

Zusammenfassung Verteilte Anwendungen

von Carsten Buschmann (kontakt@carsten-buschmann.de)
 nach der Vorlesung „Verteilte Anwendungen“
 von Prof. M. Zitterbart
 an der Technischen Universität Braunschweig
 SS 2000

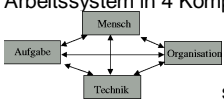
Download unter www.carsten-buschmann.de/skripte/

Bei Fehlern, Anmerkungen, Fragen oder Kritik bitte ich um Mail unter kontakt@carsten-buschmann.de. Weiterhin bin ich unter www.carsten-buschmann.de erreichbar. Die letzte Spalte der Tabelle stellt die (ungefähre, weil jedes Jahr etwas andere) Seitenzahl der Vorlesungsunterlagen dar.

1. Einführung

1.2	Begriffsbildung	<ul style="list-style-type: none"> - Gründe für Verteilung: Parallelität, Fehlertoleranz, inhärente Verteiltheit - Aspekte verteilter Anwendungen: <ul style="list-style-type: none"> - Gliederung in eine Menge kooperierender Teilkomponenten mit innerem Zustand und Operationen - Teilkomponenten können auf versch. Rechnern sein, tauschen Daten aus - Schnittstellen definieren Kommunikation und die verfügbaren Operationen mit Parametern, Ergebnissen, sichtbaren Auswirkungen und Auswirkungen auf nachfolgende Operationen - Bedarf nach übergreifenden Kontrollstrukturen, unterschiedliche Grade der Autonomie möglich 	8
	Entwurf verteilter Anwendungen	Fragestellungen: Funktionen lokal oder entfernt? Lokalisierung von Funktionen? Aktionen bei Kommunikationsunterbrechung? geforderte Sicherheit? erforderliche Homogenität? Test und Debugging schwierig	9
	CSCW	<ul style="list-style-type: none"> - Computer Supported Cooperative Work: liefert die theoretischen Grundlagen/Methoden für rechnergestützte Gruppenarbeit - darf nicht nur technische Aspekte berücksichtigen, sondern auch „menschliche“ 	10
	Groupware	<ul style="list-style-type: none"> - Implementierungen konkreter Werkzeuge - Computergestützte Systeme (verteilte Anwendungen), die Gruppen von Leuten bei der Lösung gemeinsamer Aufgaben in einer gemeinsamen, verteilten Umgebung unterstützen - Gruppenmitglieder interagieren (u.U. in unterschiedlicher Art und Weise) miteinander - Gruppenmitglieder müssen über ihre gegenseitige Existenz informiert sein (Group-awareness) - das Design von Groupware ist ausschlaggebend für die Akzeptanz der Nutzer, nur so ist wichtige Rolle in der Informationsgesellschaft möglich 	11
	Interaktionsformen	<ul style="list-style-type: none"> - explizite Interaktion: bewußter Informationsaustausch im Sinne einer Konversation - implizite Interaktion: z.B. Arbeit an einem gemeinsamen Dokument - Synchrone Interaktion: vom Benutzer initiierte Interaktion wird in Echtzeit sichtbar - Asynchrone Interaktion: Benutzer sind nicht gleichzeitig an der Interaktion beteiligt 	13
	Interdisziplinarität	Wichtige Disziplinen: Informatik, Arbeitspsychologie, Pädagogik, Gestaltung, Sozialwissenschaften	15
1.3	Klassen von CSCW-Anwendungen	<ul style="list-style-type: none"> - Kommunikationssysteme und Informationsräume: (a-)synchrone CSCW-Systeme wie Mail oder Videokonferenzen, Informationsräume - Workflow-Management, Konversations- und Koordinierungssysteme: Überwachen/ Steuern der Ausführung von Gruppenaktivitäten unter Einhaltung zeitlicher/kausaler Abhängigkeiten - Kooperationssysteme: Kooperationsunterstützung (Gruppeneditoren,...) mit Echtzeitanforderung - Agentensysteme: Kooperation mittels Agenten 	17

2. Grundlagen CSCW

2.1	Begriffsanalyse	Vorwärtsanalyse: meist ausschließlich unter Gesichtspunkten der Informatik → Rückwärtsanalyse	22
	Rückwärtsanalyse	<ul style="list-style-type: none"> - zu bewältigende Arbeit im Mittelpunkt: bei der Leavitt-Raute gliedert sich das Arbeitssystem in 4 Komponenten  <p style="text-align: center;">sowie Wirkungszusammenhänge: Gestaltung Benutzerschnittstelle, Regelung des Zugriffes auf Informationen, Granularität und Speicherung der Informationen, Parallelität der Aufgabenausführung</p> <ul style="list-style-type: none"> - kooperative Arbeit: mehrere Personen an einem Arbeitsprozeß beteiligt. Kommunikation stellt einen wichtigen Aspekt dar. Aufgaben des CSCW: Sicherung der Konsistenz, Versionsverwaltung, Sicherstellen von Zielgerichtetheit, Umsetzung sozialer Protokolle - Kooperation soll direkt durch das System unterstützt werden: Unterstützungsformen: <ul style="list-style-type: none"> - prozeßorientiert (z.B. schlichtend, ordnend/strukturierend, motivierend, kontrollierend) wie z.B. in Konferenzsystemen → Protokoll-/Ablaufnutzen - inhaltsorientiert (wie z.B. Datenbanken, Wissenbasen) → Informationsnutzen - Rollen des Rechners: Eindringling (Rechner übernimmt bekannte Unterstützungsformen wie Whiteboard), Auslöser (neue Unterstützungsformen/Kommunikationsbeziehungen möglich, z.B. Email), Dienstleister (neue Dienste möglich, z.B. weltweites Recherchieren) 	23
	Kommunikation	Kategorisierung von Zusammenarbeiten nach Kommunikationsintensität	24

		Informieren	Koordinieren	Kollaborieren	Kooperieren																	
		<p style="text-align: center;">← niedrig Grad der Kommunikation in der Gruppe hoch →</p> <p>höherer Grad ermöglicht bessere Rechnerunterstützung</p> <ul style="list-style-type: none"> - Informieren: geringer Komm.aufwand, Kaum Kontakt zw. Sender und Empfänger, Bsp: Buch - Koordinieren: direkt Kontakt, Informationen/Aktionen werden abgesprochen - Kollaborieren: Sender und Empf. haben ein gemeinsames Ziel, unregelmäßige Interaktion - Kooperieren: gemeinsames Ziel, regelmäßige Interaktion, Gruppenziele dominieren persönliche 																				
	Ks in Groupware	<ul style="list-style-type: none"> - Kommunikation: Informationsaustausch - Koordination: Abstimmen/in Einklang bringen versch. Vorgänge - Kollaboration: Zusammenarbeit in unregelmäßigen Abständen - Kooperation: Zusammenarbeit in regelmäßigen Abständen 				26																
2.2	Klassifikation von CSCW-Systemen	<p>Funktionelle Gliederung (z.B. nach Ellis)</p> <ul style="list-style-type: none"> - Nachrichtensysteme: asynchroner Austausch von Nachrichten - Gruppeneditoren - el. Sitzungsräume (realer Raum) - Rechnerkonferenzsysteme (virtueller Raum) - gemeinsame Informationsräume (Verwaltung von Gruppendokumenten) - Agentensysteme - Workflow-Management/ Koordinationssysteme <p>Problem: Kategorien nicht exklusiv</p>		<p>nach Raum und Zeit</p> <table border="1"> <thead> <tr> <th>Raum / Zeit</th> <th>Gleiche Zeit (synchron)</th> <th>Verschiedene Zeit (asynchron) vorhersehbar</th> <th>Verschiedene Zeit (asynchron) nicht vorhersehbar</th> </tr> </thead> <tbody> <tr> <td>Gleicher Ort</td> <td>Face-to-face Sitzung</td> <td>Schichtarbeit</td> <td>„schwarzes Brett“</td> </tr> <tr> <td>Verschiedener Ort (vorhersehbar)</td> <td>Video-konferenz</td> <td>E-mail</td> <td>Kollaborativ Verfassen von Dokumenten</td> </tr> <tr> <td>Verschiedener Ort (nicht vorhersehbar)</td> <td>Video-konferenz mit mobilen Mitgliedern</td> <td>Nicht-Realzeit-rechnerkonferenz</td> <td>Vorgangsbearbeitung</td> </tr> </tbody> </table> <p style="text-align: center;">Grenze Schwarmarbeit</p>		Raum / Zeit	Gleiche Zeit (synchron)	Verschiedene Zeit (asynchron) vorhersehbar	Verschiedene Zeit (asynchron) nicht vorhersehbar	Gleicher Ort	Face-to-face Sitzung	Schichtarbeit	„schwarzes Brett“	Verschiedener Ort (vorhersehbar)	Video-konferenz	E-mail	Kollaborativ Verfassen von Dokumenten	Verschiedener Ort (nicht vorhersehbar)	Video-konferenz mit mobilen Mitgliedern	Nicht-Realzeit-rechnerkonferenz	Vorgangsbearbeitung	30
Raum / Zeit	Gleiche Zeit (synchron)	Verschiedene Zeit (asynchron) vorhersehbar	Verschiedene Zeit (asynchron) nicht vorhersehbar																			
Gleicher Ort	Face-to-face Sitzung	Schichtarbeit	„schwarzes Brett“																			
Verschiedener Ort (vorhersehbar)	Video-konferenz	E-mail	Kollaborativ Verfassen von Dokumenten																			
Verschiedener Ort (nicht vorhersehbar)	Video-konferenz mit mobilen Mitgliedern	Nicht-Realzeit-rechnerkonferenz	Vorgangsbearbeitung																			
	3K-Modell	<p>nach Teufel: Einteilung nach im Vordergrund stehenden Aspekten Kommunikation, Kooperation, Koordination</p>				32																
2.3	Definition Gruppenprozeß	<p>Spezifikation von Informationen, Aktivitäten und Eigenschaften einer elektronisch unterstützten Gruppe bei der Festlegung des Rahmen, in dem Gruppenarbeit stattfindet. Hat Anfangs- und Endpunkt (Ergebnis)</p>				34																
	nach Drexler/Sibbet	<p>Entstehungsphasen</p> <ul style="list-style-type: none"> - Orientierung: Zweck/Aufgabe der Gruppe - Vertrauensbildung: Integration von Personen in Gruppenprozeß - Klärung der Ziele/Aufgaben: Ausräumen von Konflikte - Verpflichtung: Vereinbarung der weiteren Vorgehensweise <p>Festigungsphasen</p> <ul style="list-style-type: none"> - Implementierung: Zuteilung der Aufgaben an Personen - Hohe Leistung: Gruppe kann intuitiv, flexibel reagieren - Erneuerung: Engagement der Teilnehmer sinkt 				35																
	Zentrales Gruppenprozeß-Modul	<p>Informationen des Gruppenprozesse werden an zentraler Stelle gespeichert</p> <ul style="list-style-type: none"> - Gruppenprozesse durch Anwendungssoftware voneinander getrennt, Speichern von Informationen im privaten Bereich nicht notwendig - historischer Ablauf kann leicht erfaßt werden (gut, wenn später Teilnehmer hinzukommen) 				37																
	Verteiltes nichtrepliziertes Gruppenprozeß-Modul	<p>Gruppenprozess (und damit auch Gruppendokumente) ist verteilt und nicht repliziert</p> <ul style="list-style-type: none"> - von Gruppendokumenten ex. nur ein Original auf einem Heimatrechner - Zugriff auf entfernte Dokumente transparent - Zugriffs-/ortstransparent durch lokale Kopie der Gruppenprozeßsoftware 				38																
	Verteiltes repliziertes Gruppenprozeß-Modul	<p>Gruppenprozesse sind verteilt und repliziert</p> <ul style="list-style-type: none"> - Informationseinheiten pro Rechner repliziert → besser Fehlertoleranz, kürzere Antwortzeiten, Konsistenzhaltung nötig - Verteilung transparent, Zugriffs-, Orts-, Replikationstranparenz, für Gruppenteilnehmer gleiche Arbeitsweise wie beim zentralen Modell 				39																
2.4	gemeinsamer Kontext	<ul style="list-style-type: none"> - Menge von Objekten. Aktionen auf Objekte und Objekte für alle sichtbar - Darstellung: gemeinsame Sicht muß unterstützt werden, Nutzer soll Kenntnis von der Arbeit der anderen haben - zentraler Aspekt: gruppenbewußte Nutzerschnittstelle, die gemeinsame Objekte und Gruppendynamik sichtbar macht 				40																
	WYSIWIS	<p>What you see is what I see: konsistente Darstellung für alle Teilnehmer</p> <ul style="list-style-type: none"> - striktes WYSIWIS: absolut gleicher Kontext 				41																

		- Möglichkeiten zur Abschwächung von WYSIWIS: Bereitstellung privater Arbeitsfläche (Cursor), individuelle Bildschirmgestaltung, unterschiedliche Sichtweisen, zeitliche Divergenzen	
	Tele-Pointer	- erscheint auf gemeinsamen Bildschirmbereich, kann von versch. Teilnehmern bewegt werden, wird wie gemeinsames Betriebsmittel verwaltet - Probleme: mehrere Tele-Pointer unübersichtlich, Tele-Pointer müssen mit Teilnehmern identifiziert werden, bei gemeinsamen Tele-Pointer Nebenläufigkeitskontrolle nötig, Implementierung komplex („logische Position“)	43
2.5	Gruppenbewußtsein	- Verständnis der Aktionen anderer, die einen Kontext für eigene Aktivitäten liefern - Vorteile: Initiierung spontaner, informeller Kommunikation, bessere Informiertheit über den Stand der Gruppenaktivität - Arten: informelles Gruppenbewußtsein: generelles Wissen über die Gruppe (wer ist wo). Bewußtsein bzgl. Gruppenstruktur (Rollen, Verantwortlichkeiten), soziales G. (emotionaler Zustand der Mitglieder), G in gemeinsamen Aufgabenbereichen (Zugriffe/Änderungen an Dokumenten)	45

3. Nebenläufigkeitskontrolle

3.1	Einführung	- Problem: Konkurrierende Zugriffe auf Informationen können zu Inkonsistenzen führen - Situation: unterschiedliche Anforderungen bzgl. Konsistenz (2 Schreiber können an einem Dokument arbeiten, beides wird sichtbar → geringe Anforderung. Schreibender Zugriff auf Datenbank: hohe Anforderung), hohe Anforderung an Antwortzeit → Replikation beschleunigt lesen, verlangsamt schreiben. Netzpartition durch Ausfälle problematisch → höchsten eine Netzkomponente darf schreibend zugreifen	49
	Klassifikation	Gliederung in optimistische (sichern Konsistenz nicht in jedem Fall) und pessimistische Verfahren, die dann in dezentrale(mit oder ohne Votierung)/zentrale Kontrolle gegliedert werden	50
3.2	Zentrale Kontrolle	- Verfahren mit zentraler Kontrolleinheit: Zentrale serialisiert und synchronisiert Schreibzugriffe. Vorteil: keine komplexen Verfahren notwendig. Nachteil: Single-Point-of-Failure, bei Ausfall Ersetzungsstrategien komplex - Token-Verfahren (Polling): Tokeninhaber hat gleiche Rechte wie Zentrale, Rechner in virtuellem Ring organisiert. Vorteil: Serialisierung und Synchronisation einfach. Nachteil: komplexe Verwaltung der Ringstruktur	51
3.3.1	Dezentrale Kontrolle	- einfache Sperrverfahren, Floorpassing, Transformationssysteme, Transaktionssysteme	52
	einfache Sperrmechanismen	- Ressourcen werden mit Sperren versehen, sind sie gesetzt, kann kein anderer Teilnehmer zugreifen. Evtl. ergeben sich Wartezeiten. Sperrgranularität maßgeblich für Grad der Nebenläufigkeit, „Sperrschwelle“ definiert werden. - einfache Sperrverfahren ohne Replikation: Anforderung/Freigabe der Sperre bei Zentraler Einheit → kann leicht zu geringer Verfügbarkeit führen - einfache Sperrverfahren mit Caching: Benutzer arbeiten in lokalen Caches → höhere Nebenläufigkeit, ortstransparentes System - zentralisiertes Sperrverfahren mit vollst. Replikation: Vergabe der Sperren durch zentrale Stelle, Problem: Ausfall der Zentrale	53
3.3.2	Floorpassing	- Kontrolle über Ressource wechselt, keine Sperrmechanismen erforderlich, Floor meist durch Token repräsentiert, Floor repräsentiert oft Datenstrom (Verfahren häufig für Konkurrenz um Senderecht eingesetzt), Soziale Protokolle müssen beachtet werden - explizites Floorpassing: Kontrolle wird vom Benutzer aktiv weitergegeben, er kennt und kontrolliert den nächsten Benutzer → jeder muß alle Teilnehmer kennen, System muß Neuankünfte erkennen - implizites Floorpassing mit Koordinationsstelle: Übergabe der Kontrolle durch zentrale Koordinationsstelle → Fairneß (u.U. durch Zeitbegrenzung), Inaktivität kann berücksichtigt werden, kann mit Kontrolle auf Datenzugriffe kombiniert werden, kann nicht auf individuelle Bedürfnisse eingehen, Single-Point-of-Failure - implizites Floorpassing mit dezentraler Koordination: Anfrage für Kontrolle des Floors wird an alle in Frage kommenden Rechner gesendet (Broad-/Multicast), Übergabe durch Floor-Besitzer. Vorteile: Überprüfung auf Floorfreigabe wird vom System durchgeführt → Fairneß	54
3.3.3	Transaktions-Verfahren	- Transaktionen sichern Serialisierbarkeit zu - Einsatz bei asynchroner Gruppenarbeit, Kommunikationsfehler müssen vom unterliegenden transparent verarbeitet werden - Transaktionsprimitive: Begin_Transaction, Abort_Transaction, End_Transaction; Read, Write, ... bilden Rumpf der Transaktion - Unterschied zu Sperren: extrem kurze Transaktionsdauer	58
	Transaktion	- Eigenschaften: atomar (unteilbar), konsistent (verletzt keine Invarianten), isoliert (stören sich nicht gegenseitig), dauerhaft (nach Abschluß einer Transaktion werden Änderungen festgeschrieben) - gut bei kurzen Transaktionen	59
	CES	- Collaborative Editing System: Dokument (verteilt, nicht repliziert gespeichert) besteht aus Strukturbeschreibung (repliziert) und mehreren Texteinheiten (auf Rechner des verantwortlichen Benutzers) aus Identifikator, Versionskeller und Sperrzustand - Lesen: jeder kann eine Texteinheit immer lesen, im Versionskeller sind alle konsistenten Texteinheiten - Schreiben: Dokument wird gesperrt, Änderungen werden lokal angezeigt, die anderen sehen sie nach Abschluß der atomaren Transaktion	60
3.3.4	Transformations-Verfahren	- Zugriff auf kleine Informationseinheiten (z.B. Zeichen), keine strenge Serialisierbarkeit - Einsatz bei synchroner Gruppenarbeit, Änderungen sollen sofort sichtbar werden - $g = (TS, O)$ sei synchrone verteilte Gruppensitzung mit n Teilnehmern, TS: Menge von Teilnehmersystemen, O: parametrisierbare Operatoren.	62

		<ul style="list-style-type: none"> Jedes Teilnehmersystem s ist Tripel $s = (TS\text{-Proze\ss}, TS\text{-Objekt}, TS\text{-Identifikator})$. TS-Objekt: passives Datenobjekt, TS-Identifikator: eindeutiger Bezeichner für s, TS-Prozess stellt 3 Basisoperationen zur Verfügung: Erzeugen, Empfang und Ausführung von Operationsanforderungen o 	
	Definitionen	<ul style="list-style-type: none"> Präzedenzeigenschaft: wird o1 vor o2 ausgeführt, wird o1 auf allen Systemen vor o2 ausgeführt Ruhezustand: alle generierten Operationen auf allen Datenobjekt kopien ausgeführt Konvergenzeigenschaft: im Ruhezustand alle Kopien von Datenobjekte gleich Korrektheit von Transformationsverfahren: Präzedenz- und Konvergenzeigenschaft immer erfüllt 	64
	Bsp: GROVE	<ul style="list-style-type: none"> Mehrbenutzereditor, Granularität: Zeichen, Dokument liegt auf jedem Rechner vor Algorithmus basiert auf Operationstransformation: $o2 \circ o1 = o1 \circ o2$ Definition einer Teilordnung durch Prioritäten abh. von Rechner und Eigenschaften d. Anwendung Operation wird sofort lokal ausgeführt, ein Paar (Operation, Priorität) an alle versandt. Empfänger reihen Operation gemäß Priorität in Warteschlange ein. Wurden derweil konkurrierende Operationen ausgeführt, wird die eingereite ggf. transformiert. 	65

4. Konversations- und Koordinationsmodelle

4.1	Workflow-Management	<ul style="list-style-type: none"> Spezifizierung und Ausführung von zu koordinierenden Aufgaben, die einen Geschäftsprozeß repräsentieren alle Aufgaben, die zu Modellierung, Spezifikation, Simulation, Ausführung, Überwachung und Steuerung der Ausführung von Gruppenaktivitäten unter Einhaltung kausaler und zeitlicher Abhängigkeiten unter Einbeziehung des Ausführungskontextes nötig sind verbessert die gemeinsame Sicht auf eine Aufgabe und das Gruppenbewußtsein 	73																
	Grundbegriffe	<ul style="list-style-type: none"> Geschäftsprozeß: inhaltlich abgeschlossene zeitliche und sachlogische Abfolge von Funktionen, die zu Bearbeitung von betriebswirtschaftlich relevanten Objekten nötig sind Geschäftsprozeß-Reorganisation: komplexe, top-down orientierte Veränderung des Unternehmens zur dramatischen Leistungsverbesserung und Aufteilung der Unternehmensaktivitäten in essentielle, abteilungsübergreifende Prozesse Workflow: endliche Folge von Aktivitäten, wird von einer Folge von Ereignissen ausgelöst und beendet 	74																
	Workflow-Management-System (WFM)	<ul style="list-style-type: none"> System aus mehreren Werkzeugen, einfache Integration in ein Unternehmen wichtig Anforderungen: Unterstützt den Fluß von Dokumenten (Routes), Darstellung der Regeln des Geschäftsbedarfes (Rules), Aufgaben/Aktivitäten werden nicht Personen, sondern Rollen zugeteilt, (Roles), Geschäftsprozesse automatisieren (Processes), Festlegung der Unternehmenspolitik für die Abarbeitung der Prozesse (Policies), im Gegensatz dazu Realität durch Erfahrung und Historie, z.B. „kleiner Dienstweg“ (Practicies) 	75																
4.2	Konversations-Modell	<ul style="list-style-type: none"> Idee: Zwischenmenschliche Kommunikation basiert auf Wortfolgen, wesentliche Aspekte sind Syntax („Grammatik“), Semantik („Bedeutung“) und Pragmatik („Implikationen“) Sprechakttheorie: Analyse der Sprache als Bedeutungsvolle Handlung in Situationen einer gemeinsamen Aktivität, aufbauend auf pragmatischem Aspekt, dabei ist Sprechen eine Art von Handeln Konversation: Folge von Sprechakten, die ihrerseits aus Wortfolgen bestehen 	76																
	Sprechakt	<p>Aufteilung in ein Tripel:</p> <ul style="list-style-type: none"> Satzinhalt: Thema, Kommunikationsgegenstand Kategorie: Behauptung, Anweisung, Verpflichtung, Deklaration (Synergie zw. Satzinhalt und Realität), Äußerung (Beschreibung) Präsentationsgrad: höflich, fordernd, ... 	77																
	Konversations-Netze	<ul style="list-style-type: none"> Ziel: Formalisieren zw.menschlicher Kommunikation, Strukturierung von Unterhaltungen, Isolierung von wiederkehrenden Mustern von Wortfolgen Konversationsnetz: Netz von Sprechakten (Kanten = Sprechakte, Knoten = Konversationszustände) Eigenschaften: in jedem Zustand geringe Zahl möglicher Sprechakte, Menge von Endzuständen, alle Handlungen werden durch Wortfolgen ausgedrückt, es existieren implizite Handlungen, was zu tun ist/Konsequenzen werden nicht spezifiziert insbesondere Bürovorgänge können so oder als Petrinetze (Informationen = Stellen, Vorgänge = Transitionen) modelliert werden 	78																
	Petrinetze	s. Folie 81																	
	Pr/T-Netze	<ul style="list-style-type: none"> Prädikat/Transitionsnetze: beschreibt dynamische Aspekte asynchroner Konversation Akteure (Personen, Programme, Nachrichtenkanäle sowie deren Zustände) werden zu Stellen, Aktionen zu Transitionen 	<ul style="list-style-type: none"> Aktionen spezifiziert durch Anf.zust. Z, Eingangsnachricht E oder Ausgangsnachricht A, Folgezustand F Bsp: Tabelle für Auftragsgeber <table border="1"> <thead> <tr> <th>Prädikat</th> <th>Zustand</th> <th>Typ Eingangsnachricht</th> <th>Typ Ausgangsnachricht</th> <th>F-Zustand</th> </tr> </thead> <tbody> <tr> <td>Versand</td> <td>S erteilt erledigt</td> <td></td> <td>Auftrag Storno Bestätigung</td> <td>erteilt storniert bestätigt</td> </tr> <tr> <td>Empfang</td> <td>erteilt erledigt</td> <td>Ablehnung Erledigung</td> <td></td> <td>abgelehnt erledigt</td> </tr> </tbody> </table>	Prädikat	Zustand	Typ Eingangsnachricht	Typ Ausgangsnachricht	F-Zustand	Versand	S erteilt erledigt		Auftrag Storno Bestätigung	erteilt storniert bestätigt	Empfang	erteilt erledigt	Ablehnung Erledigung		abgelehnt erledigt	84
Prädikat	Zustand	Typ Eingangsnachricht	Typ Ausgangsnachricht	F-Zustand															
Versand	S erteilt erledigt		Auftrag Storno Bestätigung	erteilt storniert bestätigt															
Empfang	erteilt erledigt	Ablehnung Erledigung		abgelehnt erledigt															
4.3	Koordinations-Modell	<ul style="list-style-type: none"> Koordination: Vorgang, Abstimmen von Abhängigkeiten zwischen Aktivitäten, maßgeblich für Effizienz Koordinationsbedarf entsteht aus zu überwindenden räumlichen, zeitlichen, sachlichen und menschlichen Distanzen, Art/Struktur der Aufgabe, der Komplexität und Intensität der Abhängigkeiten zwischen Gruppenmitgliedern und Aktivitäten 	86																

	Technische Protokolle	Koordination findet auf 5 Stufen statt. - Untere Stufen: Zugriffs-, Zugangskontrolle auf Arbeitsmedien/Aufgaben → technische Protokolle (von Groupware gesteuerte Koordinationsverfahren) - obere Stufen: überwiegend soziale Protokolle (direkt Interaktion der Gruppenmitglieder mit/ohne gewohnte technische Unterstützung) - Kritik (Condon): technische Protokolle verlagern Konflikt auf Mensch-Computer-Ebene → Vorschlag: Rechner soll Konflikt nur erkennen, Lösung durch soziale Protokolle		88										
4.3.1	Koordinations-Theorie nach Malone	- Ziel: Formalisierung der Abhängigkeiten zw. Aktivitäten, Analyse/Bewertung versch. Koordinationsansätze - 4 Grundlegende Komponenten von Koordination: Ziele, Aktivitäten, Akteure, Abhängigkeiten (3 generische Typen: Voraussetzung, gemeinsame Ressource, Simultanität)	<table border="1"> <thead> <tr> <th>Komponente</th> <th>Koordinationsprozess</th> </tr> </thead> <tbody> <tr> <td>Ziele</td> <td>Ziele identifizieren</td> </tr> <tr> <td>Aktivitäten</td> <td>Abbildung von Zielen auf Aktivitäten</td> </tr> <tr> <td>Akteure</td> <td>Abbildung von Aktivitäten auf Akteure</td> </tr> <tr> <td>Abhängigkeiten</td> <td>Abhängigkeiten verwalten und handhaben</td> </tr> </tbody> </table>	Komponente	Koordinationsprozess	Ziele	Ziele identifizieren	Aktivitäten	Abbildung von Zielen auf Aktivitäten	Akteure	Abbildung von Aktivitäten auf Akteure	Abhängigkeiten	Abhängigkeiten verwalten und handhaben	90
Komponente	Koordinationsprozess													
Ziele	Ziele identifizieren													
Aktivitäten	Abbildung von Zielen auf Aktivitäten													
Akteure	Abbildung von Aktivitäten auf Akteure													
Abhängigkeiten	Abhängigkeiten verwalten und handhaben													
4.3.2	Kunden-Dienstleister-Modell	- Basis: konversationsorientierter Ansatz - Geschäftsprozess wird als Kunden-Dienstleister-Verhältnis dargestellt, typischerweise mit Verkettungen solcher Schleifen - 4 Phasen: erste Kontaktaufnahme, Kunde/Dienstleister verhandeln über zu erbringende Leistung, Dienstleister erbringt Leistung, Kunde beurteilt Leistung		91										

5. Agentensysteme

5.1	Einführung	- Agent: ein im Auftrag eines anderen in dessen Interesse Tätiger - es gibt passive Agenten (Untersteht der direkten Kontrolle des Benutzers), aktive Agenten (arbeitet teilweise autonom), intelligente (mit Weltmodell/Wissensbasis, erhält Auftrag und leitet Entscheidungen ab) und nichtintelligente Agenten (wie z.B. Software) - Einsatzbeispiele: Verteilte Terminplanung, Suchmaschinen, Groupware, ...		95
5.2	Informations-Agenten	- Unterstützt Benutzer bei der Suche nach Informationen in verteilten Systemen - Zielgerichtete Zusammenführung und Aktualisierung von Informationen, - Push-Technologie (dem Nutzer wird gezielt nutzerspezifische Information zugestellt - Bsp.: NewsWatcher		97
	Kooperations-Agent	- Lösung komplexer Probleme durch Kommunikation und Kooperation mit anderen Agenten - Bsp.: FireFly: Aufbau einer virtuellen Gemeinschaft für eine interaktive Kommunikation, Austausch in Foren, Filmfinder: Kontakt mit anderen (mit ähnlichen Filminteressen) möglich		98
	Transaktions-Agent	- Ausführung und Überwachung von Transaktionen - z.B. im E-Commerce: Kauf und Verkauf von Produkten/Services im Internet - Bsp: Jango: einfacher, bequemer und schneller Zugang zu Online-Anbietern unter einheitlicher Oberfläche, Unterstützung des Kaufvorganges, übernimmt Rolle eines Zwischenhändlers		99
5.3	Eigenschaften	- Autonomie: können eigenständige Entscheidungen treffen, ohne Interaktion ihres Benutzers handeln, semi-autonome Agenten führen Routineaufgaben aus - Soziale Kompetenz: können explizit (über Austausch strukturierter Nachrichten) oder implizit (über Effekte ihrer Aktionen auf Weltzustand) mit anderen Agenten kommunizieren - Reaktivität: nehmen Umwelt wahr und können in angemessener Weise reagieren. Echter reaktiver Agent: verfügt über Sensoren zur Aufnahme der Umwelt. Deliberativer Agent: besitzt internes Modell der Umwelt, aus dem er Schlüsse ziehen kann - Proaktivität/Zielorientiertheit: ergreifen selbständig die Initiative → brauchen wohldefinierte Ziele - Schlußfolgerungs/Lernfähigkeit: Intelligenz basiert auf 3 Komponenten: interne Wissensbasis, Schlußfolgerungen, Fähigkeit zu lernen/sich Umweltveränderungen anzupassen - Mobilität: Fähigkeit, sich innerhalb von Kommunikationsnetzen zu bewegen - Kommunikation: Agenten-Kommunikationssprache. Kooperation: komplexe Aufgaben erfordern Zusammenarbeit von Agenten → erweiterte Agenten-Kommunikationssprache - Charakter: Agent sollte nach außen ein möglichst menschähnliches Verhalten zeigen. Merkmale: Ehrlichkeit, Vertrauenswürdigkeit, Zuverlässigkeit		100

	Klassifikation	nach Handlungskompetenz : Autonomie, Kooperation, Verhandlung nach Mobilität : Stationär, Intranet, Internet nach Intelligenz : Urteilen, Planen, Lernen	104
5.4	Architektur von Agenten	<ul style="list-style-type: none"> - Prozesse eines intelligenten Agenten - Zentrale Aufgaben: Verarbeitung der aufgenommenen Informationen, Interpretation, Verfolgung eigener Ziele - Aktionen: Reaktion auf Ereignisse der Umwelt, proaktives selbständiges Handeln 	105
5.5	Mobile Agenten	<ul style="list-style-type: none"> - Programme in Form von autonomen Agenten, in einem heterogenen Netz umherwandern und dabei im Auftrag des Nutzers Dienst verrichten. A. entscheidet selbst, wann er den Rechner wechselt (Migrationsentscheidung) - Prozeßmigration: OS entscheidet, wer wohin migriert 	106
	Client-Server	für jeden Aufruf sind 2 Kommunikationsvorgänge involviert → kontinuierliche Kommunikationsverbindung erforderlich, Kontrolle wird an Serverprozeß übergeben	
	Remote Method Invokation	Kontrolle und Code werden auf entferntes System transportiert	
	Agent-Migration	<ul style="list-style-type: none"> - Agent entscheidet selbständig und abhängig vom inneren Zustand, wann und wohin er migriert - Kontrolle, Code, Daten (Zustand) werden auf entferntes System transportiert → nur 2 Kommunikationsvorgänge nötig → keine kontinuierliche Verbindung notwendig 	
	Bewertung mobiler Agenten	<ul style="list-style-type: none"> - Vorteile: Reduziert Netzlast, reduziert Ressourcenbelastung beim Client, Asynchrone Arbeitsweise, Rekonfigurierbare Dienste (Agenten statt Server stellen Funktionen zur Verfügung), Aktives Handeln, dezentrale Struktur - Probleme: Transport/Migration (Sicherheit), Effizienz (bisher wenig praktischer Erfahrungen), Standard/Interoperabilität (Einheitliche Systemumgebung für Agenten/Kommunikation), Abrechnungssysteme 	109
5.6	Kooperation zw. Agenten	<p>Kooperation mittels Nachrichten</p> <ul style="list-style-type: none"> - unstrukturiert: freies Format, natürlichsprachliche Formulierung → Agent braucht Sprachverständnis - semistrukturiert: teilweise feste Format, die ohne Sprachverständnis zugänglich sind - strukturiert: vollst. in festem, für Agenten lesbaren Format, Interessen menschlicher Leser unwichtig 	110
	Kooperationsarten	<ul style="list-style-type: none"> - zufällig: Agenten agieren unabhängig und beeinflussen unbewußt weitere Agenten - Master-Slave: Agent fordert einen anderen auf, etwas für ihn zu tun - Einwegkooperation: Agent bittet anderen um Hilfe, überläßt diesem volle Autonomie über Entsch. - gegenseitige Kooperation: für beide Seiten existiert ein Vorteil, sie entscheiden in gegenseitiger Kommunikation über den Fortschritt - gemeinsame Zielverfolgung: Kommunikation hinsichtlich Aufgabe und Zielsetzung, Ziele werden gemeinsam verfolgt 	111
	Kooperation bei gemeinsamer Zielsetzung	<p>Phasen:</p> <ul style="list-style-type: none"> - orientieren (gemeinsames Ziel finden), - Planen (gemeinsamen verbindlichen Plan finden), - zerlegen (erarbeiten der einzelnen, unterschiedlichen Aufgaben), - bearbeiten 	
5.6.1	Kontraktnetz-Protokoll	<ul style="list-style-type: none"> - Protokoll auf Anwendungsebene zur verteilten Problemlösung mit Agenten, basiert auf Konversation - Vergabe von (Teil-)Aufgaben und deren Ausführung. - Lokalisierungsproblem: finden geeigneter Partner. bei mehreren Kandidaten: Auswahl durch Verhandlung - Rollen: Aufgabensteller: fordert Ausführung der Aufgabe, überwacht Ausführung, verwendet Ergebnis. Anbeiter: verantwortlich für die Ausführung der Aufgabe - Rollenzuteilung kann sich zeitlich ändern - 5 Phasen: Aufgabenankündigung, Angebotserstellung durch evtl. mehrere Anbieter, Angebotsauswahl, Aufgabenzuteilung, Angebotsausführung 	113
	Verhandlung	<ul style="list-style-type: none"> - keine zentrale Kontrolle über Vergabe und Ausführung - jeder Agent evaluiert vom eigenen Standpunkt aus - Informationsaustausch in beide Richtungen, versch. Nachrichtentypen: Aufgabenankündigung, Aufgabenangebot, Aufgabenzuteilung, Anfrage, Bestätigung, Abbruch, Report 	114

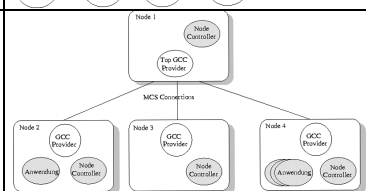
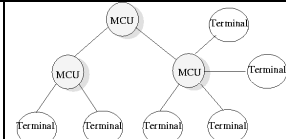
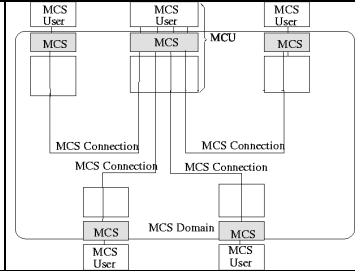
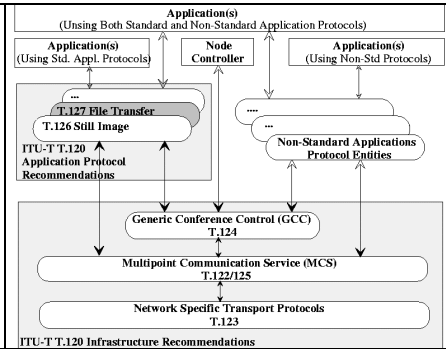
	Phase Ankündigung	<ul style="list-style-type: none"> - Nachrichtentyp Aufgabenankündigung: strukturierte Nachricht mit Feldern Aufgabenbeschreibung, Qualifikation des Anbieters, Angebotsinhalt, ... - Zielgruppenauswahl: Punkt-zu-Punkt (gezielte Verschickung an einen ausgewählten Anbieter), Multicast (gezielte Verschickung an eine Gruppe ausgewählten Anbieter), Broadcast (Problem: Skalierbarkeit) 	115															
	Phase Angebotserstellung	<ul style="list-style-type: none"> - Prozeß Contract nimmt Ankündigung entgegen - erfüllt Anbieter Anforderungen nicht, wird Auftrag verworfen - ansonsten: Agent wartet, bis Prozeß Execute gerade laufende Aufgabe beendet und schickt dann Angebot oder schickt Angebot sofort, da contract und execute parallel laufen 		116														
	Auswahl / Ausführung	<p>Angebotsauswahl: Zeitpunkt der Auswertung:</p> <ul style="list-style-type: none"> - sofort nach Erhalt - nach Erhalt einer gewissen Anz. von Angeboten - nach Ablauf einer definierten Zeit 	<p>Aufgabenausführung</p>	117														
5.6.2	Kommunikation zw. Agenten	<p>Agent Communication Language (Basis: Sprechakttheorie): besteht aus 3 Teilen</p> <ul style="list-style-type: none"> - Vokabular (ontologia): gemeinsam abgestimmte Sicht auf die zu modellierende Welt - Austauschformat für Wissensdarstellung (Knowledge Interchange Format, KIF): logische Operatoren - Metasprache und Protokoll für Agentenkommunikation (Knowledge Query and Manipulation Language, KQML) 	118															
	KQML	<p>Unterscheidung von 3 Ebenen:</p> <ul style="list-style-type: none"> - Kommunikation: Protokoll für technische Kommunikationsparameter - Nachricht: Festlegung des Sprechakttyps - Inhalt: hier wird nur ein allg. Rahmen vorgegeben 	<p>wichtige Sprechakttypen</p> <table border="1"> <thead> <tr> <th>Sprechakttyp</th> <th>Bedeutung</th> </tr> </thead> <tbody> <tr> <td>achieve</td> <td>Empfänger soll in seiner Umwelt eine Aussage wahr machen</td> </tr> <tr> <td>advertise</td> <td>Sender ist besonders geeignet für die Ausführung eines Sprechakttyps</td> </tr> <tr> <td>ask-one/all</td> <td>Sender möchte Antwort</td> </tr> <tr> <td>broker</td> <td>Empfänger soll Hilfe zur Beantwortung ausfindig machen</td> </tr> <tr> <td>subscribe</td> <td>Sender will Information über Antworten zu einem Sprechakt</td> </tr> <tr> <td>tell</td> <td>Sender überträgt eine Information</td> </tr> </tbody> </table>	Sprechakttyp	Bedeutung	achieve	Empfänger soll in seiner Umwelt eine Aussage wahr machen	advertise	Sender ist besonders geeignet für die Ausführung eines Sprechakttyps	ask-one/all	Sender möchte Antwort	broker	Empfänger soll Hilfe zur Beantwortung ausfindig machen	subscribe	Sender will Information über Antworten zu einem Sprechakt	tell	Sender überträgt eine Information	119
Sprechakttyp	Bedeutung																	
achieve	Empfänger soll in seiner Umwelt eine Aussage wahr machen																	
advertise	Sender ist besonders geeignet für die Ausführung eines Sprechakttyps																	
ask-one/all	Sender möchte Antwort																	
broker	Empfänger soll Hilfe zur Beantwortung ausfindig machen																	
subscribe	Sender will Information über Antworten zu einem Sprechakt																	
tell	Sender überträgt eine Information																	
	Dialoge in KQML	<p>einfacher Frage-Antwort-Ablauf</p>	<p>Einsatz eines Vermittlers</p>	120														
5.7	verteilte Terminplanung	<ul style="list-style-type: none"> - Aufgaben: Wann, Wie lange, Wer, Wo sollt Termin stattfinden - Einsatz von intelligente Agenten mit Wissen über Präferenzen der Person - Formale Definition: Terminvereinbarungsproblem (A, M) - $A = \{1, \dots, k\}$ Menge der beteiligten Agenten - $M = \{m_1, \dots, m_n\}$ Menge der zu vereinbarenden Termine m_i 	121															
	Spezifikation eines Termin	<p>Termin hat den Aufbau $m_i = (A_i, h_i, l_i, p_i, \tau_i, \sigma_i, d_i, T_i)$</p> <ul style="list-style-type: none"> - $A_i \subseteq A$: Menge der Teilnehmer, h_i: einladender Agent - l_i: Länge des Termins, p_i: Priorität des Termins - $\tau_i = (D, H)$ vom einladenden Agent bevorzugter Termin als Paar (Datum, Uhrzeit) - $\sigma_i = (D, H)$ spätester Termin als Paar (Datum, Uhrzeit) - $d_i = (D, H)$ spätester Zeitpunkt der Vereinbarung des Termins als Paar (Datum, Uhrzeit) - T_i: für Termin schließlich vereinbartes Intervall als Menge von Paaren (D, H) <p>Darstellung des Kalenders: Menge von Tripeln (D, L, x)</p> <ul style="list-style-type: none"> - D: Datum - L: Anzahl von Zeiteinheiten (z.B. Stunden) pro Arbeitstag - x: Zeiger auf Termin m_i oder nil 	122															
	Ablauf der Terminvereinbarung	<p>Mehrstufiger, dezentraler Prozeß, einladender Agent h_i übernimmt koordinierende Funktion, die anderen kommunizieren nicht miteinander.</p> <p>Ablauf:</p> <ul style="list-style-type: none"> - h_i bestimmt Zeitintervall im eigenen Kalender, falls nicht möglich: Absage, sonst: versenden - eingeladener Agent bestimmt lokale Lösung, sendet diese zurück - h_i sammelt Terminangebote, wertet sie aus: falls gemeinsames Intervall: Festlegung, sonst: neue Runde oder Abbruch 		123														

6. Konferenzsysteme

6.1	Synchrone Kooperation	<p>Benutzer verrichten Arbeit gleichzeitig und sind sich der Gleichzeitigkeit gewußt → Echtzeitanforderung</p> <p>Eigenschaften/Anforderungen an synchrone Groupware:</p> <ul style="list-style-type: none"> - kurze Antwortzeit: eigene Änderungen werden sofort sichtbar - kurze Benachrichtigungszeit: Änderungen anderer sollen schnell sichtbar werden - unbeständige Gruppenstruktur: Mitglieder soll kommen und gehen können - Zugriffskonflikte: Mechanismen der Nebenläufigkeitskontrolle zu Konsistenzhaltung nötig - Teilnehmer verfolgen keinen festgelegten Plan → Zugriffsvorhersage schwierig - Gruppenprozess meist verteilt - externe Informationskanäle (wie Audio und Video) 	128
6.2	Entwurf	Entwurf eines neuen gruppenbewußten Systems	129

		<ul style="list-style-type: none"> - Teilkomponenten sind verteilt; tauschen Nachrichten aus, um Konsistenz zu gewährleisten - Gruppenbewußtsein kann explizit berücksichtigt werden - hoher Implementierungsaufwand Nutzung bestehender Einzelanwendungen <ul style="list-style-type: none"> - Filtern und Sequentialisieren von Ein-/Ausgabe durch Abgreifen von Ereignissen - Geringe Flexibilität, keine Kontrolle - Geringer Realisierungsaufwand, reduzierter Lernaufwand für Nutzer 		
	Zentrale Architektur: Window-Sharing	<ul style="list-style-type: none"> - nur eine Instanz der Anwendung, an sie werden Eingaben weitergeleitet - Sequentialisieren/Ordnen/Filtern dieser erforderlich - Ausgaben müssen an alle Teilnehmer propagiert werden - Anw. Single-Point-of-Failure, Skaliert nicht - Bsp.: Xwindows: Pseudo-Windowserver: Sequentialisieren der Eingaben, Verwalten der Zuordnung zw. Eingabe und Window-Server, Verbreiten aller Ausgaben 		130
	Zentrale Architektur: Konferenz-Komponente	<ul style="list-style-type: none"> - Aufteilung der Anwendung in Präsentations- und Ausführungsteil - Konferenzkomponente: hat komplettes Wissen über Sitzung (→ Latecommer), macht Anwendung für synchrone Kooperation nutzbar: Koordiniert/Synchronisiert E/A. Zentrale Konferenzkomponente vermeidet Konsistenzprobleme, stellt aber Flaschenhals dar - Präsentationskomponente: kann Aktionen anderer ausblenden 		131
	Replizierte Architekturen	<ul style="list-style-type: none"> - eine Kopie der Anwendung befindet sich auf jedem Benutzerrechner - Vorteile: verbessert Antwortzeit, reduziert Netzbelastung (durch Zusammenfassen von Aktionen) - Probleme: unterschiedliche Anfangszustände der Anwendung möglich, deterministisches Verhalten der Applikation notwendig → identischer Zustand, Unkorrekte Reihenfolge der Eingabeereignisse, Reply-Notwendigkeit als Reaktion auf unerwünschte Ergebnisse, Wiederholungsmöglichkeit (latecommer) 		132
	Konferenz-Verwalter	<ul style="list-style-type: none"> - Anwendung: pro Benutzer repliziert → paralleles Arbeiten möglich - Konferenzschnittstelle: pro Benutzer repliziert, regelt E/A mit der gemeinsamen Anwendung, vergleichbar mit Präsentationskomponente, Kommuniziert Eingaben an andere Konferenzschnittstellen - Konferenzverwalter: nichtrepliziert, läuft auf ausgezeichnetem Rechner. Aufgaben: Verwaltung, Nebenläufigkeitskontrolle, ... 		133
	Gruppenbewußte Konferenzsysteme	<ul style="list-style-type: none"> - Verwenden keine bereits bestehenden Anwendungen - Integrale Bestandteile der Anwendung: Verwaltung der Bildschirminformation; Synchronisation und Kontrolle der Nebenläufigkeit (herausziehen zu einem Framework heute bevorzugt) 		134
6.3	Medienräume	<ul style="list-style-type: none"> - Kathedrale: gute Sichtweise, Gefahr unerwünschter Hierarchien - Bistro: informelle Arbeitsatmosphäre, Aufwendig (Mobilität der Tische, Verkabelung) - Klassisches Double: für Verhandlungen, guter Sichtkontakt 		135
	Technische Anforderungen Tele-Teaching	<ul style="list-style-type: none"> - Qualitative Anforderungen: „gute“ Audio-/Videoqualität, Synchronität zwischen den Datenströmen, Rückfragemöglichkeit - Datenströme: 3 Videostreame à ca. 200kBit/s, 3 Audioströme à 64 kBit/s, Daten ca. 20 kBit/s 		137
	Macs	Gemeinsamer virtueller Arbeitsraum: <ul style="list-style-type: none"> - mehrere Ebenen der Kommunikation: Gesamte Gruppen, private Gespräche einzelner - visuelle Sitzungskontrolle 		138

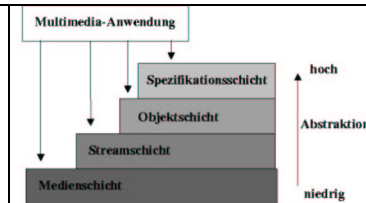
		- Versch. Anwendungen wie Chat, Audio, Vote, Tele-Editor, Whiteboard, Feedback	
6.4	Mbone	<ul style="list-style-type: none"> Multicast-Backbone, ein Overlay-Netz im Internet Konferenz-Anwendung mit Whiteboard: versch. Tool (Übertragung von Audio/Video, Whiteboard, ankündigung von Konferenzen) pubic-domain verfügbar, werden insbesondere im Schulungsbetrieb eingesetzt Unterstützung von Gruppenkommunikation im Internet Multicast IP: Gruppenadressierung mit beschränkter Ausdehnung von Multicast-Paketen IGMP: Bestandteil der IP-Schicht, „lose“ Gruppenzugehörigkeit, Informationen im Router, periodische Anfrage von Routern nach Subsystemen, die an Gruppenkommunikation teilnehmen 	140
	Mbone-Video	<ul style="list-style-type: none"> verfügbare Tool: vic, des weiteren nv und ivs unterstützen versch. Komprimierungsstandards (H.261, JPEG, ...), zum Erzielen akzeptabler Frameraten meist Hardwareunterstützung notwendig 	141
	VAT/RAT	<ul style="list-style-type: none"> VAT/RAT: Durchführung von Audiokonferenzen, jeder Teilnehmer wird aufgeführt, aktiver wird markiert RAT: Stellt Fehlerkontrolle bereit (derzeit: Sendewiederholungen), Aufzeichnung von Sitzungen 	142
	Whiteboard	Gemeinsame Arbeitsfläche mit Zeichenwerkzeugen, Tele-Pointer und –Marker, Grafik- und Postscript-Import, Zugriffskontrolle, WYSIWIS, basiert auf multicastfähigen Transportprotokoll, Nachteil: wird langsam bei vielen Teilnehmern	143
	Session Directory	<ul style="list-style-type: none"> Eintragen von Sitzungen im Mbone: Termin, Thema, beteiligte Medien, ... müssen angegeben werden, macht Vorschläge für Multicast-Adressen Auflistung von Sitzungen („Veranstaltungskalender“), Informationen werden periodisch an spezielle Gruppenadresse ausgesendet 	144
6.5	T.120	<ul style="list-style-type: none"> stellt Datenprotokolle für Multimedia-Conferencing sowie ein Rahmenwerk von Protokollen bereit Eigenschaften: Einberufen, konfigurieren und beenden von Konferenzen, Austausch von Teilnehmerinformationen (Identifizierung...), simultane Teilnahme an mehreren Konferenzen, mehrere Kommunikationskanäle innerhalb einer Konferenz, unabhängig vom unterliegenden Kommunikationssystem (Multicast oberhalb der Transportschicht) 	147
	MCS	<ul style="list-style-type: none"> dient dem Datentransport zwischen den Teilnehmer einer Konferenz Bildung von MCS-Domänen (alle mit einer MCU verbundenen Teilnehmer) Bildung logischer Kanäle innerhalb einer Domäne Verbindung zwischen Konten (=Teilnehmern) auf Basis von Punkt-zu-Punkt-Verbindungen 	149
	MCU	<p>Multipoint Control Unit:</p> <ul style="list-style-type: none"> Fungiert als zentraler Verbindungsknoten. der benötigte Funktionalität bereitstellt Verbindung zwischen MCU und Teilnehmer sowie MSCs ergibt Baumstruktur 	150
	GCC	<ul style="list-style-type: none"> Aufgaben: Einberufen, Verwalten und Beenden von Konferenzen, Registrieren und Verwalten der Teilnehmer, zentrale Datenbank mit vollständigen Konferenzdaten im Top GCC, Unterstützung moderierter Konferenzen auf Basis von Token Dienstprimitive (je als Request, Indication, response, confirm): join, create, query, ... 	151



7. Synchronisation in Multimediasystemen

7.1	Grundlegende Aspekte	<ul style="list-style-type: none"> zwischen Objekten können Beziehungen räumlicher, zeitlicher oder inhaltlicher Art bestehen diese müssen beim Abspielen (wieder-)hergestellt werden 	157
	inhaltlich	<ul style="list-style-type: none"> mehrere Objekte basieren auf dem gleichen Inhalt (z.B. Tabelle und Diagramm) Ziel: Automatische Erneuerung der verschiedenen Sichten auf die gleichen Daten (z.B. bei Änderungen) Anforderung: Explizite Definition der Abhängigkeiten; nur die Daten müssen verändert werden, die verschiedenen Sichten können dann automatisch generiert werden Implementierung basiert auf gemeinsamen Datenstrukturen oder gemeinsamen Objektschnittstellen 	158
	räumlich	<ul style="list-style-type: none"> räumliche Beziehungen werden als „Layout“-Beziehungen bezeichnet Verwendung von Layout-Frames: Position kann fest bzgl. eines Dokumentes, fest bzgl. einer Seite, relativ bzgl. der Position anderer Frames sein räumliche Beziehungen können auch für zeitabhängige Objekte bestehen (Gruppierung der Teilnehmer einer Videokonferenz, Audiopräsentation auf Stereoausgang) 	159

	zeitlich	<ul style="list-style-type: none"> - Typen: zwischen aufeinanderfolgenden Präsentationseinheiten eines Datenstromes (Intra-Objekt-Synchronisation) oder zwischen verschiedenen Datenströmen (Inter-Objekt-Synchronisation) - Beschreibende Attribute: gleichzeitig, unabhängig, nacheinander 	160
7.2	LDU	<ul style="list-style-type: none"> - für Synchronisation ist formale Darstellung notwendig → Datenströme werden in Logical Data Units gegliedert - repräsentiert eine Informationseinheit, es kann zwischen mehreren Granularitäten unterschieden werden, dies setzt eine hierarchische Zerlegung des Datenstromes voraus 	163
	offen / geschlossen	<ul style="list-style-type: none"> - geschlossen: vorhersagbare Dauer - offen: Dauer kann nur zum Präsentationszeitpunkt bestimmt werden (z.B. Live-Ton, Interaktion) 	
	„Einheit“ einer LDU	<ul style="list-style-type: none"> - physikalische Einheit: z.B. jede (geschlossene) LDU entspricht einem Bild - Sample-Blöcke: mehrere phys. Einheiten werden zu einem größeren Block zusammengefaßt, da einzeln zu klein, es sind geschlossene LDUs 	165
	Länge	Alternativen: Benutzerdefinierte Länge durch fest ausgewählte Größe (z.B. Computergenerierte Medienobjekte), LDUs wechselnder Länge (z.B. aufgezeichnete Benutzerinteraktion), offene LDUs unvorhersehbarer Länge, „leere“ LDUs (z.B. Zeitgeber alle 10ms)	166
7.3	Live-Synchronisation	<ul style="list-style-type: none"> - Ziel: Wiederherstellung der zeitlichen Beziehungen, die während der Aufzeichnung herrschten, Synchronisation erfolgt implizit - ohne Zwischenspeicher: Präsentation und Wiedergabe zeitlich nicht entkoppelt → Echtzeitanforderung an Manipulation des Datenstroms wie Anpassung der Darstellung an verfügbare Ressourcen. Mögliche Lokationen der Anpassung sind Quelle, Netz und Senke - mit Zwischenspeicher: Präsentation und Wiedergabe zeitlich entkoppelt → keine Echtzeitanforderung an Manipulation, es können andere Geräte herangezogen werden, Ändern der Abspielgeschwindigkeit und wahlfreier Zugriff möglich 	169
	Synthetische Synchronisation	<ul style="list-style-type: none"> - die beteiligten Objekte können unabhängig voneinander entstehen, die zeitlichen Beziehungen werden künstlich und explizit spezifiziert - 2 Phasen: Spezifikations- und Präsentationsphase 	172
7.4	Präsentations-Anforderungen	<ul style="list-style-type: none"> - erforderliche Genauigkeit hängt von der (individuellen) Wahrnehmungsfähigkeit des Menschen, Medium und Applikation ab. - man unterscheidet zeitl. Verschiebungen innerhalb oder zwischen Datenströmen sowie feststellbare und störende Verschiebungen - Kenntnis über auftretende Zeitverschiebungen erleichtern Implementierung und verbessern Portabilität 	172
7.4.1	Lippen-Synchronisation	<ul style="list-style-type: none"> - Audio-/Video-Synchronisation - Haupteinflussfaktoren: Inhalt (Kontinuierliche Ereignisse vs. diskrete Ereignisse, Hintergrund), Auflösung/Qualität, Ansicht (z.B. Schulteransicht), Hintergrundgeräusche/-musik, Sprache/Artikulation - +/- 80 ms unbemerkt, ab - 240 ms, + 160 ms störend (Asymmetrie durch alltägliche Erfahrung) 	174
7.4.2	Zeiger-Synchronisation	<ul style="list-style-type: none"> - diskrete vs. kontinuierliche Zeigerbewegungen - synchronisiert: - 500ms bis + 750 ms, störend ab -1000ms bis 1250ms 	179
7.5	Referenzmodell	<ul style="list-style-type: none"> - Modell notwendig, - um Anforderungen an die Synchronisation zu beschreiben, - Laufzeitmechanismen zu beschreiben und strukturieren, - zur Identifikation von Schnittstellen, - zum Vergleich von Lösungsansätzen 	182
	Medienschicht	<ul style="list-style-type: none"> - operiert auf einer Sequenz von LDUs - abstrahiert von geräteabhängiger Schnittstelle, bietet Operationen wie read(handle, LDU) - Die Anwendung ist verantwortlich für: Intra-Strom-Synchronisation durch Flußkontrollmechanismen, Ressourcenreservierung und -management, synchrone Darstellung diskreter Medien 	183
	Streamschicht	<ul style="list-style-type: none"> - operiert auf kontinuierlichen Datenströmen, Ströme innerhalb einer Gruppe werden parallel dargestellt, garantiert kontinuierliche Darstellung - abstrahiert auf Ströme mit Zeitparametern bzgl. QoS für Intra-Strom-Synchronisation, LDUs nicht mehr sichtbar - Schnittstelle nach oben: Zeitangaben für Echtzeitströme, Operationen wie start(stream), create_group(list), stop(group), ... - Anwendung ist verantwortlich für Spezifikation von Intra-Strom-Synchronisation, medienorientierten und das Zeitverhalten betreffende QoS sowie Synchronisation mit zeitunabhängigen Objekten (wie Bilder) 	185
	Objektschicht	<ul style="list-style-type: none"> - operiert auf zeitabhängigen und zeitunabhängigen Medienobjekten, verantwortlich für korrektes Scheduling der gesamten Präsentation - Schnittstelle: Spezifikation der Synchronisationsanforderungen, Operationen auf Objekten (wie prepare, start, run, destroy), Operationen verbunden mit Ereignissen 	187
	Spezifikations-Schicht	<ul style="list-style-type: none"> - stellt Werkzeuge zur Spezifikation der Synchronisation (wie Authoring-Tools) zur Verfügung - Aufgaben: konvertiert Spezifikation in das Format der Objektschicht, bildet QoS-Anforderungen des Benutzers auf die von der Objektschicht zur Verfügung gestellt QoS ab, bietet aber keine explizite Schnittstelle nach oben 	189
7.7	Spezifikation der Synchronisation	<ul style="list-style-type: none"> - Beschreibung aller zeitlichen Beziehungen der ein einem Medienobjekt enthaltenen Objekte, Teil der Beschreibung eines Medienobjektes - sollte QoS-Beschreibung für Intra- und Inter-Objekt-Synchronisation enthalten 	196



QoS für Medienobjekte	<ul style="list-style-type: none"> - einzelne Objekte: Angaben für einzelne LDUs sowie darüber, wie genau die zeitlichen Beziehungen eingehalten werden müssen - abhängige Medienobjekte: Angaben über die Zeitverschiebung (Akzeptanzgrenzen, erlaubte Synchronisationsgrenzen) 	<table border="1"> <thead> <tr> <th>Medium</th> <th>Modus, Applikation</th> <th>QoS</th> </tr> </thead> <tbody> <tr> <td rowspan="5">Video</td> <td>Animation</td> <td>Korreliert +/- 120 ms</td> </tr> <tr> <td>Audio</td> <td>Lippensynchronisation +/- 80 ms</td> </tr> <tr> <td rowspan="2">Bild</td> <td>Überlagerung</td> <td>+/- 240 ms</td> </tr> <tr> <td>keine Überlagerung</td> <td>+/- 500 ms</td> </tr> <tr> <td>Text</td> <td>Überlagerung +/- 240 ms</td> </tr> <tr> <td></td> <td>keine Überlagerung +/- 500 ms</td> </tr> <tr> <td rowspan="7">Audio</td> <td>Animation</td> <td>Ereigniskorreliert +/- 80 ms</td> </tr> <tr> <td rowspan="3">Audio</td> <td>eng gekoppelt (Stereo)</td> <td>+/- 11 µs</td> </tr> <tr> <td>schwach gekoppelt (Dialogmodus mit vielen Teilnehmern)</td> <td>+/- 120 ms</td> </tr> <tr> <td>schwach gekoppelt (z.B. Hintergrundmusik)</td> <td>+/- 500 ms</td> </tr> <tr> <td rowspan="2">Bild</td> <td>eng gekoppelt (Musik mit Noten)</td> <td>+/- 5 ms</td> </tr> <tr> <td>schwach gekoppelt (Diashow)</td> <td>+/- 500 ms</td> </tr> <tr> <td>Text</td> <td>Textanmerkungen +/- 240 ms</td> </tr> <tr> <td>Zeiger</td> <td>Audio hat Beziehung zu Gezeigtem - 500 ms + 750 ms</td> </tr> </tbody> </table>	Medium	Modus, Applikation	QoS	Video	Animation	Korreliert +/- 120 ms	Audio	Lippensynchronisation +/- 80 ms	Bild	Überlagerung	+/- 240 ms	keine Überlagerung	+/- 500 ms	Text	Überlagerung +/- 240 ms		keine Überlagerung +/- 500 ms	Audio	Animation	Ereigniskorreliert +/- 80 ms	Audio	eng gekoppelt (Stereo)	+/- 11 µs	schwach gekoppelt (Dialogmodus mit vielen Teilnehmern)	+/- 120 ms	schwach gekoppelt (z.B. Hintergrundmusik)	+/- 500 ms	Bild	eng gekoppelt (Musik mit Noten)	+/- 5 ms	schwach gekoppelt (Diashow)	+/- 500 ms	Text	Textanmerkungen +/- 240 ms	Zeiger	Audio hat Beziehung zu Gezeigtem - 500 ms + 750 ms	199
Medium	Modus, Applikation	QoS																																					
Video	Animation	Korreliert +/- 120 ms																																					
	Audio	Lippensynchronisation +/- 80 ms																																					
	Bild	Überlagerung	+/- 240 ms																																				
		keine Überlagerung	+/- 500 ms																																				
	Text	Überlagerung +/- 240 ms																																					
	keine Überlagerung +/- 500 ms																																						
Audio	Animation	Ereigniskorreliert +/- 80 ms																																					
	Audio	eng gekoppelt (Stereo)	+/- 11 µs																																				
		schwach gekoppelt (Dialogmodus mit vielen Teilnehmern)	+/- 120 ms																																				
		schwach gekoppelt (z.B. Hintergrundmusik)	+/- 500 ms																																				
	Bild	eng gekoppelt (Musik mit Noten)	+/- 5 ms																																				
		schwach gekoppelt (Diashow)	+/- 500 ms																																				
	Text	Textanmerkungen +/- 240 ms																																					
Zeiger	Audio hat Beziehung zu Gezeigtem - 500 ms + 750 ms																																						
Komplexe Spezifikation	<ul style="list-style-type: none"> - besteht aus mehr als 2 Medien, - kontinuierlichen und diskreten Medien 		199																																				
Anforderungen an Spezifikation	<ul style="list-style-type: none"> - Abstraktion des Inhaltes, die eine Beschreibung der zeitlichen Beziehungen zuläßt - Alle Synchronisationstypen sollen beschreibbar sein - die Spezifikation von QoS-Anforderungen muß möglich sein - Integration unterschiedlichen Medienobjekte muß möglich sein 		200																																				
Intervallbasiert Methode	<ul style="list-style-type: none"> - Zeitliche Beziehungen zwischen 2 Medienobjekten werden betrachtet - Vorteile: <ul style="list-style-type: none"> - offene LDUs können einfach gehandhabt werden (Interaktion kann einbezogen werden), - flexibel, erlaubt Spezifikation von Präsentationen mit versch. Laufzeiten - Nachteile: <ul style="list-style-type: none"> - Zeitverschiebungen sind nicht enthalten, Zeitbeziehungen innerhalb eines Objektes können nicht spezifiziert werden, - Flexibilität kann zu Inkonsistenzen führen 	<p>□ Operationen mit <i>einem</i> Verzögerungs-Parameter</p> <p>□ Operationen mit <i>zwei</i> Verzögerungs-Parametern</p> <p>□ Operation mit <i>drei</i> Verzögerungs-Parametern</p>	201																																				
Achsenbasierte Synchronisation	die Ereignisse einer Präsentation (z.B. Start, Stop) werden auf Achsen abgebildet, die den Objekten		205																																				
globale Achsen	<p>alle Objekte werden einer Achse zugeordnet, die eine Abstraktion der realen Zeit darstellt</p> <ul style="list-style-type: none"> - Vorteile: gute Abstraktion von der internen Struktur, leicht verständlich, Integration zeitabhängiger Objekte einfach, leichte Wartbarkeit (Objekte voneinander unabhängig) - Nachteile: Integration von Objekte unbekannter Dauer nicht möglich, Ströme müssen mit globaler Zeit synchronisieren, für Synchronisationsbeziehungen sind zusätzliche Angaben nötig 		206																																				
virtuelle Achsen	<p>Verallgemeinerung des Ansatzes mit mehreren Achsen:</p> <ul style="list-style-type: none"> - Koordinatensystem mit benutzerspezifischen Einheiten kann definiert werden - virtueller Koordinatenraum aus mehreren Achsen <p>Zur Laufzeit werden virtuelle Achsen auf reale Achsen abgebildet</p>		207																																				
Bewertung	<ul style="list-style-type: none"> - Vorteile: leicht verständlich, leichte Wartbarkeit, gute Abstraktion der internen Struktur, Integrierbarkeit zeitunabhängiger Objekte, Integration interaktiver Objekte möglich - Nachteile: Zeitverschiebung nicht direkt berücksichtigt, viele Achsen führen zu komplexer Spezifikation, Abbildung der Achsen zur Laufzeit kann komplex und zeitaufwendig sein 		209																																				
Kontrollfluß-basierte Synchronisation	<ul style="list-style-type: none"> - der Fluß von gleichzeitigen Darstellungen wird zu vordefinierten Punkten der Darstellung Synchronisiert - Varianten: hierarchische Synchronisation, Referenzpunkte, Timed Petrinetze 		210																																				
Hierarchische Synchronisation	<ul style="list-style-type: none"> - 2 Grundlegende Synchronisationsoperationen: seriell und parallel - Medienobjekte werden in einem Baum angeordnet - atomare Aktion: Darstellung eines „Single-Media“-Objektes, einer Benutzereingabe oder Verzögerung - zusammengesetzte Aktion: Kombination von Synchronisationsoperatoren und atomaren Aktionen 		211																																				

	Bewertung	<ul style="list-style-type: none"> - Vorteile: einfach verständlich, unterstützt Hierarchien, interaktive Objekte lassen sich integrieren - Nachteile: Zeitverschiebung muß separat beschrieben werden, zeitunabhängige Objekte müssen mit Präsentationsdauer versehen werden, Aufteilung von Medienobjekten zur Synchronisation kann erforderlich sein, keine adäquate Abstraktion bzgl. des Verhaltens eines Medienobjektes 	<pre> graph TD Root(()) --- Audio1[Audio 1] Root --- Video[Video] Root --- RI[RI] Root --- Bild1[Bild 1] Root --- Bild2[Bild 2] Root --- Bild3[Bild 3] Bild3 --- Anima1[Anima-tion 1] Bild3 --- Anima2[Anima-tion 2] Bild3 --- Audio2[Audio2] Bild3 --- Anima3[Anima-tion 3] Anima2 --- UI[UI] Anima2 --- Bild4[Bild 4] </pre>	212
--	-----------	---	---	-----