

CSCW in einer CORBA-basierten CA-Umgebung

Uwe von Lukas, Ute Dietrich

- 1 Motivation
- 2 Anforderungen an die Kooperationsunterstützung
- 3 Standardisierungen
 - 3.1 ODP
 - 3.2 Object Reference Model / CORBA
 - 3.3 T.120
 - 3.4 Bewertung
- 4 CSCW im Umfeld der Object Management Architecture
 - 4.1 CORBA-Dienste
 - 4.2 Ergänzende Dienste
- 5 Der TOBACO-Ansatz
 - 5.1 Umsetzung der Dienste
 - 5.2 Anwendungsszenario CAD
- 6 Zusammenfassung

Abstract

Computerunterstütztes kooperatives Arbeiten (CSCW) ist eines der wesentlichen Konzepte zur Parallelisierung des Produktentwicklungsprozesses. Mittelpunkt der Bestrebungen sind die Verkürzung der Produktentwicklungszeiten und die Verbesserung der Qualität von Erzeugnissen durch die Zusammenarbeit mehrerer spezialisierter Bearbeiter bereits zu einem sehr frühen Zeitpunkt.

Diese Form der Kooperation von, u.U. auch örtlich verteilten, Mitarbeitern eines Teams wird bislang nicht hinreichend unterstützt. Moderne CA-Architekturen zeichnen sich zwar heute durch eine weitgehende Modularität und Offenheit aus, stellen aber weder Funktionalität für eine einheitliche Kommunikation in einer heterogenen verteilten Systemumgebung noch für eine plattformübergreifende Kooperation im Sinne von Computerkonferenzen bereit. Ein offenes, einheitliches und herstellerunabhängiges Kommunikationsmodell bietet erst die Grundlage für die Realisierung offener Systemumgebungen mit integrierten CSCW-Anwendungen.

Ein allgemeines Konzept wurde von der Object Management Group standardisiert. Der Standard umfaßt die CORBA (Common Object Request Broker Architecture)-Spezifikation, welche eine objektorientierte Realisierung beliebiger verteilter Anwendungen in heterogenen Umgebungen erlaubt. Neben der bereits vollzogenen Standardisierung des ORB wurden und werden derzeit durch die OMG eine Reihe von allgemeingültigen Basisdiensten, den sogenannten Object Services, sowie optionale Dienste (Common Facilities) spezifiziert.

Im Rahmen dieses Beitrages soll ein auf CORBA basierendes Architekturkonzept vorgestellt werden, das eine kooperative Zusammenarbeit zwischen verschiedenen Bearbeitern und unterschiedlichen CA-Komponenten ermöglicht. Der Schwerpunkt liegt hierbei auf der Integration von Konferenzfunktionalität in ein CAD-Umfeld.

1 Motivation

Die heutigen Arbeitsweisen in der Produktentwicklung sind oftmals durch eine verstärkte Zusammenarbeit von Mitarbeitern sowohl innerbetrieblich als auch mit externen Partnern bzw. Zulieferern gekennzeichnet. Bekannte Kooperationszwänge sind die in vielen Bereichen gestiegene Komplexität der Produkte auf der einen und dem Bestreben nach kürzeren Produktentwicklungszyklen auf der anderen Seite. Weiterhin wird dieser Trend noch durch ein gestiegenes Outsourcing auf Unternehmensseite verstärkt. Die gewachsene Zusammenarbeit erfordert einen erhöhten Kommunikationsbedarf. Völlig neue Arbeitsformen und damit Anforderungen an Unterstützungswerkzeuge ergeben sich mit dem Einzug von Telearbeit in bisher konventionell organisierten Unternehmen.

In diesem Umfeld sind einerseits Werkzeuge für die Realisierung neuer CSCW-Anwendungen bereitzustellen. Andererseits sollen kommerziell verfügbare CA-Applikationen aus Gründen des Investitionsschutzes, nachträglich um CSCW-Funktionalität erweitert werden. Darüber hinaus besteht in vielen Unternehmen der Wunsch nach einer Integration verschiedener CA-Module, auch unterschiedlicher Hersteller, in ein Gesamtsystem [DK95]. Diese Wunschvorstellung beschreibt ihrem Wesen nach ein verteiltes, heterogenes System und läßt sich nur auf der Basis einer einheitlichen Architektur mit eindeutigen Schnittstellen realisieren.

Während die Forderung nach offenen, auf Standards beruhenden Entwicklungsplattformen in vielen Normungsgremien ihren Niederschlag bereits gefunden hat ([ISO95a],[OMG95a]), fehlten diese für die Realisierung von objektorientierten CSCW-Applikationen bislang. Vorherrschend sind hier Insellösungen und Prototypen, die oftmals nur eine ungenügende Integration in bestehende Anwendungen aufweisen.

2 Anforderungen an die Kooperationsunterstützung

Das computerunterstützte kooperative Arbeiten muß einer Vielzahl verschiedenster Anforderungen gerecht werden. Maßgeblich bedingt werden sie durch die vielfältigen Sichten auf das Thema Kooperation. Allgemein lassen sich diese Anforderungen im CA-Umfeld in konstruktionstechnische, informationstechnische und organisatorische Aspekte gliedern.

Eine optimale Einführung von Techniken des kooperativen Arbeitens ist nur bei einer genauen Kenntnis der organisatorischen Gegebenheiten in konkreten Unternehmen möglich. Die Analyse muß sich dabei auf die Arbeitsteilung, beteiligte Abteilungen und den Informationsfluß beziehen und die Ausgangsbasis für eine maßgeschneiderte Lösung bilden.

Betrachtet man CSCW unter konstruktionstechnischen Gesichtspunkten, so ist es notwendig, die verschiedenen Phasen der Produktentwicklung und -fertigung zu untersuchen. In jeder Phase sind unterschiedliche Personengruppen mit verschiedenen Bearbeitungsaufgaben involviert, deren Zusammenarbeit mit speziellen Werkzeugen unterstützt werden kann. Über den Zeitraum der Produktentwicklung, angefangen von der Angebotserstellung bis hin zur Fertigung, kann zu bestimmten Zeitpunkten die gesamte Palette von Werkzeugen sinnvoll eingesetzt werden. Dazu zählen sowohl solche für die Unterstützung synchroner als auch asynchroner Kooperation. Unter synchronem CSCW wird in unserem Kontext die gemeinsame Bearbeitung von Aufgaben innerhalb eines Konferenzszenarios verstanden. Asynchrones CSCW beinhaltet hauptsächlich die Steuerung und Automatisierung des Informationsflusses zwischen beteiligten, parallel arbeitenden Teammitgliedern. Typische Beispiele hierfür sind Workflowmanagement-Systeme mit entsprechenden Funktionen u.a. zur automatischen Benachrichtigung über Bearbeitungszustände, die Vergabe von Rollen und Zugriffsrechten.

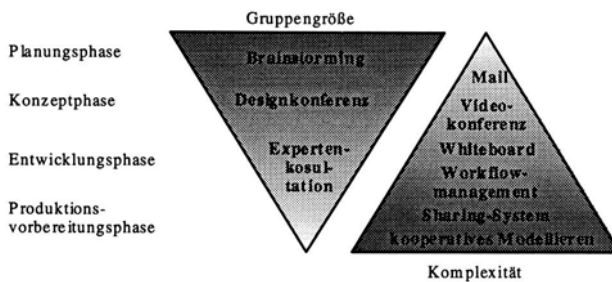


Abb. 1: Abhängigkeiten im Konstruktionsprozeß

Die Auswahl adäquater Werkzeuge kann u.a. in Abhängigkeit von der Gruppengröße und der konkreten Phase des Konstruktionsprozesses erfolgen (siehe Abb. 1). In der ersten Phase sind eine große Anzahl von Mitarbeitern aus unterschiedlichen Bereichen in den Prozeß involviert. Als unterstützende Werkzeuge können hier Email, WWW oder, zur einfachen Abstimmung, ein Audio/Video-Konferenztool eingesetzt werden. Mit zunehmender Detaillierung und Spezialisierung der zu bearbeitenden Aufgabe schrumpft die Gruppengröße, erhöht sich die Komplexität der Werkzeuge und steigen die zu berücksichtigenden Anforderungen. Die letzte Stufe ist hier die kooperative Modellierung, wo i.d.R. zwei bis drei Konstrukteure ein Modell im Rahmen einer CAD-Konferenz gemeinsam manipulieren.

Zu den informationstechnischen Aspekten zählen die speziellen Anforderungen an Hard- und Software. Betroffene Problemfelder sind hier insbesondere eine einheitliche, offene CSCW-Schnittstelle zur Realisierung der Kommunikation (Verbindungsaufbau, Synchronisation, Aktionsrechtverwaltung etc.) zwischen den Partnern, die Minimierung des im CAD-Bereich typischerweise hohen Datenaufkommens, die Einbindung von Audio- und Videoequipment sowie die Sicherung der Konsistenz der Daten. Die informationstechnischen Problemfelder

können prinzipiell in allgemeine, vom Anwendungskontext unabhängige, sowie in applikationsspezifische Anforderungen unterteilt werden. Zu denen für jede CSCW-Applikation notwendigen Forderungen gehören u.a.:

- Integration in die gewohnte Systemumgebung
- offene, flexible Lösung
- Telepräsenz durch parallele Audio- und Videoverbindung sowie Telepointer
- Multipointfähigkeit
- individuelle Granularität von Rede-/Aktionsrechten,
- angemessenes Interaktionsverhalten.

Als CAD-spezifische Anforderungen lassen sich u.a. die folgenden einordnen:

- Unterstützung echter kooperativer Modellierdialoge
- Konferenzen zwischen heterogenen, auch herkömmlichen, nachträglich um CSCW-Funktionalität erweiterten CA-Applikationen
- Konsistenzsicherung von Produktmodelldaten
- unterschiedliche Kopplungsmodi sowie
- öffentliche und private Konstruktionsräume.

CSCW-Applikationen stellen eine Sonderform verteilter Systeme dar. Somit haben alle an offene, verteilte Systeme gestellten Anforderungen und die auf diesem Gebiet unternommenen Forschungsaktivitäten auch im CSCW-Kontext Relevanz.

Die Vielfalt der Anforderungen läßt flexibel anpaßbare, generische CSCW- Werkzeuge für die Erstellung unterschiedlicher kooperativer Applikationen sinnvoll erscheinen [DLM96]. Ziel dieser Dienste ist die Erstellung konfigurierbarer CSCW-Applikationen, um so den unterschiedlichen Ansprüchen über den gesamten Produktentwicklungszyklus gerecht werden zu können. Der Fokus unserer Arbeit liegt hier insbesondere auf der Bereitstellung von Werkzeugen für synchrones CSCW.

3 Standardisierungen

Im Umfeld des kooperativen Arbeitens auf der einen und dem davon untrennbaren Themengebiet der offenen, verteilten Systeme auf der anderen Seite existieren eine Reihe von Standardisierungsbestrebungen unterschiedlicher Organisationen. Alle diese Standards bzw. sich

in der Normung befindlichen Bestrebungen aufzuführen und eingehend zu erläutern, würde den Rahmen dieser Arbeit sprengen. Aus diesem Grunde soll nur kurz auf die aus unserer Sicht wesentlichsten eingegangen werden und Standards wie OSF/DCE oder TINA-C unberücksichtigt bleiben.

3.1 ODP

Das Open Distributed Processing Referenzmodell (ODP-RM) der ISO [ISO95a,b] liefert ein generisches Modell für die Beschreibung verteilter Systeme. Durch die fünf bereitgestellten Sichtweisen (viewpoints) ist ein durchgängiger Einsatz des Modells in allen Stadien der Analyse und Spezifikation möglich. Es bildet auch die Grundlage für nationale und internationale Normungsbestrebungen auf dem Gebiet der verteilten Systeme. Innerhalb des Referenzmodells der ODP werden die folgenden Sichtweisen definiert: Die Unternehmenssicht fokussiert auf den Zweck, den Zuständigkeitsbereich und die Strategie eines verteilten Systems. Organisatorische Aspekte wie die Bildung von virtuellen Unternehmen oder der Einsatz von Diensten eines kommerziellen Anbieters stehen hier im Vordergrund. Die Informationssicht beschreibt die Semantik von Prozessen. Hierzu zählt die Modellierung des Workflows und der involvierten Dokumente in einer Domäne. Relevante CSCW-Aspekte auf dieser Ebene sind Zugriffsmechanismen und Replikationsstrategien für Dokumente. Die Verarbeitungssicht schaut in die abstrakten Prozesse und liefert eine funktionale Beschreibung auf Interface-Ebene. Die später folgende Charakterisierung von CSCW-Diensten erfolgt beispielhaft unter dieser Sichtweise. Die Engineeringsicht behandelt die Mechanismen und Funktionen, die nötig sind, um die Interaktion verteilter Objekte zu unterstützen. Modelle auf dieser Stufe lassen sich leicht in ausführbaren Code (z. B. C++) überführen. Die Technologiesicht dient der Abbildung auf unterstem, hardwarenahem Niveau. Diese Sicht und auch die Engineeringsicht gehen zu sehr ins Detail und sind im Rahmen einer allgemeinen Konzeption nicht von Interesse.

Für die Diskussion im CSCW-Bereich können die ODP-Konzepte einen Rahmen bilden, der eine Menge von Begriffen und Verfahren festlegt. Eine konkrete Spezifikation von CSCW-Diensten durch die ISO ist allerdings nicht zu erwarten.

3.2 Object Reference Model / CORBA

Eines der in unserem Rahmen wesentlichsten Konzepte wurde und wird von der Object Management Group spezifiziert. Das Ziel des 1990 gegründeten, herstellerübergreifenden Konsortiums ist die Bereitstellung einer geeigneten Architektur für die Verteilung und Zusammenarbeit von Softwarekomponenten in heterogenen Systemen. Die Spezifikation einer

objektorientierten Architektur stellt die Konzeption und Realisierung zukünftiger Software auf eine gemeinsame, objektorientierte Basis und ermöglicht damit, Softwarekomponenten verschiedener Hersteller auf verschiedenen Plattformen miteinander zu koppeln. Auf diese Weise sind CA-Systeme in der Erscheinungsform verteilter Objekte denkbar, die über Rechnergrenzen hinweg miteinander kommunizieren können. Hierdurch wird ein wesentlich stärkerer Grad der Kopplung unterschiedlicher Module möglich, als er bislang realisierbar ist.

Die OMA enthält drei Hauptkomponenten, auf deren Basis Applikationen, sogenannte Application Objects, erstellt werden können.

Der **Object Request Broker** (ORB) [OMG95a] ist die Kernkomponente der OMA. Seine Spezifikation erfolgte in der Common Object Request Broker Architecture, die sowohl Mechanismen für die Verteilung der Objekte im Netz als auch für die transparente Kommunikation zwischen den Objekten auf der Basis einer einheitlichen Schnittstellenbeschreibung definiert. Zu seinen Aufgaben zählen u.a. das Lokalisieren und Aktivieren von Servern, die Identifikation von Objekten im Netz, die Kodierung von Parametern, die Realisierung von Ausnahmebehandlungen und der Versand von Nachrichten. Um die eindeutige Beschreibung von Objekten zu erreichen, stellt die CORBA-Spezifikation eine an C++ angelehnte Sprache, die Interface Definition Language (IDL), bereit.

Die **Object Services** [OMG95b] stellen allen Komponenten allgemeingültige Basisdienste zur Verfügung, die in unterschiedlichen Bereichen benutzt werden können. Diese sind in der Regel generisch und müssen verschiedensten Ansprüchen gerecht werden. Als Beispiele für solche Dienste sollen die im CSCW-Kontext relevanten Dienste unter Abschnitt 4 vorgestellt werden.

Die **Common Facilities** [OMG95c] stellen (Branchen-)spezifische Bausteine und höherwertige Dienste bereit, die jeweils eine applikationsunabhängige Funktion oder einen Marktsektor unterstützen. Zu den Common Facilities zählen u.a. User Interface, Agent Facility, Rule Management und Marktbereiche wie CIM, Simulation, Buchhaltung, Softwareentwicklung.

Die Erstellung OMA-basierender Applikationen erfolgt durch die Kombination dieser sog. Application Objects mit den applikationsnahen Komponenten der Common Facilities und den Basisdiensten, die durch die Object Services bereitgestellt werden.

Die OMG verfolgt einen ODP-konformen Ansatz in Bezug auf die Erzielung von Offenheit, Integration und Portabilität. Die in der CORBA 2.0-Spezifikation vorgesehene Festlegung der Interoperabilität von ORBs entspricht in vielen Dingen dem in der ODP-Traderfunktion beschriebenen Förderationskonzept, über welches mehrere verschiedene Trader miteinander kooperieren. Deutlich werden die Zusammenhänge zwischen beiden Normen auch durch die teilweise Übernahme von Spezifikationen vom einen in den anderen Standard. So wird die

von der OMG spezifizierte IDL im ODP-Referenzmodell übernommen. Umgekehrt integriert die OMA die Traderspezifikation in ihre Common Object Services [PSW95].

3.3 T.120

Seitens der ISO und der International Telecommunication Union (ITU) wurde der Bedarf nach Standards auf dem Gebiet der Konferenzsysteme erkannt. Die Normung von Audio- und Videoströmen war hierzu lediglich der erste Schritt. Die jüngsten Aktivitäten in diese Richtung bestanden in der Veröffentlichung einer ersten Version der Serie T.120, die eine Reihe von Standards für audiovisuelle Konferenzsysteme beinhaltet [ITU95]. Damit steht nun u.a. eine einheitliche generische Konferenzverwaltung bereit und es können auch applikationsspezifische Daten zwischen Anwendungen im Rahmen einer Konferenz ausgetauscht werden. Beispiele hierfür sind Annotationen im Sinne einer Whiteboardanwendung oder auch Positionen eines Telepointers. Ein Vorteil des Ansatzes liegt darin, daß durch die Nutzung von sogenannten Multipoint Control Units (MCU) auch dann eine Mehrpunktverbindung aufgebaut werden kann, wenn die Arbeitsstation nur über eine Punkt-zu-Punkt Anbindung (beispielsweise über ISDN) verfügt. Terminals und MCUs werden in einer Baumstruktur vernetzt. Hierdurch wird die Grundlage für eine Kommunikationsinfrastruktur gegeben, die eine beliebige Anzahl von Konferenzpartnern unterstützt.

3.4 Bewertung

Die Normungsbestrebungen auf dem Gebiet der verteilten Systeme im allgemeinen und des kooperativen Arbeitens im speziellen sind durchaus positiv zu bewerten. Das ODP Referenzmodell steckt dabei den großen Rahmen ab und liefert Begriffe und Methoden zur konkreten Beschreibung verteilter Systeme. Die darauf aufbauende Architektur der OMG findet seitens der Industrie eine stark wachsende Beachtung. Problematischer gestaltet sich die Kombination von CORBA und T.120: Durch die asynchrone und unstrukturierte Kommunikation über die oben beschriebenen Schichten eignet sich T.120 nur bedingt für den Einsatz im Rahmen eines objektorientierten Systems. Zudem gibt es mit dem alternativen Kommunikationsdienst CORBA Überschneidungen bei der Spezifikation der Abbildung auf verschiedene Netze und Übertragungsprotokolle. Möglich ist hier eine CORBA-konforme Kapselung der von T.120 bereitgestellten Dienste. Die Kommunikation zwischen den Anwendungsobjekten und dem CSCW-Dienst kann dann über CORBA und die Kommunikation zwischen den Knoten über das T.120-Protokoll abgewickelt werden.

4 CSCW im Umfeld der Object Management Architecture

Als vielversprechende Plattform hat sich das von der Object Management Group verfolgte Konzept in vielen Forschungsaktivitäten etabliert. Sowohl das durch die OMA vorgegebene einheitliche Kommunikationsmodell als auch die Bereitstellung universeller Basisdienste und Facilities bieten eine leistungsfähige Grundlage für die Entwicklung offener objektorientierter Systeme, die auch die nachträgliche Integration externer Komponenten bzw. die Einbringung neuer, OMA-konformer Dienste komfortabel unterstützt. In der bisher vorliegenden Spezifikation ist allerdings keine direkte Unterstützung von CSCW-Anwendungen vorgesehen, daher müssen fehlende Werkzeuge in das Konzept integriert und als entsprechende Dienste angeboten werden.

4.1 CORBA-Dienste

Durch die OMG wurden bereits eine Reihe von CORBA-Diensten und Facilities vorgestellt, die eine Grundlage für die Entwicklung verteilter Systemumgebungen bilden. Für die Erstellung von Konferenzanwendungen sind hauptsächlich die folgenden Services relevant:

Der **Event Service** stellt Mechanismen für die Realisierung eines Multicast bereit. Dies ist eine wesentliche Voraussetzung für Konferenzen mit mehreren Teilnehmern. Der Event Notification Service dient der gezielten Benachrichtigung von Objekten hinsichtlich bestimmter Ereignisse und realisiert die Entkopplung der (direkten) Kommunikation zwischen Objekten. Dafür werden innerhalb des Dienstes zwei Rollen definiert: die Rolle des Lieferanten (*supplier*) und die Rolle des Verbrauchers (*consumer*). Lieferanten erzeugen Event-Daten, während Verbraucher diese behandeln können. Die Kommunikation zwischen Lieferanten und Verbrauchern erfolgt über den Austausch von Event-Daten auf der Basis von Standard-CORBA-Requests entweder in typisierter oder generischer Form.

Der **Transaction Service** besitzt bei der Standardisierung eine geringere Priorität als der oben beschriebene Event Service. Durch die Überschneidungen mit dem Gebiet der Datenbanken und der Kooperation mit den maßgeblichen Gremien, insbesondere der ODMG, sind hier noch zahlreiche Änderungen des vorliegenden Entwurfs zu erwarten. Die grundlegende Funktionalität wird davon allerdings nicht betroffen sein. Durch diesen Service soll ein möglichst breites Spektrum an Anwendungsgebieten und Modellen unterstützt werden. Dazu zählen flache und verschachtelte Transaktionen sowie domänenübergreifende Föderation von Transaktionsdiensten.

Ein **Replication Service** für die Erzeugung und Verwaltung von Objekt-Replikaten war von der OMG ursprünglich als eigenständiger Dienst geplant. Doch wird diese Funktionalität inzwischen eher als Bestandteil des ORB gesehen und dort vermutlich in spätere Versionen

aufgenommen. Von besonderem Interesse ist hier die Semantik bei der Kopie von Verweisen, wobei unterschiedliche Alternativen vorhanden sein sollten. Ein weiterer Hauptaspekt liegt in der Konsistenzsicherung der verteilten Replikate.

Der **Concurrency Control Service** ermöglicht den simultanen Zugriff mehrerer Clients unter Wahrung der Konsistenz und Koheränz sämtlicher beteiligter Objekte. Dabei wird zwischen transaktionsbasierten und einem threadbasierten Locking-Mechanismus unterschieden. Beide Modelle sollen unterstützt und in einem System parallel verwendet werden können.

Zusätzlich sind eine Reihe von weiteren Diensten einsetzbar, die aber aufgrund ihrer eher untergeordneten Funktion nicht explizit aufgeführt werden sollen. Zu diesen gehören u.a. Dienste zum Erzeugen und Löschen sowie zur Migration von Objekten.

4.2 Ergänzende Dienste

Um CORBA-konforme CSCW-Applikationen erstellen zu können, müssen zusätzliche, im Standard bislang nicht vorgesehene Dienste bereitgestellt werden. Diese müssen Funktionalitäten für eine Gruppenkommunikation, für die Bearbeitung von Streams und die Konferenzverwaltung beinhalten.

Die Standardisierung eines Dienstes für die **Gruppenkommunikation**, wie er beispielsweise in [ISI93] beschrieben wurde, ist ein wichtiger Schritt für die Implementierung CORBA-basierter, kooperativer Anwendungen. Ein solcher Dienst wird benötigt, wenn mehr als zwei Teilnehmer an einer Konferenz beteiligt sind. Unter einer Objektgruppe soll dabei eine Sammlung von kooperierenden Objekten gleichen Typs verstanden werden. Diese Gruppe sollte sich nach außen wie ein einzelnes Objekt präsentieren. Eine Folge von Nachrichten, die an die Gruppe gerichtet werden, erreicht alle Gruppenmitglieder stets in derselben Reihenfolge. Diese Isochronität ist die erste Voraussetzung für die notwendige Konsistenz innerhalb der Projektgruppe. Eine derartige Gruppenkommunikation ist nicht nur im Rahmen des kooperativen Arbeitens sinnvoll, sondern kann auch zur Steigerung der Fehlertoleranz eingesetzt werden. In unserem Kontext soll jedoch davon ausgegangen werden, daß die parallel existierenden Instanzen an unterschiedlichen Orten aktiv sind und über eine Benutzungsoberfläche mit einem Nutzer in Verbindung stehen. Sollen nun Aktionen, die durch einen Nutzer initiiert wurden, an allen Standorten gespiegelt werden, so wird ein Funktionsaufruf an die Objektgruppe abgesetzt. Solch ein Multicast erleichtert und beschleunigt die Verbreitung von Nachrichten an zahlreiche Teilnehmer. Ein weiteres Merkmal der Gruppenkommunikation ist die dynamische Mitgliedschaft. Objekte können sich jederzeit bei der Gruppe an- oder abmelden. Auf diese Weise können auch Teilnehmer, die zu Beginn einer Konferenz nicht eingebunden sind, später integriert werden. Optimalerweise erfolgt dies durch eine Benach-

richtung des Basic Object Adapters. Er muß über die Aufnahme und das Ausscheiden von Objekten informiert werden, um eine aktuelle Liste der involvierten Gruppenmitglieder halten zu können.

Für die Umsetzung von Gruppenkommunikation sind folgende Alternativen denkbar :

- Die Gruppe wird durch ein ausgezeichnetes Gruppen-Objekt repräsentiert, das alle Aufrufe weiterleitet. Dieser Ansatz stempelt die eigentlichen Gruppenmitglieder zu zweitklassigen Objekten, da nur das Gruppen-Objekt nach außen in Erscheinung tritt und beim Broker registriert werden kann. Zudem ist dieser Ansatz recht fehlerträchtig, da beim Ausfall des vermittelnden Objekts die gesamte Kommunikation scheitert. Auch Performance-Nachteile sind bei einem solchen Ansatz zu erwarten.
- Spezielle Anpassungen an der Client-Bibliothek könnten die Verteilung übernehmen. Dabei wird der Multicast über sequentielle Punkt-zu-Punkt-Verbindungen simuliert oder ein eigenes Multicast-Protokoll eingesetzt, das dann parallel zum herkömmlichen Kommunikationsmechanismus im ORB arbeitet.
- Eine eingeschränkte Gruppenunterstützung ist basierend auf dem generischen Event-Service realisierbar. Die fehlende Typisierung bei diesem Ansatz steht dabei allerdings im Widerspruch zu dem sonst streng typisierten Ansatz der OMA.

Die Integration von **Streams** in die OMA ist eine wichtige Voraussetzung für die Einbindung von Audio und Video im CSCW-Umfeld. Diese sollten einen flexiblen Mechanismus bieten, generische Datenströme zwischen definierten Endpunkten bereitzustellen. Streams erlauben es, multimediale Kommunikationsströme direkt in die Anwendung zu integrieren. Bislang erfolgt die Nutzung von Audio- und Videoapplikationen in der Regel getrennt von der CORBA-basierten Anwendung. Neben Attributen und Operationen sollte dieser spezielle Zugriffsmechanismus von der Schnittstelle bereitgestellt werden. Somit ist die aktuelle Mächtigkeit der IDL nicht ausreichend. Es wird notwendig, ein neues syntaktisches Konstrukt zu definieren. Neben dem Typ des Streams sollten dabei weitere Parameter übergeben werden können. Hierzu zählen insbesondere Anforderungen an Quality of Service (QoS) und Prioritätswerte. Anwendungen in Echtzeit, wie sie bei einer Konferenzschaltung gegeben sind, stellen dabei besondere Anforderungen, die vom Kommunikationsdienst erfüllt werden müssen. Andere, parallel übertragene Daten, sollten dann mit untergeordneter Priorität behandelt werden.

Die optimale Realisierung sowohl der Streams als auch der Gruppenkommunikation bedingen auch Erweiterungen bzw. Anpassungen des ORB und ggf. der IDL [OMG96].

Als ein Dienst, der direkt dem Entwickler von CSCW-Applikationen angeboten werden soll, wird hier eine **Konferenz-Unterstützung** beschrieben. Im Gegensatz zu Streams und der

Gruppenunterstützung ist die hier vorgestellte Funktionalität näher bei der Anwendung positioniert und besitzt nicht deren Generalität und Universalität. Somit läßt sich dieser Dienst auf der Ebene der Common Facilities ansiedeln. Die Konferenzverwaltung stellt durch eine komfortable Schnittstelle all die Funktionalität zur Verfügung, die anwendungsunabhängig zusammengefaßt werden kann. Dabei greift sie auf die beiden zuvor beschriebenen Dienste zu und verleiht ihnen Semantik. Unter der Konferenz-Unterstützung wird die folgende Funktionalität bereitgestellt:

- die Verwaltung von Nutzern und verwendeten Applikationen im Rahmen der Konferenz
- die Verwaltung von Aktionsrecht-Tokens
- Telepointer- und Annotationsoverlays
- Speicherung des Konferenzzustands für später hinzukommende Teilnehmer
- Dateiübertragung an mehrere Benutzer
- ein generischer Dienst zur Kopplung der Anwendungen

Dem letzten Punkt kommt dabei eine besondere Bedeutung zu. Dieser Dienst muß für jede Anwendung speziell angepaßt werden und beschreibt dann ein applikationsbezogenes Protokoll, das die eigentliche Grundlage für die Kommunikation und Koordination festlegt. Es bestimmt den Grad der Kopplung und die Granularität bei der Kooperation zwischen den Systemen.

5 Der TOBACO-Ansatz

Das im ZGDV Rostock bearbeitete Projekt TOBACO (Tool Based Cooperation) verfolgt die Bereitstellung von Basisdiensten für das kooperative Arbeiten [DLM95]. Dabei wird auf die objektorientierte Architektur der OMG und die bereits beschriebenen Dienste aufgesetzt. Eine Einbindung von monolithischen Systemen ist über Wrapper möglich. Es wird bislang von einem vollständig replizierten Ansatz ausgegangen: zwei komplette Instanzen einer Anwendung werden im Rahmen einer Konferenz gekoppelt.

Die entstehenden Werkzeuge sollen für zwei Anwendungsfälle genutzt werden können:

1. Die Implementierung von neuen CSCW-fähigen Applikationen.
2. Die nachträgliche Erweiterung bestehender Applikationen um Funktionalität des kooperativen Arbeitens.

Typischerweise ist der Leistungsumfang in Bezug auf das kooperative Arbeiten bei der zweiten Variante eingeschränkt. Durch die mangelnde Offenheit und fehlende Schnittstellen heutiger Systeme kann nur eine Teilmenge von Funktionen in einer Konferenz genutzt werden.

Die Konzeption der Dienste verfolgt einen universellen Ansatz und ist nicht auf ein spezielles Anwendungsgebiet beschränkt. Bisherige Arbeiten konzentrieren sich jedoch auf den CAD-Bereich. Das Konzept basiert auf der unter 4 dargestellten Aufteilung in Services, Common Facilities und benutzerdefinierten Objekten. Eine Konferenzanwendung entsteht dann wiederum durch die geeignete Kombination aus Objekten aller drei Klassen.

Im obigen Abschnitt wurden prinzipielle Anmerkungen zur Unterstützung des kooperativen Arbeitens mit CORBA gemacht. Der folgende Abschnitt konkretisiert die Aussagen, geht auf die gewählte Implementierung ein und detailliert den Leistungsumfang der Dienste.

5.1 Umsetzung der Dienste

In unserem Kontext wird die Realisierung einer Gruppenkommunikation auf der Basis des bereits spezifizierten Event-Service der OMG vorgenommen. Dieser Weg ist bislang die einzige Möglichkeit, eine standardkonforme Gruppenunterstützung bereitzustellen. Dabei können allerdings die folgenden Anforderungen an einen idealen Group Service nicht realisiert werden:

- Eine Objektgruppe sollte für den Nutzer transparent sein, die Kommunikation erfolgt also nicht über einen expliziten Kanal (Event Channel), sondern über die normale Nutzung eines entfernten Objekts.
- Es sollten Mechanismen definiert werden können, die regeln, wie sich die Objektgruppe bei Anfragen verhält. Mögliche Alternativen sind hier die Abgabe einer einzigen Antwort, die Bildung eines Mittelwerts, oder die Rückgabe aller Antworten in Form einer Sequenz. Bei der Rückgabe einer einzigen Antwort sind ebenfalls verschiedene Ansätze denkbar. Diese können sich auf Reaktionsgeschwindigkeit, Entfernung oder die explizite Auswahl eines Objektes (u.a. aufgrund entsprechender Qualitätsparameter) beziehen.
- Es sollen unterschiedliche Protokolle für die Ordnung von Nachrichten bereitstehen. Typische Beispiele sind ungeordneter, kausal oder total geordneter Multicast. Der Event Service in seiner vorliegenden Form unterstützt nur einfachen Multicast.

Diese Nachteile werden allerdings bewußt zugunsten der Konformität mit den bislang standardisierten Komponenten in Kauf genommen. Hierfür wurde ein Event-Service implementiert, der die Basis für die Kommunikation zwischen den angeschlossenen Teilnehmern bildet. Ein Kanal ist dabei stets für eine bestimmte Klasse von Ereignissen zuständig. Die Kopplung

von Instanzen einer Konferenzanwendung erfolgt durch mehrere parallele Kanäle (siehe Abb. 3).

Der im vorangegangenen Abschnitt beschriebene Stream-Service ist bislang noch nicht realisiert. Solange es von der OMG noch kein Konzept für die Unterstützung von kontinuierlichen Datenströmen gibt, erscheint eine Implementierung nicht sehr sinnvoll. Der für eine Konferenz unbedingt notwendige Audiokanal (und evtl. ein zusätzlicher Videokanal) wird durch externe Konferenzwerkzeuge bereitgestellt. Hier kommen die entsprechenden Standards aus dem Bereich der Telekommunikation zum Tragen. Eine Integration von Audio, Video und Applikationskopplung zu einer einheitlichen Dienstgruppe ist wünschenswert, allerdings bislang noch nicht möglich.

Die Umsetzung der Konferenzverwaltung erfolgt in Anlehnung an T.120, da sich etliche der Anforderungen an einen solchen Dienst dort wiederfinden. Die Funktionen werden durch sogenannte Application Protocol Entities bereitgestellt, die auf einen speziellen Multipoint Communication Service aufsetzen. Diese APEs bilden eine applikationsabhängige Gruppierung von CSCW-Funktionen. Der MCU-gestützte Multicast wird in unserer Architektur durch die Kommunikation auf der Basis von Objektgruppen ersetzt. Diese Vorgehensweise erleichtert auch eine zukünftige Kopplung von CORBA-basierten Konferenzen und T.120-konformen Systemen. Somit wurde die Architektur von T.120 weitgehend übernommen, jedoch objektorientiert gegliedert und mit CORBA als Kommunikationsschicht implementiert.

5.2 Anwendungsszenario CAD

Als erstes Anwendungsgebiet wurde der CA-Sektor ausgewählt. Hier konnte schon auf Erfahrung in der Implementierung eines kooperativen geometrischen Modellierers [DL95] zurückgegriffen werden. In TOBACO werden die Arbeiten auf diesem Gebiet weitergeführt – es wird allerdings eine komplett objektorientierte, CORBA-konforme Architektur verwendet. Das prototypische System umfaßt dabei die folgenden Komponenten: Geometriekern, User Interface und Finite State Machine. Diese bilden eine Instanz eines Einzelplatzsystems und werden im folgenden unter CAD zusammengefaßt. Die Separierung von Interaktion und interner Struktur, wie sie durch eine Finite State Machine gegeben ist, erleichtert sowohl das Abfangen und Einbringen von Ereignissen als auch die Konsistenzsicherung während der Konferenz. Zum kooperativen Arbeiten werden zusätzlich eine Konferenzunterstützung mit Telepointer und generischen Funktionen zur Verwaltung von Konferenzen genutzt. Weiterhin existiert ein CA-spezifisches Objekt zur Kopplung der Instanzen. Die zugehörige Klasse wurde in Form eines Applikationsprotokolls für den Bereich der geometrischen Modellierung spezifiziert, das die Spiegelung von Nutzeraktionen auf einem mittleren Granularitätsgrad erlaubt. Dabei werden nicht alle Teilschritte bei der Erzeugung oder Modifikation eines Kör-

pers übertragen, sondern lediglich die komplette Aktion. Beispielsweise resultiert die Erzeugung einer Kugel unter Eingabe von 3D-Koordinaten für den Mittelpunkt und einer Angabe für den Radius in einem Kommando `make_sphere(origin, rad)`.

Die folgenden IDL-Fragmente beschreiben zwei Objekte: den generischen CSCW-Teil des Applikationsprotokolls (Application Protocol Entity) und eine davon abgeleitete Beschreibung einer Applikationskopplung für den Bereich CAD. Beide nutzen (evtl. unterschiedliche) Event-Channels zur Kommunikation mit entfernten Instanzen.

```
interface APE {
    readonly attribute ConfInfo info;
    void join_conference(in UserID user);
    void leave_conference(in UserID user);
};

interface CAD : APE {
    Body make_sphere(in Coord origin, in long radius);
    Body make_cuboid(in Coord origin, in long a, in long b, in long c);
    void lock_body(in Body b);
    void delete_body(in Body b);
};
```

In den Implementierungsklassen dieser Dienste muß intern eine Umsetzung des synchronen Protokolls auf die asynchrone Arbeitsweise des Event Channels erfolgen.

Nutzt man verschiedene CA-Systeme mit demselben Geometriekern, ermöglicht das auf diesen Kern abgestimmte Protokoll mit Einschränkungen auch die Konferenz unterschiedlicher Systeme. Dabei können Eigenentwicklungen auch mit kommerziellen Komponenten kombiniert werden. Problematisch ist auch hier wieder die Einschränkung des Leistungsumfanges: kommerzielle Systeme haben eine sehr umfassende Modellierfunktionalität, wie sie bei eigenen Prototypen nie erreicht werden kann. Allerdings läßt sich die CSCW-Funktionalität im Prototyp wesentlich umfassender integrieren, da man dort den Zugriff auf alle internen Daten und Ereignisse hat. Abbildung 2 zeigt eine Konstellation mit drei Systemalternativen: Host A beherbergt ein kommerzielles CORBA-basiertes System, dessen Schnittstellen nicht vollständig offengelegt sind. Auf Host B läuft ein prototypischer Modellierer, auf dessen Funktionalität man komplett zugreifen kann. Host C benutzt ein kommerzielles CA-System, das über einen Wrapper angebunden ist. Der ORB dient als Kommunikationsbus, auf den die CSCW-Dienste in der oben beschriebenen Weise aufsetzen. Der Dienst für die Gruppenkommunikation und Teile der Konferenzverwaltung sind dabei zu einer CSCW-Komponente zusammengefaßt, die auf jedem der an einer Sitzung beteiligten Knoten instantiiert sein muß.

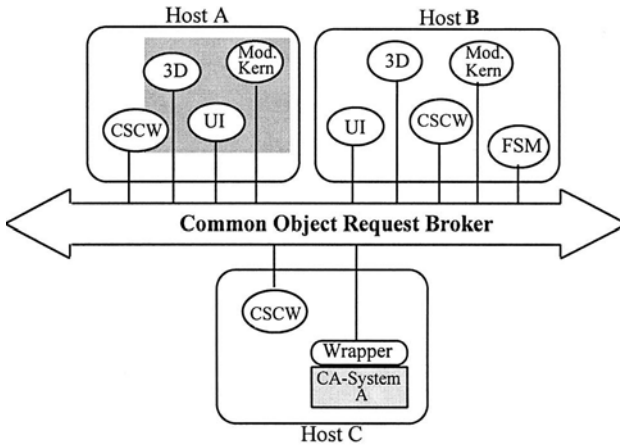


Abb. 2: TOBACO-basierte CA-Konferenz

Abbildung 3 verdeutlicht den Aufbau einer Konferenz unter Nutzung von TOBACO-Diensten.

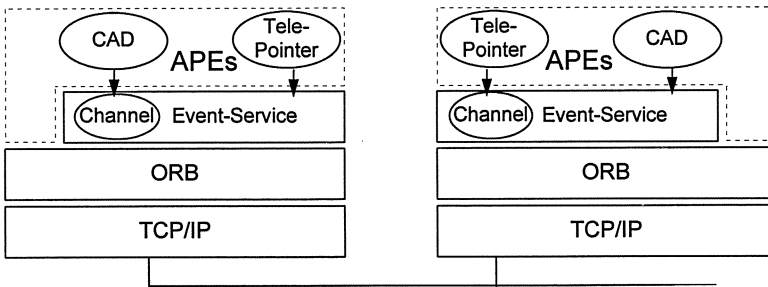


Abb. 3: Umsetzung der Objektgruppen im Schichtenmodell

Die Gruppenkommunikation soll anhand des Telepointer-APE kurz erläutert werden: Logisch betrachtet bilden die APE-Instanzen auf den unterschiedlichen Knoten eine Objektgruppe. Faktisch wird die Kommunikation über ein einziges zugehöriges Channel-Objekt umgesetzt, welches seinerseits über den ORB mit den angeschlossenen Telepointer-Objekten kommuniziert.

Eine alternative Implementierung unter Nutzung des durch T.120 spezifizierte Protokoll für die Multicast-Kommunikation ist geplant. Auf diese Weise erhält man eine voll standardkonforme Lösung. Weitere zukünftige Arbeiten umfassen die Bereitstellung unterschiedlicher

Mechanismen zur Aktionsrechtvergabe und feinkörnigere Lockingstrategien für konkurrierend bearbeitete Objekte. Zudem ist eine Übertragung der Werkzeuge auf andere Anwendungsgebiete vorgesehen.

6 Zusammenfassung

Die Realisierung unternehmensinterner und -übergreifender Kommunikation in allen Phasen der Produktentwicklung ist heute ein wesentlicher Faktor für die Wettbewerbsfähigkeit von Unternehmen. Hinzu kommen neue Trends in der Arbeitsorganisation, die mehr und mehr die Telearbeit auch vom heimischen Rechner bzw. entsprechenden Telearbeitszentren favorisieren. Diese Entwicklung fördert den Bedarf nach flexiblen CSCW-Werkzeugen für die Kommunikationsunterstützung. Dabei haben sich die Probleme, die durch die Heterogenität von Hard- und Software verursacht werden, als eine der größten Hürden bei der Erarbeitung umfassender Lösungen herausgestellt. Dieses Problem trifft insbesondere auf den CA-Sektor zu. Hier sind neue Konzepte notwendig, die eine Integration bestehender und neuer Architekturen in ein Gesamtsystem ermöglichen.

Der aufgezeigte Ansatz orientiert sich an den zukünftigen internationalen Standards der ISO/ITU und der OMG und versucht auf diese Weise zur Lösung der aufgezeigten Problematik beizutragen. Um CORBA-konforme Werkzeuge bereitstellen zu können, müssen entsprechende Dienste entwickelt werden, die bislang innerhalb der Normierung nicht hinreichend berücksichtigt sind. Ein großer Teil der Anwendungen kann zur Zeit daher nur mit neuen, eigenentwickelten Lösungen umgesetzt werden. Die in TOBACO konzipierten Dienste können dabei aufgrund ihres universellen Charakters auch auf andere Anwendungsgebiete übertragen werden.

Eine vollständige Nutzung solcher Dienste erfordert idealerweise eine neue Generation von modularen, netzübergreifend verteilten CA-Systemen, die aus beliebig miteinander kombinierbaren CA-Modulen bestehen. Diese Vorgehensweise ermöglicht die Einbeziehung vorgefertigter Komponenten und Dienste und ist somit ein wichtiger Schritt in Richtung elektronischer Märkte. Offen bleiben hier allerdings Fragen nach der Granularität und Beherrschbarkeit hochgradig verteilter Systeme sowie die rechtlichen Aspekte, die durch den Einsatz dienstorientierter Softwarekomponenten entstehen.

Literatur

- [DK95] U. Dietrich, B. Kehrer: Integration und Kooperation - Stand und Perspektiven. it+ti 5/95
- [DL95] U. Dietrich, U. v. Lukas: Kooperatives Arbeiten in einer offenen heterogenen Umgebung, in CA-Integration in Theorie und Praxis, Springer 1995
- [DLM95] U. Dietrich, U. v. Lukas und I. Morche: Rechnergestütztes kooperatives Arbeiten im CAD-Umfeld, Computer Graphik Topics 6/95 Vol. 7
- [DLM96] U. Dietrich, U. v. Lukas und I. Morche: Kooperatives Modellieren auf der Basis von standardisierten Diensten. Tagungsband CAD '96. Informatik Xpress, März 1996.
- [ESS96] K.P. Eckert, P. Schoo und G. Schürmann: Open Distributed Processing Platforms for support of telecommunication applications and their management. In Proceedings of the IFIP/IEEE International Conference on Distributed Platforms - Industrial Stream/Poster Session (Dresden, Feb. 1996), pp. 236-240.
- [FL96] K. Farooqui und L. Logrippo: Group Communication Models. To be published in Computer Communications Journal, 1996.
- [ISI93] ISIS Distributed Systems, Inc.: Object Groups: A response to the ORB 2.0 RFI. OMG Document 93-04-11, Apr. 1993.
- [ISO95a] ISO/ITU: Basic Reference Model of Open Distributed Processing Part 1: Overview. International Standard 10746-1, ITU-T Recommendation X.901, 1995.
- [ISO95b] ISO/ITU: Basic Reference Model of Open Distributed Processing Part 2: Descriptive Model. Intern. Standard 10746-2. ITU-T Recommendation X.902, 1995.
- [ITU95] ITU Study Group 8: Initial Draft of T.120: Data Protocols for Multimedia Conferencing, Oct. 1995.
- [NPC95] G. Nilsson, F. Dupuy und M. Chapman: An overview of the Telecommunication Information Networking Architecture. In Proceedings of the TINA'95 Conference (Melbourne, Feb. 1995).
- [OMG95a] Object Management Group: The Common Object Request Broker: Architecture and Specification (Revision 2.0). Framingham MA, July 1995.
- [OMG95b] Object Management Group: Common Object Services Specifications, volume 1. Framingham MA, March 1995.
- [OMG95c] Object Management Group: Common Facilities Specifications (Revision 4.0). Framingham MA, Jan. 1995.
- [OMG96] OMG Telecommunications SIG. Streams and QoS: a White Paper. Feb. 1996.
- [PSW95] C.Popien, G. Schürmann und K.-H.Weiß: Verteilte Verarbeitung in Offenen Systemen. B.G. Teubner 1995.
- [TRM94] J. Trevor, T. Rodden und J. Mariani: The Use of Adapters to Support Cooperative Sharing. In Proceedings of the Conference on Computer Supported Cooperative Work (Oct. 1994), pp. 219-230.