



# Präsentation und Kollaboration in Televeranstaltungen mit dem multimedia lecture board

Jürgen Vogel

Lehrstuhl für Praktische Informatik IV, Universität Mannheim  
vogel@informatik.uni-mannheim.de

**Zusammenfassung:** Das multimedia lecture board (mlb) ist ein sogenanntes Shared Whiteboard und ermöglicht die Präsentation und die gemeinsame Bearbeitung von Dokumenten in computergestützten Videokonferenzen. Es wurde im Rahmen des DFN-Projekts "ANETTE" speziell für den Einsatz in Teleteaching-Veranstaltungen entwickelt und findet seit dem Sommersemester 2001 Verwendung an verschiedenen Universitäten. In diesem Artikel werden die wesentlichen Funktionsmerkmale des mlb sowie die technischen Besonderheiten vorgestellt.

## 1 Einleitung



In verteilten synchronen Lernszenarien wie beispielsweise Televorlesungen werden meist drei Medien eingesetzt, um die Lehrinhalte vom Dozenten zu den Lernenden an den entfernten Standorten zu übertragen: Audio, Video und Vortragsfolien. Für die Übertragung von Audio und Video werden idealerweise spezielle Hardware-Codecs verwendet (z.B. MPEG 2- oder H.263-Codecs), die die Medienströme (in Abhängigkeit von den vorhandenen Netzwerkressourcen) in hoher Qualität erfassen und wiedergeben. Bei fehlender Ausstattung können alternativ auch reine Software- (z.B. Microsoft NetMeeting oder die Mbone-Tools vic und rat) oder auch Mischlösungen zum Einsatz kommen. Die verwendeten Lehrdokumente (z.B. Vortragsfolien, Programmcode-Auszüge, etc.) lassen sich entweder per Application Sharing oder mit einem Shared Whiteboard zu den entfernten Teilnehmern übermitteln. Beim Application Sharing (z.B. mit Microsoft NetMeeting) kann der Bildschirminhalt beliebiger Anwendungen übertragen werden. Der zentrale Nachteil ist hierbei allerdings, dass die Interaktion zwischen den entfernten Standorten nur unzureichend unterstützt wird und die gemeinsame Bearbeitung von Dokumenten kaum möglich ist. Als Alternative empfiehlt sich daher die Verwendung eines kollaborationsbewußten Shared Whiteboard, das interaktives Präsentieren und gemeinsames Bearbeiten von Dokumenten in computergestützten Videokonferenzen ermöglicht.



Das multimedia lecture board (mlb) [1] ist ein solches Shared Whiteboard und wurde speziell für den Einsatz in Teleteaching-Szenarien entwickelt. Das mlb ist im Rahmen des vom DFN-Verein geförderten ANETTE-Projekts [2] an den Universitäten Mannheim und Freiburg entstanden und kommt dort seit dem Sommersemester 2001 routinemäßig vor allem in Televorlesungen zum Einsatz. Im folgenden Abschnitt wird zunächst die grundlegende Funktionalität des mlb zum Erstellen, Bearbeiten und Präsentieren von Lehrdokumenten beschrieben. Kapitel 3 geht auf die speziellen Funktionen des mlb ein, die die Benutzer bei der kollaborativen Gruppenarbeit unterstützen. In Kapitel 4 werden die technischen



Besonderheiten des mlb dargestellt, die sich aus dessen Peer-to-Peer-Architektur ergeben. Kapitel 5 beschreibt, wie sich mit Hilfe aufgezeichneter Lehrveranstaltungen Medieneinheiten zum asynchronen Lernen erstellen lassen. Der Artikel schließt mit einer Zusammenfassung und einem kurzen Ausblick.

## 2 Das multimedia lecture board

Das mlb stellt eine Vielzahl von Funktionen zur Verfügung, um Lehrdokumente zu erstellen, zu bearbeiten und zu präsentieren. Auf einer Seite eines Lehrdokuments können eine beliebige Anzahl von graphischen Objekten wie Rechtecken, Polygonzügen, Freihandlinien und Text erzeugt werden. Außerdem ist es möglich, Grafiken in allen gängigen Formaten (z.B. GIF, PNG, JPG, BMP, ...) zu importieren. Alle graphischen Objekte können nachträglich bearbeitet oder gelöscht werden, z.B. lässt sich die Linienfarbe eines Rechtecks verändern oder die Größe einer importierten Grafik. Dokumente können so entweder vorbereitet werden oder entstehen im Laufe einer Veranstaltung. Die Benutzeroberfläche des mlb ist in Abbildung 1 zu sehen. Im Arbeitsbereich wird gerade die Seite mit dem Titel "Animations" und den dazugehörigen Objekten angezeigt. Die Auswahl der Werkzeuge und der Objektattribute erfolgt über die Werkzeugleiste.

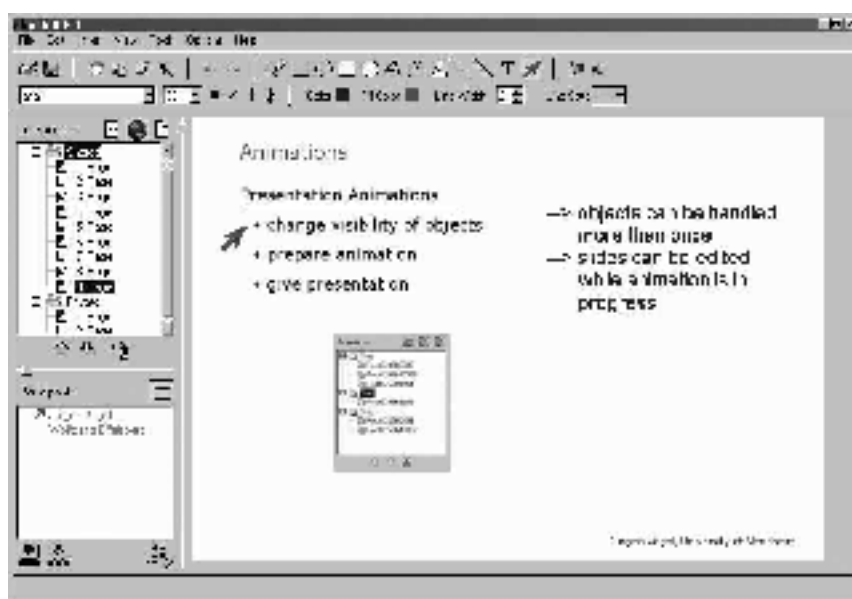


Abbildung 1: Benutzeroberfläche des mlb

Ein Benutzer kann mehrere Dokumente gleichzeitig bearbeiten, wobei für jedes Dokument ein eigener Arbeitsbereich existiert. Genau ein Dokument befindet sich dabei im gemeinsamen Arbeitsbereich, dem sogenannten Shared Workspace. Der Shared Workspace wird

über eine Gruppenadresse weltweit eindeutig identifiziert (vgl. Kapitel 3 und 4), so dass ein Benutzer, der an einer bestimmten Veranstaltung teilnehmen möchte, nur den Shared Workspace seines eigenen mlb mit der entsprechenden Adresse öffnen muss und so der Sitzung beitrifft. Alle Aktionen, die ein Benutzer im Shared Workspace ausführt, sind für alle Teilnehmer unmittelbar sichtbar. Dabei ist der Inhalt des gemeinsamen Arbeitsbereichs für jeden Teilnehmer identisch (“What You See is What I See - WYSIWIS”), und jeder Teilnehmer bekommt dieselbe aktuelle Seite angezeigt. Des Weiteren hat jeder Teilnehmer einer Sitzung die Möglichkeit, die Seiten und Objekte im Shared Workspace zu bearbeiten. Der Shared Workspace erlaubt also die Erstellung eines gemeinsamen Dokuments. Alle anderen Dokumente befinden sich in privaten Arbeitsbereichen, so dass alle Aktionen, die auf diesen Dokumenten ausgeführt werden, nur lokal sichtbar sind. Private Arbeitsbereiche werden hauptsächlich zur Vorbereitung von Dokumenten eingesetzt, so dass für die anderen Teilnehmer nur das Endergebnis sichtbar ist. In Abbildung 1 sind im Dokumentenbaum oben links ein privater und ein gemeinsamer Arbeitsbereich sichtbar.

Ein spezieller Modus erlaubt es, private Annotationen im Shared Workspace anzubringen, d.h. die so erzeugten Objekte sind nur lokal sichtbar. Dies kann in Televeranstaltungen beispielsweise von Zuhörern dazu verwendet werden, eine persönliche Mitschrift anzufertigen. Außerdem besteht die Möglichkeit, lokal im gemeinsamen Arbeitsbereich zu blättern, so dass der Benutzer eine Seite nachschlagen kann, ohne die anderen Teilnehmer zu stören.

Der Austausch von Inhalten (d.h. Seiten) zwischen privatem und gemeinsamen Arbeitsbereich ist jederzeit möglich und erfolgt am einfachsten über “Drag & Drop” - Operationen. Wird eine Seite so neu in den Shared Workspace eingebracht, wird diese automatisch an alle Teilnehmer übertragen (vgl. Kapitel 4). Falls ein Teilnehmer die Sitzung verlässt und so die Verbindung zu den anderen Benutzern abbricht, wird das Dokument des gemeinsamen Arbeitsbereichs automatisch in einen privaten Arbeitsbereich überführt, so dass die erzeugten Inhalte nicht verloren gehen. Mit Hilfe des in Kapitel 4.2 beschriebenen Mechanismus kann der Teilnehmer der Sitzung zu einem späteren Zeitpunkt wieder beitreten und den aktuellen Zustand des gemeinsamen Dokumentes erhalten. Dadurch kann ein Dozent beispielsweise nacheinander die Ergebnisse voneinander unabhängiger Arbeitsgruppen einsehen, die jeweils in einer eigenen Sitzung erstellt werden.

Größere Dokumente lassen sich mit Hilfe von Kapiteln hierarchisch gliedern, wobei Kapitel wiederum Unterkapitel enthalten können. Diese Hierarchie lässt sich mit Hilfe von “Drag & Drop” jederzeit umgestalten. Zum leichteren Auffinden von Seiten und Kapiteln im Dokumentenbaum können diese individuell mit Namen versehen werden.

Mit Hilfe von Präsentationsanimationen lassen sich die mit dem mlb gehaltenen Vorträge strukturieren oder bestimmte Sachverhalte visualisieren (z.B. der Ablauf eines Algorithmus). Eine Animation setzt sich dabei aus einer beliebigen Anzahl von Animationsschritten zusammen. Für jeden Animationsschritt kann der Benutzer eine Menge von graphischen Objekten definieren, die entweder ein- oder ausgeblendet werden sollen. Dabei ist es möglich, dass ein bestimmtes Objekt in mehreren Schritten Verwendung findet. Während der Präsentation kann der Benutzer dann die Animationsschritte nacheinander über die Maus oder die Tastatur gesteuert ausführen. Die Ein- oder Ausblendungen von graphischen Objekten sind dann für alle Teilnehmer sichtbar.

Mit dem mlb erstellte Dokumente werden in einem XML-kompatiblen Format abgespeichert, so dass die Bearbeitung von mlb-Dateien prinzipiell auch mit anderen Anwendungen möglich ist. Des Weiteren existiert eine Export-Funktion, um Dokumente im HTML-Format abzuspeichern. Alternativ kann ein Dokument auch seitenweise in einem gängigen Grafikformat exportiert werden. Der Import von Dokumenten, die mit anderen Programmen (z.B. mit Microsoft PowerPoint) erstellt wurden, erfolgt vorzugsweise auch über die unterstützten Grafikformate. Die einzelnen Seiten des Fremd-Dokuments bilden dann den Hintergrund auf den mlb-Seiten und können im weiteren Verlauf mit graphischen Objekten annotiert werden. Zur Arbeitserleichterung wurde eine spezielle Anwendung entwickelt, die aus den Einzelgrafiken eines Fremd-Dokuments automatisch ein mlb-Dokument erzeugt [1].

Die in Abbildung 1 gezeigte Benutzeroberfläche des mlb lässt sich individuell an die Bedürfnisse des Benutzers anpassen. Der Benutzer kann beispielsweise festlegen, welche Werkzeugeleiten sichtbar sind. Die Anzeigegröße einer Dokumentenseite ist beliebig einstellbar. Für die Präsentation von vorgefertigten Dokumenten empfiehlt sich die Verwendung des Vollbild-Modus. Für eine schnelle Navigation innerhalb eines Dokuments stellt das mlb außerdem eine Thumbnail-Übersicht zur Verfügung.

### 3 Unterstützung kollaborativer Gruppenarbeit

Um an einer mlb-basierten Videokonferenz teilzunehmen, benötigt ein Benutzer zum einen das mlb selbst und zum anderen die Gruppenadresse, durch die die Veranstaltung definiert ist. Dies bedeutet, dass im mlb keine zentrale Benutzerverwaltung existiert und die Teilnahme an einer mlb-Sitzung grundsätzlich jedem offen steht.

Das gemeinsame Dokument einer mlb-Sitzung kann, wie im letzten Kapitel beschrieben, von allen Teilnehmern verändert werden. Um die Kollaboration zwischen den Teilnehmern zu unterstützen und zu steuern, bietet das mlb zahlreiche Funktionen, die über die eigentlichen Editierfunktionen hinausgehen. Das Ziel dieser kollaborativen Funktionen ist es, ein Bewusstsein ("Awareness") für die anderen Teilnehmer und deren Aktionen zu erzeugen. Dies ist insbesondere bei größeren Veranstaltungen von zentraler Bedeutung, da hier die anderen Medien Audio und Video schnell an ihre Grenzen stoßen. Beispielsweise ist es für einen Dozenten bei einer größeren Anzahl von Zuhörern sehr schwierig, das Videobild jedes Einzelnen im Auge zu behalten und auf relevante Gesten (z.B. Wortmeldungen) hin zu beobachten. Im folgenden werden die kollaborativen Dienste des mlb kurz dargestellt.

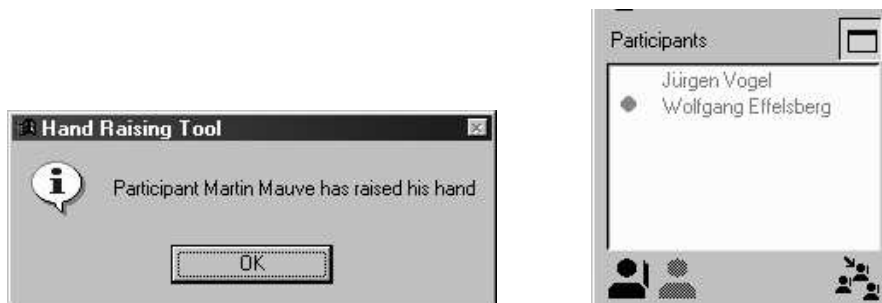
Alle mlb-Instanzen, die sich in einer bestimmten Sitzung befinden, tauschen untereinander allgemeine Teilnehmerinformationen wie z.B. Name und eMail-Adresse des jeweiligen Benutzers aus. Diese Informationen sind über den Namen eines Teilnehmers in der Teilnehmerliste unten links in der Benutzeroberfläche des mlb (siehe Abbildung 1) jederzeit abrufbar. Dadurch erhält jeder Teilnehmer einen Überblick über alle anderen Teilnehmer einer Sitzung. Mit Hilfe der Teilnehmerliste werden des Weiteren noch andere Informationen visualisiert, die das Bewusstsein für die Aktionen und den Status der anderen Teilnehmer erhöhen. Beispielsweise spiegelt sich die Aktivität eines Benutzers bezüglich Editieroperationen auf dem Shared Workspace in der Schriftfarbe seines Namens in der Teilnehmerliste wider - je dunkler, desto aktiver ist der entsprechende Benutzer.

Sollte ein Teilnehmer die Sitzung verlassen oder technische Probleme die Verbindung zu den anderen unterbrechen, wird dies über ein Symbol neben dem Namen des betroffenen Teilnehmers in der Teilnehmerliste visualisiert. Für den Benutzer ist diese Art der Rückmeldung wesentlich auffälliger, als den entsprechenden Teilnehmer einfach aus der Liste zu entfernen.

Um die Aufmerksamkeit der Teilnehmer auf einen Bereich der aktuellen Seite oder auf ein bestimmtes graphisches Objekt zu lenken, erlaubt das mlb die Verwendung eines Telepointers. Ein Telepointer dient somit der Visualisierung lokaler Zeigeoperationen, für die bei einer Einbenutzer-Anwendung der Mauszeiger verwendet würde. Letzterer ist bei einer Mehrbenutzeranwendung wie dem mlb für die entfernten Teilnehmer allerdings nicht sichtbar. Jeder Teilnehmer verfügt bei Bedarf über einen eigenen Telepointer. Damit diese voneinander unterscheidbar sind, kann die Farbe des Zeigers frei gewählt werden. Verwendet ein Teilnehmer einen Telepointer, wird dies auch in der Teilnehmerliste mit Hilfe eines Zeigersymbols neben dem entsprechenden Eintrag dargestellt. Dabei stimmt die Farbe des Symbols mit der des tatsächlichen Telepointers überein, so dass ein schneller Überblick gewährleistet ist. In Abbildung 1 verwendet beispielsweise gerade der Teilnehmer "Jürgen Vogel" einen (roten) Telepointer, um eine Textzeile hervorzuheben.

Da die Erfassung und Koordination von Wortmeldungen in Videokonferenz-Situationen mit vielen Teilnehmern mit Hilfe von Audio und Video nur schwer möglich ist, integriert das mlb ein Melde-Tool, das ebenfalls über die Teilnehmerliste gesteuert wird. Mit Hilfe spezieller Buttons kann ein Teilnehmer sich melden, seine Meldung zurückziehen, oder einen anderen Teilnehmer aufrufen, der sich zuvor gemeldet hat (siehe Abbildung 2(b) unten). Falls sich ein Teilnehmer meldet, wird dies den anderen Benutzern auf Wunsch über einen speziellen Dialog angezeigt (vgl. Abbildung 2(a)). Gleichzeitig werden alle Wortmeldungen, die zu einem bestimmten Zeitpunkt vorliegen, auch über die Teilnehmerliste durch einen blinkenden Punkt vor den jeweiligen Teilnehmer-Namen dargestellt (vgl. Abbildung 2(b)). Wird ein Teilnehmer von einem anderen aufgerufen, erhält dieser eine entsprechende Nachricht. Da die jetzige Version des mlb keine Verwaltung von Zugriffsrechten erlaubt, erfordert die Koordination von Wortmeldungen allerdings auch eine gewisse Disziplin der Teilnehmer.

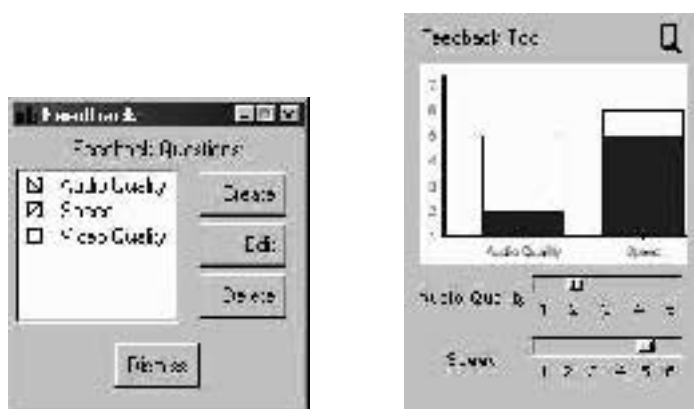
Insbesondere in größeren Televeranstaltungen kann es für den Dozenten schwierig sein, mit den anderen Teilnehmern direkt zu interagieren. Beispielsweise könnte der Dozent eine Übungsaufgabe stellen und müsste dann von jedem einzelnen Zuhörer das Ergebnis einholen. Um diesen Prozess zu vereinfachen, gibt es im mlb ein Feedback-Tool, das in allen Situationen verwendet werden kann, in denen ein Teilnehmer eine bestimmte Fragestellung von allen anderen Teilnehmern beantwortet haben möchte. In Abbildung 3(a) wird das Feedback-Tool z.B. dazu verwendet, die Qualität der Audio-Übertragung und die Geschwindigkeit des Vortrags zu evaluieren. Um eine schnelle graphische Auswertung der Ergebnisse, wie in Abbildung 3(b) dargestellt, zu ermöglichen, empfiehlt sich die Verwendung von Multiple Choice-Fragen. Jeder Teilnehmer kann seine Antwort über den entsprechenden Schieberegler (vgl. Abbildung 3(b) unten) abgeben. Alle Antworten werden dann anonym bei jedem Teilnehmer gesammelt und sind so für jedermann sichtbar. Alternativ können aber auch Fragen gestellt werden, die ausführliche Textantworten zulassen.



(a) Dialog bei Wortmeldung

(b) Visualisierung in der Teilnehmerliste

**Abbildung 2:** Das Melde-Tool des mlb



(a) Erzeugen von Fragen

(b) Auswertung der Antworten

**Abbildung 3:** Das Feedback-Tool des mlb

Um Diskussionen zwischen den Teilnehmern einer mlb-Sitzung unabhängig von einem Audio-Kanal abzuhalten, steht ein separates Chat-Fenster zur Verfügung. Dies ist beispielsweise nützlich, um Problemstellungen zu diskutieren, ohne den Dozenten zu unterbrechen.

#### 4 Architektur des mlb

Das mlb ist eine verteilte interaktive Anwendung und besitzt eine Peer-to-Peer-Architektur. Dies bedeutet, dass jeder Teilnehmer einer mlb-Sitzung über eine eigene,



vollständige Instanz des mlb verfügt und dass jede Anwendungsinstanz lokal eine Kopie der Anwendungsdaten (Anwendungszustand) hält. Wie der Zustand aller Instanzen konsistent gehalten werden kann, wird in Abschnitt 4.1 beschrieben. In Abschnitt 4.2 wird ein Algorithmus vorgestellt, der es zu spät eingetretenen Teilnehmern erlaubt, dennoch an einer Sitzung teilzunehmen. Abschnitt 4.3 gibt einen Überblick über die im mlb zum Nachrichtenaustausch verwendeten Netzwerk-Protokolle. Diese basieren auf IP Multicast und ermöglichen so eine effiziente Gruppenkommunikation.

#### 4.1 Konsistenzerhaltung

Der Anwendungszustand des mlb besteht im wesentlichen aus den Folieninhalten (Grafiken, Texte, ...) und den Attributen der oben beschriebenen kollaborativen Tools. Verändert werden kann der Anwendungszustand des mlb ausschließlich durch Benutzeraktionen (Ereignisse), wie beispielsweise dem Zeichnen einer Linie. Alle Zustandsänderungen werden zwischen den Anwendungsinstanzen ausgetauscht, so dass die zunächst lokal durchgeführten Benutzeraktionen durch alle Instanzen nachvollzogen werden können. Ein wichtiges Ziel ist es hierbei, den Austausch und die Auswertung der Nachrichten so zu gestalten, dass die jeweiligen Kopien der Anwendungsdaten bei allen Teilnehmern identisch sind und sich ein konsistenter Gesamtzustand ergibt [3].

Ohne spezielle Maßnahmen kann die Konsistenz des verteilten Anwendungszustands nicht garantiert werden, selbst wenn bei der Übertragung der Ereignisse keine Paketverluste auftreten [4]. Ursache dafür sind die unterschiedlichen Verzögerungszeiten (Delay), die sich für die einzelnen Anwendungsinstanzen zwischen dem Auftreten und dem Ausführen eines Ereignisses ergeben. Während eine Benutzeraktion bei der verursachenden (lokalen) Anwendungsinstanz ohne Zeitverzögerung ausgeführt werden kann, existiert bei den anderen Instanzen eine gewisse Zeitspanne zwischen Erzeugung und Ausführung, die durch die Netzwerkverzögerung verursacht wird. Dies kann dazu führen, dass die Reihenfolge, in der Ereignisse bearbeitet werden, bei den einzelnen Instanzen nicht mehr übereinstimmt. Beispielsweise kann es vorkommen, dass zwei Benutzer nahezu gleichzeitig die Farbe desselben Rechtecks verändern. Der erste Teilnehmer wählt die Farbe rot, der zweite blau. Zunächst wird die Änderungsoperation jeweils lokal ausgeführt und dann an den anderen Teilnehmer übertragen, wo das Ereignis dann interpretiert wird. Am Ende sieht der erste Teilnehmer ein blaues und der zweite ein rotes Rechteck, d. h. der Zustand der Anwendung ist inkonsistent.

Eine Konsistenzerhaltung ist daher ein wichtiger Bestandteil von verteilten interaktiven Anwendungen wie dem mlb. In der Literatur unterscheidet man zwischen pessimistischen und optimistischen Ansätzen. Pessimistische Ansätze verhindern Inkonsistenzen, indem sie den gleichzeitigen Zugriff von mehreren Benutzern auf gleiche oder zusammenhängende Teile des Anwendungszustands reglementieren (Locking, Floor Control). Obwohl diese Vorgehensweise sehr effektiv ist, verhindert sie jedoch echte Kollaboration zwischen den Teilnehmern. Optimistische Ansätze erlauben dagegen kurzfristige Inkonsistenzen und reparieren diese dann im Nachhinein. Größter Nachteil von optimistischen Verfahren ist ihre Komplexität.

Für das mlb wird ein optimistisches Verfahren zur Konsistenzerhaltung verwendet, das auf dem Prinzip der Serialisierung von Zustandsänderungen beruht und eine Kombination aus



den beiden Verfahren Local Lag und Timewarp ist. Die grundsätzliche Idee der Serialisierung ist es, dass ein konsistenter Zustand dann erreicht werden kann, wenn beginnend mit dem identischen initialen Zustand alle Zustandsänderungen bei allen Anwendungsinstanzen in der gleichen Reihenfolge ausgeführt werden. Um festzustellen, wie diese Reihenfolge zu sein hat, benötigt man eine Ordnungsrelation, mit deren Hilfe alle Zustandsänderungen in eine eindeutige Reihenfolge gebracht werden können. Mögliche Ordnungen kann man auf Basis von Zustandsvektoren oder Zeitstempeln definieren. Letztere werden beim mlb verwendet. Dies setzt allerdings voraus, dass die Uhren der teilnehmenden Anwendungsinstanzen synchronisiert sind (z.B. mit NTP [5] oder GPS).

Weiter oben wurde beschrieben, wie Inkonsistenzen dadurch entstehen können, dass die Zeit zwischen der Erzeugung und der Ausführung einer Zustandsänderung bei den einzelnen Teilnehmern abweicht. Die Idee des Local Lag ist es nun, diese Zeitspanne für alle Anwendungsinstanzen anzugleichen. Dies wird dadurch erreicht, dass jede Änderung eine Ausführungszeit (Local Lag) zugewiesen bekommt, die in der Zukunft liegt (typischerweise zwischen 50 und 150 ms). Die Zeitspanne bis zum Erreichen der Ausführungszeit kann dann dazu verwendet werden, die Änderung an alle Teilnehmer zu übertragen. Im Idealfall führen nun alle Instanzen die Zustandsänderung zum gleichen Zeitpunkt aus, inklusive derjenigen, die die Änderung erzeugt hat. Dadurch wird die auf Zeitstempeln basierende Ordnung zwischen Zustandsänderungen bei allen Teilnehmern hergestellt und automatisch ein konsistenter Zustand erreicht.

Während die Anzahl möglicher Inkonsistenzen mit Hilfe von Local Lag wesentlich reduziert werden kann, kann man diese dennoch nicht ausschließen, da Nachrichten aufgrund ungünstiger Netzwerkbedingungen auch weiterhin zu spät empfangen werden können. In einem weiteren Schritt wird daher das Timewarp-Verfahren eingesetzt, das wie folgt funktioniert: Jede Anwendungsinstanz speichert lokal eine Historie aller lokalen und aller empfangenen Zustandsänderungen. Zusätzlich wird periodisch eine Kopie des jeweils aktuellen lokalen Zustands in die Historie eingefügt. Empfängt man nun eine Änderung, deren Zeitstempel in der Vergangenheit liegt, setzt man die Anwendung auf den letzten Zustand zurück, der in der Historie vor der auslösenden Änderung liegt. Danach führt man alle nachfolgenden Änderungen (inklusive der verspäteten) der Historie aus, bis der aktuelle Zeitpunkt erreicht ist. Der so gewonnene Zustand ist dann bezüglich der ausgeführten Änderungen wieder konsistent. Um den Benutzer nicht zu verwirren, wird nur dieser Endzustand angezeigt und die Neuberechnung verborgen. Der zentrale Vorteil des Timewarp-Verfahrens ist es, dass Inkonsistenzen ausschließlich mit Hilfe lokaler Informationen behoben werden können.

Sowohl Local Lag als auch Timewarp können grundsätzlich von allen verteilten interaktiven Anwendungen zur Konsistenzerhaltung verwendet werden. Eine detailliertere Beschreibung des Konsistenzbegriffs, und der Verfahren Local Lag und Timewarp findet sich in [4].

## 4.2 Unterstützung von Nachzüglern

Das mlb erlaubt es einem Benutzer, einer Sitzung zu einem beliebigen Zeitpunkt beizutreten (oder sie zu verlassen). Da ein verspätet eintretender Teilnehmer aber sämtliche



Zustandsänderungen seit dem Beginn der Sitzung verpasst hat, muß der Zustand seiner Anwendungsinstanz zunächst mit dem aktuellen Zustand initialisiert werden [6]. Die Daten, die dafür benötigt werden, werden dabei von einer anderen Anwendungsinstanz bereitgestellt. Diese Initialisierung bezeichnet man auch als Late Join. Neben dem tatsächlichen Zuspätkommen von Teilnehmern ist ein Late Join aber auch dann erforderlich, wenn sich ein Teilnehmer nach technischen Störungen (z.B. Programmabsturz) wieder in die Sitzung einwählt oder wenn, wie oben beschrieben, eine Aufgabe in separaten Untergruppen bearbeitet wird. Um eine einfachere Diskussion zu ermöglichen, bezeichnen wir den Nachzügler als Late Join-Client und diejenige(n) Anwendungsinstanz(en), die den Late Join-Client mit Informationen versorgen, als Late Join-Server.

Zunächst stellt sich die Frage, welche Teile des Anwendungszustands an den Late Join-Client übertragen werden sollen. Da dieser unter Umständen eine sehr große Anzahl an Zustandsänderungen verpasst hat, wäre eine Initialisierung mit der gesamten Historie von Änderungen sehr ineffizient, zumal sich darin auch Zustandsänderungen befinden können, die sich auf mittlerweile gelöschte Objekte beziehen. Statt dessen wird der Late Join-Client mit dem aktuellen Zustand selbst initialisiert, genauer, mit den Teilen des aktuellen Zustands, den die Anwendung benötigt, um ihn dem Benutzer darzustellen. Dies umfasst im wesentlichen den Zustand der momentan aktiven Seite im Shared Workspace sowie die Hierarchie des gemeinsamen Dokuments. Die Inhalte aller anderen Seiten brauchen nur dann angefordert werden, wenn sie von einem Teilnehmer aktiviert werden sollte. Durch diese Vorgehensweise wird der Initialisierungsprozess für den Late Join-Client merklich beschleunigt sowie der Verbrauch an Netzwerk- und Anwendungs-Ressourcen minimiert [6].

Da es sich bei mlb um eine verteilte Anwendung handelt, kann grundsätzlich jeder Teilnehmer einer Sitzung die Rolle des Late Join-Servers übernehmen und den Late Join-Client mit Teilen des Anwendungszustands beliefern. Daher bestimmt ein Auswahlverfahren einen geeigneten Late Join-Server auf Basis der Netzverzögerungszeit zwischen Client und Server.

Die Initialisierungsdaten werden über eine separate Multicast-Gruppe so versendet, dass sie nur von den Late Join-Clients empfangen werden. Andere Teilnehmer, die diese Daten nicht benötigen, werden so durch den Late Join nicht belastet. Gleichzeitig verringert dies auch den Verbrauch an Netzwerkressourcen.

Der entwickelte Late Join-Algorithmus ist ebenso wie die Mechanismen zur Konsistenzhaltung für beliebige verteilte interaktive Anwendungen einsetzbar. Eine ausführliche Beschreibung des Late Join-Algorithmus findet sich in [6]. In [4] wird außerdem betrachtet, wie sich die Konsistenz im Falle eines Late Joins sichern lässt.

### 4.3 Netzwerkprotokolle

Alle Zustandsänderungen werden in der Form von Nachrichten zwischen den mlb-Instanzen aller Teilnehmer ausgetauscht. Um diese Gruppenkommunikation effizient zu realisieren, verwendet das mlb IP Multicast und UDP als Basis für die Datenübertragung. UDP selbst ist allerdings unzuverlässig, d.h. Paketverluste werden nicht repariert, und die

Reihenfolge empfangener Nachrichten kann von der Sendereihenfolge abweichen. Zuverlässige Übertragung und (eingeschränkt) Reihenfolgeerhaltung sind für eine verteilte interaktive Anwendung wie dem mlb aber essentiell. Daher integriert das mlb das zuverlässige Multicast-Protokoll SMP (Scalable Multicast Protocol), das an der Universität Mannheim entwickelt wurde und auf den Mechanismen von SRM (Scalable Reliable Multicast) [7] beruht.

Wie bei anderen verteilten interaktiven Anwendungen (verteilte Java-Animationen, simulierte 3D-Welten, Netzwerk-Computerspiele, ...) basiert das Anwendungsmodell und damit auch die Nachrichtenübertragung des mlb auf Zuständen von Objekten und Zustandsänderungen (Ereignissen) auf diesen Objekten. Auf Basis dieser Erkenntnis wurde an der Universität Mannheim das Anwendungsprotokoll RTP/I (Real Time Application Level Protocol for Distributed Interactive Media) [8] entworfen. RTP/I erlaubt die standardisierte Übertragung beliebiger interaktiver und verteilter Anwendungen, ähnlich wie RTP [9] für kontinuierliche, nicht-interaktive Medien (Audio und Video). Wie RTP besteht RTP/I aus zwei Teilen: einem Datenprotokoll für die Übertragung der eigentlichen Anwendungsdaten und einem Kontrollprotokoll zur Verwaltung von Informationen bezüglich des Zustandsraums (d.h. der vorhandenen Objekte) und der Sitzungsteilnehmer. Das Datenprotokoll dient der Einteilung des von der Anwendung erzeugten Datenstroms in Zustände, Delta-Zustände, Ereignisse und Zustandsanfragen, die jeweils als spezielle RTP/I-ADU (Application Data Unit) realisiert sind. Neben dem dem Typ enthalten RTP/I-ADUs weitere spezifische Informationen wie z.B. der Kennung des Senders, einem Zeitstempel und einem Identifizierer des betroffenen Objekts. Beim mlb wird die Erzeugung eines graphischen Objekts z.B. als Zustand übertragen und eine Änderung dieses Objekts (z.B. Veränderung der Position, Linienfarbe,...) als Ereignis. Die Kodierung der Anwendungsdaten selbst wird nicht von RTP/I festgelegt, sondern über einen Nutzdatentyp definiert.

RTP/I als standardisiertes Protokoll für die Datenübertragung von verteilten interaktiven Anwendungen bietet zahlreiche Vorteile. Beispielsweise können Datenströme verschiedener Anwendungen untereinander synchronisiert werden, indem die in den Datenpaketen enthaltenen Zeitstempel abgeglichen werden. Ein zentraler Vorteil von RTP/I besteht also darin, daß die von RTP/I zur Verfügung gestellten Informationen von anwendungsunabhängigen (generischen) Diensten genutzt werden können, ohne daß die in den Paketen transportierten Nutzdaten interpretiert werden müssen. Beispiele für Aufgaben, die als generische Dienste realisiert sind, sind die oben beschriebenen Mechanismen zur Konsistenzhaltung, der Late Join-Algorithmus sowie ein Rekorder für die Aufzeichnung und die Wiedergabe von RTP/I-Sitzungen (vgl. Kapitel 5). Ohne die gemeinsame Basis RTP/I müssten diese Problemlösungen jeweils anwendungsspezifisch implementiert werden.

Für die Kodierung der Anwendungsdaten des mlb wird ein Protokoll verwendet, das auf RTP/I aufsetzt. Dieses Protokoll legt also fest, wie Benutzeraktionen (z.B. Erzeugen und Ändern graphischer Objekte) und sonstige Informationen kodiert und über RTP/I transportiert werden.

## 5 Erzeugung von CBT-Einheiten

Zeichnet man die in synchronen Lehrveranstaltungen verwendeten Medien (mlb-Dokumente, Audio und Video) auf, können daraus Computer Based Training-Einheiten

(CBT-Einheiten) erstellt werden, so dass die Inhalte auch in asynchronen Lehrszenerien weiterverwendet werden können. Gegenüber der expliziten Erstellung von CBT-Einheiten hat diese Vorgehensweise den Vorteil, dass der zusätzliche Aufwand sehr gering ist.

Die Aufzeichnung der Audio- und Videoströme synchroner Lehrveranstaltungen ist mittlerweile sehr gut verstanden und wird beispielsweise im Rahmen des VIROR-Projekts seit dem Sommersemester 1998 im Regelbetrieb durchgeführt. Zur Aufzeichnung wird das MVoD-System [10] verwendet, das die aus dem Hörsaal übertragenen Audio- und Videoströme empfängt und synchronisiert auf der Festplatte abspeichert.



Abbildung 4: Abruf einer CBT-Einheit

Die Aufzeichnung des Datenstroms verteilter interaktiver Anwendungen wie dem mlb ist dagegen weit problematischer. Eine der Hauptschwierigkeiten bei der Implementierung eines Rekorders für interaktive Anwendungen ist die Realisierung des wahlfreien Zugriffs auf gespeicherte Aufzeichnungen, da der Datenstrom im Gegensatz zu Audio und Video interne Abhängigkeiten besitzt. Das Decodieren einer mlb-Aufzeichnung erfordert nämlich zusätzlich das Wissen über den am Zugriffszeitpunkt gültigen Zustand. So könnte ein Dozent in einer aufgezeichneten mlb-Sitzung in der Minute 12 eine Folie laden, auf der er dann in der Minute 17 eine Annotation anbringt. Wird nun auf die Aufzeichnung dieser Sitzung an der Minute 17 zugegriffen, so würde ohne weitere Vorkehrungen nur die Annotation wiedergegeben. Die der Annotation zugrundeliegende Seite würde fehlen. Diese

Seite ist an der Minute 12 im aufgezeichneten Datenstrom enthalten, d.h. weit vor dem Zugriffspunkt.

Basierend auf dem RTP/I-Protokoll wurde ein generischer Rekorder für die Medienströme verteilter interaktiver Anwendungen realisiert [11]. Der Rekorder ist zusätzlich in der Lage, Audio und Video synchronisiert zur interaktiven Anwendung aufzuzeichnen. Um den wahlfreien Zugriff auf aufgezeichneten interaktiven Medienströmen zu ermöglichen, wurde ein Mechanismus in den Rekorder integriert, der den Anwendungszustand zu Beginn einer Wiedergabe rekonstruiert. Da dieser Mechanismus ausschließlich auf den RTP/I-Protokollelementen arbeitet und keine medienspezifischen Daten interpretiert, ist es nicht möglich, den Anfangszustand eines Mediums im Rekorder direkt zu berechnen. Statt dessen kann der Rekorder existierende RTP/I-Pakete aus einer Aufzeichnung extrahieren und senden. Die Aufgabe des Rekorders ist es daher, eine Sequenz aus aufgezeichneten Zustandsinformationen und Ereignissen so zusammenstellen und zu senden, dass alle Empfänger in den für das weitere Abspielen benötigten Zustand versetzt werden. Dieser Mechanismus arbeitet ausschließlich auf den RTP/I-Protokollprimitiven und kann somit auf alle interaktiven Medienströme angewandt werden.

Verknüpft man die Medienströme einer synchronen Teleteaching-Veranstaltungen mit den traditionell im asynchronen Szenario verwendeten Dokumenten (Webseiten, Java-Animationen, ...), entsteht so eine integrierte, multimediale CBT-Einheit, die über das WWW abrufbar ist. Abbildung 4 zeigt das Beispiel einer CBT-Einheit zur Vorlesung "Rechnernetze" an der Universität Mannheim. Um die aufgezeichnete mlb-Sitzung anzuschauen, wird wiederum das mlb selbst benötigt. Die Besonderheit liegt nun darin, dass alle Objekte (Annotationen) genauso rekonstruiert werden, wie dies der Dozent ursprünglich getan hat. Theoretisch kann der Betrachter auch mit dem so generierten Dokument interagieren.

Um eine CBT-Einheit aus dem ursprünglichen Lehrdokument (Folien), das ins HTML-Format konvertiert wurde, und den Aufzeichnungen zu generieren, müssen alle Elemente miteinander verknüpft werden. Die Zuweisung der Vorlesungsmitschnitte zu Lehrdokumenten kann dabei weitgehend automatisiert werden, wenn während der Veranstaltung die Verwendung der Lehrdokumente protokolliert wird. Die Navigation innerhalb der CBT erfolgt dann anhand der Seitenstruktur des Dokuments (vgl. Browser-Fenster links in Abbildung 4). Die Einbindung weiterer multimedialer Elemente (z.B. Java-Animationen) muß vom Autor selbst durchgeführt werden. Da die CBT-Einheit eine große Menge an Lehrmaterialien enthalten kann, ist es unabdingbar, bei der Verknüpfung der Dokumente ein didaktisch sinnvolles Instruktionsdesign anzuwenden. Eine detaillierte Beschreibung dieses Verfahrens sowie des Instruktionsdesigns findet sich in [12].

## 6 Zusammenfassung und Ausblick

Das multimedia lecture board (mlb) ist ein leistungsfähiges Shared Whiteboard, das in allen computergestützten Video-Konferenzen zum Präsentieren und gemeinsamen Bearbeiten von Dokumenten verwendet werden kann. Neben der Basisfunktionalität einer Präsentations-Software wie dem Editieren von graphischen Objekten und Präsentationsanimationen bietet das mlb zahlreiche Funktionen, die die kollaborative Gruppenarbeit un-

terstützen. Beispielsweise können die Wortmeldungen von Teilnehmern koordiniert werden.

Die Peer-to-Peer-Architektur des mlb ermöglicht mlb-Sitzungen mit einer beliebigen Anzahl von Teilnehmern ohne zusätzliche Infrastruktur. Die replizierte Architektur erfordert gleichzeitig aber spezielle Mechanismen zur Konsistenzerhaltung des Anwendungszustands und zur Unterstützung von Nachzählern. Diese Problemstellungen wurden auf Basis des RTP/I-Protokolls allgemeingültig für alle verteilten interaktiven Anwendungen gelöst. Des Weiteren erlaubt RTP/I die Aufzeichnung von synchronen Tele-Veranstaltungen. Aus diesen Aufzeichnungen lassen sich dann mit geringem Aufwand CBT-Einheiten für den Gebrauch ins asynchrone Lernszenarien generieren.

Das mlb ist in C++ und Tcl/Tk implementiert und sowohl unter Linux als auch unter Windows lauffähig. Der Programmcode des mlb wurde unter der GPL-Lizenz als Open Source freigegeben. Die lauffähigen Versionen des mlb, der Programmcode und allgemeine Informationen zum mlb können unter [1] bezogen werden.

Für die Zukunft sind zahlreiche Detailverbesserungen geplant, die die Handhabung des mlb erleichtern. Schwerpunkte zukünftiger Entwicklungen liegen außerdem in einer leichtgewichtigen Version für mobile Endgeräte (ein erster Prototyp existiert bereits) und weiterer Mechanismen für die kollaborative Gruppenarbeit.

## Literatur

- [1] J. Vogel, "multimedia lecture board (mlb)", URL: <http://www.informatik.uni-mannheim.de/informatik/pi4/projects/mlb/>, 2003.
- [2] ANETTE, "Projekt Applications and Network Technology for Teleteaching", DFN-Projekt TK602-VA/T 102.1, URL: <http://www.informatik.uni-mannheim.de/informatik/pi4/projects/ANETTE/anetteIndex.html>, 2000.
- [3] C. Ellis and S. Gibbs, "Concurrency Control in Groupware Systems", in *Proc. of ACM SIGMOD*, 1989, pp. 399–407.
- [4] J. Vogel and M. Mauve, "Consistency Control for Distributed Interactive Media", in *Proc. of ACM Multimedia 2001, Ottawa, Canada*, 2001, pp. 221–230.
- [5] D. Mills, "Network Time Protocol (Version 3) specification, implementation and analysis", DARPA Network Working Group Report RFC-1305, University of Delaware, 1992.
- [6] J. Vogel, M. Mauve, W. Geyer, V. Hilt, and C. Kuhmuench, "A Generic Late Join Service for Distributed Interactive Media", in *Proc. of ACM Multimedia 2000, Los Angeles, USA*, 2000, pp. 259–268.
- [7] S. Floyd, V. Jacobson, C. Liu, S. McCanne, and L. Zhang, "A Reliable Multicast Framework for Light-Weight Sessions and Application Level Framing", *IEEE/ACM Transactions on Networking*, vol. 5, no. 6, pp. 784 – 803, 1997.
- [8] M. Mauve, V. Hilt, C. Kuhmuench, and W. Effelsberg, "RTP/I - Toward a Common Application-Level Protocol for Distributed Interactive Media", *IEEE Transactions on Multimedia*, vol. 3, no. 1, pp. 152–161, 2001.
- [9] S. Floyd, V. Jacobson, S. McCanne, C. Liu, and L. Zhang, "A Reliable Multicast Framework for Light-weight Sessions and Application Level Framing", in *Proceedings of ACM SIGCOMM'95*, 1995, Scalable Reliable Multicast, Whiteboard, Application Level Framing, Untersuchung verschiedener Topologien, zukunfftige Aenderungen.



- [10] W. Holfelder, "Interactive Remote Recording and Playback of Multicast Videoconferences", in *Proc. 4<sup>th</sup> International Workshop on Interactive Distributed Multimedia Systems and Telecommunication Services (IDMS '97)*. IDMS, September 1997.
- [11] V. Hilt, M. Mauve, C. Kuhmuench, and W. Effelsbeg, "A Generic Scheme for the Recording of Interactive Media Streams", in *Proc. of IDMS, Toulouse, France, 1999*, pp. 291–304.
- [12] C. Schremmer and V. Hilt, "A Systematic Approach to the Automatic Conversion of a LiveLecture into a Multimedia CBT Course", in *Proc. of NLT, Bern, Switzerland, 1999*.

