



Universität Karlsruhe
Fakultät für Informatik
Institut für Telematik
76128 Karlsruhe



Klausurtagung des Instituts für Telematik

Mainz, 29.–31. März 1998

Herausgeber:
Prof. Dr. Dr. h.c. Gerhard Krüger
Roland Bless
Stefan Dresler

Universität Karlsruhe
Institut für Telematik

Interner Bericht 10/98
ISSN 1432-7864

Kurzfassung

Der vorliegende Bericht gibt einen Überblick über aktuelle Forschungsarbeiten des *Instituts für Telematik der Universität Karlsruhe* in den Bereichen Hochleistungskommunikation, verteilte Systeme, Cooperation&Management und Telekooperation. Er ist in zwei Teile gegliedert. Der erste beschreibt die persönlichen Interessensgebiete der wissenschaftlichen Mitarbeiter. Danach folgt eine Darstellung der Kooperationsprojekte des Instituts. Im Anhang finden sich aktuelle Eigenveröffentlichungen der Mitarbeiter des Instituts.

Im Forschungsbereich *Hochleistungskommunikation* liegen die Schwerpunkte in der Entwicklung und Implementierung innovativer Protokollarchitekturen, in der Entwicklung fortschrittlicher Subsystemarchitekturen und auf Managementaspekten. Der Teilbereich fortgeschrittene Kommunikationssysteme befaßt sich speziell mit neuartigen Protokollmechanismen zur Unterstützung von Gruppenkommunikation, mit Aspekten der Mobilkommunikation und mit der Definition eines integrierten Dienstgütemanagements. Im Bereich innovative Subsystemarchitekturen steht die effiziente Realisierung spezieller Kommunikationsfunktionen in Hochleistungsszenarien im Mittelpunkt (z.B. durch neuartige Betriebssystemkonzepte). Der Managementbereich umfaßt traditionelles Netzwerkmanagement und Monitoring, insbesondere in Hochleistungsnetzen (FDDI, ATM), sowie qualitätsorientierte Managementfunktionen.

Der Forschungsbereich *Verteilte Systeme* deckt am Institut für Telematik das verteilte Datenmanagement, spezielle Systemunterstützung für verteilte Anwendungen und Agenten-modellierte Ansätze ab. Im ersten Bereich werden Konzepte objektorientierter Datenbanken unter den Aspekten der Adaptivität von Systemverhalten, der Integration mobiler Benutzer sowie der adaptiven Wissenspropagierung untersucht. Im zweiten Bereich stehen dedizierte Systemmechanismen wie lokationstransparente Aufrufe und dynamische Objektmigrationen in einer verteilten C++-Umgebung sowie Kooperationsmechanismen für gemeinsames Arbeiten im Rahmen verteilter Konferenzsysteme im Mittelpunkt. Im dritten Bereich wird als Erweiterung des objektorientierten Ansatzes der Einsatz von verteilten, kooperierenden Agenten in verteilten Systemen untersucht.

Im Forschungsbereich *Cooperation&Management* werden Kooperationsaspekte im Management von verteilten Systemen untersucht. Das Management umfaßt sämtliche Maßnahmen und Vorkehrungen, um die Netzkomponenten (z.B. Router, Sternkoppler, Modems) und die daran angeschlossenen Rechner (z.B. Hosts, Workstations, PCs) sowie die darauf laufenden Anwendungen effizient zu betreiben. In einer größeren Umgebung mit vielen tausend Netzanschlüssen benötigt man eine Gruppe von qualifizierten Mitarbeitern, die zur erfolgreichen Erledigung dieser Aufgaben zusammenarbeiten, also kooperieren müssen. Ein Forschungsziel besteht darin, den Kooperationsaspekt durch geeignete Workflow-Managementwerkzeuge zu unterstützen und damit effizienter zu gestalten.

Das *TecO* konzentriert sich als rein drittmittelgeförderte Technologietransfereinrichtung am Institut für Telematik auf anwendungsnahe Forschung in Kooperation mit einem breiten Spektrum von Partnern. Gegenstand der Forschungs- und Entwicklungsaktivitäten ist die softwaretechnische Unterstützung moderner verteilter Anwendungen mit den Aspekten der Multimedia-Integration, der Kooperation verteilter Teams und der Mobilität von Anwendern und Software. Die Beiträge des TecO reichen dabei von Middleware als Basis solcher Anwendungen über Werkzeuge für Entwicklung, Betrieb und Wartung bis hin zu Demonstratoren innovativer Applikationen.

Abstract

This report gives an overview of current research topics at the *Institute of Telematics of the University of Karlsruhe*. It covers work in the areas of high performance networking, distributed systems, cooperation&management, and telecooperation, and comprises two parts. After presenting the research interests of the faculty in the first part, a brief description of the cooperation projects of the institute is given.

In the area of *High Performance Communication*, the main topics of interest are advanced protocol architectures, innovative subsystem architectures, and management aspects. On the side of advanced communication systems, one deals with protocol support for group communication as well as with mobile communication and forthcoming service models to support Quality-of-Service. In the area of innovative subsystem architectures, the efficient realization of specific communication functions in high performance scenarios (e.g., by using operating systems suitable for communication) is of interest. The work in the management area focuses on traditional network management as well as on monitoring, especially in high speed networks (FDDI, ATM), and on management functions related to quality.

The research work of the *Distributed Systems Group* focusses on data management, system support for distributed applications and agent-based approaches. In the first part the main interests are concepts for building an object-oriented database system as an intelligent information backbone enabling adaptive system behaviour, integrating mobile users, and spreading the knowledge about available data services. Secondly, we investigate an extended distributed object-oriented environment supporting a uniform object model, location-independent invocation and dynamic migration of fine-grained objects. Another aspect is a multimedia collaboration system, which allows individuals to participate in an audio and video conference. In the third part, a new effort concerns agent based distributed systems, in which autonomous agent objects cooperate by following their own flow of control.

The area of *Cooperation&Management* investigates workflow aspects in IT management. This includes the entire range of steps and precautions to run all network devices (e.g., routers, hubs, modems) and the linked computers (e.g., hosts, workstations, PCs) with their applications. Large environments with several thousands of computer systems and network connections need a group of qualified employees. This group has to work together, i.e., to cooperate, to execute IT management efficiently. An important research goal is to analyze how this kind of cooperation can be supported by using customized workflow management tools.

The *TecO* as entirely third-party funded technology transfer group is focussed on applied research and collaborates with a broad a range of partners in the software and multimedia industries. Research activities are targeted at software technology for advanced distributed applications featuring multimedia-integration, collaborating teams, and mobility of users and software components. Contributions of TecO cover middle-ware as enabling technology, tools for development, operation and maintenance, and innovative applications.

Vorwort

Der vorliegende Tagungsband entstand im Rahmen der siebten Klausurtagung des Instituts für Telematik, die vom 29. bis zum 31. März 1998 in Mainz abgehalten wurde. Er dokumentiert in kurzen Beschreibungen die Forschungsgebiete der wissenschaftlichen Mitarbeiter des Instituts sowie die Beteiligung des Instituts an Kooperationsprojekten.

Die Kürze der einzelnen Beiträge erlaubt einen schnellen, aber prägnanten Einblick in die einzelnen Arbeitsgebiete. Der Preis dafür ist die Vernachlässigung konzeptioneller und technischer Detailinformationen, ohne die allerdings wissenschaftliche Arbeit nie praxisnahe Resultate liefern könnte. Daß auch diese Details existieren, zeigt ein Blick in die zitierten Veröffentlichungen, das Durcharbeiten der Studien- und Diplomarbeiten bzw. Dissertationen in den einzelnen Gebieten oder auch der persönliche Austausch mit den jeweiligen Autoren. Der vorliegende Band kann dabei eine zuverlässige Orientierungshilfe im weiten Feld der wissenschaftlichen Arbeit des Instituts sein.

Dieser Bericht verdankt seine Existenz wieder einmal der gemeinsamen Anstrengung aller Mitarbeiter des Instituts. Durch die Verwendung vorgegebener Templates konnte die Zusammenstellung der einzelnen Beiträge zu dem vorliegenden Tagungsband ohne größere Probleme durchgeführt werden. Ohne den Einsatz aller Mitarbeiter wäre das jetzige homogene Bild des Bandes nie zustande gekommen.

Während der vorliegende Band durch Mitarbeiter des „Institutsteils Krüger“ koordiniert wurde, soll nicht verschwiegen werden, daß die Planung und Durchführung (und damit der Hauptteil der Arbeit) in diesem Jahr von den Mitarbeitern des TecO, insbesondere Michael Beigl, geleistet wurde. Nicht zuletzt sei Herrn Prof. Dr. Dr. h.c. G. Krüger gedankt, ohne dessen Initiative dieser Tagungsband nicht entstanden wäre.

Karlsruhe, im März 1998

Roland Bless, Stefan Dresler

Inhaltsverzeichnis

Zusammenfassung	i
Abstract	iii
Vorwort	v
I Persönliche Beiträge	1
<i>Michael Beigl:</i>	
Ein Ubicom Interface Design Vorschlag	3
<i>Roland Bless:</i>	
Dienstqualitätsbasiertes Routing	11
<i>Elmar Dorner:</i>	
Drahtloses ATM – ATM-Architektur mit Unterstützung mobiler Endsysteme	17
<i>Stefan Dresler:</i>	
Von Unicast über Multicast und Concast zu Multipeer: Anforderungen an Kommunikationsprotokolle	23
<i>Martin Gaedke:</i>	
WebComposition: Ein objektorientiertes Unterstützungssystem für das Web Engineering	29
<i>Meng Gan:</i>	
Anwendung von Neuronalen Netzen und Fuzzy-Logik in der ATM-Verkehrslastregelung	39
<i>Hans-Werner Gellersen:</i>	
Internet Call Center – Innovative Assistenzdienstleistungen durch CTI auf dem Information Highway	45
<i>Arnd Grosse:</i>	
Bedarfsorientierte Informationsvermittlung in verteilten Trading-Systemen	49
<i>Jörn Hartroth:</i>	
Von Hamstern zu Ameisen – Ein Klon-basierter Mobilitätsansatz für Agenten	59

<i>Markus Hofmann:</i> LGMP/LGCP: Eine Protokoll-Suite für skalierbare Multicast-Kommunikation im Internet	69
<i>Christian Mayerl:</i> Prozeß-orientierter Ansatz zur Spezifikation betreibergerechter Managementwerkzeuge	75
<i>Günter Schäfer:</i> Effiziente Authentisierungsverfahren für Hochleistungsnetze	81
<i>Hartmut Ritter und Jochen Schiller:</i> Rückgekoppeltes, parametrisierbares Scheduling zur Erbringung von Dienstgüte in mehrschichtigen Kommunikationssystemen	85
<i>Albrecht Schmidt:</i> Automatisierung der WWW-Programmierung	97
<i>Robert Scholderer:</i> Assistenten zur Unterstützung des Betriebs von vernetzten Systemen	105
<i>Jochen Seitz:</i> Reaktive und adaptive Platzierung von Proxy-Objekten	111
<i>Rainer Ruggaber:</i> Quality of Service (QoS) in verteilten objektorientierten Systemen . .	117
<i>Hajo R. Wiltfang:</i> Topologie- und Dienstgütee Erfassung in ATM-Netzwerken	127
II Projekte	137
<i>Michael Beigl:</i> Mercado - ein Multimedia Pilot	139
<i>Martin Gaedke:</i> Mobilitätsdienste in Zusammenarbeit mit der Daimler-Benz AG . . .	141
<i>Martin Gaedke und Albrecht Schmidt:</i> Victor II	145
<i>Arnd G. Grosse, Jörn Hartroth und Rainer Ruggaber:</i> Sonderforschungsbereich 346 – Die Desktop-Metapher als Integrationsansatz in der rechnergestützten Fertigung	147
III Eigenveröffentlichungen des Instituts	153

Teil I

Persönliche Beiträge

Ein Ubicom Interface Design Vorschlag

Michael Beigl

Kurzfassung

Ubiquitous Computing stellt neue Anforderungen an das Design von Mensch-Computer Schnittstellen. Bisherige User-Interface Methoden und Werkzeuge sind nicht geeignet, um solche neuen Schnittstellen zu entwerfen. Deshalb müssen ausgehend von bekannten User-Interface Prinzipien neue Methoden und Regeln entwickelt werden. Ubicom Interface Design stellt einen ersten Ansatz einer Methode zur Erstellung von Ubiquitous Computing Schnittstellen dar. Der Vorschlag wird mit Hilfe eines Beispiels ausgeführt und verdeutlicht. Zum Schluß werden erste Entwurfsregeln aus allgemeinen HCI Prinzipien für den Entwurfsprozeß abgeleitet. Praktische Gesichtspunkte geben einen Ausblick auf zukünftige Forschungsarbeiten in diesem Gebiet.

1 Einleitung

Eine der neueren revolutionären Visionen im Bereich Informatik ist Mark Weisers Entwurf des Ubiquitous Computing (Ubicom) [Weis91]. Fast sieben Jahre nach dessen Idee stehen wir zwar immer noch am Anfang der vorhergesagten Entwicklung, aber erste Geräte und Technologien sind schon erhältlich. Allerdings stellen diese noch eher vereinzelte, nicht analog zu Weisers Konzept zusammenarbeitende Geräte dar, obwohl der Realisierung der Vision erster Ubicom Szenarien technisch nichts mehr im Weg stünde. Einer der Gründe dafür ist wohl, daß Ubicom Umgebungen neue Anforderungen an die Gestaltung der Benutzerschnittstellen stellen, Oberflächen heutiger Anwendungen allerdings weiterhin nach den herkömmlichen Methoden des Mensch-Computer Dialogs (Human Computer Interaction, HCI) entworfen werden.

Designer benötigen "Handreichungen" in Form von Entwurfsregeln, um gute Ergebnisse zu erzielen. Aber weder Entwurfsregeln noch Methoden oder Werkzeuge für diese neue Form der Benutzerschnittstellen sind bisher entworfen. Dieser Artikel soll neben der genaueren Problemspezifikation einen ersten Ansatz für die Erstellung von U(bicom) I(nterface) Design Anwendungen geben.

2 Unterschiede zu traditionellen UI Methoden

Ein bekanntes und in der Lehre oft verwendetes Beispiel für traditionelle Betrachtung von Mensch-Computer Schnittstellen ist das aus *Curricula for Human Computer Interaction* der ACM SIGCHI [Hefl92] entnommene Human-Interaction Modell. In dieser

Grafik sind deutlich einige Annahmen der klassischen HCI zu erkennen: Die Interaktion des Menschen mit dem Computer erfolgt via **einem** Bildschirm und **einem** Zeigergerät, Manipulationsobjekte sind zum Beispiel als Grafiken oder Menus dargestellt. Demgegenüber sind Ubicom Benutzerschnittstellen durch folgende Eigenschaften gekennzeichnet:

- Sie besitzen erheblich mehr Interaktionspunkte. Weiser geht von etwa 100 solcher Schnittstellen des Menschen zum Computer pro Raum aus.
- Eine Benutzerschnittstelle bildet ein entsprechendes reales Objekt nach. Simuliert ein "computerisiertes Objekt" ein reales Objekt, zum Beispiel ein PDA einen Notizzettel, dann ist dieser Computer mindestens für die Zeit der Anwendung "single use", im Gegensatz zu bisher verwendeten "multi purpose" Rechnern, die gleichzeitig mehrere Anwendungen repräsentieren.
- Auch passive Objekte sind Teil der Benutzerinterfaces. Sie können durch Detektoren erkannt werden. Dieses Paradigma ist eine Erweiterung der ursprünglichen Philosophie von Ubiquitous Computing, die im MediaLab entwickelt wurde.
- Einbeziehung des Hintergrunds nicht als gegebene, sondern als veränderbare Größe mit gleicher (Design)Berechtigung wie die im Vordergrund ablaufenden Prozesse. Ein gutes Beispiel ist die Symbolisierung des Datenstroms durch einen Wasserfall. Dieses Rauschen wird vom Administrator im Hintergrund wahrgenommen. Ansteigendes Rauschen zeigt höhere Netzlast an, die dann bei Überschreiten einer gewissen Schwelle vom Administrator bewußt wahrgenommen wird, aber dennoch auch unter dieser Schwelle ihre Bedeutung hat.
- Einbeziehung verschiedenartiger Manipulations- und Ausgabeobjekte anstatt eines oder mehrerer gleicher, wie sie schwerpunktmässig bei "herkömmlichen" Computer UIs betrachtet werden.

Dies erfordert keine Abkehr von Prinzipien bisheriger Benutzerschnittstellengestaltung, wohl aber von Regeln, Methoden und Werkzeugen, welche die Benutzerschnittstellen des Menschen zum Computer gestalten helfen.

Um Ubicom Schnittstellen entwerfen zu können, werden also Prinzipien, die auf den Menschen bezogene Erkenntnisse betreffen, aus der HCI weiterverwendet. Dazu gehören Erkenntnisse aus der menschlichen Wissensverarbeitung, der Sprach- und Kommunikationswissenschaften und der Ergonomie. Allerdings müssen die aus diesen Wissenschaften verwendeten Themen durch weitere Themen ergänzt werden. So werden bisher zum Beispiel Erkenntnisse aus der bewußten Wahrnehmung in der Ergonomie in den Vordergrund gestellt. Ubicom Schnittstellen bieten aber auch nicht im Vordergrund stehende Schnittstellen, die nur unterbewußt wahrgenommen werden sollen.

Bisher veröffentlichte Ansätze von Ubicom Schnittstellen stellen sich als konkrete Entwürfe und Szenarien dar. Deren Nützlichkeit wird, wenn überhaupt, statistisch über Versuche mit Testpersonen nachgewiesen; die Qualität des Designs erfolgt intuitiv durch den Ersteller der Oberflächen und hängt deshalb stark von dessen Fähigkeiten ab. Bisher gibt es keinerlei Methoden oder auch nur Vorschläge von Methoden, die den Entwurf solcher Ubicom Schnittstellen unterstützen würden. Diese Lücke zwischen HCI Theorie und der Umsetzung in Ubicom Schnittstellen soll Gegenstand des Abschnitts 5 sein.

3 Bisherige Ansätze

UbiCom definiert nicht nur eine neue Art von Computern, sondern folglich auch eine neue Art, mit Rechnern zu interagieren. Ubiquitous Computing selbst stellt deshalb per Definition [Weis91] ein neues Paradigma für die Mensch-Computer Schnittstelle dar. Bisherige Paradigmen stellen die Gestaltung virtueller Abbildungen von Objekten und Funktionen in den Vordergrund. Ubiquitous Computing dagegen versucht diese virtuellen Abbildungen und Funktionen in die reale Welt zu integrieren (embodied virtuality), möglichst ohne dem Benutzer das Gefühl zu geben, einen Computer oder gar eine Maschine zu bedienen (Calm Computers).

Tangible Objects [IsUl97] und Wearable Computer [MaFe96, FeMH97] stellen weitere revolutionäre Schnittstellen dar. In diesem Zusammenhang wurden erste Prototypen von Systemen entwickelt, welche sowohl aktive, d.h. mit Computern verbundene E/A Geräte, als auch passive, d.h. nicht mit Computern direkt verbundene E/A Geräte besitzen. Ein herausragendes Beispiel ist das metaDESK Experiment [UIIs97], in welchem ein Tisch mit eingebautem tischgroßem Display und die dort befindlichen Gegenstände wie Klötze oder eine passive Lupe zur Interaktion mit dem Computer benutzt werden.

Diese neuen Schnittstellen bzw. die Paradigmen fordern neue Konzepte in der Gestaltung von Benutzerschnittstellen, stellen aber auch die Basis für diese dar. Folgende wichtige Prinzipien lassen sich aus den Arbeiten, die direkt oder indirekt dem Thema Ubiquitous Computing zuzuordnen sind, ableiten:

- Kommunikation wird “anytime anywhere” möglich sein. Die rasante Entwicklung im Bereich Telekommunikation scheint diese These zu bestätigen.
- Es wird viele unterschiedliche Geräte und Interaktionspunkte pro Benutzer geben. Interaktionspunkte werden vom Benutzer nicht mehr bewußt wahrgenommen, sondern gehen in der Umgebung auf.
- Die Interaktion mit diesen Geräten wird auf natürlichere Weise als bisher erfolgen. Darunter darf man sich nicht nur die Steuerung per Sprache vorstellen. In vielen Fällen ist direkte Manipulation (Bewegen von Gegenständen) effizienter als Manipulationen durch gesprochene Kommandos oder Manipulation virtueller Abbilder. Zudem sollen auch nicht direkt ausgedrückte Wünsche des Benutzers erfaßt werden bzw. der Benutzer soll “unterbewußt” mit Informationen versorgt werden.

4 Verfügbare Techniken

Neben den oben erwähnten Ideen und Labor-Prototypen gibt es inzwischen einige Technologien, die dazu verwendet werden können, UbiCom Szenarien zu implementieren. Von besonderem Interesse ist das Gebiet Handheld Computing und Personal Digital Assistants (PDA). Diese Geräte, bzw. eine Vielzahl dieser Geräte bilden das wichtigste Element, um derzeit UbiCom Szenarien zu installieren. Damit diese Geräte in Kommunikation mit anderen Geräten treten können, werden die Techniken, die im Bereich Mobile Computing verfügbar sind, benutzt. Andere verfügbare Techniken, welche für

Ubicom Szenarien eingesetzt werden können, kommen aus diversen unterschiedlichen Bereichen: Dazu gehören active Batches [WaHo92], Web-basierte set-top Boxen, die gleichzeitig Internet und TV Anschluß beinhalten, wie sie zum Beispiel von Nokia angeboten werden, Liveboards von Xerox zur Konferenzunterstützung und als "intelligente" Wand, sowie moderne Display-Technologien für Flachbildschirme.

Ausgehend von existierenden Geräten, die ihren Ursprung im Bereich von Ubiquitous Computing haben, lassen sich zwei Anwendungsszenarien ausmachen:

- Die synchrone Benutzung von Geräten d.h. zeitgleiche Einbeziehung mehrerer Geräte zur Erledigung einer Aufgabe. Als Beispiel ist hier die Unterstützung eines Meetings mit Hilfe von Liveboard und PDA als Steuerungsgerät zu nennen.
- Die asynchrone Benutzung von Geräten d.h. die zeitlich versetzte Benutzung von Geräten zur Erledigung einer Aufgabe. Ein populäres Beispiel ist der Synchronisationsmechanismus für Dokumente bei PDA's.

Die zu entwickelnde Designmethode sollte beide Szenarienklassen abdecken können.

5 Design

Um die Lücke zwischen den allgemeinen Theorien und den konkreten Ubicom Szenarien zu füllen, werden Methoden benötigt, um für bestimmte Aufgabenstellungen das entsprechende Design entwerfen zu können.

Mit diesen Entwurfsmethoden müssen die in Ubiquitous Computing eingeführten Objekte und Manipulationsmethoden darstellbar sein. Notwendig ist Wissen über die zur Verfügung stehenden Objekte, deren Ein/Ausgabe und die Manipulationsmöglichkeiten. Dabei ist Bernsens [Bern93] Taxonomie hilfreich, welche eine allgemeine Klassifikation für Modalitäten enthält. Weitere Dimensionen einer Klassifikation umfassen technologische und ergonomische Aspekte sowie soziale Faktoren, wie zum Beispiel die Akzeptanz eines Ein-/Ausgabeobjektes; eine Ausführung der Klassifikation soll aber nicht Gegenstand dieses Artikels sein.

Eine Eigenschaft von Ubicom Umgebungen ist, daß sich das Aussehen, d.h. die zur Verfügung stehende E/A Schnittstelle, wechseln kann. Mit bisherigen Designmethoden kann ein solches Verhalten nicht modelliert werden. Deshalb soll in den folgenden Abschnitten ein Vorschlag vorgestellt werden, der solche Modellierungen erlaubt; damit lassen sich beide Anwendungsszenarien aus Abschnitt 4 abdecken.

5.1 Vorschlag eines Entwurfsablaufs

Der Entwurf einer Ubicom Schnittstelle beginnt nach herkömmlichen Methoden mit der Analyse der Aufgabenstellung, d.h. mit Hilfe von Prinzipien und Leitlinien wird die Idee der Aufgabe in eine Aufgabenbeschreibung überführt. Diese Aufgabenbeschreibung hat dann allerdings oft schon eine Form, die bestimmte E/A Schnittstellen präjudiziert. Ubicom Schnittstellen sollten flexibel sein, eine solche Vorbestimmung würde deshalb den Entwurfsprozeß behindern; deshalb muß bei diesem Schritt besonders auf diesen

Punkt geachtet werden. In einem weiteren Schritt sind zunächst *optimale Schnittstellen* zu entwerfen. Die optimale Schnittstelle beinhaltet eine Repräsentation der Objekte der realen Welt; diese Objekte sollen im folgenden *Assoziationen* genannt werden. Optimale Szenarien können, müssen aber nicht aus verschiedenen Entwürfen bestehen. Da dieser Prozeß große Kreativität verlangt, ist dies durch geeignete Methoden und Regeln

beziehungsweise durch entsprechende automatisierte Werkzeuge zu unterstützen. In einem weiteren Schritt werden die *minimalen Schnittstellen* für die Aufgabe spezifiziert, wobei die gleichen Methoden und Werkzeuge wie beim Design optimaler Szenarien (unter anderen Voraussetzungen) verwendet werden können. Günstigerweise werden jetzt Funktionen gefunden, welche die optimalen in die minimalen Szenarien überführen. Das Problem kann aber in der Praxis oft nicht auf diese Weise gelöst werden, da es Szenarien geben kann, welche zwar besser als das minimale Szenario sind, aber neue Assoziationen enthalten. So ist es manchmal notwendig, Szenarien, die zwischen dem optimalen und minimalen angesiedelt sind, zu entwerfen. Diese sollen im folgenden *intermediäre Szenarien* genannt werden. Zum Schluß muß die *Transformationsfunktion*, welche die Szenarien ineinander überführt, gefunden werden. Für diese ist eine Beschreibungssprache notwendig, die eine automatische Überführung ermöglicht. Der gesamte Ablauf ist nochmals in Abbildung 1 zusammengefaßt.

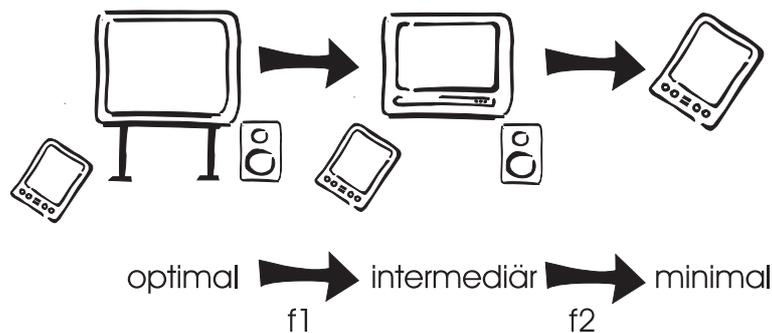


Abbildung 1: Designablauf

6 Vorgang der Modellierung am Beispiel

Ein abgewandeltes Beispiel aus [UIIs97] soll den Entwurf einer UI Modellierung zeigen. Problemstellung ist, mit Unterstützung durch den Computer eine einfache städtebauliche Planung intuitiv und effektiv zu ermöglichen. Die Ausgangsbasis ist in Abbildung 2 Punkt a) gezeigt: Ein Viertel (hier vereinfacht durch ein Haus und einen Baum dargestellt) sowie eine kleine Dokumentation mit den wichtigsten Merkmalen zu diesem Viertel, zum Beispiel den gesetzlichen Vorschriften laut Bebauungsplan. Durch den Einsatz von entsprechenden kognitiven Techniken käme man nun zu dem in Abbildung 2 b) gezeigten optimalen Szenario. Dieses besteht aus einem Handheld-Computer als Assoziation für das Papier, ein in den Schreibtisch eingelassenes Display von ausreichender Größe, welches zudem durch Sensoren weitere auf dem Tisch befindliche Objekte erkennen kann. Außerdem werden ein Würfel und eine Pyramide als passive Objekte benutzt. Die Abbildung der realen Welt erfolgt dabei so, daß der Grundriß des Viertels und einige nicht zur Disposition stehende Objekte wie zum Beispiel die Straße vom Display dargestellt werden; die Assoziation für diese Objekte ist also die

entsprechende Grafik auf dem Display. Das Papier wird vom Handheld Computer, der Baum durch die Pyramide und das Haus durch einen Würfel assoziiert. Diese Abbildung ist deshalb als optimal zu bezeichnen, da die Manipulation (also Umplazierung) der Objekte auf gewohnte, "natürliche" Weise erfolgen kann. Ein minimales Szenario

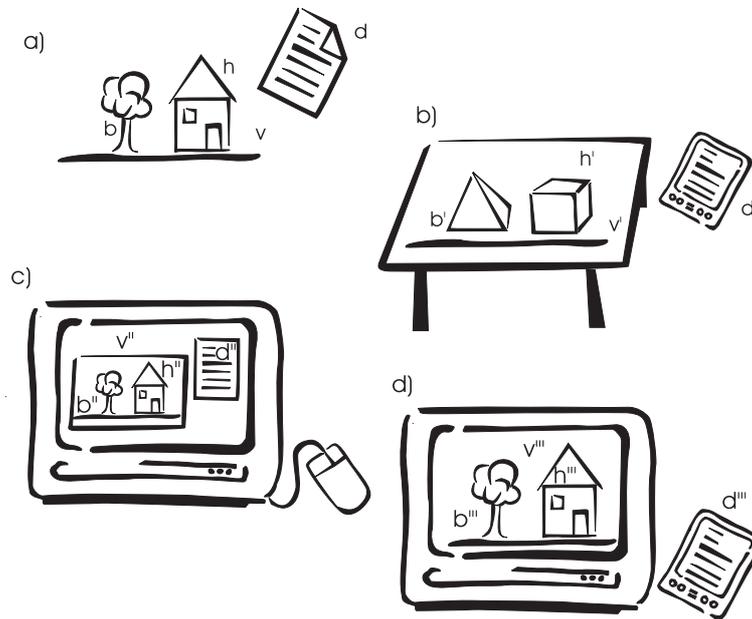


Abbildung 2: Beispiel eines Designs

wäre die Assoziation des Viertels mit einer in einem Fenster dargestellten Grafik auf einem Ausgabebildschirm einer Workstation, sowie die Manipulation dieser Objekte via Maus und Tastatur (Abbildung 2 c)). Gleiches gilt für die dazugehörige Dokumentation; auch sie kann auf ein Fenster in einem Bildschirm abgebildet werden. Ein intermediäres Szenario wäre wie in Abbildung 2 d) angegeben. Dabei könnte das Viertel den ganzen Raum des Bildschirms einnehmen und so eine übersichtlichere Repräsentation des Szenarios ermöglichen. In diesem simplen Szenario können die Funktionen für die Überführung des optimalen in das intermediäre bzw. das minimale Szenario einfach angegeben werden. Es beschränkt sich dabei auf die Umleitung des Displays auf ein Fenster, sowie die Visualisierung der im optimalen Szenario intern als Koordinatenform vorliegenden Objekte Baum und Haus. Analog ist die Überführung der Dokumentation zu betrachten. Folgende vereinfachte Beschreibung drückt die Funktion aus:

```

Wenn nicht (vorhanden Displaytisch)
    dann { benutze Bildschirm als Displaytisch und Objektanzeige}
    Wenn (vorhanden PDA) dann { benutze Vollbildmodus }
    sonst { benutze Fenstermodus }
Wenn nicht (vorhanden PDA)
    dann { benutze Bildschirm }
  
```

7 Regeln

Oben wurde ein intuitiver Entwurf einer E/A Schnittstelle gezeigt. Dieser ist wie schon erwähnt durch Prinzipien und Leitlinien zu unterstützen. Prinzipien sind aber zuvor

in Designregeln zu überführen, um konkret den Entwurf unterstützen zu können. Zwei Prinzipien aus [Pree94] scheinen besonders wichtig; diese wurden in zwei Regeln überführt, welche die Basis für obiges Design bilden. Die Prinzipien sind:

“ensure use of understanding” und

“allow input flexibility”

und die daraus abgeleiteten Regeln:

“wähle eine optimal natürliche Assoziation für jedes reale Objekt”

“wähle eine optimal natürliche Manipulation für die Assoziationen”

Neben diesen Grundregeln lassen sich aus den diversen Prinzipien, die in der HCI aufgestellt wurden, weitere Regeln ableiten.

8 Praktische Gesichtspunkte und Ausblick

Zum Abschluß sollen noch einige eher praktische Gesichtspunkte erwähnt werden, welche auf die Umsetzung einer Anwendung als Ganzes, und nicht nur auf den UI Aspekt zielt.

Um Benutzerschnittstellen wie vorgestellt realisieren zu können, müssen Anwendungen mit verschiedenen Gerätetypen umgehen können. Die Frage, ob und wie Anwendungen so erstellt werden können, daß sie sich auf verschiedenen Geräten mit sehr unterschiedlichen Eigenschaften betreiben lassen, ist oft diskutiert worden. Viele Ansätze stellen das Web als Plattform zur Lösung dieses Problems in den Vordergrund: Web-Anwendungen werden dabei für die verschiedenen Geräte umgesetzt. Mit Hilfe von Java und ActiveX lassen sich nun auch in Web-Anwendungen manipulierbare Objekte erzeugen. Insofern können diese Technologien tatsächlich als Basis für die Erstellung von UbiCom Benutzerschnittstellen dienen, zumal sich das Web als Anwendungsplattform inzwischen für die Softwareerstellung mit Hilfe entsprechender Werkzeuge wie WebComposition [GeWG97] nutzen läßt.

Es bleibt aber weiterhin das Problem des Entwurfs des UbiCom Interface Designs bestehen. Die bisher angedeuteten Entwurfsregeln und Methoden sind deshalb weiter zu detaillieren. Werkzeuge, die einen solchen Entwurf unterstützen, können dann aufbauend auf diesen Methoden entworfen werden.

Literatur

- [Bern93] Niels Ole Bernsen. Modality Theory: Supporting Multimodal Interface Design. *Proceedings of the ERCIM Workshop*, 1993, S. 13–23.
- [FeMH97] Steven Feiner, Blair MacIntyre und Tobias Höllerer. A Touring Machine: Prototyping 3D Mobile Augmented Reality Systems for Exploring the Urban Environment. *Proceedings of the International Symposium on Wearable Computing*, 1997, S. 74–81.

- [GeWG97] Hans-W. Gellersen, Robert Wicke und Martin Gaedke. WebComposition: An Object-Oriented Support System for the Web Engineering Lifecycle. *Proc. of the 6th International World-Wide Web Conference (WWW6)*, April 7–12 1997.
- [Hef92] Bill Hefley. *Cirricula for Human-Computer Interaction*. ACM. 1992.
- [IsU97] Hiroshi Ishii und Brygg Ullmer. Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms. *CHI*, 1997.
- [MaFe96] Blair MacIntyre und Steven Feiner. Future Multimedia User Interfaces. *Multimedia Systems* (4), 1996, S. 250–268.
- [Pree94] Jenny Preece. *Human-Computer Interaction*. Addison-Wesley. 1994.
- [UIs97] Brygg Ullmer und Hiroshi Ishii. The metaDESK: Models and Prototypes for Tangible User Interfaces. *Proceedings of UIST*, 1997, S. 14–17.
- [WaHo92] Roy Want und Andy Hopper. The active badge location system. *ACM Transactions on Information Systems* 10(1), January 1992, S. 91–102.
- [Weis91] Mark Weiser. The Computer for the 21st Century. *Scientific American* Band 9, 1991.

Dienstqualitätsbasiertes Routing

Roland Bless

Kurzfassung

Dienstqualitätsbasiertes Routing (QoS-Routing) ist eine fundamentale Aufgabe des Ressourcenmanagements im Netzwerk zur Erbringung garantierter Dienste. Seine Aufgabe ist es, einen Pfad im Netzwerk zu finden, der bestimmten Randbedingungen in Form von Grenzwerten für einen oder mehrere Dienstqualitätsparameter wie etwa Bandbreite oder Ende-zu-Ende-Verzögerung genügt. Einige QoS-Routing-Probleme sind praktisch nicht berechenbar. Dienstqualitätsbasiertes Multicast-Routing ist aufgrund großer, dynamischer und heterogener Empfängermengen noch komplexer als im Unicast-Fall und oft nur durch die Anwendung von Heuristiken noch handhabbar. In diesem Beitrag wird eine Einführung in die Problematik des QoS-Routings gegeben.

1 Einleitung

Um die verschiedenen Anforderungen von heutigen multimedialen und von zukünftig aufkommenden fortgeschrittenen Anwendungen unterstützen zu können, müssen diensteintegrierende Netze auch unterschiedliche Dienstypen zur Verfügung stellen. Dazu zählen insbesondere Dienste, die Garantien für die Dienstqualität (Quality of Service – QoS) von Endsystem-zu-Endsystem bieten, wie z.B. die Einhaltung eines Mindestdurchsatzes oder einer maximalen Ende-zu-Ende-Verzögerung für interaktive Anwendungsszenarien. Sind die Anforderungen an die gewünschte Dienstqualität formuliert, sorgt *dienstqualitätsbasiertes Routing (QoS-Routing)* dafür, daß zwischen den Kommunikationspartnern ein Weg im Netz gefunden wird, der solchen Randbedingungen genügt und insgesamt zu einer effizienten Gesamtnutzung der Netzwerkressourcen führt.

Wie beim traditionellen Routing muß auch beim QoS-Routing die Konnektivität der Netzwerkelemente (Topologie) bekannt sein, allerdings wird die Wegewahl zusätzlich anhand zweier Kriterien getroffen: die *momentane Verfügbarkeit der Ressourcen* auf den Verbindungswegen („Links“) sowie die spezifizierten *Anforderungen an die Dienstqualität* müssen ebenfalls berücksichtigt werden. Dazu muß zum einen Statusinformation über die (noch) verfügbaren Ressourcen gesammelt werden, wie beispielsweise die verbleibende Kapazität („Bandbreite“) eines Verbindungsweges zum Nachbarknoten, zum anderen müssen die Dienstqualitätsanforderungen propagiert werden. Daher ist die Komplexität des QoS-Routings sowohl für die Unicast-Kommunikation als auch für die Gruppenkommunikation wesentlich höher als beim traditionellen Routing. Bereits

für mehrere unabhängige additive oder multiplikative Metriken ist das Problem der Wegsuche unter den durch die Anforderungen definierten Bedingungen im Fall eines Unicasts *NP-vollständig* [WaCr96]. Im Falle des Multicast-Routings kommt noch die Dynamik und Heterogenität der Empfängergruppe hinzu.

2 Anforderungen an ein qualitätsbasiertes Routing

Allgemein lassen sich folgende Anforderungen an ein qualitätsbasiertes Routing (bzw. den zugehörigen Algorithmus) formulieren [CNRS98, WaCr96, DiDC97]:

- Bestimmung eines möglichen Weges („Feasible Path“) — Dieser Pfad muß die vorgegebenen Anforderungen erfüllen. Der Routing-Algorithmus sollte möglicherweise zunächst einen suboptimalen Pfad (z.B. länger als notwendig) finden, falls die derzeitige Ressourcenbelegung keinen besseren Weg zuläßt. Allerdings gilt es hier vorsichtig zwischen einer insgesamt effizienten Ressourcenbelegung (global für das gesamte Netzwerk betrachtet) und einer höheren Ablehnungsrate eines angeforderten Dienstes abzuwägen.
- Dynamik und Robustheit — Der Algorithmus sollte auf Ausfälle von Verbindungsstrecken und Netzwerkelementen möglichst schnell und ohne Beeinträchtigung der zu dieser Zeit bestehenden Dienste reagieren können. Werden Ressourcen entlang eines besseren Pfades wieder verfügbar, sollte unter Umständen die bereits etablierte suboptimale Route zugunsten einer effizienteren Netzwerkauslastung geändert werden. Es sollten aber häufige Änderungen der Route vermieden werden, da sie vermutlich Schwankungen in der Dienstqualität auslösen oder verstärken (Delay-Jitter). Das „Feststecken“ der Route für eine längere Zeit nennt sich „Route-Pinning“.
- Optimierung der Ressourcennutzung und Schleifenfreiheit — Der gefundene Pfad sollte die Netzwerkressourcen möglichst effizient und gleichmäßig auslasten. Konzentrationen des Verkehrs auf einzelne Verbindungsstrecken sind zu vermeiden. Der „Gesamtdurchsatz“ im Netz sollte erhöht werden und der Pfad sollte schleifenfrei sein.
- Skalierbarkeit — Der Algorithmus sollte auch für eine große Anzahl von Zwischen- und Endsystemen, und im Fall der Gruppenkommunikation, für eine große Empfängeranzahl gute Leistung zeigen.
- Minimierung des Routing-Overheads — Beispielsweise sollte der Austausch von Zustandsinformation zwischen Routern auf ein Minimum reduziert werden, ebenso wie die benötigte Zeit zur Berechnung des Pfades und die Menge der in den Routern zu verwaltenden Zustandsinformation. Hierbei spielt auch die Granularität der Routing-Entscheidung eine Rolle: wird die Entscheidung in Abhängigkeit vom Ziel, von Quelle und Ziel, oder anhand einzelner Datenströme getroffen?

Ein „optimaler Pfad“ in dem Sinne, daß alle Werte der Dienstqualitätsparameter optimal sind, muß nicht existieren. Daher ist es denkbar, daß anwendungsspezifische Prioritätenregelungen in solchen Fällen zum Einsatz kommen.

3 Grundlagen und Lösungsansätze

Im folgenden werden für QoS-Routing im Unicast-Fall kurz einige Betrachtungen zur Zeitkomplexität des Problems geschildert und bereits evaluierte Algorithmen vorgestellt.

Eine Reduktion des QoS-Routing-Problems (d.h. einen Weg zu finden, der mehreren Randbedingungen genügt) auf eine einzige Metrik ist nach [WaCr96] nicht so einfach möglich. Zum einen ist dann nicht genügend Information vorhanden, um zu beurteilen, ob sämtliche Parameterbedingungen erfüllt sind (die Abbildung ist i.a. nicht injektiv, da ein Wert für zwei oder mehr Parameterkombinationen stehen kann), zum anderen ist dadurch die Bildung von Kompositionsregeln äußerst schwierig. Man bezeichnet eine Metrik $d(i, j)$ (sei (i, j) Verbindungsstrecke) als *additiv*, wenn für den Pfad $p = (i, j, k, \dots, s, t)$ gilt $d(p) = d(i, j) + d(j, k) + \dots + d(s, t)$. Die Metrik heißt entsprechend *multiplikativ*, wenn gilt $d(p) = d(i, j) \cdot d(j, k) \cdot \dots \cdot d(s, t)$, und sie heißt *konkav*, wenn gilt $d(p) = \min(d(i, j), d(j, k), \dots, d(s, t))$. Hat man nun eine additive Metrik d_1 (z.B. Ende-zu-Ende-Verzögerung) und eine konkave Metrik d_2 in einer einzigen Metrik $f(i, j) := d_2(i, j)/d_1(i, j)$ zusammengefaßt, ist eine Kompositionsregel $f(i, k) = G(f(i, j), f(j, k))$ nicht direkt abzuleiten.

Wang und Crowcroft weisen in [WaCr96] die NP-Vollständigkeit für das Problem nach, einen Pfad zu finden, der alle Bedingungen für zwei oder mehrere additive oder multiplikative (voneinander unabhängige) Metriken erfüllen soll. Daraus läßt sich beispielsweise folgern, daß eine lösbare Parameterkombination den Parameter „Bandbreite“ und *einen* weiteren aus Ende-zu-Ende-Verzögerung, Verzögerungsschwankung, Verlustwahrscheinlichkeit umfaßt. Basierend auf diesem Resultat präsentieren die Autoren einen Algorithmus, der zunächst einen möglichen Pfad mit maximaler Restbandbreite („Bottleneck Bandwidth“) findet. Sollten mehrere solche Pfade existieren, wird derjenige mit kleinster Ende-zu-Ende-Verzögerung gewählt. Ein solcher Pfad heißt „*Shortest-widest Path*“ und ist frei von Schleifen, da die Bedingung für die Verzögerungsmetrik diese verhindert. Die Zeitkomplexität des Verfahrens ist dabei nicht höher als die eines herkömmlichen Shortest-Path-Algorithmus.

Das vorige Ergebnis wird in [MaSt97] relativiert: viele QoS-Parameter sind *nicht* unabhängig voneinander, so beispielsweise Ende-zu-Ende-Verzögerung und Bandbreite. Den Einsatz von Zuteilungsverfahren in Routern nach einem dem Weighted-Fair-Queueing verwandten Verfahren vorausgesetzt, wird ein modifizierter Bellman-Ford-Algorithmus konstruiert, der einen Pfad zu vorgegebener Ende-zu-Ende-Verzögerung und Verzögerungsschwankung sowie Pufferanzahl berechnet. Die zu reservierende Bandbreite kann dabei entweder vorgegeben oder vom Algorithmus bestimmt werden.

In [MaSt97] findet sich außerdem ein Vergleich für verschiedene Routing-Verfahren, die einen Pfad mit garantierter Bandbreite berechnen. Die Routing-Verfahren werden mittels einer Simulation und der daraus gewonnenen sogenannten *Bandwidth Blocking Rate* bewertet. Die bisher traditionell zum Vergleich verwendete *Call Blocking Rate*, d.h. das Verhältnis von Anzahl der abgewiesenen Verbindungsaufbauwünsche zur Anzahl der Verbindungsaufbauwünsche insgesamt, ist nämlich nicht sehr aussagekräftig, wenn unterschiedliche Mengen an Bandbreite angefordert werden. Die *Bandwidth Blocking Rate* (BBR) ist wie folgt definiert: $BBR := \sum_{i \in \mathcal{B}} \text{bandwidth}(i) / \sum_{i \in \mathcal{R}} \text{bandwidth}(i)$, wobei \mathcal{B} die Menge der abgewiesenen (blockierten) Verbindungsaufbauversuche bezeichnet und

\mathcal{R} die Menge aller Verbindungsaufbauversuche beschreibt. Neben dem oben erwähnten *Shortest-widest Path* werden noch *Widest-shortest Path*, *Dynamic-alternative Path* und *Shortest-Dist(P,1)* untersucht. *Widest-shortest Path* wählt zunächst einen möglichen Pfad mit minimaler Anzahl der Verbindungsstrecken (Hop-Count). Existieren mehrere solche Pfade, wird derjenige mit maximaler Restbandbreite gewählt. *Dynamic-alternative Path* versucht einen breitesten Pfad mit minimalem Hop-Count zu finden. Ist dies nicht möglich, wird ein Pfad gesucht, der um einen Hop länger ist. *Shortest-Dist(P,1)* sucht einen Pfad mit minimaler Distanz $\sum_{i=1}^k 1/R_i$, wobei R_i die verfügbare Bandbreite eines Links i entlang des Pfades bezeichnet.

Im gewählten Simulationsszenario zeigen alle Algorithmen, bis auf *Shortest-widest Path* sehr ähnliche Resultate bezüglich der BBR. Lösungen des *Shortest-widest Path* schneiden etwas schlechter ab, weil er dazu tendiert, einen längeren Pfad zu wählen, und dadurch mehr Ressourcen benötigt.

3.1 Qualitätsbasiertes Routing im Internet

Es existiert zur Zeit noch keine QoS-basierte Routing-Architektur für das Internet. Beispielsweise wurde zwar mit RSVP [ZBHJ97] bereits ein Signalisierungsprotokoll für die Ressourcenreservierung definiert, jedoch setzt dieses Protokoll bereits existierende Pfade für die Datenströme voraus, d.h. eine Routing-Entscheidung erfolgt völlig entkoppelt von der Ressourcenanforderung und Ressourcenreservierung, woraus eine suboptimale Ausnutzung der Ressourcen resultiert. Bisher existiert nur ein Rahmenwerk mit dem Status eines Internet-Drafts [CNRS98], sowie eine Erweiterung des OSPF-Verfahrens zu QOSPF (Quality of Service Extensions to OSPF) [ZSSC97].

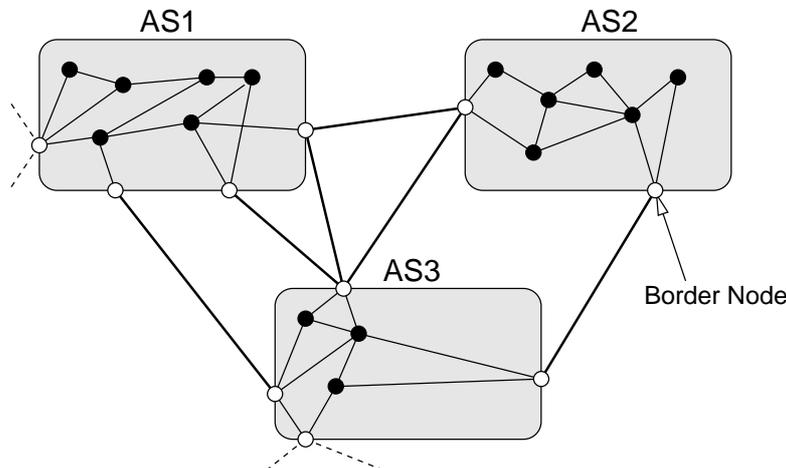


Abbildung 1: Interdomain-Routing

Unter Beibehaltung der traditionellen Einteilung des Routings im Internet in Intradomain- und Interdomain-Routing lassen sich sehr unterschiedliche QoS-Routing-Verfahren in den *Autonomen Systemen (AS)* – also im Intradomain-Bereich – implementieren, jedoch muß der Austausch der Routing-Information zwischen den AS vereinheitlicht werden, um eine Interaktion und Kooperation unter den Grenzknoten (Border Nodes) zu ermöglichen (siehe Abbildung 1). Der Vorteil des hierarchischen Routings liegt in der besseren Skalierbarkeit des Routings, die unter anderem durch *Zustandsaggregation*

der AS erreicht wird. Diese Aggregation dient dazu, die Menge der Zustandsinformation gering zu halten und kommt damit auch den Prinzipien eines AS entgegen, welches keine detaillierte Information über seine Topologie und Ressourcenverfügbarkeit preisgeben will. Weiterhin sollte die Häufigkeit des Austauschs von Routinginformationen zwischen den AS vermindert werden, was allerdings im Widerspruch zur sich häufig ändernden Ressourcenbelegung innerhalb der AS steht. Daher sollte die nach außen hin sichtbare Information über die innerhalb eines AS verfügbare Dienstgüte nur relativ statisch sein, die von der implementierten Topologie und Kapazität abhängt, anstatt von Fluktuationen der momentanen Verkehrslast durch das AS.

Ein möglicher Ansatz ist daher, gewisse Dienstqualitäten für aggregierten Transitverkehr von vorneherein vorzusehen (d.h. beim AS-Design) und bereitzustellen [CNR98]. Die Planung für diesen „virtuellen Transitpfad“ kann dabei auf der zu erwartenden Verkehrscharakteristik und den QoS-Anforderungen von benachbarten AS basieren. Verschiedene Verkehrsströme können durch den Grenzknoten am Eingang mittels Klassifikation (anhand des Zielknotens am Ausgang und den QoS-Anforderungen) zu einem einzelnen Strom zusammengefaßt werden, welcher vollständig zum Ausgangsknoten fließt. Der Eingangsknoten überprüft dabei die Ressourcenanforderungen eines jeden Transitstroms, um zu bestimmen, ob der aggregierte Strom von einem Nachbarn die vereinbarten Grenzen nicht überschreitet.

4 Qualitätsbasiertes Multicast-Routing

Im Falle des qualitätsbasierten Multicast-Routings kommen gegenüber einem QoS-basierten Unicast-Routing noch folgende Aspekte hinzu, die berücksichtigt werden müssen:

- Skalierbarkeit für große Gruppen mit dynamischer Mitgliedschaft
- Empfänger-initiierte, heterogene Reservierungen
- Robustheit bei topologischen Veränderungen
- Unterstützung geteilter, d.h. gemeinsam genutzter, Ressourcenreservierungen

Allein schon aufgrund der Tatsache, daß in den Routern Information vorgehalten werden muß, ob und welche Systeme Mitglieder einer Multicast-Gruppe sind, ergibt sich die Notwendigkeit, flußbasierte Zustandsinformation zu speichern, was wiederum einen negativen Einfluß auf die Skalierbarkeit ausübt.

In [ZSSC97] wird die Erweiterung des im Internet verwendeten Link-State-Routing-Protokolls OSPF vorgeschlagen. Der Pfad für einen Datenstrom wird basierend auf der Topologie, der Netzwerkressourceninformation (momentane Ressourcenbelegung) und den Dienstqualitätsanforderungen berechnet. Die Routen werden jeweils für ein Adreßpaar (*Quelle, Ziel*) gebildet, wobei die Zieladresse auch eine Multicast-Gruppenadresse sein kann. Daraus folgt aber auch, daß sämtliche Datenströme für ein Adreßpaar (*Quelle, Ziel*) den selben Weg durch das Netzwerk nehmen. Die Anforderungen für verschiedene Datenströme zwischen demselben Adreßpaar können aber durchaus sehr unterschiedlich sein, so daß entweder keine passende Route gefunden wird, obwohl dies

möglich wäre, oder Ressourcen ineffizient belegt werden. Weiterhin wird jeweils nur ein QoS-Parameter berücksichtigt: entweder Bandbreite oder Verzögerung. Um dem Skalierbarkeitsproblem von QOSPF, welches durch die zusätzlich ausgetauschte Information, den erhöhten Speicherplatzbedarf und die gesteigerter Rechenzeit in großen Domains und bei vielen Datenströmen entsteht, entgegenzuwirken, wird ein explizites Routing (ER) vorgeschlagen: hierbei berechnet nur der an der Quelle gelegene Router den Weg und verteilt die Routing-Information weiter „abwärts“ entlang des Pfads an die nachfolgenden Router.

5 Zusammenfassung

Dienstqualitätsbasiertes Routing ist ein außerordentlich komplexes Problem. Für einige Fälle sind Lösungen praktisch nicht berechenbar (NP-vollständig). Somit helfen teilweise nur Heuristiken weiter. Multicast-Szenarien fügen eine weitere Komplexitätsstufe hinzu. Hierfür müssen noch effiziente und zuverlässige qualitätsbasierte Routing-Verfahren entwickelt werden, ebenso wie für den Interdomain-Bereich.

Literatur

- [CNRS98] Eric Crawley, Raj Nair, Bala Rajagopalan und Hal Sandick. A Framework for QoS-Based Routing in the Internet. Internet Draft draft-ietf-qos-framework-03, 3 1998. Version vom 2. März 1998.
- [DiDC97] Cristophe Diot, Walid Dabbous und Jon Crowcroft. Multipoint Communication: A Survey of Protocols, Functions, and Mechanisms. *IEEE Journal on Selected Areas in Communications* 15(3), April 1997, S. 277–290.
- [MaSt97] Qingming Ma und Peter Steenkiste. Quality-of-Service Routing for Traffic with Performance Guarantees. In Andrew Campbell und Klara Nahrstedt (Hrsg.), *Building QoS into Distributed Systems*, London, 1997. Chapman & Hall. Proceedings of the 5th IFIP International Workshop on Quality of Service, 21.–23. Mai 1997, New York, USA.
- [WaCr96] Zheng Wang und Jon Crowcroft. Quality-of-Service Routing for Supporting Multimedia Applications. *IEEE Journal on Selected Areas in Communications* 14(7), September 1996, S. 1228–1234.
- [ZBHJ97] Lixia Zhang, Steve Berson, Shai Herzog und Sugih Jamin. Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification. Request for Comments (RFC) 2205, 9 1997.
- [ZSSC97] Zhaohui Zhang, Cheryl Sanchez, Bill Salkewicz und Eric S. Crawley. Quality of Service Extensions to OSPF or Quality Of Service Path First Routing (QOSPF). Internet Draft draft-zhang-qos-ospf-01, 9 1997.

Drahtloses ATM – ATM-Architektur mit Unterstützung mobiler Endsysteme

Elmar Dorner

Kurzfassung

Dieser Bericht beschreibt das Konzept und die Architektur von drahtlosem ATM für den Einsatz von mobilen multimedialen Anwendungen. Die Fortschritte im Entwurf und der Entwicklung von tragbaren Geräten, verbunden mit der Entwicklung des *Asynchronen Transfer Modus* (ATM) erlauben eine neue Generation von Geräten mit drahtloser Kommunikationsanbindung. ATM wurde entworfen, um multimediale Anwendungen mit hohen Bandbreitenansprüchen zu ermöglichen. Außerdem sollte ATM verschiedene Verkehrsströme integrieren, kosteneffizient arbeiten und flexible Datenübertragungen anbieten. Berücksichtigt man, daß heutige Netzwerkanwendungen auf Multimediadienste und umfassenden Informationszugriff abzielen, so kann man ATM als einen möglichen Kandidaten betrachten der solche Dienste – erweitert um die Möglichkeit tragbare und mobile Geräte mit einem drahtlosen Kommunikationssystem zu nutzen – bietet. Aus diesem Grund hat das ATM-Forum 1995 die *Wireless ATM Task-Force* gegründet, um eine Standardisierung voranzutreiben, nicht zuletzt auf Drängen von Endsystembenutzern, die die Vorzüge von ATM im Festnetzbereich bereits kennengelernt haben und sie auch im drahtlosen Umfeld nutzen wollen. Mit ersten Standards kann jedoch nicht vor Anfang 1999 gerechnet werden, so daß noch ein weites Forschungsfeld vorhanden ist.

1 Einleitung

Drahtloses ATM (*Wireless ATM*; WATM) stellt vereinfacht gesagt eine drahtlose Erweiterung der traditionellen ATM-Netzwerke, wie in Abbildung 1 gezeigt, dar. Dabei greift WATM auf Funkzellen, vergleichbar mit der GSM Struktur, zurück. Innerhalb jeder Zelle existiert eine Basisstation, die als Zugangspunkt zum ATM-Netzwerk dient. Mehrere mobile Endgeräte nutzen Verbindungen zur Basisstation und teilen sich somit das Funkmedium an der Basisstation. Es gibt mehrere technische Besonderheiten, die bei der Entwicklung von WATM berücksichtigt werden müssen. Dies liegt grundsätzlich darin begründet, daß die Spezifikationen für ATM als Grundlage von Glasfaserübertragungstrecken ausgeht. Diese zeichnen sich durch hohe Bandbreiten und geringe Fehlerraten aus. Im Gegensatz dazu, ist die Übertragungsgeschwindigkeit heutiger drahtloser Systeme eher niedrig und der drahtlose Link ist teilweise – bedingt durch sogenanntes Fading – mit hohen Bitfehlerraten versehen. Um dieses Problem zu lösen führt man eine Zwischenschicht in den Protokollturm ein: Die *Data Link Control*

(DLC) Schicht, die dann für die Fehlerbehandlung im drahtlosen Umfeld verantwortlich ist. Ein weiterer Unterschied liegt in der beschränkten Bandbreite, bedingt durch das beschränkt zur Verfügung stehende Frequenzspektrum im Funkbereich. Anders als im Festnetz läßt sich diese Ressource nicht durch Hinzufügen weiterer Komponenten erhöhen. Diese Beschränkung macht eine spezielle bedarfsgesteuerte Ressourcenkontrolle und damit Medienzugriffskontrolle (*Medium Access Control*; MAC) im drahtlosen Teilnetz notwendig. Des weiteren stellt die Endsystemmobilität einen wichtigen Punkt in der Betrachtung von drahtlosem ATM dar. Um sie zu unterstützen, müssen entsprechende Funktionen (Mobilitymanagement) in bestehende ATM-Netzwerke integriert werden.

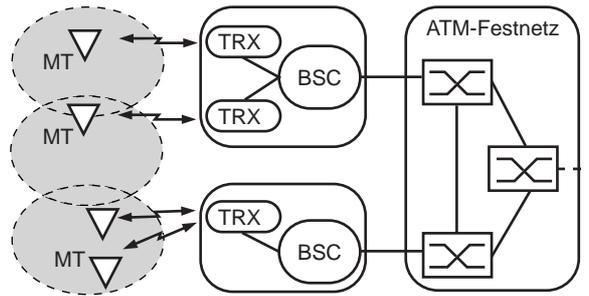


Abbildung 1: ATM Netzwerk mit drahtloser Erweiterung

2 *Native Mode* oder *TCP/IP over ATM*?

Prinzipiell kann man, entsprechend zu ATM im Festnetzbereich, zwei verschiedene Arten von drahtlosem ATM realisieren: Den sogenannten *Native Mode*, bei dem Anwendungen direkt ATM-Dienste nutzen, auf die sie über eine Anpassungsschicht zugreifen (Abbildung 2). Die zweite Art nutzt ATM als Übertragungsmedium für das existierende TCP/IP (*Transmission Control Protocol / Internet Protocol*); sie wird als *TCP/IP over WATM* bezeichnet (Abbildung 3). Im Festnetzbereich zeichnet sich der Trend ab, daß Video- und Audio-Anwendungen den *Native Mode* nutzen, während Daten-Anwendungen sowohl *Native Mode* ATM als auch *TCP/IP over ATM* nutzen werden. Für den Einsatz von drahtlosem ATM in drahtlosen LAN Umgebungen ergeben sich deutliche Unterschiede für die Anwendungen, je nachdem ob *Native Mode* ATM oder *TCP/IP over ATM* Verwendung findet. In einem drahtlosen LAN können die Benutzer bzw. Endgeräte mobil sein. Um eine unterbrechungsfreie Kommunikation zwischen den mobilen Teilnehmern bereitstellen zu können, wurden in TCP/IP verschiedene Mechanismen integriert. Diese Erweiterungen haben unter der Bezeichnung *Mobile IP* Eingang in die Internet-Protokollfamilie gefunden. Diese Erweiterungen sind aber hauptsächlich für die Datenübertragung entwickelt worden, und es ist noch nicht klar, ob sie sich auch ausreichend gut für die Audio- und Videoübertragung eignen. Hier stellt sich die Aufgabe, ein Mobilitätsmanagement für drahtloses ATM zu entwickeln. Ein weiterer Punkt bei der Gegenüberstellung der beiden Verfahren betrifft die Funkstrecke: Hier stellt sich die Frage, an welchem Ort das Segmentieren und Reassemblieren der ATM-Zellen stattfindet – kurz, wo sich die Anpassungsschicht befindet. Dies hat direkten Einfluß auf die Möglichkeit Ende-zu-Ende Dienstqualitäten anbieten zu können. Wählt man die Basisstation, dann vereinfachen sich die Anforderungen

an die Funkstrecke, weil die Paketgröße entsprechend groß gewählt werden kann. Der Hauptnachteil dieses Ansatzes wäre aber die aufwendigere Basisstation, deren Komplexität mit der Anzahl der zu versorgenden Benutzer steigt. Das Hauptaugenmerk dieses Berichts liegt deshalb auf einer *Native Mode* Infrastruktur für ATM, die eine Zellenübertragung über den drahtlosen Link unterstützt, d.h. die Anpassungsschicht befindet sich im mobilen Endsystem [Rose97, Wocj98].



Abbildung 2: *Native Mode* ATM Protokollturm für drahtloses ATM



Abbildung 3: *TCP/IP over ATM* Protokollturm für drahtloses ATM

3 Architekturmodell

Im folgenden wird ein Architekturmodell für drahtlose ATM-Endgeräte genauer betrachtet (Abbildung 4). Dabei werden die wesentlichen Aspekte der drahtlosen ATM-Architektur erläutert: Der Medienzugriff, die Dienstgüteunterstützung, das Mobilitätsmanagement und der erweiterte Handover-Mechanismus.

3.1 MAC/DLC

Der Zugriff auf das geteilte Funkmedium stellt in zweierlei Hinsicht eine Herausforderung für den Entwurf einer drahtlosen ATM-Architektur dar. Zum einen muß eine flexible Bandbreitenvergabe unter allen Endsystemen in einer Zelle gewährleistet werden. Zum anderen muß in Verbindung mit der DLC-Schicht für eine Einhaltung der Dienstgüteanforderungen gesorgt werden. Dabei sind durch die physikalischen Gegebenheiten des Funkbetriebs Grenzen vorgegeben (Fading, Signalmehrwegeausbreitung, Funklöcher, usw.), so daß man in diesem Zusammenhang von *Soft-QoS* spricht, da im

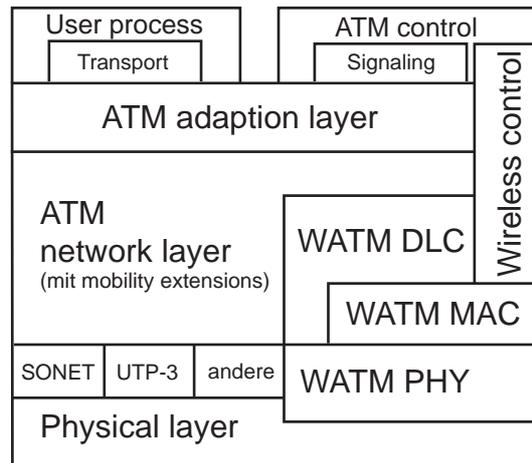


Abbildung 4: Protokollturm eines drahtlosen ATM-Endgerätes

Gegensatz zu Festnetzen, keine 100%ige Garantie gegeben werden kann. Das in [Hänl97] vorgestellte und in [Schm98] überarbeitete Protokoll basiert auf einer TDMA/FDMA-Kombination. Als Gliederungseinheit wird keine weitere Rahmenstruktur eingeführt, sondern direkt mit 53 Byte großen Einheiten (ATM-Zellen) gearbeitet. Um die zur Verfügung stehende Bandbreite optimal ausnutzen zu können, wurden *uplink* und *downlink* in ein Frequenzband gelegt. Dies nutzt die Tatsache, daß in ATM Verkehrsverträge unidirektional sind, d. h. für Send- und Empfangsrichtung können unterschiedliche Verträge abgeschlossen werden. Damit wird die Ineffizienz beseitigt, die aus einem Ungleichgewicht in der Auslastung auf up- und downlink resultiert. Um in Verbindung mit der DLC-Schicht verschiedene Dienstklassen anbieten zu können setzt das Verfahren drei grundsätzliche Mechanismen für den Mediumzugriff ein: 1. Senden nach vorausgegangener Reservierung, 2. Senden nach Erhalt der Sendemarke und 3. Konkurrierendes Senden.

3.2 Dienstgüte

Um dem Benutzer eines mobilen Endsystems dieselbe Funktionalität, die Einhaltung aller Dienstgüteeigenschaften (QoS) und eine mit dem Festnetz vergleichbare Fehler-rate bieten zu können, müssen die Verbindungsparameter Ende-zu-Ende ausgehandelt werden. Wie im Abschnitt 3.1 beschrieben, ist es schwierig, auf der Funkstrecke die Dienstgütekriterien aufrechtzuerhalten. Aus diesem Grund wird in [Schu97] ein Zwischenweg gegangen: Die Dienstgütekriterien werden zwar nach wie vor Ende-zu-Ende ausgehandelt, aber erst die Basisstation sorgt für die Einhaltung dieser Kriterien. Dies bietet einen entscheidenden Vorteil: Da die Basisstation fester Bestandteil der Aus-handlungsstrecke ist, kann sie die Dienstgütekriterien auf die Werte anpassen, die sie selbst unterstützen kann. Durch die Integration zusätzliche Mechanismen (Fehlerkorrekturverfahren, Puffer) erreicht man eine bessere Einhaltung, als es normalerweise über die Funkstrecke möglich wäre [Kris96, Engm97, Dorn98].

3.3 Mobilitätsmanagement

Die Mobilität der Endsysteme stellt innerhalb der bisher vorgeschlagenen drahtlosen ATM-Konzepte immer noch eine eher unerwünschte Eigenschaft dar. Zwar existieren Erweiterungen für die möglichst schnelle Umleitung von Verbindungen oder die Lokalisierung eines Benutzers über verschiedene Datenbankeinträge; diese reagieren jedoch nur passiv auf die Mobilität der Benutzer. Die erweiterte Funktionalität stellt das drahtlose ATM-Netz dem mobilen Benutzer also zur Verfügung, ohne zu wissen, in welchem Maße er sie in Anspruch nehmen wird. Dadurch kann es beispielsweise dazu kommen, daß beim Handover – der Verbindungsübergabe von einer Basisstation zur nächsten – QoS-Garantien einer Verbindung verloren gehen, oder diese ganz abgebrochen werden muß. Um dies zu vermeiden, wird in [Müll97, DoMü98] das Mobilitätsparameter-Management für drahtlose ATM-Netze eingeführt. Dieses Verfahren läuft verteilt in den mobilen Endsystemen, den Basisstationen und den Zonenmanagern – Verwaltungseinheiten, die das Handover für eine Gruppe von Basisstationen verwalten – ab. Die Grundlage bilden Mobilitätsparameter, durch die es einer Mobilstation ermöglicht wird, die dynamischen Aspekte ihrer Kommunikationsverbindungen anderen Netzinstanzen bekannt zu machen. Der Mobilitätsparameter einer Verbindung steuert die periodische Erzeugung von aktiven Mobilitätswahrscheinlichkeiten und deren Austausch zwischen Basisstationen und Zonenmanagern. Dadurch schafft das Mobilitätsparameter-Management die Voraussetzung, für eine effiziente Vorausplanung und gegebenenfalls Belegung der Ressourcen mobiler Verbindungen in einem drahtlosen ATM-Netzwerk.

3.4 Handover

Wie bereits erläutert, genügt es in einem drahtlosen ATM-Netzwerk nicht, das Handover nur schnell ausführen zu wollen. Vielmehr müssen zusätzliche Mechanismen in Verbindung mit einem effizienten Reroutingverfahren für die Einhaltung der Dienstgütemerkmale der Verbindung Sorge tragen. In [Lehm98, Mose98] werden Verfahren vorgestellt, die eng mit dem Mobilitätsparameter-Management zusammenarbeiten. Dem Ansatz zum Handover liegt die physikalische Tatsache zu Grunde, daß benachbarte Zellen in einem zellularen Funknetz immer einen gewissen Überlappungsbereich haben. Mobile Endsysteme die sich in diesem Bereich aufhalten, können wahlweise Verbindungen über eine der beiden Basisstationen führen. Darauf aufbauend kann ein Verdrängungsalgorithmus für mobile Endsysteme implementiert werden. Berechnet das Mobilitätsparameter-Management für ein Endsystem die Wahrscheinlichkeit eines Handovers, so kann dieses bereits im nächsten Aufenthaltsbereich vorausschauend vorbereitet werden. Herrscht im erwarteten Aufenthaltsbereich eine Ressourcenknappheit, so werden dort Endsysteme gesucht, die im Überlappungsbereich zweier Basisstationen liegen. Ein solches Endsystem wird dann seinerseits zu einem Handover angeregt, so daß in der Zelle wieder Ressourcen für das bevorstehende Handover zur Verfügung stehen.

4 Zusammenfassung

In diesem Bericht wurde knapp das Konzept und die Architektur von drahtlosem ATM vorgestellt. Es hat sich gezeigt, daß die Unterstützung von *Native WATM* be-

sonders wichtig ist. Nur damit wird es möglich, verschiedene Multimediadienste mit einem breiten Spektrum unterschiedlicher Dienstklassen zu unterstützen. Da die hohen Fehlerraten und die niedrige Bandbreite in drahtlosen Systemen die größten technischen Probleme darstellen, sind ein MAC-Protokoll, das QoS berücksichtigt, und ein DLC-Protokoll, das den Durchsatz auf fehlerbehafteten Funkstrecken verbessert, Schlüsselthemen bei der Entwicklung von drahtlosem ATM. Um den Mobilitätscharakter in drahtlosem ATM entsprechend zu unterstützen und um Verbindungen mit entsprechender Dienstqualität einhalten zu können, bedarf es eines vorausschauenden aktiven Mobilitätsparameter-Managements in Verbindung mit einem leistungsfähigen Handover-Protokoll.

Literatur

- [DoMü98] Elmar Dorner und Frank Müller. A Management Extension for Mobility Parameter Support in Wireless ATM. To be published in PWC '98, 1998.
- [Dorn98] Elmar Dorner. Support for End-to-End Quality of Service in Wireless ATM Networks. To be published in LAN/MAN '98, 1998.
- [Engm97] Steffen Engmann. Ein adaptives Fehlerkontrollverfahren zur Sicherung der Übertragungsqualität in Mobilfunknetzen. Diplomarbeit, Institut für Telematik, 1997.
- [Hänl97] Christoph Hänle. Dienstintegrierende Zugriffsprotokolle für drahtlose ATM-Netze. Studienarbeit, Institut für Telematik, 1997.
- [Kris96] Kurt Iben Kristiansen. Simulations and Analysis of Error Control Mechanisms for Wireless ATM Networks. Diplomarbeit, Institut für Telematik, 1996.
- [Lehm98] Volker Lehmann. Rerouting-Verfahren für ATM-Netze mit mobilen Endsystemen. Diplomarbeit, Institut für Telematik, 1998.
- [Mose98] Boris Moser. Verbindungsübergabe bei Bandbreitenengpässen in zellenbasierten drahtlosen ATM-Netzen. Diplomarbeit, Institut für Telematik, 1998.
- [Müll97] Frank Müller. Signalisierung in drahtlosen ATM-Netzen. Studienarbeit, Institut für Telematik, 1997.
- [Rose97] Verena Rose. Drahtloses ATM. In *Netzwerk-Management und Hochgeschwindigkeitskommunikation*, Band 15. Institut für Telematik, 1997.
- [Schm98] Stefan Schmitz-Linneweber. Entwurf und Simulation eines Zugriffsverfahrens für funkgestützte ATM-Netze. Diplomarbeit, Institut für Telematik, 1998.
- [Schu97] Tobias Schulz-Hess. Dienstqualität in ATM-Netzen mit mobilen Endsystemen. Diplomarbeit, Institut für Telematik, 1997.
- [Wocj98] Pawel Wocjan. Drahtloses ATM. In *Netzwerk-Management und Hochgeschwindigkeitskommunikation*, Band 17. Institut für Telematik, 1998.

Von Unicast über Multicast und Concast zu Multipeer: Anforderungen an Kommunikationsprotokolle

Stefan Dresler

Kurzfassung

Auf dem derzeitigen Stand der Spezifikation der ATM-Technologie (Asynchroner Transfermodus) wird Unterstützung für die Kommunikation eines Senders mit einem (Unicast) oder mehreren (Multicast) Empfängern geboten. Für eine Kommunikation innerhalb einer Gruppe, welche auch mehrere Sender enthalten kann (Multipeer), existieren jedoch noch keine Mechanismen. Die Internet-Protokolle der Versionen 4 und 6 bieten zwar Mechanismen an zur Multipeer-Kommunikation, lassen jedoch ebenfalls Fragen wie die Fluß- und Staukontrolle offen oder bieten Potential für Verbesserungen (Multicast-Routing).

In diesem Bericht wird auf noch offene Fragen der Multipeer-Kommunikation speziell über ATM eingegangen. Viele Aspekte gelten jedoch in gleicher oder ähnlicher Form sowohl für andere verbindungsorientierte Netze wie ISDN oder Frame Relay als auch für die Internet-Protokolle der Versionen 4 und 6.

1 Einleitung

Das Internet-Protokoll in der derzeit verbreiteten Version 4 bot ursprünglich – außer über rundruffähigen Medien wie Ethernet im lokalen Bereich (LAN) – lediglich die Unicast-Kommunikation (ein System sendet an einen Empfänger) an. Mit der vor einigen Jahren vorgeschlagenen Erweiterung [LyRo93, Deer89] wurde es möglich, auch über LAN-Grenzen hinweg mit einer Gruppe von Empfängern zu kommunizieren (Multicast-Kommunikation). Der zugrundeliegende Mechanismus eignet sich auch für die sogenannte Multipeer-Kommunikation, bei der mehrere Sender mit mehreren Empfängern kommunizieren. Auch nach mehreren Jahren Forschung und Testbetrieb, vornehmlich durch das sogenannte Mbone (Multicast-Backbone) [MBo], ist jedoch noch kein stabiler Betrieb gewährleistet, und auch die Verbreitung ist nur beschränkt. Multipeer kann als die Kombination von Multicast und Concast (mehrere Sender senden an einen Empfänger) betrachtet werden.

Bei der Entwicklung der Version 6 des Internet-Protokolls und der Spezifikation der ATM-Technologie (Asynchroner Transfermodus) wurde bereits Wert auf eine Integration von Mechanismen zur Multicast-Kommunikation gelegt. Einem breiten Einsatz jedes der genannten Protokollen im Zusammenhang mit Multipeer-Kommunikation stehen jedoch noch einige ungelöste Probleme im Wege.

Während die Mehrpunkt-Kommunikation bei einigen verbindungsorientierten Netzen nicht vorgesehen ist (etwa bei ISDN oder ADSL), wurde beim Asynchronen Transfermodus (ATM) bereits in der Spezifikationsphase Wert auf die Integration entsprechender Mechanismen gelegt [Foru94, Foru96]. Der Zugriff auf diese Funktionalität erfordert jedoch entsprechende Unterstützung auf Programmierseite [DHSW97].

Ziel dieses Berichts ist es, durch die Beschreibung der Besonderheiten von Unicast-, Multicast-, Concast- und Multipeer-Umgebungen noch offene Fragestellungen im Zusammenhang mit der Multipeer-Kommunikation aufzuzeigen [CaDr97].

2 Multicast-Kommunikation

Der Schritt von der Unicast- zur Multicast-Kommunikation stellt sich auf verschiedenen Ebenen des Schichtenmodells nach ISO/OSI unterschiedlich dar. Außerdem können sich Unterschiede ergeben durch die zum Einsatz kommenden Übertragungssysteme. Generelle Aufgaben, welche für die Multicast-Kommunikation zu lösen sind, sind:

- Nachrichtenverteilung und -duplizierung
- Adressierung
- Verkehrskontrolle
- Verbindungsauf- und -abbau bei verbindungsorientierten Diensten
- Zuverlässige Datenübermittlung
- Unterstützung auch heterogener Links, unterschiedlich leistungsfähiger Rechner und differierender QoS-Wünsche von Empfängern
- Unterstützung verschiedener Topologien und Netzwerk-Technologien

Zur effizienten und zuverlässigen Realisierung von Multicast eingesetzte Protokolle streben außerdem die folgenden generellen Eigenschaften an:

- Skalierbarkeit hinsichtlich großer Empfängermengen
- Robustheit bei Ausfällen von End- und Zwischensystemen und Links
- Unterstützung von Dynamik (Eintritte in und Austritte aus der Gruppe)
- Hohe Geschwindigkeit bei Übertragung, Verbindungsauf-/abbau (bei verbindungsorientierten Netzen) und Fehlererkennung

3 Concast-Kommunikation

Concast bezeichnet, wie erwähnt, ein Szenario mit mehreren Sendern und einem Empfänger. Die wichtigsten Unterschiede der Concast-Kommunikation zu Unicast-Szenarien sind im allgemeinen bzw. bei ATM:

- Geänderte Fluß-, Stau- bzw. Ratenkontrolle

- Notwendigkeit einer Senderidentifikation
- Notwendigkeit einer Rahmen-Sequentialisierung beim Zusammentreffen der Datenströme unterschiedlicher Sender (kein Mischen der Zellen unterschiedlicher Rahmen auf einem virtuellen Kanal, sofern keine Senderidentifikation pro Zelle mitgeführt wird)

4 Multipeer-Kommunikation

Die Multipeer-Kommunikation weist Charakteristika sowohl der Multicast- als auch der Concast-Kommunikation auf. Häufig läßt sich sagen, daß mit Multicast oder Concast verbundene Probleme bei der Multipeer-Kommunikation verschärft auftreten.

Bei der Multipeer-Kommunikation ergeben sich über die von der Multicast- und der Concast-Kommunikation bekannten Fragestellungen folgende Aspekte:

- Erweiterte Mechanismen der Nachrichtenverteilung (Unicast, Multicast, Multipeer)
- Modifizierte Fluß-, Stau- bzw. Ratenkontrolle
- Realisierte Ordnung (FIFO-, Kausal-, Total-Ordnung, Transaktionssemantik)
- Erweiterte Anforderungen an die Robustheit der Mechanismen
- Unterstützung von Dynamik (Eintritte in die und Austritte aus der Gruppe)
- Geschwindigkeit (bei Übertragung, Verbindungsauf-/abbau (bei verbindungsorientierten Netzen) und Fehlererkennung)

Es zeigt sich also, daß die Multipeer-Kommunikation eine über die von etablierten Protokollen angebotenen Mechanismen hinausgehende Aufmerksamkeit erfordert, welche den für die Entwicklung geeigneter Mechanismen für Multicast-Szenarien nötigen Aufwand rechtfertigt.

5 Untersuchte Architektur für die Multipeer-Kommunikation

Im folgenden wird die Architektur vorgestellt, welche im Rahmen der Forschungstätigkeit am Institut untersucht wird.

5.1 Bereitstellung einer Total-Ordnung

Einige Anwendungen profitieren von einem Multipeer-Übertragungsdienst, welcher Nachrichten bei allen Empfängern in der gleichen Reihenfolge ausliefert, also eine Total-Ordnung (globale Ordnung, Agreed Order) realisiert. Dabei ist man primär an *irgendeiner* Ordnung interessiert, solange sie bei allen Empfängern gleich ist. Daneben sind noch andere Kriterien wie möglichst geringe Verzögerung von Nachrichten von Interesse.

Eine solche Total-Ordnung läßt sich am einfachsten zweistufig realisieren: in einem ersten Schritt werden alle Nachrichten an ein zentrales System geschickt (Concast), welches diese dann z.B. per Multicast verteilt. Die Total-Ordnung ergibt sich dann ohne weitere Vorkehrungen, solange es bei der Verteilung der Nachrichten per Multicast nicht zu Umordnungen kommt.

Soll – etwa aus Gründen der Ausfallsicherheit – auf ein zentrales System verzichtet werden, so muß ein Weg gefunden werden, dezentral eine globale Ordnung zu definieren. Dies wird häufig durch ein Token erreicht, das dafür sorgt, daß nur eine Station, die gerade das Token hält, senden darf. Um *Race Conditions* bei der Weitergabe des Tokens zu vermeiden und gleichzeitig die Möglichkeit der Erkennung von Verlusten zu besitzen, wird man in der Regel außerdem eine Sequenznummer mitführen, die im Token gehalten werden kann. Bei einer großen Zahl an potentiellen Sendern kann es zu unerwünschten Sendepausen kommen, während derer eine Station nicht senden kann, da sie auf das Token wartet. Dem kann damit begegnet werden, daß ein Token nur zur Festlegung der Sequenznummern zu sendender Pakete verwendet und frühestmöglich (auch vor dem Aussenden des zugehörigen Pakets) weitergeleitet wird.

Die Vergabe einer global eindeutigen Sequenznummer kann auch durch die Verwendung einer Zeitbasis erfolgen. Die so erhaltenen Nummern sind dann in der Regel nicht mehr lückenlos vergeben. Es ist lediglich dafür Sorge zu tragen, daß die Auflösung der Uhr hinreichend fein ist, daß aufeinanderfolgende Pakete eines Senders unterschiedliche Sequenznummern erhalten. Einer (bei hoher Uhrenauflösung selten auftretenden) Vergabe der gleichen Sequenznummer an Pakete auf mehreren Sendern kann durch die Verwendung einer eindeutigen Senderidentifikation begegnet werden, auf der eine einfach festzustellende Ordnung definiert wird (etwa lexikalische Ordnung auf den Sendernamen). Basierend auf einer maximal angenommenen Drift der Uhren sowie der maximalen Übertragungs- und Bearbeitungsverzögerung von Nachrichten der verschiedenen Sender läßt sich berechnen, wie lang auf Empfängerseite gewartet werden muß, bis man sicher sein kann, daß kein „früheres“ Paket mehr kommt. Alle empfangenen Pakete mit kleinerem Zeitstempel können dann ausgeliefert werden.

5.2 Verkehrskontrolle

Wie oben bereits angedeutet, wirft die Multipeer-Kommunikation neue Fragen auf hinsichtlich der Verkehrskontrolle. Schon beim Concast stellt sich die Frage, welcher Sender wann wieviel senden darf. Beim Multicast ist zu klären, wie die Feedback-Information mehrerer Empfänger geeignet (sowohl vom Datenvolumen als auch algorithmisch bei widersprüchlichen Informationen) verarbeitet werden kann. Bei der tokenbasierten Senderechtsvergabe kann ein Kredit mitgeführt werden, der etwa von allen Systemen entsprechende ihrem Bedarf verringert und durch ein ausgezeichnetes System wieder erhöht wird. Letzteres System sammelt dann auch die Feedback-Information der Empfänger. Bei zeitbasierten Verfahren sind andere Arten der Koordination erforderlich.

Zentraler Aspekt der Verkehrskontrolle bei der Multipeer-Kommunikation ist nicht nur die protokolltechnische Realisierung einer Strategie zur Vergabe des Senderechts bzw. des zu sendenden Datenvolumens, sondern auch die grundsätzliche Diskussion des Begriffs Fairness, da unterschiedliche Interpretationen möglich sind.

5.3 Zweischichtige Hierarchie des Modells

In Abbildung 1 ist ein zweischichtiges Modell der untersuchten Architektur skizziert. Auf der unteren Stufe steht eine Anbindung sende- und empfangswilliger Rechner,

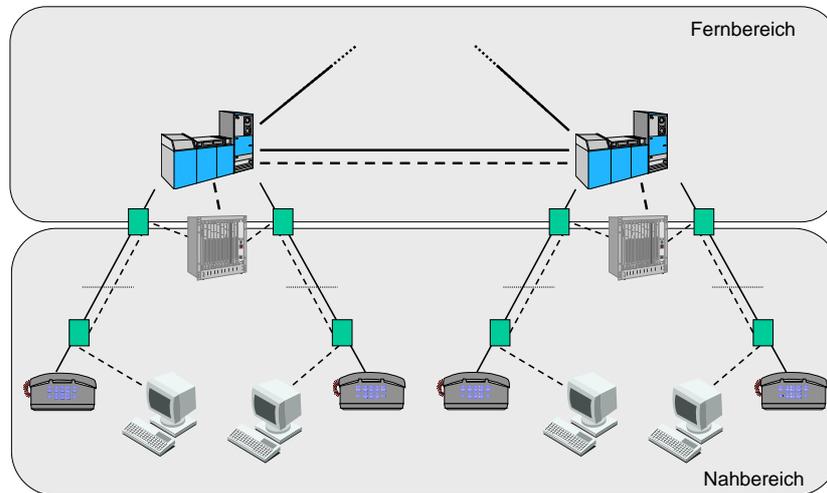


Abbildung 1: Zweischichtiges Modell der untersuchten Architektur

die in der Abbildung über ADSL (Asymmetric Digital Subscriber Line) angeschlossen sind, wobei diese Anbindung direkt oder über ein zwischengeschaltetes LAN erfolgen kann. Datenverkehr ist gestrichelt gezeichnet, während der hier nicht näher betrachtete klassische leitungsgebundene ISDN- oder Sprechverkehr durch durchgezogene Linien dargestellt wird. Die Stationen bauen eine Verbindung zu einem „in der Nähe befindlichen“ Server auf, welcher für die empfangenen Pakete Sequenznummer vergibt und sie anschließend an alle anderen beteiligten Server verteilt. Diese Verteilung realisiert die obere Stufe der Architektur.

Die Zweiteilung bietet sich an, weil im Nahbereich nur eine bedingte Verkehrskontrolle erforderlich ist und Übertragungswiederholungen verhältnismäßig schnell durchführbar sind. Außerdem können hier Unicast-Verbindungen zum Einsatz kommen, und durch die Verwendung eines Servers werden die Empfänger voneinander entkoppelt, was die Robustheit beim Ausfall von Endsystemen erhöht. Dieses Konzept kommt mit seiner Einfachheit den Erfordernissen der Anbindung von Endsystemen entgegen.

Auf der Fernebene, also zwischen den Servern, lassen sich dagegen leistungsfähige Multicast-Verteilstrategien [DrHo97, CaDS98] in Verbindung mit geeigneten Korrekturverfahren anwenden, basierend etwa auf Vorwärtsfehlerkorrektur in Kombination mit Übertragungswiederholung. Durch die in den Servern bereits aggregierten Ströme der jeweils assoziierten Endsysteme ergibt sich ein höheres und ggfs. stärker geglättetes Datenvolumen pro Zeiteinheit, dem sich bei Einsatz von Unicast-Verbindungen eher mit etablierten Verkehrskontrollverfahren wie den aus TCP bekannten Mechanismen begegnen läßt. Speziell für ATM-Netze kann durch den Einsatz der Available Bitrate (ABR) eine Verkehrskontrolle realisiert werden. Für einen Einsatz auf Multicast-Verbindungen sind die Verfahren jedoch geeignet zu modifizieren oder zu ersetzen.

6 Zusammenfassung

Die Untersuchung begann mit einer Hinführung zu den Charakteristika der Multipeer-Kommunikation, die über die von der Unicast-, Multicast- und Concast-Kommunikation hinausgehende Anforderungen an das System stellt. Für eine effiziente Bereitstellung eines ordnenden Dienstes wurde eine zweistufige Architektur vorgeschlagen, die sich zu den Endsystemen hin durch ihre Einfachheit und im Backbone durch den Einsatz leistungsfähiger Übertragungsmechanismen auszeichnet.

Literatur

- [CaDr97] G. Carle und S. Dresler. Reliable Ordered Multicast Service (ROMS) over NBMA Networks. In *Proceedings of 6th Open Workshop on High Speed Networks, Stuttgart, Deutschland, 8.-9. Oktober 1997*, 1997, S. 161–167.
- [CaDS98] G. Carle, S. Dresler und J. Schiller. Multipath FEC Scheme for the ATM Adaptation Layer AAL5. In *International Conference on Broadband Communication*, 1998.
- [Deer89] Steven Deering. Host Extensions for IP Multicast Routing. Request for Comments – RFC 1122, Internet Engineering Task Force (IETF), August 1989.
- [DHSW97] S. Dresler, M. Hofmann, C. Schmidt und H. Wiltfang. A Native ATM API Suited for Multimedia Communication. In R. Steinmetz und L.C. Wolf (Hrsg.), *Interactive Distributed Multimedia Systems and Telecommunication Services*, Nr. 1309 der Lecture Notes in Computer Science. Springer Verlag, 1997, S. 352–363.
- [DrHo97] S. Dresler und M. Hofmann. Adaptive Error Correction to Support Heterogeneous Multicast Groups. In *Proceedings of 6th Open Workshop on High Speed Networks, Stuttgart, Deutschland, 8.-9. Oktober 1997*, 1997, S. 169–174.
- [Foru94] The ATM Forum. ATM User-Network Interface Specification, Version 3.1. Technischer Bericht, The ATM Forum, September 1994. af-uni-0010.002.
- [Foru96] The ATM Forum. ATM User-Network Interface (UNI) Signaling Specification, Version 4.0. Technischer Bericht, The ATM Forum, Juli 1996. af-sig-0061.000.
- [LyRo93] Daniel C. Lynch und Marshall T. Rose (Hrsg.). *Internet System Handbook*. Addison-Wesley Publishing Company, Inc. 1993.
- [MBo] mbone.de (Homepage). im WWW erreichbar über <http://www.mbone.de/>.

WebComposition: Ein objektorientiertes Unterstützungssystem für das Web Engineering

Martin Gaedke

Kurzfassung

Die Wartung von Web-Anwendungen ist eine aufwendige und fehleranfällige Aufgabe, da auf viele Entwurfsentscheidungen nach ihrer Implementierung nicht mehr direkt zugegriffen werden kann. Trotzdem setzt sich nur langsam die Erkenntnis durch, daß für Entwicklung, Betrieb und Wartung von Web-Anwendungen ein disziplinierter Ansatz analog zum Software Engineering notwendig ist. In diesem Beitrag wird das WebComposition System als Unterstützung für das Web Engineering vorgestellt. Das System basiert auf einem objektorientierten Komponentenmodell für Web-Anwendungen und ermöglicht den feingranularen Zugriff auf die Komponenten während des gesamten Lebenszyklus einer Web-Anwendung sowie die automatische Generierung von Ressourcen für Web-Server aus dem Modell heraus.

1 Motivation und Zielsetzung

Aus der Sicht der Softwaretechnik ist das World Wide Web (WWW) eine neue Anwendungsdomäne. Das WWW, ursprünglich als Informationssystem für weltweit verteilte Forschungsgruppen konzipiert, hat sich innerhalb kürzester Zeit als Plattform für Hypertextanwendungen vom akademischen bis hin zum kommerziellen Umfeld etabliert und ist mittlerweile (fälschlicherweise) zu einem Synonym für das Internet geworden. Die rasant steigenden Nutzerzahlen sorgen gleichzeitig für ein starkes Engagement der Wirtschaft, die zusammen mit der akademischen Forschung die Entwicklung von neuen weborientierten Standards vorantreibt, bei denen es sich fast durchweg um de facto Standards handelt, da die Rate der Einführung von Neuerungen selbst für die vergleichsweise schnellen Standardisierungsprozesse im Bereich Internet zu hoch ist. Es ist nicht zu erwarten, daß die Aufgabe der Entwicklung von Web-Anwendungen in Zukunft leichter werden wird, da die Ansprüche an die Qualität der Anwendungen gestiegen sind und weiter steigen. Während es am Anfang ausreichte, „online zu sein“, ist es inzwischen besonders für kommerzielle Anbieter im Web wichtig, sich ansprechend und auf den einzelnen Kunden speziell angepaßt zu repräsentieren, um die Erfüllung des Ziels der Institution oder des Unternehmens voranzutreiben. Der zunehmenden Komplexität von Anwendungen innerhalb des Web steht eine bislang wenig disziplinierte bzw. ad-hoc Vorgehensweise bei der Entwicklung und Pflege von Web-Anwendungen gegenüber. Die Erkenntnis, daß die Erstellung von Web-Anwendungen gleichbedeutend mit

Softwareentwicklung ist, legt die Erarbeitung softwaretechnischer Methoden für Entwicklung, Betrieb und Wartung von Anwendungen im Web nahe [GeWG97]. Diese neue Disziplin, im folgenden Web Engineering (oder kurz WebE) genannt, verspricht eine Kostenreduktion bei der Entwicklung und vor allem bei der Wartung, dadurch daß der Entwicklungsprozeß als solcher planbar wird und somit hilft die Qualität des Produkts zu sichern. Wie bei anderer Softwareentwicklung lassen sich die Aufgaben des Web Engineering mehreren Entwicklungsphasen zuordnen. In dieser Arbeit wird dabei von den folgenden Phasen ausgegangen:

- *Analyse.* Hierbei werden Informationen über die Anwendungsdomäne gesammelt und bewertet. Zu klären ist hierbei insbesondere, welcher Art der zu präsentierende Informationsraum ist. Die anzusprechende Zielgruppe beeinflusst hierbei sowohl die benutzte Sprache und Darstellung als auch die Art der Bedienung der Web-Anwendung.
- *Entwurf.* Ziel des Entwurfsprozesses ist es, Inhaltselemente des durch die Analyse gefunden Informationsraumes genau zu beschreiben und ihnen eine logische Struktur zu geben. Außerdem wird eine Aufteilung der Inhalts- und Strukturelemente auf Ressourcen definiert und die konkrete Navigationsstruktur zwischen Elementen bzw. Ressourcen festgelegt. Ein weiteres wichtiges Ergebnis des Entwurfsprozesses ist die Definition von wiederverwendbaren Entwurfsartefakten, z.B. die Definition von Layoutelementen wie Logos oder Navigationsleisten, um der Web-Anwendung ein einheitliches Look&Feel zu verleihen.
- *Implementierung.* In der Implementierungsphase werden Inhalt und Struktur auf die Ressourcen des Web-Server abgebildet. Die im Entwurf definierten Navigationsstrukturen müssen durch HTML-Links in den dazugehörigen Seiten umgesetzt werden. Auf Strukturen der Anwendung, die während der Entwurfsphase gewonnen wurden, kann nach ihrer Abbildung auf Ressourcen nicht mehr als eigenständige Entwurfsartefakte zugegriffen werden. Der Verlust von Strukturinformationen bzw. der Information über Entwurfsartefakte bei der Umsetzung in Implementierungsartefakte erschwert daher die spätere Wartung der Web-Anwendung.
- *Betrieb und Wartung.* Web-Anwendungen zeichnen sich gegenüber „herkömmlicher“ Software nach ihrer initialen Implementierung durch eine hohe Änderungsdynamik sowohl der Struktur als auch der Inhalte aus. Während zu den Aufgaben der Wartung bei anderer Software hauptsächlich das Beseitigen von Fehlern zählt, die Funktionsweise und die Benutzerschnittstelle eines Programmes aber weitgehend unverändert bleiben, befinden sich Web-Anwendungen in einem Zustand kontinuierlicher Weiterentwicklung. Typische Probleme der Wartung sind z.B. Änderung replizierter HTML-Fragmente, die Manipulation an Strukturen, die Änderung der Aufteilung auf Ressourcen oder die Änderung der Navigationsstruktur.

Die Orientierung an einem bestimmten Hypertextmodell (z.B. an dem Dextermodell [HaSc94]) ist wegen der unterschiedlichen Ausprägungen einer Web-Anwendung wenig sinnvoll. Die Beschreibung einer Web-Anwendung sollte daher auf einer allgemeineren Ebene erfolgen, wobei aber die Möglichkeit der Bildung höherer Abstraktionen zur Implementierung bestimmter Hypertextmodelle gegeben sein sollte.

Die allgemeine Zielsetzung in dem Projekt WebComposition ist die Schaffung einer Grundlage für Web Engineering, d.h. ein disziplinierter Ansatz analog zum Software Engineering für Entwicklung, Betrieb und Wartung von Web-Anwendungen, bestehend aus aufeinander abgestimmten konstruktiven Maßnahmen der Qualitätssicherung. Hierzu zählen u.a. die Strukturierung von Entwicklung und Wartung in einem Prozeßmodell, der gezielte Zugriff auf Entwurfsentscheidungen über den gesamten Lebenszyklus einer Web-Anwendung hinweg sowie Modelle zur Beschreibung von Entwurfsartefakten [Gell97] und Methoden zur Wiederverwendung von daraus resultierenden Artefakten.

2 Stand der Entwicklung im Web Engineering

Web-Anwendungen können sehr unterschiedliche Ausprägungen haben. Die Bandbreite reicht von statischen, dokumentenorientierten Hypertextsystemen bis hin zu interaktiven, verteilten Anwendungen (z.B. mit Hilfe von Formularen und CGI-Skripten realisiert). Bezüglich der unterschiedlichen Ausprägungen existieren verschiedene Werkzeuge, die bei der Erstellung von statischem Hypertext, bei der Integration von Datenbanken als Informationssysteme oder bei der Entwicklung von hochdynamischen Anwendungen (z.B. sessionbasierte Anwendungen) unterstützen. Statischer Hypertext ist dadurch charakterisiert, daß nur statische Links und Seiten angewendet werden. Im Vergleich hierzu sind bei einem Web-Informationssystem die Seiten dynamisch, lediglich die Linkstruktur bleibt statisch. Dynamische Web-Anwendungen sind dadurch gekennzeichnet, daß sowohl die Seiten als auch die Linkstruktur dynamisch erzeugt werden.

Die ersten Web Engineering Werkzeuge unterschieden sich in der *Art der Entwicklungsunterstützung*. So unterstützen einige Werkzeuge die Erstellung einzelner Seiten, andere unterstützen das gesamte Management von Web-Sites [Micr98a] und wieder andere beinhalteten grundlegende Formen der Abstraktion, z.B. arbeitet das Entwicklungswerkzeug Fusion von NetObjects zur Erstellung dokumentenorientierter Webs mit dem Begriff „Style“, der alle Standardelemente des Seitenlayouts (Grafiken, Textattribute usw.) umfaßt. Die Art bzw. der Zeitpunkt der *Ressourcengenerierung* ist ein anderes Kriterium, das zur Klassifikation von Werkzeugen dienen kann. Von statischer Generierung spricht man, wenn zum Zeitpunkt der Erstellung der Web-Anwendung der Code jeder Ressource vollständig definiert ist und in einem Datenspeicher abgelegt werden kann. Im Gegensatz dazu steht die dynamische Generierung, bei der eine Ressource erst zum Zeitpunkt der Anforderung durch den Server erzeugt wird. Wichtige Vertreter dieser Klasse sind Server-Side Includes (SSI) und Server-Side Scripting. Beim SSI [fSAa95] können mehrere Dateien in eine HTML-Datei eingelesen werden. Dieser Mechanismus wird zumeist durch eine Art Präprozessor, der direkt im Web-Server integriert ist, realisiert. Bei dem darüber hinaus führenden Server-Side Scripting-Konzept handelt es sich um in HTML-Ressourcen eingebettete Befehle, die serverseitig vor Auslieferung der Ressource von einem Skriptinterpreter ausgeführt werden. Stellvertreter dieser Kategorie sind insbesondere Active Server Pages [Micr98b] und PHP/FI [Lerd98]. Eine weitere Klasse von Werkzeugen unterstützt die *Entwicklung dynamischer Anwendungen*. Hierzu zählt das WebObjects System [NeXT96], daß ebenfalls auf dem Prinzip von Server-Side Scripting basiert. Im Unterschied zu diesem stellt WebObjects dem Entwickler neben Laufzeitsystem und Skriptinterpreter auch eine umfangreiche Klassenbibliothek zur Verfügung, der eine objektorientierte Architektur

für Web-Anwendungen zugrundeliegt. Allerdings unterstützt WebObjects nicht die Organisation und Verwaltung von Ressourcen, außerdem ist die Wiederverwendung von Objekten nur eingeschränkt möglich, da sie in statischen HTML-Templates eingebettet sind. Mit Ausnahme von RMCASE sind alle betrachteten Werkzeuge auf eine der beiden Möglichkeiten der Ressourcengenerierung festgelegt, wobei die dynamische Generierung überwiegt. RMCASE soll den gesamten Lebenszyklus einer Web-Anwendung von Design über Implementierung bis hin zu Betrieb und Wartung unterstützen. Es basiert auf der in [IsSB95] beschriebenen Relationship Management Method (RMM) und ist an sich unabhängig von einer bestimmten Hypermedia-Plattform. Es ist anzumerken, daß RMCASE wegen seines zugrundeliegenden Datenmodells RMM nicht für alle Arten von Web-Anwendungen gleichermaßen geeignet ist. Es orientiert sich stark an E-R-Modellierung und ist dementsprechend gut für die Abbildung von Inhalten relationaler Datenbanken geeignet.

Zusammenfassend ist festzustellen, daß dateibasierte Ressourcen zur Modellierung von Web-Anwendungen ungeeignet sind, da sie als „Modellierungseinheit“ zu grob sind. Die Entwurfsentscheidung der Aufteilung der Inhalte auf Ressourcen ist nach der Implementierung als solche nicht zugreifbar. Für Web Engineering und insbesondere für eine bessere Unterstützung des Betriebs und der Wartung einer Web-Anwendung sind feiner granulare Strukturen notwendig, um Web Engineering über den gesamten Lebenszyklus einer Web-Anwendung anzuwenden.

3 WebComposition Component Model

3.1 Komponenten

Mit dem Ziel, eine bessere Plattform für die Beschreibung, Wiederverwendung und Wartung von Web-Anwendungen zu schaffen, wurde im Projekt WebComposition ein implementierungsunabhängiges, objektorientiertes Komponentenmodell entwickelt, das als Basis für das WebComposition Laufzeitsystem dient. Der Kern des Component Model sind Komponenten als uniformes Modellierungskonzept für Entwurfsartefakte in Web-Anwendungen. Die Idee des Component Model ist die objektorientierte Darstellung einer Web-Anwendung über ihren gesamten Lebenszyklus als Basis für ein abstraktes Entwurfsmodell und für das Laufzeitmodell. Das Component Model unterstützt zum einen eine automatische aber auch noch im Detail manipulierbare Abbildung auf Ressourcen während es auf der anderen Seite mit objektorientierten Konzepten besser mit höheren Abstraktionen korrespondiert als das Web-Laufzeitmodell. Eine Web-Anwendung wird im Modell als hierarchische, strukturelle Dekomposition von Komponenten beschrieben. Auf der höchsten Hierarchieebene können Komponenten dabei ganzen Teilwebs entsprechen, darüber hinaus korrespondieren sie mit einzelnen Ressourcen oder Fragmenten von Ressourcen. Auf dieser Stufe der Hierarchie können Komponenten anwendungs-spezifische Abstraktionen darstellen, wie z.B. Firmenlogo, Navigationsleiste, Werbespalte, oder aber auch nur Wertepaare, die das Design der Seite beeinflussen, z.B. Hintergrundfarbe und Überschriftgröße. Die Blätter in der Komponentenhierarchie werden als Primitive bezeichnet, Beispiele hierfür sind einfache Werte u.a. vom Typ Text, Integer, Datum oder URL.

Bei den Komponenten handelt es sich um voneinander unterscheidbare Objekte, die einen Zustand in Form von typisierten Attributen (die Properties genannt werden und

einfachen Name-Wert Paaren entsprechen) und ein Verhalten als Menge von Methoden besitzen. Eine Komponente erbringt mit Hilfe ihrer Methoden eine Reihe von Diensten, mindestens aber zwei Basisdienste:

- *Darstellungsdienst*. Durch die Methoden *generateCode* und *update* wird eine Repräsentation der Komponente in Abhängigkeit von ihrem Zustand erzeugt. Bei dem daraus resultierenden Code kann es sich um den Teil einer Ressource, einer ganzen Ressource oder sogar mehreren Ressourcen handeln.
- *Persistenzdienst*. Dieser wird durch die Methoden *getProperties* und *setProperty* angestoßen, die dazu dienen, den Zustand einer Komponente zu serialisieren bzw. zu deserialisieren. Dadurch wird es möglich, Komponenten in einem Datenspeicher festzuschreiben, sie also persistent zu machen, und sie später aus diesem wieder zu laden.

Diese Basisdienste ermöglichen es, eine Menge von Komponenten zu definieren, die zusammen ein ganzes Web erzeugen, und diese zusammen in Dateien oder einer Datenbank zu speichern, wie die folgende Abbildung Bild 1 darstellt.

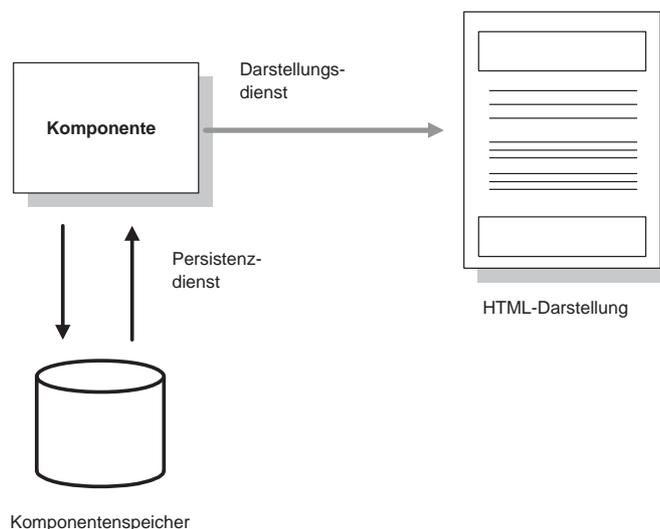


Abbildung 1: Von Komponenten erbrachte Dienste

3.2 Vererbung

Bei objektorientierten Modellen dient der Mechanismus der Vererbung zur Ableitung neuer Objekte auf der Basis existierender, indem deren Definitionen übernommen und erweitert werden. Für die Ableitung neuer Komponenten aus alten wurde ein Prototyping-Modell gewählt, wie es in [UnSm91] für die Programmiersprache *SELF* beschrieben ist. Es ist einfacher als die klassenbasierte Vererbung gängiger objektorientierter Sprachen wie Smalltalk oder Java. Beim Prototyping-Modell von WebComposition wird keine Unterscheidung zwischen Klassen und Instanzen vorgenommen. Jede Komponente kann gleichzeitig als Instanz benutzt werden und als ein Prototyp für andere Komponenten dienen, denen sie ihr Verhalten und ihren Zustand zur Verfügung stellt.

Die Beziehung ist-Klasse und ist-Instanz-von-Klasse der klassenorientierten Vererbung werden hier also auf eine einzige Relation *ist-Instanz-von-Prototyp* abgebildet.

Jede Komponente kann mehrere Prototypen haben, mindestens jedoch einen, den `WebCompositionComponentPrototype`. Dieser Prototyp besitzt keine Properties, bietet aber einen einfachen Darstellungs- und Persistenzdienst an, der durch komplexere Dienste ersetzt werden kann. Ob auf ein Prototyp-Attribut oder ein lokal definiertes Attribut zugegriffen wird, ist für die Methoden der Komponente transparent. Allerdings sind die Werte von Prototyp-Attributen weiterhin nur einmal innerhalb des jeweiligen Prototyps vorhanden (vergleichbar mit Klassenvariablen in Programmiersprachen wie Smalltalk oder Java). Eine Änderung der Attribute einer Komponente wirkt sich transitiv auf den Zustand aller von ihr als Prototyp referenzierten Komponenten aus. Prototyp Attribute können von lokalen Attributen gleichen Namens überladen werden.

Bezüglich der Repräsentation der Komponenten durch HTML-Code, kann man sich Prototypen als eine Art HTML-Schablone denken, die eine bestimmte Struktur vorgibt und Bereiche definiert, wo konkrete Inhalte eingefügt werden können. Der Vorteil von Prototypen ist ähnlich dem der gemeinsamen Nutzung von Komponenten zur Replikation von Repräsentationen. In diesem Fall werden die Prototypen und damit ihr Zustand und Verhalten gemeinsam genutzt (sog. Component Sharing). Das Component Sharing führt zu einer Reduzierung des Entwicklungsaufwandes durch Wiederverwendung von Prototypen und zu einer Erleichterung der Wartung, da gemeinsam benutzte Properties an nur einer Stelle geändert werden müssen.

4 WebComposition System

Für eine Nutzung des Component Model für Entwicklung und Wartung von Web-Anwendungen ist eine Systemunterstützung notwendig. Das System stellt persistenten Speicher für das Component Model zur Verfügung und verwaltet den feingranularen Zugriff auf die Entwurfsartefakte während des gesamten Lebenszyklus der Web-Anwendung. Außerdem bildet das System das Component Model inkrementell auf datei-basierte Ressourcen ab, so daß das System für den verwendeten Web-Server transparent bleibt.

Die System Architektur zur Umsetzung dieser Aufgaben ist in der folgenden Abbildung Bild 2 dargestellt.

Der Kern des Systems ist der Component Store, der für die persistente Speicherung der Komponenten sorgt. In der ersten Implementierung des WebComposition System basiert der Component Store auf einem relationalen Database Management System (RDBMS). Das Prototyping-Modell, das dem Component Model zugrunde liegt, läßt sich sehr einfach in einer relationalen Datenbank implementieren, so daß trotz des objektorientierten Ansatzes kein OODBMS für die Datenhaltung benötigt wird. Zu den Diensten des Component Store gehören neben Persistenz- und Konsistenzhaltung des Component Model auch Dienste wie Revisionsmanagement, um die Entwicklungsgeschichte jeder Komponente aufzuzeichnen.

Der Component Server ermöglicht Werkzeugen und Anwendungen den geordneten Zugriff auf das im Component Store gehaltene Component Model. Der Zugriff auf Komponenten kann im Rahmen der Wartung erfolgen, um explizite Änderungen an einer Anwendung vorzunehmen, oder aus der Anwendung selbst, indem durch den Web-Server

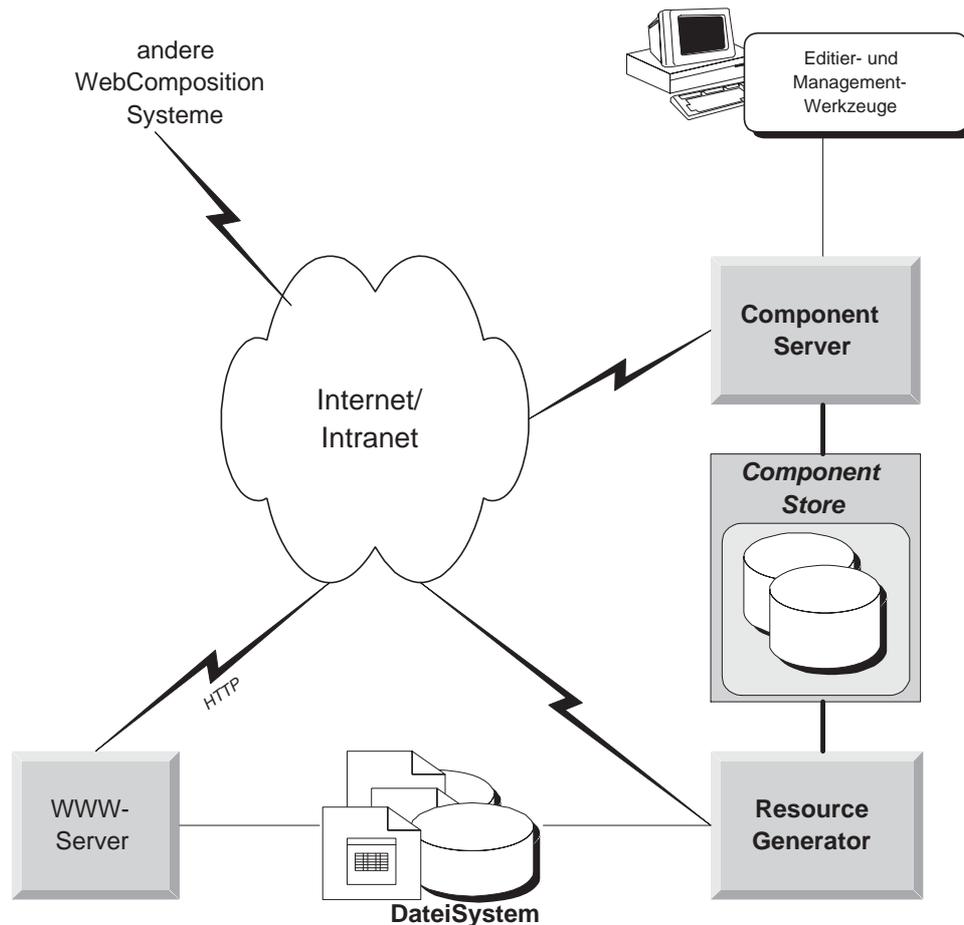


Abbildung 2: Das WebComposition System

Prozesse angestoßen werden, die mit dem Component Store kommunizieren. Der Zugriff auf das eigene Komponentenmodell einer Web-Anwendung kann z.B. dazu genutzt werden, um Komponenten dynamisch zu erzeugen. Das WebComposition System fungiert somit nicht nur als Repositorium in einer Entwicklungsumgebung sondern auch als im Betrieb manipulierbare Anwendungsplattform.

Der Resource Generator des WebComposition System realisiert die Integration des Component Model mit dem eigentlichen Web, in dem eine Anwendung schließlich abläuft. Der Dienst des Resource Generator ist die selbständige Abbildung einer im Modell beschriebenen Anwendung auf die Ressourcen im Dateisystem, auf denen Standard-Web-Server operieren können. Das Dateisystem fungiert dabei quasi als Cache zwischen Web und Component Model. Die Abbildung erfolgt durch die Nutzung des Darstellungsdienstes von Komponenten und unter Berücksichtigung der zwischen den Komponenten bestehenden Abhängigkeiten, die durch Prototyp-Instanz-Beziehungen und Revisionen auftreten. Der Resource Generator ermöglicht neben der Erzeugung der gesamten Anwendung auch die inkrementelle Erzeugung von Teilen einer Anwendung. Die inkrementelle Erzeugung kann durch eine benutzergesteuerte (z.B. nach Wartungsarbeiten), ereignisgesteuerte (z.B. bei Eintreffen einer Bedingung) oder zeitgesteuerte (z.B. zur regelmäßigen Aktualisierung einer Implementierung bei fortlaufender Weiterentwicklung am Component Model) Aktivierung angestoßen werden. Durch die ereignisgesteuerte Erzeugung (hier bei dem Ereignis der Änderung) von Ressourcen läßt sich

beispielsweise erreichen, daß dynamische Ressourcen nicht mehr bei jedem Benutzerzugriff erzeugt werden müssen, sondern zwischen tatsächlich auftretenden Veränderungen statisch vorgehalten werden können.

Das WebComposition System ist ein Unterstützungssystem für das Web Engineering. Es ist keine integrierte Entwicklungsumgebung sondern eine Plattform, die Werkzeugen und Anwendungen einen feingranularen Zugriff auf Web-Anwendungen während ihres gesamten Lebenszyklus ermöglicht. Die tatsächliche Arbeit mit dem WebComposition System sollte durch Editoren zur Manipulation von Komponenten unterstützt werden, darüberhinaus ist die Integration existierender Werkzeuge sinnvoll.

5 Stand der Arbeiten

Ein erster Prototyp des WebComposition Systems mit Revisionskontrolle und Mehrbenutzermanagement wurde implementiert und kam zur Erstellung interner Web-Seiten zum Einsatz. Der Prototyp arbeitet in einem heterogenen Umfeld basierend auf Windows NT und Unix. Die plattform-abhängigen Systemkomponenten sind ein Web-Server unter Unix und der Component Store, der auf dem Microsoft SQL-Server realisiert wurde. Der Component Store unterstützt neben dem Zugriff auf die Komponenten, Revisionskontrolle, Mehrbenutzermanagement und ein einfaches Transaktionsmanagement zur Konfliktbehandlung beim gleichzeitigen Zugriff auf Komponenten. Der Component Server und Resource Generator wurden in Java implementiert, wobei der Zugriff auf das DBMS mit JDBC/ODBC realisiert wurde. Darüber hinaus wurden die ersten Erfahrungen aus WebComposition für den Betrieb und die Wartung des an Ressourcen sehr umfangreichen Baden-Württemberg-Servers eingebracht. Die Entwicklung dynamischer Web-Anwendungen auf Basis eines an das WebComposition Component Model angelehnten Modells wurde im Projekt MobilBot mit Daimler-Benz Research erfolgreich erprobt [Gaed97]. Für den weiteren Projektverlauf ist eine Validierung von dem gesamten System geplant, die im Rahmen eines Projektes mit HP durchgeführt werden soll.

Literatur

- [fSAa95] National Center for Supercomputing Applications at the University of Illinois at Urbana-Campaign. NCSA: Tutorial on Server-Side Includes. <http://hoohoo.ncsa.uiuc.edu/docs/tutorials/includes.html>. September 1995.
- [Gaed97] Martin Gaedke. ActiveX: Componentware im Internet. *Theorie und Praxis der Wirtschaftsinformatik* 34(197), September 1997.
- [Gell97] Hans-W. Gellersen. Web Engineering: Softwaretechnik für Anwendungen im World-Wide Web. *Theorie und Praxis der Wirtschaftsinformatik* 34(196), Juli 1997.
- [GeWG97] Hans-W. Gellersen, Robert Wicke und Martin Gaedke. WebComposition: An Object-Oriented Support System for the Web Engineering Lifecycle.

In *Proc. of the 6th International World-Wide Web Conference (WWW6)*, Santa Clara, April 7–12 1997.

- [HaSc94] F. Halsz und M. Schwartz. The Dexter Hypertext Reference. Februar 1994.
- [IsSB95] T. Isakowitz, T. Stohr und A. Balasubramanian. RMM: A Methodology for Structured Hypermedia Design, in *Communications of the ACM*. August 1995.
- [Lerd98] R Lerdorf. Server Parsed Scripting language with RDBMS support. <http://modules.apache.org/>. Februar 1998.
- [Micr98a] Microsoft. FrontPage Home Page. <http://www.microsoft.com/frontpage>. Februar 1998.
- [Micr98b] Microsoft. Using Active Server Pages with Microsoft Internet Information Server. <http://www.microsoft.com/iis>. Februar 1998.
- [NeXT96] Inc. NeXT Software. WebObjects Developer's Guid. Januar 1996.
- [UnSm91] D. Ungar und R. B. Smith. SELF: The Power of Simplicity. *LISP and symbolic Computation: An International Journal* 4(3), 1991.

Anwendung von Neuronalen Netzen und Fuzzy-Logik in der ATM-Verkehrslastregelung

Meng Gan

Kurzfassung

Die Verkehrslastregelung in ATM-Netzen hat die Aufgabe, die Netzwerke und Teilnehmer zu schützen, um dadurch die vordefinierte Dienstgüte hinsichtlich der Leistung des Netzwerks zu erreichen. Wegen der Vielfalt der Verkehrscharakteristiken ist eine solche Aufgabe in ATM-Netzen nicht leicht zu lösen. Neuronale Netze stellen Lerneigenschaften und Anpassungsfähigkeiten dar, die besonders geeignet für die Approximierung von komplexen Systemen ohne genaue mathematische Modelle sind. Mit der Verwendung von Fuzzy Logik hat man noch zusätzlich die Möglichkeit, die ungenaue oder unvollständige, möglicherweise linguistische Informationen durch Fuzzy-Mengen präzise darzustellen. Die Anwendung von Neuronalen Netzen und Fuzzy Logik in ATM-Netzen, insbesondere bei der Verkehrslastregelung, gewinnt immer mehr Bedeutung.

1 Einleitung

ATM wurde als Übermittlungsprinzip für Breitband-ISDN entwickelt. Dabei hat man besonderen Wert darauf gelegt, die Menge der Dienst- bzw. Verkehrsarten gegenüber konventionellen Netzen beträchtlich zu erweitern, insbesondere betrifft dies Dienste mit unterschiedlicher, fester Bitrate und Datenströme mit variabler Datenrate. Dazu braucht man eine geeignete Verkehrslastregelung, um eine optimale Ausnutzung von Netzwerk-Ressourcen zu erreichen und die Überlastung des Netzwerks zu vermeiden.

Die Ziele der ATM-Verkehrslastregelung lassen sich wie folgt zusammenfassen [Pryc96]:

- **Flexibilität:** Sie sollte eine Reihe von ATM-vermittelten Dienstqualitätsklassen für alle Dienste unterstützen.
- **Einfachheit:** Die Herausforderung ist die Realisierung einer einfachen ATM-vermittelten Verkehrslastregelung, die die Komplexität von Netzwerkgeräten reduziert und die Kapazität von Netzwerk effizient ausnutzt.
- **Robustheit:** Die Ressourcen müssen bei allen Belastungsintensitäten optimal ausgenutzt werden, aber die Implementierung von Verkehrslastregelung muß einfach bleiben.

Das Erreichen obiger Ziele setzt zwei Funktionen bei der Verwaltung und Regelung der Verkehrslast in ATM-Netzwerken voraus: Regelung der Verbindungsaufnahme (Connection Admission Control, CAC) und Überwachung der Benutzungsparameter (Usage Parameter Control, UPC). CAC besteht aus einer Reihe von Maßnahmen, die vom Netzwerk während des Verbindungsaufbaus ergriffen werden, um eine ATM-Verbindung aufzunehmen oder abzulehnen. UPC besteht aus einer Reihe von Maßnahmen, die vom Netzwerk ergriffen werden, um den Durchsatz in einer ATM-Verbindung zu überwachen und zu steuern.

Viele Lösungen sind zur ATM-Verkehrslastregelung vorgeschlagen worden [Toy96], aber es ist nicht einfach, eine leistungsfähige Verkehrslastregelung wegen der Vielfalt der Verkehrscharakteristiken aufzubauen. Jeder Benutzer hat unterschiedliche Charakteristiken von Zellenerzeugung und jeder Dienst hat verschiedene Anforderungen für Dienstgüte wie CLR (Cell Loss Ratio), CTD (Cell Transfer Delay) und CDV (Cell Delay Variation). Um ein praktisches System zu entwerfen, wird es auf verschiedenen Annahmen und vereinfachten Verkehrsmodellen beruhen. Solche Modelle basieren meistens auf traditionellen mathematischen Berechnungen und Rechner-Simulationen. Im Gegensatz dazu bieten Neuronale Netze und Fuzzy Logik viele mächtige Methoden und Werkzeuge an, um eine optimale, effiziente Verkehrslastregelung zu entwerfen.

1.1 Neuronale Netze

Künstliche neuronale Netze sind den Informationsverarbeitungs- und Speicherungsmechanismen im Gehirn nachgebildet. Ein neuronales Netz besteht aus vielen Neuronen aus, die miteinander über gewichtete Verbindungen verknüpft werden können. Die Verarbeitung der Signale geschieht parallel in allen Neuronen, damit können neuronale Netze eine sehr hohe Leistung erreichen. Das Wissen eines Netzes steckt in den Gewichten der Verbindungen, die in einer Lernphase eingestellt werden. Die wesentliche Aufgabe neuronaler Netze ist es, bei einer bestimmten vorgegebenen Zielfunktion und vorhandenen Daten ihre Systemstruktur so zu verändern, daß die Zielfunktion erfüllt wird. Ein großer Vorteil von neuronalen Netzen ist, daß solche Zielfunktionen beliebige nichtlineare Funktionen sein können, die besonders sinnvoll bei der Verkehrslastregelung in ATM-Netzen sind.

Die Adaption der Netzwerkstruktur erfolgt durch die Variation der Kopplungsgewichte zwischen den Neuronen. Die richtige Auswahl der Lernregel stellt ein außerordentlich weites Feld zur Systemstrukturierung dar. Man verwendet einfache, problemunabhängige Algorithmen, die die Gewichte in der Lernphase geeignet anpassen [Zimm95, TsUh97].

1.2 Fuzzy-Logik

Fuzzy-Logik ist eine Technologie, die menschliche Erfahrung und Intuition zur Problemlösung nutzt. Vorteile sind vor allem dann vorhanden, wenn die mathematische Modelle komplex und rechenaufwendig sind, nichtlinearitäten existieren, oder sich Prozeßparameter ändern. Fuzzy Logik erweiterte die gewöhnliche Mengenlehre um die Beschreibung und Verknüpfung "unscharfer" Mengen, wobei die Zugehörigkeit von Elementen zu Mengen graduell sein kann.

Ein Fuzzy-System besteht aus vier Komponenten, nämlich Fuzzifizierer, Regelbasis, Inferenzmaschine und Defuzzifizierer. Die Aufgabe des Fuzzifizierers ist die Abbildung von scharfen Werten auf eine linguistische Werteskala. Die Regelbasis hat eine endliche Menge von linguistischen Regeln der Form *if Menge von Bedingungen then Menge von Konsequenzen*. Für jede Eingabe gibt es für jeden Regel einen Erfülltheitsgrad, der von der Inferenzmaschine bestimmt wird. Letztlich berechnet der Defuzzifizierer aus den unscharfen Größen, die die Inferenzmaschine liefert, einen scharfen Ausgangswert [Zimm95, TsUh97].

2 Neuronale Netze in ATM-Netzen

Die Anwendung von neuronalen Netzen in ATM-Netzen kann man seit Jahren in der Literatur finden [YuAn94, DoDe97, PaLe95]. Die wesentlichen Vorteile der Verwendung neuronaler Netze in ATM-Netzen sind:

- Es ist nicht erforderlich, die Charakteristiken von Benutzer und Diensten zu kennen. Die NN-basierte Verkehrslastregelung hat dann die Fähigkeit, sich für neue Benutzer und Dienste selbst anzupassen.
- Verkehrslastregelung mit neuronalen Netzen kann mit VLSI-Chips implementiert werden. Mit den inhärenten Eigenschaften von paralleler Verarbeitung ist es sehr geeignet für Hochleistungskommunikation.

2.1 NN-basierte Zugangssteuerung

CAC (Connection Admission Control) mit neuronalen Netzen ist eine der ersten Anwendungen von neuronalen Netzen in ATM-Netzen. [Hira90] benutzte ein 3-Schichten Back-Propagation Neuronales Netz, um eine optimierte CAC zu erreichen. Simulationen zeigten, daß die CLR (Cell Loss Rate) viel kleiner ist als die, die durch traditionelle Methoden erreicht werden kann. In [HaTY97] wurde eine sogenannte NNCAC vorgeschlagen, die zuerst ein neuronales Netz mit Musterdaten von einem Warteschlangenmodell trainiert. Die trainierte NNCAC wird dann zur Abschätzung der Zellenverzögerung und CLR benutzt, um den entsprechenden Verbindungswunsch zu akzeptieren oder nicht.

2.2 NN-basierte Verkehrssteuerung und -Vorhersage

In [TaHS94] wurde eine Verkehrssteuerung mit 2 miteinander verbundener neuronalen Netzen dargestellt, um die Zustände von Überlastung zu überwachen. Mit der inhärenten Fähigkeit von "self-learning" kann diese Verkehrssteuerung schnell zum optimalen Zustand kommen und sich an neue Situationen in Netzen anpassen. Viele Flußsteuerungsmechanismen brauchen Verkehrsvorhersagen, um eine schnelle und genaue Regelung durchzuführen. Neuronale Netze wie MLP (Multilayer Perceptrons) werden für diesen Zweck am meisten verwendet. [MCCL95] stellte eine der möglichen Ansätze dar.

2.3 Andere Anwendungen in ATM-Netzen

Man kann die neuronalen Netze auch in anderen Gebieten der Hochleistungskommunikation einsetzen. Einige davon sind: Switching [PaCL94], Routing [GeGh97], Ressourcen Management [FBHB97], Netzwerk-Topologie [FaDD97], usw.

3 Fuzzy-Logik in ATM-Netzen

Fuzzy-Logik ist sehr geeignet für eine Entscheidungsfindung mit ungenauen Informationen. In ATM-Netzen gibt es auch viele Ungenauheiten, wie Mean Bit Rate während der Verbindungsaufbaus. Die wesentlichen Vorteile von Verwendung der Fuzzy-Logik in ATM-Netzen sind:

- Fuzzy-Logik kann einen elastischeren Entscheidungsfindungsprozeß anbieten, indem sie die menschliche Inferenz simuliert. Mit der Veränderung von sogenannten Membership-Funktionen der Fuzzy-Variable kann man die Entscheidungsfindung verbessern.
- Fuzzy Inferenz kann auch durch VLSI-Chips implementiert werden, damit ist sie auch geeignet für die Hochleistungskommunikation.

3.1 Fuzzy-basierte Zugangssteuerung

[UeHi97] stellte eine Fuzzy-Logik-basierte CAC Methode vor, die die beobachteten Daten von CLR in einem Fuzzy-Inferenz-Schema benutzt. Sie haben noch die Fähigkeit von "self-learning", indem die weiteren beobachteten Daten von CLR für eine Anpassung der Regelbasis benutzt werden. In [HSMN⁺97] wurde eine Fuzzy-CAC mit der Fähigkeit der Auswertung der sogenannten "äquivalenten Bandbreite" an Hand von PCR (Peak Cell Rate), SCR (Sustainable Cell Rate) und MBS (Maximum Burst Size) vorgeschlagen. Die Fuzzy-CAC von [RaSc96] benutzt noch Genetische Algorithmen, um die Membership-Funktionen von Fuzzy Variablen zu verfeinern.

3.2 Fuzzy-basierte Verkehrssteuerung

In [CFPP96] wurde ein Fuzzy-Regelungsmechanismus definiert. Die Eingabe der Fuzzy-Steuerung sind die angekommenen Zellen nach der Aufnahme der Verbindung, die angekommenen Zellen innerhalb einem bestimmten Zeit (Fenster) und die erlaubten Zellen (N) im nächsten Fenster. Die Ausgabe ist die notwendige Änderung von N . Ein Vergleich zwischen Fuzzy-basierter Regelung und Leaky Bucket und Exponential-Weighted-Moving-Average ist durchgeführt worden. Die Ergebnisse zeigten die Leistungsfähigkeit der Fuzzy-Methode. Weiterhin ist in [ACFP97] eine Implementierung mit einem Fuzzy VLSI-Chip präsentiert worden. Ein Flußsteuerungsschema wurde in [PiSR97] vorgestellt. Die sogenannte FERM (Fuzzy Explicit Rate Marking) wird mit einem Fuzzy-Logik-Kontroller berechnet und an die ABR-Quelle in RM-Zellen zurückgesendet. Die Experimente zur Leistungsbewertung haben gezeigt, daß FERM eine bessere Leistung als EPRCA (Enhanced Proportional Rate Control Algorithm) erreichen kann.

3.3 Andere Anwendungen in ATM-Netzen

Die Verwendung von Fuzzy-Logik in der Hochleistungskommunikation kann man auch in weiteren Gebieten sehen. Eine Möglichkeit davon ist die Abbildung von Parameter zwischen Endbenutzer und Netzwerken. Die Endbenutzer von Anwendungen in ATM-Netzen, wie Video On Demand, können kaum genau wissen, welche Parameter wie CLR, PCR und SCR sie eingeben sollten. Sie wollen gerne nur einige direkte Parameter wie Rahmen/Sekunde einstellen. Fuzzy-Logik ist sehr geeignet für eine solche Abbildung. Weitere Anwendungsgebiete wie Ressourcen-Verteilung in ATM-Netzen oder drahtlosen Netzwerken sind auch denkbar.

4 Ausblick

Die meiste Forschungsarbeiten in diesem Bereich konzentrieren sich auf analytische Modelle und Simulationen. [ACFP97, CFPP96] hat aber auch eine Implementierung von einem Fuzzy-Traffic-Policing Mechanismus mit einem VLSI-Chip realisiert. Mit der Hilfe von ASIC-Technologie oder vorhandenen neuronalen und Fuzzy Chips könnten weitere Algorithmen durch Hardware implementiert werden.

Die gemischten Systeme von Fuzzy Logik, Genetische Algorithmen und Neuronalen Netzen haben immer mehr Bedeutung, da solche Systeme die Vorteile von einzelnen Technologien miteinander kombinieren können.

Viele Aspekte von neuronalen Netzen und Fuzzy Logik, wie systematischer Entwurf, Stabilität und Konvergenz, sind in dem Anwendungsbereich von ATM-Netzen noch nicht ausführlich behandelt worden. Mit Berücksichtigung solcher Aspekte, versucht man, effizientere, robustere und schnellere Algorithmen zu entwickeln.

Literatur

- [ACFP97] G. Ascia, V. Catania, G. Ficili und S. Palazzo. A VLSI Fuzzy Expert System for Real-Time Traffic Control in ATM Networks. *IEEE Transactions on Fuzzy System* 5(1), 1997, S. 20–31.
- [CFPP96] V. Catania, G. Ficili, S. Palazzo und D. Panno. A Comparative Analysis of Fuzzy Versus Conventional Policing Mechanisms for ATM networks. *IEEE/ACM Transaction on Network* 4(3), 1996, S. 449–459.
- [DoDe97] C. Douligeris und G. Develekos. Neuro-Fuzzy Control in ATM Networks. *IEEE Communications Magazine*, Mai 1997, S. 154–162.
- [FaDD97] H. Fahmy, G. Develekos und C. Douligeris. Application of Neural Networks and Machine Learning in Network Design. *IEEE JSAC* 15(2), 1997, S. 226–237.
- [FBHB97] A. Farago, J. Biro, T. Henk und M. Boda. Analog Neural Optimization for ATM Resource Management. *IEEE JSAC* 15(2), 1997, S. 156–163.

- [GeGh97] E. Gelenbe und A. Ghanwani. Improved Neural Heuristics for Multicast Routing. *IEEE JSAC* 15(2), 1997, S. 147–155.
- [HaTY97] J. Hah, P. Tien und M. Yuang. Neural-Network-Based Call Admission Control in ATM Networks with Heterogeneous Arrivals. *Computer Communications* Band 20, 1997, S. 732–740.
- [Hira90] A. Hiramatsu. ATM Communications Network Control by Neural Networks. *IEEE Transactions on Neural Network* 1(1), März 1990, S. 122–130.
- [HSMN⁺97] H. Hellendoorn, R. Seising, W. Metternich, M. Nissel und C. Thomas. Verkehrslastregelung in ATM-Netzwerken mit Fuzzy-Methoden. *Informatik Forschung und Entwicklung*, Dezember 1997, S. 23–29.
- [MCCL95] W. Moh, M. Chen, N. Chu und C. Liao. Traffic Prediction and Dynamic Bandwidth Allocation over ATM: a Neural Network Approach. *Computer Communication* 18(8), 1995, S. 563–571.
- [PaCL94] Y. Park, V. Cherkassky und G. Lee. Omega Network-based ATM Switch with Neural Network Controlled Bypass Queueing and Multiplexing. *IEEE JSAC* 12(9), 1994, S. 1471–1480.
- [PaLe95] Y. Park und G. Lee. Applications of Neural Networks in High Speed Communication Networks. *IEEE Communication Magazine*, Oktober 1995, S. 68–74.
- [PiSR97] A. Pitsillides, Y. Sekercioglu und G. Ramamurthy. Effective Control of Traffic Flow in ATM Networks Using Fuzzy Explicit Rate Marking. *IEEE JSAC* 15(2), 1997, S. 209–225.
- [Pryc96] M. Prycker. *Asynchronous Transfer Mode*, Kapitel 7, S. 341–384. Prentice Hall. 1996.
- [RaSc96] M. F. Ramalho und E. M. Scharf. Fuzzy Logic Tool and Genetic Algorithm for CAC in ATM Network. *Electronics Letters* 32(11), 1996, S. 973–974.
- [TaHS94] A. Tarraf, I. Habib und T. Saadawi. A Novel Neural Network Traffic Enforcement Mechanism for ATM Networks. *IEEE JSAC* 12(6), 1994, S. 1088–1096.
- [Toy96] M. Toy (Hrsg.). *Development and Applications of ATM*, Kapitel 3, S. 157–302. Institute of Electrical and Electronics Engineers, Inc. 1996.
- [TsUh97] L. Tsoukalas und R. Uhrig. *Fuzzy and Neural Approaches in Engineering*. John Wiley & Sons Inc. 1997.
- [UeHi97] K. Uehara und K. Hirota. Fuzzy Connection Admission Control for ATM Networks Based on Possibility Distribution of Cell Loss Ratio. *IEEE JSAC* Band 15, 1997, S. 179–190.
- [YuAn94] B. Yuhua und N. Ansari. *Neural Networks in Telecommunications*. Kluwer Academic Publishers. 1994.
- [Zimm95] H. Zimmermann (Hrsg.). *Neuro+Fuzzy Technologien und Anwendungen*. VDI Verlag. 1995.

Internet Call Center – Innovative Assistenzdienstleistungen durch CTI auf dem Information Highway

Hans-Werner Gellersen

1 Motivation und Zielsetzung

Ziel des Forschungsvorhabens *Internet Call Center* ist die Entwicklung einer Infrastruktur für innovative Assistenzdienstleistungen, die durch eine Verschmelzung von Beratung und Information gekennzeichnet sind. Informationsdienste und Beratungsdienste sind Extreme in einem Kontinuum der Problemlösungsunterstützung für Kunden, wobei Informationsdienste Kosteneffizienz und eigenständige asynchrone Informationsbeschaffung betonen, während bei Beratungsdiensten die menschliche Komponente der Unterstützung, sowie der inkrementelle, kooperative Dialog zur Lösung auch komplexerer und schwieriger spezifizierbarer Probleme im Vordergrund steht (vgl. Abb. 1).



Abbildung 1: Charakteristika von Informations- und Beratungsdiensten

Die Verschmelzung von Informations- und Beratungsdiensten hat das Potential, existierende Dienste zu verbessern (sowohl hinsichtlich Kosteneffizienz als auch Dienstgüte) und neue Dienstleistungsformen zu schaffen:

- Mehrwert-Informationsdienste durch Assistenz bei der Informationsbeschaffung; damit verbunden Steigerung der Dienstgüte und Individualisierung der Kundenunterstützung (unter Betonung der „menschlichen Beratungskomponente“ in klarer Abgrenzung zu agentenbasierten Ansätzen)

- Kosteneffizientere Beratungsdienste durch Auslagerung weniger erklärungsbedürftiger Teilvorgänge in Informationsdienste (z.B. Ersetzen langer Beratungssitzungen durch Vorgänge, die sich aus kürzeren Beratungseinheiten und asynchroner Informationsbereitstellung zusammensetzen, z.B. einleitende Beratung, anschließender Informations-Push zum Kunden und in der Folge Wiederaufnahme der Beratungssitzung nur nach Bedarf)
- Enge Verzahnung von eigenständiger Informationsbeschaffung und Beratung durch Assistenten/ Experten durch Schaffung eines gemeinsamen Kontextes (Beratungsdienste können den Kontext der Informationsbeschaffung z.B. zur Auswahl geeigneter Assistenten nutzen)
- Schaffung eines Migrationspfads in der Kundenbetreuung von kostenintensiver Beratung hin zu effizienterer eigenständiger Informationsbeschaffung (z.B. im Kontext der Beratung Kunden sukzessive zur eigenen Informationsbeschaffung anleiten)

Im Bereich Telekooperation oder weiter gefaßt CSCW (Computer-Supported Cooperative Work) gibt es Ansätze, synchrone Zusammenarbeit in asynchrone Vorgänge (Workflows) einzubetten. Die Entwicklung von CSCW-Technologien läßt dabei aber bislang ganz außer Betracht, daß Beratungsdienste heute über fortgeschrittene Call Center-Technologien der Telefonie erbracht werden, die CSCW-Werkzeugen für synchrone Zusammenarbeit, d.h. A/V- und Anwendungskonferenzsystemen, in vielerlei Hinsicht weit überlegen sind (Ubiquität der Telefonie, Beherrschung hoher Gesprächsaufkommen, wirtschaftliche Bedeutung). Call Center-Technologien selbst basieren auf Computer-Telefon-Integration (CTI) und umfassender Rechnerunterstützung haben sich aber historisch unabhängig von Informatiktrends wie CSCW und Internet entwickelt. Die Rechnerunterstützung bezieht sich allein auf die effiziente Handhabung hoher Gesprächsaufkommen mit Technologien wie z.B. automatischer Anrufverteilung auf Operatorplätze (Automatic Call Distribution, ACD) oder interaktiver Menüführung mit Spracherkennung (Interactive Voice Response, IVR) und bietet bis heute keine Ansätze für die Integration mit Diensten in Rechnernetzen.

2 Lösungsansatz

Die Umsetzung der beschriebenen Zielsetzung erfordert zwei Arbeitsschwerpunkte:

- Entwicklung einer neuen Dimension der CTI zur Integration mit dem „Information Highway“
- Technologie- und Betreibermodelle für innovative Assistenzdienstleistungen

2.1 Computer-Telefon-Integration

Voraussetzung für die Verschmelzung mächtiger Beratungsdienste mit Informationsdiensten zu neuen Dienstleistungsformen ist die Integration von Call Center-Technologien in globale Informationsinfrastrukturen, was de facto Integration mit

Internet-Technologien bedeutet. Obwohl die Computer-Telefon-Integration ein Bereich ist, der sich bereits seit den 60er-Jahren kontinuierlich entwickelt, ist bisher noch nicht der Schritt vollzogen worden, CTI-Netze, d.h. über CTI-Systeme (z.B. ACD-Systeme) gekoppelte Rechner weitergehend mit Rechnernetzen zu verbinden. Bisherige CTI-Systeme bieten in aller Regel noch keine Gateways zu LANs. Eine neue CTI-Hardwaregeneration wird PC-basiert sein und kleine Call Center-Lösungen auf Basis von PC-Einschubkarten ermöglichen. Auch wenn das Produktmarketing dazu bereits mit einer Internet-Integration wirbt (vgl. z.B. Hicom Integrated von Siemens), kann nicht übersehen werden, daß noch grundlegende Arbeiten zur Schaffung von Modellen und Abstraktionen für die Entwicklung neuer CTI-Anwendungen erforderlich sind. Internet-Integration ist bisher tatsächlich auf die sehr einfache Möglichkeit beschränkt, den Aufbau von Calls aus Internetdiensten heraus durch Knopfdruck in Webseiten zu initiieren (z.B. Produkte der Firmen Lucent, Genesys und NetCall). Hieraus ergibt sich für das Vorhaben ein Handlungsbedarf zur Entwicklung von CTI-Komponenten zur Integration mit Rechnernetzen und Diensten in Rechnernetzen. Dazu sind grundlegende Modelle zu definieren und Abstraktionen auf verschiedenen Ebenen zu realisieren, insbesondere „höhere“ Abstraktionen für die Anwendungsentwicklung wie z.B. Anwendungsschnittstellen für die Einbettung von Calls in Workflows.

2.2 Modelle für Assistenzdienstleistungen

Weitergehendes Ziel des Vorhabens ist die Entwicklung von innovativen Assistenzdiensten durch Verschmelzung von Informationsbereitstellung und Beratung. Abbildung 2 stellt das grundlegende Modell für innovative Assistenzdienste dar.

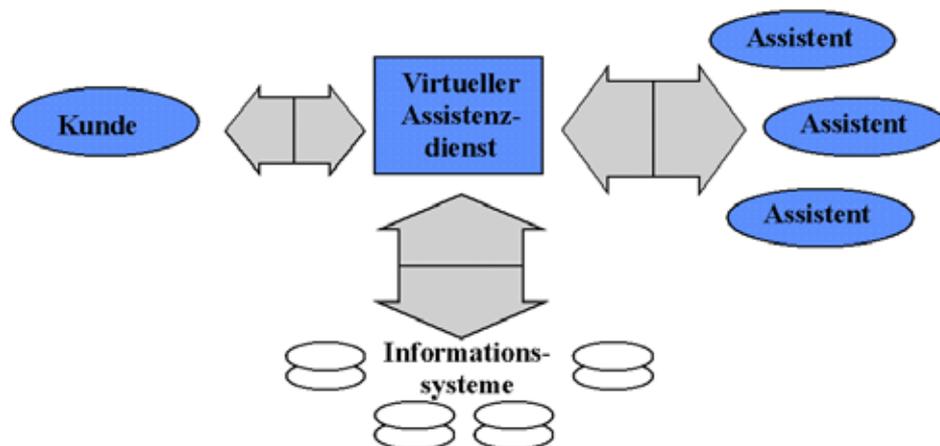


Abbildung 2: Modell für innovative Assistenzdienste.

Die wesentlichen Merkmale sind die Virtualisierung der Dienstleistung unter Bezugnahme auf sowohl Informationssysteme als auch Assistenten/Experten, wobei sowohl Systeme als auch Assistenten geographisch beliebig verteilt sein können. Das Modell ist bewußt allgemein gehalten und kann insbesondere sowohl für externe Assistenzdienste (z.B. Kundenbetreuung) als auch für interne Dienste (Workflows, Intranets) zugrundegelegt werden. Die Virtualisierung der Dienstleistung unter Nutzung sowohl von Informations- als auch Beratungsquellen kann ein Ansatzpunkt für die Verbesserung/Optimierung existierender Dienstleistungen sein (z.B. im Bereich der Kundenbetreuung, wie 1995 vom TecO in einer Studie für die SAP AG demonstriert, oder

im Bereich innerbetrieblicher Vorgangsbearbeitung). Darüberhinaus bietet sie aber Potential für innovative Dienstleistungen, wie z.B.:

- Assistenz-Brokering: Erweiterung und Ergänzung des Konzepts des Information-Brokering um das Brokering von Beratung
- Virtuelle Call Center: Zusammenschluß von Unternehmen zur gemeinsamen Vermarktung von Expertise unter Nutzung von Call Center-Technologien für Call Distribution auf verteilte Berater
- Expertenkonferenzen: High-end Assistenzdienste für Problemstellungen, die das Zusammenschalten verschiedener verteilter Experten erfordern, z.B. bei Fehlersuche in komplexen Systemen

Für verbesserte und vor allem neue Dienstleistungen sind sowohl Technologie- als auch Betreibermodelle zu entwickeln. Als Kern eines Technologiemoells ist eine offene Systemarchitektur zu entwickeln, die auf CTI-Komponenten aufbaut aber flexible Schnittstellen für weitere grundlegende Komponenten für den Dienstleistungsbetrieb bietet, z.B. für Abrechnungs- und Bezahlungskomponenten. Das Technologiemoell soll weiter eine Klassifikation entsprechender existierender Technologien umfassen sowie darauf aufbauende Konfigurationsleitfäden für verschiedene Dienstleistungsorganisationen. Neben technologischen Aspekten sind Betreiber- und Geschäftsmodelle zu untersuchen. Während das technologische Risiko für die Entwicklung neuer Assistenzdienste als vergleichsweise gering einzuschätzen ist, sind Fragen der Profitabilität und Akzeptanz sowohl seitens der Betreiber als auch seitens der Kunden kritisch, insbesondere im Kontext virtueller Organisationen.

3 Erwartete Ergebnisse

Die folgenden Ergebnisse werden erwartet:

- CTI-Modelle, -Komponenten, und -APIs für die Integration von Call Center-Technologien mit Internet-Technologien und -Diensten
- Offene Systemarchitektur für Internet Call Center mit Schnittstellen für die Integration insbesondere von Komponenten für elektronischen Handel die für den Dienstleistungsbetrieb relevant sind
- Technologiemoelle für verschiedene Dienstleistungsorganisationen
- Bibliotheken von Assistenzdienstfunktionen
- Exemplarische Demonstration verschiedener innovativer Assistenzdienste
- Analyse von Geschäfts- und Betreibermodellen

Bedarfsorientierte Informationsvermittlung in verteilten Trading-Systemen

Arnd Grosse

Kurzfassung

Die zunehmende arbeitsteilige Integration zwischen weiträumig verteilten Anwendungen bei gleichzeitiger Desintegration zuvor monolithischer Anwendungen in feingranularere (Dienst-)Objekte, erfordert eine systemtechnische Unterstützung, um zur Laufzeit flexibel auf das jeweils benötigte Dienstobjekt zuzugreifen. Diese Flexibilität wird in verteilten Systemen durch eine Vermittlungsschicht auf Basis verteilter Trading-Dienste bereitgestellt. Ein Mangel dieser Lösung stellt die bestehende unzureichende Integration einzelner Trading-Systeme zu einem Verbund kooperierender Trader dar, um hierdurch bereichsübergreifend anwendungsorientiert eine durchgängige effiziente Vermittlungsfunktion anzubieten. Die folgende Arbeit beschreibt das *Scalable Object Flow Tree (SOFT)-Konzept*, welches ein Lösungsmodell für die genannte Problematik anbietet.

1 Modularisierung monolithischer Anwendungen

Ehemals riesige, monolithisch aufgebaute und zumeist isoliert ablaufende Softwareanwendungen werden zukünftig durch modular gestaltete und miteinander über große Distanzen kooperierende *Anwendungskomponenten* ersetzt. Dahinter steht der Anspruch, eine stärkere Verknüpfung der einzelnen Aufgabenbereiche von z.B. räumlich verteilten Unternehmensbereichen zu einem integrierten Verbund zu erreichen als auch einzelne Komponenten flexibel wiederzuverwenden [BBKL⁺97].

Da aufgrund der dann gegebenen Komponentenverteilung die resultierendem Anwendungsverbunde sehr komplex und zunehmend unternehmenskritisch sind, werden hohe Anforderungen an eine flexible Konfiguration und an eine schnelle Aktualisierung einzelner Programmteile gestellt. Die Komponenten unterliegen dabei zudem kontinuierlichen Änderungen und Anpassungen. Dieses kann soweit führen, daß eine Aktualisierung einer Komponente während des Betriebs möglich sein muß, ohne den gesamten Anwendungsverbund oder eine einzelne Teilapplikation herunterzufahren [LaVe97].

Solcherart in Teilfunktionen aufgebrochene Anwendungskomponenten werden heutzutage zu Bibliotheken zusammengefaßt und an die unterschiedlichen Applikationslokalitionen verteilt [Rost97]. Um dabei eine hohe Wiederverwendung einzelner Softwarekomponenten zu erzielen, dürfen die Komponenten eine kritische Größe nicht unterschreiten. So hat eine Untersuchung bei Hewlett-Packard [GrWo95] gezeigt, daß Bibliotheken

mit wenigen, dafür aber sehr großen Komponenten die besten Ergebnisse hinsichtlich der Wiederverwendungsrate erzielen. Da hierdurch die Größe einzelner Komponenten nach unten hin begrenzt ist, besitzen die einzelnen Komponenten den Umfang auch entfernt aufrufbarer Serverobjekte. Problematisch dabei ist, daß für die Komponenten ein kontinuierlicher Änderungsbedarf bestehen kann, und daß bei Änderungen an einer einzelnen Komponente die gesamte Bibliothek zentral zu ersetzen und neu zu verteilen ist. Dieses kann einen hohen und eventuell redundanten Kommunikationsaufwand zur Folge haben. Auch werden dabei unterschiedliche Versionen einzelner Komponenten oder die Verwendung verschiedener kontextbehafteter Komponenten in existierenden Systemen nicht unterstützt.

2 Flexibilisierung durch verteiltes Trading

Ein Lösung für die zuvor genannte Problematik stellt die Verwendung einer Vermittlungsschicht in verteilten Anwendungssystemen dar, in welcher die als *Dienstobjekte* modellierten Komponenten von Anwendungen bedarfsabhängig zur Laufzeit eingebracht bzw. angefordert werden können. Eine Flexibilisierung des Auswahlprozesses wird hierbei durch die Attributierung der Dienstobjekte erreicht, wodurch einzelne Anwendungen gezielt einen spezifischen Bedarf oder ein bestimmte Angebot unterbreiten können. Eine Systemfunktion, die einen solchen Vermittlungsdienst erbringt, wird als *Trader* bezeichnet. Die Definition eines Standards für eine solche Systemfunktion befindet sich derzeit kurz vor Abschluß durch die International Organization for Standardization (ISO) [JTC197].

Der in diesem Standard vorgesehene Trading-Dienst erlaubt das Anbieten (*Export*) von Dienstobjekten, den Zugriff (*Import*) darauf sowie die Zusammenarbeit zwischen einzelnen Tradern über Verbindungen (*Links*). Die Menge der bei einem Trader verwalteten Dienstangebote einschließlich des Trader wird als *Trading-Domain* bezeichnet. Einzelne Trading-Domains können unter einem gemeinsamen Ziel (*Policy*) zusammenarbeiten und bilden dann eine Trading-Gemeinschaft (*Trading Community*), den sog. Trader-Verbund (*Federation of Traders*). Dieser Trader-Verbund tritt nach außen als eine transparente Trading-Schicht für die Anwendungen auf und kann intern durch ein Trader-Netz modelliert werden. Die einzelnen Trader in einem solchen Netz unterliegen dabei gemeinsamen Regeln (*Community Rule*). Die Bereichsgrenzen der Community sind nicht festgelegt und können administrative, technologische oder sonstige Grenzen überspannen. Für weitere Details sei auf den Standardentwurf [JTC197] verwiesen.

Der Nachteil des geplanten Trading-Standards liegt in seiner losen Definition eines Trader-Netzes. Für die Verknüpfung und Zusammenarbeit der im Netz verbundenen Trader gibt es im Standard nur einen sehr rudimentären Ansatz zur manuellen Konfiguration der Trader-Links mit Hilfe eines Trading-Administrators. Für die Steuerung der Verbindungen wird dabei auf die Definition von entsprechenden Regeln verwiesen, wobei hier jedoch keine tiefergehenden bzw. anwendungsspezifischen Vorschläge unterbreitet werden. Obgleich dieser Mangel durch weitergehende Ansätze zur Trader-Zusammenarbeit teilweise aufgegriffen und angegangen wird (z.B. [BeRa91, Burg95, LeBe95, MeZP96]), existiert kein verteilter Trader-Ansatz, der gezielt auf anwendungsorientierte Bedürfnisse bzw. unternehmensspezifische Anforderungen eingeht.

3 SOFT-Konzept

Die nachfolgenden Unterkapitel stellen das SOFT-Konzept (Scalable Object Flow Tree-Konzept) vor, das eine Erweiterung zum angedachten Standard darstellt, um eine Informationsvermittlung zwischen vernetzten Tradern gezielt nach anwendungsabhängigen Bedürfnissen automatisch und zugleich effizient zu unterstützen. Hierzu wird das existierende Trading-Konzept in den nachfolgenden drei Bereichen integriert erweitert und damit die in [GHHK⁺97] beschriebene Flexibilität gewährleistet:

1. Trading sieht die Vermittlung von Dienstobjekten in Form von Servern vor. Daneben soll es einer Anwendung möglich sein, auch personalisierte Instanzen des Serverobjekts bei sich lokal instantiiieren zu können.
2. Es gibt eine Reihe unterschiedlicher Verfahren, um die Informationen zwischen kooperierenden Tradern zu vermitteln. Auf Systemebene sollte es möglich sein, unterschiedliche Anforderungen und Bedürfnisse einzelner Anwendung jeweils gezielt durch die effizienteste Vermittlungsform zu unterstützen.
3. Eine Anwendung ist in einen Unternehmenskontext eingebunden und besitzt zur Erbringung ihrer Funktionen Anforderungen hinsichtlich dem Vorhandensein bestimmter Informationsressourcen. Dieses muß im Vorfeld durch geeignete Konfiguration der Trader auf Anwendungsebene unterstützt werden.

3.1 Skalierbare Dienstobjekte

Der Nachteil des bestehenden Trader-Standards stellt seine Beschränkung der Informationsmenge auf den klassischen Client/Server-Ansatz dar, indem Angebote als Server instantiiert und von Anwendungen in der Rolle eines Client verwendet werden müssen. Ein Server exportiert ein Dienstobjekt an ein Vermittlungssystem, indem er eine weltweit eindeutige Dienstobjektreferenz, einen eindeutigen Bezeichner für den zugehörigen Dienstobjekttyp sowie eine Liste beschreibender Dienstigenschaften beim Trader registriert. Eine einzelne Eigenschaft wird hierbei durch ein $\langle \text{Name, Wert} \rangle$ -Tupel beschrieben [JTC197]. Ein Client kann das Dienstobjekt von dem Vermittlungssystem importieren und es danach über einen entfernten Aufruf verwenden. Vor der Registrierung eines Dienstobjekts muß dessen zugehöriger Dienstyp beim Trader bekanntgemacht worden sein. Dienstypen können dabei in eine Klassenhierarchie eingebunden sein. Für weitere Details sei auf die Referenz verwiesen.

Die SOFT-Architektur greift diesen Ansatz auf und erweitert ihn um die Möglichkeit, auch Implementierungen eines Dienstyps sowie Objektzustandsinformationen bei einem Trader registrieren und damit über diesen vermitteln zu können. Hierzu wird durch das SOFT-Konzept eine Aufteilung des zu vermittelnden Dienstobjekts in unterschiedliche Facetten des zugehörigen Dienstyps und daraus instantiiierter Dienstobjekte vorgenommen. Die Ausprägung einer einzelnen Facette wird im SOFT-Konzept als *Part* und die solcherart erweiterten Dienstobjekte, die sich aus Parts zusammensetzen, werden als skalierbare (Dienst-)Objekte (*Scalable Objects*) bezeichnet [GrKH97] (vgl. hierzu auch Abbildung 1).

Für ein skalierbares Objekt lassen sich drei zugehörige Facetten zu einem Dienstyp im Trading-Standard unterscheiden:

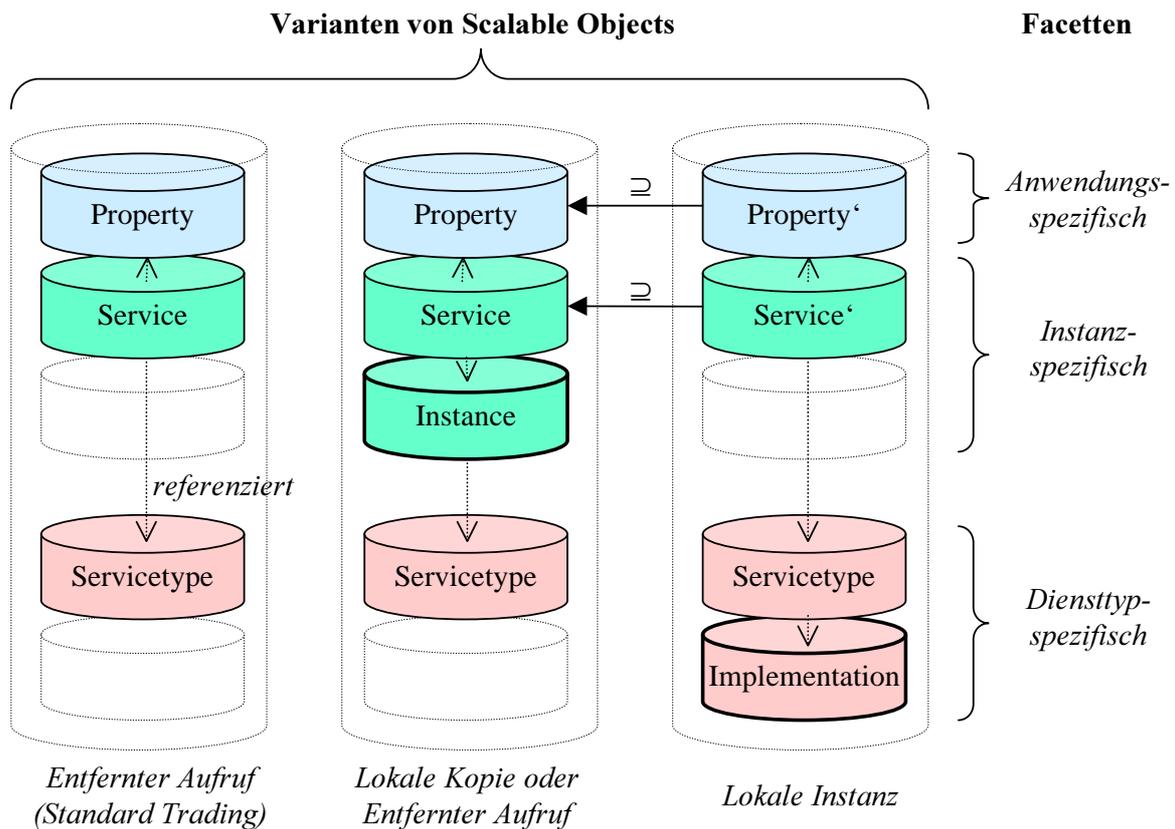


Abbildung 1: Skalierbare Objekte im SOFT-Konzept bestehend aus Parts

Diensttypspezifische Facette: Jedem Dienstobjekt ist eindeutig ein Diensttyp zugeordnet, der über einen Diensttypbezeichner (Service Type Name) referenziert wird. Der Bezeichner dient als Zeiger auf eine Struktur, die einen Verweis auf die Schnittstellenbeschreibung des Diensttyps in einem zugrunde liegenden Interface-Repository, die Bezeichner der Basisdiensttypklassen sowie die Bezeichner der Eigenschaftstypen des Diensttyps festlegt und muß demgemäß bei einem Export eines Dienstobjektes existieren. Da Diensttypen im Trader mittels eines Diensttyp-Verzeichnisses (Service Type Repository) verwaltet werden, ist somit über den Diensttypbezeichner jedem Dienstobjekt dieser (diensttypspezifische) Part durch den Trading-Standard zugeordnet. Durch Ein- und Austragen dieses Parts im Diensttypverzeichnis kann dieser dynamisch durch Anwendungen geändert werden, solange von diesem Diensttyp keine weiteren Diensttypen abgeleitet wurden. Der Part, der den Diensttyp und damit die Schnittstelle beschreibt, wird im SOFT-Konzept als *Servicetype-Part* bezeichnet und ist somit eine Ausprägung der diensttypspezifischen Facette.

Anwendungsspezifische Facette: Jeder Dienstanbieter kann sein Dienstangebot mit beschreibenden Eigenschaften versehen. Welche dieser Eigenschaften von dem Anbieter unbedingt vorzusehen sind, welche eingeschränkt als nur lesbar angegeben und welche gar optional sind, wird innerhalb des obigen Servicetype-Part vorgegeben. Der Part der anwendungsspezifischen Facette, der die beschreibenden Eigenschaften zu einem Dienstobjekt enthält, wird *Property-Part* bezeichnet und kann durch den Anbieter beim Trader dynamisch geändert werden.

Instanzspezifische Facette: Eine *Service-Part* genannte Ausprägung dieser Facette enthält die instanzspezifischen Informationen zu einem Dienstobjekt. Hierbei handelt es sich im Standard um die eindeutige Objektreferenz, unter der das Dienstobjekt aufgerufen wird, sowie den Dienstypbezeichner, der als Referenz auf den zugehörigen Servicetype-Part dient. Eine Änderung dieses Parts während der Lebenszeit des Dienstobjektes ist im Standard nicht möglich, da bei erneuter Instantiierung eines Dienstobjektes auch eine neue Objektreferenz durch das Laufzeitsystem erzeugt und zudem vom Trader eine neue Angebotsreferenz (OfferID) vergeben wird.

Ein Dienstangebot in Form eines Dienstobjektes bei einem Trader wird von einem Anbieter somit durch den Service-Part, den zugehörigen Property-Part und den referenzierbaren Servicetype-Part festgelegt. Eine eindeutige Zuordnung der Parts erfolgt über die vom Trader vergebene OfferID. Da sowohl Servicetype-Part als auch Property-Part dynamisch Änderungen unterliegen, muß dieses von einem Vermittlungssystem geeignet unterstützt werden, indem die Änderungen innerhalb des Systems propagiert werden oder sichergestellt wird, daß immer auf die aktuellen Daten zugegriffen wird.

Das SOFT-Konzept erweiterte diese Ausprägungen der Facetten um zwei weitere Parts, die mit Hilfe entsprechender SOFT-Trader vermittelt werden. Der erste Part wird *Implementation-Part* genannt und enthält eine Implementierung zu einem Dienstyp. Eine SOFT-Anwendung kann so eine Instanz des Dienstobjektes über eine erweiterte Importanweisung bei sich anlegen und darauf Operationen lokal aufrufen. Implementation-Parts werden über den Dienstypbezeichner referenziert und sind damit unabhängig von instantiierten Dienstobjekten. Sie können von einer Anwendung beliebig angepaßt werden, was zum Überschreiben der alten Version führt. Im Gegensatz hierzu stellt der zweite erweiterte Part, der sog. *Instance-Part*, eine Momentaufnahme des beim Trader registrierten Dienstobjektes dar (d.h. eine Zustandskopie). Eine SOFT-Anwendung kann hierdurch eine Kopie des entfernten Dienstobjektes bei sich anlegen und auf dieser Kopie lokal arbeiten, anstelle entfernte Aufrufe darauf zu tätigen. Das ist insbesondere bei schlechten Kommunikationsverbindungen zur Einsparung des Kommunikationsaufwands vorteilhaft oder für abgekoppelte Operationen sinnvoll, wie sie in [Kott96] beschrieben werden.

Der Unterschied des Instance-Part zum Implementation-Part liegt darin, daß so auch Dienstobjekte lokal einer Anwendung vermittelt werden können. Hierbei existiert im Fall kontextbehafteter Dienstobjekte das Problem der Definition von geeigneten Synchronisationspunkten innerhalb des Ausführungspfades des Dienstobjektes zum Abspeichern eines Zustands oder dem späteren konsistenten Zusammenführen von Berechnungsergebnissen. Diese Problematik wird in [Kott96] eingehend beschrieben und liegt nicht im Aufgabenbereich der Vermittlungsschicht, sondern ist Bestandteil der Anwendungsschicht, da hierzu das Wissen über die Anwendungssemantik notwendig ist. Während der Implementation-Part der diensttypspezifischen Facette zuzuordnen ist, überdeckt der Instance-Part sowohl die diensttypspezifische, die anwendungsspezifische als auch die instanzspezifische Facette. Die Besonderheit dabei liegt darin, daß sich die im Instance-Part enthaltene Implementierung von dem zuvor beschriebenen Implementation-Part unterscheiden kann.

Diese beiden erweiterten Parts ermöglichen somit beide eine lokale Instantiierung von Dienstobjekten bei einer Anwendung und unterstützen damit sowohl die in [Rost97]

beschriebene ungeplante Wiederverwendbarkeit als auch das in [OrHa97] definierte lokale und entfernte Verwendungsmuster für Serverobjekte. Eine (SOFT-)Anwendung legt gegenüber der Vermittlungsschicht fest, welches Verwendungsmuster von ihr benötigt bzw. priorisiert wird, und die so erweiterte Vermittlungsschicht stellt die hierfür notwendigen Parts der Anwendung zur Verfügung.

3.2 Verteilungsstrategien zwischen vernetzten Tradern

Wie aus dem vorangegangenen Unterkapitel deutlich wurde, läßt sich die zu einem Dienstobjekt gehörende Information in einzelne Parts zerlegen. Die einzelnen Parts werden je nach Verwendungskontext mehr oder weniger häufig von Anwendungen verwendet bzw. von den Anbietern geändert. Dieses muß für eine effiziente bedarfsorientierte Vermittlung durch ein verteiltes Trading-System berücksichtigt werden. Als alternative Strategien zur Verteilung in verteilten Systemen lassen sich hierbei entweder die aktive Weitergabe von Parts zwischen den Trader (*Push-Strategie*) oder das passive Verhalten des Part verbunden mit seiner bedarfsgetriebenen Weitergabe (*Pull-Strategie*) unterscheiden. Dazwischen liegende Lösungen bedienen sich eines Zwischenspeichers, dessen Konsistenz entweder durch Invalidierungsnachrichten oder durch Polling-Verfahren sichergestellt werden kann [ZhFJ97].

Im Rahmen des SOFT-Konzepts wurde zur Bewertung einer effizienten Verteilung und Aktualisierung von Parts in einem verteilten Trader-Netz ein analytisches Kostenmodell auf Basis des resultierenden Kommunikationsaufwands entwickelt [GrKo97, GrKH97]. Dieses Modell verwendet neben dem Verhältnis zwischen der Zugriffshäufigkeit und der Änderungshäufigkeit von Parts zusätzlich die jeweilige meßbare Part-Größe. Die Zufallsverteilung der Häufigkeiten wird mit Hilfe einer Poisson-Verteilung abgeschätzt. Auf Grundlage dieser Maßzahlen kann die jeweilige optimale Verteilungsstrategie zwischen der Push-Strategie, der Pull-Strategie und einer auf Zwischenspeichern basierten Strategie (*Lazy-Push-Strategie*) zwischen zwei Tradern analytisch festgelegt werden. Obgleich die volumenabhängigen Kommunikationskosten zwischen zwei Tradern mit in das analytische Modell integriert und in den Berechnungen berücksichtigt wurden, wirken sich diese nicht auf die optimale Auswahlentscheidung aus.

Weiterhin wurden die Betrachtungen auf einen Baum von miteinander vernetzten Tradern erweitert. Ebenso wie im bilateralen Verbindungsfall werden durch das SOFT-Konzept Entscheidungsregeln festgelegt und analytisch bewiesen. Diese Regeln erlauben es, innerhalb eines gegebenen Verteilungsbaums für die einzelnen Verbindungen zwischen den Knoten des Baums, die optimale Strategie zu bestimmen. Da die in das Modell einfließenden Maßzahlen Änderungen unterworfen sind und dieses eine Anpassung einzelner Strategien zwischen jeweils zwei Tradern zur Folge haben kann, wurde durch ein Simulationsmodell [Arma97] eine Vereinfachung erarbeitet, die auf einem Schwellwert bezüglich der Part-Größe basiert. Bis zu diesem Schwellwert wird nur zwischen der Pull- und der Push-Strategie eine Auswahl getroffen und ab diesem nur noch die Lazy-Push-Strategie verwendet, da die Lazy-Push-Strategie mit zunehmender Part-Größe die anderen Strategien dominiert. Ein weiteres wichtiges Ergebnis dieser Untersuchungen ist, daß sich durch die Verwendung dieses Schwellwertes die notwendigen Änderungsnachrichten zur Anpassung der Verteilungsstrategien signifikant senken bzw. im Fall des Überschreiten des Schwellwerts sogar ganz reduzieren lassen.

3.3 Logische Informationsflüsse

Eine adaptive Steuerung der Konfiguration der Trader-Verbindungen innerhalb des Vermittlungssystems stellt den anwendungsbezogenen Bereich des SOFT-Konzepts dar. Die Festlegung von Verbindungen zwischen den Tradern wird im Trading-Standard sowohl durch lokale als auch durch administrative Regeln bestimmt. Da im Trading-Standard hierfür keine übergeordnete Strategie vereinbart ist, entsteht so mit der Zeit ein redundantes teilvermaschtes Netz miteinander kooperierender Trader, welches einzelne Trader bzw. ihre Administratoren vor eine Auswahlproblematik bei der Suche nach geeigneten Dienstobjekten stellt (Auswahlproblem einer Verbindung bei Anfragen). Zudem muß ein Trader-Administrator wissen, welche Verbindungen er pflegen, aufbauen bzw. abbauen soll, um die effiziente Vermittlungsfunktion für die Anwendungen zu gewährleisten. Das im Trader-Standard [JTC197] vorgesehene Link-Verfahren stellt hier einen viel zu grobgranularen und damit unzureichenden Mechanismus dar.

Zur Lösung werden auf Anwendungsebene im SOFT-Konzept hierzu Stellen, an denen Parts in das Vermittlungssystem eingebracht werden, als Informationsquellen und Stellen, an denen Parts von Anwendungen benötigt werden, als Informationssenken modelliert. Ein Part "fließt" dabei von der Quelle durch das Vermittlungssystem, d.h. über die miteinander verbundenen Trader, zu einer Senke. Bevor ein solcher Informationsfluß tatsächlich stattfindet, läßt sich bereits im Vorfeld dessen prinzipielle Möglichkeit und nach erfolgten Fluß dessen weitere Möglichkeit bestimmen. Hierzu wurde der Begriff des *Logischen Informationsflusses* zwischen einer Informationsquelle und einer Informationssenke definiert [Waib98], der einerseits das Potential zu einem wirklichen Informationsfluß durch das Vermittlungssystem beschreibt und andererseits zur Definition der mit dem Informationsfluß verbundenen Kommunikationskosten dient.

Ein Logischer Informationsfluß beschreibt diejenigen Informationen, die beim Austausch eines Parts zwischen Quelle und Senke fließen bzw. für dessen Fluß notwendig sind. Diese Informationen lassen sich in einen anwendungsbezogenen Teil und einen vermittlungssystembezogenen Teil unterscheiden. Innerhalb des Anwendungsteils findet eine weitere Trennung in angebotsbezogene und nachfragebezogene Informationen statt, worin sich z.B. die genannten Änderungsraten und Zugriffsraten von Parts wiederfinden. Im Vermittlungssystemteil wird zwischen administrativen (z.B. Weitergabebeschränkung durch Hop-Counter), topologischen (z.B. Art der Kommunikationsverbindung) und verhaltensbezogenen (z.B. Auslastung oder Angebotsmenge) Informationen differenziert. Der eine Teil dieser Informationen ist minimal notwendig, damit ein Logischer Informationsfluß zwischen Quelle und Senke existieren kann. Er muß bei einer Rekonfiguration der Verbindungen weiter vorhanden sein. Der andere Teil der Informationen wird zwischen den einzelnen Tradern über die Verbindungen ausgetauscht und verursacht damit unterschiedliche Kommunikationskosten in Abhängigkeit der in Unterkapitel 3.2 genannten Verteilungsstrategien.

Soll nun eine Rekonfiguration der Trader-Verbindungen stattfinden, so wird im SOFT-Konzept im ersten Schritt sichergestellt, daß der Logische Informationsfluß bei einer neuen Verbindung weiterhin erhalten bleibt. In einem zweiten Schritt lassen sich die resultierenden Kosten für die alternative Verbindung mit der ursprünglichen Verbindung vergleichen. Bei einem nachfolgenden Aufruf wird dann die jeweils kostengünstigste Verbindung ausgewählt. Finden über eine Verbindung keine Aufrufe mehr statt, so kann sie automatisch entfernt werden. Ein versuchsweiser Aufbau von Verbindungen

kann hier verwendet werden, um die Trading-Community eines Traders zu erweitern und die Verbindungseffizienz auf Basis von Kommunikationskosten zu reduzieren. Betrachtungen hierzu finden sich in [LeBe95].

4 Zusammenfassung und Ausblick

Der vorliegende Beitrag hat in kurzer Form das SOFT-Konzept vorgestellt, daß einen Beitrag zur effizienten Kopplung von verteilten Trading-Systemen liefert. Der Schwerpunkt des Konzept liegt dabei auf der bedarfsorientierten Vermittlung von skalierbaren Dienstobjekten in verteilten Systemen. Zur Steuerung dient dabei ein auf Kommunikationskosten basierendes analytisches Kostenmodell. Die darin entwickelten Ergebnisse wurden mit Hilfe eines Simulationsmodells eingehend überprüft. Zur Bewertung und Steuerung der Trader-Verbindungen wurde das Konzept der Logischen Informationsflüsse eingeführt. Eine Validierung des SOFT-Konzepts findet augenblicklich durch die Implementierung entsprechender SOFT-Trader im Rahmen zweier Diplomarbeiten am Institut für Telematik statt und soll als Basis für reale Systemmessungen dienen. Eine Integration des Vermittlungssystem in die SFB346-Infrastruktur ist geplant.

Literatur

- [Arma97] L. Armand. Simulation von vernetzten Trading-Systemen. Diplomarbeit, Universität Karlsruhe, Fakultät für Informatik, Institut für Telematik, Dezember 1997.
- [BBKL⁺97] F. Bronsard, D. Bryan, W.V. Kozacynski, E.S. Liongosari, J.Q. Ning, A. Olafsson und J.W. Wetterstrand. Toward Software Plug-and-Play. In *Proceedings of the ACM SIGSOFT – 1997 Symposium on Software Reusability (SSR'97)*, Boston, Massachusetts (USA), 17–19 May 1997.
- [BeRa91] M. Bearman und K. Raymond. Federating Traders: an ODP Adventure. In *International IFIP Workshop on Open Distributed Processing*, Berlin, Germany, October 8–11 1991.
- [Burg95] C. Burger. Cooperation policies for traders. In *3rd IFIP TC6/WG6.1 International Conference on Open Distributed Processing (ICODP'95)*, Brisbane, Australia, 1995.
- [GHHK⁺97] A.G. Grosse, J. Hartroth, G. Hillebrand, D.A. Kottmann, G. Krüger und P.C. Lockemann. SFB 346 — Integrationstechnologie als Innovationsmotor für Maschinenbauanwendungen. In *Informatik '97 – Informatik als Innovationsmotor – 27. Jahrestagung der Gesellschaft für Informatik*, Aachen, 24.–26. September 1997. S. 185–194.
- [GrKH97] A.G. Grosse, D.A. Kottmann und J. Hartroth. Disseminating Object-Oriented Applications in Large Scale Environments. In *International Conference on Object Oriented Information Systems (OOIS'97)*, Brisbane, Australia, 10 – 12 November 1997.

- [GrKo97] A. Grosse und D. Kottmann. Optimale Interworking-Strategien für verteilte Trader-Netze. Technischer Bericht 8/97, Universität Karlsruhe, Fakultät für Informatik, 1997.
- [GrWo95] M.L. Griss und M. Wosser. Making Reuse Work at Hewlett-Packard. *IEEE Software*, Januar 1995, S. 105–107.
- [JTC197] ISO/IEC JTC1/SC21. *Draft Rec. X.950-1/ISO/IEC DIS 13235-1 – ODP Trading Function: Specification — Information technology – Open Distributed Processing – Trading Function-Part 1: Specification, Revised 2nd DIS text*. International Organization for Standardization, Januar 1997.
- [Kott96] D. Kottmann. *Replikation in vernetzten Systemen mit mobilen Teilnehmern*. Dissertation, Universität Karlsruhe, 1996.
- [LaVe97] R. Laddaga und J. Veitch. Dynamic Object Technology. *Communications of the ACM* 40(5), Mai 1997, S. 37–38.
- [LeBe95] O.-K. Lee und S. Benford. An Explorative Model for Federated Trading in Distributed Computing Environments. In *3rd IFIP TC6/WG6.1 International Conference on Open Distributed Processing (ICODP'95)*, Brisbane, Australia, 1995. S. 197–207.
- [MeZP96] B. Meyer, S. Zlatinitsis und C. Popien. Enabling interworking between heterogenous distributed platforms. In *IFIP/IEEE International Conference on Distributed Platforms: Client/Server and Beyond: DCE, CORBA, ODP and Advanced Distributed Applications (ICDP'96)*, London, UK, 1996. Chapman & Hall, S. 329–341.
- [OrHa97] R. Orfali und D. Harkley. *Client/Server Programming with JAVA and CORBA*. John Wiley & Sons, Inc. 1997.
- [Rost97] J. Rost. Wiederverwendbare Software. *Wirtschaftsinformatik* 39(4), April 1997, S. 357–365.
- [Waib98] J. Waibel. Logische Informationsflüsse zwischen Objekten in Tradervermittelten Client-Server-Systemen. Diplomarbeit, Universität Karlsruhe, Fakultät für Informatik, Institut für Telematik, Januar 1998.
- [ZhFJ97] L. Zhang, S. Floyd und V. Jacobsen. Adaptive Web Caching. In *NLANR Web Caching Workshop*, Boulder, Colorado, USA, 9–10 Juni 1997.

Von Hamstern zu Ameisen – Ein Klon-basierter Mobilitätsansatz für Agenten

Jörn Hartroth

Kurzfassung

Mobile Agenten erlauben die Plazierung spezifischer Anwendungsfunktionalität an den Ort entfernter Daten, um diese unter Vermeidung von Netzwerkkommunikation in einem lokalen Kontext auswerten zu können. Ein wesentliches Ziel mobiler Agenten besteht dabei in der Reduktion des innerhalb einer verteilten Anwendung auftretenden Kommunikationsvolumens. Der verbreitete Migrationsansatz für mobile Agenten erweist sich in der analytischen Kostenbewertung als schlecht skalierbar bezüglich großer Anzahlen von Informationsanbietern, wie sie in realen Anwendungsszenarien auftreten. In diesem Beitrag wird ein alternatives Mobilitätskonzept auf der Basis geklonter Agenten vorgestellt, das der Skalierungsbeschränkung nicht unterliegt, und in seiner praktischen Realisierung in einer Laufzeitumgebung für mobile Agenten betrachtet.

1 Einleitung

Der mobile Agentenansatz realisiert eine Form der Codemobilität, bei der eine Anwendungsfunktion mitsamt ihrem Code, ihren Daten und ihrem Ausführungszustand in eine Einheit, den mobilen Agenten, eingekapselt wird. Ein mobiler Agent kann auf eigene Veranlassung seine Ausführung unterbrechen, durch *Migration* an eine andere Lokation innerhalb eines Kommunikationsnetzwerks wechseln und dort seine Ausführung fortsetzen [BiDF98]. Eine der wichtigsten Zielsetzungen der Codemobilität innerhalb verteilter Anwendungen besteht darin, anwendungsspezifische Funktionen zu einem entfernten Datenbestand zu übertragen, wo sie innerhalb eines lokalen Kontextes auf die Daten zugreifen können. Diese Vorgehensweise ermöglicht eine Minimierung der Netzwerkkommunikation während der eigentlichen Auswertung und kann damit eine Reduzierung bestimmter kommunikationsbezogener Kosten – vordringlich Übertragungsvolumen und Latenzzeit – gegenüber dem herkömmlichen Client/Server-Ansatz erzielen [HaCK95]. Die primären Anwendungsbereiche für mobile Agenten finden sich in der Vision eines globalen Informationsverbunds mit einem elektronischen Markt, digitalen Bibliotheken und drahtlos angebundenen Anwendern [RuGK96], da hier die anwendungsspezifische Datenauswertung und die Optimierung der Kommunikation über schmalbandige Weitverkehrs- und Funknetze besondere Bedeutung einnimmt. Das Potential der Minimierung von Kommunikationskosten durch mobile Agenten ist bereits mehrfach nachgewiesen worden [StSc97], jedoch überwiegend in Szenarien sehr

begrenzter Größe mit i.allg. unter 20 beteiligten Instanzen. Das beabsichtigte Anwendungsumfeld für mobile Agenten zeichnet sich jedoch durch um mehrere Größenordnungen höhere Instanzzahlen aus. Wie in [CaPV97] bereits angedeutet wurde, bedarf das Verhalten des mobilen Agentenansatzes in großen Szenarien der genauen Prüfung, um Skalierungsprobleme zu vermeiden.

In diesem Beitrag soll auf Basis eines gegenüber bisherigen Untersuchungen [CaPV97, StSc97] verfeinerten analytischen Kostenmodells die Eignung des mobilen Agentenansatzes für die Kommunikationsoptimierung datenintensiver verteilter Anwendungen in großen Szenarien betrachtet werden. Für das verbreitete Migrationsverfahren der Agentenmobilität wird dabei eine unzureichende Skalierbarkeit erkannt. Als Alternative wird ein auf dem Klonen von Agenten basierendes Mobilitätsschema vorgestellt, das die festgestellten Einschränkungen vermeidet. Nach dieser analytischen Motivation wird der Klon-Ansatz in seinen praktischen Realisierungsaspekten näher beleuchtet, indem eine organisatorische Struktur und eine Architekturkomponente zur Unterstützung von Klongruppen innerhalb einer Laufzeitumgebung entworfen wird. Daran anschließend werden weitere Optimierungsmöglichkeiten der Agentenkommunikation durch die Verwendung von Multicast-Protokollen betrachtet, die sich mit dem Klon-Ansatz eröffnen. Nach der Behandlung der Kostenaspekte werden weitere Motivationen aus dem Bereich der Lokalisierung mobiler Agenten und der Agentensicherheit angeführt, in denen der Klon-Ansatz vorteilhaft einsetzbar ist.

2 Kostenanalyse für Mobilitätsparadigmen

Für die Bewertung des Kostenverhaltens verteilter agentenbasierter Anwendungen wird ein analytisches Modell zugrunde gelegt, das auf einer vereinfachten Sichtweise der realen Netzwerktopologie basiert. Als Hintergrund wird eine Klasse von *datenintensiven Anwendungen* betrachtet, deren Aufgabe darin besteht, die bei einer Menge von entfernten Informationsanbietern (*IAs*) vorliegenden Daten auf spezifische Weise auszuwerten. Konkrete Beispiele hierfür bieten die Informationssuche in digitalen Bibliotheken, die Suche nach Handelsgütern in einem elektronischen Markt oder die Sammlung von Statusinformationen in einem weiträumigen Netzwerkmanagement. Da sich im globalen Netzbereich noch kein explizites Abrechnungssystem durchgesetzt hat, muß zunächst ein sinnvoller Kostenparameter festgelegt werden. Eine naheliegende Wahl, die auch an anderen Orten verwendet wird [CaPV97, StSc97], besteht im aufsummierten Datenvolumen über alle im Rahmen der Anwendung auftretenden Kommunikationsvorgänge hinweg, da dieser Parameter

- sehr einfach auf Basis jeweils lokal verfügbarer Daten zu ermitteln ist,
- in guter Übereinstimmung mit existierenden Ansätzen zur Kostenabrechnung im Internet liegt [MMVa95] und
- in enger Beziehung zu weiteren Kostenparametern steht, so etwa bei Annahme einer uniformen Bandbreite mit der Latenzzeit der Übertragung.

Zur Veranschaulichung der nachfolgenden Formelansätze sind in Abbildung 1 die angenommene Topologie und die Kommunikationsmuster des Client/Server-Ansatzes und des Migrationsansatzes für mobile Agenten dargestellt.

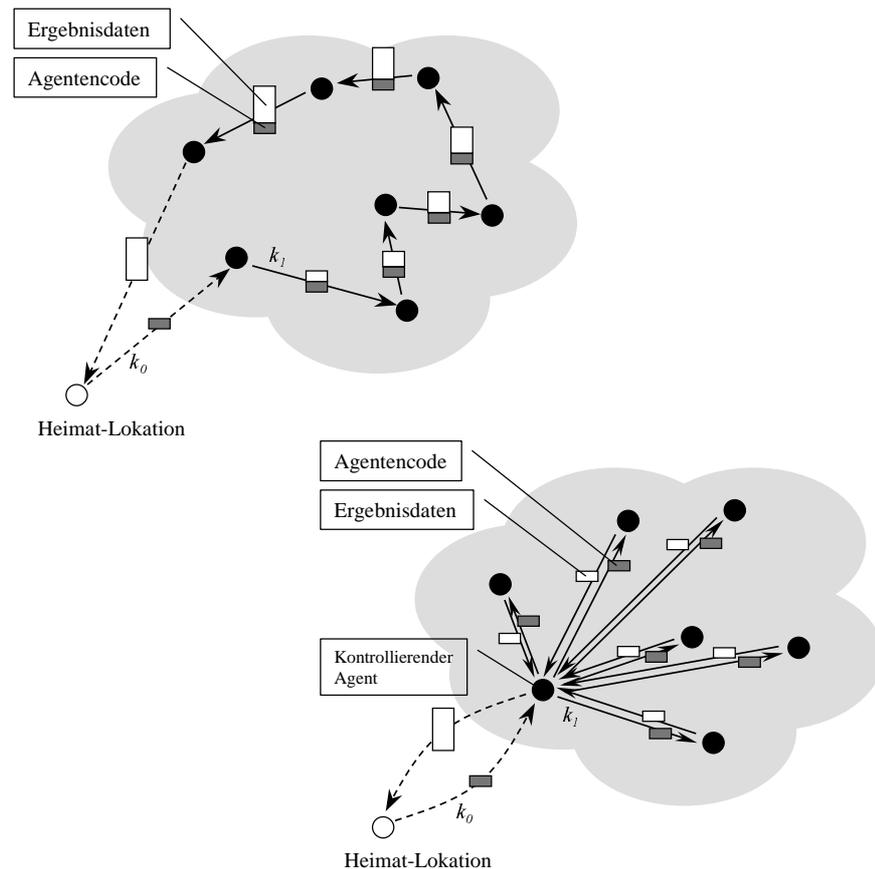


Abbildung 1: Migrations- (*Multi-Hop*) und Klonschema (*One-Hop*) mobiler Agenten

Als beteiligte Instanzen treten die kontrollierende verteilte Anwendung an ihrem Heimatort und eine Menge von n entfernten Informationsanbietern $IA_{1...n}$ auf. Die Menge der Informationsanbieter wird als vorab – z.B. nach einer Trader-Anfrage – bekannt vorausgesetzt, womit auf die Fähigkeit der mobilen Agenten zur dynamischen Entdeckung von Anbietern verzichtet wird. Jeder Informationsanbieter IA_i kontrolliert einen anwendungsrelevanten Datenbestand der Größe D_i . Die Kommunikationskosten werden durch Multiplikation des anfallenden Volumens mit einem konstanten Kostenfaktor bestimmt, der zwischen den Informationsanbietern ungeachtet ihrer relativen Entfernung einheitlich k_1 beträgt. Für die Strecke zwischen dem Heimatort der Anwendung und den Informationsanbietern wird der gesonderte Faktor $k_0 > k_1$ definiert, um den wichtigen Fall einer schmalbandig über Telefon oder drahtlose Netze angebotenen Anwendung zu berücksichtigen¹.

Im Client/Server-Ansatz wird auf Anfrage bei einem Informationsanbieter hin dessen relevanter Datenbestand D_i jeweils vollständig zum Heimatort der Anwendung übertragen, wobei Kosten von $k_0 \cdot D_i$ entstehen. (Das bei der Anfrage auftretende Kommunikationsvolumen wird hier als vernachlässigbar klein angenommen.) Im mobilen Agentenansatz hingegen ist durch spezifische Bearbeitung der Daten beim Informationsanbieter eine Reduktion des Ergebnisses um den Selektionsfaktor $s < 1$ auf $s \cdot D_i$ möglich, worin sich das Optimierungspotential des Ansatzes manifestiert. Zusätzlich müssen jedoch im Rahmen der Agentenmigration Code (und Ausführungszustand) des Volumens c_{sel} sowie die bislang bei anderen Anbietern gewonnenen Ergebnisdaten zum Informationsanbieter übertragen werden. Im letzten Migrationsschritt braucht der Co-

¹Für Anwendungen im Festnetz wähle man $k_0 = k_1$.

de des Agenten nicht wieder zur Anwendung zurück übertragen zu werden.

Als weitere Vereinfachung wird hier zunächst angenommen, daß die Daten der einzelnen Informationsanbieter untereinander disjunkt sind. Eine Betrachtung teilweise redundanter und replizierter Daten ist in [Hart97] angestellt worden, auf die Angabe der Kostenfunktionen wird hier jedoch aus Platzgründen verzichtet. In diesem Rahmen ergeben sich die Kostenfunktionen C_{CS} für den Client/Server-Ansatz und C_{mig} für den auf Migration beruhenden mobilen Agentenansatz aus den folgenden Gleichungen. Die exakten Formeln sind jeweils in einen geschlossenen Ausdruck entwickelt worden (hinter dem \simeq , mit der weiteren Vereinfachung $D_i = \text{const}$), um den Kostenverlauf in Abhängigkeit von der Anzahl n der Informationsanbieter graphisch darstellen zu können (Abbildung 2).

$$\begin{aligned}
C_{CS} &= \sum_{i=1}^n k_0 D_i \\
&\simeq k_0 n D \\
C_{mig} &= k_0 c_{sel} + \sum_{i=1}^{n-1} k_1 (c_{sel} + \sum_{h=1}^i s D_h) + k_0 \sum_{h=1}^n s D_h \\
&\simeq k_0 (c_{sel} + n s D) + \boxed{k_1 (n-1) (c_{sel} + \frac{1}{2} n s D)} \\
C_{klo} &= k_0 c_{sel} + \sum_{i=2}^n k_1 (c_{sel} + s D_i) + k_0 \sum_{i=1}^n s D_i \\
&\simeq k_0 (c_{sel} + n s D) + k_1 (n-1) (c_{sel} + s D)
\end{aligned}$$

Das zentrale Ergebnis dieses Vergleichs besteht im Auftreten einer quadratischen Abhängigkeit im hervorgehobenen k_1 -Term der Kostenfunktion C_{mig} . Auch wenn für datenintensive Anwendungen $c_{sel} \ll D$ angenommen werden kann und damit die Kommunikationskosten des mobilen Agentenansatzes auf der teuren k_0 -Strecke günstiger ausfallen als beim Client/Server-Fall, wird der quadratische Beitrag die Gesamtkosten mit zunehmender Szenariogröße dominieren (siehe Abbildung 2).

Die begrenzte Skalierbarkeit des migrierenden Agenten rührt offensichtlich nicht von einem prinzipiellen Mangel des Agentenansatzes her, sondern vom der Ansammlung und Mitnahme von Teilergebnissen über den Migrationspfad, der sogenannten *Multi-Hop* Mobilität. Abhilfe verspricht daher ein alternatives Mobilitätsmuster, in dem nicht ein einzelner Agent sequentiell alle Informationsanbieter besucht, sondern in dem mehrere Agenten parallel jeweils einen *IA* bearbeiten (*One-Hop* Mobilität). Als naheliegende Optimierung werden diese parallelen Agenten im hier vorgeschlagenen Ansatz nicht am Heimatort der Anwendung eingesetzt, sondern erst nach Überwindung der teuren k_0 -Strecke von einem Agenten im Festnetz geklont. Die Kostenfunktion dieses Klon-orientierten Ansatzes ist als C_{klo} aufgeführt. Diese Form mobiler Agenten vermeidet das für migrierende Agenten beobachtete Skalierungsproblem und erzielt daher auch für große Szenarien einen Kostenvorteil durch anwendungsspezifische Datenselektion. Die im bisherigen Modell extrem vereinfachte Netztopologie wurde um eine räumliche Struktur erweitert. Die Kosten im Festnetz werden jetzt zusätzlich in "lokale" Verbindungen hoher Bandbreite mit Kostenfaktor k_2 , etwa in Form von LANs oder MANs, und "globale" Weitverkehrsverbindungen mit Kosten k_1 unterschieden ($k_1 > k_2$). Es bilden sich so "lokale Inseln" von günstig verbundenen Informationsanbietern aus, zwischen denen teurere "globale" Verbindungen bestehen. Das Kommunikationsmuster für den Client/Server Ansatz ändert sich hierbei nicht. Für den migrierenden Agenten wird ein optimaler Migrationspfad angenommen, und der Klon-Ansatz in eine zweistufige Hierarchie erweitert (Abbildung 3).

In diesem verfeinerten Modell gewinnen mobile Agenten gegenüber der einfachen Topologie einen Vorteil, da sie im Gegensatz zu den Nachrichten im Client/Server-Ansatz die besonders günstigen "lokalen" Kommunikationswege nutzen können. Der migrierende Ansatz verliert jedoch durch das quadratische Kostenverhalten den möglichen Vorteil gegenüber Client/Server bei großen Anbieterzahlen (Abbildung 4).

Aus dieser Analyse ist das Fazit zu ziehen, daß die Verwendung eines einzelnen migrierenden Agenten für datenintensive verteilte Anwendungen nicht skaliert. Auch bei Annahme von Redundanz innerhalb der entfernten Datenbestände ergibt sich kein prinzipiell anderes Verhalten (siehe [Hart97]). Mit dem Klon-basierten Mobilitätsschema für Agenten wurde eine Alternative vorgestellt, in der das Potential des mobilen Agentenansatzes zur Kostenoptimierung auch auf große Szenarien skaliert.

3 Laufzeitunterstützung für Klon-basierte Agentenmobilität

Die dynamische Erzeugung von Agenten durch Klone ist in aktuellen Implementierungen mobiler Agentensysteme umgesetzt [LaCh96]. Die Unterstützung beschränkt sich hier jedoch bislang auf die grundsätzlichen Mechanismen der Klonerzeugung. Speziell existiert kein Konzept für die organisatorische Kontrolle der Ausführung einer zusammenhängenden Anwendungsaufgabe durch eine Gruppe mobiler Agenten. Die in der vorstehenden Analyse auf zwei Ebenen eingeführte Klon-Hierarchie kann bei einer Erweiterung auf unbegrenzt viele Ebenen dazu dienen, eine funktionelle Gliederung der Anwendung in abhängige Teilaufgaben flexibel auf die Ausführungstopologie abzubilden. Für die dabei entstehenden komplexen Beziehungen ist die bisherige Lösung, die Koordination zwischen mehreren Agenten vollständig der Anwendungsprogrammierung zu überlassen, unpraktikabel.

Aus diesem Grund wird eine neue Architekturkomponente, der *Klondienst* eingeführt, um die Funktionen der Koordination von Klongruppen hinter einer einfachen Schnitt-

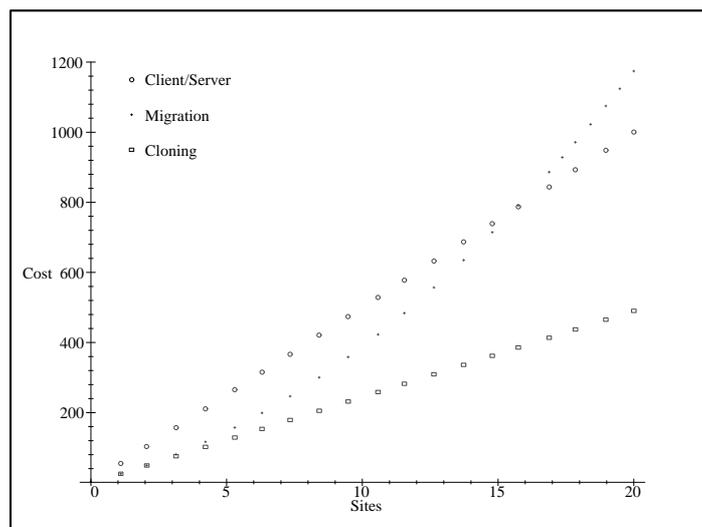


Abbildung 2: Kommunikationskosten für verschiedene Mobilitätsschemata, $k_0 = 5$, $k_1 = 1$, $c_{sel} = 0.6$, $s = 0.4$, $D = 10$

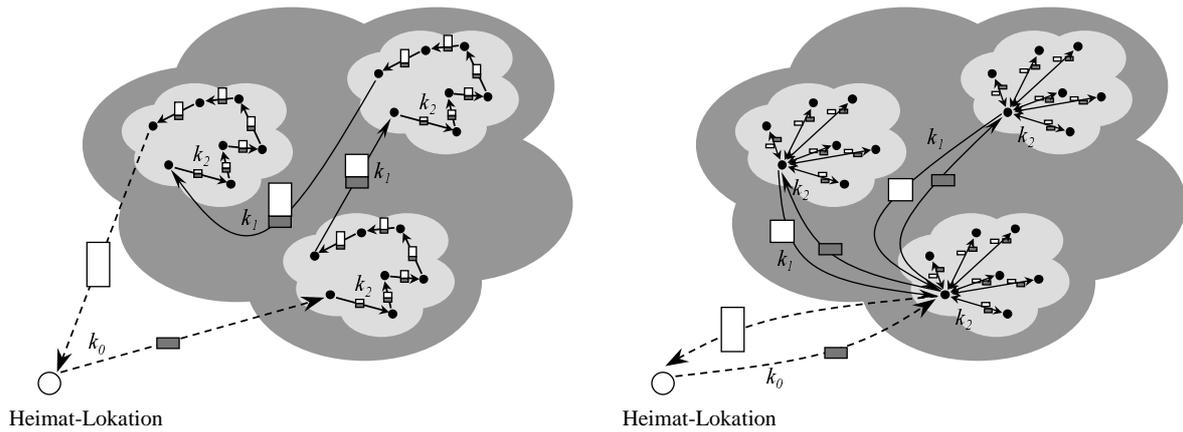


Abbildung 3: Migrierende Agenten und Klon-Ansatz in strukturierter Topologie

stelle vor der Anwendung zu verbergen. Die Aufgaben des Klondienstes sind in drei unterschiedliche Phasen eingeteilt:

Phase 1: Initialisierung Aufbau und Platzierung einer Klonhierarchie nach den Vorgaben der Anwendung

Phase 2: Verteilte Berechnung Überwachung der entfernt ablaufenden Klone, dabei ggf. dynamische Erweiterung der Klongruppe bei neu entdeckten Informationsanbietern

Phase 3: Ergebnisintegration Unterstützung der Anwendung bei der Integration der von den einzelnen Klonen beigetragenen Teilergebnisse

In der abschließenden Phase der klonbasierten Berechnung ist die Anwendung nach wie vor für die spezifische Auswertung der Ergebnisdaten zuständig. Für die Beziehung der Einzelergebnisse treten jedoch in der Anwendungssemantik drei typische Fälle auf, die vom Klondienst unterstützt werden können:

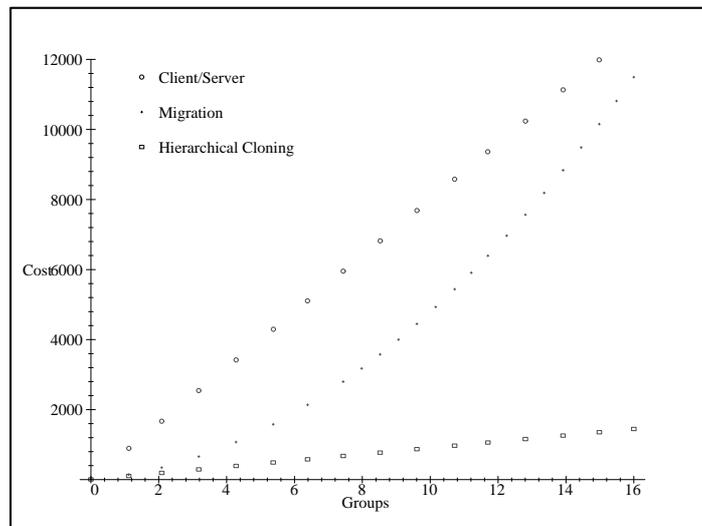


Abbildung 4: Kommunikationskosten in strukturierter Topologie, $k_0 = 10$, $k_1 = 3$, $k_2 = 1$

AND-Semantik Die Anwendung benötigt die Teilergebnisse aller Klone. Eine Integration kann erst nach ihrem vollständigem Vorliegen erfolgen, und das Versagen einzelner Klone muß speziell behandelt werden (etwa durch Wiederholung).

OR-Semantik Die Anwendung bearbeitet die Teilergebnisse möglichst vieler Klone, kann jedoch auf einzelne verzichten. Dies erlaubt eine verkürzte Gesamtausführungszeit und vereinfacht die Fehlerbehandlung, da die Anwendung ggf. nur die nach einem bestimmten Zeitraum eingegangenen Ergebnisse berücksichtigt.

XOR-Semantik Die Anwendung benötigt ein bestimmtes Ergebnis und kann die Berechnung nach dessen Eingang sofort abschließen. Dieses Verhalten dient der Vermeidung unnötigen Ressourcenverbrauchs.

Die in einem Klondienst konzentrierte Gesamtsicht auf Klongruppen ermöglicht eine weitere Optimierung der Agentenkommunikation durch die Verwendung von Multicast-Protokollen. Die Hierarchie der Klongruppen auf Anwendungsebene wird dabei mit einer entsprechenden Hierarchie von Multicastgruppen auf Transportebene unterlegt. Die initiale Migration der Klongruppe, Koordinationsnachrichten zwischen Gruppenmitgliedern und Aktualisierungsnachrichten der Anwendung an die Gruppe können damit deutlich günstiger abgewickelt werden als mit den heute üblichen Unicast-Nachrichten [HaHo98]. Gerade die oben erwähnte XOR-Semantik für Teilergebnisse wird erst durch einen solche effizienten Kommunikationsmechanismus für die Verbreitung einer Terminierungsnachricht an nicht mehr benötigte Klone sinnvoll einsetzbar.

Die Auswahl eines geeigneten Multicast-Protokolls ist dazu von großer Bedeutung, da die existierenden Protokolle meist für Anwendungscharakteristiken entworfen wurden, die von mobilen Agenten deutlich abweichen. Zuverlässige Auslieferung, dynamisch veränderliche Empfängerpopulationen, transaktionaler Datenverkehr und mehrere Sender werden derzeit am besten vom Protokoll SRM [FJML⁺95] unterstützt.

4 Weitere Aspekte

Neben den vorstehend nachgewiesenen Vorteilen des Klon-Ansatzes bei den Kommunikationskosten entstehen auch Synergieeffekte in anderen Aufgabenbereichen der Laufzeitunterstützung für mobile Agenten. Durch geeignete Anpassung der unterliegenden Multicastgruppen kann eine transparente Lokalisierung von Klonen innerhalb einer Familie erzielt werden. Dieser Mechanismus kann auch auf allgemeine Agentengruppen ausgedehnt werden, die temporär kooperieren und sich dazu gegenseitig lokalisieren müssen. Für die notorisch aufwendige Aufgabe der Lokalisierung mobiler Agenten während ihrer entfernten Ausführung kann so in einem Teilbereich eine effiziente neue Lösung erzielt werden.

Im Rahmen der Sicherheit in mobilen Agentensystemen besteht ein bislang ungelöstes Problem im Schutz des mobilen Agenten gegen böswillige Informationsanbieter. Da die ausführende Laufzeitumgebung effektiv totale Kontrolle über einen mobilen Agenten besitzt, setzt ein verbreiteter Lösungsvorschlag die Existenz einer Infrastruktur von vertrauenswürdigen Umgebungen (sogenannten *trusted hosts*) voraus, die aber nur einen geringen Anteil des globalen Netzwerks ausmachen wird. Die Beurteilung der Integrität eines migrierenden Agenten ist in einer solchen Welt unterschiedlicher Vertrauenswürdigkeit bereits nach wenigen Schritten nicht mehr zu handhaben [Ordi96]. Hier

ist ein hierarchischer Klon-Ansatz ideal geeignet, wichtige Funktionen (z.B. elektronische Zahlungsmittel) mit den kontrollierenden Agenten an vertrauenswürdigen Orten zu plazieren und Klone mit geringeren Vertrauensanforderungen für den eigentlichen Datenzugriff zu verwenden.

5 Zusammenfassung

In diesem Beitrag wurde in einem analytischen Modell nachgewiesen, daß die weit verbreitete Verwendung eines einzelnen migrierenden Agenten für datenintensive Anwendungen mit vielen Informationsanbietern nicht skaliert. Mit dem Konzept des hierarchischen Klonens wurde ein neuer Ansatz vorgestellt, der die Skalierungsprobleme vermeidet und die angestrebte Ersparnis von Kommunikationskosten auch für große Szenarien gewährleistet. Nachfolgend wurde der neue Ansatz über einen Klondienst in die Architektur einer Laufzeitumgebung für mobile Agenten eingebettet. In diesem Rahmen wurde auch eine weitere Optimierungsmöglichkeit der Agentenkommunikation durch Multicast-Protokolle diskutiert. Abschließend wurde der Beitrag des Klon-orientierten Ansatzes zur Lösung existierender Probleme bei der Lokalisierung und der Sicherheit angesprochen.

Literatur

- [BiDF98] Lubomir F. Bic, Michael B. Dillencourt und Munehiro Fukuda. Mobile Network Agents. In *Encyclopedia of Electrical and Electronics Engineering*. John Wiley & Sons, Inc., 1998.
- [CaPV97] A. Carzaniga, G. Picco und G. Vigna. Designing Distributed Applications with Mobile Code Paradigms. In *Proceedings of the 19th International Conference on Software Engineering, Boston, MA, 1997*.
- [FJML⁺95] Sally Floyd, Van Jacobson, Steven McCanne, Ching-Gung Liu und Lixia Zhang. A Reliable Multicast Framework for Light-weight Sessions and Application Level Framing. In *Proceedings ACM SIGCOMM '95, August 28 - September 1, 1995, Cambridge, MA, SIGCOMM*. ACM, 1995, S. 342–356.
- [HaCK95] Colin Harrison, David Chess und Aaron Kershenbaum. Mobile Agents: Are they a good idea? Technischer Bericht RC19887, IBM Research Division, Watson Research Center, Yorktown Heights, NY, März 1995.
- [HaHo98] Jörn Hartroth und Markus Hofmann. Using IP multicast to improve communication in large scale mobile agent systems. In *Proceedings of the 31st Hawaiian International Conference on System Sciences (HICSS'31), Kohala Coast, Hawaii, Jan. 6-9, 1998*, Januar 1998.
- [Hart97] Jörn Hartroth. Dynamisches Cloning als optimierter Mobilitätsmechanismus für autonome Agenten – Analytisches Kostenmodell. Technischer Bericht 21, Universität Karlsruhe (TH), September 1997.

-
- [LaCh96] Danny Lange und Daniel Chang. IBM Aglets Workbench: Programming Mobile Agents in Java. <http://www.ibm.co.jp/trl/aglets>, September 1996.
- [MMVa95] J. MacKie-Mason und H. Varian. *Public Access to the Internet*, Kapitel Pricing the Internet. MIT Press, Cambridge, MA. 1995.
- [Ordi96] Joann J. Ordille. When agents roam, who can you trust? In *Proceedings of the First Conference on Emerging Technologies and Applications in Communications, Portland, OR*, Mai 1996.
- [RuGK96] Daniela Rus, Robert Gray und David Kotz. Autonomous and Adaptive Agents that Gather Information. In *AAAI '96 International Workshop on Intelligent Adaptive Agents*, August 1996, S. 107–116.
- [StSc97] Markus Straßer und Markus Schwehm. A Performance Model for Mobile Agent Systems. In *Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications PDPTA'97, Las Vegas*, Band II, 1997, S. 1132–1140.

LGMP/LGCP: Eine Protokoll-Suite für skalierbare Multicast-Kommunikation im Internet

Markus Hofmann

Kurzfassung

Neben der klassischen Zweiparteien-Kommunikation gewinnt die Datenübertragung von einem Sender an mehrere Empfänger, die sogenannte *Multicast-Kommunikation*, zunehmend an Bedeutung. Ein großes Spektrum moderner Anwendungen aus den Bereichen verteilte Systeme, verteilte rechnergestützte Gruppenarbeit, Konferenzsysteme und Verteildienste basiert auf der Kommunikationsform des Multicast. Um den Qualitätsanforderungen dieser Anwendungen zu genügen, müssen Kommunikationssysteme über Mechanismen zur Fehlerkorrektur und zur Verkehrssteuerung verfügen. Multicast-Kommunikation erfordert im Gegensatz zur klassischen Zweiparteien-Kommunikation die Unterstützung unterschiedlich großer Empfängermengen. Die Gruppengröße kann von einigen wenigen Kommunikationsteilnehmern bis hin zu einigen tausend Empfängern variieren. Neben einer Erweiterung um Funktionen des Gruppenmanagements erfordert dies den Entwurf neuer Protokollmechanismen, die hinsichtlich der Empfängerzahl skalierbar ausgelegt sind und die Koordination großer Empfängermengen ermöglichen.

1 Einleitung

Die bisherigen Ansätze zur Realisierung von Multicast-Diensten basieren auf erweiterten Mechanismen der klassischen Zweiparteien-Kommunikation [Stra95], [KoZa96], [MRTW97]. Diese sind durch eine senderorientierte Ausrichtung gekennzeichnet, wobei die Verantwortung sowohl für die Steuerung des Datenflusses als auch für die Korrektheit der Datenübertragung der sendenden Instanz obliegt. Zur Durchführung einer Verkehrssteuerung wertet der Sender die Statusmeldungen aller Empfänger aus und regelt den Datenfluß entsprechend dem Zustand der Empfängermenge. Ist diese Vorgehensweise bei der klassischen Zweiparteien-Kommunikation noch unproblematisch, so führt die Bearbeitung einer Vielzahl von Statusmeldungen im Falle großer Kommunikationsgruppen zu einer Überlastung des Senders. Zudem erfordert die Übermittlung mehrerer Statusmeldungen zusätzliche Bandbreite, was insgesamt in einer deutlichen Verminderung der effektiven Übertragungsleistung und in einer ineffizienten Ressourcenauslastung resultiert. Auch die heute bekannten Verfahren zur Fehlerkorrektur sind nicht geeignet für die Multicast-Kommunikation in Weitverkehrsnetzen. Bei der klassischen Fehlerkorrektur basierend auf Übertragungswiederholungen fordern Empfänger fehlende Datenpakete stets direkt beim Sender an. Dieser überträgt daraufhin die angeforderten Dateneinheiten erneut über das Kommunikationsnetz. Eine solche Vorgehensweise eignet sich im Falle der Multicast-Kommunikation lediglich für den Einsatz

in lokalen Netzwerken. Die Kosten zur Durchführung von Übertragungswiederholungen fallen hier vergleichsweise gering aus. Dagegen ist in Weitverkehrsnetzen die Reduzierung der Netzbelastung ein vorrangiges Ziel. Mit zunehmender Empfängerzahl wächst die Wahrscheinlichkeit eines Paketverlustes und damit die Anzahl notwendiger Übertragungswiederholungen. Messungen im Internet zeigen, daß oftmals bis zu 70% der Datenpakete aufgrund eines Paketverlustes bei mindestens einem Empfänger wiederholt zu übertragen sind [YaKT96]. Eine senderorientierte Durchführung von Übertragungswiederholungen führt in diesem Fall zu einem erheblichen Mehrbedarf an Bandbreite und damit zu höheren Kosten. Dies verdeutlicht die Notwendigkeit neuer Fehlerkorrekturmechanismen zur Realisierung skalierbarer Multicast-Dienste.

Aus der geschilderten Problematik ergab sich als Zielsetzung des Projekts die Entwicklung und die Realisierung eines fortschrittlichen Kommunikationssystems, welches unter Berücksichtigung anwendungsspezifischer Anforderungen an die Dienstqualität einen adaptiven und skalierbaren Multicast-Dienst bereitstellt. Das entwickelte Kommunikationssystem übernimmt dabei Funktionen zum Gruppenmanagement und zur dynamischen Strukturierung der Gruppenmitglieder. Dies ermöglicht erstmals die Berücksichtigung der Gruppentopologie zur Optimierung sowohl der Fehlerkorrektur als auch der Quittungsbearbeitung. Im Gegensatz zu klassischen Ansätzen löst sich das im Rahmen des Projekts entwickelte Konzept vom senderorientierten Paradigma und bezieht statt dessen alle beteiligten Komponenten aktiv in die Fehlerkorrektur und in die Steuerung der Datenübertragung ein. Dies ermöglicht eine faire Verteilung der Bearbeitungslast auf unterschiedliche Kommunikationsteilnehmer, wodurch der Sender im Vergleich zu klassischen Verfahren deutlich entlastet wird. Ein wichtiger Entwurfsaspekt besteht dabei in der Strukturierung der Gesamtgruppe. Hierfür wurde ein Verfahren entwickelt, das unter Berücksichtigung der Anwendungsanforderungen und der aktuellen Netzauslastung die Kommunikationsgruppe in mehrere funktionale Einheiten aufteilt und deren Koordination übernimmt. Die entworfenen Mechanismen wurden maßgeblich durch die Eigenschaften paketvermittelter Weitverkehrsnetze beeinflusst. Diese sind durch einen verbindungslosen Übertragungsdienst charakterisiert, der einen nicht fehlergesicherten Datenaustausch zwischen beliebigen Rechnern ohne vorherigen Verbindungsaufbau ermöglicht.

2 Konzeptentwurf

Ausgehend von einer Analyse existierender Multicast-Protokolle wurden mehrere Algorithmen zur Unterstützung einer skalierbaren Gruppenkommunikation erarbeitet [Hofm98]. Die entwickelten Konzepte resultieren in einem generischen Rahmenwerk für skalierbare Multicast-Kommunikation in Weitverkehrsnetzen, dem sogenannten *Local Group Concept (LGC)* [Hofm96b]. Die grundlegende Idee besteht dabei in einer logischen Strukturierung der globalen Kommunikationsgruppe. Diese wird in mehrere lokale Untergruppen aufgeteilt, welche ihrerseits in einer baumartigen Hierarchie angeordnet werden. Dabei faßt jede lokale Gruppe nahe beieinander gelegene Kommunikationsteilnehmer zusammen. Als Entfernungsmaß werden mehrere Metriken herangezogen, die in Abhängigkeit vom jeweiligen Anwendungskontext unterschiedlich gewichtet werden. Im Gegensatz zu heute eingesetzten Kommunikationsprotokollen können beim Local Group Concept auch Empfänger Übertragungswiederholungen ausführen. Durch

Kooperation der Mitglieder einer lokalen Untergruppe werden Übertragungsfehler zunächst ohne Beteiligung des Senders korrigiert. Haben alle Mitglieder einer lokalen Gruppe bestimmte Datenpakete nicht korrekt erhalten, so ist eine lokale Fehlerkorrektur nicht möglich. In diesem Fall werden die fehlenden Daten bei der übergeordneten Teilgruppe bzw. direkt beim Sender angefordert. Das entwickelte Verfahren beschränkt den überwiegenden Teil der notwendigen Übertragungswiederholungen auf den lokalen Bereich einer Untergruppe, was sich in einer deutlichen Entlastung der Weitverkehrsverbindungen niederschlägt. Lokale Übertragungswiederholungen werden durch einen geeigneten Empfänger, den sogenannten *Repräsentanten* der lokalen Gruppe, koordiniert. Dieser wertet zudem die Statusmeldungen der angeschlossenen Empfänger aus, faßt diese in einer einzelnen Nachricht zusammen und übermittelt diese dem Repräsentanten der übergeordneten lokalen Gruppe bzw. dem Multicast-Sender. Diese Vorgehensweise ermöglicht zusammen mit der Dekomposition der Gesamtgruppe in voneinander unabhängige Teilbereiche eine verteilte und damit parallele Steuerung des Datenflusses, was die Skalierbarkeit hinsichtlich großer Empfängerzahlen verbessert.

3 Protokollspezifikation

Ausgehend von den entwickelten Konzepten wurden zwei voneinander unabhängige Protokolle entworfen, deren Gesamtfunktionalität dem Benutzer über eine erweiterte Anwendungsschnittstelle bereitgestellt wird [Hofm98]. Das *Local Group based Multicast Protocol (LGMP)* ist für den eigentlichen Datentransfer verantwortlich. Es operiert auf einer hierarchischen Strukturierung der globalen Kommunikationsgruppe, welche vom *Local Group Configuration Protocol (LGCP)* erstellt wird. Die Anordnung der Untergruppen und damit der Aufbau der Hierarchie erfolgt in Abhängigkeit von den anwendungsspezifischen Anforderungen. Somit wird eine Optimierung der für die Anwendung wichtigen Kenngrößen erreicht. Dies kann beispielsweise die Minimierung der durchschnittlichen Übertragungsverzögerung oder die Reduzierung der globalen Netzlast sein. Aufgrund dynamisch veränderbarer Gruppenzugehörigkeit und schwankender Netzauslastung muß die definierte Gruppenstruktur während der Kommunikation an die aktuellen Gegebenheiten angepaßt werden. Das vom Datentransfer entkoppelte Management der Gruppenhierarchie erlaubt eine weitgehende Unabhängigkeit von Verwaltungs- und Datentransferfunktionen und damit deren parallele Bearbeitung. Der Austausch von Verwaltungs- und Statusinformation erfolgt periodisch, wodurch implizit eine hohe Robustheit und Stabilität des Gesamtsystems erreicht wird.

4 Implementierung und Bewertung

Zur Umsetzung des Entwurfs wurden die Protokolle LGMP und LGCP als erweiterte endliche Automaten modelliert, die über einen asynchronen Nachrichtenaustausch miteinander kommunizieren [Hofm98]. Die Implementierung bildet die Automatenmodelle auf eigenständige Prozesse ab, welche über einen gemeinsamen Speicherbereich miteinander kommunizieren. Beide Prozesse zerfallen intern in mehrere Leichtgewichtsprozesse (Threads), was einen hohen Grad an Parallelität ermöglicht. Die derzeitige Realisierung ist ausgehend von einer anwendungsorientierten Parametrisierung in der Lage,

selbständig und ohne manuelles Eingreifen eine geeignete Gruppenhierarchie zu etablieren und diese gemäß den Anforderungen der Anwendung dynamisch an Änderungen im Systemumfeld anzupassen. Neben dem eigentlichen Kommunikationssystem wurde zu Demonstrationszwecken ein Monitorsystem entworfen und implementiert, welches über eine grafische Schnittstelle die jeweils aktuelle Gruppenstruktur visualisiert und ausführliche Informationen zu den einzelnen Kommunikationsteilnehmern bereitstellt. Ebenso wurde ein Java-basiertes Meßsystem entwickelt und implementiert, welches eine automatisierte und zentral gesteuerte Durchführung von Messungen zur Bewertung von Multicast-Protokollen erlaubt [HoHa97]. Durch seine einfache Handhabung und die hohe Portabilität eignet es sich für den Einsatz in großen globalen Kommunikationsgruppen.

Messungen im Mbone haben die Leistungsfähigkeit der erstellten Protokollimplementierungen unter Beweis gestellt [Hofm98]. Bei der Übermittlung von Daten an mehrere Empfänger werden mit LGMP wesentlich bessere Werte erreicht als mit den heute im Einsatz befindlichen Internet-Protokollen. So konnten im Vergleich zu senderorientierten Multicast-Protokollen die Übertragungszeit einer Datei auf weniger als die Hälfte und das Datenaufkommen auf etwa ein Viertel reduziert werden. Der Anwender profitiert bei Verwendung von LGMP von einer merklich schnelleren Kommunikation und einem wesentlich kostengünstigeren Datenaustausch. Im Vergleich zum klassischen TCP sind die Vorteile des LGMP besonders deutlich. Obwohl LGMP als Multicast-Protokoll keinen Nutzen aus der Einfachheit der Punkt-zu-Punkt-Kommunikation zieht, werden mit der vorliegenden Implementierung im Unicast-Fall annähernd gleiche Leistungswerte erreicht wie mit den kommerziellen und jahrelang optimierten TCP-Implementierungen. Bereits bei zwei Empfängern erweist sich LGMP als leistungsfähiger und kostengünstiger. Während mit zunehmender Gruppengröße die Kosten der TCP-basierten Kommunikation deutlich ansteigen, ist mit LGMP auch bei einer großen Anzahl von Empfängern ein kostengünstiger Datentransfer möglich. Viele Multicast-Anwendungen können dadurch erst realisiert werden.

Simulative Untersuchungen der entworfenen Protokollmechanismen zeigen wesentlich bessere Leistungsdaten im Vergleich zu Verfahren mit ähnlicher Funktionalität. Um die Leistungsfähigkeit und die Skalierbarkeit des entwickelten Konzeptes zu belegen, wurden ausgehend von Meßergebnissen im Internet mehrere Simulationsmodelle erstellt. Diese ermöglichen eine realitätsnahe Untersuchung von Szenarien mit mehreren tausend Empfängern, was durch Messungen mit einer realen Implementierung kaum zu erreichen ist. Die Ergebnisse zeigen, daß durch die Methode der lokalen Übertragungswiederholungen die globale Netzbelastung deutlich unter derjenigen klassischer Ansätze bleibt, wobei zugleich die durchschnittliche Übertragungsverzögerung merklich reduziert wird.

Zusammenfassend kann festgehalten werden, daß die hierarchische Strukturierung globaler Kommunikationsgruppen deutliche Vorteile hinsichtlich des Datenaufkommens sowie des erreichbaren Durchsatzes bringt. Die Messungen im Mbone bestätigen damit die getroffenen Entwurfsentscheidungen. Derzeit werden die Messungen im Mbone fortgeführt, wobei auch größere Kommunikationsszenarien unter Beteiligung aller Testpartner angestrebt werden.

Weitere Informationen und die Multicast-Software sind im Internet kostenlos abrufbar unter <http://www.telematik.informatik.uni-karlsruhe.de/~hofmann/LocalGroups.html>

Literatur

- [DHSW97] S. Dresler, M. Hofmann, C. Schmidt und H. Wiltfang. A Native ATM API Suited for Multimedia Communication. In R. Steinmetz und L.C. Wolf (Hrsg.), *Interactive Distributed Multimedia Systems and Telecommunication Services*, Nr. 1309 der Lecture Notes in Computer Science. Springer Verlag, 1997, S. 352–363.
- [DrHo97] S. Dresler und M. Hofmann. Adaptive Error Correction to Support Heterogeneous Multicast Groups. In *Proceedings of 6th Open Workshop on High Speed Networks, Stuttgart, Deutschland, 8.-9. Oktober 1997*, 1997, S. 169–174.
- [HaHo98] J. Hartroth und M. Hofmann. Using IP Multicast to Improve Communication in Large-Scale Mobile Agent Systems. In *Proceedings of 31st Annual Hawaii International Conference on System Sciences (HICSS)*, Band VII, 1998, S. 64–73. Hawaii, USA, 6.-9. Januar 1998.
- [Hofm96a] M. Hofmann. Adding Scalability to Transport Level Multicast. In A. Danthine G. Ventre, J. Domingo-Pascual (Hrsg.), *Multimedia Telecommunications and Applications*, Nr. 1185 der Lecture Notes in Computer Science. Springer Verlag, 1996, S. 41–55. Proceedings of Third International COST 237 Workshop, Barcelona, Spanien, 25.-27. November 1996.
- [Hofm96b] M. Hofmann. A Generic Concept for Large-Scale Multicast. In B. Plattner (Hrsg.), *Broadband Communications*, Nr. 1044 der Lecture Notes in Computer Science. Springer Verlag, 1996, S. 95–106. Proceedings of International Zurich Seminar on Digital Communication, Zürich, Schweiz, 21.-23. Februar 1996.
- [Hofm96c] M. Hofmann. Hierarchische Strukturierung globaler Kommunikationsgruppen. In M. Zitterbart (Hrsg.), *Kommunikation in verteilten Systemen*, Informatik aktuell. Springer Verlag, 1996, S. 328–342.
- [Hofm97] M. Hofmann. Enabling Group Communication in Global Networks. In *Proceedings of Global Networking'97, Calgary, Alberta, Kanada, 15.-18. Juni*, Band II, 1997, S. 321–330.
- [Hofm98] M. Hofmann. *Skalierbare Multicast-Kommunikation in Weitverkehrsnetzen*. Dissertation, Fakultät für Informatik, Universität Karlsruhe, Februar 1998.
- [HoHa97] M. Hofmann und J. Hartroth. Ein Meßsystem zur Bewertung von Multicast-Protokollen im Internet. In *GI/ITG Fachtagung - Messung, Modellierung und Bewertung von Rechen- und Kommunikationssystemen (MMB'97), Freiberg, Deutschland, 17.-19. September, 1997*, S. 99–109.
- [HoRo97] M. Hofmann und M. Rohrmüller. Impact of Virtual Group Structure on Multicast Performance. In A. Danthine und C. Diot (Hrsg.), *From Multimedia Services to Network Services*, Nr. 1356 der Lecture Notes in Computer Science. Springer Verlag, 1997, S. 165–180. Proceedings of Fourth

International COST 237 Workshop, Lissabon, Portugal, 15.-19. Dezember 1997.

- [HoSc97] M. Hofmann und C. Schmidt. Integration von ATM in bestehende Netze. In *Theorie und Praxis der Wirtschaftsinformatik - HMD*, Nr. 196 der Lecture Notes in Computer Science. Hüthig Verlag, Juli 1997, S. 96–108.
- [KoZa96] A. Koifman und S. Zabele. A Reliable Adaptive Multicast Protocol. In *Proceedings of IEEE Infocom'96*, März 1996.
- [MRTW97] K. Miller, K. Robertson, A. Tweedly und M. White. StarBurst Multicast File Transfer Protocol (MFTP) Specification. Technischer Bericht, Work in Progress, Internet Draft, 1997.
- [Stra95] W.T. Strayer. Xpress Transport Protocol Specification, Revision 4.0. Technischer Bericht, XTP Forum, Santa Barbara, USA, 1995.
- [YaKT96] M. Yajnik, J. Kurose und D. Towsley. Packet Loss Correlation in the MBoone Multicast Network. Technischer Bericht 96(32), University of Massachusetts at Amherst, 1996.

Prozeß-orientierter Ansatz zur Spezifikation betreibergerechter Managementwerkzeuge

Christian Mayerl

Kurzfassung

In der Vergangenheit wurde ein Schwerpunkt in der Informatik auf die effektive und effiziente Entwicklung von Systemen zur Informationsverarbeitung (IV) gelegt. Aufgrund der immer größer werdenden Bedeutung der IV werden heute ebenso Konzepte benötigt, die einen effektiven und effizienten Betrieb dieser IV-Systeme ermöglichen. Derartige Konzepte sehen u.a. den Einsatz von rechnergestützten Managementwerkzeugen vor. Bestehende Werkzeuge sind jedoch zu sehr aus Sicht der Entwickler entstanden und erfüllen daher nur bedingt die Bedürfnisse heutiger Betreiber vernetzter Systeme. Dieser Beitrag zeigt ein pragmatisches Vorgehen auf, wie Managementwerkzeuge aus der Sicht des Betreibers spezifiziert und entwickelt werden können.

1 Einleitung

In den letzten Jahren hat sich in vielen IV-Bereichen der Trend weg von Großrechnern hin zu vernetzten Systemen vollzogen. Die Kosteneinsparungen bei der Beschaffung der Hardware sind dabei mit dem Mehraufwand für den Betrieb von vernetzten Systemen erkauft worden. Gleichzeitig hat die Bedeutung der IV innerhalb und zwischen Unternehmen zugenommen. Um die steigenden Kosten und Anforderungen an die IV bewältigen zu können, entwickelt sich der Betreiber vernetzter Systeme vom einstigen Systemexperten immer mehr zum zentralisierten Dienstleister.

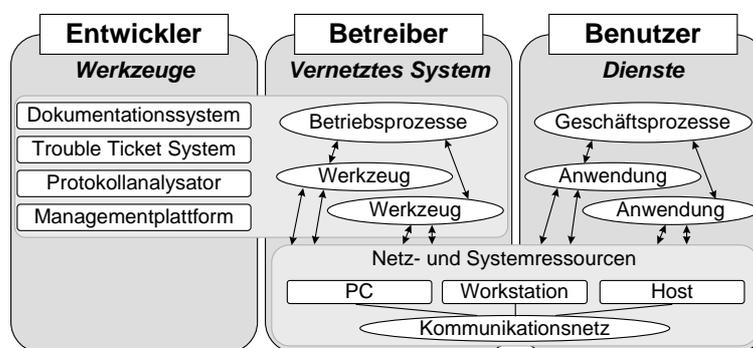


Abbildung 1: Rollen und Sichten auf das vernetzte System

Abbildung 1 zeigt einen Überblick über die an vernetzten Systemen beteiligten Rollen und deren Sichten. Zwischen dem Betreiber und seinen Kunden, den Benutzern

von Anwendungsdiensten und weiteren Dienstleistungen wie z.B. Schulungen, werden Dienstkataloge [Wall93] definiert, anhand der darin enthaltenen Dienste der Betreiber Dienstleistungsvereinbarungen (DLVs) mit den Benutzern abschließt. Diese Vereinbarungen legen Bedingungen bzgl. der Abrechnung, Verfügbarkeit, Qualität usw. der zu benutzenden Dienste fest. Das Ziel des IV-Betreibers ist es, das vernetzte System gemäß diesen DLVs zu betreiben. Um den Betrieb dabei effektiv und effizient zu gestalten, setzt er Managementwerkzeuge, u.a. für die Überwachung und Steuerung von Netz- und Systemkomponenten, ein. Jedoch sind bestehende Managementwerkzeuge aus der Sicht der Netz-, System- und Anwendungsentwickler entstanden und erfüllen daher nur bedingt die Anforderungen des Betreibers. Eine aus der Lücke zwischen den entwicklergetriebenen Managementwerkzeugen und den Anforderungen seitens des IV-Betreibers sich ergebende Fragestellung ist die Spezifikation und Entwicklung von betreibergerechten Managementwerkzeugen, mit denen ein vernetztes System effektiv und effizient gemäß den abgeschlossenen Dienstleistungsvereinbarungen betrieben werden kann.

2 Betrieb von vernetzten Systemen

Eine Voraussetzung für die Spezifikation und Entwicklung von betreibergerechten Managementwerkzeugen ist das hinreichende Verständnis des Betriebs von vernetzten Systemen [AbMa97]. Dieses Verständnis kann nicht durch das Aufstellen mathematischer Formalismen abgeleitet, sondern muß aus den in der Praxis vorhandenen Erfahrungen gewonnen werden. Insbesondere hat man im Bereich der Telekommunikation die Notwendigkeit erkannt, den Betrieb vernetzter Systeme auf die Erbringung von IV-Diensten auszurichten [ChKo97]. Eine umfangreiche Vorarbeit stellt dabei die *Information Technology Infrastructure Library (ITIL)* [CCTA94] der *Central Computer and Telecommunications Agency (CCTA)* dar, die die wesentlichen Betriebsaspekte für den Betrieb von IV-Systemen beschreibt.

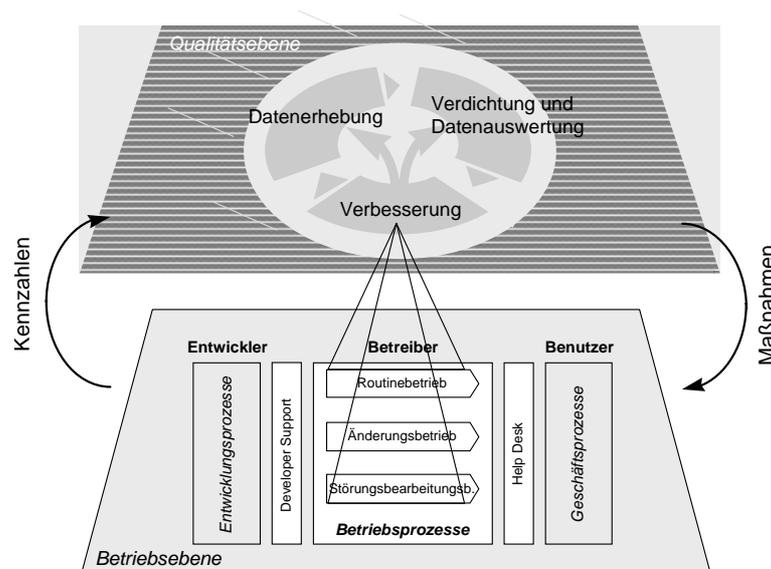


Abbildung 2: Prozeßmodell für den Betrieb vernetzter Systeme

In zahlreichen Projekten mit der Industrie, insbesondere mit Betreibern großer vernetzter Daten- und Telekommunikationssysteme, wurden Abläufe, Rollen und einzusetzende

Hilfsmittel für den Betrieb von vernetzten Systemen analysiert. Aus den Ergebnissen dieser Untersuchungen lassen sich grundsätzliche Vorgehensweisen und Strukturen für den Betrieb erkennen [Tele97]. Abbildung 2 faßt die Abläufe rund um das vernetzte System in einem Prozeßmodell [Zars98] zusammen: Über das **Help Desk** kommuniziert der Betreiber mit dem Benutzer bzgl. der benutzten Dienste und schließt die DLVs ab. Zu den Entwicklern hin wird eine mit dem Help Desk vergleichbare Schnittstelle, der **Developer Support**, definiert, über die der Betreiber mit den Entwicklern kommuniziert. Die über das Help Desk geforderten Dienste werden den Benutzern gemäß den DLVs durch das Zusammenspiel der *Betriebsprozesse* (Routinebetrieb, Änderungsbetrieb, Störungsbearbeitungsbetrieb) bereitgestellt. Ein Betriebsprozeß wird dabei mit der Bearbeitung bestimmter Aufgaben beauftragt:

- Der **Routinebetrieb** (engl. **Day-to-Day Operations**) ist für den Normalbetrieb des vernetzten Systems verantwortlich. Dabei führt er alltäglich wiederkehrende und sich etablierte Tätigkeiten am vernetzten System durch. Beispiele für Tätigkeiten des Routinebetriebs sind das Überwachen der Systemzustände, der Refresh von Systemen sowie das regelmäßige Backup aller wichtigen Daten.
- Der **Änderungsbetrieb** (engl. **Operations of Change**) führt Änderungen im vernetzten System koordiniert und weitestgehend geplant durch. Derartige Eingriffe überführen das vernetzte System in einen neuen technologischen Zustand und stellen eine Gefahr für die Einhaltung bestehender DLVs dar. Beispiele für Tätigkeiten des Änderungsbetriebs sind die Einführung neuer Software sowie die Einführung neuer Hardware.
- Der **Störungsbearbeitungsbetrieb** (engl. **Operations of Trouble**) nimmt Fehlermeldungen entgegen und bearbeitet sie. Seine Aufgabe ist es, durch eine möglichst schnelle Beseitigung einer Störung das vernetzte System in einen fehlerfreien Zustand zu versetzen, in dem es die DLVs erfüllt. Beispiele für Störungen sind der Ausfall von Hardware- und Software-Systemen sowie die Unterbrechung einer Kommunikationsverbindung.

Den genannten Betriebsprozessen sind Prozesse zur Überwachung und Sicherstellung der Qualität des IV-Betriebs übergeordnet. Diese *Qualitätsprozesse* besitzen einen globalen Überblick über die Betriebsprozesse. Zyklisch werden Informationen (Kennzahlen) aus den Betriebsprozessen gewonnen und ausgewertet. Aufgrund der Analyse der Auswertungsergebnisse werden Maßnahmen und Verbesserungen in den Betriebsprozessen angestoßen.

Für ein hinreichendes Verständnis des Betriebs vernetzter Systeme werden die Prozesse in Teilprozesse (Phasen und Aktionen) zerlegt. Beispielhaft seien an dieser Stelle die Phasen des Änderungsbetriebs genannt: In der **Request-Phase** wird der Änderungswunsch konkretisiert und an den Änderungsbetrieb gerichtet. In der **Commitment-Phase** werden Grobabschätzungen bzgl. Aufwand und Risiken einer Änderung getätigt sowie technische Leistungsanforderungen an die Systeme festgelegt. Die Eingriffe in das vernetzte System erfolgen in der **Performance-Phase** zunächst im Testbetrieb, bevor die Änderung in den Wirkbetrieb übernommen wird. Die durchgeführte Änderung wird in der **Evaluation-Phase** beurteilt und für den Wirkbetrieb freigegeben. Beispiele für Aktionen in den einzelnen Phasen sind in der Request-Phase das

Formulieren des Änderungswunsches sowie das Klassifizieren der eingereichten Änderungen in der Commitment-Phase. Anhand der Aktionen werden Hilfsmittel identifiziert, die den Betreiber bei der Durchführung seiner Tätigkeiten unterstützen. Aufgrund der Ausrichtung dieser Hilfsmittel werden diese im folgenden **Prozeß-orientierte Managementmittel (PoM)** genannt. Ein Beispiel für ein PoM ist das Änderungsauftragsformular, das den Initiator bei der Formulierung seines Änderungswunsches unterstützt. Ein weiteres PoM ist der Änderungsreihenfolgeplan bei der Klassifizierung der Änderungen, der Hilfestellungen bei der Planung anstehender Änderungen z.B. eines Jahres gibt. Eine umfangreiche Beschreibung des Änderungsbetriebs sowie der einzusetzenden PoMs ist [Zars98] zu entnehmen. Die Beschreibungen dieser PoMs sowie der Betriebsprozesse stellen Anforderungsspezifikationen für betreibergerechte Managementwerkzeuge dar.

3 Betreibergerechte Managementwerkzeuge

Aus der Analyse der Abläufe für den Betrieb von vernetzten Systemen sowie der einzusetzenden PoMs haben sich grundlegende Anforderungen für eine betreibergerechte und Prozeß-orientierte Werkzeugumgebung ergeben. Einige dieser Anforderungen werden im folgenden genannt; die Liste erhebt nicht den Anspruch auf Vollständigkeit, sondern ist je nach Szenario bzw. Betreiber zu konkretisieren und zu erweitern.

- Die PoMs spezifizieren Werkzeugmodule, die den Betreiber in den entsprechenden Aktionen unterstützen. Da sich die Prozesse von Betreiber zu Betreiber unterscheiden können und sich einer Evolution unterwerfen, muß die Möglichkeit gewährleistet sein, die Werkzeugumgebung an das konkrete Szenario anzupassen (*Customizing*). Die Werkzeugumgebung muß entsprechend **modular** und **skalierbar** sein.
- Die einzelnen Werkzeugmodule müssen in der Lage sein, Informationen untereinander austauschen und bearbeiten zu können. Dazu müssen sie, u.U. auch über größere Entfernungen hinweg, miteinander **kommunizieren** sowie **kooperieren** können. Die Kommunikation und Kooperation auf der Werkzeugebene unterstützen zudem die Kommunikation und Kooperation der im Prozeß beteiligten Rollen.
- Damit die PoMs effizient im Sinne der Prozeßdefinition eingesetzt werden, bedarf es einer Anleitung der Rollen. Die Anleitung durch die Werkzeugumgebung **koordiniert** dabei den Einsatz der Werkzeugmodule. Jedoch sollte die Anleitung jeder Zeit die Möglichkeit zur Behandlung von Ausnahmesituationen gewährleisten.

Bestehende Ansätze zur Entwicklung von Werkzeugen für den Betrieb von vernetzten Systemen erfüllen die genannten Anforderungen nur bedingt oder gar nicht. Im Bereich des Netz- und Systemmanagement [HeAb93] sind Werkzeuge vorwiegend aus der Sicht von Entwicklern entstanden und haben ihren Schwerpunkt in der Überwachung und Steuerung der technischen Netz- und Systemkomponenten. Beispiele dieser Werkzeuge sind Netz- und Systemmonitore sowie Dokumentationssysteme, die dynamische bzw. statische Informationen über das vernetzte System liefern. Meist sind diese Werkzeuge in Managementplattformen, wie OpenView von Hewlett Packard oder Spectrum von Cabletron, integriert. Derartige Werkzeuge leiten den Betreiber i.d.R. nicht an, seinen

Prozeß effizient zu durchlaufen. Werkzeuge, die die (verteilte) Gruppenarbeit unterstützen, realisieren zwar die Kooperation sowie die Kommunikation verteilter Rollen, jedoch garantieren sie auch nicht den Einsatz dieser Werkzeuge im Sinne eines effizienten Prozeßdurchlaufs. Eine derartige Koordination muß außerhalb der Werkzeugumgebung erfolgen. Workflow Management Systeme [Holl94] hingegen schreiben den Einsatz von Werkzeugen gemäß einer detaillierten Ablaufbeschreibung (*Workflow*) vor, die zur *Build-time* erstellt wurde. Zur *Run-time* ist der Prozeß strikt nach den spezifizierten Workflows abzuarbeiten. Ausnahmen, die zur *Build-time* nicht spezifiziert wurden, können zur *Run-time* mit bestehenden Workflow Management Systemen nur bedingt oder gar nicht bearbeitet werden.

Ziel ist es, eine Werkzeugumgebung zu entwickeln, die den Anforderungen des Betreibers vernetzter Systeme entspricht. Dabei soll die Werkzeugumgebung den Einsatz der modularen Werkzeuge dahingehend koordinieren, daß ein Prozeß effizient durchlaufen werden kann, ohne die Möglichkeit zu verwehren, jeder Zeit Ausnahmen zu behandeln. Das *Guided Cooperation Concept (GCC)* sieht dazu die Definition von richtungsweisenden Richtlinien vor, die jedoch genügend Freiraum bei der Durchführung von Tätigkeiten gewährleisten.

4 Ausblick

Zur Unterstützung des Änderungsbetriebs wird prototypisch die betreibergerechte Werkzeugumgebung *Cooperative IT Change Control (CICC)* entwickelt. Für die Architektur von CICC lassen sich aus den Anforderungen des Betreibers funktionale Bausteine ableiten.

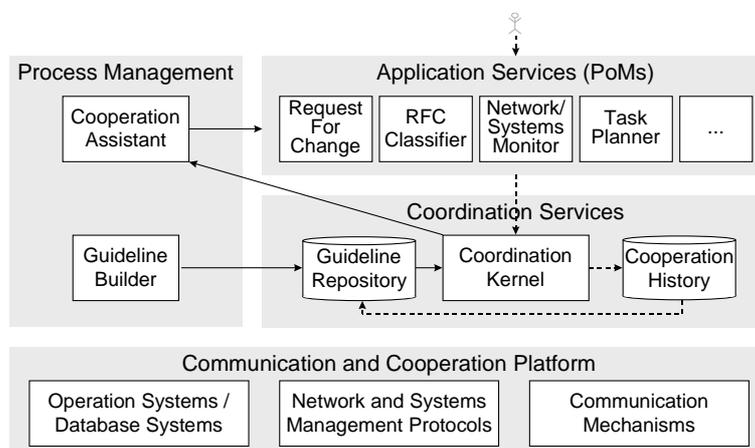


Abbildung 3: Architektur der Werkzeugumgebung CICC

Abbildung 3 zeigt die Architektur der Werkzeugumgebung CICC. Die Grundlage der verteilten Werkzeugumgebung CICC bildet eine Plattform für die Kommunikation und Kooperation. Diese stellt Basisdienste für die persistente Speicherung (**Operation Systems / Database Systems**) sowie für den Austausch und die Verarbeitung von Daten bereit. Die Kommunikation kann dabei zwischen dem zu überwachenden und zu steuernden vernetzten System (**Network and Systems Management Protocols**) bzw. zwischen den beteiligten Rollen und den eingesetzten Werkzeugmodulen (**Communication Mechanisms**) erfolgen. Eine weit verbreitete und bevorzugte

Plattform für die Kommunikation in verteilten Umgebungen ist die *Common Object Request Broker Architecture (CORBA)*. Darauf aufbauend setzt der **Coordination Kernel** als zentraler Baustein auf, der den Einsatz der Werkzeugmodule (PoMs) protokolliert (**Cooperation History**) sowie die technische Koordination der Werkzeugmodule übernimmt. Der Coordination Kernel greift dabei auf das **Guideline Repository** zurück, das Richtlinien beinhaltet, nach denen die Werkzeugumgebung angeleitet werden soll. Der **Guideline Builder** unterstützt den Prozeßverantwortlichen bei der Definition der Richtlinien. Ein **CICC Assistant** schließlich greift die technische Koordination durch den Coordination Kernel auf und leitet die Rollen beim Einsatz der Werkzeugmodule (PoMs) an. Dabei vermittelt er der jeweiligen Rolle einen schnellen Überblick über den aktuellen Handlungsbedarf und spricht Empfehlungen für den Einsatz der PoMs aus. Zusätzlich gewährt er alternative Handlungsmöglichkeiten, indem weitere Ansprechpartner bzw. PoMs angeboten werden, um z.B. Ausnahmesituationen behandeln zu können. Auf die genannten Bausteine setzten die als PoMs spezifizierten Anwendungsmodule auf. Beispiele für Anwendungsmodule der Werkzeugumgebung CICC sind das **RFC-Modul**, welches das Änderungsauftragsformular als JAVA-Applet realisiert, der **Network / Systems Monitor** sowie der **Task Planner**, der u.a. bei der Erstellung des Änderungsreihenfolgeplans Verwendung findet.

Literatur

- [AbMa97] Sebastian Abeck und Christian Mayerl. Prozeßbeschreibungen als Basis für einen qualitätsgesicherten Betrieb von vernetzten Arbeitsplatzrechnern. *Proceedings der 4. ITG/GI-Fachtagung Arbeitsplatz-Rechensysteme (APS'97)*, Mai 1997, S. 217–225.
- [CCTA94] CCTA. *Central Computer and Telecommunications Agency: The IT Infrastructure Library: An Introduction*. HMSO, London. 1994.
- [ChKo97] G. Chen und Q. Kong. Integrated TMN Service Provisioning and Management Environment. *Fifth IFIP/IEEE International Symposium on Integrated Network Management*, Mai 1997.
- [HeAb93] Heinz-Gerd Hegering und Sebastian Abeck. *Integriertes Netz- und Systemmanagement*. Addison-Wesley. 1993.
- [Holl94] D. Hollingsworth. The Workflow Reference Model. Specification tc00-1003, Workflow Management Coalition, 1994.
- [Tele97] Deutsche Telekom. *Qualitätshandbuch der Service- und Computer-Zentren*. 1997.
- [Wall93] D. Wall. *Rechner, Netze, Spezialisten: Leistungsangebot der GWDG*. Gesellschaft für wissenschaftliche Datenverarbeitung. 1993.
- [Zars98] Boris Zarske. Entwurf eines Prozeßmodells für den Betrieb von verteilten Systemen. Diplomarbeit, Institut für Telematik, Universität Karlsruhe, 1998.

Effiziente Authentisierungsverfahren für Hochleistungsnetze

Günter Schäfer

Kurzfassung

Dieser Text ist eine Kurzfassung der Dissertation *Effiziente Authentisierung und Schlüsselverwaltung in Hochleistungsnetzen*. In der Arbeit wurden ausgehend von den Anforderungen an die *Authentisierung* und die eng hiermit zusammenhängende *Schlüsselverwaltung* in modernen Hochleistungsnetzen Verfahren für die effiziente Authentisierung und Schlüsselverwaltung entwickelt, die über eine reine Betrachtung der *Sicherheit* hinaus auch den in Hochleistungsnetzen wichtigen Aspekten *Leistungsfähigkeit* und *Verwaltbarkeit* gerecht werden.

Zukünftige Telekommunikationsdienste wie Teleshopping, Telebanking oder Teleworking stellen erheblich höhere Anforderungen an die Sicherheit und zumeist auch an die Leistungsfähigkeit der zugrundeliegenden Übertragungsdienste als die herkömmliche Sprach- bzw. Datenkommunikation. Vertraulichkeit der übermittelten Daten, Authentizität der Kommunikationspartner, Nachweisbarkeit von Kommunikationsvorgängen und kontrollierter Zugang zu den Diensten sind wichtige Voraussetzungen für den sicheren Betrieb fortgeschrittener Kommunikationsdienste [Ford94]. Bei der Integration der hierfür erforderlichen Sicherheitsmechanismen in die Kommunikationsinfrastruktur ist jedoch zu gewährleisten, daß deren Leistungsparameter nicht zu sehr herabgesetzt werden. Diese umfassen neben Verkehrsparametern wie beispielsweise der erzielbaren Datenübertragungsrate, der Ende-zu-Ende-Verzögerung sowie der Varianz der Ende-zu-Ende-Verzögerung auch den Zeitaufwand für die Einrichtung einer gesicherten Kommunikationsbeziehung. Aus diesem Grund sollten Sicherheitsmechanismen möglichst in den unteren Schichten der Protokollarchitektur realisiert werden, in denen sie durch dedizierte Hardware optimiert werden können. Eine besondere Bedeutung kommt bei dieser Integration der Realisierung des Authentisierungsdienstes und der in großen Netzen durch sie beeinflussten Schlüsselverwaltung zu. Der Authentisierungsdienst ermöglicht kommunizierenden Geräten die gegenseitige Überprüfung der Identität. Es ist somit zum einen zu gewährleisten, daß den Protokollinstanzen der unteren Schichten die für einen authentisierten Verbindungsaufbau benötigten Schlüssel zur Verfügung stehen. Zum anderen ist darauf zu achten, daß sowohl der hierfür in den Systemen des Netzes erforderliche Ressourcenbedarf (Speicher, Prozessorleistung) als auch die durch die Authentisierung eingeführte Verbindungsaufbauverzögerung nicht zu hoch ausfallen.

Das Ziel der vorliegenden Arbeit ist es, ein Authentisierungsverfahren für den Einsatz in Hochleistungsnetzen und eine darauf abgestimmte Schlüsselverwaltung zu entwerfen, die über die reine Betrachtung der Sicherheit hinaus den in Hochleistungsnetzen wichtigen Aspekten Leistungsfähigkeit und Verwaltbarkeit gerecht werden. Grundlegende

Merkmale der beiden Verfahren sind neben hoher Verarbeitungsgeschwindigkeit bei der Authentisierung auch die Eignung der Schlüsselverwaltung für den Einsatz in Netzwerken globaler Ausdehnung. Die Verfahren unterstützen eine flexible Aushandlung der zu verwendenden kryptographischen Algorithmen und sind damit auch in bezug auf zukünftige Erkenntnisse der kryptographischen Forschung erweiterbar.

Die in Authentisierungsprotokollen verwendeten kryptographischen Algorithmen können in die drei grundlegenden Kategorien asymmetrische Verschlüsselung, symmetrische Verschlüsselung und kryptographische Hash-Funktionen eingeteilt werden [Schn96]. Hierbei ermöglicht nur die Kategorie der asymmetrischen Chiffrierverfahren die Realisierung einer Schlüsselverwaltung auch für sehr große Benutzergruppen auf der Grundlage zertifizierter Schlüssel, die in öffentlichen Schlüsselverzeichnissen gespeichert werden können. Diese Verfahren erfordern jedoch einen sehr hohen Rechenaufwand im Vergleich zu den Verfahren der beiden anderen Kategorien. Mit symmetrischen Chiffrierverfahren kann etwa um einen Faktor 100 bis 1000 schneller verschlüsselt werden. Sie ermöglichen allerdings nur Benutzergruppen mittlerer Größe, da bei ihnen die Anzahl erforderlicher Schlüssel quadratisch mit der Anzahl möglicher Authentisierungspartner wächst. Mit kryptographischen Hash-Funktionen kann noch schneller signiert werden, sie eignen sich jedoch nicht für die Verschlüsselung von Daten.

Der grundlegende Ansatz dieser Arbeit zur Erreichung einer möglichst effizienten Authentisierung beim Verbindungsaufbau liegt darin, die Verfahren der einzelnen Kategorien miteinander zu kombinieren und so ihre spezifischen Vorteile zu vereinen. Die Arbeit entwirft ein auf der ITU-T-Empfehlung X.509 [ITU93] beruhendes Authentisierungsprotokoll, das wahlweise mit zwei oder drei ausgetauschten Nachrichten durchgeführt wird, und bei dem der für die Signatur der Nachrichten verwendete kryptographische Algorithmus aus einer der drei genannten Kategorien gewählt und zwischen den Systemen flexibel ausgehandelt werden kann. Hierbei werden initiale und erneute Authentisierungen zwischen zwei Systemen unterschieden. Initiale Authentisierungen handeln zusätzlich einen sogenannten Authentisierungskontext aus, der lokal in beiden Systemen zwischengespeichert wird und bei zukünftigen erneuten Authentisierungen Schlüssel für die effiziente Authentisierung auf der Grundlage symmetrischer Chiffrierverfahren bzw. kryptographischer Hash-Funktionen zur Verfügung stellt. Mit diesem Ansatz werden erneute Authentisierungen erheblich beschleunigt, die in Kommunikationsnetzen zum einen als Reaktion auf Fehler im Netz und zum anderen anwendungsbedingt (z.B. für WWW-Verkehr) sehr häufig erforderlich sind.

Die zu zertifizierten öffentlichen Schlüsseln gehörenden privaten Schlüssel werden sicher in Prozessorchipkarten gespeichert, da bei ihrer Kompromittierung ein erheblicher Aufwand für die Invalidierung in den öffentlichen Schlüsselverzeichnissen entsteht und auch der Aufwand für die Überprüfung eines Schlüssels mit der Anzahl invalidierter Schlüssel steigt. Bei der initialen Authentisierung verwendet ein Endsystem jedoch nicht direkt den privaten Schlüssel auf der Chipkarte, da Prozessorchipkarten erheblich langsamer rechnen als integrierte Prozessoren, sondern einen periodisch vom Endsystem selbst generierten und mit Hilfe der Chipkarte zertifizierten asymmetrischen Arbeitsschlüssel, dessen Gültigkeitsdauer so kurz ist, daß er nicht invalidiert werden muß. Insgesamt definiert die Arbeit somit eine hybride Schlüsselhierarchie für die effiziente Authentisierung in Hochleistungsnetzen.

Mit der Integration von Sicherheitsdiensten in die unteren Protokollschichten der Kommunikationssysteme entsteht auch die für den Betrieb und die Wartung größerer Netz-

werke unverzichtbare Forderung nach einer *Verwaltungsunterstützung* sicherheitsspezifischer Parameter. So ist es zum einen wichtig, den Status der Systeme auch in bezug auf sicherheitsrelevante Aspekte fortlaufend überwachen zu können, zum anderen wird ein Mechanismus benötigt, der es erlaubt, Parameter im laufenden Betrieb von einer Managementkonsole aus zu verändern. Hierfür sollten möglichst existierende Verfahren des Netzwerkmanagements eingesetzt werden, um den Betriebskräften eine integrierte Werkzeugunterstützung bieten zu können. In der vorliegenden Arbeit werden die Verwaltungsaspekte des Authentisierungsdienstes und der eingeführten hybriden Schlüsselhierarchie identifiziert und eine entsprechende Managementdatenbasis spezifiziert. Mit Hilfe dieser Datenbasis werden beispielsweise statistische Daten über erfolgte Authentisierungen erfaßt und Schwellwerte für die Generierung von Alarmen ab einer gewissen Anzahl fehlgeschlagener Authentisierungen festgelegt. Des weiteren können auch die zu verwendenden kryptographischen Algorithmen zielbezogen vorgegeben und die Gültigkeitsdauer zwischengespeicherter Authentisierungskontexte, Zertifikate sowie selbst erzeugter Schlüssel eingestellt werden. Um diese Parameter vor unberechtigtem Zugriff zu schützen, werden gesicherte Managementprotokolle für die Kommunikation zwischen der Managementanwendung und den zu verwaltenden Einheiten eingesetzt.

Das in der Arbeit entworfene Authentisierungsprotokoll wurde in einer formalen Betrachtung unter Verwendung der nach ihren Erfindern Gong, Yahalom und Needham benannten GNY-Logik [GoNY90] auf mögliche Sicherheitslücken hin analysiert. Hierbei wurde für die Untersuchung des Zwei-Wege-Dialogs eine Regel für die Behandlung von Zeitstempeln in der GNY-Logik ergänzt, die auf Gültigkeitsintervallen und dem Vergleich mit einer lokalen Systemzeit beruht. Es konnte gezeigt werden, daß sowohl die initiale als auch die erneute Authentisierung jeweils in der Zwei-Wege- und der Drei-Wege-Variante die intendierten Ziele des Authentisierungsprotokolls bei beiden Partnern erreicht.

Der Einsatz des in dieser Arbeit entwickelten Ansatzes wurde in den beiden wichtigsten aktuellen Protokollarchitekturen, dem Asynchronen Transfermodus (ATM) und der dem Internet zugrundeliegenden TCP/IP-Protokollfamilie, untersucht. Für ATM-Netze wurde aufgezeigt, wie das Authentisierungsprotokoll und die zugehörige Schlüsselverwaltung in die Sicherheitsarchitektur [ATM97] des ATM-Forums, einem internationalen Konsortium zur Entwicklung und Förderung der ATM-Technologie, integriert werden kann. Hierzu wurde eine formale Spezifikation der ATM-Sicherheitsarchitektur in der von der ITU-T genormten Beschreibungssprache SDL (Functional Specification and Description Language) erstellt und um das entwickelte Authentisierungsprotokoll und die Schlüsselverwaltung ergänzt. Die Spezifikation wurde mit dem kommerziellen SDL-Entwicklungswerkzeug ObjectGeode implementiert, wobei kryptographische Algorithmen sowie der Zugriff auf Prozessorchipkarten in der Programmiersprache C realisiert wurden. Das resultierende Gesamtsystem wurde mit dem Entwicklungssystem simuliert und sein Verhalten anhand von Message Sequence Charts nach ITU-T-Empfehlung Z.120 dokumentiert. Insgesamt konnte somit die Eignung der Verfahren für die ATM-Protokollarchitektur anhand einer formalen Spezifikation untersucht und qualitativ nachgewiesen werden.

Eine quantitative Untersuchung wurde anhand einer Implementierung der entwickelten Verfahren als Schlüsselverwaltungsprozeß für ein sicheres Internet-Protokoll gemäß der IP-Security-Architektur *IPSec* [Atki95] der Internet Engineering Task Force (IETF), dem internationalen Normungsgremium für die Internet-Technologie, durchgeführt.

Hierzu wurden die Sicherheitsmechanismen von IPsec in den Kern des Betriebssystems Linux integriert und die im Rahmen dieser Arbeit entwickelten Authentisierungs- und Schlüsselverwaltungsverfahren als zugehöriger Key Management Daemon implementiert. Durch detaillierte Messungen der Verbindungsaufbauverzögerung und der Verarbeitungszeiten einzelner Schritte unterschiedlicher Authentisierungsvarianten konnten die Vorteile der hybriden Schlüsselhierarchie gegenüber existierenden Verfahren nachgewiesen werden. So ist bei Verschlüsselung in Software und einer angenommenen Höhe der Zertifizierungshierarchie von vier Hierarchiestufen die erneute Authentisierung etwa um einen Faktor 200 schneller als die initiale Authentisierung. Dieser Unterschied kann zwar durch Zwischenspeicherung von Zertifikaten deutlich reduziert werden. Jedoch ist eine mittels kryptographischer Hash-Funktion durchgeführte erneute Authentisierung bereits aufgrund des geringeren algorithmischen Berechnungsaufwands um einen Faktor 10 bis 100 schneller als eine auf asymmetrischer Verschlüsselung basierende initiale Authentisierung. Insbesondere bei Berechnung der asymmetrischen Verschlüsselung in einer Prozessorchipkarte kann die in fortgeschrittenen Hochleistungsnetzen erforderliche Leistungsfähigkeit nicht ohne die in dieser Arbeit entwickelten Verfahren erreicht werden.

Verglichen mit anderen Ansätzen zur Authentisierung und Schlüsselverwaltung in Netzwerken zeigt sich, daß die in der Arbeit entworfenen Verfahren sich nicht – wie bisher üblich – nur auf eine Spezifikation zu verwendender Nachrichtenformate beschränken, sondern auch eine Strategie für die Auswahl des in einer bestimmten Situation effizientesten kryptographischen Algorithmus vorgeben und die hierfür erforderlichen Schlüssel in den Systemen bereitstellen. Zu den hervorzuhebenden Eigenschaften gehört, daß zertifizierte Schlüssel sicher in Chipkarten gespeichert werden und dennoch eine effiziente Authentisierung pro Kommunikationsbeziehung realisiert werden kann. Mit einer formalen Spezifikation in der genormten Sprache SDL und einer umfangreichen Implementierung für das Betriebssystem Linux wurde über die theoretische Fundierung hinaus auch die praktische Eignung der entwickelten Verfahren nachgewiesen.

Literatur

- [Atki95] R. Atkinson. *Security Architecture for the Internet Protocol*, August 1995. RFC 1825.
- [ATM97] ATM Forum. *Phase I ATM Security Specification*, Februar 1997. (Draft).
- [Ford94] Warwick Ford. *Computer Communications Security – Principles, Standard Protocols and Techniques*. Prentice Hall. 1994.
- [GoNY90] L. Gong, R. M. Needham und R. Yahalom. Reasoning about Belief in Cryptographic Protocols. In *Symposium on Research in Security and Privacy*. IEEE Computer Society, IEEE Computer Society Press, May 1990, S. 234–248.
- [ITU93] ITU-T. *X.509: Information Technology – Open Systems Interconnection – The Directory: Authentication Framework (4)*, 1993.
- [Schn96] B. Schneier. *Applied Cryptography Second Edition: Protocols, Algorithms and Source Code in C*. John Wiley & Sons. 1996.

Rückgekoppeltes, parametrisierbares Scheduling zur Erbringung von Dienstgüte in mehrschichtigen Kommunikationssystemen

Hartmut Ritter und Jochen Schiller*

Kurzfassung

Traditionelle Verfahren zur Erbringung von Dienstgüte in Kommunikationssystemen scheitern oft aus Gründen der Komplexität, der Inkompatibilität zu existierenden Anwendungen oder auch des Bedarfs grundlegend neuer Rechnerarchitekturen. Soll ein Ansatz erfolgreich sein, so muß er in weit verbreitete Systeme integrierbar, einfach und beherrschbar sein. Der folgende Beitrag zeigt einige grundlegende Probleme bisheriger Ansätze auf und stellt ein Kommunikationssystem vor, das im wesentlichen auf rückgekoppelten, parametrisierbaren Schedulingern basiert. Damit kann dynamisch auf Änderungen von Systemparametern und äußeren Einflüssen eingegangen werden. Eine notwendige Komponente zur Erbringung von Dienstgüte für Kommunikationsanwendungen ist ein Scheduler für das Senden von Datenpaketen, welcher detaillierter vorgestellt wird. Abschließend findet eine Diskussion über weitere notwendige Schritte hin zu einem einfachen, dienstgüteunterstützenden System statt.

1 Einleitung

Eine Grundfrage, die schon einige Arbeiten am Institut motiviert hat, ist die Erbringung einer bestimmten, eventuell garantierten Dienstgüte für Kommunikationsanwendungen durch das darunterliegende Kommunikationssystem. Die generelle Motivation für Dienstgüte, kurz QoS (Quality of Service), liegt in den Anforderungen vielfältiger multimedialer Anwendungen, die über ein gewisses Spektrum an Kommunikationsmedien (Funk, Glasfaser, Kupferkabel etc.) miteinander kommunizieren. Diese Medien bieten alle aufgrund natürlicher Gegebenheiten, der verschiedenen Medienzugriffsprotokolle, Teilnehmerzahlen, Distanzen etc. unterschiedliche Dienstqualitäten für die Protokolle der höheren Schichten an. Diese versuchen nun, die Unterschiede zwischen den Medien zu verdecken und definieren ein abstraktes Medium mit spezifischen Qualitätsparametern. Ein Beispiel ist die ATM-Technologie, die Parameter wie Zellverlustrate, durchschnittliche Bandbreite, Verzögerungsschwankung und vieles mehr erlaubt festzulegen.

*Die Beiträge wurden bewußt zusammengefaßt, da sie unter den gleichen Voraussetzungen und Annahmen stehen und sich gegenseitig ergänzen.

Eine Vielzahl von Arbeiten weltweit in den letzten Jahren hat nun versucht, diese Parameter vom Netz durch das Endsystem hindurch den Anwendungen zur Verfügung zu stellen. Dazu wurden Mechanismen zur Abbildung von sog. QoS-Parametern der Anwendungen (z.B. Bildrate, Audioqualität) auf QoS-Parameter des Endsystems (z.B. CPU und Speicherbedarf) entwickelt, letztere wurden dann wiederum auf Parameter abgebildet, die von der Netzwerktechnologie „verstanden“ werden konnten (z.B. maximale Bandbreite, maximale Verzögerung). Das Resultat dieser Bestrebungen waren meist sehr komplizierte sog. QoS-Architekturen, die oft eine schichtenweise Abbildungen der Parameter vorsahen, eine Vielzahl von Funktionen und Regeln für die Abbildung aufstellten und im wesentlichen kaum eine breite Verbreitung fanden ([Camp94]). Dies hat mehrere Gründe:

- Blickt man zurück auf die Techniken, die im Bereich der Datenkommunikation erfolgreich waren, so arbeiteten die meisten nach dem sog. KISS-Prinzip¹. IP, Ethernet sind hierfür die besten Beispiele. Nicht die komplizierteste, alle Fälle abdeckende Technik hat sich durchgesetzt, sondern diejenige, die billig und einfach war, dennoch die Hauptaufgabe erfüllte. Selbstverständlich hat man gesehen, daß mit diesen Techniken, die alle nach der „best-effort“-Methode arbeiten, nur sehr schwer bestimmte Dienstgütern eingehalten bzw. garantiert werden können. Diese werden jedoch benötigt, sollten die neuen Anwendungen wirklich einen Qualitätsgewinn darstellen. Dennoch bleibt festzuhalten, daß komplexe Architekturen im Ende viel zu teuer für den Markt waren.
- Die meisten der neuen QoS-Architekturen benötigten gleich auch neue Betriebssysteme, neue Anwendungsprogramme etc., um ihre Leistungsfähigkeit voll auszuspielen. Auch hier hat die Geschichte gezeigt, daß Anwender nur sehr selten bereit sind (außer für Spezialanwendungen), auf ein komplett neues System umzusteigen. Auf vielen neu entwickelten Systemen konnten zwar die Mechanismen der QoS-Architekturen demonstriert werden, jedoch existierten keine weiteren Anwendungsprogramme bzw. die Programme waren inkompatibel zu existierenden, da sie komplett neu für die neue Architektur erstellt werden mußten.
- Aus wissenschaftlicher Sicht der Hauptgesichtspunkt ist die Beherrschbarkeit komplexer Systeme. Während die beiden vorigen Argumente eher auf historischer, marktorientierter Erfahrung basieren, so soll nun ein prinzipielles Problem aufgezeigt werden. Heutige Rechner bestehen nicht mehr aus einer einzelnen CPU, einem physikalischen Speicher und führen auf diesem System gleichzeitig nur ein Programm aus. Schon der einfachste PC beinhaltet eine Vielzahl von teilautonom agierenden Prozessoren, jeder oft mit eigenem Speicherbereich, eigenen Programmen etc. Beispiele hierfür sind Netzwerkadapter, Videokarten, SCSI-Adapter. Die meisten QoS-Architekturen decken lediglich die Haupt-CPU, das darauf laufende Betriebssystem (im wesentlichen den Scheduler) und die kommunikationsorientierten Anwendungen ab. Nicht berücksichtigt werden z.B. Unterbrechungen der Abarbeitung von Programmen durch eingehende Datenpakete vom Netz, Kopieraktionen zwischen z.B. Festplatten und Hauptspeicher können nur mit sehr

¹KISS principle: „Keep It Simple, Stupid“. Eine Maxime, die oft dann bei Entwurfsentscheidungen angewendet wird, wenn zu starker Reichtum an Fähigkeiten (featurism) und eine zu hohe Komplexität der Steuermechanismen eines Systems abgewehrt werden sollen. Der Begriff ist vermutlich aus dem Schlagwort für Verkaufspräsentationen „Keep It Short and Simple“ entstanden.

großem Aufwand vorhergesagt bzw. gesteuert werden, Programme wie Compiler, die nichts mit der Kommunikation zu tun haben, werden vernachlässigt. Daher kann eine beherrschbare QoS-Architektur nicht alle Parameter erfassen, muß aber dennoch versuchen, möglichst präzise Aussagen über den Systemzustand zu treffen und entsprechende Abbildungsfunktionen zwischen Parametern anzuwenden. Häufig wird also versucht, mit exakten Regeln basierend auf unscharfen Parametern wiederum scharfe Aussagen zu treffen. Unvorhersagbare Ereignisse machen dann häufig diese Systemprognosen und -entscheidungen hinfällig.

Faßt man diese 3 Punkte zusammen, so läßt sich eine neue Vorgehensweise bei der Architektur von Kommunikationssystemen beschreiben, die mit der Ungenauigkeit zurechtkommt, auf Veränderungen im System reagieren kann und dennoch bestimmte Garantien über das Systemverhalten leistet:

- Die Architektur muß einfach sein und ohne komplexe Parametersätze auskommen.
- Bestehende Anwendungen sollten weiterhin einsetzbar, die Neuerungen in verbreitete Systeme (z.B. PCs mit Unix/Windows) integrierbar sein.
- Anwender sollten möglichst wenig von den neuen Fähigkeiten „belastet“ werden, Anwendungen sich weitgehend autonom an Veränderungen anpassen.

Im folgenden wird eine generelle Systemarchitektur, basierend auf 2 Rückkopplungsschleifen vorgestellt, die in bestehende Systeme integriert werden kann und die Rücksicht auf nicht exakt bestimmbare Parameter nimmt. Im anschließenden dritten Kapitel folgt eine detailliertere Betrachtung eines Schedulers, der Grundlage zur Erbringung von QoS auf unterster Ebene, dem Versenden von Datenpaketen bzw. Zellen über das Netzwerk, ist. Anschließend werden weitere Schritte aufgezeigt, die noch zu erbringen sind, um die Architektur mit Leben zu erfüllen. Hier ist vor allem die Integration in Betriebssysteme, die Beherrschung der Rückkopplung und die Untersuchung adaptiver Anwendungen von Interesse.

2 Systemarchitektur für rückgekoppelte Kommunikationssysteme

In diesem Kapitel soll die Betrachtung verbreiteter Ansätze zu umfassenden QoS-Architekturen vertieft und anschließend eine flexible Systemarchitektur vorgestellt werden, die in der Lage ist, einige Nachteile der klassischen Ansätze zu vermeiden. Im traditionellen QoS-Management in Endsystemen herrschen zwei grundlegende Ansätze vor: Explizite Ressourcenreservierung auf der einen, Adaptivität der Anwendungen auf der anderen Seite. Während ersteres genaue Informationen über das zugrundeliegende Betriebssystem und die Architektur benötigt, verlangt letzteres von den Anwendungen die Anpassung an Änderungen der Ressourcen.

2.1 Zwei grundlegende Konzepte

Betrachten wir die Ansätze und ihre Vor- und Nachteile noch einmal genauer:

- **Adaptivität der Anwendungen**

Mit diesem Ansatz gibt das System den Anwendungen keinerlei Garantien. Statt dessen wird die Anwendung anfänglich mit gewissen Ressourcen ausgestattet und versucht dann, mit den gegenwärtig zur Verfügung stehenden Ressourcen, die vom System zugeteilt werden, die bestmögliche Dienstgüte zu erreichen [CoTe94], [JoBF96]. Es ist offensichtlich, daß der Endbenutzer keine konstante und vorher-sagbare Dienstqualität erwarten kann, da die Anwendungen ja stark von der wechselnden Systemauslastung abhängen.

- **Ressourcenreservierung**

Explizite Ressourcenreservierung in allen Teilen des Endsystems scheint der einzige Weg zu sein, um konstante Dienstgütegarantien abgeben zu können. Typisch für Ansätze in diesem Bereich sind folgende Mechanismen [AuCH98]:

- Dienstgüteabbildung (*QoS-mapping*)

Dienstgüteparameter der Anwendungen, z.B. Bildrate, Audioqualität, Synchronität, sind nicht ohne weiteres auf jeder Ebene des Endsystems und des Netzwerks direkt verarbeitbar. Das Betriebssystem, die Transportschicht oder das verwendete Übertragungsverfahren benötigen eine jeweils andere Darstellung der einzuhaltenden Parameter. Um den Benutzer von der Spezifikation der Dienstgüteparameter unterhalb der Anwendungsebene entlasten zu können, sehen diese Ansätze eine schichtenweise Abbildung der Dienstgüteparameter von der Anwendung durch das Betriebssystem hindurch bis hinunter auf Netzwerkebene vor. Die Anforderungen des Anwenders müssen so z.B. auf Dienstgüteklassen und Werte für maximale und durchschnittliche Zellrate abgebildet werden oder auf einzelne Zeitgrenzen und Perioden für das Scheduling von Zellen durch den Netzwerkadapter. Dies erfordert eine Vielzahl von Regeln und Funktionen für die Abbildung.

- Zulassungstests (*admission testing*)

Zur Bestimmung der Parameter der Abbildungsfunktionen werden oft Zulassungstests vorgeschlagen. Zulassungstests vergleichen den Ressourcenbedarf, der durch die Anforderungen des Anwenders entsteht, mit den verfügbaren Ressourcen des Endsystems und nehmen eine Vorreservierung vor. Manche Ansätze ermitteln den Ressourcenbedarf aus gegebenen Abbildungsfunktionen, andere Ansätze schlagen zur Anpassung der Parameter dieser Abbildungsfunktionen auch die Durchführung von Tests vor, um die jeweilige zugrundeliegende Hardware (z.B. Prozessorgeschwindigkeit, Speicherzugriffszeiten) in die Berechnungen einbeziehen zu können [NaSt95].

- Ressourcenreservierung (*resource reservation*)

Im Anschluß an den erfolgreichen Zulassungstest müssen die Ressourcen im Endsystem endgültig reserviert bzw. vorreservierte Ressourcen eventuell freigegeben werden. Die reservierten Ressourcen stehen dem System nicht mehr zur freien Verfügung, auch wenn sie nicht vollständig ausgeschöpft werden.

Neben den bereits geschilderten Nachteil der Notwendigkeit komplexer Abbildungsfunktionen tritt also noch ein hoher Aufwand durch die Ermittlung der aktuellen Auslastung der Ressourcen. Um harte Garantien bieten zu können, wird es in den meisten Fällen zu Überbuchungen von Ressourcen kommen, da dynamische Anpassungen schlecht unterstützt werden.

Sowohl das Konzept der Anwendungsadaptivität als auch die explizite Ressourcenreservierung weisen gewichtige Nachteile auf: Während ersteres die Anpassung aller Anwendungen verlangt und noch dazu keinerlei Garantien zu bieten vermag, ist die explizite Ressourcenreservierung vom Konzept her in der Lage, harte Garantien zu bieten. Allerdings gehen damit gewichtige Nachteile einher: Es handelt sich um ein sehr komplexes Rahmenwerk, das nur mit hohem Aufwand weitgehend hardwareunabhängig zu gestalten ist. Insbesondere die Abbildung von QoS-Parametern der Anwendungsschicht auf tieferliegende Schichten korreliert in starkem Maße mit der Ausstattung des jeweiligen Endsystems.

2.2 Rückkopplungsmechanismen in Endsystemen

Um die beschriebenen Nachteile traditioneller QoS-Management-Rahmenwerke zu umgehen, schlagen vereinzelte neuere Ansätze einfache Rückkopplungsmechanismen und einen Satz von adaptiven Ressourcenverwaltern vor ([ChSe96], [LaYa97], [DiSe97]). Die Grundidee dabei ist, mit Rückkopplungsmechanismen innerhalb des Betriebssystems eine stabile Dienstgüte ohne detailliertes Wissen um die gegenwärtige Auslastung oder die zugrundeliegende Hardware anbieten zu können. Jede Anwendung, die Dienstgüteunterstützung anfordert, vergleicht ihre Anforderungen mit der tatsächlich gebotenen Dienstgüte und generiert aus der Differenz eine Regelgröße. Anhand dieser Größe ändert das Betriebssystem die Ressourcenzuteilung für den Anwendungsprozeß zu Lasten anderer Prozesse. Dieser Ansatz erfordert weder exaktes Vorwissen der Anwendung über benötigte Ressourcen und Systemauslastung noch sind umfangreiche Verhandlungen über die Dienstgüte nötig. Damit können die Nachteile eines komplexen Rahmenwerkes vermieden werden, ohne umfangreiche Änderungen an den Anwendungen vornehmen zu müssen. Allerdings können natürlich keine sogenannten harten Garantien erwartet werden. In einer Überlastsituation des Systems erfolgt eine Verschlechterung der Dienstgüte; dies ist unvermeidbar in Systemen ohne harte Ressourcenreservierung. Für eine große Zahl von Anwendungen ist jedoch die Garantie von Dienstgüte unter dem Vorbehalt, daß der Endanwender nicht z.B. gleichzeitig eine weitere ressourcenintensive Anwendung startet, sicher ausreichend. Dies gilt insbesondere dann, wenn der Benutzer die Auswirkungen beispielsweise in einem einfachen graphischen Display selber verfolgen kann. Auch die Koexistenz von harten Reservierungen und derartigen Rückkopplungsmechanismen ist denkbar, soll hier jedoch nicht betrachtet werden. Statt dessen soll die Grundidee weiter ausgebreitet und angepaßt werden.

2.3 Ausdifferenzierte Rückkopplungsmechanismen

Bei genauerer Betrachtung der Systemarchitektur eines Endsystems wird deutlich, daß es nicht möglich ist, eine übergreifende Rückkopplungsschleife aufzustellen, wie das verschiedene Arbeiten anzunehmen scheinen ([ChSe96], [Beat97]). Denn während der

Scheduler und der Verkehrsformer des Netzwerkadapters wie in Kapitel 3 beschrieben in zeitlichen Größenordnungen von Nanosekunden bis Mikrosekunden arbeiten, liegen die Zeitanforderungen des CPU-Schedulers eher im Millisekundenbereich. Interaktion mit dem Benutzer kann, wenn er nicht überfordert werden soll, lediglich im Bereich von mehreren Sekunden bis Minuten erfolgen. Daher identifizieren wir in der Darstellung der Architektur eines Endsystems in Abbildung 1 drei verschiedene Schichten, die sich durch die jeweiligen zeitlichen Größenordnungen, in denen ihre Änderungen typischerweise verlaufen, unterscheiden.

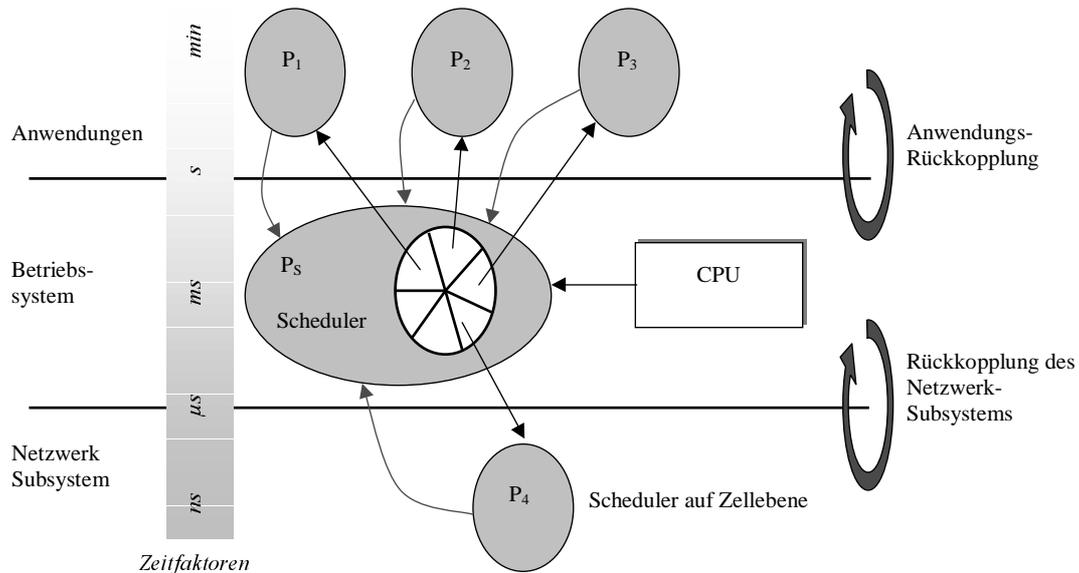


Abbildung 1: Prozesse und Rückkopplungsmechanismen im Endsystem

Die drei identifizierten Schichten sind die Anwendungsschicht, die Schicht des Betriebssystems und die Schicht des Netzwerk-Subsystems. Die Größenordnungen der typischen Zeitfaktoren sind in der senkrechten Achse angegeben. Die vorgeschlagene Architektur entkoppelt also kurzfristige Ereignismeldungen des Schedulers auf Ebene des Netzwerk-Subsystems von längerfristigen Änderungen auf Anwendungsebene. Die Verwaltung der Rechenzeit der CPU als einer zentralen Ressource in jedem Endsystem wird durch den Scheduling-Prozess im Betriebssystem vorgenommen, der sich damit an der Schnittstelle zwischen beiden Rückkopplungsschleifen befindet. Während in Kapitel 3 auf die unterste der drei Ebenen eingegangen wird und Mechanismen zum Erbringen der Dienstgüte im Netzwerk-Subsystem vorgestellt werden, soll hier kurz auf Einzelheiten der beiden Rückkopplungsschleifen eingegangen werden. Dabei wird deutlich, daß die Grundidee sicher noch weitere Verfeinerungen in der Konfrontation mit Problemen der praktischen Umsetzung erfahren muß.

2.3.1 Rückkopplungsschleife im Netzwerk-Subsystem

Der Scheduler des Netzwerk-Subsystems bestimmt, welches Datenpaket oder welche Zelle gesendet wird. Ein Software-basierter Ansatz, wie er im folgenden vorgestellt wird, benötigt dazu eine gewisse Zahl von CPU-Zyklen zur Durchführung der notwendigen Berechnungen. Erhält der Scheduler nicht ausreichend CPU-Zeit, werden Zellen verzögert. Aus dieser Verzögerung kann ein Rückkopplungssignal (Regeldifferenz) gewonnen werden, das über ein Regelglied auf den CPU-Scheduler einwirkt. Der CPU-Scheduler

kann dann dem Scheduler des Netzwerk-Subsystems mehr Prozessorzeit zuteilen und ihm damit ermöglichen, seine Aufgaben zu erfüllen. Diese Rückkopplung kann sehr rasch geschehen und vor allem nahezu transparent für den Anwender; zusätzlicher Aufwand durch exakte Erfassung der gegenwärtigen Systemauslastung kann vermieden werden.

2.3.2 Rückkopplungsschleife auf Anwendungsebene

Die Idee ist nun, analog zur eben geschilderten Rückkopplung einen Rückkopplungsmechanismus zwischen Anwendungen und CPU-Scheduler einzusetzen. Um den Anwender nicht zu überfordern, sollte er lediglich einen generischen Grad an Dienstgüteunterstützung (*QoS-level*) auswählen können, beispielsweise indem er einen entsprechenden Schieberegler für jede relevante Anwendung zur Verfügung gestellt bekommt. Diese Größe, die Führungsgröße oder Sollgröße im Sinne der Regelungstechnik, wird verglichen mit der aktuellen Dienstgüte, die der Anwendung vom zugrundeliegenden System geboten wird (Istgröße oder Meßwert). Wenn eine Differenz besteht, beispielsweise eine Dienstgüteverschlechterung, wird mehr Rechenzeit für die jeweilige Anwendung vom CPU-Scheduler angefordert. Es ist offensichtlich, daß diese Anforderung anderen Prozessen Rechenzeit wegnehmen muß. Das heißt, lediglich die Verteilung der Ressourcen ändert sich. Endliche Ressourcen vorausgesetzt ist jedoch nichts anderes möglich.

Diese Rückkopplungsmechanismen arbeiten in zeitlichen Größenordnungen von Sekunden und können immer noch ohne direkte Interaktion des Benutzers erfolgen. Wenn länger anhaltende Dienstgüteverschlechterungen beobachtet werden, kann der Anwender jedoch eingreifen, indem er die Verteilung der Systemunterstützung für die Anwendungen neu ausbalanciert. Dies kann über die erwähnten Schieberegler passieren, aber auch einfach durch Verschieben des Fokus auf ein anderes Fenster in einer graphischen Oberfläche.

3 Software-basiertes Scheduling für Netzwerkadapter

Während im vorigen Abschnitt eine generelle Architektur für Kommunikationssysteme mit Rückkopplungsmechanismen vorgestellt wurde, soll nun detaillierter auf Mechanismen des Netzwerkadapters eingegangen werden, der in der untersten Schicht in Abbildung 1 einzuordnen ist. Der Netzwerkadapter und der dazugehörige Prozeß des Schedulers für Zellen bzw. Datenpakete spielt bei der Betrachtung eine ausgezeichnete Rolle, da er von jeher als ein Flaschenhals bezüglich der Einhaltung von Dienstgüte betrachtet wurde.

Zur generellen Erläuterung der Aufgaben von Netzwerkadaptoren sollen im folgenden einige damit verbundene Funktionen, Begriffe und Probleme in Erinnerung gerufen werden. Für aktuelle ATM-basierte Netze und auch die zukünftige Internet-Generation ist eine Unterstützung von Dienstgüten in Netzwerkadaptoren elementar. Während heutige Adapter im wesentlichen dafür sorgen, daß sendebereite Datenpakete über ein Netz übertragen werden (und damit einen sehr stoßartigen Verkehr erzeugen), müssen fortgeschrittene Adapter zusätzlich dafür sorgen, daß der Strom von Zellen bzw. Teilen von Paketen eine bestimmte Form (*shape*) aufweist [RiSc97]. Im folgenden wird die Terminologie von ATM-Netzen übernommen, ähnliches kann auf andere paketorientierte Netze übertragen werden.

Um das Innere eines Netzes vor fehlerhaften Datenquellen oder bewußten Attacken zu schützen, wird in ATM eine sog. UPC (Usage Parameter Control) eingeführt. Diese überwacht den eingehenden Zellstrom und prüft für jede Verbindung, ob der zeitliche Abstand zwischen je zwei Zellen einem vertraglich festgelegten Mindestabstand entspricht. Hierbei werden dann Parameter wie maximale Zellrate (PCR, peak cell rate), mittlere Zellrate (SCR, sustainable cell rate) oder auch eine Varianz der Verzögerung (CDV, cell delay variation) beim Verbindungsaufbau festgelegt. Das Einhalten dieser Parameter kann von der UPC sehr genau überwacht werden, Hardware-Lösungen für 16000 Verbindungen mit je 622 Mbit/s existieren.

Traditionelle Netzwerkadapter versuchen vor allem eine hohe Geschwindigkeit bei der Datenübertragung von einer Anwendung auf das Netz zu erzielen. Heute ist nicht nur ein hoher Durchsatz, sondern vor allem auch die Unterstützung einer Vielzahl von Dienstgüteparametern gefragt. Die Mechanismen der Verkehrsformung auf Seiten eines Endsystems sind unabdingbar, sollen nicht die einzelnen Zellen beim Netzzugang durch die UPC verworfen werden. Lösungen in dieser Richtung sind derzeit vor allem auf dedizierte Hardware konzentriert. Beim Vergleich unseres Ansatzes mit den vorhandenen wollen wir vor allem auf die folgenden drei Punkte eingehen:

- *Effizienz*: Wie nutzen die Mechanismen des Scheduling (Auswahl einer Zelle zum Senden) und Shaping (Formung der Verkehrscharakteristik) die zur Verfügung stehende Bandbreite, um einen möglichst hohen Gewinn durch das Multiplexen zu erzielen?
- *Isolation*: Können die einzelnen Verbindungen gegenseitig abgeschirmt werden oder beeinflussen sie sich gegenseitig, so daß evtl. eine kurzzeitige Überlast verursacht durch eine Verbindung auch die Zellen einer anderen außerhalb des Verkehrsvertrags bringt?
- *Skalierbarkeit*: Wie viele Verbindungen können individuell gleichzeitig unterstützt werden?

3.1 Existierende Ansätze

Existierende Hardware-Lösungen sind meist hinsichtlich der Anzahl unterstützter Verbindungen mit unterschiedlichen Parametersätzen stark beschränkt. Verbindungen werden bestimmten Warteschlangen zugeordnet, die mittlere erzeugte Zellrate wird von der maximalen als Bruchteil abgeleitet etc. Dies beruht im allgemeinen darauf, daß Verkehrsparameter direkt auf Hardware-Register abgebildet werden, die Anzahl der Register jedoch aufgrund des beschränkten Platzes auf einem Chip sehr begrenzt ist. Daher resultiert die schlechte *Skalierbarkeit*. Die *Effizienz* wird weiterhin durch zu kleine Puffer eingeschränkt, Zellen werden erst kurz vor dem Senden aus dem Hauptspeicher geholt, optimale Übertragungsgrößen über interne Systembusse nicht richtig ausgenutzt oder auch das Endsystem unnötig beim Eingang von Zellen unterbrochen. Schließlich ist meist keine *Isolation* der unterschiedlichen Verbindungen gewährleistet, typischerweise teilen sich viele Verbindungen eine Warteschlange pro Prioritätsstufe, Überlastungen dieser Warteschlange wirken sich dann auf alle Verbindungen der Schlange aus.

Zusammenfassend läßt sich feststellen, daß existierende Lösungen hinsichtlich der Fähigkeit Verkehr zu formen und eine Vielzahl unabhängiger Verbindungen zu unterstützen, stark eingeschränkt sind. Die UPC-Chips auf Seiten der Netzbetreiber hingegen können sehr detailliert den Verkehr kontrollieren, das Ergebnis sind hohe Zellverlust-raten obwohl die Verkehrsparameter beim Verbindungsaufbau ausgehandelt wurden. Diese Problematik ist natürlich in der begrenzten Hardware zu sehen, kann aber nur teilweise durch leistungsfähigere Bausteine oder größere Puffer ausgeglichen werden. Solche Lösungen wären dann nämlich wesentlich teurer und für den breiten Markt nicht mehr akzeptabel. Für einfache, billige Netzwerkadapter mit QoS-Unterstützung kommt nur eine Software-basierte Lösung in Frage, die rechenintensive, standardisierte Aufgaben (Segmentierung, Reassemblierung, Prüfsummenberechnung, Vorwärtsfehlerkorrektur etc.) auf dedizierte Hardware auslagert.

3.2 Elemente eines Software-basierten Netzwerkadapters

Eine CPU muß für den Netzwerkadapter folgende Prozesse abarbeiten: ein Scheduler auf Zellebene, ein Shaper für die Verkehrsströme und eine Datentransfereinheit. Welche CPU dies macht ist von sekundärem Interesse, es könnte die Haupt-CPU eines Endsystems oder eine separate CPU auf dem Adapter sein. Dies ist kein grundlegender Unterschied sondern lediglich eine Frage der Leistungsfähigkeit der CPUs und der internen Verbindungsstrukturen. Eine Neuheit des Entwurfs ist die Art der Kooperation zwischen Scheduler, Shaper und DMA-Einheit, die sicherstellt, daß die richtige Zelle zum richtigen Zeitpunkt gemäß des Verkehrsvertrages und der aktuellen Last auf dem Adapter an das Netz gesendet wird. Zusätzlich wird durch eine „just-in-time“-Übertragung von einer auf die optimale Übertragungsgröße von Daten im System (*burst-size*) abgestimmten Datenmenge per DMA für eine geringe Systemlast gesorgt.

Die Interaktion der 3 Elemente der Software-Architektur wird in Abbildung 2 aufgezeigt. Der Shaper speichert die Kontextinformationen für alle existierenden Verbindungen, u.a. PCR, SCR, CDV Toleranz, bereits gesendete Daten etc. Er versorgt auch den Scheduler mit einer Zeitangabe, zu der frühestens die nächste Zelle einer aktuellen Verbindung gesendet werden darf ohne den Verkehrsvertrag zu verletzen. Daraufhin aktualisiert der Scheduler eine nach Zeitpunkten geordnete Liste von Verbindungskennungen und übergibt dem Shaper wiederum einen Zeiger auf Kontextinformationen für die nächste Verbindung, von der eine Zelle auf das Netz gesendet werden muß. Da der Shaper zusätzlich Informationen über die aktuell für eine Verbindung gespeicherten Zellen auf dem Adapter und die Anzahl der wartenden PDUs (Protokolldateneinheiten) besitzt, kann vorausberechnet werden, wann der nächste Datentransfer vom Hauptspeicher zum Adapter stattfinden muß. Diese Information wird dann an die DMA-Einheit weitergeleitet.

Die Implementierung des Shapers verwendet eine Kombination des Virtual Scheduling Algorithmus (VSA) und des Leaky Bucket Algorithmus (LBA). Beide wurden von den entsprechenden Spezifikationen für einen möglichen Algorithmus für die UPC, nämlich dem GCRA (Generic Cell Rate Algorithm), für den Netzzugang zu einem öffentlichen ATM-Netz abgeleitet [Foru96]. Anstatt diese Algorithmen für die Überwachung von Zellströmen am Netzzugang einzusetzen, verwenden wir sie, um damit aktiv den Verkehr zu formen. Werden zusätzlich die Parameter für die Toleranz der Verzögerungs-

schwankung (CDVT, Cell Delay Variation Tolerance) richtig gesetzt, so garantiert unser Vorgehen, daß alle Zellen am Netzzugang akzeptiert werden.

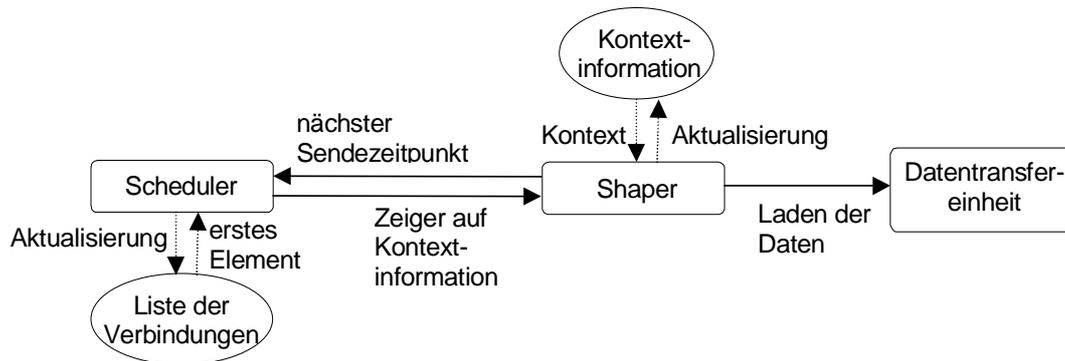


Abbildung 2: Interaktion von Shaper, Scheduler und Datentransfereinheit

3.3 Implementierung und Leistungsmessung

Der Algorithmus wurde in C auf verschiedenen Plattformen implementiert (Pentium-Pro, Alpha, UltraSPARC, [BGKK⁺97]). Das Hauptinteresse der Messungen lag hierbei in der Anzahl der CPU-Zyklen, die für das Senden einer Zelle benötigt wird. Für jede Zelle muß ein Zeitpunkt für das Senden berechnet, die Liste mit aktiven Verbindungen aktualisiert, eventuell ein DMA-Transfer initiiert und schließlich ein ATM-Chip zum Senden aktiviert werden. Diese Aufgaben müssen für einen 155 Mbit/s-Adapter innerhalb von $2,8 \mu s$, für einen 622 Mbit/s-Adapter innerhalb von $680 ns$ erledigt werden.

Die hier vorgestellten Messungen wurden unter Zuhilfenahme spezieller Register auf den CPUs, z.B. dem Time Stamp Counter des PentiumPro, durchgeführt. So kann ohne jegliche Beeinflussung des Algorithmus die Anzahl der benötigten CPU-Zyklen, der Speicherzugriffe etc. bestimmt werden. Die Implementierung wurde mit sog. worst-case-Szenarios belastet, um die Reaktion hinsichtlich der oben aufgeführten Kriterien Effizienz, Isolation und Skalierbarkeit zu untersuchen. Für solche Szenarios wurden in dem vorgestellten Beispiel über 1000 Verbindungen gleichzeitig gestartet, die alle die gleichen Verkehrscharakteristika hatten. Dadurch ist für das Senden einer Zelle jedesmal der maximale Aufwand erforderlich. Das Beispiel in Abbildung 3 zeigt dennoch deutlich, daß bereits mit heutigen Standardprozessoren die gestellte Aufgabe für 155 Mbit/s-Adapter erfüllbar ist: im Schnitt werden unter 330 CPU-Zyklen für das Senden einer Zelle benötigt, bei einer Taktfrequenz von 200 MHz und damit einer Zykluszeit von $5 ns$ entspricht dies deutlich weniger als die geforderten $2,8 \mu s$.

Hierbei ist zu beachten, daß alle Verbindungen individuell behandelt wurden, eine vollständige Isolation ist somit gewährleistet. Zusätzlich gewinnt die vorgestellte Implementierung bei jeder neuen Generation von Prozessoren im Gegensatz zu reinen Hardware-Lösungen. Der Algorithmus ist unabhängig von der Zahl der aktiven Verbindungen, Messungen mit 64000 Verbindungen zeigen eine mittlere Zyklenzahl von 620. Wesentlich für die Leistungsfähigkeit ist, daß möglichst die gesamte Verbindungsliste im Cache des Prozessors liegt. Pro Verbindung werden hier jedoch lediglich 64 bit benötigt, so daß eine entsprechend große Anzahl an Verbindungsinformationen in einem Cache z.B. des PentiumPro mit 512 kByte gehalten werden kann.

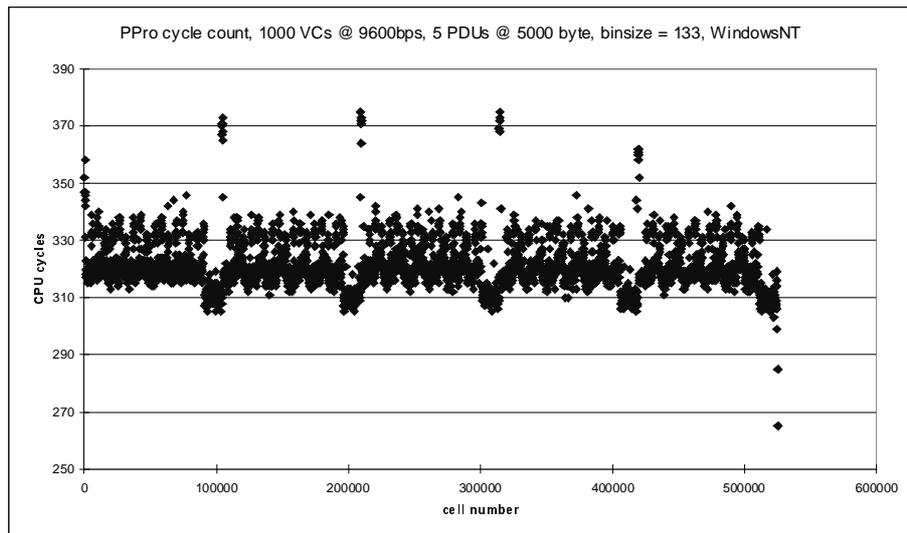


Abbildung 3: Benötigte CPU-Zyklen für das Senden von ATM-Zellen

4 Zukünftige Schritte

Der im vorigen Abschnitt vorgestellte Software-basierte Netzwerkadapter stellt nur eine Komponente innerhalb des Gesamtsystems dar. Wie bereits im zweiten Abschnitt aufgezeigt, müssen weitere Schritte folgen:

- *Betriebssystemunterstützung:* Für die Unterstützung von Dienstqualitäten und die Integration von Rückkopplungsmechanismen müssen existierende Betriebssysteme erweitert werden. Diese Erweiterung betrifft vorrangig den Scheduler eines Betriebssystems. Der grundlegende Scheduler muß echtzeitfähig sein, dann können innerhalb einzelner Zeitscheiben weitere Scheduler nach verschiedenen Gesichtspunkten integriert werden (Hierarchisches Scheduling). Zielplattformen sind hier Linux und WindowsNT. Wichtig ist dabei aber, daß alle bisherigen Anwendungen weiterhin ungehindert unterstützt werden, lediglich die neuen Anwendungen, die eine bestimmte Dienstgüte für die Kommunikation erfordern, sollen von den erweiterten Möglichkeiten profitieren.
- *Adaptive Anwendungen:* Bereits heute sind Multimediaanwendungen zu komplex, um sie für normale Anwender einzusetzen. Man kann nicht von einem Benutzer verlangen, daß er unterliegende Mechanismen erkennen und entsprechend darauf in einer „technischen“ Form reagieren kann. Beispielsweise sind Benutzer überfordert, wenn sie bei Problemen mit der Datenrate mit Verringern der Farbtiefe oder Auflösung reagieren sollen. Hier müssen sich Anwendungen automatisch anpassen, ein Benutzer soll nur noch durch z.B. Klicken auf ein Fenster seine Präferenzen für die zugehörige Anwendung ausdrücken.
- *Messungen in realen Netzen:* Die bisher implementierten Algorithmen müssen auf reale ATM-Systeme gebracht und vermessen werden. Zusätzlich wird das Verfahren auch auf Internet-basierte Netze angewandt, um auch hier Erfahrungen mit Software-basiertem Scheduling von Paketen variabler Länge zu sammeln.
- *Untersuchung der Rückkopplungsmechanismen:* Auf eher theoretischem Gebiet stehen diese Untersuchungen, die außerordentlich wichtig sind, um unerwünschte

Schwingungen des Systems zu vermeiden. Aktuelle Untersuchungen im Internet haben bereits gezeigt, daß auch in bestehenden Netzen Schwingungen vorhanden sind, mit der Einführung weiterer Rückkopplungen soll die Situation auf keinen Fall verschlechtert werden.

Literatur

- [AuCH98] C. Aurrecochea, A. Campbell und L. Hauw. A Survey of QoS Architectures. *Multimedia Systems Journal, Special Issue on QoS Architecture*, Mai 1998.
- [Beat97] G. Beaton. A Feedback-Based Quality of Service Management Scheme. *Proceedings of Third International Workshop on High Performance Protocol Architectures*, Juni 1997.
- [BGKK⁺97] H. Bryhni, P. Gunningberg, E. Klovning, Ø. Kure und J. Schiller. DMA Scheduling for High Capacity Data Transfer and Traffic Shaping. Technischer Bericht, Telenor Research and Development, Juni 1997.
- [Camp94] A. Campbell. A Quality of Service Architecture. *ACM SIGCOMM Computer Communication Review*, April 1994.
- [ChSe96] H. Cho und A. Seneviratne. Dynamic QoS Control without the Knowledge of Resource Requirements. *submitted to publish*, September 1996.
- [CoTe94] D. Compton und D. Tennenhouse. Collaborative Load Scheduling for Media-Based Applications. *Proceedings of the International Conference on Multimedia Computing and Systems*, Mai 1994.
- [DiSe97] C. Diot und A. Seneviratne. Quality of Service in Heterogeneous Distributed Systems. *Proceedings of HICSS'97, 30. Hawai'ian International Conference on System Sciences*, Januar 1997.
- [Foru96] The ATM Forum. Traffic Management Specification version 4.0. af-tm-0056.000, April 1996.
- [JoBF96] M. Jones, J. Barrera und A. Forin. An Overview of the Rialto Real-Time Architecture. *Proceedings of the 7th ACM SIGOPS European Workshop*, September 1996.
- [LaYa97] K. Lakshman und R. Yavatkar. Integrated CPU and Network-I/O QoS Management in an Endsystem. *Proceedings of fifth IFIP International Workshop on Quality of Service (IWQOS 97)*, Mai 1997.
- [NaSt95] K. Nahrstedt und R. Steinmetz. Resource Management in Networked Multimedia Systems. *IEEE Computer*, Mai 1995.
- [RiSc97] H. Ritter und J. Schiller. Software-based ATM cell-level scheduling. *Proceedings of 6th Open Workshop on High Speed Networks*, Oktober 1997.

Automatisierung der WWW-Programmierung

Albrecht Schmidt

Kurzfassung

Im folgenden Beitrag wird ein Ansatz zur Automatisierung der WWW-Programmierung vorgestellt. Ausgehend von einer Analyse von bestehenden Webserviceangeboten werden verschiedene Aspekte der Automatisierung diskutiert. Mit diesem Ansatz soll das Erstellen interaktiver Web-Applikationen ohne Programmierkenntnisse möglich sein. Im weiteren wird ein Prototyp - der MetaWebService, der in Zusammenarbeit mit dem HP-IT Service Center entwickelt wurde, vorgestellt. Dieser Prototyp ermöglicht es ohne Programmierkenntnisse interaktive Webangebote, wie Bestellannahmen oder Konferenzzanmeldungen, zu erstellen. Der MetaWebService ist ebenfalls als Web-Applikation implementiert und ermöglicht es so, daß neue Webserviceangebote von beliebigen Webbrowsern aus erstellt werden können.

1 Einleitung

Das Internet und insbesondere das World Wide Web (WWW) bietet viele neue Möglichkeiten, um eine effiziente computergestützte Zusammenarbeit zu ermöglichen.

Besonders die Formen der asynchronen Zusammenarbeit auf der Basis des Internets werden heute in vielen Unternehmen erfolgreich eingesetzt. Zwei wichtige Gruppen sind hierbei Dienste, welche über Email realisiert werden, und Dienste, welche mittels eines WWW-Frontends angeboten werden.

Ein typisches Beispiel für eine Kombination aus beiden Formen ist ein elektronischer Antrag. Die Datenerhebung wird mittels eines HTML-Formulars im Web durchgeführt. Abhängig von den Daten wird dann der Antrag an die bearbeitende Abteilung via Email weitergeleitet. Hierbei ist auch die Realisierung einfacher Workflows in der Praxis durchaus gängig.

Ein einfaches Beispiel für das oben beschriebene Szenario ist ein Antrag für ein Rechner-Account, der firmenintern behandelt wird. Der Benutzer gibt über einen Webbrowser seine Daten in das entsprechende Formular ein. Aus diesen Daten werden dann Emails an die EDV-Abteilung, welche den Antrag bearbeitet, an den Vorgesetzten, welcher sein Veto gegen den Antrag einlegen kann, und an den Benutzer selbst, erzeugt und versandt.

Die Erstellung solcher Dienstangebote besteht im allgemeinen aus zwei wesentlichen Teilen. Zuerst werden die Webseiten für die Datenerhebung gestaltet und die Emailtexte

entworfen, hierfür sind prinzipiell keine Programmierkenntnisse notwendig. Im zweiten Teil wird dann das eigentliche Programm - oft ein CGI-Skript - entwickelt. Dieser Schritt setzt Kenntnisse der Web-Programmierung voraus.

Die beschriebene Vorgehensweise impliziert, daß die Erstellung von neuen Webserviceangeboten, insbesondere der Teil der Programmierung, nur von EDV Fachpersonal durchgeführt werden kann. Im folgenden wird ein Ansatz zur Automatisierung der Programmierung vorgestellt. Der MetaWebService (MWS) hat das Ziel, die Erstellung von Standardwebangeboten soweit zu vereinfachen, daß diese vom eigentlichen Serviceanbieter selbst durchgeführt werden kann.

Ein Beispielszenario ist hierbei der Sachbearbeiter, welcher die Reisekosten bearbeitet. Er erstellt das Formular und die Rückemails. Diese werden dann an das System übergeben; hier wird nun automatisch der neue Dienst erstellt und aktiviert.

Um die Erstellung von interaktiven Webangeboten zu vereinfachen gibt verschiedene Ansätze im Bildungsbereich [Mahe96], [CCLO96]. Hierbei wird insbesondere die Erstellung interaktiver Lehrrangebote unterstützt. Eine weitere Möglichkeit, die Erstellung von Webanwendungen zu vereinfachen, bieten Schablonen [Joha97].

Die Anwendung von Methoden des Software Engineering im Kontext des WWW wird zunehmend zum Forschungsgegenstand. Beispiele hierfür sind in [GeWG97] und [Eich97] zu finden.

Im kommerziellen Bereich werden diverse Produkte in Form von grafischen Entwicklungsumgebungen vertrieben (z.B. [Prog97]). Diese Software ist darauf ausgerichtet, die Entwicklung großer Internetanwendungen zu unterstützen. Diese Produkte sind zur Erstellung von einfachen Serviceanwendungen durch *nicht-Programmierer* meist nicht geeignet. Ein weiteres Problem beim Einsatz solcher Werkzeuge stellt ein heterogenes Soft- und Hardwareumfeld dar.

2 Analyse bestehender Webserviceangebote

Um die Anforderungen an den MWS definieren zu können, ist es notwendig eine Analyse von Webserviceangeboten durchzuführen. Im allgemeinen bestehen Serviceangebote aus einem Frontend und einem Backend. Mit Hilfe des Frontend wird der Kunde informiert. Außerdem wird hier die Datenerhebung durchgeführt. Im Backend werden die erhobenen Daten verarbeitet und evtl. an andere Prozesse weitergegeben, siehe Abbildung 1.

In den Serviceangeboten werden im folgenden drei Teilbereiche unterschieden und diskutiert, das Frontend, das Backend sowie die Validierungsfunktion.

2.1 Analyse des Frontends

Anhand der folgenden Kriterien ist es einfacher, die Funktionalität des Frontends zu analysieren und zu verstehen.

1. Anzahl der Seiten für den Webservice. (Unterschieden nach reinen Informationsseiten und Formularseiten, sowie deren Abfolge.)

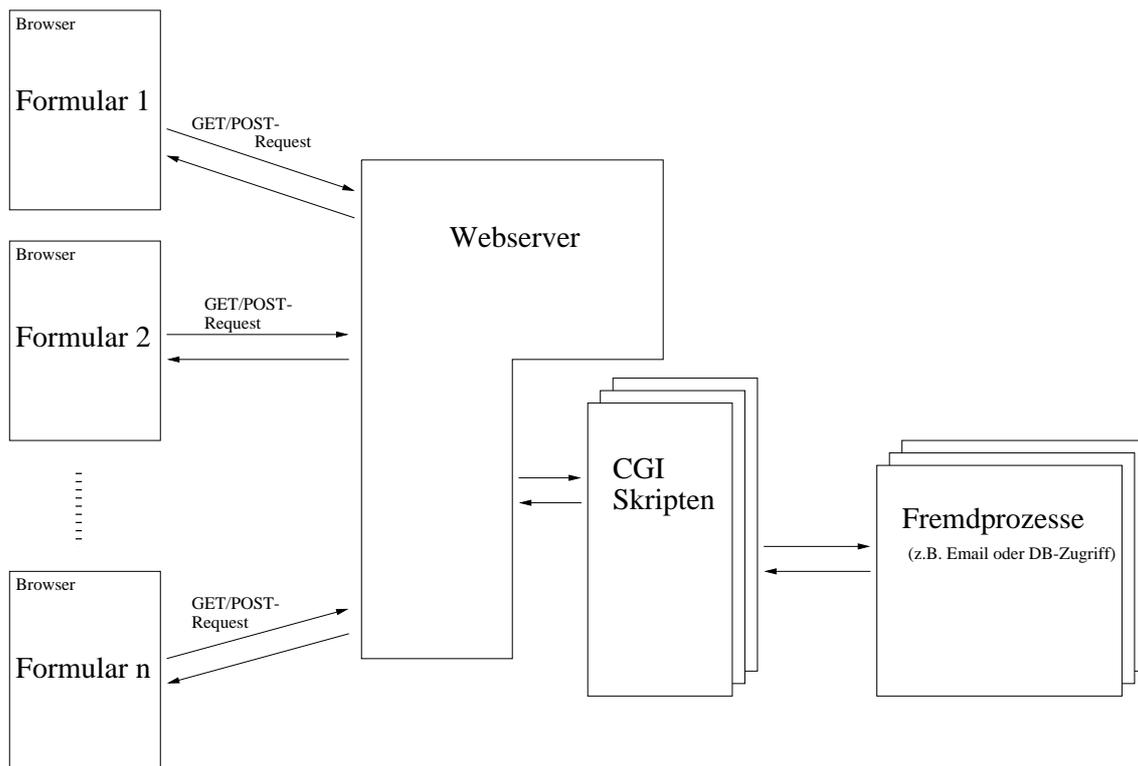


Abbildung 1: Allgemeiner Aufbau eines interaktiven Webserviceangebots.

2. Datenaustausch zwischen Seiten. (Wird eine erhobene Information auf einer folgenden Seite verwendet? Z.B. auf der ersten Seite wird der Name des Kunden abgefragt und auf allen folgenden Seiten wird der Kunde dann persönlich mit seinem Namen angesprochen.)
3. Datenaustausch zwischen Seiten und weiteren Prozessen. (Wird eine erhobene Information durch einen weiteren Prozeß bearbeitet und auf einer folgenden Seite verwendet? Z.B. auf der ersten Seite wird die Personalnummer abgefragt und auf der folgenden Seite wird dann das Formular schon mit den persönlichen Daten, die identifiziert durch die Personalnummer aus einer Datenbank gelesen wurden, ausgefüllt bereitgestellt.)
4. Ablaufsteuerung (Werden aufgrund der eingegebenen Daten verschiedene Seiten gezeigt? Z.B. gibt ein Kunde auf der ersten Seite seine Kundennummer an, erhält er als nächste Seite die Bestellseite, gibt er keine Kundennummer an, erhält er eine Seite zur Registrierung.)

Im einfachsten Fall beinhaltet das Frontend des Services nur eine Formularseite. Für diese Kategorie haben alle weiteren genannten Punkte (2.-4.) keine Relevanz. Dieser einfache Fall trifft für sehr viele Serviceangebote zu, welche in der Praxis benutzt werden. Als Beispiele hierfür sind Konferenzanmeldungen, Abonnieren von Mailinglisten, oder Registrieren von Software zu nennen.

Für Angebote mit mehreren Seiten wird oft die Funktionalität, die in den Punkten 2. und 3. beschrieben wurde, verwendet. Hierbei ist zu unterscheiden, daß für die Funktion, beschrieben 2., nur die Kenntnis des Konzepts einer Variablen beim Servicedesigner

vorhanden sein muß, wohingegen die Funktionalität, beschreiben in 3., auch elementare Programmierkenntnisse voraussetzt.

Webservices, die eine Ablaufsteuerung sowie die Interaktion zwischen Seiten beinhalten, sind vergleichsweise selten. Solche Angebote sind in der Regel WWW-Applikationen; ein Beispiel hierfür sind auf CGI-Basis erstellte Online Shops. Diese können ohne Programmierkenntnisse nicht erstellt werden. Aus diesem Grund wurden solche Angebote im Design des MWS nicht berücksichtigt. Komplexere Webservices können aber auch ohne diesen Mechanismus als eine Menge einfacher Services erstellt werden.

Auf Grund dieser Analyse wurde die Funktionalität, die in 1. und 2. beschrieben wurde, für das Design des MWS berücksichtigt, d.h. es können mit Hilfe des MWS Angebote mit einer beliebigen Anzahl von Seiten erstellt werden. Hierbei wurde vor allem berücksichtigt, daß mit dieser Funktionalität ein Großteil der Standard-Webservices erstellt werden kann, wobei keine Programmierkenntnisse vorausgesetzt werden müssen.

2.2 Analyse des Backends

Die CGI-Skripten haben eine sehr unterschiedliche Funktionalität. Bei genauerer Betrachtung läßt sich folgende Abstraktion finden. Skripten (Programme) werden durch Variablenwerte, welche durch die Formulareingabe erhoben wurden, parameterisiert. Die Erstellung solcher Skripten impliziert Programmierkenntnisse. Da diese Programme als Unikate entwickelt werden, sind sie der Hauptkostenfaktor.

Bei genauerer Betrachtung können innerhalb der Skripten zwei funktionale Einheiten unterschieden werden. Ein Teil beschäftigt sich mit der Extraktion der Variablen aus den gegebenen Formularinformationen, der andere Teil ist das eigentliche Programm (z.B. Verschicken einer Mail, SQL-Statement, Eintrag in eine Bestelldatei). Diese funktionalen Blöcke werden im allgemeinen nicht getrennt, d.h. in den Skripten ist keine klare Trennung zu erkennen. Diese Verschmelzung erschwert die Wartung und Erweiterung vorhandener Angebote.

Bei weiterer Analyse zeigt sich, daß das Sammeln der Informationen aus den verschiedenen Formularen und die Extraktion der Variablenwerte ein großes Automatisierungspotential bieten. Hingegen ist die Erstellung der eigentlichen Programme in allgemeiner Weise nicht zu automatisieren, da sie direkt auf einen Service abgestimmt ist.

Ausgehend von diesen Erfahrungen wird im MWS das Skript zum Sammeln von Informationen und zur Extraktion von Variablen automatisch erstellt. Die gesammelten Informationen werden dann verwendet um eine Rückantwort zu generieren. Diese wird dann per Email oder Fax an eine definierte Adresse geschickt. Die Ausführung von parameterisierten Programmen wird ebenfalls durch einen auf Email basierten Ansatz gelöst. Dieses Verfahren wird in Abschnitt 3.3 beschrieben.

2.3 Analyse der Validierungsfunktion

Viele der betrachteten Webservices beinhalten Validierungsfunktionen für einzelne Eingabefelder. Grundsätzlich wird hierbei geprüft, ob die Eingabe einem bestimmten Muster entspricht. Einige Beispiele hierfür sind die Prüfung, ob eine Emailadresse ein

@-Symbol enthält, ob eine Postleitzahl fünf Stellen hat oder eine Zahl in einem bestimmten Intervall ist.

Diese Form der Validierung kann auf Formularseite (im Frontend) oder auf Serverseite (im Backend) realisiert sein.

Im Formular wird dies durch ein Skript (z.B. JavaScript) implementiert. Dieses Skript überprüft die Werte bevor sie an den Server geschickt werden und produziert evtl. Warnhinweise oder Fehlermeldungen und gibt dem Benutzer die Möglichkeit seine Eingabe zu korrigieren. Manche HTML-Editoren (z.B. MS-Frontpage) unterstützen die Erstellung solcher Eingabevalidierungsfunktionen.

Auf Serverseite können die eingegangenen Variablen ebenfalls auf ihre Zulässigkeit geprüft werden. Dies geschieht nach dem Sammeln der Variablen und vor der weiteren Verarbeitung.

Beide Ansätze bieten Vor- und Nachteile. Eine Prüfung im Formular ist im allgemeinen schneller, da sie lokal abläuft. Der Nachteil hierbei ist, daß die verwendete Skriptsprache vom Browser unterstützt werden muß. In der Praxis zeigt sich jedoch, daß oft durch die verwendeten Sicherheitskonzepte diese Sprachen nicht zur Verfügung stehen. Um trotzdem sicherzustellen, daß die Variablenwerte gültig sind, muß also eine serverseitige Prüfung zusätzlich durchgeführt werden.

Aus diesen Ergebnissen wurde beim Design des MWS die Validierungsfunktion auf Serverseite implementiert. Dem Benutzer, welcher die Webformulare erstellt, ist aber freigestellt klientenseitig Skripten zur Validierung zu integrieren.

3 Das MWS-System

Unter Berücksichtigung der Ergebnisse der obigen Analyse sowie beeinflusst durch die gegebenen Randbedingungen wurde der MWS entworfen und implementiert.

Wesentliche Randbedingungen, die beim Entwurf berücksichtigt wurden, sind:

- Neue Standardwebservices sollen ohne Programmierkenntnisse erstellt werden können.
- Die Erstellung soll weitestgehend automatisiert werden.
- Der Einsatz von Standardsoftware zur Erstellung von Formularen und Rückantworten soll möglich sein.
- Neue Services sollen von jedem Ort mit Webzugang erstellt werden können.
- Die Erstellung soll Plattformunabhängig möglich sein.
- Die erstellten Webservices sollen weitestgehend webserver- und plattformunabhängig eingesetzt werden können.

Die Berücksichtigung dieser Anforderungen soll sicher stellen, daß der entworfene MWS die Effizienz der Erstellung neuer Webangebote steigert. Insbesondere soll die Erstellung von Webservices aus der EDV-Abteilung in die eigentlichen Servicebereiche verlagert werden.

Dies birgt ebenfalls ein besonders großes Potential für Mittlere und Kleine Unternehmen, die bis jetzt auf interaktive Webangebote verzichtet haben, da sie nicht das erforderliche EDV-Know-how besaßen.

3.1 Designentscheidungen und Architektur

Es wird die Erstellung von Webservices mit beliebig vielen Webseiten unterstützt. Dabei kann in der $(n + 1)$. Webseite auf alle Variablen, der Seiten von eins bis n zugegriffen werden (Punkte 1. und 2. aus Abschnitt 2.1). Die Entscheidung, nur diese Funktionalität zu unterstützen, ist darin begründet, daß somit Benutzer ohne Programmierkenntnisse neue Services erstellen können. Als Rückantwort können beliebige Texte (somit auch Programme/Skripten) verwendet werden. Die in den Texten verwendeten Variablen müssen auf den Formularseiten definiert sein. Die Adressen für die Rückantworten können entweder über in den Formularen definierte Variablen oder fixe Zuordnungen festgelegt werden. Die HTML-Seiten und Rückantworten können mit beliebigen Editoren erstellt werden. Lediglich die Variablen müssen in einer vereinbarten Weise markiert werden, siehe Abschnitt 3.2.

Das System zum einfachen Erstellen neuer Webservices ist ebenfalls als Web-Applikation entworfen worden. Die Benutzerschnittstelle basiert auf reinem HTML. Dadurch können von jedem beliebigen Ort mit Webzugang neue Webservices eingerichtet werden. Der Zugang setzt lediglich einen HTML-fähigen Browser voraus. Der serverseitige Teil des Programms führt eine Analyse der gegebenen Formulare, Informationseiten und Rückantworten durch. Interaktiv wird dann der Benutzer nach weiteren Information, wie z.B. Muster für die Feldvalidierung, gefragt. Unter Verwendung dieser Daten wird dann automatisch der Code für den neuen Webservice generiert.

3.2 Implementierung

Die implementierte Benutzerschnittstelle verwendet HTML Eingabefelder um, die Daten vom Benutzer zu erfragen. Die vom Benutzer erstellten Dateien (Formulare, HTML-Seiten und Rückantworten) werden mit dem in RFC 1867 [NeMa95] spezifizierten *Form-based File Upload in HTML* direkt aus der Webseite mit HTTP-POST an den Server übertragen. Zu jeder Rückantwort wird eine Adresse angegeben. Die Adresse ist entweder eine gültige Emailadresse oder eine Variable der Form `%VariablenName%`. Durch die Benutzung eines Fax-Gateways können Rückantworten auch als Fax ausgeliefert werden.

Für die Formulare gelten folgende Konventionen:

- Jedes Formular ist eine gültige HTML-Seite.
- Jedes Formular enthält genau ein `<FORM>` Tag, das alle Eingabefelder umfaßt.
- Die Formulare enthalten keine Schaltflächen (Buttons), sondern die Variable `%BUTTON%`, die dann durch eine Knopfleiste ersetzt werden.
- Auf bereits definierte Variablen kann mit `<%VariablenName%>` zugegriffen werden.

Für Rückantworten gelten folgende Konventionen:

- Jede Rückantwort ist eine gültige HTML-Seite.
- Auf definierte Variablen kann mit `%VariablenName%` zugegriffen werden.

In einem weiteren Schritt werden mit einem CGI-Skript alle in Formularen definierten Variablen gesucht. Diese werden dann mit der Menge der auszugebenden Variablen in den HTML-Seiten sowie in den Rückantworten verglichen. Sind alle verwendeten Variablen definiert, wird der Vorgang fortgesetzt.

Im nächsten Schritt werden interaktiv die Validitätsbedingungen für die einzelnen Variablen abgefragt. Hierbei wird dem Benutzer die Möglichkeit gegeben, bestimmte Muster vorzugeben, welche die Gültigkeit der Variablen im entstehenden Service definieren.

Nachdem alle Daten erhoben wurden, wird von einem weiteren CGI-Skript der neue Service erzeugt. Dabei werden die einzelnen HTML-Seiten angepaßt, der Code für die CGI-Skripten generiert, und die Rückantworten aufbereitet. Als Ergebnis entstehen eine HTML-Eingangsseite für den Service, mehrere Resourcedateien, die aus den HTML-Dateien und Rückantwortdateien gewonnen wurden, sowie eine Menge von Perl-CGI-Skripten. Diese Skripten haben die Aufgabe, die Daten der einzelnen Formulare zu sammeln, Variablensubstitutionen vorzunehmen und dann am Ende die Rückantwort zu erstellen.

3.3 Ausführung beliebiger Skripten

Das bisher beschriebene System ermöglicht es dem Serviceanbieter beliebige Emails als Rückantwort zu verschicken. Dieser Mechanismus ist sehr mächtig; wird als Rückemiltext ein Programm oder Skript, das Variablen enthält, verwendet (z.B. ein SQL-Ausdruck oder ein Shell-Skript), kann dieses durch den Einsatz von automatischen Emailfiltern direkt ausgeführt werden. Natürlich empfiehlt es sich vorher gewisse Sicherheitsüberprüfungen durchzuführen, da auf Serverseite eine textuelle Substitution der Variablen vorgenommen wurde.

4 Zusammenfassung und Ausblick

Durch die Automatisierung der Erstellung von Webservices ist ein breiterer Einsatz dieser Technologie einfach machbar. Serviceangebote können somit wesentlich kostengünstiger realisiert werden.

Mit dem beschriebenen System kann nun der einzelne Mitarbeiter, welcher einen Service anbietet, diesen Service auch selbst einrichten und dies ohne Programmierkenntnisse. Dies macht Serviceangebote wesentlich flexibler.

Besonders in international tätigen Unternehmen wie HP ist die Bereitstellung von Dienstleistungen in verschiedenen Sprachen von elementarer Bedeutung. Mit dem MWS können nun Mitarbeiter die Dienstbeschreibungen und Rückantworten in ihrer Sprache erstellen und dann diese Angebote auf einem zur Verfügung gestellten Server selbst publizieren.

In den nächsten Entwicklungsschritten ist eine weitere Flexibilisierung des MWS geplant. Es sollen dem Benutzer Werkzeuge an die Hand gegeben werden, die es ihm ermöglichen, auch komplexere Web-Applikationen ohne Programmierkenntnisse zu erstellen. Hierbei wird ein Komponentenmodell auf der Basis von WebComposition [GeWG97] zu Einsatz kommen.

Literatur

- [CCLO96] Chih-Kai Chang, Gwo-Dong Chen, Baw-Jhiune Liu und Kou-Liang Ou. Language for developing collaborative learning activities on World Wide Web. In *Proc. IEEE 20th Annual International Computer Software & Applications Conference, COMPSAC'96*, Seoul, S Korea, August 21–23 1996.
- [Davi95] Andrew Davison. Coding with HTML forms. *Dr. Dobb's Journal of Software Tools for Professional Programmer* 20(6), Juni 1995, S. 5pp.
- [Eich97] David Eichmann. Software engineering (on) the World Wide Web. In *Proc. IEEE 19th International Conference on Software Engineering*, Boston, MA, USA, Mai 17–23 1997.
- [GeWG97] Hans-W. Gellersen, Robert Wicke und Martin Gaedke. WebComposition: An Object-Oriented Support System for the Web Engineering Lifecycle. In *Proc. of the 6th International World-Wide Web Conference (WWW6)*, Santa Clara, April 7–12 1997.
- [Joha97] Jay Johansen. Internet Programming — Template Driven Web Pages: A CGI program that parses templates. *Dr. Dobb's Journal of Software Tools* 22(11), November 1997, S. 74, 76–78, 93–94.
- [Mahe96] Robert C. Maher. Simple but useful tools for interactive WWW development. In *Proc. 26th Annual Conference on Frontiers in Education, FIE'96. Part 3 (of 3)*, Salt Lake City, UT, USA, November 6–9 1996.
- [NeMa95] E. Nebel und L. Masinter. Form-based File Upload in HTML. Request for comments: 1867, Network Working Group, Xerox Corporation, November 1995.
- [Prog97] Progress. WebSpeed. White paper, Progress Software Corporation, www.progress.com/internet/WebSpeed/, 1997.

Assistenten zur Unterstützung des Betriebs von vernetzten Systemen

Robert Scholderer

Kurzfassung

Der Betrieb eines vernetzten Systems ist dadurch gekennzeichnet, daß das Bedienpersonal eine Vielzahl von Aspekten bewältigen muß. Zur Unterstützung lassen sich Assistenten einsetzen, die Benutzer im Rahmen ihrer Tätigkeit anleiten und ggf. erforderliche Managementwerkzeuge bereitstellt. Im Mittelpunkt stehen nicht mehr Werkzeugfunktionen, sondern betriebliche Prozesse und gerade bestehende betriebliche Situationen, zu deren Bewältigung auf Managementfunktionen, aber auch auf andere nicht rechnergestützte Hilfsmittel zurückgegriffen werden muß. Assistenten ersetzen nicht die unterliegenden Managementwerkzeuge, sie sind vielmehr als eine Ergänzung zu verstehen, durch die der Einsatz von bestehenden Werkzeugen erleichtert wird.

Dieser Beitrag analysiert die Methoden zur rechnergestützten Anleitung des Bedienpersonals des Betriebs von vernetzten Systemen und stellt einen strukturierten Ansatz zum effektiven und effizienten Einsatz bestehender Managementwerkzeuge vor. Der Ansatz basiert auf prozeßorientierten Konzepten und verfolgt konsequent die betreibergerechte Sicht auf den Betrieb eines vernetzten Systems.

1 Einführung

Der effiziente und effektive Einsatz von traditionellen Managementwerkzeuge ist von der Arbeitsweise des Bedienpersonals, das den Betrieb von vernetzten Systemen erbringt, abhängig. Organisationssysteme werden hierzu nur in sehr überschaubaren Bereichen eingesetzt, wobei nur ein geringer Teil rechnergestützt erbracht wird. Aspekte, wie teilautomatisierte Jobläufe, Kalendersteuerungen und Ereignisverwaltungen, werden mit Applikationen erbracht und bieten nach wie vor eine komponentenorientierte Sicht auf das vernetzte System. Diese fast ausschließliche Sicht auf Netzkomponenten liefert keinen ausreichenden Ansatz, um den Betrieb eines vernetzten Systems sicher zu erbringen. Erforderlich wird ein Ansatz zur Assistenz bei betrieblichen Situationen, so daß der Betrieb nachvollziehbar erkennbar wird. Der prozeßorientierte Ansatz ermöglicht es, den Betrieb von vernetzten Systemen betreibergerecht zu modellieren [Thom97].

2 Ziele

Ziel ist es, die wesentlichen Aspekte für Assistenten aus Betriebsprozessen zu gewinnen und in prozeßunterstützende Managementwerkzeuge umzusetzen. Insbesondere wird die Frage, welches Werkzeug ein Benutzer im Rahmen seiner Tätigkeit benötigt, anhand von konkreten Beispielen analysiert. Betriebsprozesse zeichnen sich dadurch aus, daß sie abhängig von der jeweiligen betrieblichen Situation stark differieren und eine einheitliche Abarbeitung von Tätigkeiten und somit einen homogenen Einsatz von Werkzeugen nicht zulassen. Adaptive Eigenschaften sind von zentraler Wichtigkeit, denn nur so kann gewährleistet werden, daß Benutzern weiterhin Hilfestellungen auch in unvorhergesehenen Situationen zukommen [Mech95].

3 Assistenten im Betrieb

Assistenten ersetzen keine bestehenden Managementwerkzeuge, vielmehr bieten sie den Benutzern eine Unterstützung zur Orientierung in der Vielfalt der zu bearbeitenden Aspekte eines vernetzten Systems an. Derzeit werden Assistenten z.B. zur Störungsbearbeitung eingesetzt, um Fehlerursachen im vernetzten Systemen aufzufinden. Auf Management-Plattformen dienen Assistenten dazu, weitere Informationen über Netzkomponenten, etc. bereitzustellen.

Typische Bausteine von bestehenden Assistenten sind Diagnosefunktionen und Entscheidungsbäume, die auf Wissensbanken aufsetzen und diese mit dedizierten Regeln auswerten oder graphisch darstellen [Ante94, Bunn95]. Die Hilfestellungen, die das Betriebspersonal durch Assistenten erfährt, sind bisher vorwiegend auf die Entscheidungsfindung in komplexen IV-Systemen oder auf die Informationsbereitstellung beschränkt [Maie95]. Ein einheitlicher Aufbau von Assistenten ist bisher nicht erkennbar. Zunächst wird ein Anforderungskatalog für Assistenten erstellt, der die Grundlage für einen einheitlichen Aufbau bildet.

3.1 Anforderungen an Assistenten

Im weiteren sind die Eigenschaften formuliert, die Assistenten Benutzern bereitstellen müssen, um betriebliche Situationen zu erfassen.

- **Prozeßorientierte Anleitung:** Betriebsabläufe sind in jedem Prozeß begleitend zu unterstützen. Das Betriebspersonal hat aufgrund der Vielzahl von Workflows keinen Überblick, welche Schritte als nächstes erforderlich sind. Für den Assistenten bedeutet dies, daß dem Benutzer ein Überblick über seine Workflows zur Verfügung gestellt werden muß und zu jedem Workflow die nächsten Schritte vorgezeichnet werden [Dres97].
- **Diagnosefunktionen und Viewbildung:** In betrieblichen Situationen ist eine genaue Erfassung der Sachlage oft nur mit Expertenwissen oder einer genauen Recherche möglich. Deshalb müssen dem Bedienpersonal verschiedene Möglichkeiten bereitgestellt werden, die Entscheidungen anbieten oder zumindest die Erfassung der Sachlage unterstützen.

- **Verschatten von technischen Spezifika:** Die Heterogenität des vernetzten Systems erschwert es dem Betriebspersonal, ein einheitliches Wissen aufzubauen und den Betriebsprozeß effizient zu erbringen. Der Assistent abstrahiert von technischen Systemspezifika und stellt nur die wesentlichen Informationen und Funktionen bereit.
- **Entlastung der Verwaltungsaufgaben:** Ein Großteil der Arbeitszeit geht durch Verwaltungsaufgaben (z.B. Suche und Zugriff auf Informationen oder Werkzeuge) verloren. Das Bedienpersonal verliert bereits nach wenigen Arbeitsschritten den Überblick über die Aufbewahrungsorte. Dynamische Systeme, die die Informationen an strategisch wichtigen Stellen im Prozeß ablegen und bereitstellen, sind im Assistenten zu integrieren.
- **Lernfähigkeit:** Die betrieblichen Situationen, die in einem vernetzten System entstehen, sind vielfältig und einem ständigen Wandel unterworfen. Damit Benutzer möglichst optimal angeleitet werden kann, muß sich ein Assistent auf neue Situationen einstellen und das Bedienpersonal bei der Ausübung ihrer Tätigkeiten mit dem aktuellen Informationsstand über die Situationen begleiten und anleiten können.
- **Flexibilität:** Viele betriebliche Situationen sind dadurch gekennzeichnet, daß sie nicht einheitlich durchzuführen sind, also abhängig von den aktuellen Anforderungen erbracht werden müssen. Ein Assistent muß auf Abweichungen von vorgezeichneten Workflows vorbereitet sein und sich beim erneuten Einstieg oder Rückkehr in den Workflow mit den Werkzeugen und/oder Benutzern aufsynchronisieren und das weitere Vorgehen erneut unterstützen.
- **Offene Workflows:** Die Bearbeitung von Workflows ist dadurch gekennzeichnet, daß eine sequentielle Abarbeitung nicht möglich ist, denn Unterbrechungen und das Wechseln auf andere Workflows gehören zum Tagesgeschäft des Betriebs von vernetzten Systemen. Die verzahnte und unregelmäßige Abarbeitung von Workflows erhöhen die Komplexität des Aufgabengebietes von Benutzern. Zur Unterstützung bietet der Assistent alle vom Bedienpersonal begonnenen Workflows an und liefert auch den Status mit, mit dem der Workflow verlassen wurde.

Diese Anforderungen können von Assistenten nur dann erfüllt werden, wenn Assistenten mit den eingesetzten (Management-)Werkzeugen so kooperieren, daß sie die Informationen erhalten, die für eine Aussage über die bestehende betriebliche Situation herangezogen werden können. Der Assistent ist hierzug ständig im Hintergrund aktiv, nimmt Informationen entgegen und reagiert ggf. auf die neue Situation.

3.2 Struktureller Aufbau eines Assistenten

Damit Assistenten zur Unterstützung konkreter Szenarien implementiert werden können, entstand eine neue Architektur, die den prinzipiellen Aufbau von Assistenten beschreibt.

Von einzelnen Inhalten konkreter Szenarien wird hier abgesehen, vielmehr soll aufgezeigt werden, aus welchen Bausteinen Assistenten aufgebaut sind.

Die im Rahmen dieser Arbeit entworfene Architektur wird in zwei Teile eingeteilt. Den ersten Teil bildet die **Assistenten-Plattform**, die eine umfassende Infrastruktur bereitstellt. Insbesondere ist sie für den transparenten Zugriff auf Werkzeuge über verschiedene Systeme hinweg verantwortlich. Hierzu berücksichtigt sie die unterschiedlichen physischen Gegebenheiten und stellt einheitliche Dienste bereit.

Die zur betrieblichen Assistenz erforderlichen **Anwendungen** setzen auf der Assistenten-Plattform auf und stellen den Benutzern Funktionen über eine graphische Bedienoberfläche bereit. Die Funktionen unterteilen sich in die Kategorien Ausführung und Anleitung. Die ausführenden Funktionen ermöglichen dem Bedienpersonal aktiv Workflows mit den daran gekoppelten Werkzeugen anzustoßen, während zur Anleitung Methoden eingesetzt werden, die die nächsten Schritte in einem Ablauf vorzeichnen.

3.3 Architektur von Assistenten-Anwendungen

In diesem Abschnitt wird die Architektur der Assistenten-Anwendungen aufgezeigt. Die Bausteine können so konfiguriert werden, daß sie Benutzer in konkreten Szenarien unterstützend anleiten (Abbildung 1).

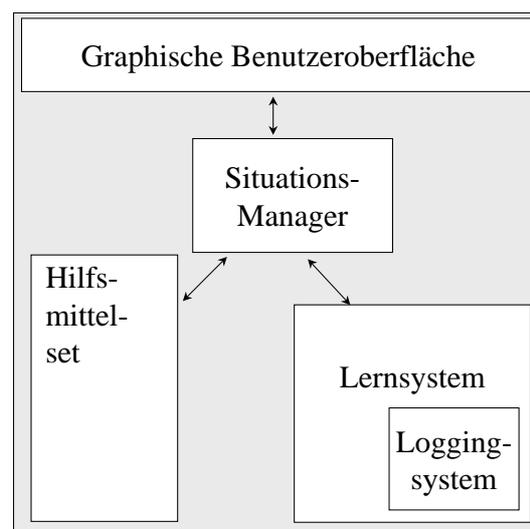


Abbildung 1: Aufbau von Assistenten-Anwendungen

Situationsmanager: Dieser Manager nimmt eine zentrale Position ein, da er im wesentlichen auf aktuelle betriebliche Situationen reagiert, indem er sie interpretiert und entscheidet, wie der Benutzer in einer veränderten Situation anzuleiten ist. Hierzu wird auf ein bestehendes Repertoire (Hilfsmittelset) zugegriffen und ein Hilfsmittel ausgewählt. Insbesondere ist der Situationsmanager damit beschäftigt, die Synchronisationspunkte bei Abweichungen festzulegen [GIGC97].

Hilfsmittelset: Dieses Set stellt für Routine- sowie für Ausnahmefälle eine Reihe von Funktionen zur Anleitung des Bedienpersonals bereit. Es beinhaltet Visualisierungsmethoden, Beschreibungen, Verweise auf bestehendes Dokumentationsmaterial, Diagnosefunktionen und Ablageverwaltungen von zu bearbeitenden Informationen. Situationsmanager und Benutzer können auf dieses Set zugreifen und die angebotenen Funktionen nutzen.

Lernsystem: Für eine ständige Adaption an neue Situationen unterstützt ein Lernsystem den Situationsmanager, in dem es die Aktionen von bearbeiteten Workflows mitprotokolliert und auswertet. Die Ergebnisse werden bei wieder auftretenden betrieblichen Situationen dem Situationsmanager zur Verfügung gestellt.

Die Bausteine der Anwendungen arbeiten eng mit der Plattform zusammen. Für diese Kooperation stellt die Assistenten-Plattform einheitliche Schnittstellen bereit, über die der Zugriff auf (Management-)Werkzeuge geregelt wird. Im nächsten Abschnitt wird unter Einbeziehung der aufgestellten Eigenschaften eine tragfähige Assistenten-Plattform entworfen, auf die die Basiskomponenten eines Assistent aufsetzen und somit einer Anleitung von Benutzern gerecht werden kann.

3.4 Wesentliche Dienste der Assistenten-Plattform

Auf der Plattform sind alle wesentlichen zur Assistenz erforderlichen Informationen integriert. Die Assistenten-Plattform stellt diese Informationen über verschiedene Grunddienste bereit, die von Assistenten benutzt werden können.

Ereignisdienst: Dieser Dienst ist permanent aktiv, sammelt und empfängt alle Meldungen des vernetzten Systems und leitet sie an den entsprechenden Assistenten weiter. Im Anschluß daran werden diese Meldungen nach den bekannten Methoden Rule-based-, Case-based- oder Model-based-System korreliert. Ergebnisse hieraus werden als Startereignis interpretiert, um Workflows zu initiieren.

Standardisierter Informationsaustausch: Damit der Assistent mit der Assistenten-Plattform plattformunabhängig arbeiten kann, wird ein standardisierter Informationsaustausch zwischen den verschiedenen Netzkomponenten gefordert. Für den Assistenten bedeutet dies, daß er mit allen Managementwerkzeugen so arbeitet und seine Informationen so bezieht, daß er eine koordinierte Anleitung des Workflows gewährleisten kann.

Steuerung der Workflows: Die einzelnen Teilschritte der Workflows liegen in einer vom Betreiber festgelegten Reihenfolge vor. Zur Einhaltung dieser Reihenfolge ist eine Steuerung erforderlich, die einzelne Teilschritte des Workflows koordiniert und bei Abweichungen Alternativen wählt oder mit vordefinierten Synchronisationspunkten die Konsistenz der Workflows wiederherstellt.

Verwaltung von Objekten: Ähnlich der in den vorangegangenen Kapiteln beschriebenen Werkzeugumgebungen sind in der Assistenten-Plattform Behälter bereitgestellt, in denen die Workflows und die damit verbundenen Zusatzinformationen abgelegt und verwaltet werden können. Die Verwaltung beinhaltet vor allem den konkreten Status eines Workflow, also welche Workflows sind aktiv, bzw. wurden vorübergehend verlassen.

Die einem Prozeß zugeordneten (Management-)Werkzeuge sind an die Assistenten-Plattform angebunden und können entweder über den Assistenten oder direkt genutzt werden.

4 Weiteres Vorgehen

Weiterführende Arbeiten auf diesem Gebiet sind die Modellierung der Arbeitsweise des Assistenten und dessen betrieblicher Einsatz. Hierzu werden Konzepte erstellt, die

z.B. bei Abweichungen von vorgezeichneten Abläufen Lösungen anbieten, so daß eine Unterstützung des Bedienpersonals weiterhin erfolgt. Das zentrale Problem ist hierbei, wie die Wechselbeziehung zwischen Managementwerkzeugen und Assistenten betreiberorientiert modelliert werden kann, da sich der Assistent auf neue oder alternative Werkzeuge einstellen muß. Arbeiten an der Assistenten-Plattform betreffen die Gestaltung von Managementwerkzeugen, um eine erfolgreiche Anbindung und somit eine Anleitung zu gewährleisten.

Hierzu wird ein Prototyp erstellt, der bestehende im Rahmen der bisherigen Arbeiten entwickelten Werkzeuge exemplarisch einbezieht. Die Assistenten-Plattform wird mittels einer CORBA-Architektur realisiert, an die Werkzeuge angebunden werden [OrHa97]. Ein Assistent, der auf der CORBA-Architektur aufsetzt, wird für ein betriebernahes Szenario implementiert.

Literatur

- [Ante94] S. Anteneh. *Zur Lösung komplexer mehrkriterieller Entscheidungsprobleme mittels Decision-Support-Systemen*. Frankfurt am Main : Lang. 1994.
- [Bunn95] D. W. Bunn. *Decision technology and intelligent decision support*. Amsterdam : North-Holland. 1995.
- [Dres97] M. Dressler. *Modellbasierte Navigationsstrategien in Executive Support Systems*. Universitätsverlag. 1997.
- [GIGC97] P. Gluchowski, R. Gabriel und P. Chamoni. *Management Support Systeme*. Springer. 1997.
- [Maie95] F. Maier. *Die Integration wissens- und modellbasierter Konzepte zur Entscheidungsunterstützung im Innovationsmanagement*. Berlin : Duncker und Humboldt. 1995.
- [Mech95] B. Mechler. *Intelligente Informationssysteme*. Addison-Wesley. 1995.
- [OrHa97] R. Orfali und D. Harkey. *Client/server programming with Java and CORBA*. New York : Wiley. 1997.
- [Thom97] F. Thomas. *Methodische Unterstützung organisatorischer Strukturierungsprozesse mit Hilfe eines clusteranalytisch gestützten Planungssystems*. Düsseldorf : VDI-Verl. 1997.

Reaktive und adaptive Platzierung von Proxy-Objekten

Jochen Seitz*

Kurzfassung

Mit dem Siegeszug der Internet-Dienste, insbesondere des *World Wide Web*, und den wachsenden Übertragungsraten auch im drahtlosen Bereich steigt die Nachfrage nach Multimedia-Anwendungen auch auf mobilen Endgeräten, die über eine drahtlose Anbindung (beispielsweise Funk) an das Festnetz verfügen. Dennoch muß man damit Einbußen in der Kommunikationsdienstgüte in Kauf nehmen, die vor allem Durchsatz, Übertragungsverzögerung und Fehlerrate betreffen. Somit müssen die Anwendungen auch auf sich verändernde Kommunikationsdienstgüte reagieren können. Die in diesem Beitrag vorgestellte Architektur liefert die Grundlage dafür, wobei sie sich auf CORBA-konforme verteilte, objektorientierte Anwendungen konzentriert.

1 Einführung und Motivation

Multimedia-Anwendungen stellen besondere Anforderungen an Kommunikationseigenschaften. Um dem Benutzer einer solchen Anwendung angenehmes Arbeiten zu ermöglichen, muß die Kommunikationsdienstgüte (*Quality of Service*) ausgehandelt und durch ständiges Überwachen und notfalls Eingreifen in die Übertragung garantiert werden.

Ist dies bereits in leitungsgebundenen Netzen eine sehr komplexe Aufgabe, ergibt sich im drahtlosen Umfeld durch die damit einhergehende nicht vorhersagbare Schwankung der Dienstgüte bezüglich Verzögerung oder Fehlerrate eine weitere Schwierigkeit. Vorkehrungen im Transportsubsystem können meist nicht alles abfangen, da die Kommunikationsdienstgüte bereits durch physikalische Effekte wie Signalreflektion oder Mehrwegeausbreitung beeinflusst wird. Neben einer an das drahtlose Umfeld angepaßten Kommunikationsarchitektur müssen daher auch die Anwendungen entsprechend auf sich verändernde Dienstgüten reagieren. Dies äußert sich zumeist in einer Reduktion des zu übertragenden Datenvolumens (z.B. durch Herausfiltern redundanter Daten oder durch Anfordern multimedialer Daten in geringerer Qualität) oder durch zeitlich verzögerte Kommunikationsvorgänge (beispielsweise das Absenden einer E-Mail erst bei einer meßbaren Verbesserung der aktuellen Kommunikationsdienstgüte).

*zur Zeit beurlaubt für einen Forschungsaufenthalt am Computing Department (*School of Engineering, Computing and Mathematical Sciences*), Lancaster University, United Kingdom.

Gerade im Bereich des *World Wide Web* mit den damit verbundenen Diensten wurden Einrichtungen (sogenannte “Proxies”) entwickelt, welche die Möglichkeit einer Anpassung an die schwankende Qualität drahtloser Übertragungsstrecken ermöglicht. Leider sind diese Ansätze zu sehr spezialisiert, als daß sie ohne weiteres allgemein auf den Bereich der verteilten multimedialen Anwendungen im mobilen Umfeld übertragbar sind. Gerade dafür soll eine Infrastruktur entwickelt werden, welche die Realisierung mobiler verteilter Anwendungen erleichtert, ohne daß dies eine zu große Anpassung der verteilten Anwendung bedarf. Als Plattform für derartig zu unterstützende verteilte Anwendungen wurde die *Common Object Request Broker Architecture (CORBA)* als zukunftssträftig ausgewählt, wobei deren Möglichkeit, auch JAVA als Programmiersprache zu unterstützen, bei der Implementierung Einsatz findet. In Abschnitt 2 wird die projektierte Architektur, welche eine reaktive und adaptive Platzierung von Proxy-Objekten (daher deren Bezeichnung RAPP) ermöglichen soll, vorgestellt und anhand eines Ablaufbeispiels näher erläutert. Der darauffolgende Abschnitt 3 stellt eine der Hauptkomponenten dieser Architektur, nämlich das *Lancaster Mobile Inter-ORB Protocol (LMIOP)* vor, das in einer Diplomarbeit konzipiert und implementiert wurde [Wolf98]. Abschließend werden der weitere Verlauf dieses Forschungsprojektes umrissen und die einzelnen Meilensteine vorgestellt.

2 Die RAPP-Architektur

Ausgangspunkt: Bei einer CORBA-basierten Anwendung soll die Dienstnehmerseite, d.h. der *Client*, mobil sein. Der Dienstgeber (*Server*) sei — der Einfachheit halber — im Festnetz angesiedelt. Da zwischen den beiden multimediale Daten ausgetauscht werden sollen, hat der Client, der als CORBA-konformes Objekt über ein Binding mit dem Server-Objekt kommuniziert, QoS-Anforderungen, die er an der Schnittstelle zum *Object Request Broker*, der die Kommunikationsinfrastruktur für die verteilte Anwendung bietet, angibt (siehe Abbildung 1).

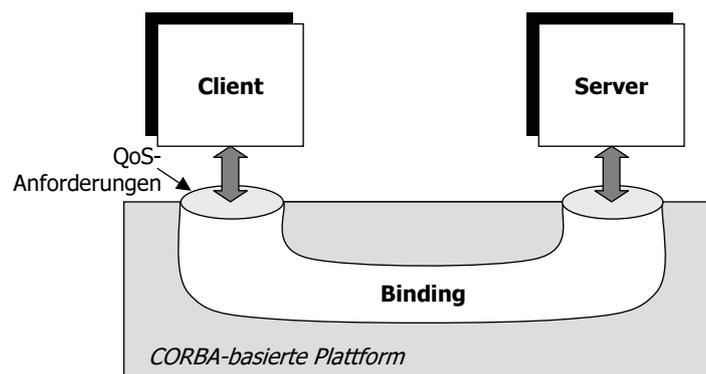


Abbildung 1: Ausgangslage

Wie bereits in Abschnitt 1 ausgeführt, kann ein Kommunikationssystem im mobilen Umfeld allein nicht die Kommunikationsdienstgüteeanforderungen befriedigen. Es ist vielmehr auf die Kooperation der Anwendung, d.h. deren Adaptivität angewiesen. Das bedeutet, daß das Client-Objekt nicht nur Anforderungen stellen darf, sondern auch auf Änderungen der Kommunikationsdienstgüte reagieren muß, wenn diese vom Kommunikationssystem nicht mehr abgefangen werden können. Andererseits müßte jedes

Client-Objekt mit einer spezifischen Adaptivität ausgestattet werden, die im Netz ein Gegenstück haben muß: Wenn sich beispielsweise eine Anwendung aufgrund sich verschlechternder Kommunikationsdienstgüte dazu entscheidet, den von ihr verwendeten Videostrom in bandbreitensparender geringer Qualität zu empfangen, so muß senderseitig ein entsprechender Filter eingesetzt werden. Daher ist anzustreben, daß die Adaptivität möglichst allen Anwendungen genügt und größtenteils durch die (somit erweiterte) CORBA-basierte Plattform erbracht wird.

Idee: Unterschiedliche Filter werden in sogenannten *Proxy*-Objekten realisiert, die in *Proxy Repositories* zur Verfügung stehen. Eine verteilte Anwendung spezifiziert die Art des von ihr erwarteten Kommunikationsdatenstroms. Bei einer Verschlechterung der Kommunikationsdienstgüte wird dann aufgrund dieser Information ein *Proxy*-Objekt ausgewählt und in den Datenstrom eingefügt, so daß der darin realisierte Filter die Anforderungen bezüglich Bandbreite oder Durchsatz senken kann.

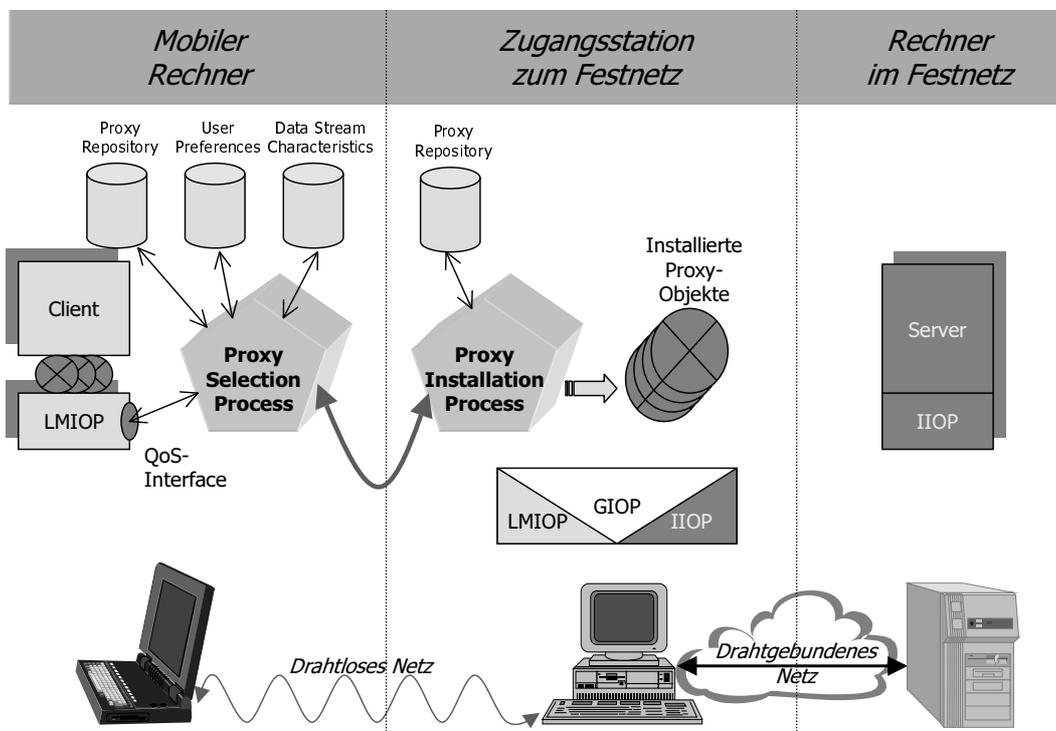


Abbildung 2: Die RAPP-Architektur

Aus diesen Überlegungen heraus wurde eine Architektur entwickelt (siehe Abbildung 2), die eine CORBA-Implementierung entsprechend erweitert. Die Hauptbestandteile dieser Architektur sind:

- der *Proxy Selection Process (PSP)*, der für die Auswahl eines Filters zuständig ist,
- der *Proxy Installation Process (PIP)*, der den Filter in den Datenstrom einfügt, ihn entsprechend parametrisiert und ihn bei Bedarf wieder löscht, und
- das *Lancaster Mobile Inter-ORB Protocol (LMIOP)*, welches das für CORBA definierte *General Inter-ORB Protocol (GIOP)* an die Gegebenheiten der drahtlosen Strecke anpaßt.

Die nachfolgenden beiden Unterabschnitte erläutern kurz die Funktionsweisen von *PSP* und *PIP*, wohingegen *LMIOP* im separaten Abschnitt 3 genauer vorgestellt wird.

2.1 Auswahl von *Proxy*-Objekten

Um eine möglichst generische Unterstützung der Adaptivität verteilter Anwendung zu ermöglichen, muß sich das Client-Objekt vor dem Aufbau der Kommunikation mit einer Charakterisierung der von ihm erwarteten Datenströme an der Plattform anmelden. Dazu wurde eine entsprechende Klassifikation von Datenströmen definiert, welche auf dem Mechanismus zur Spezifikation des Formats von Internet Mails gemäß dem *Multipurpose Internet Mail Extensions (MIME)*-Standard [BoFr96] basiert.

Aufgrund dieser Spezifikation und den Benutzervorgaben, die als *User Preferences*-File vorliegen, kann der *PSP* bei einer Verschlechterung der Kommunikationsdienstgüte, die er über das QoS-Interface von *LMIOP* angezeigt bekommt, einen Kommunikationsdatenstrom bestimmen, der an die verschlechterten Bedingungen adaptiert werden soll. Aufgrund der Klassifikation dieses Stromes wird eine Beschreibung eines dafür in Frage kommenden Filters abgeleitet, also z.B. ein Komprimierungsmechanismus für einen MPEG-Videostrom. Diese Beschreibung wird dann an den *PIP* geschickt.

2.2 Instanziierung von *Proxy*-Objekten

Der *PIP* erhält die Beschreibung eines zu instanzierenden Filters und bestimmt aufgrund der geforderten Eigenschaften und der möglichen *Proxy*-Objekte, die den Filter realisieren und in seinem *Proxy Repository* vorrätig sind, das am besten passende *Proxy*-Objekt. Dieses fügt er dann an einer eigens dafür konzipierten Schnittstelle in den betroffenen Kommunikationsdatenstrom ein. Dabei kann es durchaus dazu kommen, daß auf der Empfängerseite, d.h. auf dem mobilen Rechner, ein entsprechendes Gegenstück installiert werden muß. Dieses geschieht durch den *PSP* entweder, indem er sich das passende *Proxy*-Objekt aus dem dortigen *Proxy Repository* beschafft, oder indem der *PIP* das Objekt zum *PSP* schickt.

Mitunter kann es allerdings auch ausreichen, daß ein bereits installiertes *Proxy*-Objekt durch den *PIP* neu parametrisiert wird. So kann beispielsweise die oben angesprochene Komprimierung von MPEG-Videoströmen, die zuerst die Sendung der *B-Frames* unterdrückt hat, durch zusätzliches Auslassen der *P-Frames* noch verstärkt werden, was allerdings auf die Qualität des übertragenen Videostroms Auswirkungen hat [Yead96].

Desweiteren sollte bei einer wieder zunehmenden Dienstgüte dies auch Auswirkungen auf die installierten *Proxy*-Objekte haben. Das bedeutet, daß der *PIP* ebenfalls dafür zuständig ist, überflüssige *Proxy*-Objekte zu löschen bzw. die Filterbedingungen entsprechend zu ändern.

3 Das Lancaster Mobile Inter-ORB Protocol (LMIOP)

Bezüglich des Kommunikationssubsystems wurde für CORBA in der ersten Version keine Standardisierung vorgeschlagen. Mit der Tatsache, daß auch Objekte, die auf

unterschiedlichen *Object Request Broker*-Implementierungen laufen, miteinander kommunizieren sollen, wurde die Notwendigkeit eines Kommunikationsstandards erkannt und in Version 2 mit dem *General Inter-ORB Protocol (GIOP)* verabschiedet. Dieses Protokoll wird nicht nur zur Kommunikation zwischen verschiedenen ORBs verwendet, sondern auch als Kommunikationsgrundlage innerhalb eines ORBs eingesetzt. Die am weitesten verbreitete Umsetzung des allgemein gehaltenen *GIOP*-Standards ist das *Internet Inter-ORB Protocol (IIOP)* (siehe Abbildung 3).

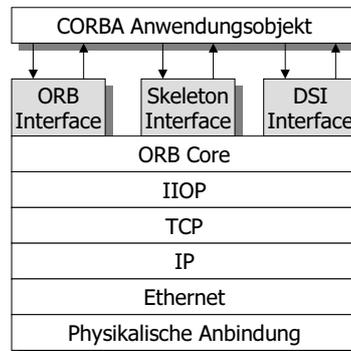


Abbildung 3: Instanziierung der mit CORBA definierten Protokollarchitektur

Dieses *IIOP* ist aber aufgrund des darunterliegenden Protokollstacks mit dem verbindungsorientierten *Transmission Control Protocol (TCP)* nur bedingt für den Einsatz im drahtlosen Umfeld geeignet. Da zudem im RAPP-Projekt die Möglichkeit bestehen mußte, an einer wohldefinierten Schnittstelle an aktuelle Kommunikationsdienstgütwerte zu kommen, mußte *IIOP* durch eine andere Protokollsuite ersetzt werden, die besser an das drahtlose Umfeld angepaßt ist. Diese unter dem Namen *Lancaster Mobile Inter-ORB Protocol (LMIOP)* definierte Protokollsuite hat folgende Eigenschaften:

- *Verwendung des User Datagram Protocols:*
Da das *User Datagram Protocol (UDP)* für den drahtlosen Übertragungsweg aufgrund seiner Einfachheit besser geeignet ist als *TCP*, wurde es als Transportprotokoll ausgewählt. Dies widerspricht allerdings dem *CORBA*-Standard, der die Verwendung eines verbindungsorientierten Transportprotokolls vorsieht. Daher muß *LMIOP* die Eigenschaften eines verbindungsorientierten Transportprotokolls selbst nachbilden.
- *Segmentieren:*
Für die drahtlose Kommunikation ist eine kleine Paketlänge besser geeignet, da öfters mit fehlerhaft übertragenen Paketen gerechnet werden muß, die dann zu wiederholen sind.
- *Fehlerkontrollmechanismen:*
Die Absicherung gegen Übertragungsfehler sind in *UDP* nur sehr rudimentär vorhanden, so daß *LMIOP* selbst entsprechende Mechanismen zur Verfügung stellen muß.
- *Übertragungswiederholung:*
Erkannte Paketfehler führen zu Übertragungswiederholungen, die — aufgrund der Eigenschaften drahtloser Übertragungswege — aggressiv durchgeführt werden müssen, da ein derartiger Fehler nicht auf Stausituationen hinweist, wie dies herkömmliche Transportprotokolle wie z.B. *TCP* annehmen.

- *Dienstgüteüberwachung und -meldung:*
Schließlich muß *LMIOF* wie gefordert die aktuelle Kommunikationsdienstgüte überwachen und bei einer Verschlechterung den entsprechenden *Proxy Selection Process* darüber informieren.

4 Meilensteine und Ausblick

Der aktuelle Stand dieses Projektes vom 28. Februar 1998 umfaßt die Konzipierung der Architektur, die Definition der formalen Spezifikation von Kommunikationsdatenströmen, eine Implementierung zur Eingabe von Benutzerpräferenzen sowie die Realisierung von *LMIOF* [Wolf98].

Folgende Meilensteine im RAPP-Projekt stehen noch an:

- eine formale Beschreibung zur Auswahl von *Proxy*-Objekten;
- Realisierung des *Proxy Selection Process*;
- Implementierung des *Proxy Installation Process*;
- prototypische Implementierung unterschiedlicher *Proxy*-Objects;
- Entwurf eines RAPP-spezifischen QoS-Monitors.

Mit Hilfe dieser vorgestellten Architektur soll es CORBA-basierten verteilten Anwendungen ermöglicht werden, auch im drahtlosen, mobilen Umfeld eingesetzt zu werden. Mit Hilfe eines veränderten Transportsubsystems wird auf die Problematik der drahtlosen Übertragungsstrecke eingegangen. Desweiteren ermöglicht die Überwachung der Kommunikationsdienstgüte einer Anwendung, auf deren Verschlechterung zu reagieren, indem die Anforderungen an die Bandbreite bzw. den benötigten Durchsatz durch Einfügen von *Proxy*-Objekten, die durch Filter den Datenstrom manipulieren, gesenkt werden.

Literatur

- [BoFr96] N. Borenstein und N. Freed. Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types. Request for Comments 2046, Network Working Group, November 1996.
- [Wolf98] C. Wolf. Adaptive Mobile Applications based on a CORBA Infrastructure. Diplomarbeit, Institut für Telematik, Universität Karlsruhe und Computing Department, SECAMS (*School of Engineering, Computing and Mathematical Sciences*), Lancaster University, United Kingdom, Februar 1998.
- [Yead96] N.J. Yeadon. *Quality of Service Filtering for Multimedia Communications*. Ph.D. Thesis, Computing Department, Lancaster University, Lancaster LA1 4YR, UK, Mai 1996.

Quality of Service (QoS) in verteilten objektorientierten Systemen

Rainer Ruggaber

Kurzfassung

Im Bereich von unternehmenskritischen Anwendungen oder Anwendungen, die Realzeitanforderungen stellen, müssen angebotene Dienste mit einer garantierten Qualität erbracht werden. Die Qualitätsparameter und Mechanismen zu deren Erbringung in verteilten Systemen unterscheiden sich jedoch wesentlich von denen auf Netzwerkebene oder in verteilten Multimedia-Systemen, so daß erheblicher Forschungsbedarf besteht. Gegenwärtig sind erste Untersuchungen und Standardisierungsbemühungen in diesem Bereich im Gange. In diesem Bericht werden die Anforderungen an die Umsetzung von QoS in verteilten Anwendungen, zwei Konzepte zur Umsetzung von QoS in verteilten CORBA-Anwendungen und die Standardisierungsbemühungen der ISO und der OMG in diesem Bereich vorgestellt.

1 Einleitung

Eine zunehmende Anzahl Anwendungen aus den Bereichen Medizin, Multimedia, Finanzdienstleistungen, Vermittlungssystemen oder verteilten Fertigungssystemen fordern Dienstgarantien wie beispielsweise Echtzeit-Betrieb, Konsistenz, Fehlertoleranz oder Sicherheit. Die Forschung in Bezug auf Quality-of-Service hat sich bisher auf den Bereich der Netzwerkebene und die Unterstützung von Multimedia-Anwendungen beschränkt. In diesen Bereichen sind die notwendigen Parameter und Mechanismen gut untersucht und verstanden. Die Anforderungen zur Unterstützung von QoS in verteilten Anwendungen unterscheiden sich jedoch wesentlich von denen in verteilten Multimedia-Systemen. Verbreitete Middleware-Plattformen wie beispielsweise CORBA bieten jedoch keine Unterstützung für Garantien bezüglich der Diensterbringung. Aktuelle Arbeiten und Standardisierungsbemühungen in diesem Bereich versuchen diese Lücke zu schließen.

2 Grundlagen

Um in einem verteilten System auf Anwendungsebene Dienste mit einer garantierten Qualität erbringen zu können, müssen alle Schichten eines Systems ihre Dienste mit einer garantierten Qualität zu erbringen. Auf Anwendungsebene konzentrierten sich die

Untersuchungen bisher auf Multimedia-Anwendungen (2.3). Auf Netzwerkebene sind verschiedene Technologien verfügbar, jedoch ist im Augenblick nur ATM in der Lage Dienste mit einer garantierten Qualität zu erbringen (2.2).

2.1 Definitionen

Die *Dienstqualität* eines Dienstes beschreibt die Eigenschaften, mit denen der Dienst erbracht wird. Die Dienstqualität wird durch eine Reihe von *Qualitätsparametern* und *weiteren Dienstmerkmalen* beschrieben. *Qualitätsparameter* beschreiben quantitativ erfassbare Eigenschaften, wie beispielsweise Durchsatz, Verzögerung, Jitter oder Zuverlässigkeit. Demgegenüber dienen die *weiteren Dienstmerkmale* zur Formulierung nicht quantifizierbarer Dienstmerkmale, wie Sicherheit, Synchronisation oder Kommunikationsform.

Eine detaillierte Beschreibung der benötigten Qualitätsparameter erfordert neben deren quantitativer Beschreibung auch eine Beschreibung der benötigten *Dienstklasse*. Die *Dienstklasse* beschreibt qualitativ, wie Qualitätsparameter interpretiert und erbracht werden sollen. Unterschieden wird eine *deterministische* Dienstklasse, die auch im schlechtesten Fall die geforderte Qualität erbringt, eine *statistische* Dienstklasse, die die Erfüllung nur mit einer bestimmten Wahrscheinlichkeit garantiert, und eine *bestmögliche* Dienstklasse, die keinerlei Garantien übernimmt [Zitt95].

Freiheitsgrade bestehen bei der Dynamik, mit der sich die Qualitätsanforderungen einer Anwendung ändern können. Bei *statischen* QoS wird zum Entwicklungszeitpunkt der Anwendung deren QoS durch verschiedene Design- und Implementierungsentscheidungen festgelegt. Damit sind diese für jede Instantiierung der Anwendung gleich, und können zur Laufzeit nicht mehr geändert werden. Im Falle von *dynamischen* QoS werden diese zur Laufzeit ausgehandelt und gelten somit nur für eine bestimmte Zeit. Die Zeitdauer, für die die ausgehandelten QoS gelten, kann dabei von einem Methodenaufruf über einen Block von Methodenaufrufen bis zu einem Bindevorgang gelten.

Ein weiterer Freiheitsgrad ist der Ansatz zum Dienstmanagement. Dieser reicht von *Managementansätzen* bis zu *anwendungsorientierten Ansätzen*. Beim *anwendungsorientierten* Dienstmanagement wird die gesamte Funktionalität des Managements in die Anwendung verlagert. In diesem Fall muß die Anwendung auf veränderte Ressourcen reagieren. Es entstehen adaptive Anwendungen. Demgegenüber stellen *Managementansätze* der Anwendung Funktionen zum Dienstmanagement zur Verfügung.

Eine spezielle Art von Anforderungen, sind diejenigen der Realzeitsysteme. Gemäß DIN 44300 ist unter *Echtzeit-* bzw. *Realzeitbetrieb* eines Rechnersystems die Fähigkeit eines Rechnersystems zu verstehen, daß *Verarbeitungsergebnisse innerhalb einer vorgegebenen Zeitspanne verfügbar sind* [ReLe94]. Die Anforderungen der Realzeitsysteme beziehen sich auf den Qualitätsparameter Verzögerung. Realzeitsystemen bilden somit eine Teilmenge der verteilten Systeme, die Garantien bezüglich der Dienstqualitäten benötigen.

2.2 QoS auf Netzwerkebene

Im Internet Protocol (IP) besteht die Möglichkeit mit Hilfe des *Type-of-Service*-Feld die Dienstqualität in IP-Datagrammen zu beschreiben. Dadurch ist eine Spezifikation

der Priorität oder des geforderten Durchsatzes möglich. Das Problem an diesem Ansatz ist jedoch, daß das *Type-of-Service*-Feld meist nicht berücksichtigt wird.

RSVP (Resource ReSerVation Protocol) [BrZh94] ist ein Kontrollprotokoll zur Ressourcenreservierung im Internet. Dazu werden entlang eines Weges von Sender zu Empfänger Ressourcen reserviert. Der Datentransfer wird weiterhin über IP abgewickelt, und nutzt die Reservierungen entlang des Weges. Jedoch kann RSVP keine Garantien für bestimmte QoS übernehmen.

In ATM [Zitt95] bietet die Dienstschnittstelle der ATM-Adaptionsschicht höheren Schichten ein vielfältiges Dienstespektrum an. Es werden asynchrone, synchrone und isochrone Dienste angeboten. Die zu erbringenden Dienste werden in vier Klassen (A-D) eingeteilt. Zur Definition dieser Klassen werden die Merkmale Bitrate, Zeitbeziehung zwischen Sender und Empfänger sowie die Verbindungsart herangezogen. Eine Anwendung teilt dem ATM-System beim Verbindungsaufbau die benötigte Qualität mit. Sind genügend Ressourcen im Netzwerk vorhanden, um die Anforderung zu erfüllen, wird eine Reservierung vorgenommen und die Verbindung aufgebaut.

2.3 QoS in verteilten Multimedia-Systemen

Charakteristische Kommunikationsform in verteilten Multimedia-Systemen ist der *Datenstrom*. Ein Datenstrom ist eine unidirektionale Punkt-zu-Punkt- oder Punkt-zu-Mehrpunkt Kommunikationsverbindung. Jeder Datenstrom ist mit einer Dienstqualität assoziiert. Logisch zusammengehörende Datenströme können zu *Sitzungen* zusammengefaßt werden, zwischen denen Beziehungen definiert werden können [Schm97].

Die Datenströme in Multimedia-Systemen dienen zur Übertragung von komprimiertem oder unkomprimiertem Video oder Ton. Die unkomprimierte Übertragung setzt dabei einen konstanten Durchsatz voraus, wohingegen der benötigte Durchsatz bei der Übertragung von komprimierten Daten nicht eindeutig vorhergesagt werden kann. Insbesondere bei der Übertragung von Ton, in geringerem Maße auch für Video, ist auf eine geringe Verzögerung und einen kleinen Jitter zu achten. Bei Multimedia-Daten ist ein kleiner Verlust (1-2%) von Informationen meist tolerierbar [Zitt95].

3 QoS in verteilten objektorientierten Systemen

Die Anforderungen und Probleme die bei der Integration von QoS in verteilte objektorientierte Systeme entstehen unterscheiden sich wesentlich von denen verteilter Multimedia-Systeme. Aufgrund der weiten Verbreitung von CORBA werden speziell die Anforderungen an eine Integration in CORBA vorgestellt. Die Aktivitäten im Bereich von QoS in verteilten objektorientierten Systemen konzentrieren sich auf die Standardisierung und die Entwicklung erster Systeme die Unterstützung für QoS bieten.

3.1 Middleware

Eine Middleware bietet Dienste an, die die Realisierung verteilter Anwendungen vereinfachen. Wichtige Dienste in diesem Rahmen sind die Möglichkeit, Objekte auf entfernten Knoten zu referenzieren, Aufrufe transparent an Objekte auf entfernten Knoten

weiterzuleiten und das Ergebnis des Aufrufs zurückzuliefern. Dazu müssen die Parameter eines Methodenaufrufs auf der Client-Seite in ein Format verpackt werden, das über das unterliegende Netzwerk übertragen werden kann (Marshalling). Auf der Server-Seite werden die Parameter wieder entpackt (Demarshalling) und der Aufruf an das entsprechende Objekt weitergeleitet. Daneben ist es möglich, unterschiedliche Programmiersprachen, Betriebssysteme und Hardware-Plattformen für Client und Server, sowie die verwendeten Netzwerkprotokolle zu verdecken.

CORBA (Common Object Request Broker) [OMG97a] stellt eine von der OMG (Object Management Group) standardisierte Middleware dar. Im Rahmen von CORBA werden oben genannte Dienste vom ORB (Object Request Broker) erbracht. Weitere Dienste werden im Rahmen der CORBA-Spezifikation von den Common Object Services und den Common Facilities zur Verfügung gestellt. Aufgrund der großen Bedeutung von CORBA werden im folgenden speziell die Arbeiten im Bereich von CORBA vorgestellt.

3.2 Anforderungen

Eine Reihe von Unterschieden verteilter Multimedia-Systeme und verteilter objektorientierter Systeme zeigt Tabelle 1. Aus dieser Tabelle lassen sich eine Reihe von Anforderungen ableiten.

Eigenschaft	Multimedia-Systeme	verteilte OO-Systeme
Kommunikationsform	Datenstrom	Request-Response
Bitfehlerrate	geringe Fehlerrate tolerierbar (1-2 %)	keine Fehler tolerierbar
Durchsatz	Sprache: gering (64 kBit/s) Video: hoch (1,5 MBit/s)	meist gering
Jitter	sehr gering	egal
Übertragungsverzögerung	mittel	sehr gering, entscheidender Leistungsfaktor
Nutzungsmuster	a priori bekannt	selten bekannt oder ableitbar
Zuverlässigkeit Ausfallsicherheit	mittel	anwendungsabhängig, meist hoch
Konsistenz	mittel, Zellen verwerfen erlaubt	anwendungsabhängig, meist hoch
Interaktionsmuster	einfach zu beschreiben	komplex
Struktur der Kommunikationspartner	Sender-Empfänger Beziehung	Aufruf kann sich rekursiv fortpflanzen

Tabelle 1: Vergleich Multimedia-Systeme und verteilte objektorientierte Systeme

Ausgangspunkt der Integration von Qualitätsparametern in CORBA ist die Identifikation der Qualitätsparameter, die für verteilte Anwendungen relevant sind. Von der ISO und der OMG wurde bisher nur eine Reihe von elementaren Qualitätsparametern vorgeschlagen (siehe Abschnitt 3.3). Eine zentrale Eigenschaft von Qualitätsparametern muß deren Orthogonalität sein.

Im nächsten Schritt muß es für eine Anwendung möglich sein, mit Hilfe der identifizierten Qualitätsparameter ihre Anforderungen zu formulieren. Die Formulierung der komplexen Interaktionsmuster zwischen Client und Server stellt ein großes Problem dar. Auf der einen Seite sind die Informationen über typische Interaktionsmuster ungenügend, auf der anderen Seite wurden noch keine Verfahren entwickelt, diese zu formulieren und dem System mitzuteilen.

Die Qualitätsparameter einer verteilten Anwendung lassen sich nicht direkt auf solche der Netzwerkebene abbilden. Eine einfache Abbildung wäre möglich, falls man sich bei der Definition der QoS-Parameter auf Anwendungsebene an denen der Netzwerkebene orientiert. Dies führt jedoch dazu, daß der Anwendungsentwickler die Anforderungen der Anwendung nicht auf einer semantisch angepaßten Ebene formulieren kann. Löst man sich bei der Definition der QoS-Parametern jedoch von denen der Netzwerkebene, entstehen Abbildungsprobleme. Probleme, die bei der mehrfachen Abbildung von QoS-Parametern in geschichteten Systemen entstehen, sind in der Literatur beschrieben [Schm97].

Mit dem ORB existiert eine zusätzliche Schicht, die bei der Erbringung der Qualitätsparameter berücksichtigt werden muß. Einerseits existiert auf diese Weise eine zusätzliche Abbildungsschicht, andererseits muß auch der ORB in der Lage sein, seine Dienste mit einer gewissen Garantie zu erbringen. Innerhalb des ORB auf der Client- und der Server-Seite sind aufwendige Marshalling- und Demarshalling-Operationen durchzuführen. Zusätzlich muß auf Server-Seite die gewünschte Implementierung ausgewählt werden. Diese Operationen müssen bei der Aushandlung von Qualitätsparametern berücksichtigt werden.

Auf Server-Seite besteht zusätzlich das Problem, daß gleichzeitig mehrere Methodenaufrufe aktiv sein können. Die Bearbeitung des aktuellen Methodenaufrufs kann also durch bereits aktive Methodenaufrufe verzögert werden.

Weiterhin ist zu berücksichtigen, daß ein Server, um einen Dienst zu erbringen, Methoden auf weiteren Objekten aufruft. Auf diese Weise kann sich ein Methodenaufruf rekursiv fortsetzen. Die Abbildung der Qualitätsparameter auf diese Kette von Methodenaufrufen muß möglich sein und festgelegt werden.

3.3 Standardisierung

Die Standardisierungsbemühungen im Bereich QoS in verteilten Systemen finden im Augenblick in drei verschiedenen Gruppen statt. Die OMG bemüht sich um eine Erweiterung von CORBA um QoS. Die ISO versucht das ODP (Open Distributed Processing) um entsprechende QoS-Parameter und Mechanismen zu erweitern. Vor diesen beiden Gruppen hat sich bereits eine RT-SIG gebildet mit dem Ziel CORBA um Echtzeitfähigkeiten zu erweitern. In keiner der drei Gruppen sind die Arbeiten abgeschlossen, jedoch bemühen sich die OMG und die ISO bei der Standardisierung zusammenzuarbeiten.

Ziel der OMG ist die Definition und Spezifikation der Erweiterungen die notwendig sind um QoS in CORBA-basierten Systemen anzubieten [OMG97b]. Hierzu sollen keine Änderungen, sondern nur Erweiterungen an der CORBA-Architektur durchgeführt werden. Zusätzlich sollen keine Einschränkungen bezüglich der unterstützten Qualitätsparameter vorgenommen werden. Die OMG will sowohl Managementansätze wie

anwendungsorientierte Ansätze zum Dienstemanagement unterstützen. Zur Unterstützung des anwendungsorientierten Ansatzes wird die Möglichkeit zur Verfügung gestellt, mehrere Implementierungen zu einem Interface anzubieten, die jeweils unterschiedliche QoS unterstützen. Daneben soll die Möglichkeit, sowohl statische wie dynamische QoS zu realisieren, angeboten werden. Weitere Anforderungen die identifiziert wurden sind das aushandeln und überwachen von QoS, die Benachrichtigung bei Verletzung ausgehandelte QoS und die Unterstützung verschiedener Dienstklassen. Bisher wurden noch keine Qualitätsparameter oder weitere Dienstmerkmale vorgeschlagen.

Die ISO [ISO97] verfolgt bei der Standardisierung von QoS in ODP einen umfassenderen Ansatz im Vergleich zur OMG, so daß die ISO die gesamte von der OMG vorgeschlagene Funktionalität ebenfalls zur Verfügung stellt. Ein Schwerpunkt der aktuellen Arbeit ist die Integration der QoS-Funktionalität in das Konzept der ODP-Viewpoints. Als elementare Qualitätsparameter, die sich auf die Interaktion zwischen zwei Objekten beziehen, wurden von der ISO identifiziert: Verzögerung, Jitter, Durchsatz, Fehlerwahrscheinlichkeit, Verfügbarkeit, Zuverlässigkeit, Integrität, Sicherheit und Priorität. Auch von der ISO wurde noch keine Notation für QoS-Parameter vorgeschlagen.

Daneben versucht die RT CORBA SIG die Standardisierung von Echtzeit-CORBA voranzutreiben [WDGS⁺97]. Für die Realisierung von Echtzeit-Anwendungen auf Basis von CORBA muß es möglich sein dem System die Zeitanforderungen mitzuteilen, die durchgesetzt werden müssen. Zwei Anforderungsbereiche wurden identifiziert. Die Anforderungen an die Systemumgebung sind Synchrone Uhren auf allen Knoten, eine maximale Nachrichtenverzögerung die vom Netzwerk zugesichert wird, ein prioritätsbasiertes Scheduling und die Unterstützung von Priority Inheritance auf Seiten des Betriebssystems. Die Anforderungen an den ORB sind ein Zeit-Typ, die Unterstützung einer knotenübergreifenden globalen Priorität, spezielle Echtzeit-Ausnahmen, Priority Inheritance, Erbringung der Dienste des ORB mit Qualitätsgarantie und die Spezifikation von Echtzeitereignissen. Jedoch bilden die Qualitätsparameter, die zur Formulierung der Anforderungen für den Realzeitbetrieb benötigt werden, lediglich eine Teilmenge der von der ISO vorgeschlagenen Qualitätsparameter.

3.4 Systeme

In diesem Abschnitt werden zwei Systeme vorgestellt, die auf unterschiedliche Weise CORBA-Anwendungen QoS zur Verfügung stellen und realisieren.

3.4.1 The ACE Orb (TAO)

Das Ziel von TAO [ScLM97] war die Entwicklung eines CORBA 2.0 kompatiblen Echtzeit ORBs. Im Rahmen dieses Projektes wurden zunächst kommerzielle ORBs (Orbix, Visibroker) auf ihre Echtzeittauglichkeit untersucht [GoSc97]. Dabei wurde festgestellt, daß deren Verarbeitungszeit aufgrund nicht optimierter Algorithmen mit der Last ansteigt. Aufgrund der Ergebnisse der Analyse wurden speziell für die entdeckten Engpässe optimierte Algorithmen entwickelt, die einerseits ein besseres Verhalten unter Last zeigen, wichtiger jedoch, sich deterministisch verhalten.

Einsatzgebiete für TAO sind Anwendungen aus dem harten Realzeitbetrieb. Möglich wird das deterministische Verhalten aufgrund mehrerer Maßnahmen. Zum einen ist der

Betrieb von TAO über ein ATM-Netzwerk vorgesehen, das auf Netzwerkebene einen hohen Durchsatz, geringe Verzögerung und Unterstützung für QoS bietet. Daneben kommt eine optimierte Variante des GIOP (General Inter-ORB Protocol) zum Einsatz. RIOP (Real-Time Inter-ORB Protocol) erweitert GIOP/IIOP um QoS-Attribute. Zusätzlich ist eine Anmeldung der QoS-Anforderungen vor Ausführung des Programms notwendig. Kann das System die gewünschten QoS bieten, werden die notwendigen Ressourcen in den Endsystemen und dem Netzwerk reserviert. Daneben kann mit Hilfe dieses a priori Wissens beispielsweise eine optimale Hash-Funktion für die Auswahl der Server-Implementierung bestimmt werden. Auf der anderen Seite bietet TAO keine komplexen Schnittstellen wie das Dynamic Invocation Interface (DII) an.

3.4.2 Quality Objects (QuO)

Zinky, Bakken und Schantz von BBN zielen mit ihrem QuO (Quality Objects) Ansatz [ZiBS97] auf einen Einsatz von CORBA im WAN-Bereich. Gegenüber TAO (3.4.1) wählen sie einen anwendungsorientierten Ansatz zum Dienstemanagement. Ihrer Ansicht nach sind die Verhältnisse bei der Nutzung von CORBA-Anwendungen im WAN-Bereich nicht vorhersehbar oder steuerbar, so daß die Anwendung entsprechend der vorhandenen Ressourcen reagieren muß. Voraussetzung für eine solche Vorgehensweise ist, daß CORBA mehrere Implementierungen zu einem Interface unterstützt.

Um die gewünschte Adaptivität der Anwendung zu erreichen, muß mit Hilfe der QDL (Quality Definition Language), die sich in mehrere Spezialsprachen aufteilt und eine Erweiterung der IDL ist, spezifiziert werden, unter welchen Bedingungen die Anwendung in der Lage ist zu arbeiten und welche Implementierung in welchem Bereich der Ressourcenverfügbarkeit gewählt werden muß. Nachdem dem ORB diese Informationen mitgeteilt wurden, kann dieser abhängig von der aktuellen Ressourcensituation die geeignete Implementierung auswählen. QuO bietet keine harten Echtzeitgarantien. QuO bietet die Möglichkeit mit Hilfe von RSVP auf Netzwerkebene gewisse QoS sicherzustellen.

3.4.3 Vergleich

Die vorgestellten Systeme (TAO, QuO) versuchen auf unterschiedliche Weise die QoS auf Anwendungsebene sicherzustellen. In QuO versucht die Anwendung durch Adaption an die vorhandenen Ressourcen und Reservierung von Übertragungskapazitäten auf Netzwerkebene durch RSVP bestimmte Garantien geben zu können. TAO hingegen versucht durch Nutzung von ATM zur Erlangung von Garantien auf Netzwerkebene, Anmeldung und Reservierung der Anforderungen beim ORB vor Start der Anwendung und einer Optimierung des ORBs harten Echtzeitanforderungen zu genügen. Beide Systeme verfolgen einen statischen Ansatz, was die Möglichkeit der Verteilungssteuerung angeht.

4 Zusammenfassung

In diesem Bericht wurde über die aktuellen Arbeiten auf dem Gebiet der QoS für verteilte objektorientierte Anwendungen berichtet. Dazu wurden die wesentlichen Unter-

schiede von QoS in verteilten Multimedia-Systemen und verteilten objektorientierten Systemen vorgestellt. Daraus abgeleitet wurde eine Reihe von Anforderungen, die eine Middleware-Plattform bieten muß, um einer Anwendungen Garantien bezüglich der Dienstleistung zu bieten.

Die Standardisierungsbemühungen der ISO und der OMG zielen im Augenblick auf eine integrierte Architektur zur Unterstützung von QoS. Bisher wurde noch keine konkreten Qualitätsparameter festgelegt, die unterstützt werden müssen, noch wurde eine verbindliche Art festgelegt, auf welche Weise dem System die geforderten QoS mitgeteilt werden sollen.

Keines der zwei vorgestellten Systeme unterstützt jedoch bisher die speziellen Möglichkeiten, die sich durch verteilte objektorientierte Systeme ergeben. Insbesondere die Lockerung bestimmter QoS wie Konsistenz oder Aktualität und die Berücksichtigung alternativer Verteilungspolitiken sind noch nicht untersucht.

Literatur

- [BrZh94] Bob Braden und Lixia Zhang. RSVP: A Resource ReSerVation Protocol. *Connexions*, August 1994.
- [GoSc97] Aniruddha S. Gokhale und Douglas C. Schmidt. Measuring and Optimizing CORBA Latency and Scalability Over High-speed Networks (not yet published). *IEEE Transaction on Computers*, 1997.
- [ISO97] Open System Interconnection, Data Management and Open Distributed Processing. *ISO/IEC JTC1/SC21 N QoS1, Working Draft*, Juli 1997.
- [OMG97a] *The Common Object Request Broker: Architecture and Specification, Revision 2.1*, August 1997.
- [OMG97b] OMG Green Paper: Quality of Service (QoS). *QoS Green Paper Version 0.4a*, Juni 1997.
- [ReLe94] Ulrich Rembold und Paul Levi. *Realzeitsysteme zur Prozeßautomatisierung*. Carl Hanser Verlag. 1994.
- [Schm97] Claudia Schmidt. *Integrierte Managementarchitektur für qualitätsorientierte Kommunikationsdienste*. Dissertation, Universität Karlsruhe, 1997.
- [ScLM97] Douglas C. Schmidt, David L. Levine und Sumedh Mungee. The Design of the TAO Real-Time Object Request Broker. *Computer Communications Journal*, Summer 1997.
- [WDGS⁺97] Victor Fay Wolfe, Lisa Cingiser DiPippo, Roman Ginis, Michael Squadrito, Steven Wohlever, Igor Zykg und Russel Johnston. Real-Time CORBA. In *Proceedings of the 3. IEEE Real-Time Technology and Application Symposium*, Montreal, Canada, Juni 1997.
- [ZiBS97] John A. Zinky, David E. Bakken und Richard D. Schantz. Architectural Support for Quality of Service for CORBA Objects. *Theory and Practice of Object Systems* 3(1), 1997.

- [Zitt95] Martina Zitterbart. *Hochleistungskommunikation*. Oldenburg Verlag, 1995.

Topologie- und Dienstgütererfassung in ATM-Netzwerken

Hajo R. Wiltfang

Kurzfassung

Netzwerkmanagement ist ein wichtiger Aspekt beim Betrieb von rechnerbasierten Netzwerken, vor allem für neue und recht komplexe Technologien wie ATM. Für alle Komponenten im Netzwerk sollte zu jeder Zeit deren Zustand und Funktionsfähigkeit durch das Netzwerkmanagement überwacht werden, damit an einem zentralen Punkt eine globale Übersicht besteht und Reaktionen auf festgestellte Fehler sofort erfolgen können. Leider sind für die ATM-Technologie die derzeit verfügbaren Managementschnittstellen von einzelnen Komponenten aufgrund der andauernden Standardisierung und herstellerspezifischer Eigenschaften zum Teil sehr verschieden. Dieser Tatsache begegnet die in diesem Beitrag vorgestellte hierarchische Managementarchitektur für ATM-Netzwerke mit einer einheitlichen Schnittstelle auf der untersten Ebene, die durch Proxy-Managementagenten erreicht werden kann. Darauf aufbauend kann die mittlere Ebene der Architektur unterstützende Dienste für das Management anbieten, die für eine Vielzahl von Anwendungen auf der oberen Ebene von Interesse sind. Der ATM-Netzwerkmonitor stellt eine wesentliche Instanz dieser mittleren Ebene dar. Die vom ATM-Netzwerkmonitor realisierte Topologie- und Dienstgütererfassung wird zusammen mit seiner Struktur im Hauptteil dieses Beitrags vorgestellt.

1 Einleitung

Die heutige Gesellschaft ist durch ein mittlerweile grundlegendes Bedürfnis zur globalen Kommunikation geprägt. Dies bezieht sich nicht nur auf die persönliche Kommunikation mit Hilfe der Telefonie oder auf die Versorgung mit Information über Radio und Fernsehen, sondern vermehrt auch auf die rechnergestützte Kommunikation über Netzwerke. Die steigende Ausdehnung sowie die zunehmende Komplexität der rechnerbasierten Netzwerke macht dabei ein geeignetes Netzwerkmanagement unumläßlich. Die wesentlichen Abläufe in den Netzwerken müssen jederzeit erfaßt werden können und notwendige Konfigurationsänderungen sollten direkt vorzunehmen sein. Die Bedeutung des Netzwerkmanagements steigt zusätzlich mit der Komplexität der eigentlichen Netztechnologie, wie dies beispielsweise bei ATM-basierten Netzwerken der Fall ist.

Gerade bei so neuen Technologien wie ATM ist jedoch am Anfang das Augenmerk der Entwickler und Hersteller primär auf die wesentliche Funktionalität der Technologie gelegt, um schnell einsetzbare Produkte am Markt plazieren zu können. Darunter

leidet in der frühen Phase der Produktentwicklung oft die Managementfunktionalität dieser Komponenten. Bei der ATM-basierten Technik kommt noch hinzu, daß sich viele Aspekte dieser Technik noch immer bzw. fortlaufend in Standardisierungsprozessen befinden, so daß teilweise keine abschließenden Lösungen für das Management erstellt werden können. Aus den beschriebenen Gründen kann die heute verfügbare Managementfunktionalität von ATM-Komponenten sehr unterschiedlich ausfallen. Die *Art der Managementschnittstelle* kann in-band (ATM) oder aber out-of-band (z.B. Ethernet oder serieller Anschluß) sein. An den unterschiedlichen Schnittstellen können *verschiedene Managementprotokolle* unterstützt werden, wie beispielsweise standardisierte Managementprotokolle (z.B. SNMP oder CMIP), andere standardisierte Protokolle (z.B. Telnet) oder aber proprietäre Managementprotokolle (z.B. GSMP, das General Switch Management Protocol [NEHH⁺96]). Zusätzlich kann die *verfügbare Managementinformation* variieren. Es können standardisierte Einheiten an Managementinformation (z.B. ATM-MIB [AhTe94]) angeboten werden, wobei zumeist nur Teile der gesamten Einheit implementiert sind (d.h. nur Teile der MIB). Ferner werden von Herstellern Managementinformationen zur Verfügung gestellt, die speziell auf ihre jeweiligen Geräte zugeschnitten sind.

Anwendungen für das Management von einzelnen oder sogar mehreren Komponenten in einem ATM-Netzwerk können aufgrund der beschriebenen Heterogenität an Managementfunktionalität nur unter großem Aufwand erstellt werden. Derzeit verfügbare Managementanwendungen sind daher in der Regel auf genau eine spezielle ATM-Komponente zugeschnitten und höchstens noch für ähnliche Komponenten desselben Herstellers zu verwenden. Wünschenswert sind jedoch Anwendungen, die in einem heterogenen ATM-Umfeld für verschiedene ATM-Komponenten gleichzeitig eingesetzt werden können. Dabei kann es sich dann um einfache Ausprägungen handeln, die zu einem Zeitpunkt immer nur eine Komponente verwalten, oder aber um komplexere Anwendungen, die zur Erfüllung ihrer Aufgaben gleichzeitig auf viele ATM-Komponenten zugreifen müssen.

Ein weiterer Aspekt ist gerade auch für komplexere Anwendungen beim Management wichtig. Es gibt eine Vielzahl von grundlegenden Funktionen und Informationen, die in der Regel für mehr als nur eine Anwendung benötigt werden. Die in Kapitel 2 vorgestellte *hierarchische Managementarchitektur für ATM-Netzwerke* sieht hierfür die Management-Middleware vor, die als eine Art Zwischenebene unterstützende Dienste für das Management von ATM-Netzwerken bietet. Als eine beispielhafte Instanz der Management-Middleware wird dann in Kapitel 3 der *ATM-Netzwerkmonitor* präsentiert, der eine Topologie- und Dienstgütee Erfassung für ATM-basierte Netzwerke realisiert. Eine Evaluierung des ATM-Netzwerkmonitors in zwei verschiedenen Umgebungen liefert im abschließenden Kapitel 4 eine Einschätzung der praktischen Einsetzbarkeit der vorgestellten Ideen.

2 Hierarchische Managementarchitektur

Als Lösung für die Heterogenität im Managementbereich heutiger ATM-Komponenten wird in [Wilt97b, Wilt97a] eine Architektur präsentiert, die drei logische Abstraktionsebenen für das Management von ATM-basierten Netzwerken definiert (vgl. Abbildung 1):

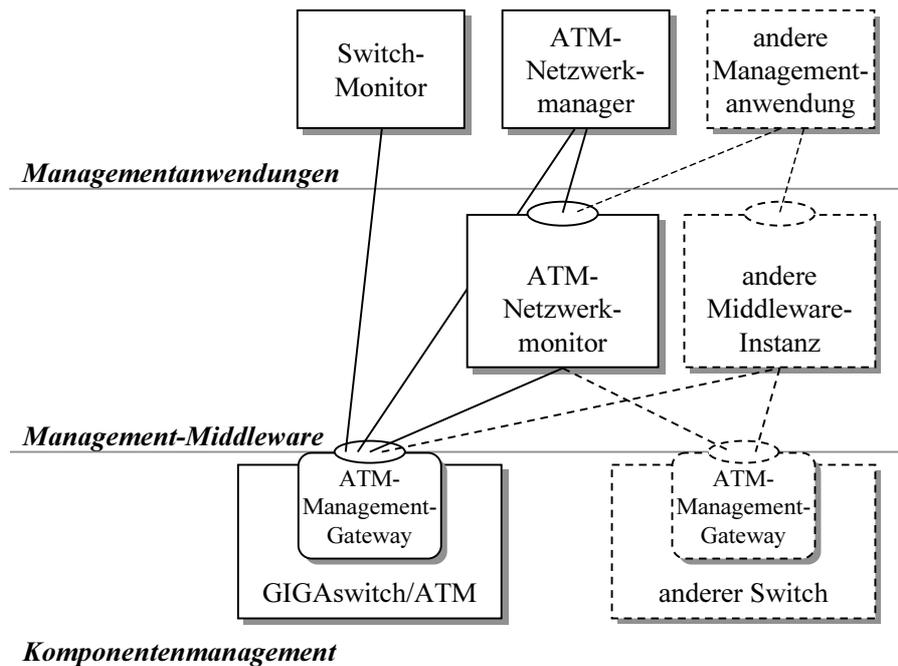


Abbildung 1: Hierarchische Managementarchitektur für ATM-Netzwerke

- Komponentenmanagement** – Die unterste Ebene, das Komponentenmanagement, beschreibt das Management einzelner Netzwerkkomponenten. Hier ist das Management durch die spezifischen Eigenschaften der einzelnen Geräte geprägt, die abhängig vom Hersteller und vom Stand der Produktentwicklung sind. Auf der Ebene des Komponentenmanagements ist eine einheitliche Managementschnittstelle anzustreben, die in ihrer ersten Version eine minimale Menge an grundsätzlich notwendiger Managementinformation anbieten muß. Die Integration der unterschiedlichen Managementfunktionalität bei verschiedenen ATM-Komponenten kann dabei mit Hilfe des ATM-Management-Gateways (AMG) [Ritt97] erreicht werden, das in Form eines Proxy-Managementagenten [CFSD90, HeAb93] die Managementkommunikation für die jeweilige ATM-Komponente umsetzt. Durch das ATM-Management-Gateway wird so für die übrigen Instanzen auf den höheren Abstraktionsebenen der Managementarchitektur ein wohldefinierter komponentenunabhängiger Zugriff auf Managementinformation ermöglicht. Als Managementprotokoll für den definierten Zugriff auf das Komponentenmanagement wird in der präsentierten Architektur das Simple Network Management Protocol (SNMP [CFSD90]) eingesetzt, da SNMP ein einfaches Protokoll darstellt, von sehr vielen Anwendungen im Bereich des Internets bereits genutzt wird und von derzeit verfügbaren ATM-Komponenten am häufigsten unterstützt wird.
- Management-Middleware** – Die nächsthöhere Abstraktionsebene, die Management-Middleware, realisiert unterstützende Dienste für das Management ATM-basierter Netzwerke. Diese Dienste sind komplexer als die einfache Managementfunktionalität des Komponentenmanagements, da sie sich in der Regel auf mehrere oder sogar alle Komponenten eines ATM-Netzwerks beziehen. Dienste der Management-Middleware decken beispielsweise die Bereiche Konfigurations-, Fehler- oder Leistungsmanagement für ATM-Netzwerke ab. Der wesentliche Vorteil der Management-Middleware besteht dabei in der Konzentration wichtiger unterstützender Dienste in einer oder wenigen zentralen Instanzen, die dann von

vielen Managementanwendungen gleichzeitig genutzt werden können. Somit müssen die unterschiedlichen Managementanwendungen die durch die Middleware angebotenen Managementinformationen nicht jeweils einzeln und für sich lokal zusammentragen und berechnen, sondern können hierzu den Dienst der Middleware in Anspruch nehmen. Zur Erbringung ihrer Dienste nutzen alle Instanzen der Management-Middleware die wohldefinierte Managementschnittstelle des Komponentenmanagements, die durch die Proxy-Managementagenten realisiert sein kann.

- *Managementanwendungen* – Die Managementanwendungen auf der dritten Abstraktionsebene stellen Endanwendungen für das Management ATM-basierter Netzwerke dar. Sie sind zumeist mit einer Oberfläche grafischer oder textueller Art ausgestattet, über die eine Interaktion mit einem menschlichen Benutzer erfolgt. Für die Umsetzung der von einem Benutzer geforderten Aktionen oder Aufgaben nehmen die Managementanwendungen sowohl die elementaren Dienste des Komponentenmanagements als auch die komplexeren Dienste der Middleware in Anspruch. Die durch das Komponentenmanagement erreichte Unabhängigkeit von den spezifischen Eigenschaften einzelner ATM-Komponenten macht es möglich, Anwendungen für viele ATM-Komponenten unterschiedlichen Typs zu entwerfen und einzusetzen. Hierbei lassen sich Anwendungen unterscheiden, die das Management einzelner ATM-Komponenten zum Ziel haben, und solche, die mehrere oder alle ATM-Komponenten eines Netzwerks überwachen bzw. verwalten. In gleicher Weise können auch kommerzielle Managementsysteme, die das Managementprotokoll SNMP unterstützen, als Managementanwendungen gemäß der präsentierten Architektur eingesetzt werden.

3 Der ATM-Netzwerkmonitor

Der ATM-Netzwerkmonitor (ANEMON) [Wilt97a, WiFo98] kann als Management-Middleware gemäß der in Kapitel 2 vorgestellten Managementarchitektur klassifiziert werden. In diesem Kapitel werden die Aufgaben des ATM-Netzwerkmonitors motiviert (Abschnitt 3.1), die Struktur des ANEMONS präsentiert (Abschnitt 3.2) sowie die Subsysteme zur Topologie- und Dienstgütere Erfassung (Abschnitte 3.3 und 3.4) im Detail vorgestellt.

3.1 Motivation und Aufgaben

Management-Middleware bietet Unterstützung beim Management ATM-basierter Netzwerke, indem sie Anwendungen von solchen Aufgaben entlastet, die für eine Vielzahl unterschiedlicher Anwendungen in gleicher oder ähnlicher Form interessant sein können und daher besser an zentraler Stelle durchgeführt werden sollten. Der ATM-Netzwerkmonitor deckt einige elementare Aufgaben ab, die als Unterstützung für das Management von ATM-Netzen einzustufen sind und in die Bereiche Leistungs- und Fehlermanagement eingeordnet werden können.

Die erste Aufgabe des ATM-Netzwerkmonitors, die Erfassung und Aufbereitung von Topologiedaten, kann als grundlegende Unterstützung für das Management von ATM-

Netzwerken charakterisiert werden. Viele sinnvolle Managementfunktionen oder -operationen basieren auf der Verfügbarkeit von solchen Topologieinformationen. Ein weiteres Aufgabenfeld des ATM-Netzwerkmonitors ist die Lokalisierung einzelner ATM-Verbindungen innerhalb der erfaßten Topologie. Die aufwendigste Aufgabe des ATM-Netzwerkmonitors besteht in der bedarfsgesteuerten Überwachung von Dienstgüteparametern für zuvor erfolgreich lokalisierte ATM-Verbindungen. Dabei wird für jeden Abschnitt der ATM-Verbindung innerhalb des ATM-Netzwerks die dort gegebene Dienstgüte ermittelt.

3.2 Struktur des ATM-Netzwerkmonitors

Die in Abschnitt 3.1 beschriebenen Aufgabenfelder des ANEMONs finden sich auch in seiner Struktur (vgl. Abbildung 2) wieder. Es sind zwei Module zu erkennen, die sich jeweils auf die Beschaffung und Aufbereitung einer Teilmenge der vom ATM-Netzwerkmonitor angebotenen Managementinformation konzentrieren. Das *Topologiemodul* realisiert dabei die Topologieerfassung und stellt die in der Topologiedatenbasis erfaßten Daten zur Verfügung. Das *Dienstgütemodul* beinhaltet die verbindungsbezogenen Daten in der QoS-Datenbasis. Hier wird die bedarfsgesteuerte Nachverfolgung einzelner ATM-Verbindungen und die Dienstgüteüberwachung für solche Verbindungen realisiert. Das Rahmenwerk des ATM-Netzwerkmonitors bildet ein SNMP-Agent, der mit Hilfe der AGENT++-Bibliothek [Fock97] erstellt worden ist. Die objektorientierte Implementierung besteht aus einer Protokollmaschine, einer Verteilertabelle und der Managementinstrumentierung. Die Protokollmaschine empfängt an den ANEMON gerichtete SNMP-Anfragen und dekodiert sie. Ebenso werden Antworten kodiert und versendet. Die Verteilertabelle übernimmt die Weiterleitung der Anfragen innerhalb der internen objektorientierten Modellierung der MIB an das Objekt, das für die angefragte Managementinformation verantwortlich ist. Die Managementinstrumentierung ist auf die einzelnen internen Objekte verteilt und sorgt dort dann für die eigentliche Beschaffung der Managementinformation. Im Falle des ANEMONs ist hierzu hauptsächlich eine SNMP-Kommunikation mit den ATM-Komponenten notwendig, weshalb der ATM-Netzwerkmonitor auch die Implementierung eines SNMP-Managers beinhalten muß.

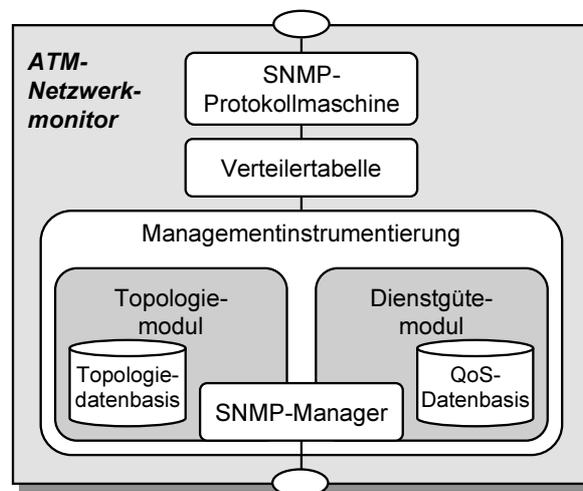


Abbildung 2: Struktur des ATM-Netzwerkmonitors

3.3 Topologieerfassung in ATM-Netzwerken

Die Erfassung von Topologiedaten stellt eine grundlegende Aufgabe für das Management von ATM-Netzwerken dar, die für viele unterschiedliche Anwendungen von Nutzen sein kann. Beispielsweise ist es sinnvoll, daß eine Anwendung für das Management eines ATM-Netzwerks eine grafische Darstellung des Netzes erzeugen kann, um basierend auf dieser Darstellung weitere Funktionen anbieten zu können [Chri97]. Folglich sollte eine Topologieerfassung als Management-Middleware realisiert werden, damit sie als unterstützender Dienst einer Vielzahl von Anwendungen gleichzeitig zur Verfügung stehen kann.

Der ATM-Netzwerkmonitor beinhaltet ein Topologiemodul (vgl. Abbildung 2), das eine Topologieerfassung für ATM-Netzwerke realisiert und die berechneten Topologiedaten Managementanwendungen als Management Information Base (MIB) modelliert über SNMP zur Verfügung stellt. Die Erfassung der Topologieinformation durch einen speziellen Erfassungsalgorithmus basiert auf der Nachbarschaftsinformation, die ATM-Komponenten in-band über die ILMI-Schnittstelle (Integrated Local Management Interface [ATM-96]) austauschen und danach in der ATM-MIB [AhTe94] zur Verfügung stellen. Folglich muß der ATM-Netzwerkmonitor hierzu jede einzelne ATM-Komponente per Management ansprechen und die dort lokal vorhandenen Nachbarschaftsinformationen auslesen. Der gesamte Erfassungsalgorithmus läuft dabei als Tiefensuche ab, die bei einem einstellbaren Startpunkt beginnt und bis zu einer ebenfalls einstellbaren maximalen Suchtiefe fortgesetzt wird. Eine spezielle Erweiterung sorgt bei dem Erfassungsalgorithmus dafür, daß ein Einsatz des ANEMONS in einem hierarchischen Umfeld möglich wird. Hierzu kann der Algorithmus bei seiner Suche gefundene Topologieinformation eines anderen ANEMONS in die eigenen Topologiedaten einfügen. Auf diese Art ist ein Szenario denkbar, bei dem einige ANEMONS in ihrem lokal beschränkten Bereich eine Topologieerfassung durchführen und ein weiterer ANEMON die globale Topologie berechnet, indem er die Ergebnisse der anderen ANEMONS mit einbezieht und in Zusammenhang bringt.

3.4 Dienstgüteüberwachung ausgewählter ATM-Verbindungen

Eine weitere sinnvolle Aufgabe für Management-Middleware ist die Identifizierung und Überwachung einzelner ATM-Verbindungen innerhalb eines ATM-Netzwerks. Diese Aufgabe wird von dem Dienstgütemodul des ATM-Netzwerkmonitors (vgl. Abbildung 2) erbracht. Im ersten Schritt wird eine ATM-Verbindung auf Anfrage einer Managementanwendung identifiziert, indem sie von einem Startpunkt aus innerhalb der Netzwerktopologie lokalisiert wird. Der Startpunkt muß dabei von der Managementanwendung in der Anfrage angegeben werden. Dann werden alle an der Verbindung beteiligten ATM-Komponenten identifiziert und die entsprechenden Verbindungsabschnitte ermittelt. Folglich ist die Information in der Topologiedatenbasis des ANEMONS für eine erfolgreiche Lokalisierung von Verbindungen eine notwendige Voraussetzung. Im zweiten Schritt können für zuvor erfolgreich lokalisierte ATM-Verbindungen Dienstgüteparametern überwacht werden. Dabei wird für jeden Abschnitt der ATM-Verbindung innerhalb des ATM-Netzwerks die dort gegebene Dienstgüte ermittelt. Nur so kann bei einer Beeinträchtigung der Dienstgüte genau die verursachende Komponente innerhalb des Netzwerks bestimmt werden. Eine detaillierte Überwachung der Dienstgüte von

ATM-Verbindungen macht aufgrund des erheblichen Aufwandes nur im Einzelfall Sinn, wenn an anderer Stelle ein Problem erkannt wurde bzw. vermutet wird. Folglich wird sie vom ATM-Netzwerkmonitor auch nur auf Anfrage und für einen einstellbaren Zeitraum durchgeführt. Zur Erfassung sämtlicher Dienstgüteparametern, die für einzelne ATM-Verbindungen sinnvoll erscheinen, sind bei heutigen ATM-Komponenten jedoch noch Erweiterungen zu integrieren, da beispielsweise für die Messung zeitbezogener Werte wie Verzögerung und Jitter die Integration von Zeitstempeln in den regulären Datenfluß der Verbindungen nötig ist. Hierfür bietet zum einen das bei der ATM-Technologie bereits vorhandene OAM-Konzept (Operation and Maintenance [ITU-95]) Möglichkeiten zur Umsetzung, indem spezielle Managementzellen auf der Ebene von ATM-Verbindungen (OAM F5 Level) definiert und eingesetzt werden. Zum anderen empfiehlt sich auch der Einsatz eines neuen SSCS-Protokolls (Service Specific Convergence Sublayer) in der AAL-Schicht von zu überwachenden Verbindungen. Zu diesem Zweck wurde für AAL5-basierte Verbindungen in [WiSc97] das QM-SSCS-Protokoll vorgeschlagen, das pro Benutzerdateneinheit (AAL5-SDU) einige zusätzliche Werte zur Unterstützung der Dienstgüteüberwachung in den Datenstrom einfügt. Die Felder sind dabei so angeordnet, daß alle wichtigen Informationen nach der Segmentierung der AAL5-Dateneinheit in nur einer ATM-Zelle lokalisiert sind. Da hierfür die letzte Zelle der AAL5-Dateneinheit gewählt wurde, die bei AAL5 eine spezielle Kennzeichnung im Zellkopf mitführt, ist eine Erkennung und Auswertung solcher Zellen mit Dienstgüteinformation in ATM-Zwischensysteme recht einfach zu integrieren. Die basierend auf den hier vorgeschlagenen Erweiterungen für ATM-Zwischensysteme vom ATM-Netzwerkmonitor erfaßbaren Dienstgüteparameter einzelner ATM-Verbindungen werden parallel zu den Topologiedaten in einer weiteren MIB den Managementanwendungen zur Verfügung gestellt.

4 Evaluierung und Bewertung

Die Evaluierung des ATM-Netzwerkmonitors erfolgte auf zwei unabhängige Arten. Zum einen wurden Tests in einem lokalen ATM-Testfeld durchgeführt. Dieses Testfeld umfaßte einen ATM-Switch von Digital Equipment, einige ATM-Switches des Herstellers Netbridge und eine Vielzahl an Endsystemen des Typs Alpha von Digital Equipment (vgl. Abbildung 3). Aufgrund der beschränkten Größe und Heterogenität des verfügbaren Testfeldes wurde parallel eine virtuelle Testumgebung entwickelt, in der die erstellten Implementierungen evaluiert werden konnten. Die virtuelle Testumgebung basiert auf einem speziellen Managementagenten, der anstelle des beim Komponentenmanagement vorgesehenen ATM-Management-Gateways eingesetzt wird. Dieser Managementagent ist in der Lage, durch interne Daten das Vorhandensein einer realen ATM-Komponente zu simulieren. Durch eine speziell entwickelte Steueroberfläche für diese Managementagenten konnte erreicht werden, daß viele parallel eingesetzte Agenten gleichzeitig ein großes virtuelles ATM-Netzwerk simulieren.

In beiden Testumgebungen konnte der implementierte ATM-Netzwerkmonitor seine Funktionalität und Leistungsfähigkeit unter Beweis stellen. Da das Dienstgütemodul des ANEMONS in der aktuellen Version noch nicht implementiert ist, bezogen sich die durchgeführten Tests bis jetzt nur auf die Topologieerfassung. Hier konnte jedoch nicht nur die allgemeine Leistungsfähigkeit gezeigt werden, sondern in einer speziell

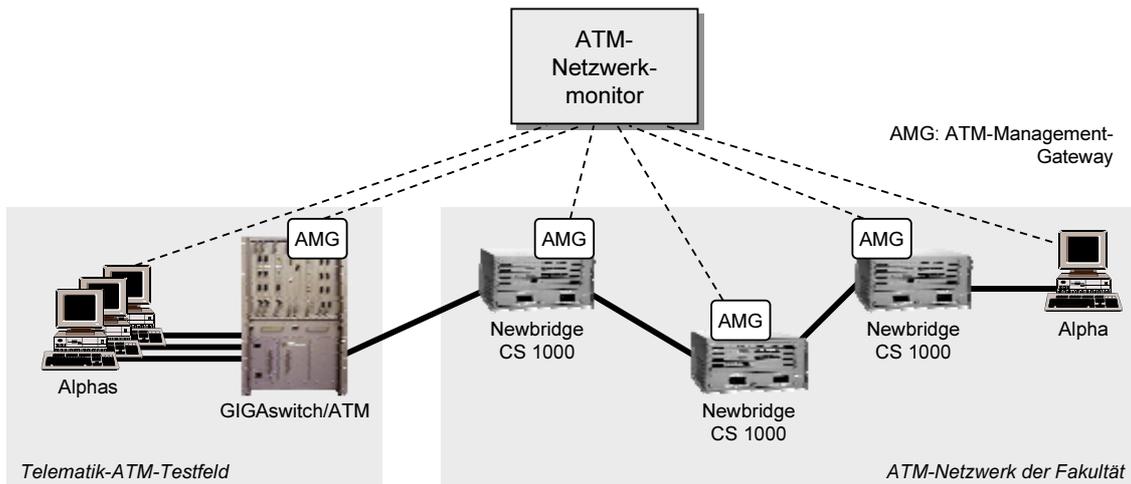


Abbildung 3: Testfeld für den ATM-Netzwerkmonitor

aufgesetzten virtuellen Umgebung auch die Vorteile der hierarchischen Nutzung von ANEMON-Instanzen.

5 Zusammenfassung

In diesem Beitrag wurde der ATM-Netzwerkmonitor vorgestellt, der eine Topologieerfassung für ATM-Netzwerke sowie eine bedarfsgesteuerte Lokalisierung und gezielte Dienstgüteüberwachung für einzelne ATM-Verbindungen realisiert. Da es sich bei diesen Aufgaben um unterstützende Dienste für das Management von ATM-basierten Netzwerken handelt, kann der ATM-Netzwerkmonitor als Management-Middleware gemäß der ebenfalls in diesem Beitrag vorgestellten hierarchischen Managementarchitektur für ATM-Netzwerke klassifiziert werden. Er setzt auf die einheitliche Managementchnittstelle des Komponentenmanagements auf und bietet seine Dienste den Managementanwendungen auf der obersten Ebene an.

Die Topologieerfassung des ATM-Netzwerkmonitors basiert auf Nachbarschaftsinformationen, die ATM-Komponenten automatisch in-band über die ILMI-Schnittstelle untereinander austauschen. Der Erfassungsalgorithmus sammelt diese im Netzwerk verteilten Daten in Form einer Tiefensuche ein, die bei einem vorgegebenen Startpunkt beginnt. Dabei kann der Algorithmus gefundene Topologiedaten eines anderen ATM-Netzwerkmonitors miteinbeziehen, was den Einsatz in einem hierarchischen Umfeld möglich macht. Während die Topologieerfassung bereits implementiert ist und in den vorgestellten Tests erfolgreich evaluiert werden konnte, ist die Lokalisierung und Dienstgütererfassung für einzelne ATM-Verbindungen derzeit noch in der Entwurfsphase. Hier sind für eine vollständige Umsetzung der präsentierten Ideen auch noch Erweiterungen in ATM-Komponenten (vor allem Switches) zu integrieren. Dieser Teilbereich stellt somit neben weiteren Tests den Schwerpunkt zukünftiger Arbeiten dar.

Literatur

- [AhTe94] M. Ahmed und K. Tesink. Definition of Managed Objects for ATM Management Version 8.0 using SMIV2. RFC 1695, August 1994.
- [ATM-96] The ATM-Forum. Integrated Local Management Interface (ILMI) Specification Version 4.0. af-ilmi-0065.000, September 1996.
- [CFSD90] J. Case, M. Fedor, M. Schoffstall und J. Davin. A Simple Network Management Protocol (SNMP). RFC 1157, Mai 1990.
- [Chri97] Jörg Christ. Web-basiertes Management von ATM-Netzwerken. Diplomarbeit, Institut für Telematik, Universität Karlsruhe, Dezember 1997.
- [Fock97] Frank Fock. Entwicklung von Managementkomponenten zur Topologieerfassung in ATM-Netzwerken. Diplomarbeit, Institut für Telematik, Universität Karlsruhe, Dezember 1997.
- [HeAb93] H.-G. Hegering und S. Abeck. *Integriertes Netz- und Systemmanagement*. Addison-Wesley. 1993.
- [ITU-95] ITU-T. B-ISDN Operation and Maintenance – Principles and Functions. ITU-T Recommendation I.610, November 1995.
- [NEHH⁺96] P. Newman, W. Edwards, R. Hinden, E. Hoffman, F. Ching Liaw, T. Lyon und G. Minshall. Ipsilon's General Switch Management Protocol Specification Version 1.1. RFC 1987, August 1996.
- [Ritt97] Hartmut Ritter. Entwurf und Implementierung einer Monitoring-Schnittstelle für ATM-Netzwerke. Diplomarbeit, Institut für Telematik, Universität Karlsruhe, Mai 1997.
- [WiFo98] H. R. Wiltfang und F. Fock. Topology Discovery for ATM Networks. In *Proceedings of the 9th IEEE Workshop on Local and Metropolitan Area Networks*, Banff, Kanada, 17. - 20. Mai 1998.
- [Wilt97a] Hajo R. Wiltfang. An Approach to QoS Monitoring in ATM Networks. In D.W. Faulkner und A.L. Harmer (Hrsg.), *Proceedings of the European Conference on Networks and Optical Communication – Part II: Core and ATM Networks*, Antwerpen, Belgien, 17. - 20. Juni 1997. IOS Press, Amsterdam, S. 328–335.
- [Wilt97b] Hajo R. Wiltfang. Management und Monitoring von ATM-Netzen. *Praxis der Informationsverarbeitung und Kommunikation (PIK)* 20(2), Juni 1997, S. 68–75.
- [WiSc97] Hajo R. Wiltfang und Claudia Schmidt. QoS Monitoring for ATM-based Networks. In R. Boutaba und A. Hafid (Hrsg.), *Management of Multimedia Networks and Services*, Montreal, Kanada, 8. - 10. Juli 1997. Chapman & Hall, London, S. 55–66.

Teil II

Projekte

Mercado - ein Multimedia Pilot

Michael Beigl

Kurzfassung

Das EU Projekt Mercado zeigt als Multimedia Pilot den Einsatz von Web-Technologie auf einem Kiosksystem. Im Vordergrund steht dabei die Abwicklung von kostenpflichtigen Transaktionen z.B. der Kauf von Tickets via GeldKarte. Darüber hinaus soll das System prinzipiell die Möglichkeit bieten, per Internet auf die Anwendungen des Kiosksystems zugreifen zu können.

1 Einleitung

Kiosksysteme stellt man sich gemeinhin als Einzelsystem im Baumarkt oder auf öffentlichen Plätzen vor, die einen sehr eingeschränkten Zweck verfolgen und dabei eher "standalone" sind. Mercado soll als EU Multimedia Pilot zeigen, daß in bisherige Strukturen integrierte und vernetzte Kiosksysteme bessere Bedingungen sowohl für die Betreiber der Kiosksysteme als auch für die Benutzer solcher Systeme bieten. An diesem Projekt beteiligt sind deshalb neben den Firmen und Institutionen Digital/CEC Karlsruhe, Tellux GmbH, Colibri GmbH und TecO, welche für die technische Realisierung zuständig sind, die Städte Nottingham, Oulu, Athen und Karlsruhe zur Durchführung des Piloten.

2 Mercado System

Ziel des Mercado Systems war es nicht, ein weiteres Kiosksystem zu implementieren, sondern ein System zu entwickeln, mit dem sich Inhalte verschiedenster Ausprägung darstellen lassen. Dabei sollte der gleiche Inhalt ebenfalls im Internet angeboten werden können. Die Basis stellt dabei die Web-Technologie dar, wobei Java und ActiveX Elemente benutzt werden, um zentrale Funktionalitäten des Kiosks, wie beispielsweise kostenpflichtige Dienste und Transaktionen, zu ermöglichen.

Mercado besteht aus drei Komponenten: Dem (Multimedia-)Kiosk, dem Service Center und dem Service Provider. Auf dem Kiosk werden dabei die Inhalte dargestellt und die Transaktionen getätigt, z.B. ein Ticket gekauft. Die derzeitige Version des Kiosks besteht aus einem Digital-PC mit Touchscreen, entsprechendem Drucker (je nach Verwendungszweck) und einem Kioskgehäuse. Der Inhalt des Kiosks wird von einem speziell entwickelten Webbrowser gesteuert, welcher gleichzeitig für die Abwicklung der

Transaktionen zuständig ist. Der Service Center stellt den Verbindungspunkt der Kioske dar. Dort werden die Transaktionen gesammelt und weitergeleitet. Auch befindet sich dort der Originalinhalt, der auf den Kiosken angeboten wird. Die Service Provider sind für die Erstellung des Inhalts und die eventuelle Auslieferung der Waren zuständig.

3 Stand des Projekts und Aufgaben des TecO

Mercado befindet sich in der zweiten Phase der Entwicklung, d.h. erste Prototypen sind schon im Einsatz oder stehen kurz davor; darunter befindet sich das Stadtinformationssystem im Rathaus Karlsruhe, der Verkauf von Offerta Tickets bei der letzten Offerta und der Einsatz von Mercado Kiosken zum Verkauf von persönlichen Audio-CD's, wie er auf der diesjährigen CeBIT gezeigt wird.

Aufgaben des TecO in diesem Projekt betreffen vor allem die Client-Seite, also den Kiosk. Das TecO entwickelt den MercadoBrowser mit Hilfe von Microsoft's ActiveX Technologie. Mit Hilfe dieses Browsers lassen sich Funktionen wie Bezahlen via Geld-Karte, aber prinzipiell auch andere Bezahlmöglichkeiten sicher abwickeln. Außerdem wurde von uns eine virtuelle Tastatur zur Eingabe von Daten über den Bildschirm entwickelt. Ein weiterer Schwerpunkt ist die Abwicklung des Geld- und Warenflusses auf der Seite des Service Centers.

Im weiteren Verlauf des Projektes wird sich der Schwerpunkt zunehmend auf die Durchführung des Piloten sowie die Auswertung der Ergebnisse des Piloten verlagern. Aufgrund der Vielfalt der Einsatzgebiete vom Stadtinformationszentrum bis hin zum Bankterminal, werden sich viele Fragestellungen ergeben, deren Lösungen für zukünftige Kiosksysteme wichtig sein wird.

Mobilitätsdienste in Zusammenarbeit mit der Daimler-Benz AG

Martin Gaedke

Kurzfassung

MobilBot ist ein Projekt der Daimler-Benz Forschung, das Systeme zur intelligenten Reiseunterstützung untersucht. Im Rahmen dieses Projektes wurde am TecO ein webbasiertes System entwickelt, das unter Berücksichtigung personenbezogener und gelernter Daten den Zugriff auf verschiedene Mobilitätsdienste ermöglicht, um multimodale Reisenketten vorzuschlagen oder zu buchen.

1 Einleitung

Informationsdienstleistungen im Umfeld der Mobilität sind heutzutage ein unentbehrlicher Helfer geworden, um die Vielfalt an erhältlichen Informationen noch verarbeiten zu können. Der Ausgangspunkt ist hierbei für die folgenden Überlegungen die These „Information ist der Schlüssel zur Verbesserung der Mobilität“. Das Projekt MobilBot dient dazu, zukunftsgerichtete Ideen und entsprechende Trends für neue Informationsdienstleistungen im Reisebereich zu ergründen und aufgrund der gesammelten Ergebnisse, Fakten und Ideen den Dienst zu skizzieren. Ein wichtiges Ergebnis ist die Beschreibung von möglichen, zukünftigen Formen der Mobilitätsplanung und die entsprechende Einbettung in Szenarien. Die Szenarien sollen realitätsnah genug sein, daß die Bezüge zur heutigen Mobilität und den damit verbundenen Problemen vorhanden sind. Trotzdem sollen langfristige Visionen einbezogen werden können. In der heute sehr turbulenten Entwicklung unterschiedlichster Informationssysteme im Bereich Tourismus und Reise trägt diese offene Perspektive wesentlich dazu bei, bei der Entwicklung von MobilBot möglichst große Offenheit und Flexibilität zu erzielen.

Ziel des von TecO und Daimler-Benz Research durchgeführten Projektes war es, ein System auf Webbasis zu entwickeln, das als Basis für die Implementierung oder Anbindung neuer Szenarien dient. Als ein Szenario sollte auf dieser Basisplattform ein anwendungsnaher Prototyp für das Kundensegment „Geschäftsreisende“ entwickelt werden [Gaed97].

2 Projektübersicht

In der ersten Projektphase wurde am TecO das MobilBot-Basissystem entwickelt. Das System basiert auf einer Multi-Tier Architektur und besteht aus einer Menge von

Diensten, die durch verschiedene Anwendungen und Datenbanken geboten werden, sowie aus einer Webanwendung, die den Zugang zum System ermöglicht. Für den Entwurf der Webanwendung und ihre spätere Wartung wurden Methoden, die im Rahmen des Projektes WebComposition entwickelt wurden, angewandt [GeWG97]. Wesentliche Anforderungen, die das Basissystem erfüllt, sind Dienste zur automatischen Gewinnung von Benutzerprofilen und zur automatischen Interaktion des Systems mit dem Benutzer aufgrund von beobachteten Ereignissen. Das Zusammenspiel von Benutzerprofilen und Interaktion [ShMa95] kann z.B. dazu genutzt werden, daß das System festgestellte Veränderungen im Reiseverhalten interaktiv bei dem Benutzer oder der Benutzerin hinterfragt, um die festgestellten Wünsche im System als Reisepräferenz zu speichern. Im weiteren Projektverlauf wurde das System durch eine offene Architektur für Mobilitätsdienste erweitert. Diese Architektur wird dazu genutzt Mobilitätsdienste wie z.B. Flugbuchungs- oder Hotelreservierungssysteme von dem MobilBot-Metarouter anzusprechen, um aus den gewonnenen Daten multimodale Reiserouten, also Reiserouten mit verschiedenen Verkehrsträgern, zu ermitteln. Darüber hinaus bietet das System auf Benutzerseite, die Möglichkeit der Integration von Fremdsoftware. So kann man bei der Planung einer Reise beispielsweise auf die Daten aus der persönlichen Kontaktdatenbank des Programms Outlook der Firma Microsoft zugreifen oder eine gebuchte Reise im Terminkalender eintragen lassen. Die folgende Abbildung zeigt einen kleinen Ausschnitt der Gesamtarchitektur.

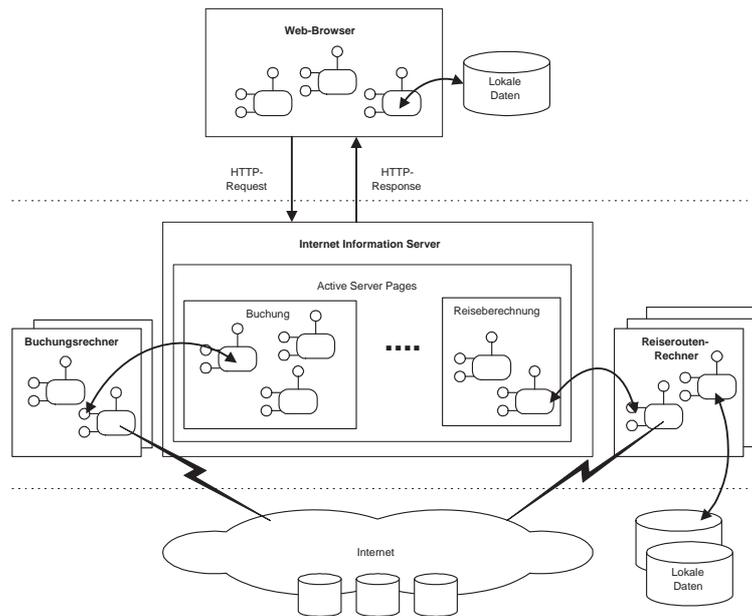


Abbildung 1: Die MobilBot Systemarchitektur

Während einer Reise besteht dann die Möglichkeit sich mit einem PDA im System anzumelden, um z.B. eine Reise umzubuchen oder zu stornieren.

3 Ausblick

Nach Abschluß der Arbeiten wurde das TecO gebeten, auf der offenen Architektur für Mobilitätsdienste einen existierenden Dienst der Daimler-Benz zu integrieren. Die bei

dem Projekt entstandenen Erkenntnisse über die Möglichkeiten von Benutzerprofilen werden im Hinblick auf ihre Anwendbarkeit in WebComposition weiter untersucht.

Literatur

- [Gaed97] Martin Gaedke. ActiveX: Componentware im Internet. *Theorie und Praxis der Wirtschaftsinformatik* 34(197), September 1997.
- [GeWG97] Hans-W. Gellersen, Robert Wicke und Martin Gaedke. WebComposition: An Object-Oriented Support System for the Web Engineering Lifecycle. In *Proc. of the 6th International World-Wide Web Conference (WWW6)*, Santa Clara, April 7–12 1997.
- [ShMa95] Upendra Shardanand und Pattie Maes. Social Information Filtering: Algorithms for Automating 'Word of Mouth'. In *Proceedings of the CHI-95 Conference*, ACM Press, Denver, Mai 1995.

Victor II

Martin Gaedke und Albrecht Schmidt

Kurzfassung

Das im folgenden beschriebene Projekt beschäftigt sich mit der Automatisierung von Serviceangeboten. Dabei werden die bereits vorhandenen Serviceangebote im HP-Intranet im Rahmen des Projekts analysiert. Basierend auf dieser Analyse und unter Verwendung neuer Entwicklungsmethoden für Websoftware soll die Erstellung und Wartung der Angebote optimiert werden.

1 Einleitung

Die Automatisierung von Serviceangeboten und deren Integration ins Web/Intranet birgt ein enormes Rationalisierungspotential in sich. HP versucht derzeit möglichst viele seiner internen Serviceangebote ins Intranet zu integrieren. Man erhofft sich daraus eine einfachere, schnellere und kostengünstigere Abwicklung dieser Vorgänge. Die Webintegration soll darauf ausgerichtet sein, daß die Erstellung neuer Services, die Pflege und Wartung sowie die Erweiterung bereits vorhandener Serviceangebote möglichst einfach ist. In diesem Kooperationsprojekt bringt das TecO sein Wissen und seine Kompetenz aus den Bereichen Web, Internet und Intranet ein.

2 Projektübersicht

2.1 Projektpartner

Am Projekt sind das HP-IT Service Center in Böblingen und das Telecooperation Office an der Universität Karlsruhe (TecO) beteiligt.

2.2 MetaWebService

In der ersten Projektphase wurde ein MetaWebService entwickelt, siehe auch 4. Die Zielsetzung hierbei war es, ein Werkzeug zu schaffen, das es erlaubt Webservices ohne Programmierkenntnisse zu erstellen. Eine weitere Randbedingung ist, daß neue Webservices von beliebigen HP-Niederlassungen in Europa aus erstellt werden können müssen. Um ein Höchstmaß an Flexibilität und Plattformunabhängigkeit zu erreichen, wurde das Frontend für den MetaWebService in HTML implementiert. Die weiteren Komponenten wurden in Perl geschrieben.

2.3 Requirements Analyse und System Design

In der nächsten Phase wird nun eine umfassende Analyse der bereits vorhandenen Webservices vorgenommen. Außerdem wird eine Analyse der Services für die eine Webintegration geplant ist, durchgeführt. Das Ziel dieser Analyse ist es die Obermenge von Komponenten zu spezifizieren, die für den Aufbau von Webservices im HP-Intranet verwendet wird. Insbesondere sollen Probleme von multilingualen Serviceangeboten, Abhängigkeiten zwischen Angeboten, Integration von Datenbanken, Unterstützung von Fremdsoftware und Einsatzmöglichkeiten von Benutzer- und Businessprofilen untersucht werden. Als Ergebnis wird ein System Design, das die gegebenen Anforderungen berücksichtigt und darüber hinaus Konzepte für das Webservice Management enthält, erstellt.

2.4 Umsetzung und Aufbau der Service Infrastruktur

Das durch die Analyse der Anforderungen entstandene System soll mit adaptierten Konzepten aus WebComposition [GeWG97] für den Praxisbetrieb umgesetzt werden. Diese Implementierung wird von Mitarbeitern am TecO in enger Kooperation mit dem HP-IT Service Center durchgeführt werden.

Literatur

- [GeWG97] Hans-W. Gellersen, Robert Wicke und Martin Gaedke. WebComposition: An Object-Oriented Support System for the Web Engineering Lifecycle. In *Proc. of the 6th International World-Wide Web Conference (WWW6)*, Santa Clara, April 7–12 1997.

Sonderforschungsbereich 346 – Die Desktop-Metapher als Integrationsansatz in der rechnergestützten Fertigung

Arnd G. Grosse, Jörn Hartroth und Rainer Ruggaber

Kurzfassung

Der Sonderforschungsbereich 346 verfolgt die Zielsetzung, Methoden der Rechnerintegration für Konstruktion und Fertigung im Maschinenbau zu untersuchen. Das Institut für Telematik arbeitet in diesem Rahmen an der Entwicklung einer kommunikationstechnischen Infrastruktur. Zur Unterstützung dynamischer Organisationsformen und Abläufe wird dabei konkret die Befriedigung spontan entstehenden Informationsbedarfs, die optimierte Vermittlung von Diensten und die optimierte Koordination unabhängig entwickelter lokaler Abläufe untersucht. Der zweite Schwerpunkt liegt in der Entwicklung einer einheitlichen Benutzerschnittstelle mit verteiltem Zugriff auf unterliegende Daten und Werkzeuge zur Schaffung einer zweiten Integrationsebene oberhalb der bereits bestehenden Integration auf Datenrepräsentationsebene.

1 Einleitung

Der Sonderforschungsbereich 346 verfolgt die Zielsetzung, Methoden der Rechnerintegration für Konstruktion und Fertigung im Maschinenbau zu untersuchen. Das Institut für Telematik arbeitet in diesem Rahmen an der Entwicklung einer kommunikationstechnischen Infrastruktur. In den ersten beiden Projektphasen wurde im SFB 346 die Entwicklung und nachfolgende Konsolidierung des integrierten Produkt- und Produktionsmodells PPM als konzeptionelle Basis der informationstechnischen Erfassung aller relevanten Produktionsabläufe von der Auftragsakquisition über Erzeugnisentwicklung und -konstruktion bis hin zur Fertigungsplanung und -durchführung betrieben. In der derzeitigen dritten Projektphase soll in den Jahren 1996 bis 1999 dieses Modell in Richtung dynamischer Organisationsformen und Abläufe erweitert werden. Im Institut für Telematik wird dabei die Unterstützung dynamisch veränderlicher Anwendungsbedingungen auf Ebene der Verteilungsinfrastruktur betrachtet. Konkret verfolgte Ansätze sind hier die Befriedigung spontan entstehenden Informationsbedarfs durch die Verwendung von Codemobilität für Funktionseinheiten [GrHK96, HaKG97], die optimierte Vermittlung von Diensten auf Basis einer Modellierung dynamischer Informationsbeziehungen [GrKH97] und die Verwendung von Verteilungsmechanismen zur optimierten Koordination unabhängig entwickelter lokaler Abläufe.

Als zusätzliche Entwicklungsrichtung des Sonderforschungsbereichs ist innerhalb dieser Projektphase in die Planung aufgenommen worden, die bisher auf der Ebene des

unterliegenden Datenmodells im PPM erfolgte Integration der verschiedenen Anwendungsabläufe um eine zweite Integrationsschiene auf Ebene der graphischen Benutzeroberfläche zu ergänzen. Zentraler Gedanke ist die Verwendung einer *Desktop-Metapher*, die die Steuerung aller im Rahmen des SFB 346 entstandenen Software-Werkzeuge in eine einheitlich gestaltete, individuell konfigurierbare Benutzeroberfläche, das *SFB346-Desktop*, zusammenfaßt. Die einheitliche Oberfläche soll auch den Zugriff auf Weiterentwicklungen des PPM bieten, hier insbesondere die geplante Einteilung des ehemals monolithischen PPMs in unterschiedliche Sichten entsprechend den verschiedenen Phasen der Produktentstehung.

Da das Konzept des Desktops innerhalb der aktuellen Projektphase entstanden ist und einen wesentlichen Einfluß auf die Orientierung des gesamten Sonderforschungsbereichs nimmt, soll es in diesem Beitrag schwerpunktmäßig behandelt werden.

2 Das SFB346-Desktop

Als Folge der bislang bislang nur auf Ebene des unterliegenden Datenmodells erfolgten Integration der verschiedenen Produktionsphasen entstanden die Anwendungswerkzeuge in den verschiedenen Teilprojekten des SFBs im wesentlichen getrennt. Dies führte zu einer Menge größtenteils monolithischer Anwendungen mit jeweils eigenem Erscheinungsbild, die überdies in verschiedenen Sprachen (C++, TCL, Java) auf unterschiedlichen Systemplattformen (Windows-PCs, Workstations mit Digital Unix und Solaris) vorlagen. Diese Struktur erwies sich als zu unflexibel für die weiteren Vorhaben des SFB.

Für eine Umstrukturierung der Softwarearchitektur im Rahmen der Zielsetzung einer Dynamisierung wurden die folgenden Verbesserungsmöglichkeiten identifiziert:

- Verwendung flexibler Komponenten anstelle der monolithischen Applikationen, um eine bessere Konfigurierbarkeit zu erreichen und die Möglichkeit zur Wiederverwendung von Funktionen zu schaffen.
- Schaffung einer einheitlichen Benutzerschnittstelle, um eine Integration verschiedener Phasen der Produktentstehung auch in der Benutzersicht zu erreichen.
- Individuelle und flexible Arbeitsplatzgestaltung nach der bekannten Desktop-Metapher fensterorientierter Benutzeroberflächen.
- Trennung von Präsentation und Anwendungslogik, um verschiedene Systemplattformen unter einem einheitlichen Desktop zu unterstützen.

Da das Konzept einer individuell konfigurierbaren einheitlichen Präsentationform den Kern des neuen Architekturkonzepts ausmacht, wurde für die Umgestaltung die Bezeichnung *SFB346-Desktop-Architektur* gewählt.

3 Entwurfsrahmenbedingungen

Die Festlegung des konkreten Systementwurfs für das SFB346-Desktop wurde unter Berücksichtigung der drei wesentlichen geforderten Rahmenbedingungen der Lokations-, Sprach- und Plattformunabhängigkeit getroffen.

Lokationsunabhängigkeit Im Hinblick auf die Unterstützung dezentralisierter und dynamischer Organisationsformen sollte ein Zugriff auf das SFB346-Desktop auch außerhalb eines lokalen Netzkontextes möglich sein. Darüberhinaus soll die Zugriffsmöglichkeit auch unabhängig davon sein, ob sich der Anwender innerhalb oder außerhalb der organisatorischen Domäne des Unternehmens befindet. Eine attraktive Plattform für diese Zugriffsart bildet die Kombination eines WWW-Browsers mit der Programmiersprache Java, die die Verwendung spezifischer Anwendungen ohne spezielle Voraussetzungen auf Nutzerseite erlaubt.

Sprachunabhängigkeit Da bei der Entwicklung der Anwendungswerkzeuge verschiedene Programmiersprachen zum Einsatz gekommen sind, muß zumindest für die die Anwendungslogik enthaltenden Applikationsmodule eine sprachunabhängige Anbindung geschaffen werden. Transparente Kommunikation zwischen Programmen in verschiedenen Sprachen kann über den CORBA-Ansatz erreicht werden.

Plattformunabhängigkeit Die implementierungstechnische Integration der im SFB entstandenen Software innerhalb der Desktop-Architektur muß auf den unterschiedlichen im Einsatz befindlichen Systemplattformen lauffähig sein. Ein Übergang auf eine einheitliche Plattform ist wegen Abhängigkeiten bei einigen Anwendungswerkzeugen nicht möglich. Mit der Programmiersprache Java existiert eine auf Portabilität ausgelegte Sprachumgebung, die auf den relevanten Systemplattformen verfügbar ist.

Der Rahmenentwurf des SFB-Desktops sieht demnach vor, die präsentationsbezogenen Komponenten (Desktop und Views) durch eine Implementierung in Java für eine Ausführung im WWW-Browser geeignet zu gestalten. Die Kommunikation mit den funktionsbezogenen Komponenten und der GOM-Datenbank erfolgt über eine ORB-Implementierung.

3.1 Architekturkomponenten

Das in dieser Projektphase neu entwickelte Architekturkonzept enthält folgende Komponenten, die in Abbildung 1 veranschaulicht werden:

Desktop Das Desktop bildet die umschließende Klammer der Benutzerschnittstelle sowohl in der graphischen Erscheinung als auch in organisatorischer Sicht. Es enthält und kontrolliert mehrere Teilelemente, sog. *Views*, die die Zugriffsschnittstelle des Anwenders auf PPM-Objekte, Anwendungen und Systemdienste wie Mail und Ereignisdienste realisieren. Ein Desktop ist jeweils genau einem menschlichen Anwender zugeordnet, so daß im Rahmen der Bearbeitung unterschiedlicher Phasen des Produktionsablaufs bei mehreren parallel verfolgten Aufträgen zu einem Zeitpunkt eine Anzahl verschiedener Desktops gleichzeitig auf dem PPM und den Entwicklungswerkzeugen operieren kann.

Das Desktop leistet weiterhin Unterstützung bei der unten vorgestellten Signalisierung von Änderungsnachrichten von PPM-Objekten an zugeordnete Views. Um die Anzahl der Kommunikationsvorgänge bei einer Objektänderung zu reduzieren, wird jeweils nur eine Änderungsmeldung pro Desktop versandt, eine spezielle Komponente im Desktop übernimmt dann die lokale Weiterleitung an interessierte Views.

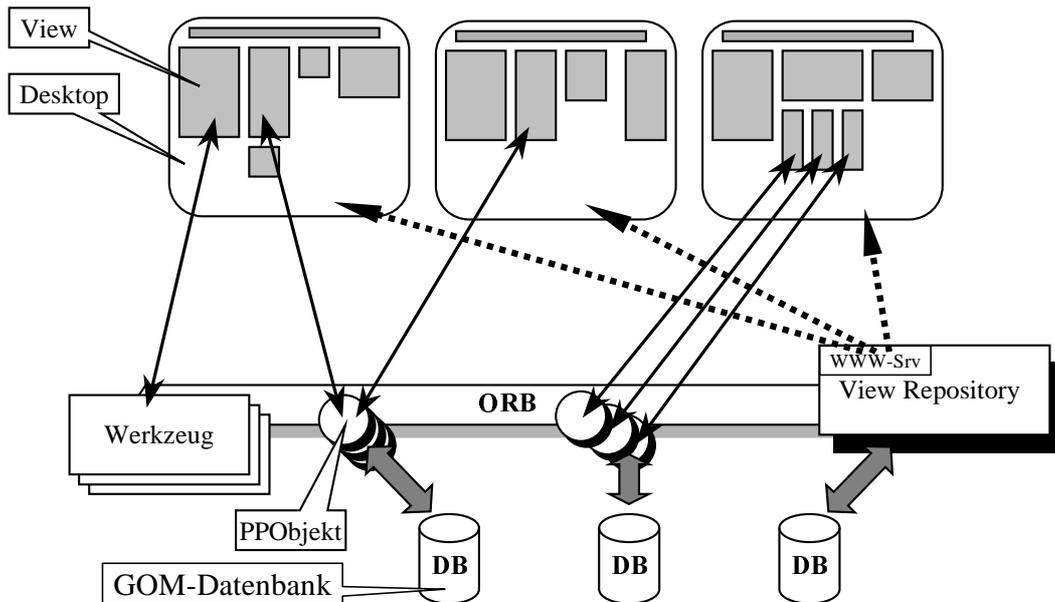


Abbildung 1: Architektur des SFB346-Destops

View Ein View realisiert die Schnittstelle für den Zugriff eines menschlichen Bedieners auf die im Rahmen der Produktentstehung benötigten Daten und Werkzeuge. Es existieren mehrere Typen von Views, von denen die wichtigsten die *PPMObjectViews* sind, die an ein PPM-Objekt gebunden sind und den Zugriff auf Methoden und Variablen dieses Objekts erlauben. Ein View-Typ kann für die Darstellung mehrerer PPM-Klassen geeignet sein, so existiert z.B. ein generischer View, der jedes PPM-Objekt einer beliebigen Klasse in einer kanonischen Weise darstellen kann und einfachen Zugriff auf seine Operationen und Variablen gewährt. Ein menschlicher Benutzer kann einen View entsprechend den Fähigkeiten seines Typs individuell konfigurieren.

Zu einem PPM-Objekt können zu einer Zeit mehrere Views in unterschiedlichen Desktops zugeordnet sein. Wird der Zustand eines beobachteten PPM-Objekts durch direkten Einfluß eines Views oder durch ablaufende Anwendungen geändert, so signalisiert das Objekt die Tatsache der Veränderung an alle beobachtenden Views, die daraufhin den aktualisierten Zustand auslesen und darstellen.

Weitere View-Typen übernehmen entsprechend dem Modulkonzept die Darstellung der Ausgaben von Anwendungswerkzeugen, die nicht in Form von Methoden der PPM-Objekte realisiert sind. Nach Möglichkeit sollen schrittweise alle bislang eingesetzten Werkzeuge mit einer Schnittstelle für Java-basierte Views ausgestattet werden, was im wesentlichen die Erfordernis einer IIOP-Schnittstelle für die CORBA-gerechte Kommunikation bedeutet. Die Werkzeug-Views kommunizieren dann über einen ORB mit der ausgelagerten Berechnungsfunktionalität ihres Werkzeugs.

ORB Zur Kommunikation zwischen räumlich getrennten Architekturkomponenten, z.B. zwischen einem View und seinem zugeordneten PPObjekt, dem Desktop und dem ViewRepository, einem View und einer Applikation oder zwischen einer Applikation und relevanten PPObjekten wird ein Middleware-System in Form eines Object Request Brokers (ORB) eingesetzt.

Auf dieser Kommunikationsplattform soll in einer weiteren Ausbaustufe eine verteil-

te Komponente zur Ablaufsteuerung aufgesetzt werden, deren Aufgabe darin besteht, die innerhalb bestimmter Arbeitsschritte entstandenen Ergebnisobjekte transparent an nachgelagerte Bearbeiter weiterzuleiten. Dies entspricht grob einer verteilt realisierten dynamischen Workflow-Engine.

PPObjekt PPObjekte sind einzelne Einheiten innerhalb des im gesamt-PPM spezifizierten Modells der Produktentwicklung. Sie realisieren die eigentlichen Anwendungsdaten und -funktionalität (soweit diese nicht von dedizierten Werkzeugen erbracht wird) der verschiedenen Abläufe und werden in einer GOM-Datenbank gespeichert. Nach dem bekannten Model-View-Controller-Ansatz der Benutzerschnittstellen entspricht somit das PPObjekt dem *Model*, der zugehörige View vereinigt die Komponenten *View* und *Controller* in sich. Ein PPObjekt bietet generische Funktionen zur Verwendung durch den generischen View an. Je nach Verwendungskontext können spezialisierte Views existieren, die auf bestimmte Abläufe zugeschnitten sind, indem sie z.B. Mengen von PPObjekten gemeinsam bearbeiten.

Die Architektur der GOM-Datenbank wird derzeit auf die Verwendung von CORBA-IDL als Spezifikationsprache umgestellt, aus der eine Implementierung der GOM-Objekte als CORBA-Server abgeleitet werden kann. Damit steht das für den Datenzugriff aus den Desktop-Views erforderliche Gegenstück bereit.

ViewRepository Das View Repository verwaltet die Personalisierungen der einzelnen Benutzer für die von ihnen verwendeten Views. Es wird vom Desktop für die individuelle Zuordnung von View-Typen zu PPObjekt-Klassen und die Einstellung spezifischer Konfigurationsparameter eines Benutzers für seine Views verwendet.

Bei der Darstellungsanforderung für ein PPM-Objekt wird von diesem zunächst eine Liste möglicher View-Typen angefordert. Die im View-Repository verwalteten individuellen Präferenzen eines Benutzers werden unter Berücksichtigung der aktuellen Rolle des Benutzers mit dieser Liste abgeglichen. Die Vorgabe der Präferenzen kann dabei flexibel geregelt werden und z.B. auf Einstellungen von Referenzpersonen zurückgreifen. Für die Übertragung des View-Codes zum Desktop wird der in Java vorgesehene HTTP-orientierte Mechanismus verwendet, so daß das ViewRepository auf einen zusätzlichen Web-Server zurückgreift. Nachfolgend kann der gewählte View-Typ individuell konfiguriert werden, z.B. durch Auswahl einer bestimmten Detaillierungsstufe der Darstellung.

Werkzeug Ein Werkzeug enthält die Funktionslogik einer Anwendung, die nach der oben angesprochenen Moduleinteilung von der in einem View realisierten Präsentationsfunktion getrennt wird. Das Zusammenspiel zwischen Werkzeug und Werkzeug-View erfolgt ebenfalls über CORBA-IIOP, kann jedoch ansonsten weitgehend frei realisiert werden, ohne die für PPMObjectViews bestehenden Vorgaben beachten zu müssen.

Die vom Insitut für Telematik zu realisierenden Komponenten für die SFB346-Desktop-Architektur umfassen die Kommunikationsinfrastruktur der Laufzeitumgebung für das Desktop-Frontend, die Kommunikation zwischen Views und zugeordneten PPM-Objekten oder Werkzeugen und die spezielle Systeminstanz des *View Repository* zur Durchführung der benutzer- und rollenspezifischen Auswahl und Anpassung von Views. Die Erstellung von spezialisierten Views für bestimmte Aufgaben obliegt den in der betreffenden Produktionsphase tätigen Teilprojekten des SFB 346.

3.2 Implementierung

Als Kommunikationsmechanismus zwischen den in Java implementierten Komponenten des Desktops und den Anwendungen und im PPM modellierten Objekten wurde als konkrete Middleware-Plattform OrbixWeb 3.0 von IONA ausgewählt. Die Entscheidung für dieses Produkt ist zum Großteil darin begründet, daß OrbixWeb einen speziellen *Callback*-Mechanismus anbietet, der es erlaubt, Rückaufrufe von CORBA-Servern an Client-Objekte (typischerweise Views) abzusetzen. Dies vermeidet die in anderen erhältlichen Java-ORBs – darunter der Visigenic ORB und die Sun-eigene Lösung JavaIDL – bestehende unakzeptable Erfordernis eines ORBs auf Clientseite.

Für die Realisierung des Desktops wird das Java Development Kit in Version 1.1 verwendet, das die grundlegenden Elemente einer graphischen Benutzeroberfläche bereitstellt. Die einzelnen Views sollen der *Java Beans* Komponentenarchitektur entsprechen, womit im wesentlichen die Mechanismen zur Weiterleitung von Nachrichten zwischen verschiedenen Desktop-Komponenten festgelegt sind. Für den weiteren Ausbau des SFB346-Desktops sind einige Erweiterungen des Java-APIs interessant, darunter besonders die *swing*-Bibliothek mit weiteren graphischen Standardelementen und die *dnd*-Bibliothek, die mit einem Drag-and-Drop-Mechanismus die Möglichkeiten der Benutzerinteraktion verbessert. Diese Erweiterungen liegen derzeit erst als Vorabversionen mit unzureichender Stabilität vor. Mit der Verfügbarkeit der JDK-Version 1.2 werden sie auch im SFB346-Desktop verwendbar sein.

Literatur

- [GrHK96] Arnd G. Grosse, Jörn Hartroth und Dietmar A. Kottmann. Delegating Client Functionality for Optimized Access to Object-Oriented Databases. In *Proceedings of the 1996 International Conference on Object Oriented Information Systems*, London, Dezember 1996. S. 123–134.
- [GrKH97] Arnd G. Grosse, Dietmar A. Kottmann und Jörn Hartroth. Disseminating Object-Oriented Applications in Large-Scale Environments. In *Proceedings of the 1997 International Conference on Object Oriented Information Systems*, Brisbane, November 1997. S. 493–503.
- [HaKG97] Jörn Hartroth, Dietmar A. Kottmann und Arnd G. Grosse. Zugriffsmodalitäten für objektintensive Anwendungen in verteilten Systemen. In *GI/ITG-Fachtagung Kommunikation in verteilten Systemen*, Braunschweig, Februar 1997. S. 282–296.

Teil III

Eigenveröffentlichungen des Instituts

Eigenveröffentlichungen des Instituts

- [AbMa97] Sebastian Abeck und Christian Mayerl. Prozeßbeschreibungen als Basis für einen qualitätsgesicherten Betrieb von vernetzten Arbeitsplatzrechnern. *Proceedings der 4. ITG/GI-Fachtagung Arbeitsplatz-Rechensysteme (APS'97)*, Mai 1997, S. 217–225.
- [BDHS⁺97] Roland Bless, Elmar Dorner, Markus Hofmann, Claudia Schmidt und Jochen Seitz (Hrsg.). Netzwerk-Management und Hochgeschwindigkeitskommunikation – Teil XV. Interner Bericht 5, Institut für Telematik, 1997.
- [BDSS⁺97] Roland Bless, Stefan Dresler, Günter Schäfer, Claudia Schmidt und Hajo Wiltfang (Hrsg.). Netzwerk-Management und Hochgeschwindigkeitskommunikation – Teil XVI. Interner Bericht 18, Institut für Telematik, 1997.
- [Bles98] Roland Bless. Flexibles Ende-zu-Ende QoS-Monitoring in Hochleistungsnetzen. In Klaus Heidtmann und Bernd Wolfinger (Hrsg.), *Leistungs-, Zuverlässigkeits- und Verlässlichkeitsbewertung von Kommunikationsnetzen und verteilten Systemen*, Nr. 209 der Berichte des Fachbereichs Informatik, Universität Hamburg, März 1998. Fachbereich Informatik, S. 1–3.
- [BlSc97] Roland Bless und Claudia Schmidt. Service-Tailored QoS Management in High Performance Networks. In Andrew Campbell und Klara Nahrstedt (Hrsg.), *Building QoS into Distributed Systems*, London, 1997. Chapman & Hall. Proceedings of the 5th IFIP International Workshop on Quality of Service, 21.–23. Mai 1997, New York, USA.
- [CaDr97] G. Carle und S. Dresler. Reliable Ordered Multicast Service (ROMS) over NBMA Networks. In *Proceedings of 6th Open Workshop on High Speed Networks, Stuttgart, Deutschland, 8.-9. Oktober 1997*, 1997, S. 161–167.
- [CaDS98] G. Carle, S. Dresler und J. Schiller. Multipath FEC Scheme for the ATM Adaptation Layer AAL5. In *International Conference on Broadband Communication*, 1998.
- [DDHR⁺98] Elmar Dorner, Stefan Dresler, Markus Hofmann, Hartmut Ritter, Jochen Schiller und Jochen Seitz. Netzwerk-Management und Hochgeschwindigkeitskommunikation – Teil XVII. Interner Bericht 6, Institut für Telematik, 1998.
- [DHSW97] Stefan Dresler, Markus Hofmann, Claudia Schmidt und Hajo R. Wiltfang. A Native ATM API Suited for Multimedia Communication. In R. Steinmetz und L.C. Wolf (Hrsg.), *Proceedings of the International Workshop on Interactive Distributed Multimedia Systems and Telecommunication Services*, Darmstadt, Deutschland, 10. - 12. September 1997. Springer-Verlag, Berlin Heidelberg, S. 352–363.
- [DrGR97] S. Dresler, A.G. Grosse und A. Rösner. Realisierung und Optimierung der Informationsbeschaffung von Internet-Suchmaschinen am Beispiel von www.crawler.de. In *Hypertext, Information Retrieval, Multimedia (HIM'97)*, Dortmund, September 1997. S. 237–250.

- [DrHo97] S. Dresler und M. Hofmann. Adaptive Error Correction to Support Heterogeneous Multicast Groups. In *Proceedings of 6th Open Workshop on High Speed Networks, Stuttgart, Deutschland, 8.-9. Oktober 1997*, 1997, S. 169–174.
- [Gaed97] Martin Gaedke. ActiveX: Componentware im Internet. *Theorie und Praxis der Wirtschaftsinformatik* 34(197), September 1997.
- [Gell97] Hans-W. Gellersen. Web Engineering: Softwaretechnik für Anwendungen im World-Wide Web. *Theorie und Praxis der Wirtschaftsinformatik* 34(196), Juli 1997.
- [GeWG97] Hans-W. Gellersen, Robert Wicke und Martin Gaedke. WebComposition: An Object-Oriented Support System for the Web Engineering Lifecycle. In *Proc. of the 6th International World-Wide Web Conference (WWW6)*, Santa Clara, April 7–12 1997.
- [GHHK⁺97] A.G. Grosse, J. Hartroth, G. Hillebrand, D.A. Kottmann, G. Krüger und P.C. Lockemann. SFB 346 — Integrationstechnologie als Innovationsmotor für Maschinenbauanwendungen. In *Informatik '97 – Informatik als Innovationsmotor – 27. Jahrestagung der Gesellschaft für Informatik*, Aachen, 24.–26. September 1997. S. 185–194.
- [GrHa97] A. Grosse und J. Hartroth (Hrsg.). Architektur vernetzter Systeme. Technischer Bericht 2/98, Universität Karlsruhe, Fakultät für Informatik, 1997.
- [GrHK96] Arnd G. Grosse, Jörn Hartroth und Dietmar A. Kottmann. Delegating Client Functionality for Optimized Access to Object-Oriented Databases. In *Proceedings of the 1996 International Conference on Object Oriented Information Systems*, London, Dezember 1996. S. 123–134.
- [GrKH97] A.G. Grosse, D.A. Kottmann und J. Hartroth. Disseminating Object-Oriented Applications in Large Scale Environments. In *International Conference on Object Oriented Information Systems (OOIS'97)*, Brisbane, Australia, 10 – 12 November 1997.
- [GrKo96] A. Grosse und D. Kottmann (Hrsg.). Daten in verteilten Systemen. Technischer Bericht 38/96, Universität Karlsruhe, Fakultät für Informatik, 1996.
- [GrKo97a] A. Grosse und D. Kottmann (Hrsg.). Architektur vernetzter Systeme. Technischer Bericht 31/97, Universität Karlsruhe, Fakultät für Informatik, 1997.
- [GrKo97b] A. Grosse und D. Kottmann. Optimale Interworking-Strategien für verteilte Trader-Netze. Technischer Bericht 8/97, Universität Karlsruhe, Fakultät für Informatik, 1997.
- [HaHo98] Jörn Hartroth und Markus Hofmann. Using IP multicast to improve communication in large scale mobile agent systems. In *Proceedings of the 31st Hawaiian International Conference on System Sciences (HICSS'31)*, Kohala Coast, Hawaii, Jan. 6-9, 1998, Januar 1998.

- [HaKG97] Jörn Hartroth, Dietmar A. Kottmann und Arnd G. Grosse. Zugriffsmodalitäten für objektintensive Anwendungen in verteilten Systemen. In *GI/ITG-Fachtagung Kommunikation in verteilten Systemen*, Braunschweig, Februar 1997. S. 282–296.
- [Hart97] Jörn Hartroth. Dynamisches Cloning als optimierter Mobilitätsmechanismus für autonome Agenten – Analytisches Kostenmodell. Technischer Bericht 21, Universität Karlsruhe (TH), September 1997.
- [Hofm96a] M. Hofmann. Hierarchische Strukturierung globaler Kommunikationsgruppen. In M. Zitterbart (Hrsg.), *Kommunikation in verteilten Systemen*, Informatik aktuell. Springer Verlag, 1996, S. 328–342.
- [Hofm96b] M. Hofmann. Scalable Multicast Communication in the Internet. *Connections* 10(10), 1996, S. 24–32.
- [Hofm97] M. Hofmann. Enabling Group Communication in Global Networks. In *Proceedings of Global Networking'97, Calgary, Alberta, Kanada, 15.-18. Juni*, Band II, 1997, S. 321–330.
- [Hofm98] M. Hofmann. *Skalierbare Multicast-Kommunikation in Weitverkehrsnetzen*. Dissertation, Fakultät für Informatik, Universität Karlsruhe, Februar 1998.
- [HoHa97] Markus Hofmann und Jörn Hartroth. Ein Meßsystem zur Bewertung von Multicast-Protokollen im Internet. In *9. ITG/GI-Fachtagung Messung, Modellierung und Bewertung von Rechen- und Kommunikationssystemen*, Freiberg, September 1997. S. 99–109.
- [HoRo97] M. Hofmann und M. Rohrmüller. Impact of Virtual Group Structure on Multicast Performance. In A. Danthine und C. Diot (Hrsg.), *From Multimedia Services to Network Services*, Nr. 1356 der Lecture Notes in Computer Science. Springer Verlag, 1997, S. 165–180. Proceedings of Fourth International COST 237 Workshop, Lissabon, Portugal, 15.-19. Dezember 1997.
- [HoSc97] M. Hofmann und C. Schmidt. Integration von ATM in bestehende Netze. In *Theorie und Praxis der Wirtschaftsinformatik - HMD*, Nr. 196 der Lecture Notes in Computer Science. Hüthig Verlag, Juli 1997, S. 96–108.
- [LaGe97] Markus Lauff und Hans-Werner Gellersen. Multimedia Client Implementation on Personal Digital Assistants. In *IDMS 97: Interactive Distributed Multimedia Systems and Telecommunication Services*, Lecture Notes in Computer Science, 1997.
- [PSSW98] G. Partsch, A. Schmidt, A. Specker und M. Weber. Multimedia technologies and Internet services for optimizing civil engineering projects. In *Proc. The 16th IASTED International Conference APPLIED INFORMATICS - AI'98*, Garmisch-Partenkirchen, Germany, Februar23–25 1998.
- [ScBa97] Albrecht Schmidt und Zuhair Bandar. A modular neural network architecture with additional generalisation abilities for large input vectors. In

Proc. Third International Conference on Artificial Neural Networks and Genetic Algorithms (ICANNGA), Norwich/England, April1–4 1997.

- [Seit97] J. Seitz. Meta Managed Objects. In *Fifth IFIP/IEEE International Symposium on Integrated Network Management ISINM'97, San Diego, CA, USA, May 12–16, 1997*. IFIP/IEEE, 1997.
- [SSPW⁺98] A. Schmidt, A. Specker, G. Partsch, M. Weber und S. Hoeck. An Agent-Based Telecooperation Framework. In *Proceedings of CoBuild'98. Darmstadt, February 1998. Lecture Notes in Computer Science 1370. Springer: Heidelberg, 1998.*, Darmstadt, Februar 1998.
- [WeIS98] M. Weber, T. Illmann und A. Schmidt. WebFlow - Decentralized workflow management in the World Wide Web. In *Proc. The 16th IASTED International Conference APPLIED INFORMATICS - AI'98*, Garmisch-Partenkirchen, Germany, Februar23–25 1998.
- [Wilt97a] Hajo R. Wiltfang. An Approach to QoS Monitoring in ATM Networks. In D.W. Faulkner und A.L. Harmer (Hrsg.), *Proceedings of the European Conference on Networks and Optical Communication – Part II: Core and ATM Networks*, Antwerpen, Belgien, 17. - 20. Juni 1997. IOS Press, Amsterdam, S. 328–335.
- [Wilt97b] Hajo R. Wiltfang. Management und Monitoring von ATM-Netzen. *Praxis der Informationsverarbeitung und Kommunikation (PIK)* 20(2), Juni 1997, S. 68–75.
- [WiSc97] Hajo R. Wiltfang und Claudia Schmidt. QoS Monitoring for ATM-based Networks. In R. Boutaba und A. Hafid (Hrsg.), *Management of Multimedia Networks and Services*, Montreal, Kanada, 8. - 10. Juli 1997. Chapman & Hall, London, S. 55–66.