verwendet werden, ist es notwendig, dass die Initialisierung von Nachzüglern in skalierbarer und effizienter Weise durchgeführt wird. Daher wurde basierend auf dem Medienmodell und RTP/I ein generischer Dienst mit diesen Eigenschaften für die Unterstützung von Nachzüglern entwickelt [VMG+00]. Die Architektur dieses Dienstes ist in Abbildung 2 dargestellt. Der Dienst analysiert die Daten, die von den anderen Anwendungsinstanzen übertragen werden. Da die Anwendungen hierfür RTP/I verwenden, ist der Dienst in der Lage, deren Semantik soweit zu interpretieren, wie es für die Unterstützung von

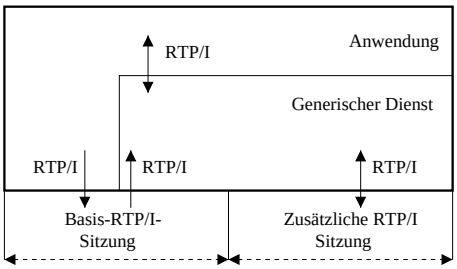

Abbildung 2: Architektur des generischen Dienstes zur Unterstützung von Nachzüglern

Nachzüglern notwendig ist. Kenntnis der medienspezifischen Kodierung von Zuständen und Ereignissen ist nicht notwendig. Nachdem der Dienst die Daten untersucht hat, leitet er sie unverändert an die Anwendung weiter.

Wenn eine Anwendung einer laufenden Sitzung beitritt, dann wird der Dienst durch das Kontroll-Protokoll RTCP/I von den in der Sitzung vorhandenen Sub-Komponenten erfahren. Immer wenn eine neue Sub-Komponente entdeckt wird, informiert der Dienst die Anwendung. Diese kann ihrerseits festlegen ob, und wenn ja, wann der Zustand dieser Sub-Komponenten nachgefragt werden soll. Diese Kriterien bezeichnet man als Politiken. Beispiele für Politiken sind: frage den Zustand sofort nach, erfrage den Zustand, wenn ein Ereignis für die Sub-Komponente auftritt, oder ignoriere die Sub-Komponente.

Sobald für eine Sub-Komponente die Bedingung eintritt, die von der ihr zugeordneten Politik beschrieben ist, wird der Dienst aktiv. Er erfragt den Zustand der Sub-Komponente über eine separate Sitzung. In dieser Sitzung sind nur diejenigen Anwendungen präsent, die selbst Nachzügler sind, sowie einige Anwendungen, die den Nachzüglern antworten. Die große Mehrzahl der Anwendungen ist daher von dem Vorgang nicht betroffen. Insbesondere wenn Gruppenkommunikation (Multicast) vom Netzwerk unterstützt wird, ist dieses Vorgehen sehr effizient. Von einer einzigen Anfrage und der darauf folgenden Übertragung des Zustandes können dann alle Anwendungen profitieren, die verspätet der Sitzung beigetreten sind.

Sollte in der Sitzung für die Unterstützung von Nachzüglern keine Anwendung präsent sein, die eine Anfrage beantworten kann, so wird ein Teilnehmer der regulären Sitzung aufgefordert, der Sitzung für die Unterstützung von Nachzüglern beizutreten und die gewünschten Informationen zu übertragen.

Sobald der Dienst eines Nachzüglers die Zustandsinformationen erhält, werden diese an die Anwendung weitergeleitet. Wenn über einen bestimmten Zeitraum hinweg keine neue Sub-Komponenten entdeckt werden und der Zustand aller relevanten Sub-Komponenten empfangen wurde, dann gilt der Nachzügler als in die Sitzung integriert. Folglich kann der Dienst die spezielle Sitzung für Nachzügler verlassen.

Mit Hilfe dieses generischen Dienstes können beliebige verteilte interaktive Anwendungen einer laufenden Sitzung auf skalierbare und effiziente Weise beitreten. Die Möglichkeit der Anwendung, für jede Sub-Komponente eine eigene Politik zu spezifizieren, erleichtert

dabei die Anpassung des Dienstes an die speziellen Bedürfnisse der Anwendung.

4.3 Aufzeichnung von Sitzungen

Das Aufzeichnen von Sitzungen wird aus vielerlei Gründen benötigt, sei es zur Archivierung, um Abwesenden die Möglichkeit zu geben, nachzuvollziehen, was während einer Sitzung geschehen ist, oder um im Streitfall nachweisen zu können, wie eine Sitzung abgelaufen ist. Das generische Aufzeichnen von Audio- und Videoübertragungen ist lange untersucht worden und ist inzwischen gut verstanden [Hol97]. Für verteilte interaktive Medien ist dies jedoch bislang nicht der Fall. Durch das Medienmodell und mit Hilfe von RTP/I ist es nun erstmals möglich geworden einen generischen Dienst zur synchronen Aufzeichnung von beliebigen RTP/I-basierten Sitzungen mit verteilten interaktiven Medien und RTP-basierten Audio- und Videoströmen zu entwickeln. Der Dienst erlaubt einen eingeschränkten wahlfreien Zugriff und benötigt keine Informationen über die medienspezifische Kodierung von Ereignissen oder Zuständen [HMKE99].

Während der Aufzeichnung speichert der Dienst die Ereignisse und Zustände, die in der Sitzung übertragen werden. Zusätzlich fordert er regelmäßig Zustandsübertragungen an, die später als Aufsetzpunkte für den wahlfreien Zugriff verwendet werden. Nachdem die Aufzeichnung beendet ist, können diese Informationen wieder abgespielt werden. Durch die Informationen im RTP/I-Rahmen ist es möglich, die korrekte Reihenfolge und den richtigen zeitlichen Abstand für das Abspielen zu bestimmen. Der Datenstrom, der auf diese Weise entsteht, kann von einer gewöhnlichen Anwendung empfangen und dargestellt werden, genauso wie der Datenstrom der während einer regulären Sitzung erzeugt wird.

Eine wesentliche Herausforderung für die Entwicklung eines generischen Aufzeichnungsdienstes ist die Bereitstellung von Funktionalität für einen eingeschränkten wahlfreien Zugriff. Das Hauptproblem bei der Realisierung eines wahlfreien Zugriffes ist, dass eine Anwendung mit einem geeigneten Zustand initialisiert werden muß, bevor sie den aufgezeichneten Medienstrom darstellen kann. Da der Dienst generisch ist, kann er diesen Zustand nicht selbst berechnen. Vielmehr ist es notwendig, aus dem aufgezeichneten Datenmaterial eine Sequenz von Zuständen und Ereignissen zu erzeugen, welche die Anwendung in den gewünschten Ausgangszustand versetzt. Sollte das Medium nur eine Sub-Komponente besitzen, ist dieser Vorgang sehr einfach: Es wird bis zu dem ersten Zustand vor dem Zugriffszeitpunkt zurück gegangen und dann mit dem Abspielen dieses Zustandes begonnen. Da dies in der Regel nicht genau der vom Anwender spezifizierte Zugriffszeitpunkt ist, spricht man von eingeschränktem wahlfreien Zugriff. Komplexere Mechanismen für Anwendungen mit mehreren Sub-Komponenten oder sehr großen Zuständen finden sich in [HMKE99].

5 TeCo3D

Basierend auf dem Medienmodell, RTP/I und den generischen Diensten wurden bislang drei Anwendungen entwickelt: ein Shared-Whiteboard-System [Vog01], ein einfaches verteiltes Computerspiel [Fri01] und eine Anwendung für Telekooperation mit 3D-Modellen [Mau99]. Letztere trägt den Namen TeCo3D und wird im Folgenden kurz vorgestellt.
TeCo3D erlaubt es einer räumlich verteilten Benutzergruppe, gemeinsam kooperationsunbewusste 3D-Modelle zu betrachten. Dabei bedeutet kooperationsunbewusst, dass die Modelle nicht speziell für einen verteilten Einsatz entwickelt werden müssen. Es können also beliebige 3D-Modelle, die mit Hilfe von VRML (Virtual Reality Modeling Language) spezifiziert wurden, von mehreren Benutzern gleichzeitig verwendet werden. Diese 3D-Modelle werden in der Regel von CAD-Anwendungen oder 3D-Editoren generiert und sind meist sowohl interaktiv als auch dynamisch, d.h. ihr Zustand ändert sich über die Zeit und durch Benutzerinteraktionen.
Für das 3D-Rendering wird in TeCo3D ein erweiterter VRML-Browser verwendet. Wie in verteilten interaktiven Medien üblich ist die Architektur von TeCo3D repliziert, d.h. eine Anwendungsinstanz läuft auf dem Computer eines jeden Anwenders. Für die Kommunikation kommt ein zuverlässiges Multicast-Transportprotokoll zum Einsatz.
Wenn ein Benutzer ein VRML-Objekt in den verteilten Arbeitsbereich einfügt, wird die Beschreibung dieses Objektes analysiert. Die Bestandteile, welche für die Benutzerinteraktion verantwortlich sind, werden durch spezielle Funktionalität ersetzt. Dies verwandelt das kooperationsunbewusste Objekt in ein kooperationsbewusstes Objekt. Benutzerinteraktionen mit dem Objekt stellen Ereignisse dar und werden von der eingefügten Funktionalität an alle Instanzen der Anwendung mit Hilfe von RTP/I übertragen und dort ausgeführt. Der Zustand des Mediums ist über eine Erweiterung des VRML-Browsers zugreifbar.
Das Medienmodell paßt sehr gut auf TeCo3D. Die Sub-Komponenten sind die einzelnen 3D-Modelle im verteilten Arbeitsbereich. Der Zustand eines solchen Modells ist der Zustand der entsprechenden Sub-Komponente, während die Benutzerinteraktionen Ereignisse darstellen. TeCo3D benutzt RTP/I als Anwendungsprotokoll und verwendet alle hier vorgestellten generischen Dienste. Ein Beispiel, wie ein Modell mit Hilfe von TeCo3D verteilt wird, ist in Abbildung 3 zu sehen. Ein voll funktionsfähiger Prototyp wurde in Java entwickelt und wird zur Zeit von Siemens für die Integration in deren Call-Center-Produktlinie erweitert.

6 Zusammenfassung und Ausblick

Die Kernaussage dieser Arbeit ist, dass es neben den bekannten Medienklassen der diskreten Medien und der kontinuierlichen Medienströme eine weitere Medienklasse gibt, die über Computernetzwerke übertragen wird: die verteilten interaktiven Medien. Es wurde gezeigt, dass es mit Hilfe eines Medienmodells und eines Anwendungprotokolls mit dem Namen RTP/I möglich ist, gemeinsame Probleme für die gesamte Medienklasse zu identi-

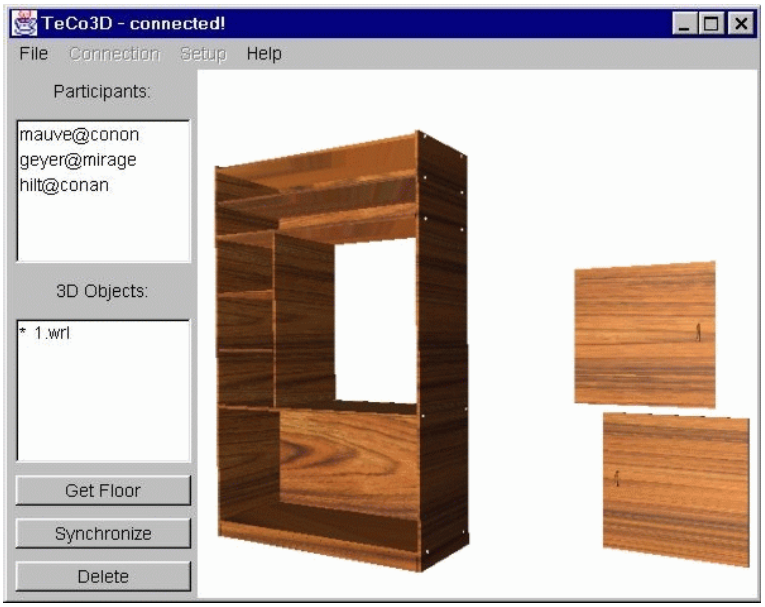

Abbildung 3: Screenshot der TeCo3D-Anwendung

fizieren und in Form von generischen Diensten zu lösen. Basierend auf diesem Fundament wurden bereits mehrere Anwendungen für verteilte interaktive Medien entwickelt. Beispielhaft wurde eine Anwendung für die Kooperation mit interaktiven und dynamischen 3D-Modellen näher betrachtet.

Zur Zeit wird insbesondere die Standardisierung von RTP/I weiter vorangetrieben. Ein überarbeiteter Internet Draft steht kurz vor der Vollendung. Die meisten Veränderungen ergaben sich dabei aus der Erfahrung mit der Entwicklung und dem Einsatz mehrerer Anwendungen für verteilte interaktive Medien, sowie dem Entwurf der generischen Dienste. Parallel dazu wird an einem geeigneten Transportprotokoll für diese Medienklasse gearbeitet. Sowohl TCP als auch UDP scheinen im hohen Maße ungeeignet. Ein geeignetes Transportprotokoll sollte den Echtzeitcharakter der Medienklasse unterstützen, passende Mechanismen zur Reparatur von Paketverlusten besitzen und Überlastkontrolle auf eine Art durchführen, die für verteilte interaktive Medien geeignet ist.

Literaturverzeichnis

[Fri01] M. Friedrich. Entwurf und Implementierung einer Beispielanwendung für die Synchronisation in verteilten interaktiven Anwendungen mit kontinuierlichen Zustandsänderungen. Master's thesis, Universität Mannheim, Lehrstuhl für Praktische Informatik IV, 2001.

[HMKE99] V. Hilt, M. Mauve, C. Kuhmünch, and W. Effelsberg. A generic scheme for the recording of interactive media streams. In *Proc. of the International Workshop on Interactive Distributed Multimedia Systems and Telecommunication Services (IDMS) 1999*, pages 291–304, 1999.

[Hol97] W. Holfelder. Interactive remote recording and playback of multicast videoconferences. In *Proc. of the International Workshop on Interactive Distributed Multimedia Systems and Telecommunications Services (IDMS) 1997*, pages 450–463, 1997.

[Mau99] M. Mauve. Teco3d: a 3d telecooperation based on vrml and java. In *Proc. of SPIE Multimedia Computing and Networking (MMCN) 1999*, pages 240–251, 1999.

[Mau00a] M. Mauve. Consistency in replicated continuous interactive media. In *Proc. of the ACM Conference on Computer Supported Cooperative Work (CSCW) 2000*, pages 181–190, 2000.

[Mau00b] M. Mauve. *Distributed Interactive Media*. PhD thesis, Praktische Informatik IV, Universität Mannheim, Germany, L15,16, D-68131 Mannheim, Germany, 2000.

[Mau00c] M. Mauve. How to keep a dead man from shooting. In *Proc. of the 7th International Workshop on Interactive Distributed Multimedia Systems and Telecommunication Services (IDMS) 2000*, pages 199–204, 2000.

[MHKE01] M. Mauve, V. Hilt, C. Kuhmünch, and W. Effelsberg. Rtp/i - toward a common application level protocol for distributed interactive media. *IEEE Transactions on Multimedia*, 3(1):152–161, March 2001.

[SCFJ00] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. Rtp: A transport protocol for real-time applications. Internet Draft: draft-ietf-avt-rtp-new-08.txt. Work in Progress., July 2000.

[SJZ+98] C. Sun, X. Jia, Y. Zhang, Y. Yang, and D. Chen. Achieving Convergence, Causality Preservation and Intention Preservation in Real-Time Cooperative Editing Systems. *ACM Transactions on Computer-Human Interaction*, 5(1):63–108, 1998.

[VMG+00] J. Vogel, M. Mauve, W. Geyer, V. Hilt, and C. Kuhmünch. A generic late join service for distributed interactive media. In *Proc. of the 8th ACM Conference on Multimedia (ACM MM) 2000*, pages 259–268, 2000.

[Vog01] J. Vogel. multimedia lecture board (mlb) homepage. URL: http://www.www.informatik.uni-mannheim.de/informatik/pi4/projects/ANETTE/anetteProject2.html, 2001.

Martin Mauve, wurde am 17. März 1971 in Köln geboren. Er erlangte sein Abitur 1990 am Hohenstaufengynasium Eberbach. Von 1991 an studierte er Wirtschaftsinformatik an der Universität Mannheim. Zwischen 1996 und 1997 war er Diplomand am International Computer Science Institute (ICSI) an der University of California in Berekely. Sein Studium schloß er 1997 mit einem Diplom in Wirtschaftsinformatik ab. Von 1997 bis 2000 war er Doktorand am Lehrstuhl für Praktische Informatik IV an der Universität Mannheim. Seine Arbeit als Doktorand beendete er erfolgreich 2000 mit einer Dissertation über das Thema “Distributed Interactive Media”. Zur Zeit ist er wissenschaftlicher Assistent an der Universität Mannheim.