Studiengang:	Informatik
Prüfer:	Prof. Dr. Kurt Rothermel
Betreuer:	Dipl. Inf. Arno Wacker
begonnen am:	01.11.2002
beendet am:	30.04.2003
CR-Klassifikation:	C.2.2, C.2.4, H.5.3, K.3.1, K.6.3

Studienarbeit Nr. 1872

Fernsteuerung von Anwendungen

Holger Cermann

Institut für Informatik
Universität Stuttgart
Breitwiesenstraße 20–22
D–70565 Stuttgart

Zusammenfassung

Vorgestellt wird die Konzeption und Entwicklung einer im Rahmen des NUSS-Projektes der Universität Stuttgart erstellten Software zur Fernsteuerung von beliebigen Anwendungen, beispielhaft an Microsoft PowerPoint.

Um dies zu erreichen, werden grundlegende Techniken der Kommunikation sowie Arten der Fernsteuerung beschrieben. Daraufhin werden die verschiedenen Möglichkeiten für einen Einsatz mit PowerPoint diskutiert und für das resultierende Fernsteuerungsmodell eine entsprechende Software entworfen und implementiert.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Kapitel 1

Einleitung

Nach einer Motivation für diese Arbeit folgt in diesem Kapitel die daraus entstandene Aufgabenstellung. Im letzten Abschnitt dieses Kapitels werden die Anforderungen einerseits für eine generische Fernsteuerung, andererseits für eine PowerPoint-Fernsteuerung gesammelt.

1.1 Motivation

Multimedia und Computerunterstützung hält immer mehr Einzug in Lehrveranstaltungen. So werden Folien immer häufiger elektronisch mittels Beamer und Notebook präsentiert. Ebenso werden auf Gruppentreffen - z.B. von Lern- oder Übungsgruppen - mehr und mehr elektronisch Notizen erstellt. Solche Softwaresysteme sind jedoch (vor allem in Lehrveranstaltungen) meist nicht miteinander verbunden, so dass bei Zwischenfragen oder Anmerkungen oft verbal und mittels Flipchart bzw. Tafel gearbeitet werden muss. Daher wäre eine lokale wie auch entfernte Unterstützung einer gemeinsamen Nutzung von Anwendungen (und im speziellen von Präsentationssoftware) wünschenswert.

Für die gemeinsame Nutzung einer Anwendung gibt es unter anderem folgende Gründe:

- Befinden sich die Teilnehmer nicht am gleichen Ort, so ist grundsätzlich eine technische Unterstützung nötig.
- Wird eine Anwendung von mehreren Personen gemeinsam genutzt und besitzt diese keine Fernsteuerungsunterstützung, so müßten sich alle vor dem ausführenden Rechner befinden. Dies ist bei einer großen Personenzahl nicht mehr möglich.
- Sollen viele Personen gleichzeitig an einer Präsentation teilnehmen, so können diese (als Alternative zu einem Beamer) auf einzelnen Computern die Anwendung verfolgen.

1.2 Aufgabenstellung

Da sich die Anforderungen an diese Aufgabe während ihrer Bearbeitung geändert haben, sei hier zunächst die ursprüngliche Aufgabenstellung, wie sie in der Ausschreibung der Studienarbeit zu finden ist, angeführt. Danach werden Gründe sowie die daraus resultierenden Änderungen aufgezählt.

1.2.1 Ursprüngliche Aufgabenstellung

Die ursprüngliche Aufgabe dieser Studienarbeit war die Erweiterung der SASCIA Architektur [SASCIA] um eine Fernsteuerungsfunktionalität:

Am Markt verfügbare Produkte wie VNC und Netmeeting bieten bereits eine generische Anwendungsfernsteuerung. Hier erhält eine Person ein Schreibrecht zur Anwendung, die übrigen Personen können die Anwendung lesend verfolgen. In der Gruppenarbeit ist aber auch die konkurrierende Nutzung einer Anwendung mit rollenbezogenen, abgestuften Berechtigungen wünschenswert.

In der angebotenen Studienarbeit sollen die vorhandene SASCIA Architektur und das in SASCIA bereitgestellte Rahmenwerk derart überarbeitet werden, dass beliebige Anwendungen, die nach dem Model-View-Control Paradigma aufgespalten werden können,

an SASCIA angebunden werden können, wobei auf der existierenden Komponente einer generischen Floor Control aufzusetzen ist. Die entworfene Architektur soll am Beispiel einer verbreiteten Standardanwendung, nach Möglichkeit PowerPoint, prototypisch implementiert und validiert werden. Dabei ist zu untersuchen,ob PowerPoint gemäß des MVC Paradigmas aufgebrochen werden kann.

1.2.2 Geänderte Aufgabenstellung

Durch die Entstehung des NUSS-Projektes (Notebook University Stuttgart, [NUSS]) wurde das SASCIA-Projekt abgelöst. Es beschäftigt sich interdisziplinär mit der Untersuchung einer verstärkten Beteiligung von Studenten in der Lehre mittels mobiler Geräte, Funktechnologien und gemeinsam benutzten Anwendungen.

Dadurch entstand ein geändertes Systemumfeld. Während ursprünglich das Rahmenwerk von SASCIA verwendet werden sollte, wird nun das parallel entwickelte NUSS-Framework [BCNW02] unterstützt. Da dieses im Gegensatz zu seinem Vorgänger nicht über einen Dienst zum Datentransport verfügt, werden hier auch diese Aspekte diskutiert.

Die zweite Änderung betrifft die (ursprünglich geforderte) reine Aufsplittung nach dem MVC-Paradigma (siehe Kapitel 2.1.1). Da sich während früher Phasen bereits herausgestellt hat, dass sich kommerziell verfügbare Software in den meisten Fällen nicht nach diesem Paradigma zerlegen lässt und demnach auch nicht danach fernsteuerbar ist, wurde diese Einschränkung aufgelöst.

Aus diesem Grund werden die unterschiedlichen Möglichkeiten der Fernsteuerung diskutiert und exemplarisch an Microsoft PowerPoint mit der hierfür am besten erscheinende Lösung realisiert.

1.3 Anforderungen

Anforderungen an ein System zur Fernsteuerung anderer Anwendungen lassen sich grundsätzlich in zwei Bereiche unterteilen:

1. Anforderungen, welche an eine generische Fernsteuerung gestellt werden (also ohne jegliche Kenntnis über deren Programmlogik) sowie
2. Anforderungen an eine einzelne zu steuernde Anwendung.

Da sich im generischen Fall zum Beispiel keine feingranularen Rechte (ohne Wissen über Semantik von Benutzeraktionen) realisieren lassen, ist dieser zweite Punkt für eine wie in der Ausschreibung geforderten Funktionalität unabdingbar. In diesem Dokument soll diese Anforderungsanalyse, exemplarisch an einer gemeinsamen Benutzung von PowerPoint-Präsentationen, erfolgen.

1.3.1 Generische Fernsteuerung

Grundsätzlich werden folgende Anforderungen an eine Fernsteuerung gestellt:

Echtzeit Aktionen, welche an einem Computer durchgeführt werden, sollen innerhalb einer gewissen Grenze - z.B. einer Sekunde bei Benutzereingaben - auch auf allen anderen Rechnern erscheinen. Geschieht dies nicht oder zu langsam, so besteht die Gefahr, dass Gruppenteilnehmer nicht wissen, worüber der agierende Benutzer gerade spricht oder sie versuchen selber Aktionen durchzuführen.

Konsistenz/Konvergenz Durch das gleichzeitige Ausführen von Aktionen sollen weder Konsistenz noch Konvergenz der Anwendung zerstört werden. Unter Konsistenz ist hier zu verstehen, dass niemals Zustände in der Anwendung vorkommen, welche (aufgrund der Programmlogik oder intuitiver Ansicht) nicht erlaubt sind.

Die Konvergenz fordert dagegen, dass am Ende einer Aktion (z.B. einer Texteingabe) alle ferngesteuerten Anwendungen den gleichen Zustand besitzen, also im Falle von PowerPoint alle Folien gleich aussehen.

Hierzu sind Probleme wie die Reihenfolgeerhaltung zu betrachten. Aktionen, welche zum Beispiel kausal (siehe hierzu Kapitel 2.3.1) zeitlich vor einer anderen geschahen, dürfen auf keinem Rechner in der verkehrten Reihenfolge aufgeführt werden.

Skalierbarkeit/Ressourcenschonung Ein System zur Fernsteuerung besteht oftmals nur aus zwei Rechnern. Im NUSS-Projekt werden jedoch gerade Szenarien mit vielen Rechnern gleichzeitig verwendet. Zum Beispiel bei einer Vorlesung, in der der Vortragende gemeinsam mit 100 Studenten eine Anwendung bedient. Demnach muss ein solches System skalierbar sein. Dies bedeutet, dass die Komplexität im Idealfall kleiner als O(n) ist; das System bei einer Verdoppelung der Rechnerzahl also höchstens doppelt so viele Ressourcen (z.B. Bandbreite) verwendet. Ist dies nicht der Fall, so ist darauf zu achten, dass das System so ausgelegt wird, so dass die innerhalb der angedachten Szenarien maximal benötigten Ressourcen ausreichend vorhanden sind.

Sicherheit/Rechte Letztendlich sollte immer eine Einschränkung der Rechte für einzelne Benutzer möglich sein. Während dies bei einer vollkommen generischen Anwendung nur in sehr eingeschränktem Maße geschehen kann, soll zumindest der jeweils höchste Grad angestrebt werden. Grundsätzlich ist zumindest immer eine Floor Control realisierbar: Nur ein Benutzer darf (beliebig) agieren, alle anderen nur betrachten. Dieser Benutzer kann dieses Recht dann je nach Realisierung zurück- oder weitergeben.

1.3.2 Präsentationsfernsteuerung

Im Folgenden werden nun die Anforderungen einer spezifischen Anwendung an eine Fernsteuerung betrachtet. Hierbei wird von einer PowerPoint-Fernsteuerung ausgegangen.

Für eine einfache Fernsteuerung einer PowerPoint-Präsentation ist die Eigenschaft des "Gemeinsamen Blätterns" (Wenn eine Person blättert, so wird dies auf allen zugehörigen Rechnern durchgeführt) ausreichend. Das in dieser Arbeit entwickelte System soll jedoch weitere Anforderungen unterstützen, um den in der Motivation beschriebenen interaktiven Einsatz zu ermöglichen.

Zur PowerPoint-Fernsteuerung gehört hier demnach mehr als das reine Blättern. Auch Funktionen für gemeinsame Annotationen (wie das Markieren mittels Text und Stift) mehrerer Benutzer gleichzeitig sollen möglich sein. Es handelt sich demnach nicht um ein ferngesteuertes PowerPoint, sondern um ein **Shared**PowerPoint.

Geforderte Eigenschaften der späteren Anwendungen:

- Automatische Verteilung beliebiger PowerPoint-Präsentationen
- Verteilte Präsentationen
  Es wird eine Fernsteuerung von Folienabläufen (Globales Blättern) inklusive aller Effekte (wie z.B. Animationen) ermöglicht. Hierbei kann die Präsentation von zentraler Stelle, aber auch von beliebigen Clients gesteuert werden. Außerdem ist ein von der aktuellen Präsentationsfolie losgelöstes Blättern innerhalb der Präsentation möglich. Die Navigation erfolgt mittels der in PowerPoint gewohnten Möglichkeiten (Bild Auf/Ab, Leertaste/Backspace, durch numerische Folieneingabe mit anschließendem Return sowie bei Wunsch mittels linker Maustaste).

- Annotationen
  Innerhalb der Präsentation werden folgende Annotier-Möglichkeiten zur Verfügung gestellt:

  - Temporärer Stift: Der aus PowerPoint bekannte Zeichenstift, welcher sich beim Wechseln der Folie automatisch löscht.
  - Persistenter Stift: Mittels dieses Werkzeuges lassen sich Zeichnungen in Folien eintragen, welche fest (als Objekte innerhalb der aktuellen Folie) in Form

von einzelnen Strichen in die Präsentation eingebettet werden und dadurch präsentationsübergreifend erhalten bleiben.

- Persistenter Text: Wie beim persistenten Stift werden Textobjekte in die Präsentation eingefügt.
- Zeiger: Es wird ein Symbol zur Kennzeichnung eine Position auf der aktuellen Folie angezeigt.

Bei den Annotationen lässt sich jeweils die Farbe sowie (abgesehen vom Temporären Stift) auch die jeweilige Größe (Schriftgrad, Stiftdicke) wählen.

- Shared Whiteboard
  Neben der Möglichkeit, Annotationen lokal zu erstellen, können diese gemeinsam und global für alle sichtbar erstellt werden.

- Untergruppen
  Um Gruppenarbeit zu unterstützen, ist es möglich, Untergruppen zu bilden. Annotationen innerhalb einer Untergruppe werden nur zu deren Teilnehmern verteilt.

- Modifikation von Folien
  Grundsätzlich werden folgende Funktionalitäten zur Verfügung gestellt:

  - Löschen des Temporären Stiftes
  - Löschen aller Annotationen auf einer Folie
  - Durch Picking von Objekten existieren weitere objektbasierende Aktionen:
    * Löschen von einzelnen Annotationen (also das Entfernen von Texteinträgen sowie von Teilstrecken einer Zeichnung)
    * Verschieben von Annotationen (durch Setzen der neuen Position mittels Mausklick)
    * Verschieben von vorhandenen Folienobjekten
      Dadurch lassen sich Objekte (Texte, Zeichnungen, Objektgruppen), wel-

che zur ursprünglichen Präsentation gehören, auf der zugehörigen Folie neu positionieren.

Zusätzlich lässt sich mittels Picking der Urheber einer einzelnen Annotation ermitteln.

- Kommentare
  Es besteht die Möglichkeit, zu den einzelnen Folien Kommentare in Freitextform abzuspeichern. Diese sind mit den aus PowerPoint bekannten Notizen auf den Handzetteln voll kompatibel. Dort lassen sich diese unterhalb der Folie eintragen.

- Lokale Präsentationskopien
  Nach einer Präsentation erhält der Benutzer die Möglichkeit, diese abzuspeichern. Dadurch werden im Gegensatz zu herkömmlichen Präsentationen zwei Dateien erstellt: Einmal die Präsentation, wie sie am Anfang der Sitzung erhalten wurde und zusätzlich die Präsentation, wie sie am Ende aussah (also inklusive allen Modifikationen und Annotationen). Diese Dateien lassen sich danach wie gewohnt in PowerPoint öffnen und weiter editieren.

- NUSS-Anbindung
  Das Programm baut auf das NUSS-Framework der Universität Stuttgart auf und verwendet damit dessen Features wie Session Management sowie dessen graphischen Konfigurationstools. Das NUSS-Framework wird in Kapitel 3.3 genauer beschrieben.

- NUSS-Rechteverwaltung
  Aufgrund der NUSS-Anbindung wird dessen Rechteverwaltung (mit Rollendefinitionen, Policies, Kompatibilitäten, ...) zur Steuerung der Annotations- und Präsentationsfunktionen unterstützt.

- DONUT-Unterstützung
  Das DONUT-System ist ein von Michael Nagy im Rahmen einer Diplomarbeit

[NAGY03] entwickeltes System zur kontextsensitiven verteilten Annotation in Lehrumgebungen.
Zusätzlich zu den eigenen Annotationsmöglichkeiten ist eine Schnittstelle zu dem DONUT-System vorhanden. Mittels dieser lassen sich zusätzliche kontextabhängige Feedback-Informationen an ein Interaction Tracking System weiterleiten. Das Programm fungiert hierbei als Client für DONUT sowie als Kontextprovider (siehe dafür [NAGY03]).

- Präsentationsverteilung durch Multicast auf aggregierten Clients
  Die Verteilung von Präsentationen kann per Multicast mittels des "Light-wight Reliable Multicast Protocol" [LRMP] erfolgen. Sollte dies aufgrund einer fehlenden Multicast-Unterstützung des Netzwerkes nicht möglich sein, so wird auf eine Verteilung per Unicast zurückgegriffen. Dies erfolgt für den Benutzer ohne einem Zwang zur zusätzlichen Interaktion.

- Einzelstehende Ausführung
  Neben der in NUSS eingebundenen Funktionalität ist es dem Programm möglich, ohne weitere Systemvoraussetzungen in einem Broadcast-Modus zu starten. Hierbei gibt es dann jedoch weder eine Rechteverwaltung noch eine Präsentationsverteilung.

Kapitel 2

Grundlagen

In diesem Kapitel werden zunächst die grundsätzlichen Möglichkeiten der Fernsteuerung von vorhandenen Anwendungen aufgeführt. Danach werden im Related Work einige existierende Anwendungen mit ihren Fähigkeiten und Problemen besprochen.

Im zweiten Teil dieses Kapitels werden Grundlagen der Netz- und Anwendungskommunikation erklärt, welche im späteren Verlauf dieses Dokumentes verwendet oder diskutiert werden.

2.1 Methoden der Fernsteuerung

Im Folgenden werden 4 Methoden vorgestellt, mittels derer eine Fernsteuerung prinzipiell funktionieren kann. Hierbei wird zur Vereinfachung von einem Szenario mit zwei Rechnern ausgegangen. Läuft die zu steuernde Anwendung nur auf einem Rechner, so wird dieser hier als "Server", der Andere als "Client" bezeichnet.

2.1.1 MVC-Prinzip

Das Model-View-Controller-Paradigma [KrPo88] wurde ursprünglich mit der objektorientierten Sprache Smalltalk-80 eingeführt. Es teilt eine Software mit Benutzeroberfläche in 3 Bereiche auf, wie die folgende Abbildung (nach [KrPo88]) verdeutlicht:

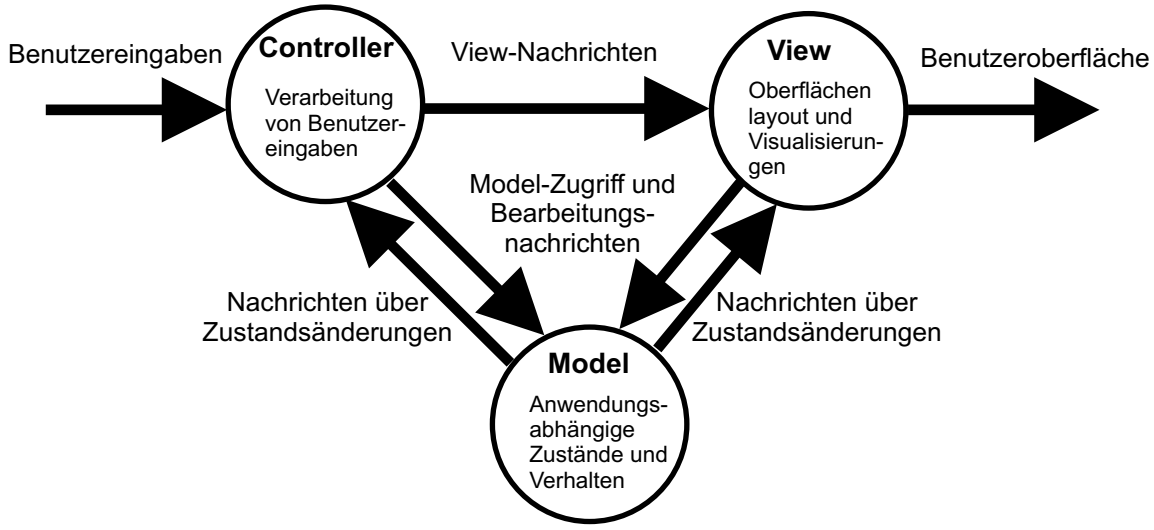

Abbildung 2.1: MVC-Prinzip

Die einzelnen Bereiche haben folgende Aufgaben:

Model Das Model liefert die Informationen zu dem, was dargestellt werden soll. Es kann die Anwendungssemantik beinhalten oder einfach nur die zugehörigen Zustände. So ist bereits eine einfache Zahl in Form eines Integers ein mögliches Model für eine Stoppuhr.

View Die View ist für alles graphische zuständig. Sie kann über Super- und Subviews verfügen, mittels derer z.B. verschiedene Ansichten auf das Model möglich sind.

Controller Der Controller bekommt Eingaben von Peripheriegeräten wie Maus und Tastatur. Seine Aufgabe ist die Verarbeitung dieser Daten und das entsprechende

Ändern von View bzw. Model.

Durch eine solche Aufteilung lässt sich eine Anwendung einfach fernsteuern, indem das lokale Model mit entfernten Views und Controllern verbunden wird. Eine gemeinsame Nutzung eines Models von mehreren Views und Controllern lässt sich ebenso realisieren, da abgesehen vom Model an keiner Stelle zustandsspezifische Daten geändert werden können.

2.1.2 Schnappschuss des Fensterinhaltes

Eine andere Möglichkeit, beliebige Anwendungen fernzusteuern, ist das Übertragen des Fensterinhaltes in Form eines Bildes. Die Bildinformationen werden entweder periodisch oder nur bei einer Änderung des Inhaltes übertragen. Die letztere Alternative müssen das Betriebssystem (zum Beispiel durch das Senden von WM_PAINT-Nachrichten in Microsoft Windows) sowie das Fernsteuerungssystem (zum Beispiel durch Abfangen eben dieser Nachrichten) unterstützen. Durch diese Erweiterung wird nicht nur weniger Netzwerklast erzeugt, es kann auch schneller auf Ereignisse reagiert werden. Um Netzwerkressourcen weitergehend zu schonen, stehen zudem folgende Möglichkeiten zur Verfügung:

- Verringern der Updatehäufigkeit
  Darunter wird bei periodisch übertragenen Bildern eine Erhöhung des Intervalls verstanden.
  Wird nur bei Änderungen übertragen, so lässt sich in gewissem Masse mit diesem Verfahren zusätzliche Netzlast verringern, in dem abgewartet wird, ob eine Änderung nur kurzzeitig existiert und somit ignoriert werden kann. Diese Optimierung ist mit dem damit verbundenen Informationsverlust abzuwägen.

- Abstraktion von Graphikprimitive
  Der Unterschied zum allgemeinen Ansatz besteht hier darin, dass nicht die letztendlich erzeugten Bilder übermittelt werden, sondern die für deren Generierung

im Betriebssystem verwendeten Graphikprimitive. Dieses Verfahren benötigt jedoch eine weitere Unterstützung durch das Betriebssystem, welches die Fernsteuerungssoftware über Zeichen- oder andere Oberflächenevents informieren muss. Ein Beispiel für diese Methode ist das X-System [X].

- Übertragung von Differenzen
  Geht man davon aus, dass sich in einer typischen Anwendung meist nur Teile des Fensterinhaltes ändern, so entfällt bei einer reinen Übertragung von Bilddifferenzen beziehungsweise von geänderten Regionen ein Grossteil der sonst gesendeten Daten. Das VNC-System (siehe Kapitel 2.2.1) verwendet unter anderem dieses sowie die folgenden Erweiterungen.

- Komprimierung der Informationen
  Die Datenmenge der resultierenden Graphikinformationen lässt sich weiterhin durch eine Komprimierung verringern. So lassen sich beispielsweise bei der Codierung von durchschnittlichen Fensterbildern (32-bit) in das GIF-Format Faktoren von 1:50 erreichen.

- Auslassen bestimmter Informationen
  Als letzte Möglichkeit zur Optimierung des Ressourcenverbrauchs sei hier das gezielte Entfernen von (für nicht notwendig erachteten) Bildinformationen angeführt. Dies kann z.B durch das Verringern der Farbtiefe von 32 Bit auf 8 Bit oder das Nichtübertragen des Desktophintergrundes realisiert werden. Das VNC-System bietet diese Möglichkeiten an.

Die so übertragenen Informationen werden auf einem Clientrechner wieder zusammengesetzt und in einem Anzeigefenster dargestellt. Die Rückrichtung besteht aus Kontrollinformationen wie Maus- und Tastatureingaben. Diese wird durch ein Weiterleiten der Events (Mauskoordinaten, Tastendrücke) an die zu steuernde Anwendung realisiert. Das Schnappschussverfahren hat den Vorteil, dass die steuernde Anwendung keinerlei Wissen über die Semantik der fernzusteuernden Anwendung haben muss und auf der Clientseite keinerlei weitere Software (ausser der Fernsteuerungssoftware) vorhanden sein muss.

2.1.3 Betriebsystemoperationen

Setzt man voraus, dass:

- die zu steuernde Software auf beiden Rechnern installiert ist,
- von der Anwendung benötigte Ressourcen jeweils gleich anzusprechen sind,
- sich die Anwendung deterministisch verhält,
- sich zu Beginn alle Programminstanzen im gleichen Zustand befinden,

so lässt sich durch das Starten der Anwendung auf beiden Rechnern und eine gleiche Versorgung mit Nachrichten ein identischer Ablauf auf beiden Rechnern realisieren.

2.1.4 Anwendungsoperationen

Die höchste Art der Abstraktion ist das Übermitteln von Anwendungsoperationen. Solche Operationen können z.B. in einer Präsentationssoftware die Bedeutung "Blättere auf der 5. Folie auf die Animation Nummer 3" enthalten. Im Gegensatz zu den bisher genannten Möglichkeiten muss die zu steuernde Anwendung hierbei jedoch explizit eine Möglichkeit zu dieser Fernsteuerungsart bieten.

2.1.5 Zusammenfassung

Die folgende Tabelle fasst die Eigenschaften der erwähnten Möglichkeiten nochmals, angelehnt an [BURG97], zusammen.

Verfahren	Netzbelastung	Ressourcen-verfügbarkeit	Anforderungen an die Anwendung
MVC	Gering	nicht erforderlich	nach MVC-Paradigma erstellt
Schnappschuss	hoch/einstellbar	nicht erforderlich	keinerlei
Betriebsystem-operationen	gering	erforderlich	benötigte Kriterien
Anwendungs-operationen	gering	erforderlich	Fernsteuerungs-unterstützung

Tabelle 2.1: Eigenschaften von Fernsteuerungsmethoden

2.2 Related Work

In diesem Kapitel werden einige auf dem Markt existierende Lösungen zur Fernsteuerung von Anwendungen und zur verteilten Präsentation vorgestellt. Hierbei werden sie den oben angeführten Methoden zugeordnet und ihre Vor- bzw. Nachteile erläutert. Als erstes werden zwei Systeme zur generischen Fernsteuerung vorgestellt: VNC und Netmeeting. Danach folgen zwei weitere Systeme zur verteilten Präsentation: ConferenceXP und SASCIA.

2.2.1 VNC

VNC (Virtual Network Computing) ist ein Client-Server-basierendes System, welches Clients die Sicht und optional auch die Kontrolle über einen anderen Rechner gibt.

Die Software arbeitet nach der Methode das Schnappschusses. Als Optimierungen werden die Bilder in einstellbaren Stärken komprimiert und es können Informationen wie Hintergrundbilder und hohe Farbtiefen reduziert werden.

Dadurch, dass die einzigen Rechte für einen Client die Freischaltung von Maus und

Tastatur sind, hat ein Client - auch wenn er nur einen Kommentar in eine Anwendung eingeben soll - die volle Kontrolle über den Server. Aus diesem Grund ist ein Einsatz dieser Software innerhalb einer Vorlesung nicht sinnvoll.

Nach dem gleichen Prinzip arbeiten auch andere Programme wie zum Beispiel PC-Anywhere oder DesktopDelivery.

2.2.2 Netmeeting

NetMeeting ist ein System zur Unterstützung verteilter Konferenzen. Es verfügt neben der Anwendungsfernsteuerung noch über eine Chat, Audio- und Videoübertragung, sowie über ein Whiteboard. Da diese weiteren Bestandteile nicht im Fokus dieser Arbeit liegen, werden sie nicht erläutert.

Die Technik der Fernsteuerung erfolgt nach dem gleichen Prinzip wie in VNC: durch Screenshots.

In der Rechteverwaltung ist Netmeeting jedoch feingranularer. Neben dem grundsätzlichen Aktivieren von Maus und Tastatur lässt sich hier noch eine Liste von Anwendungen definieren, welche für den Client sichtbar und kontrollierbar sind.

Diese Software ist zwar im Gegensatz zu VNC eher für verteilte Vorlesungen geeignet, da ein Benutzer nicht mit 3 Mausklicks den Server herunterfahren kann. Innerhalb der einzelnen Anwendung gibt es jedoch keinerlei Rechteabstufungen. Dadurch haben aus Sicht des Rechtemanagements der Fernsteuerung das Übertragen einer Textanmerkung, das Löschen einer Folie und das Ausführen eines VBA-Skriptes zum Löschen eines Verzeichnisses die gleiche Bedeutung.

2.2.3 ConferenceXP Presenter

Der ConferenceXP Presenter ist ein Shared Whiteboard, aufbauend auf dem Framework von Microsofts ConferenceXP. Das Programm ist in 3 Unterprogramme gegliedert:

DeckBuilder Der DeckBuilder ist nötig, um z.B. PowerPoint-Folien in das proprietäres Format des Presenters umzuwandeln. Dabei kann es multimediale Inhalte oder Animationen nicht übernehmen. Des weiteren lassen sich vorher feste Feedbackmöglichkeiten angeben: Der Teilnehmer kann später unter anderem entweder Kommentare schreiben, Geschwindigkeitsinformationen angeben oder mit Ja/Nein antworten.

Presenter Der Presenter wird vom Vortragenden ausgeführt und bietet die Kontrolle über den Folienablauf und Annotationen. Des weiteren lassen sich mittels Werkzeugen (wie Stiften) innerhalb der Präsentation Bemerkungen erstellen.

Viewer Der Viewer wird von den Teilnehmern benutzt. Diese melden sich an dem jeweiligen Presenter an, wofür eine Multicastfähigkeit vorrausgesetzt wird. Nach der Anmeldung haben die Teilnehmer, abgesehen von der im DeckBuilder eingestellten Annotationsmöglichkeit, keine weiteren zur Interaktion.

Es ist festzustellen, das dieses System zwar zur entfernten Präsentation geeignet ist, jedoch keinerlei Möglichkeit zum gemeinsamen Arbeiten durch kooperatives Ändern von Folien bietet. Des weiteren ist das vorherige Umwandeln in ein Zwischenformat unnötig kompliziert und das Datenformat (Die Folien werden als Bilder übertragen) ungünstig für Performance und Qualität. Insgesamt macht der Viewer aufgrund schlechter Bildqualität, Grafikfehlern und fehlender Automatisierung beim Foliendownload ein negatives Bild.

2.2.4 SASCIA WhiteBoard

Bei "**S**ystem **A**rchitecture **S**upporting **C**ooperative and **i**nteractive **A**pplications" handelt es sich um ein Framework zur Unterstützung kooperativer Arbeit [OBER01]. Für dieses System wurde innerhalb einer Diplomarbeit eine Architektur zur gemeinsamen Anwendungsnutzung vorgestellt, welches prototypisch ein Whiteboard zur Verfügung stellt (siehe [OBER02]).

Innerhalb dieser Software werden bei dem Verbinden mit dem Server (bzw. beim weiteren Hinzufügen von Folien) Bilder aller vorhandenen Folien übertragen. Eine "Folie" ist dabei als Bild realisiert, was die Übertragung als langwierig (teilweise mehrere Minuten) gestaltet. Das Einbinden von PowerPoint-Folien ist dabei nur über Screenshots oder durch einen vorherigen Export der Folien in Bilddateien möglich. Animationen sind nicht vorgesehen.

Des weiteren können Folien nicht abgespeichert, sondern nur mittels einer proprietären Log-Datei erneut abgespielt werden.

Als Annotationsmöglichkeiten stehen verschiedene Stift- und Textwerkzeuge zur Verfügung. Annotationen werden als Zeichenbefehle übertragen und bei allen Clients durchgeführt. Dadurch ist keine Objektstruktur vorhanden - welche im Nachhinein modifiziert werden könnte - sondern nur ein Bild.

Letztendlich ist festzustellen, das dieses System über alle nötigen Funktionen für eine verteilte Präsentation verfügt, jedoch aufgrund der nicht vorhandenen Unterstützung von PowerPoint, fehlenden Exportfunktionen sowie schlechter Performance bei der Präsentationsübertragung und mangelnder Stabilität unattraktiv erscheint.

2.3 Netzkommunikation

Im Folgenden werden Grundlagen der Kommunikationen erläutert, welche während der Entwicklung des Systemdesigns benötigt werden.

2.3.1 Semantiken der Kommunikation

Werden Pakete über ein Netzwerk an mehrere Rechner verschickt, so kann dies unterschiedliche Semantiken haben.

Einen Überblick bietet die Abbildung 2.2 nach [ROTH01]:

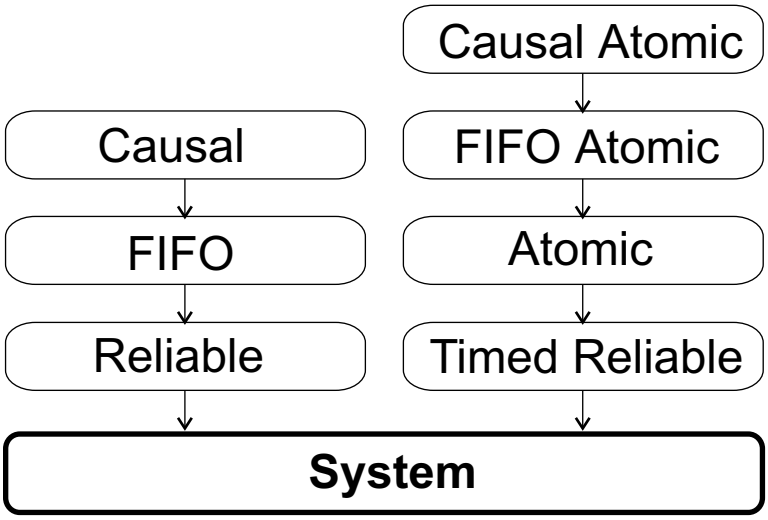

Abbildung 2.2: Semantiken der Kommunikation

Pfeile bedeuten hierbei, das eine Semantik auf eine andere aufbaut, also alle Anforderungen übernimmt. Die Begriffe Reliable und FIFO werden hierbei auch in der Kommunikation zwischen zwei Computern verwendet. Die restlichen Semantiken sind nur bei Multi- und Broadcasts (siehe Kapitel 2.3.3) von Bedeutung.

Die einzelnen Semantiken haben die im folgenden beschriebene Bedeutung. Eine weiterführende Erklärung sowie Beispiele sind in [ROTH01] zu finden.

Reliable Ist eine Kommunikation reliable, so wird sichergestellt, das eine abgeschickte Nachricht an allen korrekten Empfängern auch ausgeliefert wird. Durch den Zusatz "korrekt" wird sichergestellt, dass fehlerhafte Prozesse nicht betrachtet werden. Ein "Empfänger" sei hier dadurch definiert, dass er die Nachricht erhalten soll (und nicht zwangsweise bereits empfängt). Es müssen folgende Eigenschaften erfüllt sein:

- Validität

  Wenn ein korrekter Prozess eine Nachricht aussendet, dann liefern letztendlich alle korrekten Empfängerprozesse diese Nachricht auch aus.

- Übereinkunft
  Wenn ein korrekter Prozess eine Nachricht ausliefert, so liefern diese alle Empfängerprozesse aus.

- Integrität
  Jede Nachricht wird von jedem korrekten Prozess nur einmal ausgeliefert und das nur, wenn sie vorher an ihn gesendet wurde.

FIFO Die FIFO-Semantik ("First in, first out") sagt aus, dass sich Nachrichten nicht "überholen". Es gilt zusätzlich zu den Reliable-Eigenschaften noch folgende Bedingung:

Wird eine Nachricht m_1 vor einer Nachricht m_2 gesendet, so empfängt kein korrekter Prozess m_2 vor m_1.

Causal Hängt der Kontext einer Nachricht nicht nur von den vorher vom gleichen Prozess gesendeten Nachrichten, sondern auch von den Nachrichten anderer Prozesse ab, so muss eine kausale Ordnung verwendet werden. Die Definition wird dementsprechend erweitert:

Wurde eine Nachricht m_1 kausal vor einer Nachricht m_2 verschickt, so empfängt kein korrekter Prozess m_2 vor m_1.

Hierbei ist der Begriff "kausal" folgendermassen definiert: Ein Ereignis a geschieht kausal vor b, $a \rightarrow b$ gdw:

1. Ein Prozess erzeugt die Ereignisse a und b, wobei a vor b erzeugt wurde.
2. a ist ein Sendeereignis einer Nachricht und das Ereignis b ist deren Empfang.
3. Es gibt ein Ereignis c, für das gilt: $a \rightarrow c$ und $c \rightarrow b$.

Ereignisse sind (insofern nicht weiter eingeschränkt) Sende- und Empfangsereignisse. Ein Sendeereignis kann z.B. mittels Multicast und Broadcast mehrere Empfänger spezifizieren (siehe Kapitel 2.3.3).

Timed Reliable Ist das Verschicken von Nachrichten "Timed Reliable", so kann man eine Konstante Δ in der Art angeben, dass jeder korrekte Empfängerprozess bei einem Sendezeitpunkt t die Nachricht bis spätestens $t + \Delta$ ausliefert.

Atomic Ein Broadcast oder Multicast ist Atomic, wenn jeder Prozess die Nachrichten in der gleichen Reihenfolge bekommt. Die Reihenfolge der Nachrichten untereinander spielt hierbei keine Rolle.

FIFO Atomic Zusätzlich zu der Atomic-Eigenschaft haben die Nachrichten noch eine FIFO-Ordnung.

Causal Atomic Zusätzlich zu der Atomic-Eigenschaft haben die Nachrichten noch eine kausale Ordnung.

2.3.2 TCP/UDP

Die Transportschicht des TCP/IP-Referenzmodelles bietet der Anwendung 2 Protokolle an: TCP (Transmission Control Protocol) und UDP (User Datagram Protocol). Da über diese Techniken im weiteren Verlauf diskutiert wird, werden hier deren Eigenschaften kurz erläutert. Details, deren Implementierung sowie Optimierung und darunter liegende Schichten werden hier nicht erklärt und es sei hierfür auf [TANE96] verwiesen.

TCP ist ein verbindungsorientiertes, datenstrombasierendes Protokoll. Das heisst, dass es nach einem expliziten Verbindungsaufbau auf beiden Seiten einen Bytestrom entgegen nimmt und auf der jeweils anderen Seite abliefert. TCP hat die folgenden Eigenschaften:

- FIFO
  Da es sich um einen Datenstrom handelt, werden die einzelnen Pakete bzw. Bytes

wieder in der richtigen Reihenfolge zusammengesetzt.

- Reliable
  Werden Daten gesendet, so erreichen diese entwerder ihr Ziel oder es wird ein Fehler signalisiert, der zum Verbindungsabbruch führt.

Kommunikation ist bei TCP immer in 3 Phasen unterteilt: Zuerst erfolgt der Verbindungsaufbau. Ist die Verbindung aufgebaut, so kann in der Datenphase über die TCP-Verbindung kommuniziert werden. Das Beenden der Kommunikation erfolgt letztendlich in der Phase Verbindungsabbau.

UDP hingegen ist ein verbindungsloses, paketbasiertes Protokoll. Während das TCP-Protokoll mit einem Telefonat vergleichbar ist (Verbindungsaufbau → Daten beliebiger Länge senden → Verbindungsabbau), gleicht UDP der traditionellen Post (Brief in Umschlag stecken, Absender angeben und abschicken). Es besitzt dementsprechend weder die FIFO-Eigenschaft, noch ist die Kommunikation reliable.

Die Vorteile der jeweiligen Protokolle liegen auf der Hand: Während TCP der Anwendung durch seine Semantik viele Überprüfungen abnimmt, ist UDP ein äußerst schlankes Protokoll: Es wird nur eine Nachricht gesendet und auf keine Bestätigung gewartet. Der Overhead für Verbindungsaufbau, Fehlerkontrolle und Reihenfolgenerhaltung entfällt.

2.3.3 Unicast, Multicast, Broadcast

Eine Nachricht kann an einen Empfänger verschickt werden, oft werden jedoch mehrere Empfänger benötigt (z.B. in einer von vielen Personen gleichzeitig genutzten Anwendung). Dabei kann es vorkommen, dass dem Sender die einzelnen Empfänger nicht einmal alle bekannt sind (z.B. bei der Bekanntgabe von Diensten). Dadurch ist das Verschicken an mehrere Ziele an sich verbindungslos realisiert. Das UDP-Protokoll unterstützt diese Zwecke durch spezielle IP-Adressen.

- Unicast
  Unter Unicast ist die klassische 1:1-Kommunikation zu verstehen. Adressiert wird die IP-Adresse des Zielrechners

- Broadcast
  Ein Broadcast ist eine Kommunikation an alle vorhandenen Rechner in einem Subnetz. Adressiert wird das gesamte Subnetz.

- Multicast
  Sollen Nachrichten an eine Teilmenge aller verfügbaren Rechner gesendet werden, so bietet sich die Technologie des Multicastes an. In der TCP/IP-Architektur wurde dies so gelöst, dass spezielle Adressbereiche für diesen Zweck definiert sind (Class D Netz: 224.0.0.0 - 239.255.255.255, siehe [IANA]).

Da Multicast nur vom UDP-Protokoll unterstützt wird, besitzt es weder eine Reliable- noch eine höher liegende Semantik. Es existiert jedoch darauf aufbauende Software wie z.B. das Light-wight Reliable Multicast Protokoll [LRMP] oder das ISIS-System [ISIS].

2.4 Anwendungskommunikation

Während in den vorherigen Kapiteln die Eigenschaften der Kommunikation auf Netzebene besprochen wurden, wird nun die Kommunikation auf Anwendungsebene erläutert.

2.4.1 Nachrichten

Die einfachste Art der Kommunikation ist das direkte Versenden von Nachrichten. Die Eigenschaften einer solchen Kommunikation sind die gleichen, wie sie bei dem UDP-Protokoll erwähnt wurden: Sie ist unbestätigt und unreliable. Der Vorteil einer solchen schlanken Kommunikation mit der damit verbundenen geringen Netzlast lässt sich zum Beispiel bei folgendem Anwendungsszenario erkennen: Audio- und Videokonferenzen.

Bei der Echtzeitübertragung ist es nötig, dass die Daten innerhalb einer gewissen Zeitspanne (Jitter) beim Empfänger vorliegen. Dabei werden zugunsten einer geringen Latenz Fehler im Bild oder kurze Störungen im Ton in Kauf genommen. Würden komplexe Fehlerkontrollmechanismen eingeführt, müssten verlorene Pakete erneut gesendet werden, was zu einer Vergrösserung des Jitters führen würde. Ein Protokoll zur Echtzeitkommunikation mittels Nachrichten ist z.B. das Real-Time Transport Protocol [SCFJ96].

Weiterhin können Nachrichten (aufgrund der unidirektionalen Kommunikation ohne Rückantworten) ohne der Verwendung komplexer Protokolle oder weiterer Nachrichten an mehrere Rechner gleichzeitig gesendet werden.

2.4.2 TCP-Sockets

Sollen größere Datenströme über eine bestehende Verbindung übertragen werden, so bieten sich TCP-Sockets an. Da das TCP-Protokoll strombasiert, reliable und FIFO ist, lassen sich Datenmengen wie Dateiinhalte oder Texte einfach übertragen. Durch das Puffern von Paketen und performanten Algorithmen wie das Sliding Window-Verfahren ist TCP geeignet, sicher und schnell große Datenmengen zu übertragen.

Da TCP Fehlerkorrektur besitzt und über einen Slowstart-Mechanismus im Fehlerfalle verfügt, sind für Echtzeitkommunikation andere Verfahren, wie bereits im vorherigen Kapitel festgestellt, besser geeignet.

Das TCP-Protokoll wird ausführlich in [TANE96] beschrieben.

2.4.3 RPC

Im "Remote Procedure Call", dem entfernten Prozeduraufruf, wird die Kommunikation auf Prozeduraufrufe abstrahiert. Im Falle von Corba [OMG] geschieht dies beispielsweise durch eine Middleware, die Aufrufe lokationstransparent entgegennimmt und

weiterleitet. Im Unterschied zu bisher genannten Verfahren sind in den Aufrufen neben den Argumenten noch zusätzliche Semantiken wie Datentypen oder Funktionsnamen enthalten. Der Programmierer verwendet die Prozeduren hierbei wie im lokalen Fall.

Bei RPCs können Prozeduraufrufe verschiedene Aufrufsemantiken haben:

- **exactly-once**: der Befehl wird genau einmal ausgeführt.
- **may-be**: es werden keinerlei Aussagen über den Aufruf gemacht.
- **at-most-once**: der Befehl wird null oder einmal ausgeführt.
- **at-least-once**: der Befehl wird mindestens einmal ausgeführt.

2.4.4 Java RMI

"Java Remote Method Invocation" [RMI] ist ein auf der Sprache Java basierendes System, welches im Gegensatz zum prozedualen RPC objektorientiert funktioniert. Es ist direkt in Java eingebunden und benötigt durch die Verwendung von Java-Interfaces keine weitere "Interface Description Language" (IDL). RMI-Klassen werden einzig durch einen weiteren Compiler (rmic) verarbeitet, welcher entsprechende Stub-Klassen erstellt.

Durch die Integration in die Java-Umgebung lassen sich sämtliche serialisierbaren Datentypen ohne weitere Spezifikationen übertragen. Das RMI-System bietet lediglich eine at-most-once-Semantik an. Bei erkannten Fehlern liefert es eine spezielle Exception.

Kapitel 3

Entwurf

Wurden bisher grundsätzliche Möglichkeiten der Fernsteuerung diskutiert, wird nun ein System zur Fernsteuerung von Powerpoint mit den in Kapitel 1.3 genannten Anforderungen entworfen. Hierbei werden die Aspekte Fernsteuerungsmethode, NUSS-Anbindung sowie die benötigte Kommunikation genauer analysiert.

3.1 Diskussion der Methoden

Im Nachfolgenden werden die Methoden, wie sie in Kapitel 2.1 aufgezählt wurden, auf ihre Tauglichkeit für diesen Fall untersucht.

3.1.1 MVC-Prinzip

Wie bereits festgestellt wurde, benötigt eine Fernsteuerung nach dem MVC-Paradigma eine dementsprechend programmierte und offengelegte Anwendung. Zwar wäre diese Methode (gerade durch die Möglichkeit von verschiedenen Sichten ...) ideal, nur sind die wenigsten verfügbaren Anwendungen hierfür ausgelegt.

Da erstens PowerPoint ferngesteuert und zweitens ein generisches Prinzip entwickelt

werden soll, fällt eine direkte Umsetzung dieser Methode aus.

3.1.2 Schnappschuss des Fensterinhaltes

Die Methode des Schnappschusses ist eine für diese Anwendung mögliche Realisierung und wird deshalb genauer untersucht. Der Aufbau eines solchen System würde folgendermassen aussehen: Auf einem PC läuft eine PowerPoint-Instanz sowie ein Programm zur

1. Erstellung und Versendung der Screenshots.
2. Rückübertragung von Aktionen.
3. Realisierung einer Floorcontrol oder anderer Rechtesysteme.

Auf Clientseite müsste sich hingegen nur eine Anwendung zur Darstellung der Bilder und dem Übertragen von Aktionen befinden.

Vorteile einer solchen Architektur wären

- die volle Kontrolle über zu steuernde Applikation
- eine Plattformunabhängigkeit
- ein generisch für beliebige Applikationen verwendbares Prinzip
- die Notwendigkeit des Erwerbs nur einer Programmlizenz

Bei einem solchen Ansatz ergeben sich jedoch folgende Nachteile und Probleme:

- Rechteverwaltung grobgranular oder komplex
  Da die interne Programmlogik nicht bekannt ist, ist entweder nur eine "alles oder nichts"-Semantik realisierbar, oder mögliche Aktionen müssen in die Fernsteuerungssoftware implementiert werden. Dadurch würde einerseits die Generizität verloren gehen, andererseits kämen somit wieder weitere Anforderungen

bezüglich Steuerbarkeit an die Anwendungen, da die jeweilige Semantik einer Benutzeraktion erkannt und unterschieden werden müsste.

- Probleme bei mehreren Aktoren gleichzeitig. (z.B. bei der Maussteuerung)
- Hoher Traffic
  Da unter Windows das Abfangen von Grafik-Primitiven nicht vorgesehen ist, müssen Screenshots erstellt werden.
- Anwendungsfenster muss serverseitig ständig sichtbar sein

Da in den Anforderungen eine feingranulare Rechtevergabe sowie das Agieren von mehreren Clients gleichzeitig vorgesehen ist, kann dieser Ansatz so nicht verwendet werden.

3.1.3 Betriebsystemoperationen

Werden Aktionen wie Mausbewegungen oder Tastatureingaben übertragen, so ergeben sich die gleichen Probleme wie bei Schnappschüssen: Die Rechtevergabe lässt sich nur sehr grobgranular gestalten (z.B. "darf Maus bewegen" oder "darf Maus nicht bewegen") und die Anwendung muss auf allen Rechnern ständig sichtbar sein, damit unter anderem Mausevents angenommen werden. Zusätzlich ergeben sich noch weitere Nachteile:

- Unterschiedliche Programmversionen können unterschiedlich auf Nachrichten reagieren.
- Unterschiedliche Bildschirmauflösungen verursachen ggf. verschiedene Anordnungen im Programmfenster.

Letztendlich muss festgestellt, dass sich dieses Verfahren grundsätzlich nicht zur interaktiven Nutzung durch mehrere Personen gleichzeitig in einem heterogenen Umfeld eignet.

3.1.4 Anwendungsoperationen

Diese Fernsteuerungsart hat folgende Vorteile:

- Geringe Netzlast.
- Feingranulere Rechteverwaltung durch Analyse der Anwendungsoperationen.
- Gleichzeitige Steuerung von mehreren Aktoren (Entweder durch atomare Aktionen oder durch deren Serialisierung).

Da bei dieser Methode eine Unterstützung von Seiten der Anwendung vorrausgesetzt wird (was z.B. in PowerPoint nicht gegeben ist), lässt sich diese Methode nicht verwenden.

3.2 Resultierendes Modell

Als Ergebnis wurde eine Mischung aus dem MVC-Paradigma, der Methode der Betriebssystem- sowie der Anwendungsoperationen gewählt.

Hierbei wird die fernzusteuernde Anwendung direkt durch einen anwendungsspezifischen Wrapper erweitert, welcher Betriebssystemoperationen abfängt und in Anwendungsoperationen (und umgekehrt) transformiert. Diese Operationen werden danach an alle Anwendungsinstanzen verteilt und somit aus Sicht des MVC-Paradigmas als Controllernachrichten an ein (an jedem Client replizierten) Model mit direkt verbundener View-Komponente gesendet. Dieses Verfahren wird im weiteren genauer erläutert.

Das resultierende Modell hat den in Abbildung 3.1 beschriebenen Aufbau.

Direkt auf dem Rechner mit der zu steuernden Anwendung befindet sich eine weitere (Wrapper-)Anwendung, welche mittels

- Betriebssystemoperationen (Fensternachrichten)

- von der Anwendung bereitgestellten Methoden

mit der ersteren kommuniziert. Microsoft PowerPoint bietet hierfür ein ActiveX-Object (siehe hierzu [MSDN]) an, mittels dessen sich eine Präsentation steuern lässt. Da PowerPoint jedoch über nahezu keine Ereignismeldungen über lokal ausgeführte Aktionen (z.B. Mausklicks, Tastatureingaben oder gar Objekterstellungen) verfügt, werden Eingaben direkt vom Betriebssystem abgegriffen und nicht an PowerPoint weitergeleitet. Nach aussen hin besitzt der Wrapper anwendungsspezifische Methoden und Ereignismeldungen (wie Blättern, Linien zeichnen oder ähnliches). Da der Wrapper somit die Anwendungslogik kennen muss (wann erzeugt ein Mausklick ein Event?), wurde das fernzusteuernde Fenster auf die Präsentationssicht von PowerPoint beschränkt und Objekte werden nicht wie üblich, sondern durch eine eigene GUI erzeugt.

Der Wrapper ist demnach nicht generisch und muss für jede zu steuernde Anwendung neu geschrieben werden. Die Schnittstelle nach aussen liefert eine fernzusteuernde Anwendung, welche das weitere System mit anwendungsspezifischen Nachrichten in einem generischen Format versorgt.

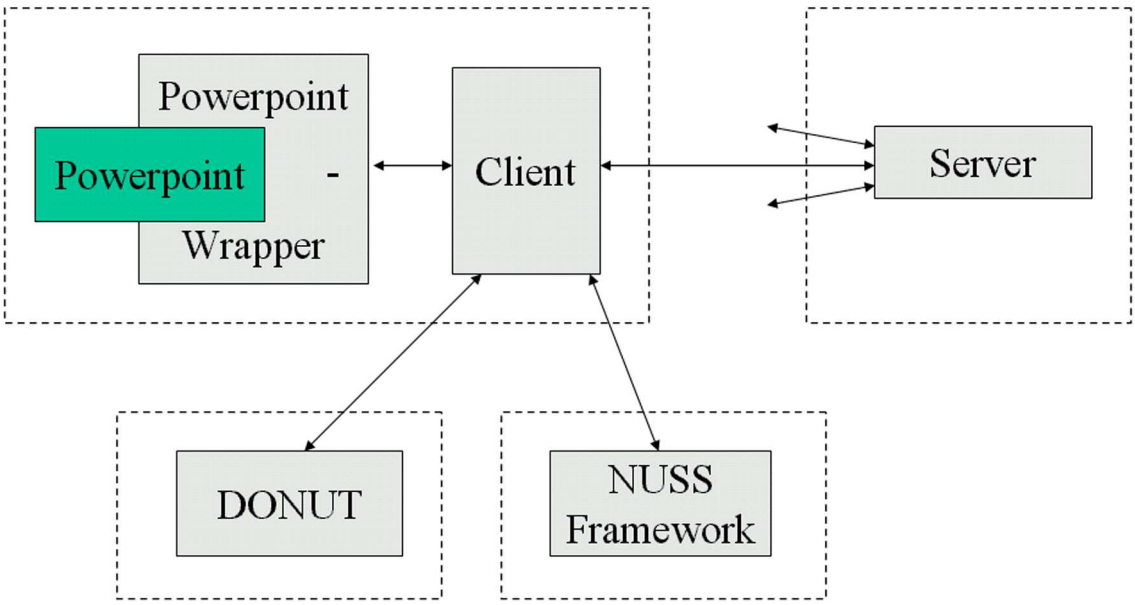

Abbildung 3.1: Systemdesign

Der Client ist eine Komponente, welche für die Kommunikation mit dem Fernsteuerungsserver zuständig ist. Seine Aufgabe ist es,

- mit dem Server zu kommunizieren.
- die Verteilung von Dateien zu organisieren.
- Server zu erstellen oder einem bestehenden beizutreten.
- Gruppen bekanntzugeben und zu suchen.
- Rechte zu überprüfen.

Als letzte Komponente existiert noch ein Server, welcher die Nachrichten der einzelnen Clients koordiniert und verteilt.

Der Server an sich kommuniziert nicht mit dem NUSS-Framework. Da der Server von einem Client aus initiiert wird, übernimmt dieser das Eintragen der Gruppe in die Registry.

3.3 NUSS-Anbindung

Das System soll an das NUSS-Framework, wie in [NOLL02], [NOLL03] und [BoZh02] beschrieben, angebunden werden. Dabei soll es dessen Möglichkeiten zur Authentifizierung, der Gruppenverwaltung und dem Rechtemanagement verwenden. Einen Überblick über den NUSS-Framework liefert Abbildung 3.2 aus [NOLL03].

Im Folgenden wird das NUSS-Framework genauer erläutert.

3.3.1 NAP - Der NUSS Access Point

Der NUSS Access Point bildet eine zentrale Anlaufstelle für alle Funktionen des NUSS-Frameworks. Um mit ihm zu kommunizieren, muss einmalig an einer wohldefinierten

RMI-Registry ein entsprechender Stub empfangen werden, über welchen jegliche weiteren Objekte für die NUSS-Kommunikation übertragen werden.

3.3.2 Authentifizierung

Das von der RMI-Registry erhaltene NAP-Objekt verfügt ursprünglich nur über Funktionen zur Benutzerkontrolle. Diese sieht derzeit nur eine Benutzername-Passwort-Authentifizierung vor, welche von einer entspechenden Oberfläche in SharedPowerPoint unterstützt werden muss. Intern ist in der Authentifizierungskomponente ein PlugIn-Konzept realisiert, welches das Anbinden beliebiger Methoden (wie z.B. der Authentifizierung mittels einer lokalen Textdatei, einem Active Directory-Server oder einem LDAP-Verzeichnis) gestattet. Im Gegensatz zu einer applikationsabhängigen Authentifizierung lässt sich demnach ein einheitliches System für alle NUSS-Anwendungen realisieren.

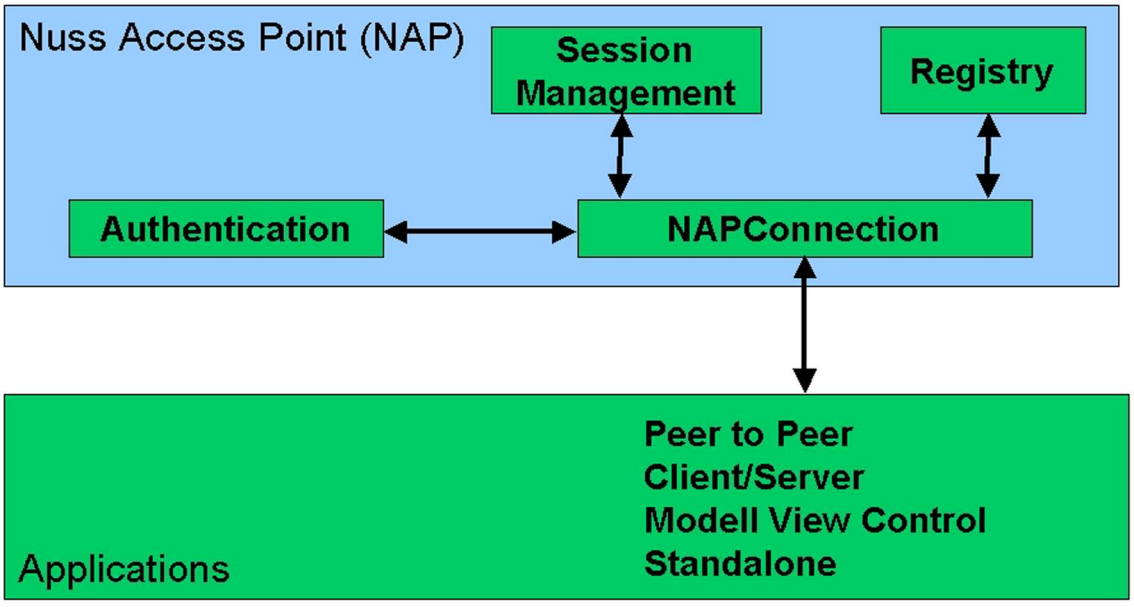

Abbildung 3.2: NUSS Access Point

3.3.3 Gruppenmanagement

Nach erfolgter Authentifizierung stehen dem Benutzer Befehle zur Auswahl und Erzeugung einer Gruppe aus der NUSS-Registry zur Verfügung. Eine Gruppe spezifiziert zum Beispiel eine SPP-Präsentation. Daher soll ein dementsprechender Eintrag bei Erstellung einer solchen gemeinsam genutzten Präsentation dort eingetragen werden. Wurde einer Gruppe beigetreten, so stehen nun gruppenspezifische Befehle zur Verfügung. Dies sind Befehle des Rechtemanagements (siehe nächstes Kapitel) sowie Möglichkeiten zur Erstellung von Untergruppen für Teilmengen der Teilnehmer.

3.3.4 Rechte

Innerhalb einer Gruppe und Untergruppe gibt es ein Rechtemanagement. Die Grundeinstellungen der Rechte (Mengen, Bezeichnungen, Einschränkungen, Kompatibilitäten, Maximalwerte, Defaultwerte) werden bei dem Erstellen der Hauptgruppe übergeben.

Innerhalb des Rechtemanagements gibt es neben Rechten noch definierbare Rollen, welche durch eine Menge von Rechten modelliert sind. So kann es in SPP beispielsweise eine Rolle "Akteur" geben, welcher mit Text und Linien annotieren und ausserdem blättern darf.

Besitzt ein Benutzer ein Recht zu Beginn nicht, so muss er innerhalb der Anwendung ein Recht beantragen können. Das Verhalten des Rechtemanagements hängt von den übergebenen Einstellungen ab. Es kann je nach vorhandener Policy sofort abgelehnt, vergeben oder nachgefragt werden. Details zu der Einführung und Realisierung dieser Konzepte sind bereits detailliert in [NOLL02] und [NOLL03] erläutert und werden hier deshalb nicht vertieft.

Zur Verwaltung von Rechten, Policies oder Rechteanfragen existiert eine generische Rechteverwaltung von [KINZ03]. Deshalb muss nur nur eine einfache Benutzerschnittstelle zum Anzeigen und Anfordern von Rechten innerhalb jeder einzelnen Anwendung

realisiert werden.

3.4 Dateiverteilung

Wird die fernzusteuernde Applikation wie in dem hier gewählten Modell bei jedem Rechner ausgeführt, so müssen auch von jedem dieser Rechner die benötigten Ressourcen verfügbar sein. Da nicht grundsätzlich von einem gemeinsamen Dateisystem ausgegangen werden kann (schließlich soll die Software in einem möglichst heterogenen Umfeld nahezu überall zum Einsatz kommen können), muss eine dementsprechende Verteilungsmöglichkeit im System bestehen. Im Fall von PowerPoint muss einzig die Präsentationsdatei (.ppt) verteilt werden.

3.4.1 Skalierbarkeit

Würden die Dateien durch den Server zu jedem Client einzeln übertragen werden, so wäre die zu übertragende Datenmenge das Produkt aus Clientzahl und Dateigrösse, sie wäre also linear von der Anzahl der teilnehmenden Clients abhängig.

Geht man von dem Szenario einer Vorlesung mit 100 Teilnehmern aus, in der ein Foliensatz mit einer Grösse von 4 MB verteilt werden muss, so würde deren Übertragung bei einem zugrundeliegendem 11 MBit WLAN und einzelnen Verbindungen zum Server weit über 5 Minuten dauern; eine Zeit, die z.B. in einer Vorlesung nicht zumutbar ist. Daher muss für die Dateiverteilung auf eine andere Technologie zurückgegriffen werden.

Aus diesem Grund wurde eine Verteilung über Multicast vorgesehen. Da es innerhalb einer Dateiübertragung keine komplexen Anforderungen wie ein Gruppenmanagement gibt, wurde das Light-wight Reliable Multicast Protocol, [LRMP] gewählt. Es handelt sich dabei um ein Protokoll, welches unter anderem folgende (für seine Auswahl entscheidende) Eigenschaften hat:

- Kommunikation rein über Multicast-Adressen.
  Das Protokoll benötigt von zwischenliegenden Routern keine weitere Funktionalität.

- Kommunikation ist FIFO und Reliable
  Dadurch lassen sich Daten wie z.B. Dateien einfach und ohne weiterer Anwendungslogik übertragen.

- Fehlerkontrolle über NACKs
  Durch das reine Rücksenden von NACKs (Not Acknowledge) entsteht bei der Verteilung keinerlei weitere Kommunikation, solange keine Fehler entstehen. Sollten diese entstehen, so werden fehlende Datenpakete mittels einer automatisch erstellten Baumstruktur von einem 'nahen' Teilnehmer der Gruppe nachgesendet, wodurch das System auch bei Fehlern skaliert.

- Angabemöglichkeit von einer Mindestdatenrate.
  Durch die Angabe einer Mindestdatenrate wird der Fall ausgeschlossen, dass durch die Teilnahme eines besonders langsamen Teilnehmers (z.B. durch Anbindung mittels Modem) die gesamte Gruppe ausgebremst wird. Zu langsame Gruppenteilnehmer signalisieren dies der Anwendung durch einen Fehler.

Da jedoch nicht grundsätzlich von einem multicastfähigen Netzwerk ausgegangen werden kann (z.B. ist die derzeitige VPN-Software von Cisco nicht multicastfähig, aber auch die meisten Internetprovider bieten dies nicht an), muss hier grundsätzlich eine Fallbacklösung mittels Unicast existieren.

Letztendlich erfolgt (wenn eine Multicast-Übertragung gewünscht wird) die Übertragung von Dateien derart, dass zuerst versucht wird, die Datei per Multicast zu empfangen. Ist dies aufgrund von Timeouts, Fehlern oder anderen Gründen nicht möglich, werden die Dateien per Unicast empfangen.

3.4.2 Kommunikationsprotokolle

Da im vorliegenden Fall von PowerPoint nur eine Datei übertragen werden soll, wird hier nur ein dementsprechend vereinfachtes Protokoll aufgezeigt. Eine Möglichkeit zur Erweiterung wird am Ende dieses Kapitels vorgestellt.

Multicast Übertragung Um den Benutzer nicht durch weitere Einstellungen zu irritieren und da es keine 1:1 Beziehung zwischen 'normalen' (Class A-C) und Multicastadressen (Class D) gibt, soll es möglich sein, dass mehrere oder alle Gruppen die gleiche Adresse benutzen.

In dem folgenden Protokoll gibt es 5 Pakettypen, welche durch einen zusätzlichen Header unterschieden werden:

- Anfrage
- Antwort
- Sendebeginn
- Daten
- Ende

Die Übertragung läuft folgendermassen ab:

1. Der Client sendet seinen Wunsch, Dateien von einem Server zu empfangen, an die Multicastgruppe (durch ein Paket vom Typ "Anfrage"). Da die Clients die Unicast-IP des Servers kennen, können sie ihn eindeutig adressieren.
2. Der Server antwortet sofort durch ein "Antwort"-Paket, in dem er seine eigene Unicast-IP zusätzlich mitsendet. Da das LRMP-Protokoll bereits über eine eindeutige Senderkennung verfügt, können Clients mit diesem Wissen alle weiteren

Pakete von dem Server ausfiltern. Da die Antwort unabhängig vom Zustand des Servers sofort zu erfolgen hat, können Clients ohne lange Wartezeit feststellen, ob der zuständige Server erreichbar ist. Ist dies der Fall, so warten diese nun auf ein Paket vom Typ "Sendebeginn".

3. Hat der Server seinen, vor der Anfrage begonnenen, Sendeprozess beendet oder ist er bereits in einem Wartezustand, so beginnt er mit dem Datentransfer. Dies geschieht durch ein Paket vom Typ "Sendebeginn" mit Metainformationen (wie Dateiname), gefolgt von entsprechend der Dateigrösse vielen "Daten"-Paketen. Danach schickt er noch ein "Ende"-Paket, um dem Client über den komplettierten Transfer zu informieren.

Um mehrere Dateien zu versenden, so müsste dieses Protokoll nur leicht abgewandelt werden. Nach dem Versenden einer Datei schickt der Server z.B. nicht ein "Ende"-Paket, sondern ein "NeueDatei"-Paket mit dem nächsten Dateinamen.

Unicast Übertragung Um den Dateitransfer mittels Unicast zu realisieren, wurde eine TCP-Verbindung gewählt.

Ein Client verbindet sich auf dem Server (auf einem durch die SPP-Server Adresse bestimmten wohldefinierten Port). Der Server beginnt darauf ohne jegliche weitere Kommunikation mit dem Übertragen der Datei. Nach Beendigung des Transfers schliesst dieser die Verbindung.

Um mehrere Dateien zu übertragen, müssten weitere Informationen in dem TCP-Strom eingeflochten werden. Dies kann beispielsweise durch eine gewisse Bytefolge (oder Bitfolge) geschehen, die signalisiert, dass als nächstes

- ein Dateiname
- ein Dateiinhalt

übertragen wird. Da beliebige Binärdateien übertragen werden sollen, muss davon ausgegangen werden, dass diese speziellen Folgen auch in den Dateien vorkommen können.

Diese Vorkommnisse müssen dann entsprechend erweitert werden, so dass diese beim Empfang erkannt und zurückkonvertiert werden. Dieses Verfahren, das "character stuffing", wird unter anderem in [TANE96] genauer erläutert.

Eine andere Alternative zur Übertragung mehrerer Dateien wäre z.B. das Schliessen der TCP-Verbindung nach jeder Datei und das automatische Senden der nächsten bei einem erneuten Verbindungsaufbau.

3.5 Fernsteuerung

Die Anbindung an die Anwendung geschieht grundsätzlich durch das in Kapitel 3.2 beschriebene Verfahren mit dem Abfangen von Events und dem Weiterleiten von anwendungsspezifischen Nachrichten. In diesem Kapitel wird der weitere Verlauf dieser Daten betrachtet.

Es wird ein reines Client-Server Modell beschrieben, in dem der Server für die Verteilung und die Benutzerverwaltung zuständig ist. Die Kommunikation wird durch eine TCP-Verbindung realisiert.

Da die Daten zu dem Zeitpunkt der Erzeugung noch einen direkten Anwendungsbezug haben (z.B. "blätter auf Folie 4"), werden zuerst die vorhandenen Rechte überprüft. Bei einem negativen Ergebnis wird der Benutzer darüber informiert. Besitzt er die entsprechenden Rechte, so werden die strukturierten Daten in ein Bytearray umgewandelt (siehe Kapitel 3.5.2) und an den Server geschickt. Dieser hat einzig die Aufgabe, die Daten an alle Clients zu verteilen. Der Umweg über den Server geschieht, da

- dadurch gesichert ist, dass alle Clients die Datenpakete in der gleichen Reihenfolge erhalten. Dies ist wichtig, weil Aktionen gegenseitig kausal abhängig sein können (z.B. das Löschen eines von einem anderen Client erstellten Objektes).
- durch das Senden eines Paketes an den Server - mittels einer TCP-Verbindung - gesichert ist, dass entweder **alle** anderen (korrekt funktionierenden) Clients oder

keiner der Clients dieses Paket bekommt.

- die Verwaltung der Clients nur an einer Stelle geschehen muss.

Empfängt ein Client ein solches Paket, so wandelt er es zurück in das ursprüngliche Datenformat und führt die darin spezifizierte Anwendungsaktion durch.

3.5.1 Skalierbarkeit

Betrachtet man die Gründe für die Einführung des Servers, so muss man feststellen, dass die benötigte Semantik für die Fernsteuerung die eines "Atomic Multicastes" ist. Da die übertragenen Datenmengen relativ klein sind, wurde auf die Verwendung eines dementsprechend komplexen Multicast-Protokolles wie z.B. vom ISIS-Protokoll angeboten (siehe [ISIS]) abgesehen. Diese Protokolle benötigen ebenfalls einen Prozess, welcher die Reihenfolge festlegt (siehe [ROTH03]). Die Nachrichtenzahl an sich würde sich demnach durch den Multicast-Protokoll-Einsatz nicht verringern.

Die Anzahl der übertragenen Nachrichten pro Aktion beträgt $\mathbf{n + 1}$, also eine Nachricht an den Server und von dort aus je eine Nachricht an jeden der Clients. Geht man von einer durchschnittlichen Paketgrösse von 40 Byte, eine maximalen Clientzahl von 100 Rechnern und 100 Aktionen pro Sekunde aus, so kommt man auf eine Last von

$100 \frac{1}{s} * (100 + 1) * 40 \text{ Byte} = 404000 \frac{B}{s} = 404 \frac{kB}{s}$

Da eine solche Datenmenge über ein 11 MBit WLAN transportiert werden kann und die meisten Szenarien weitaus kleiner sind, wurde dieses Modell gewählt. Da sich die Szenarien wie auch die verfügbaren Netze während des Einsatzes ändern können, so ist in der Implementierung darauf zu achten, das die Client-Server-Kommunikation streng gekapselt wird.

3.5.2 Kommunikationsprotokolle

Der Server verfügt über 2 Funktionen. Die Erste ist das reine Weiterleiten von Datenpaketen. Die Zweite ist eine Zuordnung zwischen der in allen Paketen enthaltenen Benutzernummer und dem Namen, unter welchem sich dieser Client beim Verbindungsaufbau angemeldet hat.

Im Folgenden wird zuerst auf das Datenformat eingegangen, in welches die Aktionen transformiert werden (marhalling).

Übertragen werden grundsätzlich nur die folgende Grunddatentypen:

Bezeichnung	Inhalt
Byte	1 Byte unsigned (0..255)
Int	2 Byte unsigned (0..65535), Aufbau [MSB—LSB]
String	Stringlänge Int + Zeichenfolge ISO 8859-1

Tabelle 3.1: Datentypen

Ein zu übertragendes Paket hat **immer** den folgenden Anfang:

- Int Size - Die Groesse des Paketes inklusive diesem Header
- Int Type - Die Art des Befehles.

Abhängig vom Wert des "Type"-Feldes folgen befehlsabhängige Inhalte.

Durch einen solchen Aufbau der Datenpakete lassen sich auf einer TCP-Verbindung (welche bereits die Semantiken "FIFO" und "Reliable" (siehe Kapitel 2.3.1) bietet) einzelne Pakete transportieren und einzeln dekodieren. Für den Aufbau der einzelnen Pakete sei auf die Codedokumentation verwiesen.

Kapitel 4

Implementierung

Während sich der Entwurf hauptsächlich mit grundlegenden Designprinzipien beschäftigt hat, wird nun die konkrete Realisierung vorgestellt. Da hier nur die wichtigsten Komponenten beschrieben werden, ist bei Detailfragen die Codedokumentation zu konsultieren

4.1 Verwendete Programmiersprachen

Als Programmiersprachen wurden Java 2 1.4.01 [JAVA] sowie Microsoft Visual Basic 6 Service Pack 5 [VB] verwendet. Die Aufteilung in 2 Sprachen hat die folgende Gründe:

Java

- Java bietet eine direkte Unterstützung von Java RMI [RMI]. Da diese vom NUSS-Framework sowie der meisten weiteren Anwendungen darin (wie DONUT) ausschließlich verwendet wird, bietet sich Java an.
- Aufgrund seiner Portabilität lassen sich Teile des Codes für weitere Anwendungen (auch auf Nicht-Microsoft Betriebssystemen) einsetzen. Da die gesamte Client-

Server-Kommunikation auf Java-Ebene läuft, lassen sich weitere Java-Systeme (wie derzeit ein Audio-Aufzeichnungsprogramm) ohne weitere Probleme in das System einbinden.

Visual Basic (VB)

- Visual Basic bietet eine einfache Unterstützung von ActiveX Objekten. Da PowerPoint auf diese Art gesteuert wird, ist eine einfache Anbindung wünschenswert.
- Da bei der Steuerung einer fremden Anwendung relativ viele Betriebssystemaufrufe nötig sind (Fensterhandles suchen, Fenster verstecken oder umpositionieren, Events abfangen, etc.), eignet sich eine Sprache wie JAVA weniger. Selbst bei der Verwendung von JNI (Java Native Interface, siehe [JNI]) müssten spezielle Wrapperklassen (in beispielsweise C) erstellt werden.

4.2 Komponenten

Das SPP-System besteht aus 3 getrennten Komponenten:

- Die graphische Oberfläche sowie die PowerPoint-Fernsteuerung.
  Diese Komponente ist komplett in Visual Basic geschrieben. Ihre Aufgabe ist das Führen sämtlicher Benutzerdialoge zu Beginn. Während der Präsentation erfolgt zusätzlich das Abfangen von PowerPoint-Eingaben sowie deren Umwandlung in Befehle.
- Der Server.
  Der Server hat die Aufgabe, Befehle an die Clients zu verteilen und ein Verzeichnis der Benutzernamen zu führen.
- Der Client.
  Hierunter ist der Teil "unter" der VB-Komponente zu verstehen. Diese ober-

flächenlose Komponente empfängt einerseits Befehle der GUI und leitet diese je nach Art

- an den Server
- an den NAP
- an DONUT

weiter. Dasselbe gilt für die Gegenrichtung.

4.3 Kommunikation zwischen den Komponenten

Als Technologie für die Verbindung der Komponenten wurde bei der Kommunikation VB↔Java, wie auch bei Client↔Server eine Socketverbindung gewählt. Diese wird zu Beginn der Kommunikation aufgebaut. Bei einem Verlust einer der Verbindungen werden die jeweiligen Komponenten beendet. Dadurch wird gewährleistet, dass das Programm beim Schliessen von PowerPoint oder bei Fehlern komplett terminiert.

Ausserdem wurde jeweils die gleiche Kommunikationsart gewählt, um somit das gleiche Protokoll verwenden zu können. Für die Kommunikation zwischen der Oberfläche und dem Client sind jedoch zusätzliche Nachrichten spezifiziert.

Die Nachrichten definieren hierbei entfernte Methodenaufrufe. Diese sind je nach aufgerufener Methode teils blockierend (z.B. die Nachfrage nach einem Recht), teils nicht blockierend (z.B. eine DONUT-Annotation) und können in beide Richtungen erfolgen.

4.4 Systemübersicht

Im folgenden werden die einzelnen Teile des in Kapitel 3.2 beschriebenen Modells vertieft.

4.4.1 Visual Basic

Die Visual Basic Komponente besteht im Grundsatz aus den folgenden Modulen:

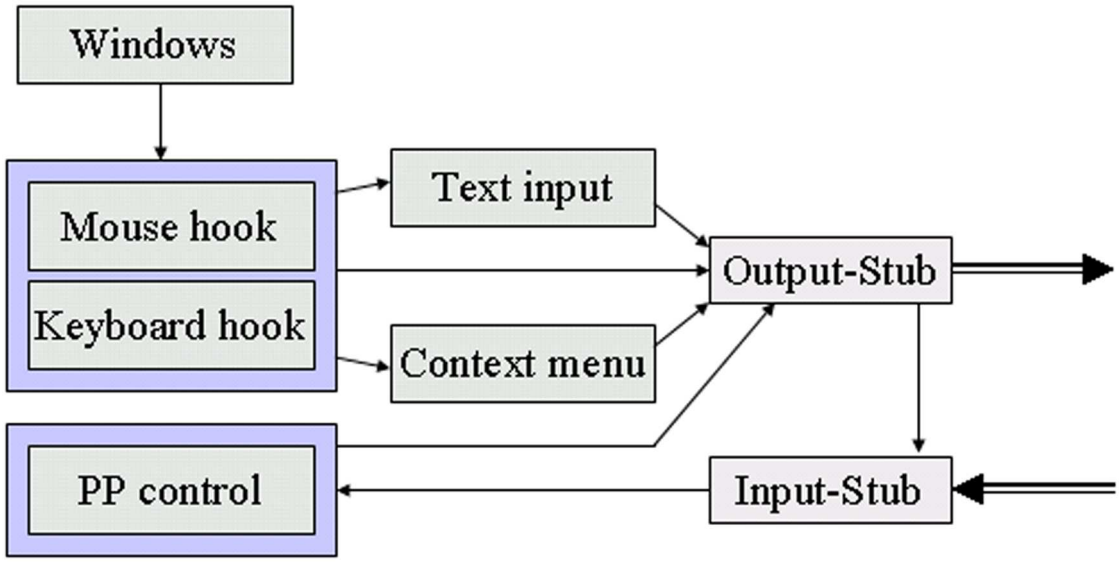

Abbildung 4.1: Visual Basic Architektur

Die Anbindung an PowerPoint erfolgt durch 3 Module:

- Die direkte Steuerung sowie die komplette Präsentation sind über das von PowerPoint angebotene ActiveX-Objekt anzusprechen. Das Modul ***PP control*** verwendet diese und besitzt dementsprechend folienspezifische Funktionen zum Erstellen von Texten und Linien, dem Blättern sowie inhaltsspezifische Funktionen, wie das Picken von Objekten an bestimmten Positionen.

- Das Modul ***Keyboard hook*** wird verwendet, um Tastaturevents, die in PowerPoint erfolgen, abzufangen und an einen eigenen Eventhandler weiterzugeben. Damit können Befehle zum Blättern abgefangen und in SPP verteilt werden. Da Windows zum Umleiten von Events verlangt, dass sich die Zieladresse im gleichen Speicherbereich wie die Anwendung befindet, wurde hier auf eine externe DLL zurückgegriffen (siehe auch Kapitel 4.5 sowie [TUEM]). Diese bietet die Umleitung auf einen eigenen Prozedurrumpf an.

- Das ***Mouse hook***-Modul vollendet das Umschliessen der Anwendungen, in dem es sämtliche Mausevents an PowerPoint abfängt und diese ebenfalls an eine interne Prozedur leitet. Ab dieser Stelle ist sichergestellt, dass die Anwendung vom Benutzer direkt keine Eingaben entgegen nehmen kann, welche den Systemzustand einseitig verändern können.

Werden Eingaben abgefangen, so muss aufgrund des gewählten Modells die Systemlogik nachimplementiert werden. Im Fall von SPP ist dies unter anderem das Zeichnen von Linien und die Eingabe eines Textes. Um den jeweiligen Modus festzulegen, steht neben einem Fenster mit Werkzeugleiste auch ein Kontextmenü zur Verfügung. Dieses Modul (***Context menu***) wird aufgerufen, wenn ein Tastaturevent "Rechte Maustaste gedrückt" empfangen wird. Anstelle des PowerPoint-Kontextmenüs wird nun ein eigenes angezeigt. Ist darin der Modus "Textannotation" gewählt, so wird bei einem linken Mausklick auf einer beliebige Stelle der Präsentation ein Fenster geöffnet, welches eine Eingabe von Fliesstext und einer Schriftgrösse erlaubt. Dieses Fenster erzeugt bei einem Klick auf "OK" eine Nachricht zum Erstellen der entsprechenden Textannotation.

Alle Nachrichten (Textannotationen, Blättern, Linien) werden an den sogenannten ***Output-Stub*** gesendet. Dieser überprüft zuerst, ob "Lokale Annotationen" ausgewählt ist und schickt die Nachricht in diesem Fall ohne eine weitere Überprüfung an den *Input-Stub*. Ist dies nicht der Fall, so wird nun geprüft, ob alle benötigten Rechte vorhanden sind und es wird im negativen Fall eine Benutzermeldung angezeigt. Sind alle Rechte vorhanden, so wird letztendlich der Befehl, wie in Kapitel 3.5.2 beschrieben, in ein Bytearray verpackt und das Paket an den Java-Teil geschickt.[1]

Der ***Input-Stub*** hat die Aufgabe, empfangene Pakete wieder in seine ursprüngliche Form zu bringen (unmarshalling) und diese zu verteilen. Dafür ruft das Modul entsprechend des Pakettyps die vorgesehenen Funktionen im **PP control**-Modul auf. So erstellt das Modul beispielsweise bei einem Paket vom Typ "Textannotation" ein neues

[1] Wird kein Java verwendet, so besteht die Möglichkeit, das Paket an eine IP-Adresse zu broadcasten. In diesem Fall kommt das gesendete Paket danach bei allen Clients, die es empfangen, direkt im *Input-Stub* an.

Textobjekt, setzt darin die Eigenschaften Text, Farbe und Schriftgrösse und positioniert diese Annotation an der ursprünglich angegebenen Position.

4.4.2 Java

Der verwendete Java-Teil besteht im Kern aus den folgenden Klassen:

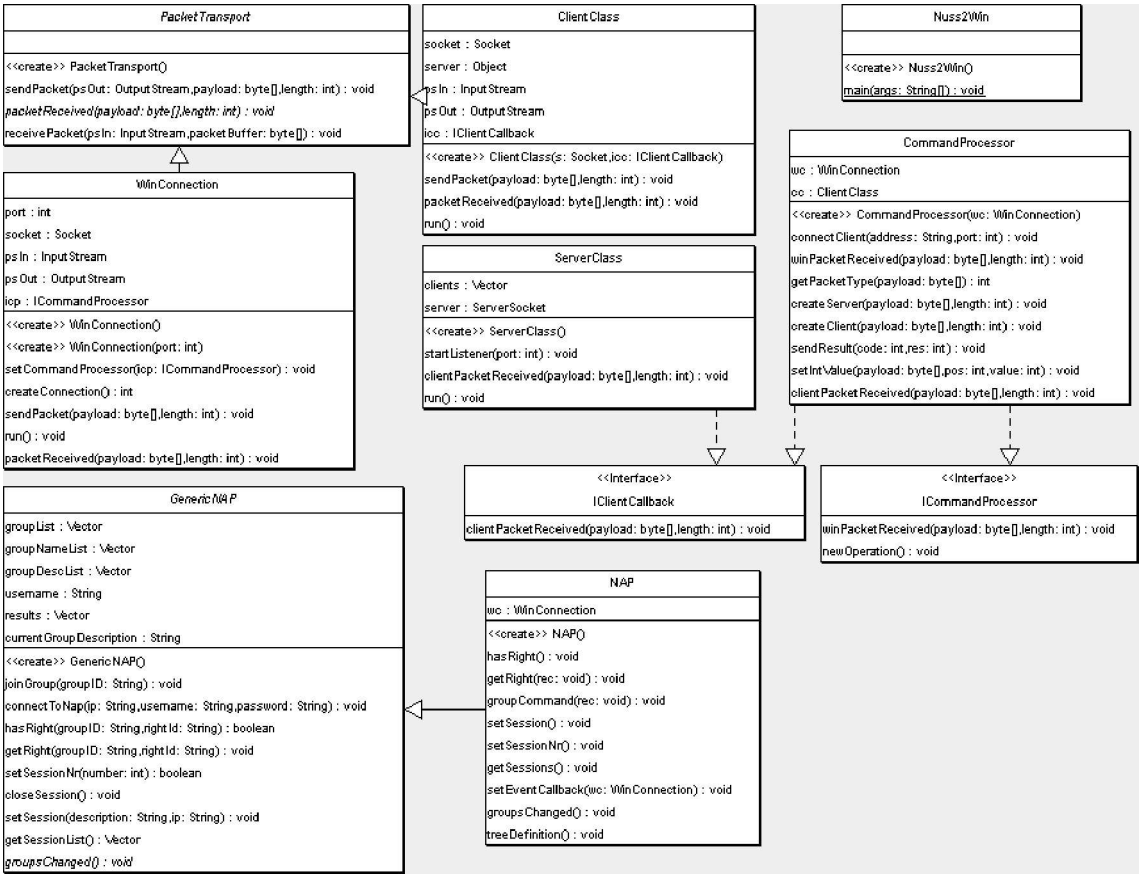

Abbildung 4.2: Java Architektur

Klassen für die Dateiverteilung oder die Donut-Anbindung sind hierbei der Übersichtlichkeit halber nicht enthalten.

Im Folgenden werden die einzelnen grundlegenden Klassen kurz beschrieben:

- ***PacketTransport***

  Die Klasse *PacketTransport* dient zur Übertragung von Paketen über einer bestehenden TCP-Verbindung. Da diese Kommunikationsart zwischen Client↔Server sowie zwischen VB↔Java verwendet wird, stellt sie eine grundlegende Basisklasse dar. Ihre Hauptaufgabe ist das Lesen von eingehenden Daten, das Dekodieren des Headers und die Weiterverteilung vollständiger Pakete an abgeleitete Klassen.

- ***ClientClass, WinConnection***

  Diese beiden Klassen sind die Realisierungen der Klasse *PacketTransport* und dienen der jeweiligen Kommunikation. Eingegangene Pakete melden sie an eine jeweils übergebene Klasse. Die *WinConnection* hat zur Anbindung an das VB-Programm eine andere Callbackfunktion sowie ein erweitertes Verbindungsmanagement.

- ***CommandProcessor***

  Der *CommandProcessor* bildet das zentrale Bindeglied zwischen den beiden Verbindungen. Es empfängt von *ClientClass* wie *WinConnection* Pakete und leitet diese entsprechend weiter. Befehle für SPP-Aktionen werden beispielsweise an die jeweils andere Kommunikationskomponente, Rechteanfragen an die NAP-Komponente und DONUT-Befehle an den *DonutClient* weitergeleitet. Meldet eine der Netzwerkverbindungen einen Verbindungsabbruch, so wird entsprechend auch die andere Verbindung geschlossen.

- ***ServerClass***

  Diese Klasse realisiert den Server. Sie wartet auf Verbindungswünsche und erstellt daraus eine Liste von *ClientClass*-Instanzen, welche mit der korrespondierenden *ClientClass* auf dem Client-Rechner verbunden ist. Wie der *CommandProcessor* ist auch diese Klasse für die Nachrichtenverteilung zuständig. SPP-Aktionen schickt diese jedoch an alle vorhandenen *ClientClass*-Instanzen. Des weiteren werden Pakete zur Namensauflösung durch ein Ergebnispaket an die empfangende *ClientClass* beantwortet

- ***GenericNAP***
  Der *GenericNAP* ist eine Klasse mit grundlegenden Funktionen zur Kommunikation mit dem NUSS Access Provider. So bietet sie Funktionen zum Login, dem Erstellen einer Sessionliste, zum Überprüfen von Rechten und zur Untergruppenverwaltung.

- ***NAP***
  Während der *GenericNAP* eine allgemeine NAP-Anbindung realisiert, bietet diese Erweiterung anwendungsspezifische Funktionen an. So sendet es beispielsweise bei RPC-basierten Aufrufen (wie z.B. Rechteabfragen) direkt das Antwortpaket an die WinConnection zurück.

4.5 Verwendete Externe Module

Für die Entwicklung von SPP wurde auf folgende existierende Produkte zurückgegriffen. Nachstehend werden die Produkte, deren Einsatszweck sowie die Gründe für deren Verwendung diskutiert:

- NUSS-Framework, NUSS Access Point
  Da SharedPowerPoint eine Anwendung, entwickelt im Rahmen des NUSS-Projektes ist, ist eine Anbindung an deren Framework sowie die Nutzung dessen Funktionalität implizit gegeben. Es wird für Authentifizierung, Rechteverwaltung, Gruppen und Untergruppenmanagement verwendet.

- Microsoft PowerPoint
  In SPP wird auf jedem Rechner eine laufende PowerPoint Instanz verwendet.

- DSKEYBRD.DLL von Dr. Jürgen Thümmler, [TUEM] Diese Freeware-DLL bietet die Möglichkeit, Keyboardhooks (Abfangen von entsprechenden Meldungen des Betriebssystems) bei externen Programmen zu verwenden. Da sich die dementsprechende Funktion im Speicherbereich der abzufangenden Anwendung befinden muss, erschien eine Eigenimplementierung als zu aufwendig.

- DSMOUSE.DLL von Dr. Jürgen Thümmler, [TUEM] Diese DLL ist das Pendant zur DSKEYBRD.DLL, jedoch zum Abfangen von Mausevents.

- Light-weight Reliable Multicast Protocol, [LRMP] Das Light-weight Reliable Multicast Protocol ist ein freies vom Massachusetts Institute Of Technology (MIT), INRIA entwickeltes Protokoll zur Realisierung eines Reliable Multicast. Die Lizenzbedingungen sind in C.2 zu finden.

Kapitel 5

Evaluation

In diesem Kapitel werden zunächst die möglichen Einsatzszenarien erarbeitet. Danach wird auf den bisherigen Einsatz eingegangen und die daraus resultierenden Ergebnisse analysiert.

5.1 Mögliche Szenarien

Im Folgenden werden Beispielszenarien für die Verwendung von SharedPowerPoint genannt. Dabei wurde versucht, jeweils auf Einzelaspekte einzugehen. Die Möglichkeit, das System in allen oder mehreren Szenarien gleichzeitig zu betreiben, ist jedoch vorgesehen.

5.1.1 Präsentationsunterstützung

Ausstattung: 1 PC an Beamer angeschlossen, 1 Notebook im Blickfeld des Präsentierenden.
Ablauf: Während auf dem Beamer eine normale Präsentation abläuft, werden auf dem Bildschirm (den der Vortragende in seinem unmittelbaren Blickfeld hat) zusätz-

lich Notizen, die Uhrzeit, der Stand in einem vorher festgelegten Zeitplan und ggf. sogar modifizierte Präsentationsfolien angezeigt. Dadurch kann der Vortragende besser auf angedachte Punkte eingehen und die Präsentation wirkungsvoller planen. Zusätzliche Handzettel werden damit nicht mehr benötigt.

5.1.2 Interaktive Lehrveranstaltungen

Ausstattung: 1 PC ggf. an einen Beamer angeschlossen, auf Wunsch Notebooks bei den Teilnehmern.
Ablauf: Da sich persistente Annotationen nachträglich fest in der PowerPoint-Präsentation befinden, lassen sich Ergebnisse während der Veranstaltung interaktiv erarbeiten und sie sind danach auch für die Anwesenden (im Gegensatz zu klassischen Tafelaufschrieben) zur Nachbereitung verfügbar. Beispiele hierfür sind Zusammenfassungen, Ansammlungen von Schlagwörtern, Brainstorming-Ergebnisse, Ablaufdiagramme oder Fragebögen. Diese können entweder direkt von den Anwesenden oder (sollte nur ein Rechner zur Verfügung stehen) durch den Vortragenden als Mittelsmann eingetragen werden. SharedPowerPoint stellt für dieses Szenario (zum Beispiel durch die Möglichkeit des Verschiebens bereits vorhandener Objekte) eine Ansammlung von Werkzeugen bereit.

5.1.3 Papierlose Anmerkungen

Ausstattung: Notebooks bei den Teilnehmern.
Ablauf: Durch die Möglichkeit der privaten Annotation können Teilnehmer an Vorträgen einerseits Zeichnungen und Texte in Folien einfügen; zusätzlich aber auch Freitexte (welche sich später in den Notizen zur Folie befinden) notieren und in der Nachbereitung verwenden.

5.1.4 Unterstützte Fragen

Ausstattung: 1 PC an Beamer angeschlossen, Notebooks bei den Teilnehmern.
Ablauf: Werden Fragen zu Folien gestellt, so bietet SharedPowerPoint eine Reihe von Hilfen:

- Zeiger: Ein Zuhörer kann für alle sichtbar auf eine Stelle zeigen.
- Temporärer Stift: Es können Abläufe eingezeichnet und Bereiche markiert werden, ohne das diese Markierungen später abgespeichert werden. Dadurch sinkt zusätzlich die Hemmschwelle der Interaktion.

5.1.5 Gruppenarbeit

Ausstattung: Notebooks bei den Teilnehmern.
Ablauf: Teilnehmer können sich Aufgaben - wie das Kommentieren oder das Zusammenstellen von Ergebnissen - teilen, indem diese eine Untergruppe bilden und in eben dieser annotieren. Für die spätere Vorstellung der Ergebnisse sendet ein Gruppenteilnehmer diese an den Präsentationsrechner.

5.2 Einsatz in Vorlesungen

Seit dem 09.01.2003 wurde das Programm im Rahmen der Vorlesung "Kooperation in verteilten Systemen" von Dr. Burger im Rahmen eines Beta-Tests eingesetzt. Es löste hierbei das bis dahin verwendete SASCIA-System [SASCIA] ab. Abbildung 5.1 zeigt dessen Verwendung in der Vorlesung: Während die Vorlesungsfolien primär auf einem Smartboard präsentiert werden, stehen den Teilnehmern zusätzlich Notebooks zur Verfügung. Wer kein eigenes Notebook besitzt und auch nicht von der Möglichkeit des Ausleihens Gebrauch gemacht hat, kann alleine oder in Zweiergruppen an zusätzlich aufgestellten Notebooks arbeiten.

Abbildung 5.1: Notebookeinsatz

Innerhalb dieser Vorlesung wurde SharedPowerPoint für folgende Zwecke verwendet:

- Nachträgliches Vervollständigen von Folien
  Zusammenfassungen und Sammlungen von Eigenschaften wurden interaktiv gelöst. Dabei wurden die Inhalte auf vorher (teilweise) leeren Folien von den Teilnehmern mündlich und letztendlich schriftlich erarbeitet.

- Präsentationsfolien mit kleingedruckten Inhalten
  In grösseren Beispielen musste nicht auf die Lesbarkeit von Details in den hinteren Reihen Rücksicht genommen werden. Dadurch liessen sich komplexe Diagramme oder Codebeispiele anhand einer Übersichtsfolie realisieren.

- Unterstützung der Zuhörer
  Teilnehmern der Vorlesung war es möglich, sich die letzten Folien anzusehen, ohne vorher ein Skript oder einen Foliensatz ausdrucken zu müssen. Dadurch liessen sich Verständnisprobleme vermeiden und Zwischenfragen besser vorbereiten.

- Bessere Nachbereitung
  Die während der Vorlesung erstellten Foliensätze wurden von Dr. Burger für die Teilnehmer erreichbar in das Internet gestellt. Dadurch liessen sich auch für diejenigen, die kein Notebook besitzen oder an einer Vorlesung nicht teilnehmen konnten, die Folienkommentare downloaden.

- Einsatz von DONUT
  Ab dem 13.01.2003 wurde zusätzlich das aus SPP verwendbare DONUT-System eingesetzt. Dadurch konnten Teilnehmer Fragen zu bestimmten Punkten auf den einzelnen Vorlesungsfolien auch über dieses System stellen.

5.3 Feedback

Im Rahmen des NUSS-Projektes hat die Abteilung "Pädagogik" des Institutes für "Erziehungswissenschaft und Psychologie" unter anderem auch während und nach der oben beschriebenen Vorlesung Interviews mit den Teilnehmern geführt.

Es kristalliesierten sich dabei folgende Meinungen heraus:

- SharedPowerPoint wurde positiv entgegengenommen und als eine Verbesserung zu der vorherigen Software empfunden.
- Gerade die Möglichkeit, private Kommentare einfach eingeben zu können, wurde von den Vorlesungsteilnehmern für wichtig empfunden.
- Programmfunktionen sind teilweise nicht intuitiv oder zu komplex zu bedienen. So haben manche Studenten nicht die privaten Kommentarfelder gefunden.
- Viele Teilnehmer empfinden den verwendeten Softwareeinsatz bei einer kleinen Vorlesung für übertrieben, sahen aber die Vorteile in grossen Vorlesungen.
- Eine Smartboard-Unterstützung sowie die Eingabe von handschriftlichen Notizen wäre zu begrüssen.

- Gerade im Lehreinsatz, wo Programme wie SPP unterstützend im Hintergrund laufen sollen, ist es nötig, dass diese wenig Fehler besitzen und mit einem geringen Zeitaufwand zu betreiben sind.

Kapitel 6

Ausblick

6.1 Mögliche Erweiterungen und Verbesserungen

Aus den bisherigen Tests bei unterschiedlichen Benutzergruppen haben sich die folgenden Wünsche einer Erweiterung ergeben:

- Editieren von Texten
  Ein immer wieder geäusserter Wunsch während des Einsatzes war die Editiermöglichkeit von Textannotationen und auch Folieninhalten.

- Anbindung an einen Protokollierungsdienst zur späteren Wiedergabe.
  Derzeit in der Entwicklung ist ein Protokollierungsdienst, mit dem sich eine SPP-Sitzung inklusive Audio mitloggen lässt und später inklusive aller SPP-Aktionen wieder abspielbar ist.

- Einführung von Sicherheitskonzepten.
  Da sich im NUSS-Framework bisher nur ein schwaches Sicherheitskonzept befindet, wurde dieses in SPP übernommen. Ausgegangen wird hierbei von "well behaved" Anwendungen, es wird also davon ausgegangen, dass keine modifizierten Programme eingesetzt werden.

- Persistente Präsentationen
  Ein weiterer Verbesserungspunkt ist die Möglichkeit persistenter Präsentationen. Hierbei kann entweder das reine Fortsetzen einer Präsentation verstanden werden, aber auch das "Einfrieren" einer kompletten Gruppe inklusive aller Benutzer und deren Rechte.

- Auswertung von NAP-Callbacks
  In der aktuellen Version werden die vom Rechtemanagement angebotenen Callbacks nicht ausgewertet.

- Unterstützung der Microsoft TabletPC-Erweiterungen.
  Microsoft Powerpoint XP verfügt in seiner Version für TabletPC's über sogenannte Ink-Objekte. Diese lassen sich in PowerPoint auch während der Präsentationsansicht erstellen. Dadurch können zusätzlich Freihandzeichnungen erstellt werden. Da sich die in SPP angebotenen Linieneingaben aufgrund der langsamen Geschwindigkeit nicht für eine Schrifteingabe eignen, ist deren Integration ein klares Ziel für die weitere SPP-Entwicklung.

- Wechseln des Foliensatzes während einer Vorlesung
  Wird ein Foliensatz beendet, so sollte es möglich sein, innerhalb einer existierenden Gruppe einen neuen Foliensatz zu verteilen.

6.2 Rückblick

Die Entwicklung dieser Software war im Vergleich zu den Vorgehensplänen des Software Engineerings ein eher untypisches Projekt. Die Anforderungen an die Software ergaben sich oftmals erst im direkten Test (mittels Prototypen) mit Vorlesungsteilnehmern und Vortragenden. Ebenso war durch die gleichzeitige Konzeption und Erstellung des NUSS-Frameworks nur eine stark evolutionäre Entwicklung möglich, in der nach und nach Features eingeplant sowie eingebunden wurden.

Die Auswirkungen sind im Zeitplan zu erkennen, in dem die einzelnen Abschnitte nur

einen zentralen Fokus darstellen und Entwicklung sowie Einsatz sich ständig abwechseln.

6.3 Weitere Schritte

Der weitere Ablauf ist geprägt durch die Weiterentwicklung von SharedPowerPoint (siehe 6.1), die Wartung sowie die Unterstützung des Einsatzes in verschiedenen Vorlesungen ab dem Sommersemester 2003.

6.4 Fazit

Abschliessend ist festzustellen, das die Entwicklung einer PowerPoint-Fernsteuerung mit Annotationsmöglichkeiten erfolgreich verlief. Das System wird eingesetzt und als Verbesserung zu dem bis dahin verwendeten SASCIA-System angesehen. Durch die Weiterentwicklung wird gewährleistet, dass die Software verbessert wird und weitere Anforderungen von Vorlesungsteilnehmern in das Produkt einfliessen.

Literaturverzeichnis

[BCNW02] Detlef Bosau, Holger Cermann, Jochen Noller, Arno Wacker. NAP Spezifikation. Online verfügbar unter `http://www.informatik.uni-stuttgart.de/ipvr/vs/de/projects/NUSS/index.html`

[BoZh02] D. Bosau, Y. Zhou: *NUSS Registry Spezifikation.* Online verfügbar unter `http://www.informatik.uni-stuttgart.de/ipvr/vs/de/projects/NUSS/index.html`

[BURG97] C. Burger. *Groupware.* dpunkt.verlag, 1997

[LRMP] INRIA. *Light-wight Reliable Multicast Protocol.* Online verfügbar unter `http://webcanal.inria.fr/lrmp/`

[OMG] Object Management Group. Online verfügbar unter `http://www.omg.org`

[ROTH01] K. Rothermel. *Skript zur Vorlesung "Grundlagen der Verteilten Systeme".* SS 2001

[SASCIA] SASCIA-Projekt der Universität Stuttgart. Online verfügbar unter `http://www.informatik.uni-stuttgart.de/ipvr/vs/projekte/Festival/sascia.html`

[TANE96] A. Tanenbaum. *Computer Networks.* Prentice Hall, 3rd edition, 1996

[IANA] IANA: *Internet Multicast Addresses.* Online verfügbar unter `http://www.iana.org/assignments/multicast-addresses`

[ISIS] Cornell University. *The ISIS Project*. Online verfügbar unter `http://www.cs.cornell.edu/Info/Projects/ISIS/`

[JAVA] SUN Microsystems. *The Source for Java Technology*. Online verfügbar unter `http://java.sun.com`

[JNI] SUN Microsystems. *Java Native Interface (JNI) Specification*, Revision 1.1, 2003. Online verfügbar unter `http://java.sun.com/j2se/1.4/docs/guide/jni/`

[KINZ03] D. Kinzler. *Nutmin-Beschreibung*. Online verfügbar unter `http://www.informatik.uni-stuttgart.de/ipvr/vs/de/projects/NUSS/index.html`

[KrPo88] G. Krasner, S. Pope. *A Description of the Model-View-Controller User Interface Paradigm in the Smalltalk-80 System*, 1988

[MSDN] Microsoft Corporation. *Working with Microsoft PowerPoint Objects*. Online verfügbar unter `http://msdn.microsoft.com/library/default.asp?url=/library/en-us/modcore/html/deovrworkingwithmicrosoftpowerpointobjects.asp`

[NAGY03] M. Nagy. *Zwischenfragen in Augmented Lectures*. Diplomarbeit Nr. 2039, IPVS, Universität Stuttgart

[NOLL02] J. Noller. *Sessionmanagement Spezifikation*. Online verfügbar unter `http://www.informatik.uni-stuttgart.de/ipvr/vs/de/projects/NUSS/index.html`

[NOLL03] J. Noller. *Hierarchische Floorcontrol und Filterung von Primitiven - Generische Rechteverwaltung in hierarchischen Gruppen*. Studienarbeit Nr. 1873, IPVS, Universität Stuttgart

[NUSS] *NUSS-Projekt der Universität Stuttgart*. Online verfügbar unter `http://www.informatik.uni-stuttgart.de/ipvr/vs/en/projects/NUSS/index.html`

[OBER01] P. Oberparleiter. *Vergabe von Zugangsberechtigungen für die gemeinsame Nutzung von Smartboards in der Vorlesung*. Studienarbeit Nr. 1807, IPVR, Universität Stuttgart

[OBER02] P. Oberparleiter. *Eine Architektur zur Nutzung von Anwendungen in der Lehre mit prototypischer Realisierung von elektronischer Tafel und Leinwand*. Diplomarbeit Nr. 1950, IPVR, Universität Stuttgart

[RMI] SUN Microsystems. *Java Remote Method Invocation Specification*, Revision 1.8, 2002. Online verfügbar unter `http://java.sun.com/j2se/1.4/docs/guide/rmi/`

[ROTH03] K. Rothermel. *Skript zur Vorlesung "Verteilte Systeme 2"*. WS 2002/2003

[SCFJ96] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobsen. *RTP: A Transport Protocol for Real-Time Applications*. RFC 1889. 1996.

[TUEM] Dr. Thümmler-DLL Pack. Online verfügbar unter `http://www.it-berater.org/ThueDownloads/index.shtml`

[VB] Microsoft Corporation. *Visual Basic 6 Dokumentation*. Online verfügbar unter `http://msdn.microsoft.com/library/default.asp?URL=/library/devprods/vs6/vbasic/vbcon98/vbstartpage.htm`

[X] X Window System. Online verfügbar unter `http://www.x.org/X11.html`

Anhang A

Benutzerhandbuch

A.1 Installation und Starten von SPP

Zur Installation ist die Datei "SharedPowerPoint.msi" auszuführen. Diese befindet sich in dem zu entpackenden Zip-Archiv, welches beispielsweise auf der NUSS-Projekt-Homepage unter http://www.informatik.uni-stuttgart.de/ipvr/vs/en/projects/NUSS/ zur Verfügung steht. Danach befindet sich ein entsprechender Eintrag im Startmenü, welcher zum Starten ausgewählt werden kann.

A.2 Systemvoraussetzungen

- PC mit Betriebssystemen
  - Windows 98 (SE)
  - Windows XP
  - Windows NT
  - Windows 2000

Weitere Microsoft-Betriebssysteme sind möglich, wurden jedoch nicht getestet. Wird SPP in heterogenen Umgebungen verwendet, so ist bei der Erstellung der Folien darauf zu achten, dass sich z.B. Animationen aufgrund verschiedener Office-Versionen **nicht** unterschiedlich verhalten.

- Die empfohlene Mindestgeschwindigkeit der PC's beträgt 500 MHz.
- Microsoft PowerPoint 2000 oder PowerPoint XP (2002) jeweils auf jedem Rechner installiert
- Mindestens Java 2 1.4 auf jedem Rechner installiert (und der Aufruf mittels "java" bzw. "javaw" muss ohne weitere Pfadangabe erfolgen können).
- Bildschirmauflösung mindestens 800*600, empfohlen wird 1024*768 oder höher.
- Mindestens 50 MB freier Speicherplatz

Weitere Softwarekomponenten (wie Visual Basic Runtimes, DLL's für das Subclassing von PowerPoint sowie dem inria.net.lrmp-Paket für Reliable Multicast) werden durch Setup-Pakete zur Verfügung gestellt.

Folgende Softwarekomponenten werden von SPP genutzt, liegen dem Installationspaket jedoch **nicht** bei:

- Wenn mit dem NUSS-Framework kommuniziert werden soll, so muss auf einem erreichbaren Rechner ein NAP gestartet werden. Innerhalb dieser NAP-Installation muss die Rechtedefinitionsdatei von SPP installiert sein.
- Für die integrierte DONUT-Unterstützung muss ein ITS entsprechend dessen Dokumentation installiert und gestartet sein.

A.3 Erstellen von Powerpointfolien

Die PowerPoint-Folien können wie gewohnt erstellt und verwendet werden. Hierbei sind zwar alle Features von PowerPoint (Animationen, ...) möglich, jedoch dürfen keine Verbindungen zu externen Dateien bestehen!

Zusätzlich können in den Folienkommentaren - diese finden sich in PowerPoint in der Textbox unterhalb der Folienansicht - folgende SPP-Befehle eingegeben werden:

- "♯"
  Steht in einer einzelnen Zeile das Nummernsymbol, so werden während einer SPP-Session alle Kommentare, die sich dahinter befinden, nicht mit angezeigt. Dies eignet sich z.B. für die Angabe von weiteren Befehlen.

- "DONUT x1 y1 x2 y2 Kommentar"
  Dieser Befehl gibt ein DONUT-Keyword (vgl. Diplomarbeit M. Nagy "Zwischenfragen in Augmented Lectures") an, welches für den Bereich in dem virtuellen Rechteck mit den Ecken oben-links (x1;y1) und unten-rechts (x2;y2) gilt. Die Koordinaten geben ganzzahlige Prozentwerte ("0" - "100") von dem Folienanfang oben links an. Für weitere Infos ist die DONUT-Dokumentation, welche sich bei dessen Installationsdateien befindet, zu verwenden.

- "TIME n"
  Mittels des TIME-Befehls lassen sich Präsentationen vorher zeitlich planen. "n" (eine positive ganze Zahl) gibt hierbei an, für wie viele Sekunden das Anzeigen der Folie vorgesehen ist. In dem Uhrzeit-Fenster lässt sich danach jederzeit feststellen, wie gut man sich im Plan befindet.

A.4 Einstellungen bei Programmstart

A.4.1 Moduswahl

Zu Beginn muss der Benutzer auswählen, in welchem Programmmodus er starten will. Die Vor- und Nachteile der jeweiligen Modi sind im Folgenden beschrieben.

A.4.2 NAP-Login

Im NAP-Login geschieht die nötige Authentifizierung beim NUSS Access Point. Hierbei sind die IP-Adresse des NAP anzugeben, der eigene Benutzername sowie das zugehörige Passwort. Nach einer korrekten Anmeldung stehen die Bereiche NAP-Server und NAP-Client zur Verfügung.

A.4.3 NAP-Server

Der NAP-Server dient dem Erstellen einer neuen NAP-Gruppe. Diese stellt genau eine SPP-Präsentation und eine Powerpoint-Datei dar. Zusätzlich wird eine Beschreibung angegeben, damit Teilnehmer die richtige Gruppe finden können. Um eine SPP-Gruppe zu erstellen, muss zusätzlich das Netzwerkinterface, auf dem der Server agieren soll, ausgewählt werden (bzw. im RUS-VPN die VPN-IP eingetragen werden).

A.4.4 NAP-Client

Im NAP-Client befindet sich eine Liste aller SPP-Gruppen. Durch das Verbinden (Join) mit einer Gruppe startet sich die SPP-Präsentation.

A.4.5 Server

Soll kein NAP verwendet werden (aber eine Java-Umgebung steht bereit), so lässt sich eine Gruppe auch ohne die durch das NUSS-Framework bereit gestellten Möglichkeiten erzeugen. Dadurch stehen folgende Features **nicht** zur Verfügung:

- Gruppenfindung
- Benutzerauthentifizierung
- Rechte
- DONUT-Anbindung

A.4.6 Client

Der Client bildet das Gegenstück zum oben genannten Server. In ihm muss die IP-Adresse des Servers angegeben werden.

A.4.7 Broadcast

Der Broadcast-Modus benötigt weder eine Java-VM, einen laufenden Java-Teil noch eine Infrastruktur wie den NAP. Dadurch stehen folgende Features **nicht** zur Verfügung:

- Gruppenfindung
- Verteilung von Präsentationen
- Benutzermanagement (Namen und Authentifizierung)
- Rechte
- DONUT-Anbindung

- Internet-Fähigkeit (Funktion nur im aktuellen Subnetz; nicht über Router hinweg)

Da der Broadcast-Modus nicht nach dem Client-Server Modell arbeitet, bricht die Präsentation nicht beim Beenden eines Knotens ab (es gibt keinen SinglePointOfFailure). Einzugeben sind hierbei außer einer gültigen Portnummer eine Broadcast-Adresse sowie der Name der gewünschten Präsentationsdatei (von der Replikate auf allen Rechnern bereits vorhanden sein müssen und nicht auf Konsistenz untereinander geprüft werden). Der größte Bereich wird mit der Einstellung "255.255.255.255" in der Broadcast-Adresse abgedeckt.

A.4.8 Erweiterte Einstellungen

Neben den im Programm verfügbaren Möglichkeiten lassen sich weitere Einstellungen mit dem Programm "Experteneinstellungen", welches sich ebenfalls im Startmenü befindet, abändern.

Derzeit kann darin nur die Multcastübertragung konfiguriert werden. Bei der Eingabe einer IP-Adresse ist darauf zu achten, dass nur korrekte Multicast-IPs (Class D Netz) eingetragen werden. Da mehrere Präsentationen über die gleiche Gruppen-Adresse verteilt werden können, muss diese Einstellung jedoch nicht notwendigerweise verändert werden.

A.5 Steuerung während der Präsentation

A.5.1 Tastatur

Wenn das Programm nicht gerade eine Texteingabe erwartet, gelten folgende Tasten:

Falls diese Tasten nicht funktionieren, so muss entweder

Leertaste	Nächste Animation/Folie
Bild-Ab	Nächste Animation/Folie
Rücktaste	Vorherige Animation/Folie
Bild-Auf	Vorherige Animation/Folie
Q	Beenden der Präsentation/der Anwendung
Return	Nächste Animation/Folie
Zahlen+Return	Zu einer bestimmten Folie springen

Tabelle A.1: Tastenkürzel

- ein SPP-Fenster wie die Befehlsleiste oder die Uhr aktiviert werden.
- mittels Alt-Tab zu der SPP-Präsentation gesprungen werden.

Tasteneingaben sind nicht möglich, wenn das Notizenfenster fokussiert ist.

A.5.2 SPP-Befehlsleiste

Die Befehlsleiste visualisiert die am meisten benötigten Funktionen und ist für das Gruppenmanagement nötig. Die einzelnen Knöpfe haben folgende Bedeutung:
Aktionen:
Hierbei wird festgelegt, welche Aktion bei einem Links-Klick auf der Präsentationsfolie durchgeführt werden soll.

- Keine Aktion (Kreuz)
  Linke Mausklicks werden ignoriert.
- Klicks an PowerPoint weiterleiten (Weltkugel)
  Linke Mausklicks werden wie von PowerPoint gewohnt behandelt. Somit können zum Beispiel Hyperlinks geöffnet oder Filme gestartet werden. Achtung: Wenn Sie in diesem Modus mit der linken Maustaste blättern, so wird dies von SPP nicht erkannt!

- Persistenter Stift (Bleistift)
  Bei gehaltener Maustaste lassen sich Linien einzeichnen.
  Achtung: Dies geschieht (je nach Rechnerleistung) teilweise sehr langsam und zieht nach!

- Temporärer Stift (Stift mit Kringel)
  Der Temporäre Stift funktioniert wie der permanente, ist aber mit dem Präsentationsstift aus PowerPoint verwandt: Die Linien verschwinden wieder, sobald geblättert oder ein persistentes Objekt erstellt wird.

- Persistenter Text ("A")
  Bei einem Mausklick in die Präsentation erscheint ein Texteingabefenster (siehe unten), über das ein Text eingefügt werden kann.

- Picken (Pfeil)
  Durch das Picken lassen sich (persistente) Annotationen markieren, um Informationen anzuzeigen oder sie später zu verschieben bzw. zu löschen (siehe Fenster "Picking Informationen"). Es wird immer das oberste Objekt gepickt.

- Objekt Picken (Zauberstab)
  Dieser Modus ermöglicht neben dem Picken von Annotationen das Selektieren von beliebigen Objekten (wie z.B. Bildern).

- Pointer setzen (Zeigende Hand)
  Hiermit lassen sich Punkte kennzeichnen, indem ein Symbol an der angeklickten Stelle eingefügt wird. Dieses Symbol verschwindet bei einem weiteren Setzen des Pointers und wird auch nicht gespeichert.

- DONUT-Frage (D?) und DONUT-Anmerkung (D!)
  Wird ein DONUT-Dienst gefunden, so sind diese 2 Möglichkeiten zusätzlich verfügbar (außer im Broadcast-Modus). Beim Klicken auf einen Punkt öffnet sich ein DONUT-Fenster, in dem zu (in der Präsentationsdatei) angegebenen Keywords Fragen/Anmerkungen an den DONUT-Server geschickt werden (siehe DONUT-Dokumentation).

Als nächstes stehen 6 vordefinierte Farben zur Auswahl. Diese finden Verwendung beim Linienziehen und beim Setzen des Pointers (die Texteingabe hat eine separate Auswahl).

Der "¡"-Button (bzw. der "¿"-Button) dient der Verkleinerung des Fensters, wenn Gruppenbefehle nicht benötigt werden. Im Gruppenmanagement gibt es folgende Optionen:

- Gruppenauswahl (Dropdownliste) Im einfachsten Fall lässt sich hier auswählen, ob Annotationen nur auf dem eigenen Rechner ("Lokale Annotationen") oder bei allen Teilnehmern ("Globale Annotationen") sichtbar sind. Werden Untergruppen verwendet, so lassen sich hier auch diese verwenden. Annotationen können hierbei nur in den Gruppen erfolgen, in denen man auch Teilnehmer ist:
- Gruppe Beitreten/Verlassen ("+ -")
  Der Gruppenstatus ist an einem "+"-Zeichen vor dem Gruppennamen zu erkennen. Um diesen Status zu ändern, muss nur auf diesen Button geklickt werden.
- Refresh ("R")
  Erneuert die Gruppenliste.
- Neue Untergruppe erstellen ("*")
  Erstellt eine neue Untergruppe mit dem Namen, welcher in dem daraufhin erscheinenden Fenster eingegeben wird.
- Untergruppe löschen ("X")
  Hiermit können Untergruppen (insofern man dieser beigetreten ist und ebenfalls das nötige Recht besitzt) gelöscht werden.

A.5.3 PowerPoint-Oberfläche

Innerhalb der PowerPoint-Oberfläche haben die Maustasten folgende Bedeutung:

Rechte Maustaste: Öffnet ein Kontextmenü, in dem die meisten Einstellungen vorgenommen werden können (siehe Kontextmenü).

Linke Maustaste: Abhängig von der gerade ausgewählten Aktion (in der Befehlsleiste bzw. im Kontextmenü.

A.5.4 Kontextmenü

Im Kontextmenü stehen folgende Optionen zur Verfügung:

- Abbruch
  Schließt das Fenster wieder (Achtung: verwenden Sie bitte diesen Eintrag, da es sonst Probleme mit Tastatureingaben geben kann)
- Befehlsleiste einblenden
  Öffnet das Fenster "SPP-Befehlsleiste"
- Notizen einblenden
  Öffnet das Fenster "Notizen/Kommentare"
- Uhr einblenden
  Öffnet das Fenster "Uhrzeit/Timing"
- DONUT-Einträge betrachten
  Meldet DONUT, dass es ein Fenster mit allen Einträgen öffnen soll.
- Annotationsverbreitung
  Wählt die Gruppe aus, in welche die Annotationen gesendet werden.
- Aktionen
  Wählt die zu verwendende Aktion aus (siehe SPP-Befehlsleiste)
- Stiftdicke
  Wählt die Dicke der permanenten Linien aus

- Stiftfarbe
  Wählt die Farbe von Linien und Pointern aus
- Löschen, Präsentationsstift
  Löscht alle temporären Zeichnungen
- Löschen, Annotationen auf dieser Folie
  Löscht alle Annotationen der Folie.
- Beenden
  Beendet das Programm

A.5.5 Notizen/Kommentare

Im Notizfeld können einerseits die Kommentare, welche bei der Erstellung in PowerPoint vermerkt wurden, eingesehen werden. Zusätzlich können diese Kommentare geändert werden. Dadurch lassen sich hier eigene Notizen und Folienmitschriebe realisieren.

Achtung: Während dieses Fenster angewählt ist (Titelleiste dunkelblau), werden keine Tastenbefehle angenommen!

A.5.6 Text-Eingabe

In diesem Fenster lassen sich alle nötigen Informationen für eine Textannotation eingeben. Wird "OK" gedrückt, so wird die Texteingabe in die selektierte Gruppe gesendet.

A.5.7 Picking-Informationen

Dieses Fenster erscheint, wenn ein Objekt markiert wurde. Es zeigt Informationen über Art und Ersteller von Objekten an. Nach einem Klick auf "verschieben" wird das

Objekt bei einem Klick in die Präsentation auf der Folie verschoben. Durch den Button "Löschen" lässt es sich entfernen.

A.5.8 Uhrzeit/Timing

Das Uhrzeit-Fenster gibt Informationen über den Zeitverlauf wieder. Es beinhaltet:

- eine normale Uhr
- Die aktuelle Foliennummer
- Eine Stoppuhr (mit Reset-Knopf)
- Eine Anzeige, wie lange die aktuelle Folie angezeigt wird
- Eine Anzeige, wie man während der Präsentation in der vorgeplanten Zeit liegt (siehe "Erstellen von Powerpointfolien" für das Setzen der Vorgaben)

A.6 Beenden des Programmes

Wird das Programm beendet (über Kontextmenü:Beenden oder mittels der Taste "Q"), so wird gefragt, ob die Präsentation gespeichert werden soll. Wenn dies bejaht wird, so lässt sich ein Dateiname angeben. Im entsprechenden Verzeichnis befinden sich danach 2 Präsentationen, nämlich einmal die Originaldatei und einmal die Präsentation mit allen Annotationen und Kommentaren.

Achtung: Sollte während des Beendens gefragt werden, ob Überarbeitungen gekennzeichnet werden sollen, so darf dies nicht mit "Abbrechen" beantwortet werden!

A.7 FAQ

Alles funktioniert, nur das Blättern klappt nicht.
Überprüfen Sie, ob "Globales Blättern" im Kontextmenü eingeschaltet ist.

Der Server kann im Client-Betrieb nicht gefunden werden.
Überprüfen Sie mittels des Programms "winipcfg", ob Sie die richtige IP-Adresse angegeben haben. Sollten Sie sich innerhalb eines VPN's befinden, so müssen Sie dessen IP-Adresse verwenden.

Der Server kann im NUSS-Betrieb nicht gefunden werden.
Vermutlich wurde bei der Erstellung des Servers die falsche IP-Adresse angegeben. Zur Lösung siehe vorherige Frage.

Die Multicast-Verteilung funktioniert nicht.
Dies kann mehrere Gründe haben:

- Ist Ihr Netzwerk Multicast-fähig?
- Beherrscht Ihr Computer Multicastnachrichten (Dienste wie Personal Firewalls, SW-Router oder mehrere Netzwerkkarten könnten dies verhindern)
- Haben Sie in den Einstellungen Multicast abgeschaltet?

A.8 Feedback und Kontakt

Sollten während des Betriebs Fehler auftreten, welche hier nicht erwähnt wurden, so wäre ich über eine Mail an Holger.Cermann@gmx.de dankbar. Ebenso sind Probleme, Wünsche und Anregungen immer willkommen.

Anhang B

Zeitplan

Im Folgenden ist der Zeitplan angeführt, nach dem diese Studienarbeit ablief. Die einzelnen genannten Phasen überlappen sich teilweise oder laufen nach der offiziellen Beendigung weiter.

B.0.1 Phase 1: Grundlagen

Beginn:

01.10.2002

Dauer:

3 Wochen

Inhalt:

- Analyse Fernsteuerungsarten
- Ermitteln der Anforderungen

B.0.2 Phase 2: Analyse von Microsoft PowerPoint

Beginn:

19.10.2002

Dauer:

2 Wochen

Inhalt:

- Analyse der Architektur und des Objektmodells von PowerPoint
- Suche nach Möglichkeiten zum Abfangen von Nachrichten
- Erstellung eines Prototypen

B.0.3 Phase 3: Anforderungen an eine Middelware

Beginn:

29.10.2002

Dauer:

2 Wochen

Inhalt:

- Sammeln der benötigten Rechte
- Anforderungen an eine API
- Bestimmung des Ablaufes einer Sitzung
- Mitentwicklung der Schnittstelle zum NAP

B.0.4 Phase 4: Programmentwicklung

Beginn:

15.11.2002

Dauer:

6 Wochen

Inhalt:

- Grob- und Feinentwurf
- Implementierung des Programmes
- Interne Tests

B.0.5 Phase 5: Test und Weiterentwicklung

Beginn:

08.01.2003

Dauer:

6 Wochen

Inhalt:

- Einbindung des NAP's
- Verwendung im Rahmen von Vorlesungen
- Tests in verschiedenen Szenarien
- Einsammeln von Feedback

B.0.6 Phase 6: Dokumentation

Beginn:

31.01.2003

Dauer:

8 Wochen

Inhalt:

- Vervollständigen der Codedokumentation
- Schreiben der Studienarbeit

Anhang C

Rechtliches

C.1 Markenzeichen

Microsoft, PowerPoint, Netmeeting, Conference XP, Windows 98, Windows 98 SE, Windows XP, Windows NT und Windows 2000 sind eingetragene Warenzeichen der Microsoft Coporation.

In dieser Studienarbeit wurden weitere eingetragene Warenzeichen, Handelsnamen und Gebrauchsnamen verwendet. Auch wenn diese nicht als solche gekennzeichnet sind, gelten die entsprechenden Schutzbestimmungen.

C.2 Lizenz des Light-weight Reliable Multicast Protocol (LRMP)

COPYRIGHT 1997/1998 BY: MASSACHUSETTS INSTITUTE OF TECHNOLOGY (MIT), INRIA

This W3C software is being provided by the copyright holders under the following license. By obtaining, using and/or copying this software, you agree that you have

read, understood, and will comply with the following terms and conditions:

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee or royalty is hereby granted, provided that the full text of this NOTICE appears on ALL copies of the software and documentation or portions thereof, including modifications, that you make.

THIS SOFTWARE IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, COPYRIGHT HOLDERS MAKE NO REPRESENTATIONS OR WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE OR DOCUMENTATION WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS. COPYRIGHT HOLDERS WILL BEAR NO LIABILITY FOR ANY USE OF THIS SOFTWARE OR DOCUMENTATION.

The name and trademarks of copyright holders may NOT be used in advertising or publicity pertaining to the software without specific, written prior permission. Title to copyright in this software and any associated documentation will at all times remain with copyright holders.

Erklärung

Hiermit versichere ich, diese Arbeit
selbständig verfaßt und nur die
angegebenen Quellen benutzt zu haben.

(Holger Cermann)