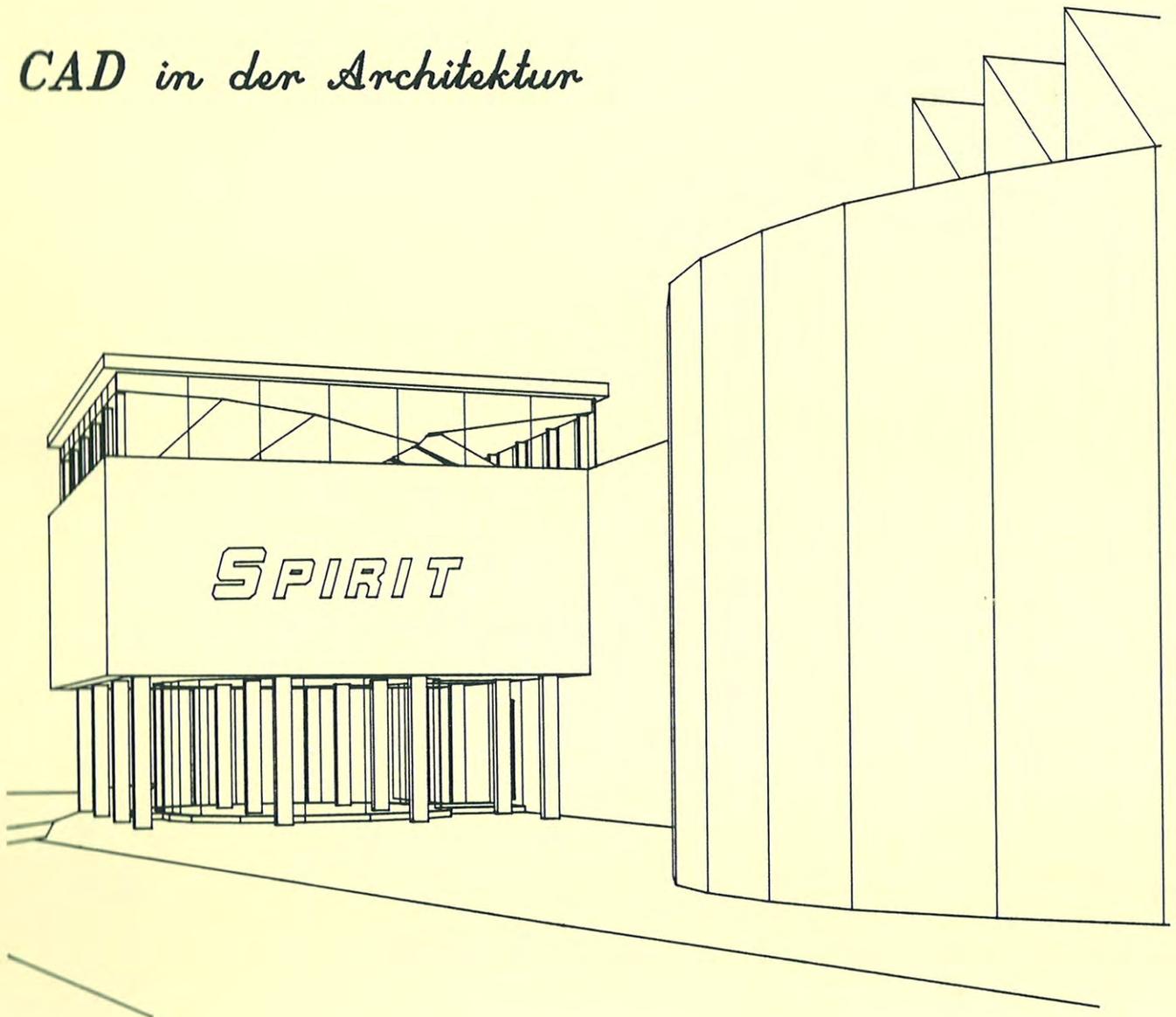

WIRTSCHAFTS

HEFT 18

1989

CAD in der Architektur



TU-PRA

*Herausgegeben von der Abt. Prozeßrechenanlage des
EDV-Zentrums der Technischen Universität Wien,
A-1040 Wien, Gußhausstraße 25*



Editorial

“Feedback” erscheint diesmal in einer höheren Auflage. Bisher wurde es an den Benutzerkreis der Abt. Prozeßrechenanlage (PRA) sowie an einige ausgewählte Adressaten versendet, um über wesentliche Schwerpunkte und Neuigkeiten aus der Arbeit der Abt. Prozeßrechenanlage zu berichten. Im Hinblick auf die derzeitige Diskussion um ein einheitlicheres Bild des EDV-Zentrums, aber auch aus der bisherigen Erfahrung, daß potentielle, aber nicht registrierte Benutzer über Dienstleistungen der Abt. Prozeßrechenanlage nicht informiert waren, ergeht die neueste Ausgabe des “Feedback” an jeden Instituts- bzw. Abteilungsleiter der **TU**-Wien. Wir hoffen, daß das “Feedback” dadurch auch in die Hände interessierter Institutsmitarbeiter gelangt, die wir bisher nicht erreichen konnten. Insbesondere erscheint die vorliegende Ausgabe prädestiniert, einen Überblick über die verschiedenen Arbeitsgebiete der Abt. Prozeßrechenanlage zu geben.

Computerunterstütztes Konstruieren (CAD) zählt seit zehn Jahren zu einem Arbeitsschwerpunkt an der PRA. Im Rahmen eines Beitrages wird der Wandel der Dienstleistungen in diesem Gebiet, vom Betrieb eines vor zehn Jahren noch sehr seltenen Leiterplatten-Entwicklungssystems zum heutigen Angebot, wie Beschaffung von Camplizenzen, Ankaufsberatung und zentralem Plotterservice, aufgezeigt.

Vier Berichte aus dem Schwerpunktgebiet Datenetze geben einen Einblick in die Arbeit in diesem heute durch die Dezentralisierung der Rechner so wichtigen Gebiet der Datenverarbeitung. Ein Beitrag beschäftigt sich mit dem erfolgreichen Betrieb des österreichischen EUnet-Hauptknotens, mit dessen Hilfe Nachrichten zwischen einzelnen Adressaten im EUnet, einem europaweiten Netz von Rechnern mit dem Betriebssystem UNIX ausgetauscht, und Neuigkeiten aus über 400 “Newsgroups” gelesen bzw. diskutiert werden können. Das Universitätsnetz Austria (UNA) ist Thema eines weiteren Beitrages, in dem einerseits über die Aufnahme neuer Dienste wie z.B. eines NOTES-Konferenzsystems oder des X.400-Maildienstes berichtet wird, andererseits neue noch in Entwicklung befindliche Arbeiten, wie Gateways zu anderen Datenetzen, oder der Anschluß von UNA an das internationale HEPNET (DECnet) vorgestellt werden.

Die immer größer werdende Computerleistung ist Grundlage dreier weiterer Beiträge dieses Heftes, auf die besonders verwiesen werden soll, da diesen Themen in Zukunft große Bedeutung zukommen wird. Die steigende Benutzerzahl von Softwarepaketen zur symbolischen Lösung mathematischer Probleme ist ein Indikator für deren Wichtigkeit im technisch-wissenschaftlichen Bereich, die durch einen Anwenderbeitrag noch verdeutlicht wird. Die gegenwärtige Verfügbarkeit leistungsfähiger Hardware und komplexer Softwaresysteme führte bei allen namhaften Herstellern zu gemeinsamen Bestrebungen nach einer einheitlichen Mensch-Maschine-Schnittstelle. Die damit zusammenhängenden Probleme werden in diesem Heft grundlegend dargestellt, und in einem weiteren Beitrag wird auf die Bedeutung des Software Engineering für die Entwicklung komplexer Softwaresysteme eingegangen.

G.W.

Inhaltsverzeichnis

Editorial	1
Inhaltsverzeichnis	2
CAD als Dienstleistung eines universitären Rechenzentrums	3
Was gibt es neues im Universitätsnetz Austria ?	6
TCP/IP am TU NET	12
EUnet, your net @ work	13
Konzept für ein X.25-Netz für Österreichs Universitäten	20
Rechnerkonfiguration an der Abt. Prozeßrechenanlage	26
Effiziente Nutzung der Computertechnologie durch neuartige Mensch-Maschine-Kommunikation	32
Software Engineering als Konstruktionsgrundlage für zeitgemäße Softwaresysteme	38
Zur Bearbeitung nichtlinearer Gleichungssysteme mit MAPLE	44

Offenlegung gemäß §25 Mediengesetz

Grundlegende Richtung

Information der Benutzer der Rechenanlagen an der Abt. Prozeßrechenanlage des
EDV-Zentrums der **TU**-Wien

Inhaber, Herausgeber

EDV-Zentrum der Technischen Universität Wien,
Abt. Prozeßrechenanlage,
A-1040 Wien, Gußhausstraße 25-29

Hersteller

Druckerei der ÖH an der **TU**-Wien, A-1040 Wien, Wiedner Hauptstraße 8-10

Redaktion/Layout

Dr. G. Wehrberger, Dipl. Ing. G. Gollmann

Titelblatt

Christian Kühn, Diplomarbeit am Institut für Gebäudelehre, Prof. Anton Schweighofer

CAD als Dienstleistung eines universitären Rechenzentrums

Günther Wehrberger

1. Entwicklung

Computer Aided Design (CAD), heute ein weit verbreiteter Begriff, hat an der Abt. Prozeßrechenanlage eine zehnjährige Tradition. Bereits 1979 übernahm die Abt. Prozeßrechenanlage die Betriebsführung des ersten graphischen Entwurfssystems der **TU**-Wien, welches vom Institut für allgemeine Elektrotechnik und Elektronik für Leiterplattenentwicklung und IC-Design aus Forschungsmitteln angeschafft wurde. Die langjährige Beschäftigung mit CAD und das damit erworbene Know How führten schließlich zur schwerpunktmäßigen Betreuung der **TU**-Institute im Bereich des computerunterstützten Konstruierens durch die Abt. Prozeßrechenanlage. Heute, zehn Jahre nach den ersten Erfahrungen mit CAD im Bereich der Elektrotechnik/ Elektronik, werden die Dienstleistungen der Abt. Prozeßrechenanlage auf diesem speziellen Gebiet der graphischen Datenverarbeitung auch von Instituten der Fakultät Maschinenbau sowie Raumplanung und Architektur in Anspruch genommen.

2. Arten der Dienstleistungen

Das Dienstleistungsangebot hat sich in dieser Zeit stark gewandelt, bzw. erweitert. Es reicht heute vom Plotter-Service und dem Betrieb eines Printplatten-Entwurfssystems über die System- und Softwareunterstützung bei Institutsrechnern sowie der Beschaffung von Campuslizenzen bis zur Ankaufsberatung und Schulung.

2.1 Betrieb eines zentralen CAD-Systems

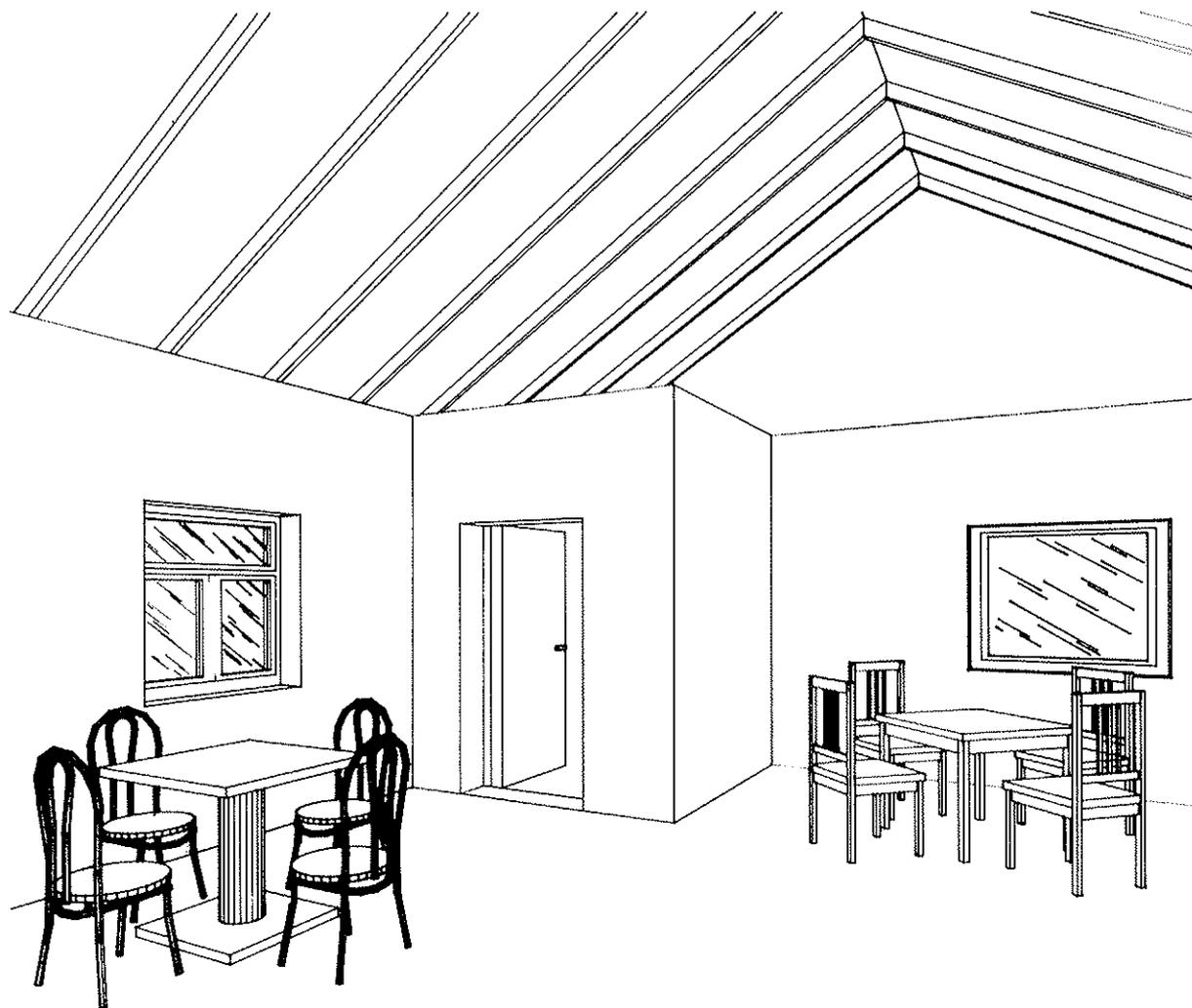
Bei dem erwähnten Printplatten-Entwurfssystem handelt es sich um ein schlüsselfertiges CAD-System der Firma CALMA, welches mit drei Arbeitsplätzen in den Räumen des EDV-Zentrums zur Benutzung zur Verfügung steht. Die Nachteile sind der mit der Verwendung des Systems verbundene notwendige Ortswechsel, sowie eine an Reservierungslisten gebundene Arbeitszeit. Durch das enorm gestiegene Preis-/Leistungsverhältnis heutiger CAD-Systeme sind immer mehr Anwender in der Lage, auf institutseigene PC's und Arbeitsstationen umzusteigen. Aus diesem Grund, sowie auch wegen der hohen Hardware-Wartungskosten dieses alten Systems soll noch heuer ein Ersatz geschaffen werden. Da die Benutzung aus oben erwähnten Gründen nicht mehr in dem Ausmaß erfolgt, wie noch vor einigen Jahren, ein Printplatten-Entwurfssystem an einer Technischen Universität aber doch allgemein zugänglich zur Verfügung stehen soll, wird die zehn Jahre alte Anlage durch ein modernes, den Erfordernissen entsprechendes CAD-System auf Workstation-Basis (ein Arbeitsplatz) ersetzt werden.

2.2 Campuslizenzen und zentrale Software-Wartung

Aus dem Betrieb eines ursprünglich ebenfalls zentralen CAD-Systems für die mechanische Konstruktion, dem 3D-Volumensystem EUCLID, ist im Lauf der letzten beiden Jahre eine neue Dienstleistung entstanden, nämlich die Installation und Wartung zentral beschaffter Software an Institutsrechnern. EUCLID stellt ein typisches Beispiel für die Entwicklung zur dezentralen Verwendung von Software dar. Einerseits ist die Zahl der Benutzer am zentralen Rechner der Abt. Prozeßrechenanlage gestiegen, da seit zwei Jahren PC's mit geeigneter Graphikausrüstung (PGA-Karte) statt der kostspieligen EUCLID-Arbeitsstationen verwendet werden können, wodurch EUCLID nun vom eigenen Schreibtisch aus zu verwenden ist. Andererseits darf EUCLID aufgrund einer Vereinbarung mit der Lieferfirma von der Abt. Prozeßrechenanlage auch auf geeigneten Institutsrechnern, die an das lokale Rechnernetz der **TU**-Wien angeschlossen sind, installiert werden.

Die Beschaffung und Installation neuer Software-Versionen sowohl am zentralen Rechner der Abt. Prozeßrechenanlage als auch an den Institutsrechnern wird in Zukunft weiter an Bedeutung gewinnen. Durch die Übernahme EDV-spezifischer Arbeiten an den Institutsrechnern durch entsprechend geschultes Personal des EDV-Zentrums wird an den Instituten der notwendige Freiraum geschaffen, um mit einem so mächtigen Werkzeug wie einem CAD-System, fachspezifische Arbeiten durchführen zu können.

Als weiteres Beispiel sei das PC-CAD-System SPIRIT angeführt, dessen Stärken vor allem im Architekturbereich liegen. Daß dieses Software-Paket bereits seit einem Jahr an den meisten Architektur-Instituten der **TU**-Wien zur Verfügung steht, ist Ergebnis der Zusammenarbeit zwischen der Fachgruppe Architektur und der Abt. Prozeßrechenanlage. Im Rahmen der vor zwei Jahren durchgeführten Beratung der Architektur-Institute wurde von der Abt. Prozeßrechenanlage auf ein neu am europäischen Markt befindliches CAD-System namens SPIRIT hingewiesen, welches durch hervorragende Eigenschaften wie 3-dimensionale Datenbasis, komfortable Wandeingabe und -korrektur, Erstellung von Perspektiven und Innenraumperspektiven (siehe Bild) für den Einsatz im Architekturbereich besonders geeignet erschien. In weiterer Folge wurde mit dem SPIRIT-Repräsentanten für Österreich vereinbart, daß SPIRIT-Lizenzen für Lehr- und Forschungszwecke von allen Instituten und besonderen Universitätseinrichtungen der **TU**-Wien um knapp über S 10.000,- (inkl. MWSt.) zu erhalten sind, sofern Beratung, Bestellung, Installation und Schulung durch die Abt. Prozeßrechenanlage erfolgt.



Mit SPIRIT erstellte Innenraumperspektive

2.3 Ankaufsberatung und Schulung

Die Beratung der Institute bei der Auswahl spezieller Software für institutseigene Rechner, sowie die Überprüfung von Angeboten ist eine weitere Dienstleistung der Abt. Prozeßrechenanlage, die immer mehr in Anspruch genommen wird (siehe auch 2.2, SPIRIT). Auch die Schulung der Benutzer ist als Dienstleistung im CAD-Bereich sehr wichtig, da die Software-Lieferfirmen eher günstige Softwarepreise gewähren, dafür aber die Schulung voll in Rechnung stellen. Die Kurskosten für zwei Mitarbeiter können mitunter die Anschaffungskosten der Software sehr schnell überschreiten.

2.4 Plotter-Service

2.4.1 DIN A0-Plotter

Die graphische Ausgabe von Entwürfen, welche im Computer und nicht am Reißbrett entstanden sind, ist im Zusammenhang mit CAD besonders wichtig, da diese Zeichnungen entweder das Endprodukt darstellen, wie z.B. ein Plan in der Architektur, oder Ausgangspunkt bzw. Kontrolle für weitere Arbeitsgänge bilden, wie z.B. eine Werkstattzeichnung im Bereich der mechanischen Konstruktion.

Ein genauer, schneller Plotter für Formate bis DIN A0 ist oft teurer als das Rechnersystem, mit dessen Hilfe die Zeichnung erstellt wurde. Aus diesem Grund bietet die Abt. Prozeßrechenanlage mit dem Plotter-Service die Möglichkeit, Zeichnungen bis zum Format DIN A0 mit Bleistift, Faserschreibern oder Tusche zentral auszugeben. Voraussetzung ist das Vorliegen der Plotterdaten im HP-GL-Format (Hewlett Packard Graphic Language). Es können alle gängigen Diskettengrößen bzw. Formate verarbeitet werden, oder die Daten direkt über das lokale Rechnernetz

der **TU-Wien (TUNET)** zur Graphik-VAX der Abt. Prozeßrechenanlage geschickt werden, um von dort an den Plotter weitergeleitet zu werden.

2.4.2 Photoplotter

Aufgrund der Komplexität moderner Leiterplatten stellt ein Photoplotter heute ein äußerst wichtiges Glied in der Produktionskette einer Leiterplatte dar. Der alte Photoplotter der Abt. Prozeßrechenanlage war einerseits zu langsam und erfüllte andererseits nicht mehr die hohen Genauigkeitsanforderungen für Leiterplattenentwürfe. Seit kurzer Zeit steht nun an der Abt. Prozeßrechenanlage ein neuer, gebrauchter Photoplotter der Spitzenklasse, Type GERBER PC40, zur Verfügung. Bei diesem neuen Plotter handelt es sich um ein besonders stabiles, schnelles und hochgenaues Gerät (Genauigkeit 0.0025 cm, Auflösung 0.001 cm), mit nach Benutzerwünschen angefertigten Blenden (siehe Tabelle). Da die Entwürfe für die Leiterplatten nicht mehr ausschließlich am Calma-System, sondern dem Trend entsprechend sehr oft an Instituts-PC's oder Workstation-Rechnern entstehen, können Daten auch direkt von Disketten (5,25", 360 KB) verarbeitet werden, oder über **TUNET** und GraphikVAX zum Photoplotter geschickt werden.

Station-Nr	D-Comm	Größe		Kommentar
			Zoll	
Linien				
1	D-10		.002	
2	D-11		.004	
3	D-12		.008	
4	D-13		.013	
5	D-14		.025	
6	D-15		.039	
7	D-16		.050	
8	D-17		.059	
9	D-18		.079	
10	D-19		.100	
Lötlagen		D1 (mil)	DA (mil)	
11	D-70	37	40	Achteck
12	D-71	44	48	Achteck
13	D-20	50	55	Achteck
14	D-21	57	62	Achteck
15	D-22	65	70	Achteck
16	D-23	70	76	Achteck
17	D-24	78	85	Achteck
18	D-25	87	94	Achteck
19	D-26	97	105	Achteck
20	D-27	110	119	Achteck
21	D-28	57	57	Rechteck
22	D-29	75/59	81/64	Spezial(Multilayer)
23	D-72	110/95	119/103	Spezial(Multilayer)
24	D-73	185	185	Paßmarke

Tabelle: Blendenbelegung des Photoplotters GERBER PC40

(D1=Innendurchmesser, DA=Außendurchmesser, 1 mil = 0.001 Zoll = 0.0254 mm)

3. Ausblick

In der mittelbaren Zukunft wird der EDV-Durchdringungsgrad an den Instituten weiter zunehmen, und somit auch die Zahl der dezentralen CAD-Systeme ansteigen. Daher wird ein zentrales Plotter-Service, die Beschaffung von Campus-Lizenzen, Ankaufsberatung und Know How-Weitergabe wichtiger sein, als der Betrieb zentraler CAD-Systeme an Rechenanlagen des EDV-Zentrums.

Was gibt es Neues im Universitätsnetz Austria ?

Christian Panigl

Abstract

Auch im vergangenen Jahr gab es wieder einen beachtlichen Zuwachs an Benutzern, Rechnern und Dienstleistungen im UNA. Weitere Schritte zur OSI-konformen Öffnung wurden gesetzt, aber auch eine internationale Öffnung wurde konkret in Angriff genommen.

1. Die 2. UNA-Donation

Im Rahmen der 2. UNA-Donation der Firma Digital Equipment Corporation konnten zwei weitere Organisationen in UNA integriert werden:

- Forschungsgesellschaft Joanneum
- Bundes-Versuchs- und Forschungsanstalt (Arsenal-Wien)

es konnten Erweiterungen vorgenommen werden:

- UNI-Klagenfurt
- Akademie d. Wissenschaften, Hochenergiephysik

und folgende Software und Lizenzen wurden von der Firma Digital für alle UNA-Bereichsrechner zur Verfügung gestellt:

- Message Router
- Message Router VMSmail-Gateway
- VAX/Notes

Darüberhinaus steht an der **TU**-Wien/PRA (EGH780) der Message Router Programmiers Kit allen Interessenten aus UNA zur Entwicklung von Mail-Gateways und User-Agents zu Verfügung, sowie die VMS/SNA-Gatewaysoftware für microVAX, zum Erfahrung sammeln mit der Kopplung an die IBM-Welt.

Die UNA-Donation II, für die wir uns im Namen des ACONET-Vereins herzlich bedanken, wird UNA außer einer sehr begrüßenswerten Erweiterung der Teilnehmer folgende neue Möglichkeiten bringen:

1.1 Einführung eines "Store and Forward"-Mechanismus im VMS-Mail-Verbund

Um per DECnet/VMSmail eine Mail verschicken zu können, muß zum Zeitpunkt der Versendung der Zielknoten verfügbar sein. Andernfalls erhält der Absender sofort eine Unzustellbarkeitsmeldung. Je größer und dezentralisierter nun ein Rechnernetz wird, desto größer wird auch die Wahrscheinlichkeit, daß ein Zielrechner zu einem bestimmten Zeitpunkt nicht erreichbar ist. Um nun nicht von Hand aus immer wieder neue Versuche unternemen zu müssen, gibt es Software, die einen sogenannten "Store and Forward"-Mechanismus implementiert. Diese Funktion bietet auch die Message Router Software der Firma Digital.

Wenn also ein Zielrechner im Moment nicht erreichbar ist, kann der Benutzer seine Mail dem Message Router übergeben, der dann in regelmäßigen Abständen versucht, diese Mail zuzustellen, bzw. nach einer bestimmten Zeit eine Unzustellbarkeitsmeldung zurückschickt. Eine weitere komfortable Möglichkeit bietet der Message Router mit seiner Routing-Tabelle: Es kann jeder Benutzer mit seinem Namen eingetragen werden und dem Verweis auf seine Mailbox-Adresse. Ein über die tatsächliche Adresse uninformatierter Absender, kann nun dem Message Router nur unter Angabe eines Namens die Mail übergeben, für die korrekte Adressierung sorgt der Message Router.

1.2 Einführung des NOTES-Konferenzsystems in UNA

Bereits beim letzten UNA-Workshop wurden die NOTES als attraktiver lokaler und überregionaler Dienst diskutiert. Zwei Hürden wurden damals erkannt, um den Dienst einzuführen:

- die UNA-Bereiche verfügten mit Ausnahme der PRA nicht über die NOTES-Software,
- hohe DATEX-P-Kosten bei überregionalen Konferenzen oder bei der Teilnahme an Konferenzen in anderen Systemen.

Die erste Hürde konnte nun beseitigt werden. Auch die zweite ist, wenn schon nicht beseitigt, so doch wesentlich geringer: Ab 1.1.89 gibt es neue DATEX-P-Tarife, die für ganz Österreich einheitliche Volums- und Zeitgebühren vorsehen:

15.- öS/kSegment	Volumsgebühr
0.09 öS/min	Zeitgebühr

Das heißt, die DATEX-P-Kosten für die bundesländer-übergreifende Kommunikation wurden auf ca. die Hälfte reduziert !

Diese reduzierte Hürde wird aber hoffentlich auch noch ganz fallen, wenn es zur Durchsetzung eines Standleistungsnetzes kommt (siehe ACONET Artikel).

Um den Überblick über die entstehenden NOTES-Konferenzen zu gewährleisten und unnötige und ungewollte Paralleldiskussionen zu vermeiden wird an der PRA ein UNA-weiter Konferenzindex geführt. Dieser Konferenzindex ist an sich wieder eine normale NOTES-Konferenz, eine Meta-Konferenz also, deren Eintragungen (also die Konferenzankündigungen) aber die angenehme Eigenschaft haben, auf die angekündigte Konferenz "zu zeigen". Beim Lesen dieses Indexes kann man dann einfach per Knopfdruck jene Konferenzen selektieren, an denen man interessiert ist.

1.3 Zugang zur IBM-Welt

Die SNA-Software eröffnet schließlich die Möglichkeit, neue Zugänge zwischen UNA und der IBM-Welt zu erschließen und den UNA-Teilnehmern damit neue Dienste zur Verfügung zu stellen.

2. Der X.400-Maildienst

2.1 X.400 in Kürze

Die CCITT X.400-Empfehlungen beschreiben die Protokolle und Services eines als "OSI-Application" laufenden Message Handling Systems (MHS). Der Ausdruck Message Handling System läßt bereits erkennen, daß es sich hierbei nicht nur um ein Electronic Mail System handelt. Vielmehr sind die vorgesehenen Protokolle so ausgelegt, daß faktisch beliebige "Mitteilungen" übertragen werden können. Die derzeitigen Spezifikationen definieren zwar nur eine winzige Untermenge der möglichen Mitteilungsformate, es kann aber nur eine Frage der Zeit sein, bis es möglich sein wird, in einer Mitteilung Text, Graphik, Sprache und andere Formate kombiniert zu verschicken.

Die X.400 Absender-/Empfänger-Adressierung unterscheidet sich (zumindest auf den ersten Blick) signifikant von den bisher in den verschiedenen E-Mail-Netzen bekannten Formaten. Sie hat eine hierarchische Struktur und unterstützt damit gestaffelte Adreßhoheiten. Die Adreßelemente haben Ähnlichkeiten mit normalen Postadressen, nur daß sie mit "Feldbezeichnern" EDV-gerecht definiert sind:

S = Zuname (Surname)

G = Vorname (Givenname)

OU = Abteilung (Organizational Unit), mehrfach möglich

O = Organisation

P = Privater Zustellbereich (Private Management Domain, PRMD)

A = Administrativer Zustellbereich (Administration Management Domain, ADMD)

C = Land (Country, nach ISO 3166)

Als Beispiel meine X.400-Adresse:

S=Panig!; G=Christian; OU=PRA; O=EDVZ; P=TU-Wien; A=ADA; C=AT

Das Message Handling System ist im wesentlichen in zwei Komponenten unterteilt, den Message Transfer Agent (MTA) und den User Agent (UA). Als User Agent kann man sich jene "Werkzeugsammlung" vorstellen, die dazu dient, eine Mitteilung zu erstellen und zu empfangen (lesen, hören ...). Der MTA kümmert sich um den Transport und die Zustellung von Mitteilungen. Als Schnittstelle zwischen diesen beiden Komponenten dient jeweils eine "Mailbox".

Jede PRMD besteht aus mindestens einem MTA der beliebig viele lokale UAs bedient. Jede PRMD ist genau einer ADMD zugeordnet, was aber nicht notwendigerweise bedeutet, daß es nur eine Verbindung zur ADMD gibt. Es können zwischen PRMDs, zwischen denen viel Nachrichtenverkehr anfällt, auch direkte Verbindungen errichtet werden. Lediglich das Vermitteln zwischen verschiedenen PRMDs bleibt der ADMD vorbehalten.

2.2 X.400 im UNA

Im Rahmen eines Forschungsprojektes, das das BMWF an den ACONET-Verein vergeben hat, wurde im UNA ein X.400 Maildienst eingeführt. Dazu wird derzeit unter dem Betriebssystem VAX/VMS die DFN-EAN-Software eingesetzt, die sowohl über einen geeigneten MTA als auch über einen geeigneten UA verfügt. Der kleine Schönheitsfehler an dieser Lösung ist die Tatsache, daß damit zwei "Mail-Ordner" gepflegt werden müssen, die X.400-(EAN-)Mailbox und die VMSmail-Mailbox. Die Firma Digital bietet derzeit jedoch nur einen X.400-fähigen MTA an (MRX). Als User-Agent dient unter anderem VMSmail, wodurch die meisten Dienste des Transfer Service und der anderen UAs nicht in Anspruch genommen werden können. An der Realisierung eines geeigneten User Agents wird jedoch gearbeitet. Eine Umstellung auf die X.400-Software der Firma Digital ist also zu einem späteren Zeitpunkt zu erwarten.

EAN wurde von der University of British Columbia (UBC) 1984 entwickelt, vom DFN-Verein als X.400-

System (Version 1984) weiterentwickelt und im DFN (Deutsches Forschungsnetz) eingesetzt. Der ACONET-Verein hat eine Lizenz bei der University of British Columbia für EAN erworben, die für 15 Installationen gilt.

Diese X.400-Software läuft derzeit auf fast allen UNA-Bereichsrechnern. Das EDV-Zentrum der TU-Graz fungiert momentan noch als Administration Management Domain (ADMD, "Zustellungsbehörde"), diese Funktion wird aber demnächst von der Firma Radio Austria als Bevollmächtigte ADMD der Österreichischen Post unter dem Admd-Namen "ADA" übernommen.

3. Vorbereitung eines X.25-Netzes auf der Basis von Standverbindungen

Das Problem der volumsorientierten Vergebührung von DATEX-P, die eine Budgetierung der Kommunikationskosten erschwert bis unmöglich macht, behinderte die allgemeine, komfortable Verwendung der interuniversitären Netzdienste. Der Einsatz eines Standleitungsnetzes wurde daher schon öfter in Erwägung gezogen. Im Auftrag des ACONET-Vereins wurde ein Konzept für einen Verbund der österreichischen Universitäts- und Forschungsorganisationen durch ein privates X.25-Netz, das auf Standverbindungen basiert, entwickelt. Dieses Konzept ist in einem anderen Beitrag dieser "Feedback"-Ausgabe wiedergegeben. Dieses Konzept wurde dem Bundesministerium für Wissenschaft und Forschung, Abt. Präs. 6, vorgestellt und wurde von dieser Stelle gutgeheißen. Der Beginn der Realisierung ist für 1990 geplant.

4. Gateways zu anderen Datennetzen

4.1 Gateways zwischen UNA und EARN

Da der X.400-Dienst erst im Aufbau begriffen ist, ist es auch noch erforderlich, UNA an andere herstellerspezifische bzw. betriebssystemspezifische Netze über Gateways heranzuführen.

Derzeit gibt es am EDV-Zentrum der Universität Wien zwei Gateways zwischen UNA und EARN:

- 1) VMS-Mail Gateway
- 2) X.400 Gateway

4.1.1 VMS-Mail-Gateway

Dieser Gateway wurde im Rahmen eines vom BMWF an den ACONET-Verein vergebenen Forschungsprojektes errichtet. Es handelt sich dabei um eine Micro-VAX II, die mit dem Namen AUNIW in UNA integriert ist. Der Rechner ist mit einer DECnet-Standleitung mit der EGH780 der Abt. Prozeßrechenanlage der TU-Wien verbunden.

AUNIW ist mit jnet-Software ausgestattet, die über die IBM-Anlage des EDV-Zentrums die Verbindung mit EARN ermöglicht. Die Weiterleitung der Mails zwischen DECnet und EARN erfolgt durch die PMDF-Software (Pascal Mail Distribution Facility).

Adressierung:

UNA → EARN: AUNIW::IN%"user@node.BITNET"

EARN → UNA: user@una-node.UNA.AT

Dieser Gateway ist seit 22.11.1988 offiziell in Betrieb. Derzeit sind in AUNIW alle jene Systeme eingetragen und damit berechtigt, den Gateway zu verwenden, die über Standleitungen erreicht werden können wie zum Beispiel alle UNA-Systeme an der TU-Wien.

4.1.2 X.400-Gateway

Ebenfalls an der UNI-Wien wurde ein Gateway zwischen EARN und X.400 in Betrieb genommen. Die Software wurde von der GMD in Darmstadt in Zusammenarbeit mit dem European Networking Centre von IBM in Heidelberg entwickelt. Durch diesen Gateway haben alle X.400-Mail-Benutzer die Möglichkeit, mit EARN-Benutzern zu kommunizieren.

Adressierung:

X.400 → EARN: <S=user;O=node;P=BITNET;A=ADA;C=AT>

EARN → X.400: surname@orgunit.org.pdmd.admd.country

Der Gateway ist derzeit im Versuchsbetrieb. Ein entsprechendes Vergebührungskonzept wurde bereits ausgearbeitet. Nach Implementierung desselben in der ADMD bzw. in den MTAs der PRMDs kann der Standardbetrieb aufgenommen werden.

4.2 Gateways zwischen UNA und EUnet

Eine Gatewayfunktion zwischen UNA und EUnet für E-Mail ist bereits seit längerer Zeit im Standardbetrieb.

5. UNA wird international

Eines der größten DECnet-basierenden Netze ist zweifellos das internationale HEPNET (High Energy Physics NETwork) mit allen seinen assoziierten und angeschlossenen Netzen. Zwei UNA-Bereiche hatten schon lange ein

besonderes Interesse an einem Anschluß von UNA an HEPNET: die Akademie der Wissenschaften, Institut für Hochenergiephysik und das Atominstitut der Österreichischen Universitäten. Dieser Wunsch wurde nun beim letzten UNA-Treffen auch von den anderen Bereichen unterstützt.

Im Zusammenhang damit wurde eine Umkonfiguration der bestehenden Area-Aufteilung in UNA nötig, um Adreßkollisionen mit bestehenden HEPNET-Bereichen zu beseitigen.

5.1 Technische Details über die Anbindung von UNA an HEPNET

Beim "erweiterten" HEPNET handelt es sich um ein inhomogenes DECnet-Netzwerk, das aus einem eindeutigen Adreß-Bereich und einem mehrdeutigen Adreß-Bereich besteht. Der eindeutige Adreß-Bereich erstreckt sich über die DECnet-Area- Nummern 1 bis 46 (inklusive). Dieser Bereich repräsentiert das HEPNET im engeren Sinn und ist homogen, das heißt, daß jeder Knoten innerhalb dieses Bereiches direkt adressierbar ist.

Um den Bedürfnissen bestehender großer DECnet-Netze, die ihrerseits bereits in mehrere Areas aufgeteilt sind, gerecht werden zu können, wird der Area-Adreß- Bereich von 47 bis 63 (inklusive) vom HEPNET nicht direkt verwaltet. Es können mehrere solche voneinander unabhängige, in sich jedoch autonome und homogene Bereiche existieren. Jeder dieser Bereiche kann über einen (oder mehrere) so genannte "Area-Filter-Knoten" an das HEPNET angebunden werden.

Es existiert also im erweiterten HEPNET genau eine "1-bis-46-Wolke" und mehrere "47-bis-63-Wolken" die jeweils über einen Area-Filter-Knoten innerhalb der 1-bis-46-Wolke an HEPNET angebunden sind.

Die Adressierung von Knoten in einer 47-bis-63-Wolke muß indirekt (zweistufig) erfolgen: Zunächst wird der entsprechende Area-Filter-Knoten adressiert und von diesem aus der Zielknoten.

5.2 Die neuen UNA-Adressen

Damit UNA in Zukunft trotz des reduzierten Adreßumfangs problemlos erweitert werden kann, werden die 1023 möglichen Adressen einer Area nicht vollständig an einen UNA-Bereich vergeben, sondern Adreß-Slots festgelegt. Damit wird erreicht, daß neu zu integrierende UNA-Bereiche einen Area-Bereich mit bereits existierenden Systemen teilen können.

Area ALT	Area NEU	Kenn- buchst.	Adreß- slot	Organisation	mögliche Assoziierte
21	47	A	1-256	Uni-Wien	Univ.Sternwarte Univ.Kliniken
22	48	B	1-256	Uni-Graz (2.Buchstabe ungleich M)	
23	49	C	1-256	Uni-Innsbruck	
24	50	D	1-256	Uni-Salzburg	Forschungszentrum
25	51	E	1-256	TU-Wien	Atominstitut Vet.Med
	51	X	1023-769	IEZ	
26	52	F	1-256	TU-Graz	
27	53	G	1-256	MU-Leoben	
30	54	J	1-256	WU-Wien	
31	55	K	1-256	Uni-Linz	
32	56	L	1-256	Uni-Klagenfurt	
36	57	P	1-256	FG-Joanneum	
53	22	Q	1-256	Akad.d.Wiss.	im HEPNET
45	59	Y	1-256	BVFA	evtl. weitere Bundes- Forschungseinrichtungen
46/47	60	Z	1-256	ÖFZ-Seibersdorf	
63	61	BM	1-256	BMWF	

Der primäre "Area-Filter-Knoten" wird an der TU-Wien/EDVZ/PRA eingerichtet und betrieben, sobald die Integration von UNA in HEPNET von den zuständigen HEPNET-Gremien endgültig genehmigt wurde und im UNA alle technischen und organisatorischen Voraussetzungen geschaffen wurden. In beratender Funktion, insbesondere was den leistungstechnischen Anschluß an HEPNET (Genf) betrifft, hat das EDVZ der Uni-Linz seine Mitarbeit angeboten.

6. Der derzeitige Status von UNA

UNA ist also im vergangenen Jahr wieder ein schönes Stück größer geworden. Es umfaßt mittlerweile 15 Bereiche mit weit mehr als zweihundert VAX-Rechnern. Wenn man alle anderen Systeme (PCs, PDPs, ...) dazurechnet, ist die Zahl sicher schon weit über 500 gestiegen.

Das folgende Verzeichnis enthält jeweils nur die Bereichskoordinationsstellen sowie den Bereichsrechner. Die Liste der angeschlossenen Systeme würde hier bereits den Rahmen sprengen.

ORGANISATION/KONTAKT	Tel.	Bereichsrechner	
		MAIL-Username	Datex-P-Nr.
UNIVERSITÄT WIEN		47.4 (APAP)	24221159
Prozeßrechenanlage Physik L: Mag. P.Karlsreiter N: A. Majerowicz Boltzmannng. 5 1090 Wien	(0222) 342630 /429	PK AM	
UNIVERSITÄT GRAZ		48.2 (BATG02)	24311094
EDV-Zentrum L: Dr. C.Leonhardt N: DI. P.Schubert Attemsg. 25/II 8010 Graz	(0316) 380 /2231 /2230	LEONHARDT SCHUBERT	
UNIVERSITÄT INNSBRUCK		49.1 (CGIE01)	25521006800
GIE Laborrechner L: Prof.Dr. G.Swoboda Technikerstr. 13 6020 Innsbruck	(05222) 748 /4280	SECY	
UNIVERSITÄT SALZBURG		50.1 (DSB750)	25629007
EDV-Zentrum L: F.Maier N: M.Feldbacher Petersbrunnstr. 19 5020 Salzburg	(0662) 8044 /6702 /6704	MAIER FELDBACHER	
TECHNISCHE UNIVERSITÄT WIEN		51.2 (EGH780)	26221070
EDV-Zentrum/PRA L: Dr. M.Paul N: C.Panigl Gußhausstr. 25 1040 Wien	(0222) 58801/3600 /3605 /3614	PAUL PANIGL	
ATOMINSTITUT DER ÖST. UNIVERSITÄTEN		51.200 (EATI)	24221155
EDV-Zentrum L: Dr. G.Westphal N: E.Seymann Schüttelstr. 115 1020 Wien	(0222) 21701 /238 /234	WESTPHAL SEYMANN	

TECHNISCHE UNIVERSITÄT GRAZ	52.12 (FINFO)	26311010
EDV-Zentrum (0316) 7061		
L: Dr. J.Theurl /6390		
N: DI. F.Haselbacher /6392		
Steyrerg. 17/P		
8010 Graz		
MONTANISTISCHE UNIVERSITÄT LEOBEN	53.1 (GMUL01)	24381049
EDV-Zentrum (03842) 42555		
L: DI. H.Maierhofer /269		
N: D.Kados	KADOS	
F.Josef-Str. 18		
8700 Leoben		
UNIVERSITÄT LINZ	55.256 (KCAMP)	24731124
Inst.f.Mathematik (RISC) (0732) 231381		
L: Prof.Dr. Buchberger /9219	BUCHBERGER	
N: DI. G.Gratzl /9221	GRATZL	
Altenbergstr. 69		
4040 Linz		
UNIVERSITÄT KLAGENFURT	56.1 (LUBW)	24411101
EDV-Zentrum (0463) 5317		
L: DI. H.Maier /288	MAIER	
N: A.Ploner /285	PLONER	
Universitätsstr. 65		
9020 Klagenfurt		
FORSCHUNGSGESELLSCHAFT JOANNEUM	57.1 (PRZJ01)	
Inst.f.dig.Bilverarbeitung (0316) 8021		
L: Dr. W.Sinna /36		
N: K.Kapper /26		
Wastiang. 6/1		
8010 Graz		
AKADEMIE D. WISSENSCHAFTEN	22.93 (QHEPHY)	24221133
Inst.f.Hochenergiephysik (0222) 557328		
L: DI. W.Mitaroff /33	MIT	
N: Dr. G.Walzel /20	WALZEL	
Nikolsdorferg. 18		
1050 Wien		
BUNDES-VERSUCHS- UND FORSCHUNGSANSTALT ARSENAL	59.1 (YGP750)	26191064
Elektrotechnisches Institut (0222) 782531		
L: HR.Dr. A.Sethy /288		
N: A.Deutsch /300	DEUTSCH	
Grillstr. 3		
1030 Wien		
ÖST. FORSCHUNGSZENTRUM SEIBERSDORF	60.5 (ZDVAXA)	26221064
Physik (02254) 80		
L: DI. E.Schoitsch /3117	SCHOITSCH	
N: K.Stellner /3261	STELLNER	
2444 Seibersdorf		

BUNDESMINISTERIUM F. WISSENSCHAFT . U. FORSCHUNG 61.1 (BMWF1)

Präs. 6 (0222) 53120
L: Dr. H. Schläger /3136 SCHLAEGER
N: H. Hatzinger /3152 HATZINGER
Freyung 1
1010 Wien

Ausblick

An einer weiteren Öffnung von UNA wird bereits gearbeitet. Durch das geplante X.25-Netz, insbesondere wenn es auf 64 kBit/s - Verbindungen ausgebaut sein wird, wird eine sehr leistungsfähige Basis für ein Akademisches Computernetz (ACONET) gegeben, auf der schrittweise sämtliche OSI-Dienste eingeführt werden können. Der Anfang dazu wurde bereits mit dem X.400-Maildienst gemacht, als nächstes wird FTAM eingeführt werden. Alle anderen Dienste (VT, JTM usw.) werden folgen, wenn dafür geeignete Software von den Herstellern verfügbar sein wird.

Parallel zur Einführung von OSI-Diensten, die UNA in ACONET überführen werden, wird an der möglichst effizienten Kopplung österreichischer Netze an internationale Wissenschaftsnetze gearbeitet. Die nächsten Schritte in diese Richtung werden die konkrete Anbindung von UNA an HEPNET und die Ankopplung österreichischer Netze an den europäischen Verbund nationaler Wissenschaftsnetze, der auf X.25 basiert und den Namen MDNS (Managed Data Network System) trägt, sein. Derzeit wird eine 64 kBit/s - Verbindung zwischen der TU-Wien und dem Schweizer MDNS-Knoten in Bern in Betrieb genommen. Das neue X.25-Netz österreichischer Universitäten wird damit über eine leistungsfähige Verbindung zu Wissenschaftsnetzen anderer europäischer Länder verfügen.

TCP/IP am TUNET

Martin Rathmayer

Mit Zunahme der Anschlüsse an TUNET hat auch TCP/IP als Netzwerkprotokoll immer mehr an Bedeutung gewonnen. Diese Protokollfamilie wurde von der DARPA Internet Group entwickelt, und es gibt Implementationen für nahezu alle Computer, vom PC bis zum Großrechner. Die wichtigsten Utilities unter TCP/IP zur Kommunikation mit anderen TCP/IP-Rechnern sind TELNET (Remote-Login), FTP (File-Transfer) und SMTP (Mail). Um mit möglichst vielen Netzwerkprotokollen erreicht werden zu können, hat das EDV-Zentrum auf nahezu allen seinen Maschinen TCP/IP Software installiert. Zusätzlich gibt es noch Terminalkonzentratoren oder Terminalserver, die den Zugang zu TCP/IP-Rechnern am TUNET ermöglichen. Für Rechner am TUNET, die zum Beispiel nur das DECnet Protokoll verstehen, gibt es nun die Möglichkeit mit TCP/IP-Rechnern zu reden, indem sie sich des neuen DECnet-TCP/IP Gateway-Rechners bedienen, der vom EDV-Zentrum zur Verfügung gestellt wird. Dieser Gateway-Rechner ist in der Lage, Netzwerkoperationen der einen Protokollwelt in jene der anderen zu übersetzen. Dazu bedient sich der Anwender, je nachdem von welcher Seite aus er seine Kommunikation startet, einer speziellen Syntax, die er aus der jeweiligen Rechnerdokumentation des EDV-Zentrums entnimmt. Dieser Gateway-Rechner ist per Knotenname EUTEST zu erreichen, und es ist kein Account auf der Maschine notwendig, um diese Dienste in Anspruch zu nehmen. Die TCP/IP-Protokolle, die konvertiert werden sind TELNET, FTP und SMTP. Mit Hilfe dieses Gateways und der anderen Rechner des EDV-Zentrums kann ein Benutzer von TUNET auch andere externe Netze wie UNA, EARN/BITNET, EUNET/USENET, usw. erreichen.

EUnet, your net @ work

Friedrich Plank

1. Einleitung

Die wachsende Verbreitung und Beliebtheit des Betriebssystems UNIX^{*)} ist nicht zuletzt auf den nicht von einem Hersteller eingeschränkten, regen Informationsaustausch von Anwendern und Entwicklern unter- und miteinander zurückzuführen. Ein zusätzliches Forum für diesen Informationsaustausch ist das Unix-Rechnernetz USENET in Nordamerika bzw. EUnet in Europa.

Dieser Beitrag soll einen Überblick über die augenblickliche Situation in Österreich geben. Es wird erläutert, welche Möglichkeiten das EUnet bereits jetzt bietet und welche Tendenzen sich für die Zukunft abzeichnen. Zunächst werden die Netzdienste und die Struktur des Netztes kurz erläutert.

2. Das EUnet

EUnet ist der europäische Teil von USENET, einem Netz von Rechnern unter dem Betriebssystem UNIX. Die EUnet-Aktivitäten werden von der European Unix Systems User Group (EUUG) und den jeweiligen nationalen Gruppen, in Österreich von der Unix User Group Austria (UUGA), koordiniert und teilweise auch finanziert. Das EUnet steht in Österreich allen Mitgliedern der UUGA offen.

Das EUnet dient dem Austausch von Nachrichten zwischen einzelnen Adressaten sowie als Medium für Informationen und Diskussionen von allgemeinem Interesse. Zu diesem Zweck stehen im Netz zwei verschiedene Kommunikationsformen zur Verfügung:

Die Datenpost der Mail und das Informationssystem der News.

3. Der Mail-Dienst

Die "Electronic Mail" ermöglicht es jedem UNIX-Benutzer, Meldungen an andere Teilnehmer im EUnet oder anderer nationaler oder internationaler Netze zu verschicken. Im österreichischen EUnet übernimmt 'tuvie' dabei die Rolle eines zentralen Vermittlungsrechners, der die elektronische Post der angeschlossenen Rechner weiterleitet bzw. zur Abholung bereitstellt.

Um elektronische Post zu versenden oder angekommene Meldungen zu lesen, bedient sich der Benutzer an seinem Rechner eines Programmes, des sogenannten 'Mailers', der unter UNIX in verschiedenen Ausführungen existiert. Die Basisfunktionen ermöglichen unter anderem

- einen Überblick der in der Mail-Box eingegangenen "Mails",
- ein einfaches und direktes Beantworten dieser Meldungen,
- das Ablegen und Verwalten von Meldungen in Dateien,
- das Versenden von Mail,
- private Verteiler als sogenannte "Mailing-Lists".

Die meisten Mailer verarbeiten außerdem die "Adreßköpfe", um die Meldungen weiterzuleiten. So bestehen z.B. auch Gateways zum CSnet, BITNET und anderen Netzen, die man über 'tuvie' erreichen kann. In Österreich ist neben dem bereits bestehenden Gateway ins UNA (UniversitätsNetz Austria) und zum BTX auch eine Verbindung zu einem MHS (Message Handling System), das sich an den X.400-Empfehlungen der CCITT orientiert, geplant.

4. Der News-Dienst

Der News-Dienst ist ein verteiltes Computer-Konferenzsystem, in dem die verschiedensten Themen netzweit diskutiert werden. Unter den derzeit etwa 400 im EUnet angebotenen "Newsgroups" kann jeder Benutzer "seine" Interessensgruppen herausuchen und abonnieren. Die Themen der Gruppen erstrecken sich etwa zur Hälfte auf UNIX, Computer, Technik und Peripheriegeräte im allgemeinen und speziellen, aber auch auf verschiedene andere Wissenschafts-, Gesellschafts- und Freizeitthemen. Von großer Bedeutung für direkte Anwender von Software sind die Gruppen, in denen Public Domain Software, in der Regel im source code, verteilt wird. In jeder Gruppe werden dem Leser auf einem "schwarzen Brett" die Anfragen, Antworten, Meinungen und Beiträge anderer Teilnehmer angeboten. Dabei gewinnen Sie durch die persönlichen Erfahrungen, Informationen, oder auch Public-Domain-Software anderer "Experten". Jeder Abonnent kann und sollte auch mit dem eigenen Wissen und Erfahrungen zur Diskussion beitragen, indem er selbst Beiträge in die entsprechende Newsgruppe einbringt. Die Artikel werden innerhalb der angegebenen Verbreitzungszone an alle Teilnehmer der entsprechenden Newsgruppe verteilt.

Will ein Benutzer News lesen, so ruft er das ReadNews-Programm auf, das alle neuen Artikel in der abonnierten

^{*)} UNIX ist ein eingetragenes Warenzeichen von AT&T in USA und anderen Ländern.

Newsgruppe anzeigt, die gelesen und dann gespeichert, gelöscht oder beantwortet werden können.

Die Mehrzahl der im EUnet angebotenen Newsgruppen werden weltweit verteilt, mit Schwerpunkt im amerikanischen USENET. Daneben bestehen die europäischen EUnet-Gruppen, die lokalen Gruppen, sowie die Möglichkeit, bei jeder weltweiten Gruppe auch kleinere Verbreitzonen zu wählen.

Im Anhang finden Sie eine Liste aller derzeit auf 'tuvie' verfügbaren Gruppen. Es besteht auch die Möglichkeit, sich kostenlos über diesen angebotenen Dienst einen Eindruck zu verschaffen. Ein kurzer Anruf unter 58801/3608 DW genügt um sich über die Modalitäten zu informieren.

5. Die Struktur des Netzes

Der Erfolg des USENET bzw. EUnet ist wesentlich darauf zurückzuführen, daß der Anschluß für Unix-Benutzer nur ein Minimum an Hardware (Akustikkoppler oder Telefonmodem) und Software benötigt. Die grundsätzlich ausreichenden Unix-Elemente für UUCP und Mail können bei Bedarf noch durch kostenlose Programmpakete verbessert werden.

Ursache für die schnelle Verbreitung ist auch die kooperative Struktur des Netzes. Um einen Anschluß zu bekommen, sollte man einen Nachbarn suchen, der bereits angeschlossen ist und über diesen auch die angebotenen Dienste verwenden. Die Effizienz dieses so vielfach vermaschten Netzes wird durch die zentralen Service-Leistungen des Backbones, der in Österreich von der Prozeßrechenanlage des EDV-Zentrums der TU Wien seit über drei Jahren erfolgreich betrieben wird, noch erhöht.

Eine Datenübertragung zu einem weiter entfernten Rechner geschieht in der Regel nicht direkt, sondern die Daten werden von Rechner zu Rechner bis zum Ziel weitergereicht. Als Übertragungsmedium wird in den meisten Fällen Datex-P (X.25) verwendet, zwischen den einzelnen Backbones in zunehmenden Maße auch Standleitungen. Im nationalen Bereich auch häufig das Telefonnetz.

6. Die Verbreitung des Netzes

Das USENET umfaßt im April 1989 über 11.000 offizielle Rechner in den USA, Kanada, Australien und Asien und somit ungefähr eine Million erreichbare Teilnehmer. Das erst später hinzugekommene EUnet umfaßt zum gleichen Zeitpunkt etwa 1000 angeschlossene Systeme, 36 davon in Österreich. Eine genaue Schätzung der Teilnehmerzahl ist nur schwer möglich, da sie sich täglich ändert und sich oft hinter einem angeschlossenen Rechner ein lokales Netz mit vielen Benutzern verbirgt.

7. Kosten

Für den Betrieb eines EUnet-Anschlusses treten neben den einmaligen Installationskosten in erster Linie Kommunikations-Kosten auf. In der Regel trägt jede Institution die anfallenden Kosten selbst. Bei den Backbones fallen durch das Anbieten der zentralen Dienste erhebliche Kosten an, insbesondere durch die internationale Kommunikation. Daher werden solche Kosten auf einer Volumen orientierten Verrechnungsbasis an die angeschlossenen Institutionen weiterverrechnet.

8. Tarife

Der Tarif, nach dem die Benützung unseres Backbones 'tuvie' abgerechnet wird, sieht nur eine vom Verkehrsvolumen abhängige Gebühr für den internationalen Mail-Austausch vor.

Mit Stand Mai 1989 gelten folgende Gebühren:

	Universitäten	Firmen
Österreich	—*)	—*)
Europa	1,30	1,30
Rest der Welt	4,00	4,00

Diese Gebühren werden je Kilobyte gesendeter bzw. empfangener Mail monatlich verrechnet.

*) Da innerhalb Österreichs alle angeschlossenen Rechner ihre Mails auf eigene Kosten zum Backbone übertragen bzw. von dort abholen, entstehen keine weiteren Kosten.

Für die Verwendung des News-Dienstes wird ein monatlicher Pauschalbetrag verrechnet, ohne Einfluß, ob alle News-Gruppen gelesen werden oder nur einige. Es gibt lediglich eine Staffelung zwischen Universitäten und Firmen.

Universitäten

Firmen

1.000,-

3.500,-

Für die interaktive Benützung des Backbones wird bei kommerziellen und privaten Anwendern eine Gebühr von derzeit 50,-/Loginstunde verrechnet.

9. Anschlußmöglichkeiten

Für den Anschluß an den österreichischen EUnet-Backbone 'tuvie' stehen derzeit folgende Zugangsmöglichkeiten zur Verfügung:

Telefon:	(0222) 587 46 92 bis 94	300 bd asynchron V.21
	(0222) 587 46 95 bis 96	1200/2400 bd asyn. V.22/V.22 bis
	(0222) 587 45 73	1200/75 bd asynchron V.23
Datex-P:	26221070121 (f-protol)	26221070131 (g-protol)

Neben diesen öffentlichen Zugängen ist 'tuvie' auch über das lokale **TUNET** erreichbar.

10. Ein Ausblick auf die Zukunft

Um die Weiterleitung in andere Datennetze effizienter und schneller bewerkstelligen zu können, wird in nächster Zukunft auch auf 'tuvie' eine Gateway-Funktion zu diesen Netzen realisiert werden. Europaweit ist eine TCP/IP Verbindung über Standleitungen zwischen den einzelnen Systemen geplant. Es existieren bereits solche Lösungen zwischen einigen Backbones. Das Ziel dieser Bestrebungen ist eine Eingliederung von EUnet in das amerikanische INTERNET, das ja ebenfalls auf TCP/IP aufbaut.

11. Kontaktaufnahme

Für Anfragen und Beratung stehe ich gerne zur Verfügung.

Die Kontaktadresse:

Friedrich Plank

Technische Universität Wien EDV-Zentrum/ Abt. PRA

Gußhausstraße 25 / 0203 A-1040 Wien

Tel: 0222 58801/3608 DW Fax: 0222 505 48 00

e-mail: plank@tuvie.at, postmaster@tuvie.at

ANHANG

general	Articles that should be read by everyone on your local system	comp.arch	Computer architecture.
alt.fax	Faxing documents — protocols, equipment, etc.	comp.archives	Descriptions of public access archives. (Moderated)
alt.fusion	Discussions on cold fusion.	comp.binaries.amiga	Encoded public domain programs in binary. (Moderated)
alt.gourmand	Recipes & cooking info. (Moderated)	comp.binaries.apple2	Binary-only postings for the Apple II computer.
alt.hypertext	Discussion of hypertext — uses, transport, etc.	comp.binaries.atari.st	Binary-only postings for the Atari ST. (Moderated)
alt.sources	Alternative source code, unmoderated. Caveat Emptor.	comp.binaries.ibm.pc	Binary-only postings for IBM PC/MS-DOS. (Moderated)
alt.sources.amiga	Technically-oriented Amiga PC sources. (Moderated)	comp.binaries.ibm.pc.d	Discussions about IBM/PC binary postings.
comp.ai	Artificial intelligence discussions.	comp.binaries.mac	Encoded Macintosh programs in binary. (Moderated)
comp.ai.digest	Artificial Intelligence discussions. (Moderated)	comp.bugs.2bsd	Reports of UNIX* version 2BSD related bugs.
comp.ai.edu	Applications of Artificial Intelligence to Education.	comp.bugs.4bsd	Reports of UNIX version 4BSD related bugs.
comp.ai.neural-nets	All aspects of neural networks.	comp.bugs.4bsd.ucb-fixes	Bug reports/fixes for BSD Unix. (Moderated)
comp.ai.nlang-know-rep	Neural Language and Knowledge Representation. (Moderated)	comp.bugs.misc	General UNIX bug reports and fixes (incl V7, uucp)
comp.ai.shells	Artificial intelligence applied to shells. (Moderated)	comp.bugs.sys5	Reports of USG (System III, V, etc.) bugs.
comp.ai.vision	Artificial Intelligence Vision Research. (Moderated)	comp.cog-eng	Cognitive engineering.
		comp.compilers	Compiler construction, theory, etc.

	(Moderated)		
comp.databases	Database and data management issues and theory.	comp.mail.misc	General discussions about computer mail.
comp.dcom.lans	Local area network hardware and software.	comp.mail.multi-media	Multimedia Mail.
comp.dcom.lans.hyperchannel	Hyperchannel networks within an IP network.	comp.mail.sendmail	Configuring and using the BSD sendmail agent.
comp.dcom.lans.v2lni	Proteon Pronet/V2LNI Ring networks.	comp.mail.uucp	Mail in the uucp network environment.
comp.dcom.modems	Data communications hardware and software.	comp.misc	General topics about computers not covered elsewhere.
comp.dcom.telecom	Telecommunications digest. (Moderated)	comp.music	Applications of computers in music research.
comp.doc	Archived public-domain documentation. (Moderated)	comp.newprod	Announcements of new products of interest. (Moderated)
comp.doc.techreports	Lists of technical reports. (Moderated)	comp.org.decus	DEC* Users' Society newsgroup.
comp.editors	Topics related to computerized text editing.	comp.org.fidonet	FidoNews digest, official news of FidoNet Assoc. (Moderated)
comp.edu	Computer science education.	comp.org.ieee	Issues and announcements about the IEEE & its members.
comp.edu.composition	Writing instruction in computer-based classrooms.	comp.org.usenix	USENIX Association events and announcements.
comp.emacs	EMACS editors of different flavors.	comp.org.usrgroup	News/discussion about/from the /usr/group organization.
comp.fonts	Typefonts — design, conversion, use, etc.	comp.os.aos	Topics related to Data General's AOS/VS.
comp.graphics	Computer graphics, art, animation, image processing.	comp.os.cpm	Discussion about the CP/M operating system.
comp.graphics.digest	Graphics software, hardware, theory, etc. (Moderated)	comp.os.cpm.amethyst	Discussion of Amethyst, CP/M-80 software package.
comp.ivideodisc	Interactive videodiscs — uses, potential, etc.	comp.os.eunice	The SRI Eunice system.
comp.lang.ada	Discussion about Ada*.	comp.os.minix	Discussion of Tanenbaum's MINIX system.
comp.lang.apl	Discussion about APL.	comp.os.misc	General OS-oriented discussion not carried elsewhere.
comp.lang.c	Discussion about C.	comp.os.os9	Discussions about the os9 operating system.
comp.lang.c++	The object-oriented C++ language.	comp.os.research	Operating systems and related areas. (Moderated)
comp.lang.clu	The CLU language & related topics. (Moderated)	comp.os.rsts	Topics related to the PDP-11 RSTS/E operating system.
comp.lang.eiffel	The object-oriented Eiffel language.	comp.os.v	The V distributed operating system from Stanford.
comp.lang.forth	Discussion about Forth.	comp.os.vms	DEC's VAX* line of computers & VMS.
comp.lang.forth.mac	The CSI MacForth programming environment.	comp.os.xinu	The XINU operating system from Purdue (D. Comer).
comp.lang.fortran	Discussion about FORTRAN.	comp.parallel	Massively parallel hardware/software. (Moderated)
comp.lang.icon	Topics related to the ICON programming language.	comp.periphs	Peripheral devices.
comp.lang.idl	IDL (Interface Description Language) related topics.	comp.periphs.printers	Information on printers.
comp.lang.lisp	Discussion about LISP.	comp.protocols.appletalk	Applebus hardware & software.
comp.lang.lisp.franz	The Franz Lisp programming language.	comp.protocols.ibm	Networking with IBM mainframes.
comp.lang.lisp.x	The XLISP language system.	comp.protocols.iso	The ISO protocol stack.
comp.lang.misc	Different computer languages not specifically listed.	comp.protocols.iso.dev-envirion	The ISO Development Environment.
comp.lang.modula2	Discussion about Modula-2.	comp.protocols.iso.x400	X400 mail protocol discussions. (Moderated)
comp.lang.pascal	Discussion about Pascal.	comp.protocols.iso.x400.gateway	X400 mail gateway discussions. (Moderated)
comp.lang.postscript	The PostScript Page Description Language.	comp.protocols.kerberos	The Kerberos authentication server.
comp.lang.prolog	Discussion about PROLOG.	comp.protocols.kermit	Info about the Kermit package. (Moderated)
comp.lang.rexx	The REXX command language.	comp.protocols.misc	Various forms and types of FTP protocol.
comp.lang.scheme	The Scheme Programming language.	comp.protocols.nfs	Discussion about the Network File System protocol.
comp.lang.scheme.c	The Scheme language environment.	comp.protocols.pcnet	Topics related to PCNET (a personal computer network).
comp.lang.sigplan	Info & announcements from ACM SIGPLAN. (Moderated)	comp.protocols.pup	The Xerox PUP network protocols.
comp.lang.smalltalk	Discussion about Smalltalk 80.	comp.protocols.tcp-ip	TCP and IP network protocols.
comp.lang.visual	Visual programming languages.	comp.protocols.tcp-ip.domains	Topics related to Domain Style names.
comp.laser-printers	Laser printers, hardware & software. (Moderated)		
comp.lsi	Large scale integrated circuits.		
comp.lsi.cad	Electrical Computer Aided Design.		
comp.mail.elm	Discussion and fixes for ELM mail system.		
comp.mail.headers	Gatewayed from the ARPA header-people list.		
comp.mail.mh	The UCI version of the Rand Message Handling system.		

- comp.protocols.tcp-ip.ibmpc TCP/IP for IBM(-like) personal computers.
- comp.risks Risks to the public from computers & users. (Moderated)
- comp.simulation Simulation methods, problems, uses. (Moderated)
- comp.society The impact of technology on society. (Moderated)
- comp.society.futures Events in technology affecting future computing.
- comp.society.women Women's roles and problems in computing (Moderated)
- comp.soft-sys.andrew The Andrew system from CMU.
- comp.software-eng Software Engineering and related topics.
- comp.sources.amiga Source code-only postings for the Amiga. (Moderated)
- comp.sources.atari.st Source code-only postings for the Atari ST. (Moderated)
- comp.sources.bugs Bug reports, fixes, discussion for posted sources
- comp.sources.d For any discussion of source postings.
- comp.sources.games Postings of recreational software. (Moderated)
- comp.sources.games.bugs Bug reports and fixes for posted game software.
- comp.sources.mac Software for the Apple Macintosh. (Moderated)
- comp.sources.misc Posting of software. (Moderated)
- comp.sources.sun Software for Sun workstations. (Moderated)
- comp.sources.unix Postings of complete, UNIX-oriented sources. (Moderated)
- comp.sources.wanted Requests for software and fixes.
- comp.sources.x Software for the X windows system. (Moderated)
- comp.std.c Discussion about C language standards.
- comp.std.internat Discussion about international standards.
- comp.std.misc Discussion about various standards.
- comp.std.mumps Discussion for the X11.1 committee on Mumps. (Moderated)
- comp.std.unix Discussion for the P1003 committee on UNIX. (Moderated)
- comp.sys.amiga Commodore Amiga: info&uses, but no programs.
- comp.sys.amiga.tech Technical discussion about the Amiga.
- comp.sys.apollo Apollo computer systems.
- comp.sys.apple Discussion about Apple micros.
- comp.sys.atari.8bit Discussion about 8 bit Atari micros.
- comp.sys.atari.st Discussion about 16 bit Atari micros.
- comp.sys.att Discussions about AT&T microcomputers.
- comp.sys.cbm Discussion about Commodore micros.
- comp.sys.cdc Control Data Corporation Computers (e.g., Cybers).
- comp.sys.celerity Celerity Computers
- comp.sys.dec Discussions about DEC computer systems.
- comp.sys.dec.micro DEC Micros (Rainbow, Professional 350/380)
- comp.sys.encore Encore's MultiMax computers.
- comp.sys.handhelds Handheld computers and programmable calculators.
- comp.sys.hp Discussion about Hewlett-Packard equipment.
- comp.sys.ibm.pc Discussion about IBM personal computers.
- comp.sys.ibm.pc.digest The IBM PC, PC-XT, and PC-AT. (Moderated)
- comp.sys.ibm.pc.rt Topics related to IBM's RT computer.
- comp.sys.intel Discussions about Intel systems and parts.
- comp.sys.intel.ipsc310 Anything related to Xenix on an Intel 310.
- comp.sys.m6809 Discussion about 6809's.
- comp.sys.m68k Discussion about 68k's.
- comp.sys.m68k.pc Discussion about 68k-based PCs. (Moderated)
- comp.sys.mac Discussions about the Apple Macintosh & Lisa.
- comp.sys.mac.digest Apple Macintosh: info&uses, but no programs. (Moderated)
- comp.sys.mac.hypercard The Macintosh Hypercard: info & uses.
- comp.sys.mac.programmer Discussion by people programming the Apple Macintosh.
- comp.sys.masscomp The Masscomp line of computers. (Moderated)
- comp.sys.misc Discussion about computers of all kinds.
- comp.sys.next Discussion about the new NeXT computer.
- comp.sys.northstar Northstar microcomputer users.
- comp.sys.nsc.32k National Semiconductor 32000 series chips.
- comp.sys.proteon Proteon gateway products.
- comp.sys.pyramid Pyramid 90x computers.
- comp.sys.ridge Ridge 32 computers and ROS.
- comp.sys.sequent Sequent systems, (Balance and Symmetry).
- comp.sys.sgi Silicon Graphics's Iris workstations and software.
- comp.sys.sun Sun "workstation" computers. (Moderated)
- comp.sys.super Supercomputers.
- comp.sys.tahoe CCI 6/32, Harris HCX/7, & Sperry 7000 computers.
- comp.sys.tandy Discussion about TRS-80's.
- comp.sys.ti Discussion about Texas Instruments.
- comp.sys.ti.explorer The Texas Instruments Explorer.
- comp.sys.transputer The Transputer computer and OCCAM language.
- comp.sys.workstations Various workstation-type computers. (Moderated)
- comp.sys.xerox Xerox 1100 workstations and protocols.
- comp.sys.zenith Heath terminals and related Zenith products.
- comp.sys.zenith.z100 The Zenith Z-100 (Heath H-100) family of computers.
- comp.terminals All sorts of terminals.
- comp.terminals.bitgraph The BB&N BitGraph Terminal.
- comp.terminals.tty5620 AT&T Dot Mapped Display Terminals (5620 and BLIT).
- comp.text Text processing issues and methods.
- comp.text.desktop Technology & techniques of desktop publishing.
- comp.theory Theoretical Computer Science.
- comp.theory.cell-automata Discussion of all aspects of cellular automata.
- comp.theory.dynamic-sys Ergodic Theory and Dynamical Systems.
- comp.theory.info-retrieval Information Retrieval topics. (Moderated)
- comp.theory.self-org-sys Topics related to self-organization.
- comp.unix Discussion of UNIX* features and bugs. (Moderated)
- comp.unix.aux The version of UNIX for Apple Macintosh II computers.
- comp.unix.cray Cray computers and their operating systems.
- comp.unix.i386 Versions of Unix running on Intel 80386-based boxes.
- comp.unix.microport Discussion of Microport's UNIX.
- comp.unix.questions UNIX neophytes group.
- comp.unix.ultrix Discussions about DEC's Ultrix.
- comp.unix.wizards Discussions, bug reports, and fixes on and for

	UNIX.		
comp.unix.xenix	Discussion about the Xenix OS.	news.announce.conferences	Calls for papers and conference announcements. (Moderated)
comp.windows.misc	Various issues about windowing systems.	news.announce.important	General announcements of interest to all. (Moderated)
comp.windows.ms	Window systems under MS/DOS.	news.announce.newusers	Explanatory postings for new users. (Moderated)
comp.windows.news	Sun Microsystems' NewS window system.	news.config	Postings of system down times and interruptions.
comp.windows.x	Discussion about the X Window System.	news.groups	Discussions and lists of newsgroups.
eunet.bugs.4bsd	Bug reports on 4.*bsd, limited (EUnet) "audience".	news.lists	News-related statistics and lists. (Moderated)
eunet.bugs.uucp	Bug reports on uucp, specifically EUUG uucp.	news.misc	Discussions of USENET itself.
eunet.esprit	Discussion/info about Esprit projects.	news.software.b	Discussion about B-news-compatible software.
eunet.general	General discussions.	news.software.nntp	The Network News Transfer Protocol.
eunet.followup	Followup group for eunet.general.	news.software.notes	Notesfile software from the Univ. of Illinois.
eunet.jokes	Jokes only Europeans can (do?) understand...	news.sysadmin	Comments directed to system administrators.
eunet.micro.acorn	Discussion about Acorn micros.	rec.arts.books	Books of all genres, and the publishing industry.
eunet.newprod	Announcements of new products of European interest.	rec.arts.movies	Discussions of movies and movie making.
eunet.news	Group for news topics, limited to EUnet.	rec.arts.movies.reviews	Reviews of movies. (Moderated)
eunet.news.group	Discussion on and proposals for new eunet newsgroups.	rec.arts.sf-lovers	Science fiction lovers' newsgroup.
eunet.politics	(European) political discussions (and flames!).	rec.games.board	Discussion and hints on board games.
eunet.sources	EUnet-wide (only!) group for posting sources.	rec.games.bridge	Hobbyists interested in bridge.
eunet.test	EUnet-wide test group.	rec.games.chess	Chess & computer chess.
eunet.works	Workstations (specifically European) topics.	rec.games.empire	Discussion and hints about Empire.
gnu.announce	Status and announcements from the Project. (Moderated)	rec.games.frp	Discussion about Fantasy Role Playing games.
gnu.chess	Discussion of the GNU Chess program	rec.games.go	Discussion about Go.
gnu.config	GNU's Not Usenet administration and configuration	rec.games.hack	Discussion, hints, etc. about the Hack game.
gnu.emacs	Editor/development environment and occasional sources	rec.games.mise	Games and computer games.
gnu.emacs.bug	GNU Emacs bug reports and suggested fixes	rec.games.moria	Comments, hints, and info about the Moria game.
gnu.emacs.gnus	News reading under GNU Emacs using GNUS (in English).	rec.games.pbm	Discussion about Play by Mail games.
gnu.emacs.lisp.manual	Working group on the GNU Emacs LISP Programmer's Manual	rec.games.programmer	Discussion of adventure game programming.
gnu.emacs.vms	Discussion of the VMS port of GNU Emacs	rec.games.rogue	Discussion and hints about Rogue.
gnu.g++	Discussion of the GNU C++ Compiler and GDB+ Debugger.	rec.games.trivia	Discussion about trivia.
gnu.g++.bug	G++ and GDB+ bug reports and suggested fixes.	rec.games.vectrex	The Vectrex game system.
gnu.g++.lib.bug	G++ library bug reports and suggested fixes.	rec.games.video	Discussion about video games.
gnu.gcc	Discussion of the GNU C Compiler	rec.ham-radio	Amateur Radio practices, contests, events, rules, etc.
gnu.gcc.bug	GNU C Compiler bug reports and suggested fixes.	rec.ham-radio.packet	Discussion about packet radio setups.
gnu.gdb.bug	GNU DeBugger bug reports and suggested fixes	rec.ham-radio.swap	Offers to trade and swap radio equipment.
gnu.ghostscript.bug	GNU Ghostscript interpreter bugs.	rec.humor.funny	Jokes that are funny (in the moderator's opinion). (Moderated)
gnu.test	GNU's Not Usenet alternative sub-network testing	rec.mag	Magazine summaries, tables of contents, etc.
gnu.utils.bug	Bugs in GNU utility programs (e.g., gnumake)	rec.mag.fsfnet	A Science Fiction "fanzine." (Moderated)
misc.jobs.misc	Discussion about employment, workplaces, careers.	rec.mag.otherrealms	Edited science fiction & fantasy "magazine". (Moderated)
misc.jobs.offered	Announcements of positions available.	rec.music.beatles	Postings about the Fab Four & their music.
misc.jobs.resumes	Postings of resumes and "situation wanted" articles.	rec.music.bluenote	Discussion of jazz, blues, and related types of music.
misc.misc	Various discussions not fitting in any other group.	rec.music.cd	CDs --- availability and other discussions.
misc.security	Security in general, not just computers. (Moderated)	rec.music.classical	Discussion about classical music.
misc.wanted	Requests for things that are needed (NOT software).	rec.music.dementia	Discussion of comedy and novelty music.
news.admin	Comments directed to news administrators.	rec.music.folk	Folks discussing folk music of various sorts.
		rec.music.gaffa	Progressive music (e.g., Kate Bush). (Moderated)
		rec.music.gdead	A group for (Grateful) Dead-heads.
		rec.music.makers	For performers and their discussions.
		rec.music.misc	Music lovers' group.
		rec.music.synth	Synthesizers and computer music.

sci.astro	Astronomy discussions and information.	soc.culture.jewish	Jewish culture & religion. (cf. talk.politics.mideast)
sci.bio	Biology and related sciences.	soc.culture.misc	Group for discussion about other cultures.
sci.bio.technology	Any topic relating to biotechnology.	soc.culture.turkish	Discussion about things Turkish.
sci.chem	Chemistry and related sciences.	soc.human-nets	Computer aided communications digest. (Moderated)
sci.crypt	Different methods of data en/decryption.	soc.politics	Political problems, systems, solutions. (Moderated)
sci.edu	Science education.	soc.politics.arms-d	Arms discussion digest. (Moderated)
sci.electronics	Circuits, theory, electrons and discussions.	unix-pc.bugs	Bug reports, fixes & workarounds.
sci.environment	Discussions about the environment and ecology.	unix-pc.general	General information and discussion.
sci.lang	Natural languages, communication, etc.	unix-pc.sources	Source code to various programs.
sci.lang.japan	The Japanese language, both spoken and written.	unix-pc.test	Test group.
sci.logic	Logic — math, philosophy & computational aspects.	unix-pc.uucp	Configuration and management of uucp on Unix-PCs.
sci.math	Mathematical discussions and pursuits.	bionet.general	General BIONET announcements.
sci.math.num-analysis	Numerical Analysis.	bionet.sci-resources	Information about funding agencies, etc.
sci.math.stat	Statistics discussion.	bionet.jobs	Scientific Job opportunities.
sci.math.symbolic	Symbolic algebra discussion.	bionet.journals.contents	Contents of biology journal publications.
sci.med	Medicine and its related products and regulations.	bionet.molbio.ageing	Discussions of cellular and organismal ageing.
sci.med.aids	AIDS: treatment, pathology/biology of HIV, prevention. (Moderated)	bionet.molbio.news	Research news of interest to the community.
sci.military	Discussion about science & the military. (Moderated)	bionet.molbio.methods-reagents	Requests for information and lab reagents.
sci.misc	Short-lived discussions on subjects in the sciences.	bionet.molbio.genbank	Info about the GenBank Nucleic acid database.
sci.nanotech	Self-reproducing molecular-scale machines. (Moderated)	bionet.molbio.embl databank	Info about the EMBL Nucleic acid database.
sci.philosophy.meta	Discussions within the scope of "MetaPhilosophy."	bionet.molbio.pir	Info about the PIR protein sequence database.
sci.philosophy.tech	Technical philosophy: math, science, logic, etc.	bionet.molbio.bio-matrix	Computer applications to biological databases.
sci.physics	Physical laws, properties, etc.	bionet.molbio.evolution	How genes and proteins have evolved.
sci.psychology	Topics related to psychology.	bionet.molbio.gene-express	How genes are regulated in cells.
sci.research	Research methods, funding, ethics, and whatever.	bionet.molbio.gene-org	How genes are organized on chromosomes.
sci.space	Space, space programs, space related research, etc.	bionet.molbio.oncogenes	Genes that cause cancer.
sci.space.shuttle	The space shuttle and the STS program.	bionet.molbio.plant	Molecular biology of plants.
soc.culture.african	Discussions about Africa & things African.	bionet.molbio.proteins	Research on proteins and protein databases.
soc.culture.arabic	Technological & cultural issues, *not* politics.	bionet.molbio.swiss-prot	Discussion on the SWISS-PROT Database.
soc.culture.celtic	Group about Celts (*not* basketball!).	bionet.molbio.yeast	Molecular biology of yeast.
soc.culture.china	About China and Chinese culture.	bionet.population-bio	Technical discussions about population biology.
soc.culture.esperanto	The neutral international language Esperanto.	bionet.software	Information about software for biology.
soc.culture.greek	Group about Greeks.	bionet.software.pc	Info on PC software for scientists.
soc.culture.indian	Group for discussion about India & things Indian.	bionet.software.pc.comm	Info on PC communications software.
soc.culture.japan	Everything Japanese, except the Japanese language.	bionet.software.contrib	Info on programs contributed to BIONET.
		bionet.technology.conversion	Use of technology to convert waste and biomass.

Konzept für ein X.25 - Netz für Österreichs Universitäten *)

Walter Kunft

1. Ausgangssituation

Zur Abdeckung des nationalen und internationalen Kommunikationsbedarfes österreichischer Universitätsinstitutionen wurden in den letzten Jahren in Österreich verschiedene überregionale Datennetze in Betrieb genommen. Zum einen wurde dadurch die Teilnahme an weltweiten Datennetzen des Wissenschafts- und Forschungsbereiches wie EARN (European Academic and Research Network) und EUnet (European UNIX systems network) ermöglicht, zum anderen wurde mit der Schaffung von zunächst noch herstellerabhängigen nationalen Netzen, wie UNA (Universitätsnetz Austria, Verbund von DEC-Systemen) und ATACNET (Austrian Academic Network, Verbund von Systemen mit IBM-Betriebssystemen) die Basis für die Kommunikation zwischen österreichischen Universitäten gelegt. In diese Netze sind auch Forschungsinstitutionen und das Bundesministerium für Wissenschaft und Forschung eingebunden.

Der 1986 gegründete ACONET-Verein hat die Koordination der Datennetze im Universitätsbereich und ihre Weiterentwicklung bzw. Überführung in ein herstellerunabhängiges Akademisches Computernetz nach den Regeln der Open Systems Interconnection von ISO zum Ziel. Der ACONET-Verein ist Mitglied von RARE (Reseaux Associes pour la Recherche Europeenne) damit dieses Ziel in Koordination mit diesbezüglichen gesamteuropäischen Bestrebungen im Rahmen von COSINE (Cooperation for OSI Networking in Europe) erreicht werden kann. Darüber hinaus arbeitet der ACONET-Verein sehr intensiv mit anderen europäischen Gremien zusammen, die die gleiche Zielsetzung haben, z.B. mit dem DFN-Verein (Verein zur Förderung eines Deutschen Forschungsnetzes).

2. Konzept für ein Trägernetz für ACONET

2.1 Protokolle und Dienste

Auf dem Trägernetz von ACONET müssen sämtliche OSI-Dienste aufgesetzt werden können. Diese Forderung wird durch ein Netz, das mit X.25-Protokollen betrieben wird, erfüllt. Auf diesem X.25-Trägernetz können aber nicht nur spätere OSI-Dienste, sondern auch gegenwärtig verwendete andere Kommunikationsprotokolle wie DECnet, SNA und TCP/IP aufsetzen. Damit ist ein solches Netz in der Lage, bestehende herstellerepezifische Kommunikationsarchitekturen zu tragen, LANs in verschiedenen Universitäten zu verbinden und als Basis für OSI-Dienste gemäß der Zielsetzung von ACONET zu dienen.

2.2 Topologie

In den nächsten Jahren wird zwischen österreichischen Universitätsinstitutionen voraussichtlich ein Datenvolumen von 5 bis 10 GByte im Monat befördert werden müssen. (In Deutschland haben gleichartige Schätzungen des DFN-Vereins ein Informationsvolumen von etwa 200 GByte pro Monat ergeben). Dieses Datenvolumen umfaßt

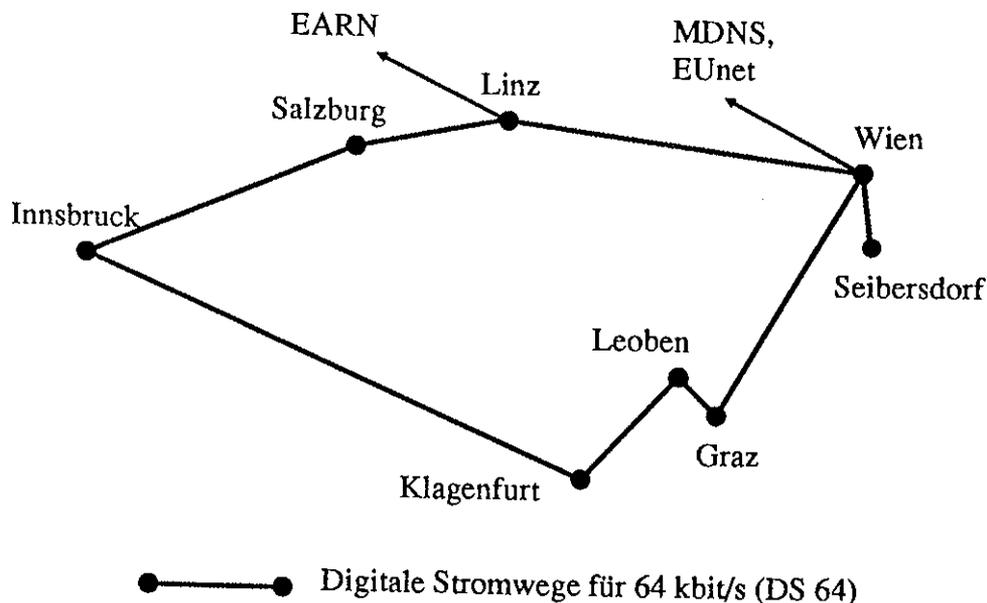


Abbildung 1: ACONET-Trägernetz

*) erstellt im Auftrag des ACONET-Vereins

einerseits die Kommunikation zwischen inländischen Kommunikationspartnern und den größten Teil der Kommunikation mit ausländischen Institutionen. Letzterer Anteil kommt dadurch zustande, daß der Zugang zu allen internationalen Datennetzen über einzelne Backbone-Systeme erfolgt, die die Weiterleitung der Information ins Ausland bzw. die Verteilung von Information aus dem Ausland im Inland durchführen. Daher muß die Information, die ins Ausland gesendet werden soll, zunächst im Inland zum betreffenden Backbone-System transportiert werden. Aus dem Ausland empfangene Information trifft beim Backbone-System ein und muß im Inland zum eigentlichen Empfänger gebracht werden. Dieses Informationsvolumen von 5 bis 10 GByte pro Monat kann durch das öffentliche Datenpaketvermittlungsnetz DATEX-P nicht mehr mit vertretbarem Kostenaufwand befördert werden. Allein die Verkehrsgebühren für die Übertragung von 5 GByte im Monat würden bei Annahme günstigster Verhältnisse (Segmente vollgefüllt) nach den derzeitigen DATEX-P-Tarifen etwa 1,17 Mill. öS monatlich betragen. Aus diesem Grunde ist für ACONET ein Standleitungsnetz erforderlich, das die Universitätsstandorte in ganz Österreich verbindet (Siehe Abb. 1).

Das Trägernetz muß hinsichtlich seines Durchsatzes flexibel ausgebaut werden können. In einer ersten Ausbaustufe werden DDL-Verbindungen für Übertragungsgeschwindigkeiten von 9600 bit/s verwendet, um die einzelnen Universitätsstandorte miteinander zu verbinden. Diese Verbindungen können dann dort, wo es das zu transferierende Informationsvolumen erfordert, durch leistungsfähige digitale Stromwege für 64 kbit/s (Dienst DS 64 der ÖPTV) ersetzt werden, so daß im Laufe der Zeit ein DS-64-Verbund entstehen wird. Die Ringstruktur des Trägernetzes sorgt für die Existenz von Alternativwegen bei Leitungs- oder Knotenausfällen.

In jedem Universitätsstandort sorgt ein leistungsfähiger privater X.25-Vermittlungsknoten (im folgenden mit Standort-Knoten oder kurz S-Knoten bezeichnet) für den Anschluß der Universitäten dieses Standortes an ACONET und für die Verbindung mit anderen Universitätsstandorten. An die S-Knoten sind X.25-Vermittlungsknoten der einzelnen Universitäten bzw. Organisationen eines Standortes (im folgenden als End-Knoten oder kurz E-Knoten bezeichnet) mit 9,6 oder 64 kbit/s angeschlossen. Die E-Knoten führen die einzelnen Institutionen der Universitäten an ACONET heran (siehe Abb. 2). An E-Knoten können Einzelsysteme, PADs, Gateways zu LANs usw. je nach den Erfordernissen des jeweiligen Bereiches angeschlossen werden. An Standorten, in denen nur eine Universität bzw. Organisation anzuschließen ist, kann der Anschluß der betreffenden Institutionen auch direkt an den S-Knoten erfolgen. Bei Bedarf können zwischen S-Knoten und zwischen S- und E-Knoten auch mehrere 9,6- oder 64 kbit/s-Verbindungen geschaltet werden.

Zur effizienten Einbindung von LANs in ACONET ist die Errichtung von Gateways oder der Einsatz von LAN-Routern, die TCP/IP über X.25-Verbindungen abwickeln können, notwendig. Langfristig gesehen sollte X.25 auch in den LANs eingeführt werden.

2.3 Vermittlungsknoten

S- und E-Knoten müssen in kleinen Stufen (z.B. 4 Ports) bis etwa 20 Ports ausbaubar sein, wobei alle Ports im Interesse einer flexiblen Übergangsmöglichkeit von DDL 9600 auf DS 64 mit Geschwindigkeiten von 1200 bit/s bis 64 kbit/s übertragen können und V.24 bzw. V.35 Schnittstellen aufweisen müssen. Die Durchsatzleistung der S-

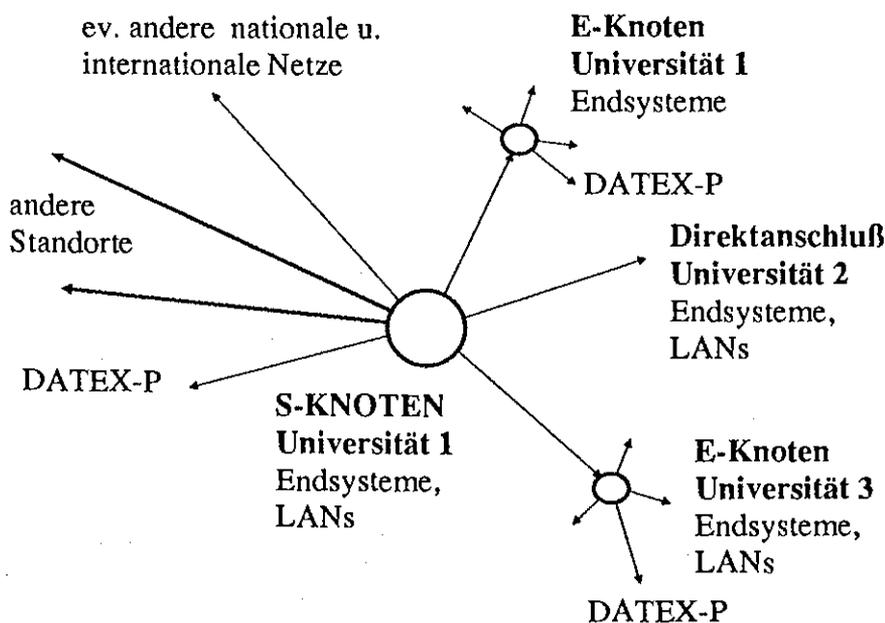


Abbildung 2: Standortkonzept

Knoten muß, ebenfalls aus Gründen einer problemlosen Erweiterungsmöglichkeit auf DS 64 zwischen 800 und 1000 Paketen (zu 128 Bytes) pro Sekunde liegen, die Zahl der möglichen Rufe sollte 20 pro Sekunde betragen. E-Knoten sollten nach Möglichkeit dieselbe Leistung aufweisen, jedoch sind auch geringere Leistungswerte zulässig. S- und E-Knoten sollten so beschaffen sein, daß für beide Kategorien ein gemeinsames, zentrales Netzmanagement durchgeführt werden kann. In den Phasen der ersten Ausbaustufe, in denen DDL 9600 verwendet wird, können auch S-Knoten die erwähnte geringere Leistungsfähigkeit aufweisen. Beim Übergang auf DS 64 sind sie jedoch durch S-Knoten mit der obigen Vermittlungsleistung zu ersetzen. Sie können dann als E-Knoten weiter verwendet werden.

Die Software in den Vermittlungsknoten muß in der Lage sein, X.25 Version 1984, abzuwickeln.

2.4 Teilnehmende Institutionen (1. Ausbaustufe)

Tabelle 1 zeigt die Universitäten bzw. Organisationen, die in der ersten Ausbaustufe in ACONET integriert werden sollen, die Art der zu errichtenden Knoten und die Zahl der Anschlüsse, die in der ersten Ausbaustufe realisiert werden sollen.

Standort	Organisation	Knoten	Anschlüsse
Wien	Universität	E	8
	Technische Universität	S	20
	Universität für Bodenkultur	E	6
	Veterinärmedizinische Universität	E	6
	Wirtschaftsuniversität	E	8
	Akademie der Wissenschaften	E	8
	Interuniversitäres EDV-Zentrum	E	8
	Atominstitut	E	6
	Bundesvers.- u. Forschungsanstalt Arsenal	E	6
	Bundesmin. f. Wissenschaft und Forschung	E	6
Seibersdorf	Österr. Forschungszentrum	S	8
Graz	Universität	E	8
	Technische Universität	S	8
	Forschungsgesellschaft Joanneum	E	6
Leoben	Montanistische Universität	S	8
Klagenfurt	Universität f. Bildungswissenschaften	S	8
Innsbruck	Universität	S	8
Salzburg	Universität	S	8
Linz	Universität	S	8

Tabelle 1: Anschlußkonzept der Ausbaustufe 1

Andere wissenschaftliche Organisationen, wie z.B. die Musik- und Kunstakademien sollen in einer späteren Ausbaustufe integriert werden. In der ersten Ausbaustufe werden an allen in Tabelle 1 unter Anschlußart mit E bezeichneten Universitäten E-Knoten errichtet. Anschlußart S bedeutet, daß die Institutionen dieser Universität/Organisation in der ersten Ausbaustufe direkt an den S-Knoten angeschlossen werden und zwar mit Geschwindigkeiten, wie sie für die einzelnen Endsysteme geeignet sind (1,2 bis 64 kbit/s).

2.5 Verbindung zu anderen Netzen

2.5.1 Verbindung zu DATEX-P

Die Verbindung von ACONET zum öffentlichen Paketvermittlungsnetz DATEX-P der ÖPTV wäre durch eine X.75-Schnittstelle am besten realisierbar. Privatnetze werden derzeit von der Post jedoch nicht über X.75 sondern ausschließlich über X.25 an DATEX-P angeschlossen. Daher muß die Heranführung der Universitätsinstitutionen an DATEX-P über die S-Knoten in den Universitätsstandorten und über die E-Knoten der Universitäten/Organisationen erfolgen. Jede Universität sollte über einen eigenen DATEX-P-Anschluß verfügen, der vom E- oder, wenn sie über den S-Knoten integriert ist, vom S-Knoten zum DATEX-P führen soll (siehe Abb. 2). An allen Verbindungspunkten

zwischen ACONET und DATEX-P ist eine geeignete Kostenerfassung und Verrechnung der Verkehrskosten von DATEX-P durchzuführen.

2.5.2 Verbindung zu internationalen Netzen

Eine weitere wichtige Forderung an das Trägernetz von ACONET ist die Anschließbarkeit an das europäische MDNS (Managed Data Network System), das im Rahmen von COSINE errichtet und verschiedene nationale Netze des Wissenschaftsbereiches zu einem europäischen Verbund zusammenschließen wird. Da es sich bei MDNS auch um ein X.25-Netz handelt, ist diese Forderung erfüllt, wenn das Trägernetz von ACONET auf X.25 basiert.

Auch alle anderen Datennetze des wissenschaftlichen Bereiches (EARN, EUnet, HEPNET, ARPA-Internet usw.) werden über Gateways, die teilweise bereits existieren, erreichbar sein.

Derzeit wird X.400 über den internationalen Gateway an der TU-Graz abgewickelt, der als Administration Management Domain (ADMD) fungiert. Da die Radio Austria AG die X.400-ADMD für Österreich übernehmen wird, werden Verbindungen zwischen dem ACONET-Trägernetz und der Radio Austria AG geschaffen werden.

3. Kosten

In den folgenden Kostenabschätzungen sind die Aufwendungen für die Integration des Österreichischen Forschungszentrums Seibersdorf, der Forschungsgesellschaft Joanneum und der Bundesversuchs- und Forschungsanstalt Arsenal nicht berücksichtigt.

3.1 Leitungskosten

Im folgenden wurden die monatlichen Leitungskosten unter Voraussetzung der Ringstruktur gemäß Abb. 1 errechnet. Dazu wurden noch insgesamt 10 Standverbindungen gerechnet, um innerhalb der Standorte Graz und Wien einzelne E-Knoten mit dem jeweiligen S-Knoten und den S-Knoten in Wien mit der Radio Austria AG zu verbinden.

3.1.1 DDL 9600

Beim Einsatz von DDL 9600 errechnen sich die Leitungskosten für die Ringkonfiguration gemäß Abb. 1 wie folgt:

- 14 x Fernbereich (je 9.800 öS)
 - 20 x Ortsbereich (je 5.200 öS)
- abz. 40% Ermäßigung 1.737,- köS/Jahr

3.1.2 DS 64

Folgende Tabelle zeigt die monatlichen Kosten für die DS-64-Verbindungen zwischen den Universitätsstandorten und innerhalb derselben.

Strecke	Entfernung (km)	köS/Monat	köS/Jahr
Wien - Linz	150	43,-	
Linz - Salzburg	110	40,2	
Salzburg - Innsbruck	140	42,3	
Innsbruck - Klagenfurt	240	49,3	
Klagenfurt - Leoben	100	39,5	
Leoben - Graz	45	29,5	
Graz - Wien	145	42,7	
		286,5	
10 DS 64 Verbindungen zu 3 km innerhalb der Standorte		50,-	
		336,5	
* 1,25 für Mitbenützung durch andere (ev. -40% Ermäßigung)		420,6	5.048,-
		252,4	3.029,-

Ob für die Benützung der DS 64 Verbindungen durch verschiedene Universitäten der 1,25- fache Betrag der Gebühren zu zahlen ist, ist noch zu klären. Zu klären ist auch, ob von der ÖPTV für den öffentlichen Dienst für DS 64 dieselben Ermäßigungen gewährt werden können, wie für Standleitungen und für DDL.

3.2 Kosten der Vermittlungseinrichtungen

Folgende Aufstellung gibt einen Überblick über die Kosten der notwendigen Vermittlungseinrichtungen, um das Konzept gemäß Punkt 2 realisieren zu können (alle Preise sind in kÖS ohne MWSt. angegeben). Die hier angeführten Vermittlungsknoten sind in der Lage, über alle Ports mit einer Geschwindigkeit von 64 kbit/s zu übertragen.

S-Knoten	
• Grundgerät m. 4 Ports (ausbaubar bis 20 Ports)	140,-
• 4 Ports Erweiterung	48,-
E-Knoten	
• Grundgerät m. 2 Ports (ausbaubar bis 10 Ports)	60,-
• 4 Ports Erweiterung	38,-
V.35/V.36-Konverter	6,8
Adapterkabel f. V.35	2,4
LAN-Router (TCP/IP-X.25)	222,-

4. Realisierungsphasen der ersten Ausbaustufe und ihre Kosten

Die erste Ausbaustufe gemäß Punkt 2 kann in folgenden Schritten realisiert werden:

4.1 Phase 1: Realisierung des südlichen Teiles des Ringes.

In Phase 1 der ersten Ausbaustufe werden die Standorte Klagenfurt, Leoben und Graz unter Verwendung von DDL 9600 Verbindungen und Vermittlungsknoten an Wien herangeführt. Die Realisierung dieses Teiles des Ringes ist deswegen von höherer Priorität als die der übrigen Teile, weil die Universitäten im Süden über keine Standverbindungen nach Wien verfügen. In dieser Phase wird auch eine Verbindung des Wiener S-Knotens mit der Radio Austria AG (X.400-ADMD) realisiert. Die Radio Austria AG hat den Universitäten für einen noch nicht definierten Übergangszeitraum die kostenlose Benützung ihrer Anlagen für einen X.400-Testbetrieb angeboten.

Kosten der Phase 1:

- Leitungskosten	
• 6 x Fernbereich (zu je 9.800)	
• 4 x Ortsbereich (zu je 5.200)	
abz. 40% Ermäßigung	573,- kÖS/Jahr
- Einmalkosten	
• 4 S-Knoten mit je 8 Ports	
• 1 E-Knoten mit 8 Ports	
Einmalkosten Phase 1:	1.128,- kÖS

4.2 Phase 2: Realisierung des vollen Ringes

In dieser Realisierungsphase wird der Ring komplettiert, so daß alle Standorte mit X.25 über DDL 9600 verbunden werden können. Nach der Umstellung derzeit unter Verwendung eigener Stromwege laufender Applikationen auf X.25 können bestehende DDL 9600 Verbindungen zwischen Universitätsstädten aufgegeben werden, so daß die folgenden Kosten nur für einen Umstellungszeitraum von maximal einem Jahr in voller Höhe zum Tragen kommen. Nach dem Umstellungszeitraum fallen zusätzliche Kosten nur mehr zwischen jenen Städten an, zwischen denen derzeit noch keine Standverbindungen existieren. Soweit derzeit abgeschätzt werden kann, ist eine Umstellung der mit SNA kommunizierenden IBM-Systeme auf X.25 technisch gesehen problemlos möglich. Dadurch kann auch der EARN-Verkehr zwischen diesen Systemen auf X.25 umgestellt werden.

Kosten der Phase 2:

- Leitungskosten	
• 14 x Fernbereich (zu je 9.800)	
• 4 x Ortsbereich (zu je 5.200)	
abz. 40% Ermäßigung	1.138,- kÖS/Jahr
- Einmalkosten (zusätzlich zu Phase 1)	
• 3 S-Knoten mit je 8 Ports	
Einmalkosten Phase 2:	677,- kÖS

4.3 Phase 3: Integration des Raumes Wien gemäß der ersten Ausbaustufe

In dieser Phase werden alle Wiener Organisationen der ersten Ausbaustufe in den Verbund integriert.

- Leitungskosten	
• 14 x Fernbereich (zu je 9.800)	
• 20 x Ortsbereich (je 5.200)	
abz. 40% Ermäßigung	1.737,- kÖS/Jahr
- Einmalkosten (zusätzlich zu Phase 2)	
• 4 E-Knoten mit je 8 Ports	
• 4 E-Knoten mit je 6 Ports	
• 12 Ports für Wiener S-Knoten	
Einmalkosten Phase 3:	1.546,- kÖS

4.4 Phase 4: Übergang auf DS 64

Diese Phase sieht die Umstellung des Trägernetzes auf DS 64 vor. Die Realisierung kann entsprechend der Verkehrsdichte schrittweise erfolgen. In dieser Aufstellung wird eine Umstellung des gesamten Verbundes angenommen.

Im Zuge der Umstellung können sämtliche DDL-Verbindungen der Phase 3 zurückgegeben werden!

- Leitungskosten	
• 7 DS 64 Verb. zw. den Standorten	
• 10 DS 64 Verb. zu 3 km innerhalb der Standorte	
* 1,25 für Mitbenützung durch andere	5.048,- kÖS/Jahr
(ev. -40% Ermäßigung)	3.029,- kÖS/Jahr
- Einmalkosten (zusätzlich zu Phase 3)	
• 32 V.35/V.36-Konverter	
• 32 Adapterkabel	
Einmalkosten Phase 4:	354,- kÖS

Nach der Umstellung der betreffenden Applikationen auf X.25 kann der DS 64 - Ring auch den Datenverkehr übernehmen, der noch auf eigenen, getrennten Stromwegen abgewickelt wird. Dadurch ergeben sich beträchtliche Einsparungen. Allein an der Universität Innsbruck können beispielsweise folgende Kosten nach Realisierung der Phase 4 eingespart werden:

Von dieser Universität werden derzeit eine DDL 9600 - Verbindung nach Wien und zwei nach Linz betrieben. Nach Realisierung der Phase 2 kann eine der Verbindungen nach Linz abgebaut werden. Nach der Umstellung auf DS 64 können auch die beiden anderen Verbindungen abgebaut werden, was eine Einsparung von 282.240,- öS/Jahr ergibt. Berücksichtigt man weiters, daß mindestens 50% des DATEX-P-Verkehrs dieser Universität auf die DS 64 Verbindungen übernommen werden können, werden weitere 111.000,- öS/Jahr eingespart. Diese Einsparungen von insgesamt 393.240,- öS/Jahr machen bereits etwa 77% der Kosten der DS 64 - Verbindung zwischen Innsbruck und Salzburg (507.600,- öS/Jahr) aus.

5. Ausblick auf die zweite Ausbaustufe, LAN-Kopplungen

Nach der Realisierung der Phase 4 der ersten Ausbaustufe steht eine Leitungsinfrastruktur zur Verfügung, die dazu benutzt werden könnte, die in den einzelnen Universitäten aufgebauten, bzw. im Aufbau begriffenen Local Area Networks durch Router miteinander zu verbinden.

Die Topologie eines solchen Netzes ist die gleiche wie die in Abb. 1, jedoch würden in den einzelnen Universitätsstandorten statt der X.25-Knoten leistungsfähige LAN-Router eingesetzt, die LANs des betreffenden Bereiches mit denen anderer Bereiche verbinden. Die Kommunikationsinfrastruktur der Phase 4 der ersten Ausbaustufe könnte in den einzelnen Universitätsstandorten weiter verwendet und durch X.25-fähige Router in die LANs der betreffenden Standorte und damit in den gesamten Verbund integriert werden.

Derzeit wird mit dem Aufbau von städteweiten LANs im Universitätsbereich begonnen. Sobald mit der dazu notwendigen Technologie genügend Erfahrung gesammelt worden ist, könnte die Kopplung verschiedener Universitätsstandorte mit dieser Technologie und damit die zweite Ausbaustufe begonnen werden.

DS 64 stellt für solche LAN-Kopplungen das Minimum an Übertragungskapazität dar. Damit Engpässe bei der Kommunikation zwischen LANs verschiedener Universitätsbereiche vermieden werden können, müßte ein Übergang auf den Dienst DS 2000 der ÖPTV, der digitale Stromwege für 2Mbit/s vorsieht, vorgesehen werden. Dieser

Übergang von DS 64 auf DS 2000 könnte schrittweise erfolgen.

Die Leitungskosten für einen Verbund mit DS 2000 gemäß Abb. 1 mit 9 Verbindungen zu je 3 km innerhalb der Standorte würden unter Zugrundelegung der derzeitigen Tarife jährlich etwa 24,4 Mill öS betragen (bei 40% Ermäßigung 14,7 Mill). Hinzu käme noch der Aufwand für die Beschaffung der notwendigen Router für DS 2000 - Verbindungen.

Rechnerkonfiguration an der Abt. Prozeßrechenanlage

Walter Niedermayer , Georg Gollmann

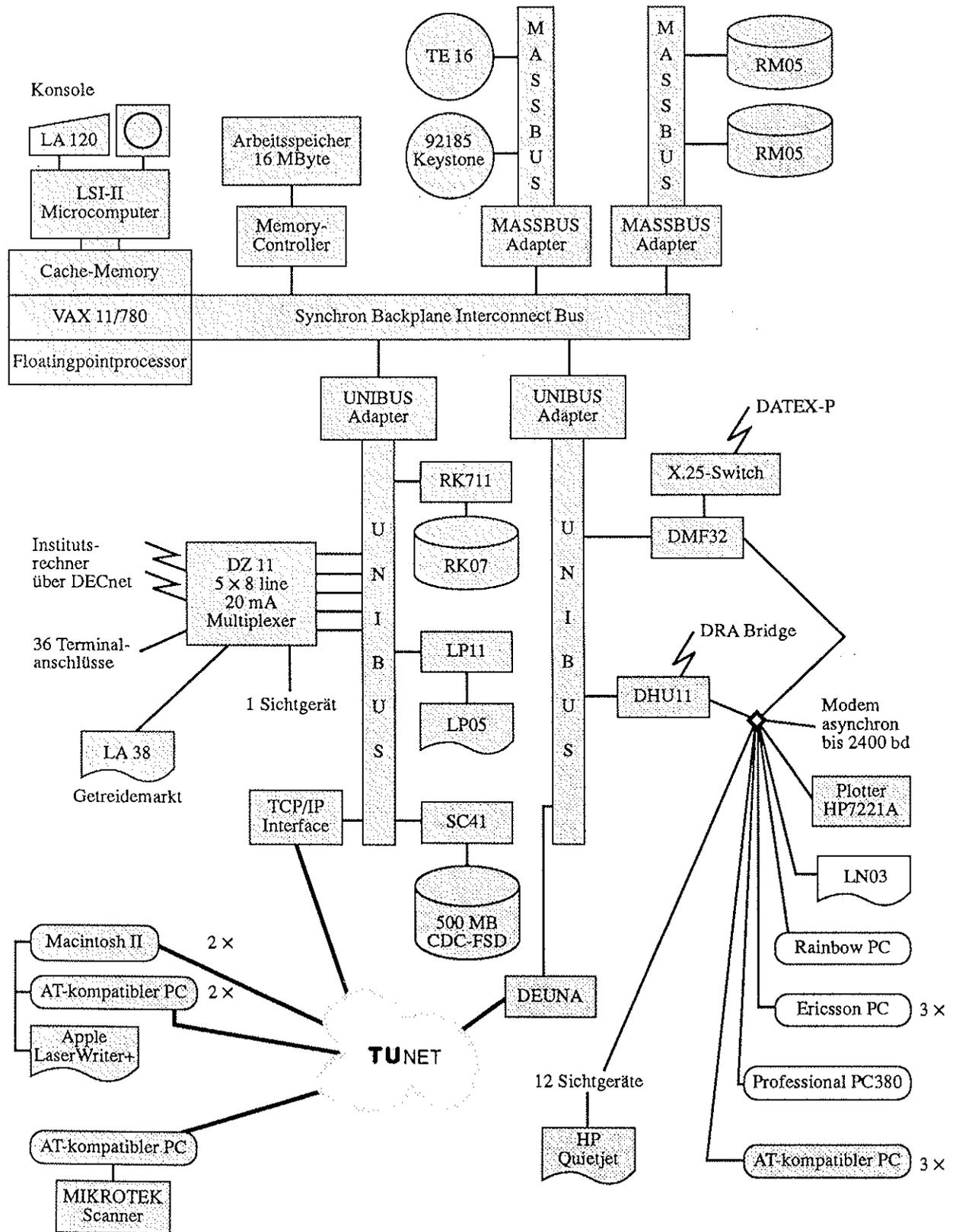
An der Abt. Prozeßrechenanlage wird im Gegensatz zu den anderen Abt. des EDV-Zentrums der **TU**-Wien an zwei verschiedenen Standorten ein breites Spektrum von Rechenanlagen betrieben, wobei dieses vom graphischen Entwurfssystem über einen zentralen Unterstützungsrechner bis zum Prozeßrechner reicht. Die nachfolgenden Konfigurationsblätter sollen einerseits diese Vielfalt zum Ausdruck bringen, andererseits einen genauen Überblick über die an der Abt. Prozeßrechenanlage zur Verfügung stehende Hardware liefern. Zumeist handelt es sich um Rechner der Firma Digital Equipment, wodurch die meisten Bezeichnungen in den folgenden Skizzen DEC-spezifisch sind. Die Rechner können miteinander **TU**-weit im Netz kommunizieren.

Am Standort Freihaus wird die Anlage VAX-11/750 (Seite 30) hauptsächlich von Instituten der Fachgruppe Physik verwendet. In der Gußhausstraße werden die 'Graphik-VAX' (Seite 29) und die CALMA-Anlage (Seite 31) für Anwendungen aus den Bereichen CAD und Graphik benötigt, eine MikroVAX II (Seite 28) dient als Backbone für EUnet, dem europäischen Teil des Netzes von Rechnern mit dem Betriebssystem UNIX, und eine weitere Mikro-VAX II (ebenfalls Seite 28) findet als Testrechner für Pilotprojekte Verwendung. Über die PDP-11/23-Plus (Seite 29) sind mit Hilfe asynchroner DECnet-Verbindungen Institutsrechner mit den Anlagen der Abt. Prozeßrechenanlage verbunden. Die PDP-11/73 (Seite 29) wird für Systemgenerierungen bzw. im Lehrbetrieb eingesetzt. Die VAX-11/780 (Seite 27), ebenfalls in der Gußhausstraße, steht im Rahmen des Aufgabenbereiches der Abt. Prozeßrechenanlage für alle übrigen Anforderungen des Benutzerkreises zur Verfügung.

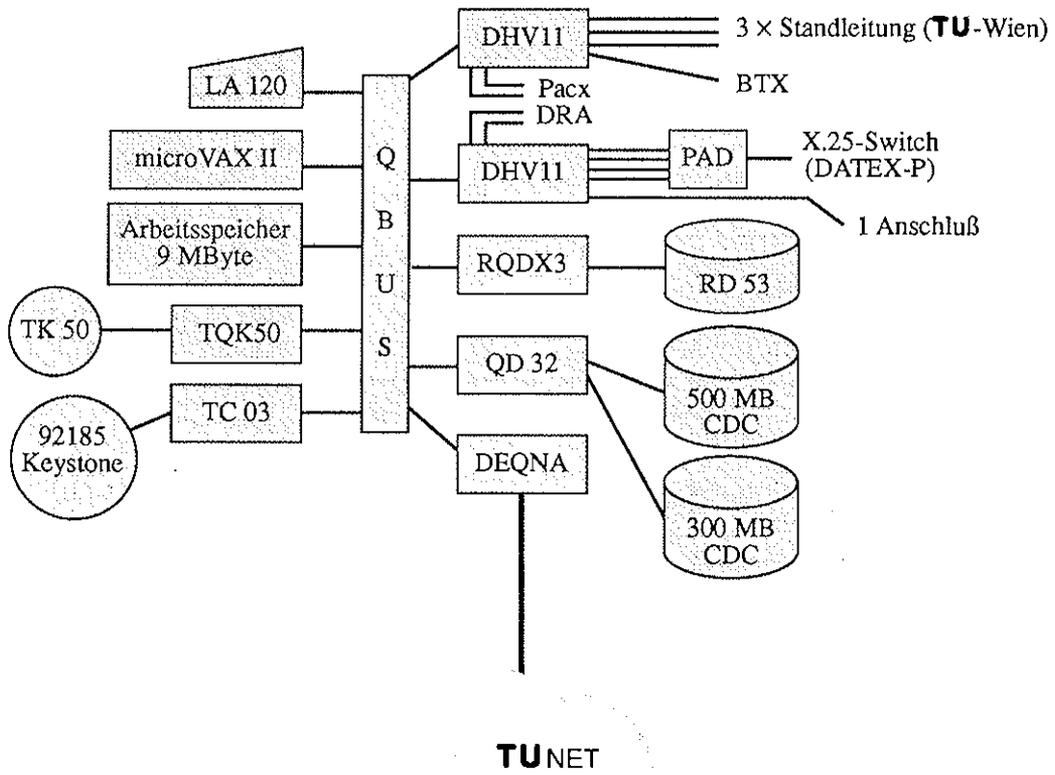
Legende

-  langfristig gebunden
-  gekauft
-  Eigentum des Institutes für
Allgemeine Elektrotechnik und Elektronik
-  gemietet

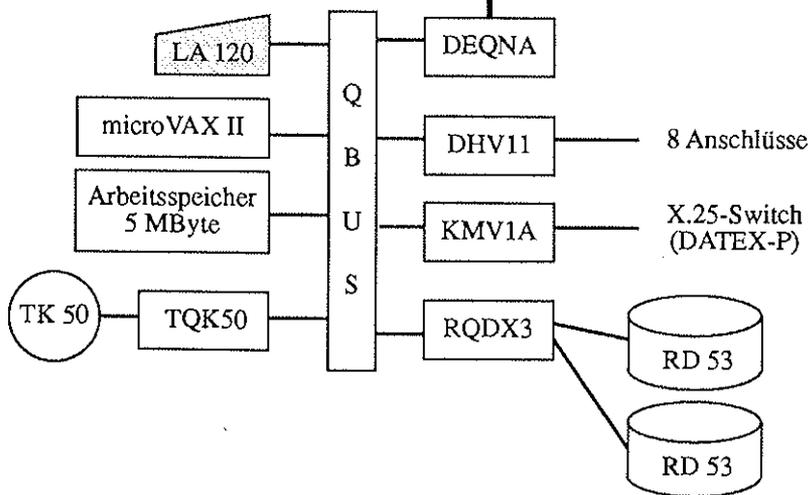
VAX-11/780



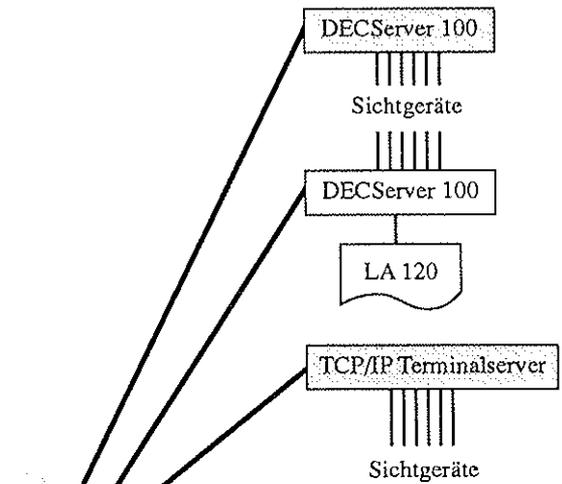
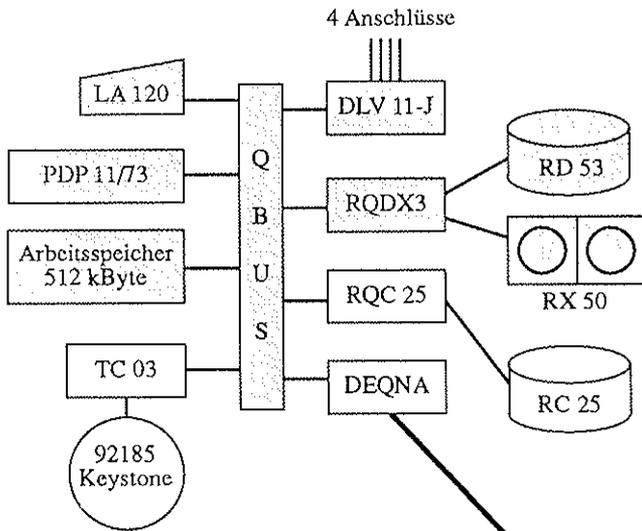
μ VAX II (Backbone - Ultrix)



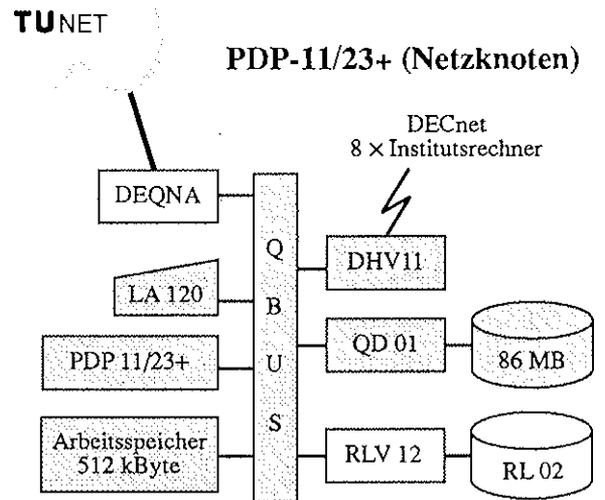
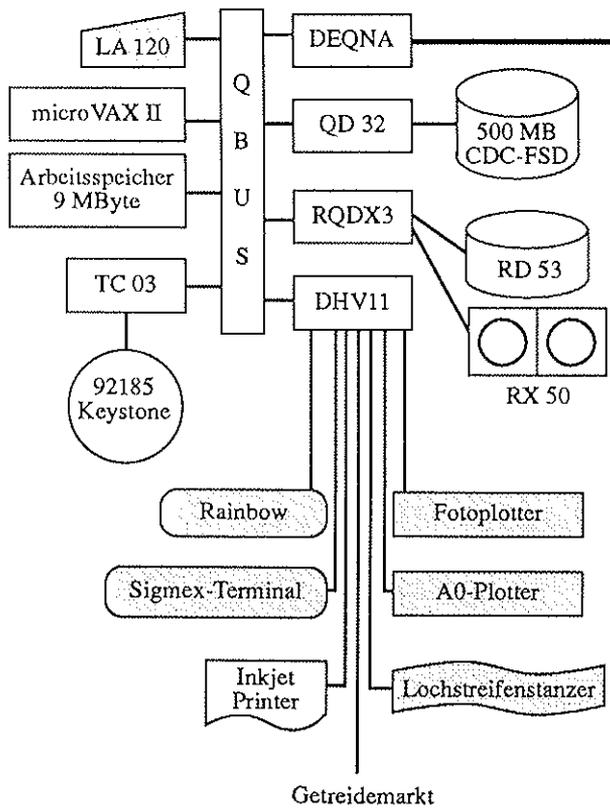
μ VAX II (Ultrix/VMS)



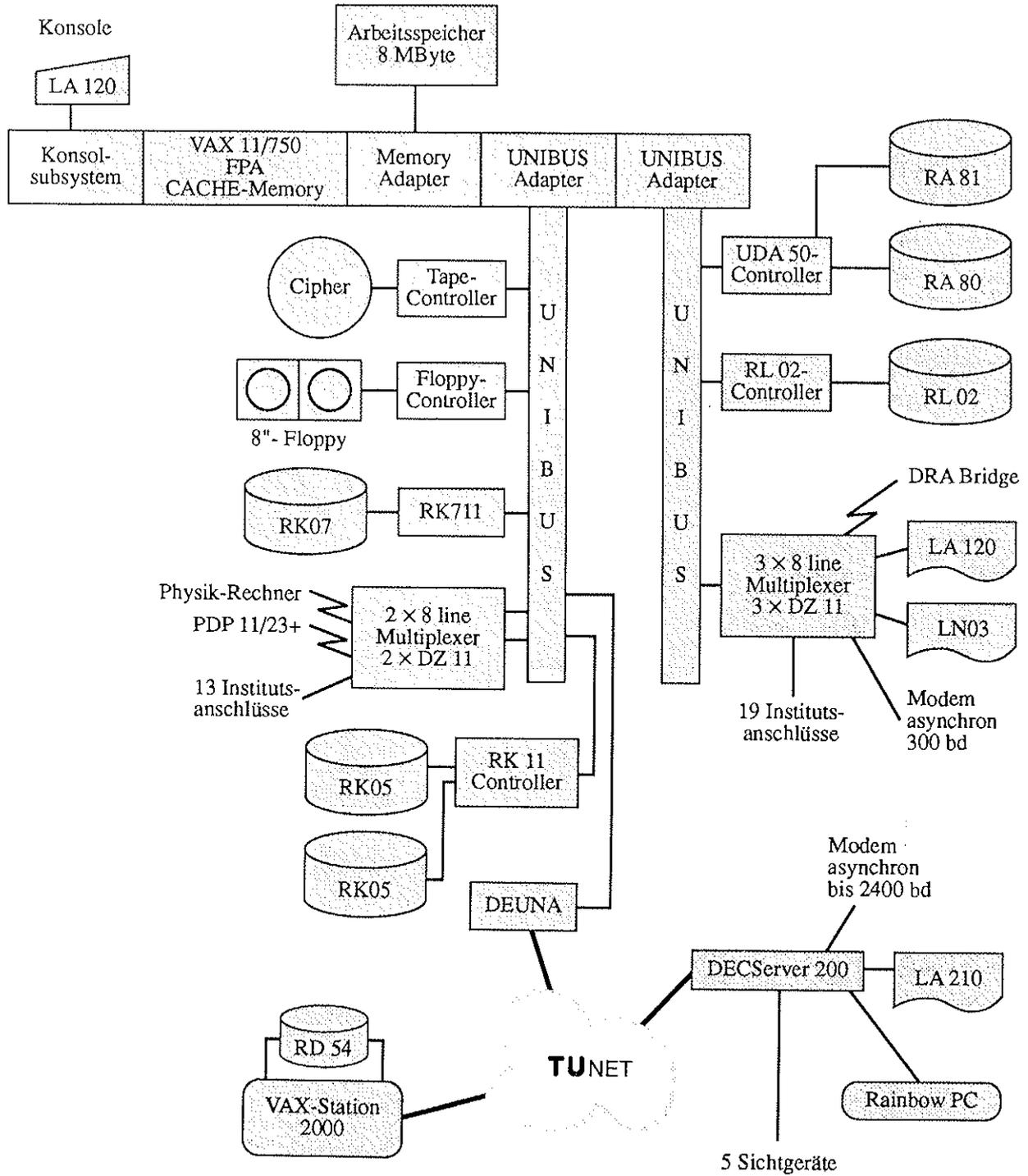
PDP-11/73



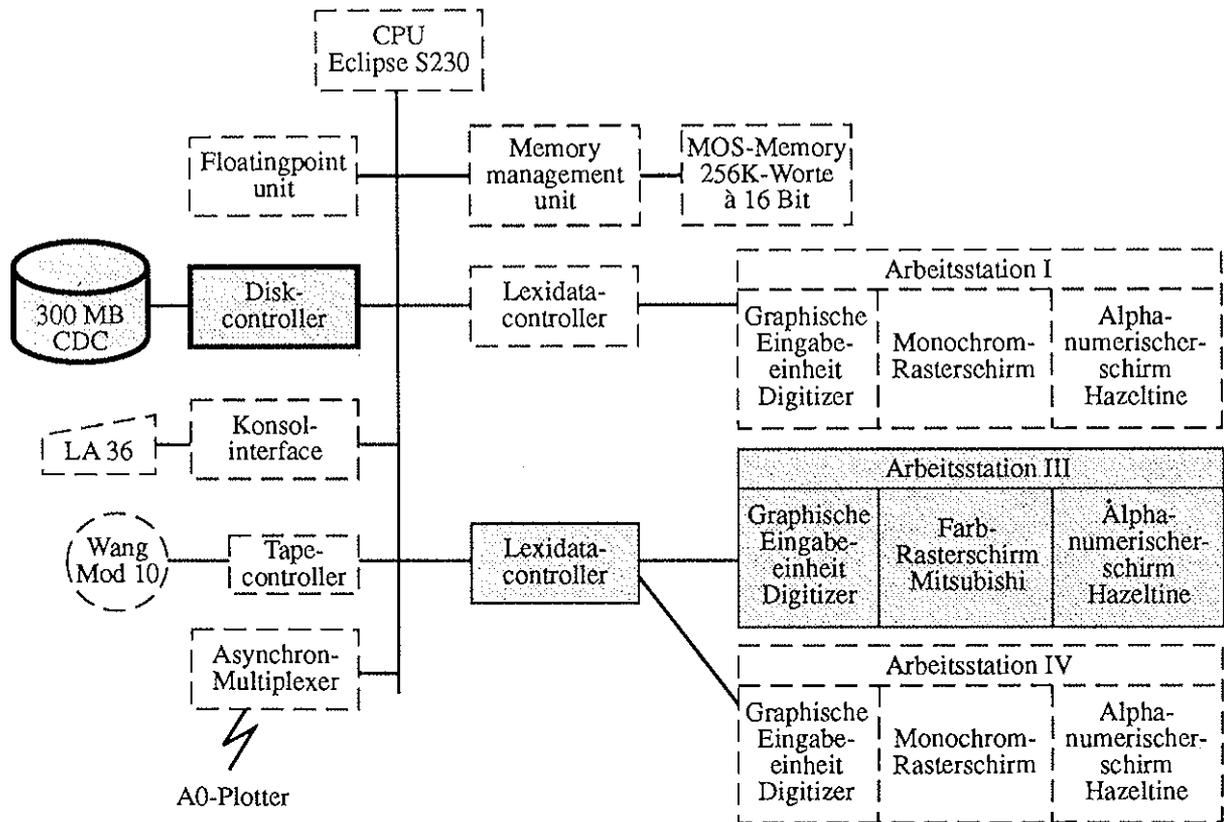
µVAX-II (Graphik)



VAX-11/750 (Standort Freihaus)



CALMA-System



Effiziente Nutzung der Computertechnologie durch neuartige Mensch-Maschine-Kommunikation

Antonin Sprinzl

*“Nicht vom Beginn an enthüllen die Götter uns Sterblichen alles.
Aber im Laufe der Zeit finden wir, suchend, das Bess're.“*

v. Xenophanes 500 v. Chr., ref. in K.R. Popper “Auf der Suche nach einer besseren Welt”

Zusammenfassung

Die Entwicklung am informationstechnologischen Gebiet der letzten Jahre wird durch zwei markante Begleiterscheinungen gekennzeichnet. Zum einen steht dem Anwender zur Lösung anstehender Probleme erfreulicherweise immer größer werdende Computerleistung zur Verfügung. Zum anderen gewinnt die Problematik adäquater Mensch-Maschine-Kommunikation zunehmend an Bedeutung. Anwenderfreundlichkeit, Effizienz, Anpassbarkeit, leichte Erlernbarkeit sowie starke Berücksichtigung des Arbeitsstils individueller Anwender im Umgang mit Computersystemen stehen im Vordergrund.

Einführende Überlegungen

Der rasante Fortschritt auf dem Gebiet der Informationstechnologie (IT) eröffnete eine Fülle neuartiger Einsatzmöglichkeiten, führte aber im gleichen Zug zur Artikulation neuer damit zusammenhängender Problembereiche. Der Komplexitätsgrad von Computersystemen nahm rapide zu und damit auch ihre Interaktionsart mit der Umwelt.

Einen wichtigen Teilbereich des informationstechnologischen Kuchens bilden Computersysteme, die der Anwender interaktiv zur Lösung spezifischer Probleme einsetzt. Derartige Computersysteme lassen sich durch zwei wichtige Merkmale charakterisieren. Zum einen durch die angebotene Funktionalität. Zum anderen durch die Art und Weise, wie ‘effizient’ (vom Anwenderstandpunkt aus betrachtet) diese Funktionalität dem Anwender zugänglich gemacht wird. Im Zusammenhang mit interaktiv eingesetzten Computersystemen gewinnt daher die Frage nach einer geeigneten Applikationsschnittstelle, die den kognitiven, menschlichen Apparat unterstützt, eminent an Bedeutung. Die Erfahrung zeigt, daß eine ‘sauber’ entworfene Applikationsschnittstelle für die Akzeptanz einer Anwenderapplikation vielfach ausschlaggebend ist. Die Anfänge der Computerentwicklung standen, grob skizziert, im Zeichen der Betonung des Funktionalitätsangebotes. Fragen der Mensch-Maschine-Interaktion (MMI, Human Computer Interaction, HCI) fanden hingegen keine sonderliche Beachtung. Wie läßt sich dieser Umstand aus heutiger Sicht begründen? Der primäre Grund dürfte in geringer Leistungsfähigkeit verfügbarer Hardware zum damaligen Zeitpunkt gelegen sein. Demzufolge kamen nur Softwaresysteme eher vom bescheidenen Umfang mit einfacher Kommunikationsstruktur in Frage, die in erster Linie die von ihnen erwartete Funktionalität abzudecken hatten. Für einfallsreiche, anwenderfreundliche Kommunikation fehlte es einfach an nötigen Ressourcen.

Die gegenwärtige Verfügbarkeit leistungsfähiger Hardware verleitet einerseits zur Implementation komplexer Softwaresysteme, andererseits aber bietet sie in ausreichendem Maß Betriebsmittel, die eine adäquate Berücksichtigung der Schnittstellenproblematik gestatten. Eine sauber entworfene Applikationsschnittstelle kann nicht mehr als eine Option ‘verkauft’ werden, sondern gehört zum entscheidenden, integralen Bestandteil einer Applikation.

Mensch-Maschine-Kommunikation, wünschenswerte Schnittstelleneigenschaften

Bisherige Erfahrungen auf dem Gebiet der Konstruktion von SW-Applikationen brachten tiefere Einsichten in die Problematik der Erfordernisse für qualitativ hochwertige Applikationsschnittstellen.

Von einer Applikationsschnittstelle, die den ‘zeitgemäßen’ Anwenderbedürfnissen entsprechen soll, wird eine Fülle von Eigenschaften erwartet. Zu diesen gehören nicht nur die Präsentationsart von Ergebnissen bzw. die Festlegung der Datenformate für die Dateneingabe. Vielmehr beginnen Fragen einer optimalen und effizienten Anpassung der Schnittstelle an die Arbeitsweise des Anwenders eine herausragende Rolle zu spielen. Insbesondere findet die Berücksichtigung des kognitiven Apparats des Anwenders sowie der von ihm bevorzugten, individuellen Betrachtungs- und Arbeitsweise, eine Applikation zu ‘sehen’ und anzuwenden, zunehmend an Bedeutung. Ferner die Möglichkeit der gleichzeitigen Ausführung mehrerer Applikationen, die Möglichkeit des Datenaustausches zwischen Applikationen, Schnittstellenkonsistenz, geringer Schulungsaufwand, Portabilität von Applikationen, Netzwerkunabhängigkeit, HW-Unabhängigkeit, u.a.m. Einige relevante Schnittstelleneigenschaften sollen im folgenden kurz diskutiert werden.

Es ist eine allgemeine Tatsache, daß jeder Anwender in der Benutzung eines Applikationssystems eine bestimmte, ihm eigene Arbeitsweise bevorzugt, der eine individuelle Betrachtungsweise (bedingt vor allem durch allgemeine Erfahrungen) sowie eine spezifische Sicht des Applikationssystems zugrunde liegt. Der Anwender neigt zu einer von ihm bevorzugten, individuellen ‘Manipulation’ eines internen, ‘zurechtgezimmerten’ mentalen Modells, das er sich über die Applikation gebildet hat. Der objektorientierte Ansatz bei der Bildung von Applikationsschnitt-

stellen scheint in diesem Zusammenhang auch hier gute Dienste zu leisten. Die an der Schnittstelle in der Regel symbolisch und graphisch dargestellten Objekte erleichtern dem Anwender grundlegend die interne Bildung eines kognitiven Applikationsmodells. Dadurch kann die Erfassung der in der Applikation begründeten, komplexen Zusammenhänge erheblich verbessert werden. Der Anwender kann dadurch nicht nur leichter und schneller begreifen, was die gegenständliche Applikation zu leisten vermag (Überschaubarkeit), sondern darüber hinaus die Applikation leichter im Gedächtnis bewahren. Der zur effektiven Applikationsnutzung erforderliche Lern- und Schulungsaufwand kann dadurch u.U. beträchtlich reduziert werden.

Von einer 'zeitgemäßen' Mensch-Maschine-Schnittstelle (MMS) darf der Anwender erwarten, daß er bei der gleichzeitigen Bearbeitung mehrerer Applikationen unterstützt wird. Eine wesentliche Rolle kommt dabei der Schnittstellenkonsistenz von Applikationen zu. Aufkommende Differenzen in der Aktionssemantik unter den Applikationsschnittstellen wirken sehr verwirrend und störend. Sie können u.U. bis zur Ablehnung einer Applikationsschnittstelle seitens des Anwenders führen (z.B. nicht einheitliches Kopieren oder Verschieben von Objekten in allen Applikationen). Demgegenüber kann auf das Konto der 'Schnittstellenkonsistenz' eine 'leichte' Erlernbarkeit des Applikationseinsatzes verbucht werden.

Die gleichzeitige Bearbeitung mehrerer Applikationen läßt den Wunsch aufkommen, Datenaustausch zwischen den Applikationen durchzuführen. Auch hier ist die einheitliche Semantik des Datentransfers von Applikation zu Applikation von besonderer Relevanz.

Eine Applikation sollte sich durch hohe Portabilität, Hardware- und Netzwerkuabhängigkeit auszeichnen. Im Zeitalter der verteilten, vernetzten Informationssysteme drängt sich der Wunsch auf, den Funktionsteil einer Applikation von der Applikationsschnittstelle logisch zu trennen. Der wesentliche Vorteil einer solchen Trennung besteht in der Möglichkeit, den Funktionsteil einer Applikation auf einer geeignet dimensionierten Hardware innerhalb des verteilten Rechnernetzes ablaufen zu lassen. Die Hardwareausstattung für die Applikationsschnittstelle schlägt sich, insbesondere bei komplexeren Applikationen mit einem geringeren Aufwand zu Buche.

Mensch-Maschine-Schnittstellen, gegenwärtige Bestrebungen, Erfolgchancen, Marktsituation

Viele der Computeranwender, die schon in der Vergangenheit dabei waren und manche Computerneuanschaffungen miterlebt haben, wissen bezüglich Software-Portabilität, Hardware-, Netzwerkuabhängigkeit phantasievoll, an Realität nicht entbehrende Stories zu erzählen.

Der Wunsch nach portablen, hardware- und netzwerkuabhängigen Applikationen ist nichts Neues und wurde seitens der Anwender bei vielen Anlässen stets oft genug artikuliert. Seitens der Computerhersteller wurde diese Angelegenheit aus verständlichen Gründen lange Zeit nicht beachtet. Erst eine seit 1984 zu beobachtende Entwicklung, die insbesondere durch die X/Open Group entscheidend geprägt wird, läßt die berechtigte Hoffnung zu, daß weitestgehende Applikationenportabilität in absehbarer Zukunft erzielt werden kann.

Die X/Open Group ist ein 'neutrales', allgemein anerkanntes Gremium, dem eine Vermittlerrolle zwischen Anwendern und Herstellern zukommt. In der X/Open sind alle namhaften Computerhersteller, Anwendergruppierungen und andere Firmen (Ölfirmer) vertreten. Das primäre Anliegen der X/Open besteht in der Schaffung eines Marktes für Offene Systeme. Von der X/Open werden diesbezüglich Empfehlungen und Richtlinien erarbeitet (im Portability Guide), die eine gemeinsame Bezugsplattform darstellen. Primär sollen dadurch neue Marktverhältnisse geschaffen werden, denen zufolge Anwender eigenständige, längerfristige Investitionspolitik betreiben können. X/Open ist kein Standardisierungsgremium, arbeitet aber sehr eng mit ISO, IEEE, NIST und vielen anderen namhaften Normungsgremien zusammen und sorgt dafür, daß einschlägige Normen und defacto Standards aufeinander abgestimmt und verbreitet werden.

Im Zusammenhang mit der Tätigkeit der X/Open Group ist es sinnvoll zwei weitere oft genannte Gruppierungen zu erwähnen, die Open Software Foundation (OSF, gegründet Mai 1988) und die Unix International (UI, Gründung Ende 1988). Bezüglich der mittlerweile allseits zu beobachtenden Bestrebungen, zur Schaffung eines Marktes für Open Systems beizutragen, herrscht Einigkeit darüber, daß dabei der X/Open Group generell die Aufgabe der Definition von Open Systems zukommt, während sich OSF und UI der Realisierung und dem Vertrieb je einer, industriellen Anforderungen entsprechenden Implementierung annehmen (zwei unterschiedliche UNIX-Implementierungen, die einer Definition entsprechen). Ja, und was hat das alles mit Mensch-Maschine-Schnittstellen (MMS) zu tun? Zum einen gehört die Problematik der MMS zu einem der zentralen Punkte im 'Portability Guide for Common Application Environment (CAE)' (zusammen mit UNIX-basierter System Interface Specification (XSI), Rechner-Rechner-Kommunikation, u.a.m.). Zum anderen berücksichtigt die X/Open Group MMS-spezifische Vorschläge, die vom X-Konsortium kommen. (X-Konsortium kontrolliert die X-Window Spezifikation).

Es ist erwähnenswert, daß die Computerhersteller DEC und IBM dem MIT-Computerlabor beträchtliche Mittel in Höhe mehrerer Mill. Dollar spendeten, um die Problematik der flexiblen, anwenderorientierten Mensch-Maschine-Kommunikation näher zu untersuchen. Es ist daraus deutlich ersichtlich, welchen Stellenwert Computerhersteller

der MMS-Problematik beimessen.

Ein neues Paradigma für Applikationsinteraktion, Grundzüge

Im folgenden sollen einige bemerkenswerte Grundzüge des neuen Paradigmas der fensterorientierten Applikationsinteraktion angedeutet werden. Stellvertretend für diese neue 'Fenster-technik' möge das zum de-facto Standard avancierte X11 Window (Version 11) Grundmodell für Mensch-Maschine-Interaktion vom MIT herangezogen werden, das mittlerweile von breiten Schichten der Computerwelt angenommen wurde. Andere bekannte Fenster-techniken (DEC Windows, Macintosh, MS Windows, Sun NeWS, u.a.) sind teils Erweiterungen vom MIT-Konzept, zum anderen Teil hängen diese mit dem MIT-Konzept nicht zusammen, bzw. sind manche wesentliche Eigenschaften des MIT-Konzeptes nicht enthalten (z.B. Netzkunabhängigkeit). Die zum Teil weit auseinander gehenden Unterschiede in den Eigenschaften genannter Produkte sollen hier nicht näher diskutiert werden. Eines haben diese Fenster-techniken allerdings sicher gemeinsam ... ein sehr ähnliches Aussehen.

Zur Erinnerung an 'klassische' Schnittstellentechnik möge kurz auf Charakteristika einer typischen Applikation eingegangen werden.

Eine klassische Applikation zeichnet sich (hier natürlich stark vereinfacht, aber für die Betrachtung ausreichend) bezüglich der Schnittstellenproblematik zumindest durch folgende wesentliche Aspekte aus.

Erstens befinden sich die Schnittstellenkomponenten (I/O-Anschlüsse) einer Applikation 'irgendwo' tief in ihrem Funktionsteil begraben. Von einer halbwegs sauberen Trennung Applikationsschnittstelle/Funktionsteil kann in den meisten Fällen kaum gesprochen werden.

Zweitens wird eine Applikationsschnittstelle fix 'einprogrammiert', die eine Anpassung ohne einen elaboraten, in der Regel recht aufwendigen Eingriff in die gesamte Applikation kaum zuläßt. Daraus ergibt sich auch der 'statische' Charakter der Applikationsschnittstelle. Eine Modifikation seitens des Anwenders ist daher kaum möglich.

Drittens, die Applikation 'läuft' prinzipiell in einer endlosen Schleife (auch hier vereinfacht), nach folgendem Grundschema :

```
WHILE appl_not_done DO
    BEGIN wait_for_kb_input;
          process_kb_io
    END
```

Viertens, der Ablauf mehrerer Applikationen hat meistens einen linearen Charakter. Nach der vollständigen Ausführung einer Applikation wird die nächste gestartet, usw. Die gleichzeitige Ausführung mehrerer Applikationen mit umschaltbaren, nacheinander darstellbaren, 'virtuellen' Bildschirmen entbehrt einer sauberen konzeptuellen Grundlage und bietet aus heutiger Sicht nur marginale Vorteile.

Fünftens, die Applikationsinteraktion ist zeichenorientiert (keine pixel-Graphik) und von einer verhältnismäßig geringen Intensität (wenig Anwenderinteraktion, Applikation rechenintensiv).

Demgegenüber läßt sich eine fensterorientierte Applikation folgendermaßen charakterisieren.

Es wird eine möglichst 'saubere', logisch übersichtliche Trennung zwischen Applikationsschnittstelle und dem dazugehörigen Funktionsteil angestrebt. (Das dies nicht immer leicht durchzuführen ist, soll hier nicht weiter erörtert werden.)

Eine fensterorientierte Applikation hat aufgrund des neuen Schnittstellen-Kommunikationskonzeptes (X Window) mehr typenunterschiedliche, asynchrone Ereignisse (events) abzufangen und auf diese entsprechend zu reagieren (nicht zu erwartende Ereignisse können 'herausgefiltert' werden). Events können z.B. von anderen Applikationen, von der Tastatur, von der Maus oder sonstiger IO-Peripherie ankommen.

Es wird von einem neuen Interaktions-Grundkonzept ausgegangen. Demzufolge wird die Art und Weise der Interaktion nicht von der Applikation geprägt (wie im 'klassischen' Interaktionsfall), sondern vom Anwender festgelegt und nach seinen Präferenzen auch modifiziert. Der Anwender prägt die Interaktionsart (u.U. dynamisch) nach seinem Geschmack. Die Kommunikationsperipherie (Bildschirm, Maus, usw.) 'gehört' sozusagen dem Anwender. In der Tat aber wird die vom Anwender bevorzugte Interaktionsart vom Window Manager (auch eine Applikation) wahrgenommen. Dem Window Manager kommt eine Vermittlerrolle zu. Der Funktionsteil bietet beispielsweise nur 'Vorschläge' (hints) an, wie etwa eine Ausgabe aussehen könnte. Die finale Präsentation der Ausgabe bestimmt dann der Window Manager, indem dieser die vom Funktionsteil kommenden Vorschläge wahrnimmt und die ihm mitgeteilten Anwenderpräferenzen berücksichtigt.

Das neue Interaktionskonzept gestattet eine gleichzeitige Bearbeitung mehrerer, voneinander unabhängiger Applikationen, wobei u.U. mehrere Applikationsschnittstellen am Bildschirm gleichzeitig dargestellt werden können. Mit Hilfe einfacher Datentransfer-Operationen von Schnittstelle zu Schnittstelle können Daten 'transparent'

zwischen zwei irgendwo im Netzwerk 'verstreuten' Applikations-Funktionsteilen übertragen werden.

Die neuen Applikationsschnittstellen sind von Haus aus graphikorientiert. Eine graphikorientierte Schnittstelle bedeutet eine erhebliche qualitative Verbesserung in der Mensch-Maschine-Kommunikation. Objektorientierte, graphische Präsentation trägt wesentlich dazu bei, das sich der Anwender ein 'Bild' von der jeweiligen Applikation (ein mentales Abbild der Applikation, ein Modell) machen kann. Eine der wesentlichen Stärken des menschlichen kognitiven Apparates besteht ja in der effizienten, internen 'Manipulation' solcher Bilder (Aufnahme, Lernvorgang), Speicherung, Abruf, Kommunikation im weiten Sinne, Präsentation, Explanation, Exploration, Modifikation, u.v.a.m.)

Wie läuft nun eine Interaktion in einer fensterorientierten Applikation ab? Anhand des X-Window-Modells möge der Vorgang kurz angedeutet werden:

```
Init_Application (* open_connection_to_X_server *);
generate_widgets (* widgets ... 'Fensterkomponenten' *);
pass_control_to_event_dispatcher;
LOOP forever BEGIN do_nothing_but_wait_for_and_process_events END
```

CPU-intensive Prozeduren sollten eine immediate_event-Abfrage (z.B. event terminate, oder sonstige wichtige requests) eingebaut haben:

```
WHILE cpu_intens_proc DO
    BEGIN IF not_immediate_event THEN do_nothing ELSE process_event;
    compute_a_while END
```

X-Window-Protokoll als Grundbaustein künftiger Anwenderapplikationen

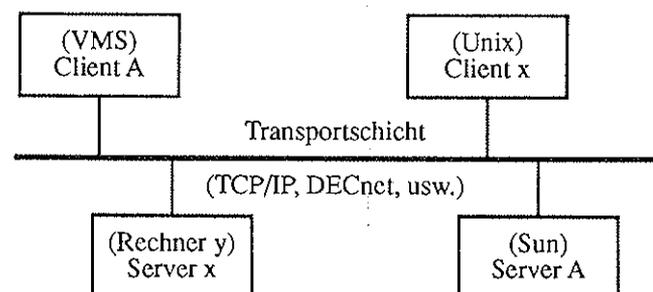
Nur in Kürze sollen hier einige markante Grundzüge des weitverbreiteten X-Window-Protokolls hervorgehoben werden, das die künftige Entwicklung auf dem Gebiet anwenderspezifischer Applikationen vermutlich stark prägen wird. Der Umfang von X-Window wird vom X-Konsortium am MIT kontrolliert.

X-Window ist zu einem industriellen de-facto Standard für Workstations geworden. Das Protokoll garantiert eine Hardware- und Netzwerkunabhängigkeit. Ferner wird vom Protokoll eine leistungsfähige, netzwerktransparente Fenstertechnik und Graphik unterstützt. Die sources (in der Programmiersprache C) sind allgemein verfügbar. Das Protokoll geht von einer logischen Trennung der Applikation in einen Funktionsteil (client) und eine Applikationsschnittstelle (server) aus. Die Applikationsschnittstelle ist WIMP-orientiert (window, icon, mouse, pointer). Es wird gleichzeitige Bearbeitung mehrerer Applikationen und mehrerer, gleichzeitig darstellbarer Fenster unterstützt. Unter Beachtung einschlägiger Konventionen kann eine intuitiv zu bedienende, leicht erlernbare einheitliche Interaktionstechnik und ein Datenaustausch zwischen verschiedenen Applikationen erzielt werden.

Fensterorientierte Applikationen mit DECwindows, Übersicht

Im folgenden sollen einige grundlegende Konzepte sowie Komponenten des DECwindows Environment erläutert werden, mit dessen Hilfe fensterorientierte Applikationen erstellt werden können.

DECwindows basiert auf dem bereits erwähnten X-Window-Modell. Demzufolge setzt sich eine Applikation aus zwei logisch voneinander getrennten Teilen, dem Client und dem Server, zusammen. Client und Server können auf einer bestimmten Hardware gemeinsam 'laufen'. Aufgrund der logischen Trennung ist aber auch eine andere Hardware-Konstellation möglich. In einem Netzwerk kann beispielsweise der Client auf einem bestimmten, u.U. leistungsfähigen Rechner laufen, der Server hingegen auf einer kostengünstigen Workstation. Die nebenstehende Abbildung möge dies verdeutlichen.



Eine DECwindows-Applikation besteht in der Regel aus mehreren Schichten. Die unterste Schicht, die von einer Applikation sinnvollerweise angesprochen werden kann, ist die Bibliothek der implementierten Grundfunktionen, Xlib des X-Protokolls. Diese Bibliothek stellt die Basisbibliothek des MIT Distribution Set dar..

Die darüber gelagerte Schicht ist die Schicht der sog. 'intrinsic functions', die Xtk-Bibliothek, die von MIT auch mitgeliefert wird. Diese Bibliothek, die auch als 'Werkzeugkiste für Werkzeugbauer' bezeichnet wird, enthält eine Kollektion grundlegender Objekte, sog. Widgets, die immer wieder in einer fensterorientierten Applikation angewandt werden (elementare buttons, menus, scroll bars, icons, u.ä.). Diese 'zusammenhanglose' Ansammlung

von Objekten kann dann von einem Werkzeugbauer (eigentlich einem Designer) nach unterschiedlichen Kriterien zu umfassenderen Objekten zusammengestellt werden. Die daraus entstehende, durch den Designer geprägte Schnittstellenkomponente (man beachte, immer noch eine Komponente, allerdings nicht von elementarer Art), zeichnet sich dann durch ein bestimmtes 'look and feel' aus. Die Xtk-Bibliothek ist 'policy free', die durch den Designer entworfenen Komponenten ('höherer Art') werden hingegen 'policy bound' genannt (weil diesen bereits eine bestimmte 'politische' Prägung anhaftet).

DECtoolkit ist im wesentlichen als Bibliothek für den Applikationsprogrammierer gedacht. Durch den Einsatz von DECtoolkit wird primär das Ziel verfolgt, die Anwenderprogrammierung zu vereinfachen und effizienter zu machen. DECtoolkit enthält demnach eine reiche Kollektion solcher, von der DEC vorgefertigter Grundkomponenten (z.B. Applikationsinput, Dialog, Menüs mit drop-down, pop-up, pull-right, Listenboxen, usw.), die sich in einem breiten Spektrum von Anwendungen verwenden lassen (man spricht dann von einem DEC 'look and feel' im Gegensatz z.B. zu einem Sun 'look and feel').

Die Verwendung von DECtoolkit bzw. Xtk ist natürlich mit Vor- und Nachteilen behaftet. Xtk stellt beispielsweise ein mächtigeres Instrumentarium für die Realisierung eigenständiger Vorstellungen zur Verfügung als DECtoolkit. Der Preis dafür ist aber ein höherer Programmieraufwand. Demgegenüber ist der Einsatz vorgefertigter Bausteine aus dem DECtoolkit verhältnismäßig einfach, wobei ein 'präfabriziertes look and feel' in Kauf zu nehmen ist (die Problematik erinnert ein wenig an die Entscheidung 'TeX oder LaTeX?'). Dessen ungeachtet soll hier die 'religiöse' Frage, welche Vor- bzw. Nachteile ein bestimmtes, einheitliches 'look and feel' auf allen oder vielen Computersystemen mit sich bringt, nicht weiter verfolgt werden (Stichworte dazu ... Schulungseffizienz, öde Einheit).

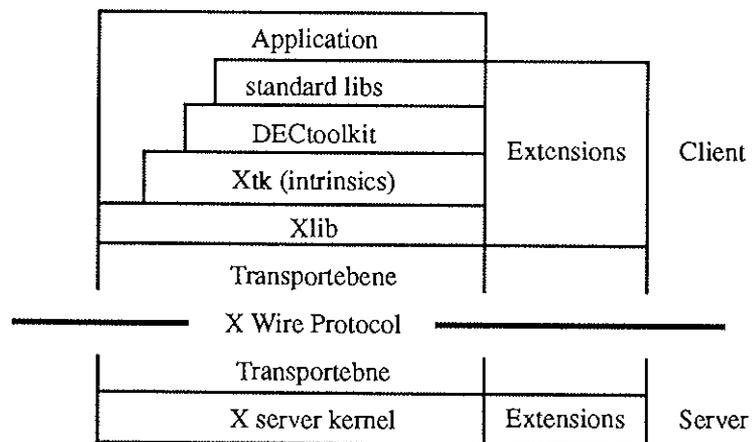
Zu den Standard-Bibliotheken gehört z.B. das GKS für 2D-Graphik, u.a.m.

Es sei erwähnt, daß zu den geplanten, in Arbeit befindlichen wichtigen Erweiterungen die Einsatzmöglichkeit von Adobes's Beschreibungssprache PostScript gehört.

Von einer Applikation können die angeführten Schichten von allen im VMS environment verfügbaren Programmiersprachen aufgerufen werden, inklusive des gleichzeitigen Ansprechens mehrerer Schichten in üblicher 'VMS-Manier' (call and lib intermixing).

Die nebenstehende Abbildung möge den geschilderten Sachverhalt in graphischer Form verdeutlichen.

DECwindows wird mit einigen grundlegenden DECwindows-Applikationen ausgeliefert, zu denen u.a. der Session Manager und der Window Manager gehören. Zum Grundpaket gehören auch Terminal emulation (VTxxx, 'virtuelle Terminals', cut and paste) und VMS User Executive (objektorientierte Schnittstelle zu einer Unter- menge des Betriebssystems VMS, die einige der häufig verwendeten VMS-Kommandos implementiert).



Der Session Manager initialisiert den Server, gestattet Login und einige lokale Anpassungen.

Der Window Manager (als 'Interessensvertreter' des Anwenders) übernimmt die Aufgabe, Anwender- und Applikationswünsche so zu koordinieren, daß der Anwender mit der Interaktionsart (z.B. gleichzeitige Darstellung mehrerer Applikationsergebnisse auf dem Bildschirm) stets zufrieden ist. Der Window Manager versieht jedes Applikationsfenster mit einer 'Dekoration' (Applikationstitel, window modification icons, usw.). Darüber hinaus wird vom Window Manager das icon-box-Fenster (üblicherweise am Bildschirm unten) angelegt und verwaltet ('Iconisierung' von Applikationen).

DECwindows und Applikations-Entwicklungswerkzeuge höherer Integrationsstufe

Konstruktion von anwenderfreundlichen, effizienten, flexiblen Schnittstellen ist zu einer nicht trivialen Angelegenheit geworden. Das Beispiel, das diesen Umstand eindrucksvoll dokumentiert, ist das Hello-Programm. Hello-Programm (die Programmausführung produziert am Bildschirm 'Hello world'), ein sog. 'Zweizeiler' in der Programmiersprache C, nimmt als eine fensterorientierte 'Applikation' etwa zwei A4-Seiten ein.

DECwindows bietet, wie bereits erwähnt, eine Menge von nun an als elementar anzusehenden Grundkomponenten dar, die der (geübte) Applikationsprogrammierer bei der Realisierung von Applikationen einsetzen kann.

Diese Grundkomponentenmenge gestattet die Konstruktion von Schnittstellen vom statischen Charakter.

Gefragt sind aber Applikationsschnittstellen vom dynamischen Charakter. Die gewünschte Dynamik äußert sich zumindest in drei unterschiedlichen Formen. Einerseits soll nach Möglichkeit der jeweilige Anwender selbst 'seine' Applikationsschnittstelle 'entwerfen' können. Andererseits soll die Möglichkeit vorgesehen sein, mehrere bereits vordefinierte Applikationsschnittstellen schnell und effizient zu tauschen. Und nicht zuletzt soll der Anwender bereits vorhandene (d.h. vordefinierte) Applikationsschnittstellen nach seinem Geschmack modifizieren können.

Es bedarf eines weiteren Überbaues, um die angedeuteten Modifikationsmöglichkeiten vorzusehen. Zu den wesentlichen Komponenten dieses Überbaues gehören derzeit User Interface Language (UIL), Digital Resource Manager (DRM) und das DWICS, DECwindows Interface Construction Set. Im folgenden werden die wesentlichen Aufgaben jeder der genannten Komponenten kurz skizziert und ein Zusammenhang aller untereinander präsentiert.

User Interface Language (UIL) ermöglicht eine einfache Trennung einer Applikation in einen Schnittstellenteil und einen Funktionsteil. Demnach wird eine Applikationsschnittstelle mit Hilfe von UIL deskriptiv definiert und mit dem UIL-Compiler in ein binäres Format (UID-File) überführt (höhere Ablaufeffizienz). Korrespondierender Funktionsteil (Applikationsroutinen) wird unabhängig vom UIL-File separat compiliert und gelinkt. Durch diese Trennung lassen sich Schnittstellen problemlos tauschen. Daraus läßt sich eine Reihe von Vorteilen ableiten, wie z.B. Prototypisierung von Schnittstellen, Internationalisierung, Anpassung an unterschiedliche Arbeitsstile, usw.). Im UIL-File werden im wesentlichen das Fenster-Layout, d.h. Widgets, ihre Hierarchie und sonstige Charakteristiken sowie aufrufbare Applikationsroutinen spezifiziert.

Die wesentliche Aufgabe des Digital Resource Manager (DRM) besteht in der Interpretation von binären UID-Files zu run-time, um die Applikationsausführung zu beschleunigen (Generieren von Widget-Instanzen). DRM erlaubt den Eingriff in die Definition der Applikationsschnittstelle zu runtime, womit sich eine dynamische Schnittstellenanpassung erreichen läßt (z.B. Berücksichtigung von Landesspezifika).

Mit der Konstruktion von DECwindows Interface Construction Set (DWICS) wird eine weitere Vereinfachung des SchnittstellenDesigns angestrebt, indem graphische Hilfsmittel zur Verfügung gestellt werden. Ein Schnittstellen-Layout kann dadurch mit Hilfe eines geeigneten, graphischen WYSIWYG-Editors durchgeführt werden. Gegenüber einer deskriptiven Beschreibung mit Hilfe einer Definitionssprache resultieren daraus für manchen sichtbare Vorteile. DWICS ist in der Grundversion von DECwindows noch nicht verfügbar.

Ausblick

Auf dem Gebiet der Mensch-Maschine-Kommunikation wird derzeit emsig geforscht. Im Mittelpunkt stehen Schnittstellen, die ökonomisch konstruiert werden können, an den Arbeitsstil eines Anwenders individuell anpaßbar sind und die sich in einem breiten Applikationsspektrum einsetzen lassen.

Die Akzeptanz von Computertechnologien in der Zukunft sowie deren effizienter Einsatz wird entscheidend von qualitativ hochwertigen, anwenderfreundlichen Schnittstellen abhängen. Darüber hinaus ist es stets zu betonen, daß das zu benutzende Computerinstrumentarium als Werkzeug an den menschlichen Intellekt anzupassen ist und nicht umgekehrt.

Weitere geplante Aktivitäten des EDV-Zentrums

Das EDV-Zentrum ist im Rahmen der Kommunikationsaktivitäten bestrebt, der Problematik der Mensch-Maschine-Kommunikation erhöhte Aufmerksamkeit zu widmen. Insbesondere soll die Thematik der tatsächlichen Portabilität der auf X Window basierenden Applikationen untersucht werden. Das EDV-Zentrum beabsichtigt im Laufe des Monats Oktober 1989 eine X-Window-Präsentation zu veranstalten, zu der eine Reihe namhafter Workstation-Hersteller eingeladen werden soll. Dabei sollen unserem Kundenkreis Vorteile dieser Neuentwicklung vorgeführt werden. Darüber hinaus ist ein Einführungskurs in die windoworientierte Applikationsprogrammierung in Vorbereitung.

Literaturhinweise

[Myers89] Myers B.A.: User-Interface Tools:Introduction and Survey, IEEE Software, Jan. 1989

[Fischer89] Fischer G.: Human-Computer Interaction Software: Lessons Learned, Challenges Ahead, IEEE Software, Jan. 1989

[Hurley89] Hurley W.D., Sibert J.L.: Modeling User Interface Application Interactions, IEEE Software, Jan.1989

[Dodani89] Dodani M.H., Hughes C.E., Moshell J.M. : Separation of Powers, Byte, March 1989

Software Engineering als Konstruktionsgrundlage für zeitgemäße Softwaresysteme

Antonin Sprinzl

"Good judgment comes from experience, and experience comes from bad judgment."

Fred Brooks

Zusammenfassung

Die ingenieurmäßige Konstruktion zeitgemäßer Anwendersysteme erfordert den Einsatz eines neuen Instrumentariums. Zu diesem gehören nicht nur die vielerorts zitierten, unterschiedlichen Tools, sondern vor allem ein fundierter methodischer und verfahrenstechnischer Rahmen, der oft vernachlässigt wird. Eine disziplinierte Anwendung einschlägiger Methoden und Techniken gehört zu den Grundvoraussetzungen für einen zielführenden CASE-Einsatz. Der SW-Konstrukteur als die treibende, Kreativität entwickelnde Kraft nimmt in dem SW-Erstellungsprozeß weiterhin die zentrale Rolle ein.

Leistungsfähige Hardware, treibendes Element der Computerentwicklung

Die Computertechnik der letzten 20 Jahre hat eine beachtenswerte Entwicklung zu verzeichnen. Insbesondere auf dem Gebiet der Hardware wurden Leistungssteigerungen erreicht, die sich über mehrere Potenzen erstrecken. Der erzielte technologische Fortschritt in der Softwareerstellung ist hingegen bei weitem nicht von einer ähnlichen spektakulären Dimension. Auf diesen Umstand ist die Prägung der 'Softwarekrise' zurückzuführen.

Die Verfügbarkeit leistungsfähiger Hardware 'verleitet' leicht zur Realisierung komplexer Systeme. Die Systemkomplexität äußert sich aber in der Regel in der Systemsoftware, die dementsprechend umfassend ausfällt.

Es wird mittlerweile allgemein anerkannt, daß sich die zu bewältigende Softwarekomplexität mit Konstruktionspraktiken der früheren Jahre kaum bewältigen läßt. Dies dokumentieren zahlreiche Systemrealisierungen, die in der Vergangenheit mit einem Desaster endeten. Wie eine amerikanische Studie feststellt, konnten in weniger als 10 Prozent der Fälle die gewünschten Systeme ohne tiefgreifende Modifikationen an den Auftraggeber ausgeliefert werden.

Was lief in den meisten Fällen schief? Bei näherer Betrachtung konnte festgestellt werden, daß die Eigenschaften der ausgelieferten Systeme, Funktionalität, Dialogart, usw., den Anwendererwartungen selten entsprechen haben. Der Anwender wollte an sich etwas anderes. Darüber hinaus wurden Anwendersysteme zu spät ausgeliefert, die finalen Kosten beliefen sich auf ein Vielfaches der geschätzten Kosten. Weitere Überraschungen brachte die sog. Wartungsphase, in der grobe Versäumnisse während der Produktrealisierung ans Licht kamen. Demnach wurde eine Anpassung an neue Anwenderbedürfnisse bzw. Hardware zu einem ernsthaften Problem. Dafür aufgewendete, zusätzliche Mittel betragen sehr oft 200% der Produktrealisierungskosten. Mangelndes Qualitätsbewußtsein sowie unzureichende Systematik in der Realisierungsphase schlugen sich zu Buche. Hat sich die gegenwärtige Situation grundlegend verändert?

Zentrales Problem der SW-Erstellung, Problemspezifikation, Systemdesign

Eines der wesentlichen Probleme in der Konstruktion von Software besteht in der genauen Spezifikation des Leistungsumfanges des künftigen Anwendersystems (Funktionalität) sowie der Ermittlung einer anwenderoptimalen Interaktionsweise (Mensch-System-Kommunikation), die eine effektive Nutzung des Anwendersystems ermöglicht. Wie die Erfahrung zeigt, ist der Anwender, insbesondere wenn komplexe Domänensysteme in Frage kommen, in der Regel außerstande, eine derartige, annähernd vollständige Spezifikation anzugeben. Dies ist auf verschiedene Gründe zurückzuführen. Einer der Gründe betrifft Kommunikationsschwierigkeiten zwischen den beiden Kontrahenden, dem Anwender ('Auftraggeber') und dem EDV-Fachmann ('Auftragnehmer'). Diese Probleme ergeben sich naturgemäß aus der Perspektive zweier unterschiedlicher Fachwelten, Interessensschwerpunkte, der Beurteilung relevanter Sachverhalte, usw. An diesem Sachverhalt ändert sich in der Praxis auch dann nicht viel, wenn sozusagen beide Kontrahenden in einer physischen Person 'vereint' auftreten. Dies kommt z.B. häufig bei SW-Vorhaben im Forschungsbereich einer Universität vor ('Ein-MannTeam').

Zum zweiten wesentlichen Problemkomplex gehört das Auffinden einer geeigneten, logischen Architektur, die eine Umsetzung des einigermaßen abgeklärten Systemkonzeptes (des Anwender-, Domänensystems) in die Realität gestattet. Die architektonische Angemessenheit äußert sich in einer passenden Modularisierung des Domänensystems unter Berücksichtigung der avisierten Implementationsarchitektur. Insbesondere die Güte des architektonischen Grobentwurfes beeinflußt nämlich maßgeblich die Eigenschaften des Endproduktes (das Softwaresystem), die sich dann entscheidend in der Wartungsphase auswirken. Zu den wichtigsten Produkteigenschaften, die zum Teil zusammenhängen, gehören u.a. Anpaßbarkeit, Modifizierbarkeit, Durchschaubarkeit, Portabilität, Erweiterbarkeit, u.v.a.m.

SW-Konstruktion, Strategien, einschlägiges Instrumentarium

Nun gibt es unbestrittenmaßen viele Wege (SW-Erstellungsstrategien), die bei der Konstruktion eines Anwendersystems beschrrieben werden können, deren Wahl aber von einer Fülle einschlägiger Kriterien abhängt. Ein 'vollständiger' Kriterienkatalog wird hier nicht diskutiert. Ein wichtiges, die 'Strategiewahl' beeinflussendes Kriterium ist z.B. die Art des Anwendersystems, sein Umfang, Komplexitätsgrad, die System-Interaktionsart u.ä. Handelt es sich um die Reimplementation eines bestehenden oder um die Schaffung eines gänzlich neuen Anwendersystems, eine Systemnovität wie z.B. häufig in der Forschung ?, usf.

Bei weiteren Überlegungen soll von der Annahme ausgegangen werden, daß ein systematisches, diszipliniertes Vorgehen nach einer gewählten 'Strategie' eher zu einem Ziel und Zielprodukt führt, als ein konzeptloses Probieren und Herumsuchen. Diese Annahme inkludiert den Einsatz einschlägiger Instrumentarien, die in einen methodischen Überbau (einen Vorgehensrahmen) und eine den Überbau effizient unterstützende Technik unterteilt werden können. Beide Kategorien, zusammengenommen, sind für einen erfolversprechenden CASE-Einsatz von zentraler Relevanz.

Oberstes Gebot der SW-Erstellung, methodische, disziplinierte Vorgehensweise

Eines der größten Probleme in der zielführenden Anwendung der bereits verfügbaren CASE-Technologie besteht vermutlich in der Reversion von Prioritäten bei der Anwendung des Instrumentariums. Im allgemeinen wird nach 'greifbaren' tools gefragt, mit einer spürbaren Überbetonung des tool handling, die 'greifbare' Ergebnisse liefern ('je früher, desto besser'). Der Beschäftigung mit dem methodischen Überbau sowie der Konzeptualisierung wird im allgemeinen geringere Bedeutung beigemessen (von manchen auch als graue 'Theorie' abgetan, für die man keine Zeit hat). Dadurch fehlt oft die wesentliche Grundlage für ein systematisches, zielorientiertes Vorgehen. In diesem Zusammenhang spricht man auch sehr oft von einer 'Schulungsproblematik'.

Die Einhaltung einiger grundlegender Prinzipien und Techniken sowie methodisches Vorgehen bei der SW-Erstellung garantieren zwar noch keinen Erfolg, sie sind aber für diesen die Grundvoraussetzung. Ausgehend von methodischen Grundlagen wäre es grundsätzlich möglich, 'gute' Software zu konstruieren, auch wenn keine effiziente Unterstützung eingesetzt wird (z.B. mit Papier und Bleistift). Das dies möglich ist, lassen einige Fälle aus den 70er Jahren ('Jahrzent der SE-Methoden') deutlich erkennen.

Es muß betont werden, daß es stets der SW-Konstrukteur ist, der den entscheidenden, treibenden Teil eines SW-Erstellungsprozesses darstellt, und zwar mit seinen Ideen, Vorstellungen und Einfällen. Die Berücksichtigung eines angemessenen methodischen Rahmens hilft ihm dabei entscheidend, auf ein 'vorgedachtes', gewähltes Ziel systematisch hinzusteuern.

Die Einhaltung dieses Rahmens erfordert jedoch eine Disziplin. Diese kann entweder vom Konstrukteur selbst auferlegt werden, oder eine disziplinierte Vorgehensweise wird durch eine andere Instanz 'erzwungen' (z.B. durch ein tool). Die Praxis zeigt jedoch, daß die letztere Variante oft zu einem gänzlich unerwünschten Effekt führt. Das Kreativitätsspektrum des Konstrukteurs kann dadurch u.U. beträchtlich beschnitten werden. Demgegenüber beläßt eine disziplinierte Einhaltung und Befolgung auch nur eines bescheidenen, methodischen Rahmens, in Eigenverantwortung, einer kreativen Tätigkeit ein breiteres Wirkungsfeld.

Effiziente Umsetzung von Konzepten durch geeignete Werkzeuge

In der Praxis stellt sich aber heraus, daß der Werkzeuffizienz gleichfalls eine wichtige Rolle zukommt. Denn z.B. die Spezifikation eines komplexeren Anwendersystems sowie dessen architektonischer Entwurf ist in der Regel ein iterativer Prozeß, bei dem eine erstaunliche Menge unterschiedlicher Objekte anfällt (Spezifikation, Design, Sourcecode, Listings, mehr oder minder umfangreich, als Zeichen, Graphik dargestellt, usw.), die oft in mehreren Versionen, laufend verwaltet werden müssen. Diese Verwaltung umfaßt Objekterstellung, Modifikation, effiziente Speicherung, u.a.m. Es ist gerade diese routinemäßig durchzuführende Verwaltungsarbeit, zu der u.a. auch unterschiedliche Konsistenzprüfungen gehören, die dem SW-Konstrukteur abgenommen werden muß (und kann).

Den tools kommt dabei die Aufgabe zu, Vorstellungen des Konstrukteurs entsprechend effizient in die Tat umzusetzen. Die tools können ihn in keiner Weise ersetzen, wie manchmal gemutmaßt wird. Insbesondere kann von den tools keine 'automatische' Abgrenzung einer gestellten Problemstellung vorgenommen bzw. die Erstellung eines 'optimalen' Systemdesigns oder sonstige kreative Aktion erwartet werden. Trotzdem kann auch eine im obigen Sinn 'beschränkte' tool-Unterstützung gewaltig sein. Dadurch wird es oft dem SW-Konstrukteur erst ermöglicht, daß er sich mit anwenderspezifischen Problemen überhaupt gründlich befassen kann.

SW-Lebenszyklus

Die Produktion eines Anwendersystems als finales, wartungsfreies, Anwenderwünsche restlos erfüllendes Produkt hat sich als eine Illusion erwiesen. Ein Anwendersystem als das sogenannte 'Endprodukt' unterliegt in der Regel einer sich über mehrere Jahre (bis Jahrzehnte) erstreckenden Evolution. Bei komplexeren Systemen ist der Erstellungsaufwand enorm, daher die Begründung für ein evolutives Vorgehen durchaus einsichtig. Insbesondere

müssen die in der 'Wartungsphase' zu erwartenden Korrekturen, Anpassungen, Modifikationen u.d.g.m. rechtzeitig, nämlich in der Spezifikations- und Entwurfsphase berücksichtigt werden, um die Gesamtkosten für das Anwendersystem so gering wie möglich zu halten. Deshalb kann die Existenz eines Anwendersystems in etwa folgende 'Lebensabschnitte' unterteilt werden: Konzeption, Spezifikation, Entwurf, Implementation, Integration, Betrieb und Wartung.

SW-Erstellungsstrategien

Je nach der Reihenfolge, in der die Abschnitte (auch Phasen genannt) während der SW-Entwicklung einander folgen, kann von unterschiedlichen Erstellungsstrategien gesprochen werden, mit mehr oder minder ausgeprägten Abschnittübergängen. Es wurden unzählige papers darüber geschrieben, welche Strategie die beste darstelle. Mittlerweile setzt sich die Meinung durch, daß die 'optimale' Erstellungsstrategie wesentlich von den Eigenschaften des zu erstellenden Anwendersystems abhängt. Im folgenden werden, natürlich nur skizzenhaft, einige Strategien kurz angedeutet.

Zu den 'ältesten' Erstellungsstrategien gehört das Phasenmodell, das von der grundlegenden Annahme ausgeht, daß sich ein Anwendersystem ähnlich wie eine Kurbelwelle (Taylor-Paradigma) erstellen läßt. Demnach werden die oben genannten Abschnitte mit 'klar' definierten Schnittstellen linear abgearbeitet. Die Praxis hat gezeigt, daß das Taylor-Paradigma nur in Ausnahmesituationen zum Erfolg führt. Das Phasenmodell ist aber aus pädagogischen Gründen weiterhin interessant.

Unter Rapid Prototyping (RP) ist eine Kategorie von Erstellungsstrategien zu verstehen, die viele Befürworter und Gegner hat. Das primäre Anliegen dieser Strategie liegt in der schnellen Erstellung eines lauffähigen, vorführbaren Prototyps, wobei dieser dann unterschiedlichen Zwecken dienen kann; als Endprodukt, das einer weiteren Evolution unterliegt, oder als 'Zwischenprodukt', das als Vorlage für eine 'saubere' Implementierung herangezogen wird. Charakteristisch für RP ist die 'oberflächliche' Behandlung einiger Abschnitte, z.B. Konzept und Spezifikation nur in Grundzügen, mit zyklischen Durchläufen Entwurf, Implementation, Integration.

Operationale Spezifikation oder transformatorische Implementation als Strategien verfolgen das Ziel, den ganzen Weg ab der Systemspezifikation durch Automatismen abzudecken. Der Weg bis zur (vorzugsweise) formalen Spezifikation des Anwendersystems bleibt dem SW-Konstrukteur und dem Domänenfachmann vorbehalten. Die verlockende Substanz dieser Strategie liegt in der Hoffnung, durch bloße 'Redefinition' des formalen Modells ein neues, lauffähiges System zu bekommen (operationale Spezifikation) oder durch entsprechende Automatismen ein solches generieren zu können (transformatorische Implementation). Aus verschiedenen Gründen ist diese Strategie nicht weit verbreitet (Anwendung von Formalismen, derzeit noch nicht allgemein einsetzbar, usw.)

Nach der Spiral-Strategie werden die Abschnitte Konzept bis Integration in einer Schleife iterativ 'abgearbeitet', wobei vom allgemeinen zum besonderen, vom globalen zum speziellen vorgegangen wird. Es werden zunächst Grundkonturen herausgearbeitet, dann folgt feinere Auflösung, ständig von der Frage nach vertretbarem Herstellungsrisiko begleitet. Nur wenn sich das Risiko in akzeptablen Grenzen hält, wird in der schrittweisen Umsetzung Konzept/Programm fortgeföhren.

Grundlegende Techniken, Methoden

Es gibt eine überschaubare Grundmenge von elementaren Techniken, die in unterschiedlichen Kombinationen in allen Abschnitten der SW-Erstellung zur Anwendung kommen. Zu diesen gehören Abstraktion, Dekomposition, Elaboration, Prüfung, Entscheidung, u.a.m., die hier nicht näher erläutert werden.

Methoden stellen im allgemeinen bewährte Denkschemata, Anleitungen, Ratschläge zur zielföhrenden Durchführung einer Tätigkeit dar.

Im Laufe der letzten 20 Jahre wurden für jeden Abschnitt des Lebenszyklus unterschiedliche Methoden entwickelt. Historisch bedingt, ist das Methodenangebot für jeden Abschnitt nicht gleich. Für die sog. 'back-end'-Phasen (Feinentwurf, Implementation, Integration) gibt es mehr Methoden als für die 'front-end'-Phasen. Die Methoden unterscheiden sich in 'Schnittstellen', die für den Anwender sichtbar sind, insbesondere dann aber im Formalisierungsgrad, in der textuellen oder graphischen Darstellungsart, in der Mächtigkeit zu unterstützender Konzepten, in der Benutzerfreundlichkeit, in der Erlernbarkeit, usw. Es gibt derzeit keine Methode, mit der sich der gesamte Lebenszyklus abdecken läßt. Der gegenwärtig diskutierte objektorientierte Ansatz ist in dieser Hinsicht allerdings vielversprechend.

Die Wahl einer 'optimalen' Methode richtet sich nach Lebenszyklus-Abschnitt und nach der Art des Anwendersystems. Dies hat im allgemeinen zur Folge, daß zur 'optimalen' Abdeckung des gesamten Lebenszyklus eine Methodenkollektion herangezogen wird. Dies führt wiederum zu einer Vielfalt von Methoden-Schnittstellen. Im Rahmen einer bestimmten Methodologie wird versucht, geeignete Methoden auszuwählen und aufeinander abzustimmen (Integrationsaspekt). Dabei wird primär darauf geachtet, daß die Methoden möglichst 'nahtlos' ineinander übergehen, die Anwenderschnittstelle zu allen Methoden gleich oder ähnlich ausfällt bei gleichzeitiger Verfügbarkeit der Methodenvorzüge für den jeweiligen Abschnitt.

CASE-Einsatz, derzeitige Situation

Einem breiten Einsatz von CASE-Werkzeugen steht derzeit noch der verhältnismäßig hohe Einstiegspreis im Wege. Bei der Anschaffung von CASE-Tools ist vieles zu beachten. Im folgenden wird auf einige wichtige Punkte hingewiesen. Die Aufstellung ist nicht vollständig, die Reihung nicht gezielt.

Es ist zuerst sehr darauf zu achten, was sich hinter dem Begriff 'ein integrales CASE-Tool' wirklich verbirgt. Im Prospektmaterial der CASE-Tools anbietenden Firmen wird damit gerne und häufig operiert. 'Ein integrales CASE-Tool' kann demnach etwa folgende Bedeutungen haben; häufig eine Programmierumgebung (sehr oft Editor, Compiler, Debugger in einem, was als 'integral' bezeichnet wird), ein Instrument für Projektmanagement und Projektkommunikation, ein Instrument, das eine oder mehrere (ähnliche) Methoden unterstützt, u.ä.m. 'Ein integrales CASE-Tool' sollte die Abschnitte Konzepterstellung, Spezifikation, Entwurf, Integration und Implementation, entsprechend methodisch und 'nahtlos' mit einer konsistenten Anwenderschnittstelle unterstützen. Derzeit gibt es tatsächlich nur einige Werkzeuge von dieser Art.

Der potentielle Käufer eines CASE-Tools sollte sich zuerst zwei primäre Fragen stellen.

Erstens sollte die Frage geklärt werden, ob eine SW-Entwicklung im 'Großen', unter Einsatz eines größeren Teams erfolgen wird, wobei eine Projektteilung, das Projektmanagement und die Projektkoordination, Qualitätskontrollen, Kommunikationsprobleme, usw. mitberücksichtigt werden müssen. Das kennzeichnende Merkmal einer SW-Entwicklung im 'Kleinen' hingegen ist ein 'Wenig-Mann-Team' (sehr oft 'Ein-Mann-Team'). Natürlich sind dann nur die CASE-Komponenten von Relevanz, die vor allem den SW-Entwicklungs- und Erstellungsprozeß unterstützen. Projektmanagement-Komponenten kommt in dem Zusammenhang eine eher untergeordnete Rolle zu.

Die zweite generelle Frage betrifft die Art der zu erstellenden Anwendersoftware. Steht die ökonomische Erstellung einer bekannten, dagewesenen oder ähnlichen Anwendersoftware im Vordergrund (Reimplementation, Erweiterung, Modernisierung, usw.), oder handelt es sich um ein Forschungsvorhaben, die Realisierung eines neuartigen Domänensystems, ein Software-Neuland?

Die Benutzungsart eines CASE-Werkzeuges sollte nicht vom Werkzeug selbst 'erzwungen', sondern den Präferenzen des Anwenders überlassen werden. Ein 'gutes' CASE-Werkzeug bietet dem Anwender vor allem eine einheitlich aufgebaute, konsistente (fensterorientierte) Schnittstelle. Die Ergebnisse der einzelnen Abschnitte sollten auf Konsistenz hin jederzeit geprüft werden können. Es sollte möglich sein, jederzeit nach Anwenderpräferenzen formatierte (Status-)Reports automatisch zu generieren. Die Fähigkeit eines transparenten Zugriffes auf ein Netzwerk kann u.U. von Nutzen sein. Im Zeitalter leistungsfähiger, kostengünstiger PC's und Workstations ist eine SW-Entwicklung nach dem Host/Target-Konzept günstig (Entwicklung am Host-, Betreiben am Target-Rechner). Dies gilt auch für eine Entwicklung im 'Großen', wobei auf die zentrale Objektbank übers Netzwerk zugegriffen werden kann.

Ausblick

Die Konstruktion komplexer, durchschaubarer, wartbarer Anwendersysteme, die auf Akzeptanz stoßen sollen, wird ohne den Einsatz geeigneter Methoden und Werkzeuge zu einem immer größer werdenden Problem. Es ist zu hoffen, daß eine weiter zu erwartende Verbesserung des Preis/Leistungsverhältnisses für die angebotene Computerleistung den breiten Zugang zur CASE-Technologie ermöglicht.

CASE-Einführungskurse am EDV-Zentrum

Das EDV-Zentrum der TU Wien hat ab Anfang dieses Jahres zwei Einführungskurse in das Ausbildungsprogramm aufgenommen, die auf die Problematik des Software Engineering (SE) eingehen. Es soll damit den Kollegen, die sich im Rahmen ihrer Tätigkeit mit der SW-Entwicklung befassen, eine Unterstützung angeboten werden.

Der erste Kurs in Software Engineering (SE, Dauer ca. 15 Stunden), 'Einführung in Computer-Aided Software Engineering' (CASE), ist für Angehörige aller Fakultäten gedacht. Es werden keine speziellen EDV-Kenntnisse vorausgesetzt, einschlägige EDV-Erfahrungen sind jedoch von Vorteil.

Dieser Einführungskurs geht von der primären Zielsetzung aus, die Problematik der SW-Erstellung von unterschiedlichen Standpunkten aus zu beleuchten sowie verfügbare, praktisch anwendbare Techniken, Methoden und Konzepte zu diskutieren. Der Einführungskurs ist herstellerunabhängig konzipiert. Die Befassung mit 'Kochrezepten' steht nicht im Vordergrund.

Folgende Themen bilden den Kursinhalt:

- Zielsetzung, Ausgangslage, Problembereiche
- Ein geschichtlicher Abriß
- System, Umwelt, Modelle, Formalismen
- Grundlegende Techniken, Methoden
- SW-Erstellungsstrategien, bekannte Modelle

- Integrated Project Support Systems (IPSE's), Eigenschaften
- IPSE ProMod, Präsentation

Der zweite Einführungskurs (Dauer ca. 9 Stunden), 'Einführung in das IPSE ProMod', befaßt sich mit der Benutzung des konkreten SW-Entwicklungswerkzeuges ProMod. Dieses Werkzeug unterstützt primär die 'front-end'-Phasen eines Entwicklungsprozesses, d.h. die Systemanforderungs-Definition, Systemgrobentwurf, Feinentwurf. Die Einsatzmöglichkeiten dieser Umgebung sollen an konkreten Beispielen demonstriert werden.

Für beide SE-Kurse sind schriftliche Unterlagen vorbereitet. Kurstermine können im Sekretariat des EDV-Zentrums, Abt. PRA, Fr. Pulzer, Kl.3606, oder bei mir, Kl. 3612 erfragt werden.

Literaturhinweise

- <Agrewala79> Agrewala T.: Putting Petri Nets to Work, Computer, Dec. 1979
- <Bernstein87> Bernstein P.A.: Database System Support for SW Engineering-An Extended Abstract-, ACM-0270/5257/87/300/166
- <Bersoff80> Bersoff E.H., Henderson V.D., Siegel S.G.: Software Configuration Management, Prentice-Hall, 1980
- <Blom88> Bjorn-Andersen N.: Are 'Human factors' Human ?, The Computer Journal, Vol.31, No.5 1988
- <Boehm87> Boehm B.W.: Improving SW Productivity, 0018-9162/87/0900-0043, IEEE Software, Sep. 1987
- <Boehm88a> Boehm B.W.: A Spiral Model of SW Development and Enhancement, 0018-9162/88/0500-0061
- <Borstow87> Borstow D.: AI and SE, ACM-0270/5257/87/300/200
- <Caine75> Caine S.H. Gordon E.W.: PDL: A Tool for SW Design, AFIPS Proc. NCC, Vol. 44, 1975
- <Chikofsky88> Chikovsky E.J.: SW Technology People, 0740-7459/88/0300/0050, IEEE Software
- <Chikofsky88a> Chikovsky E.J., Rubenstein B.L.: CASE: Reliability Engineering for Information Systems, 0740-7459/88/0300/0011, IEEE Software
- <Clement83> Clement T.: The formal specification of a conference organising system, Oxford Univ. Computing Lab., Techn. monograph PRG-36, Aug. 1983, Auszüge in <Mitchell84>
- <Dahl72> Dahl D.J., Dijkstra E.W., Hoare C.A.R.: Structured Programming, Academic Press, 1972
- <Cotterman81> Cotterman W.W.: Systems Analysis and Design: a Foundation for the 1980's, IEEE workshop, Elsevier NH, 1981
- <Curtis87> Curtis B., Krasner H., Shen V., Iscoe N. (MCC): On Building Software Process Models Under the Lamppost, ACM 0270-5257/87/0300/0096
- <Dart87> Dart S.A. at al.: SW Development Environments, 0018-9162/87/1100-0018, Computer, Nov. 1987
- <DeMarco78> DeMarco T.: Structured Analysis and System Specification, Yourdon press, 1978
- <Denert88> Denert E.: Ein Vierteljahrhundert Software Engineering, online 9/88
- <DoD82> DoD: Ada Methodologies: Concepts and Requirements, Nov. 1982
- <Dombrine81> Dombrine, M.: The Warmer Approach to EDP Problems, in Systems Analysis and Design: A Foundation for the 1980's, Ed. Cotterman W.W. at al, North Holland, 1981
- <Dowson86> Dowson M.: ISTAR - An Integrated Project Support Environment, ACM 0-89791-212-8/86/0012/0027
- <Feigenbaum83> Feigenbaum E. A., McCorduck: The Fifth Generation, PAN BOOKS, 1983
- <Freeman81> Freeman P.: "Why Johnny Can't Analyse ?", in <Cotterman81>
- <Frenkel85> Frenkel K.A.: Toward Automating the SW-Development Cycle, ACM Comm., Nr.6, June 1985
- <Gane79> Gane C., Sarson T.: Structured Systems Analysis, Englewood Cliffs, Prentice Hall, 1979
- <Hass84> Hasse V. H., Hammer R.: Softwaretechnologie für die Industrie, Proc's of the European Seminar on Industrial Software Engineering and EWICS, 1984
- <Hatley88> Hatley D.J., Pirbhei I.A.: Strategies for real time system specification, Dorset House Pub.s, NY, 1988
- <Hoare87> Hoare C.A.R.: An Overview of Some Formal Methods for Program Design, 0018-9162/87/0900-0085, Computer, Sep. 1987
- <Henderson87> Henderson P.B.: Integrated Design and Programming Environments, Computer, Nov. 87
- <Hruschka87> Hruschka P.: Methoden zur Entwicklung von Echtzeitsystemen, Design & Elektronik, Nr.3, 1987.
- <Hudson88> Hudson S.E., ing R.: The CACTIS Project: Database Support for Software Environments, ACM-98-5589/88/600/709
- <Humphrey88> Humphrey W.S.: Characterising the Software Process: A Maturity Framework, 0740-7459/88/0300/0073, IEEE Software, Mar. 1988
- <Fischer87> Fischer G., Lemke A.C., Rathke Ch.: From Design to Redesign, ACM-270/5257/87/300/369
- <Hurley83> Hurley R.B.: Decision Tables in SW Engineering, Van Nostrand Reinhold, 1983
- <Kaiser87> Kaiser: An Architecture for Intelligent Assistance in Software Development, ACM-270/5257/87/300/180
- <Jackson75> Jackson M.: 'JDM', 1975
- <Kelly87> Kelly J.C., Feiler P.H. : A Comparison of four Design Methods for Real-Time Systems, ACM-0270-5257/87/300/238 (Methoden im Vergleich : Str. Design for Real-time Systems, Object Oriented Design, PAMELA, SCR)
- <King88> King M.C.: Interdisciplinarity and Systems Thinking, European Journal of Engineering Education, Vol.13, Nr.3, 1988
- <Koen88> Koen B.V.: Toward a Definition of the Engineering Method, European Journal of Engineering Education, Vol.13, No.3, Fall 1988
- <Knuth84> Knuth E., Ludwig J.: On specification techniques, in Softwaretechnologie für die Industrie, ÖCG Schriftenreihe, Bd. 23, Proc. EWICS, 1984
- <Lehman85> Lehman M.M., Belady L.A.: Program Evolution, Academic Press, 1985

- <Lehman87> Lehman M.M.: Process Models, Process Programs, Programming Support, ACM 0270-5257/87/0300/0014
- <Lubars87> Lubars M.D., Harandi M.T.: Knowledge-Based SW- Design Using Design Schemas, ACM 0270/5257/87/300/253
- <Luqi88> Luqi Berzins V.: Rapidly Prototyping Real-Time Systems, 0740-7459/88/0900/0025, Computer, Sep. 1988
- <Martin88> Martin C.F.: Second-Generation CASE Tools: A Challenge to Vendors, 0740-7459/88/0300/046
- <Mills86> Mills H.D.: Structured Programming: Retrospect and Prospect, IEEE Software, Nov. 1986
- <Mitchell84> Mitchell R. J.: The Technical Foundations of Software Engineering, in <Hasse84>
- <McMebamin88> McMenamin S.M., Palmer J.F.: Strukturierte Systemanalyse, Hanser 1988
- <MüllerDom88> Müller-Dom G., Martin W.: Ein Fall im Rampenlicht: CASE, Elektronik Dez. 1988
- <MüllerMerb83> Müller-Merbach H.: Systemanalyse als gelenkter kreativer Prozeß, in <Schelle83>
- <Myers75> Myers G.J.: Reliable Software Through Composite Design, Van Nostrand Reinhold, 1975
- <Nassi73> Nassi, Shneidermann: NS charts, 1973
- <Orr77> Orr K.T.: Structured Systems Development, Yourdon Press, NY, 1977
- <Orr81> Orr K.T.: Structured Requirements Definition, Ken Orr & Assc's, Inc., Topeka, KS, 1981
- <Osterweil87> Osterweil L.: Software Processes Are Software Too, ACM-0270/5257/87/300/002
- <Pamas71> Pamas D. L.: Information Distribution Aspects of Design Methodology, Proc. IFIP Congress 71, North Holland, 1971
- <Pamas85> Pamas D.L., Clemens P.C., Weiss D.M.: The Modular Structure of Complex Systems, IEEE Trans. on SW-Eng., März, 1985
- <Penedo88> Penedo M.H.: Guest Editors' Introduction in SEE Architectures, IEEE Trans. on SW Engineering, June 1988
- <Puncello88> Puncello P.P., et al.: ASPIS: A Knowledge-Based CASE Environment, 0740-7459/88/0300/058
- <Rich88> Rich, Ch., Watters C.: Automatic Programming: Myths and Prospects, 0018-9162/88/0800-0040, IEEE, Computer, Aug. 1988
- <Ross77> Ross D.T.: Structured Analysis (SA) : A Language for Communicating Ideas, EH0205-5/83/0000/0096, IEEE Trans. on SW Eng. Jan. 1977
- <Ross77a> Ross D.T., Schoman K.E.: Structured Analysis for Requirement Definition, EH0205-5/83/0000/0096, IEEE Trans. on SW Eng. Jan. 1977
- <Royce70> Royce W.W.: Managing the Development of Large Software Systems: Concepts and Techniques, 1970, reprint, proc's of the intern. conf. of SE, Monterey CA, Apr. 1987
- <Scharf88> Scharf A.: Spaghetti ade ?, c't Heft 11, 1988
- <Shatz87> Shatz S.M., Wang J-P.: Introduction to Distributed Software Engineering, 0018-9162/87/0900-0023, Computer, Oct. 1987
- <Schelle83> Schelle H., Molzberger P.: Psychologische Aspekte der Software-Entwicklung, Oldenbourg, 1983
- <Shere87> Shere K.D.: Myths Related to the Software Development Live Cycle, TH0181-8-87/0000/0134
- <Snodgrass87> Snodgrass J.G., Yun D.Y.Y.: Challenging the "Conventional" Software Engineering Paradigm, TH0181-8/87/0000/0139
- <Simon86> Simon H.A.: Whether SW Engineering Needs to be Artificially Intelligent, 0098-5589/86/0700-0726, IEEE Trans. on SW Eng., July 1986
- <Smestad88> Smestad, T., Andersen K.: Introduction of the Concept "Integrated Projection Illustrating", SW-Eng. Notes, Jan. 1988
- <Stenning87> Stenning V.: On the Role of an Environment, ACM 0270/5257/87/300/340 <Tully87> Tully C.J.; Prospects for Future Environments, panel session, ACM 0270/5257/87/300/340
- <Stephens84> Stephens M., Whitehead K.: The "Analyst" - An Expert Systems Approach to Requirements Analysis, in ÖCG-Schriftenreihe, Bd.23, Softwaretechnologie für die Industrie, Proc. EWICS, 1984
- <Sorenson88> Sorenson P.G., Tremblay J-P.: The Metaview System for Many Specification Environments, 0740-7459/88/0300/0030
- <Taylor85> Taylor B.: A Database Approach to Configuration Management of Large Projects, Proc. Conf. Software maintenance, IEEE, Nov. 1985
- <Teichroew77> Teichroew D., Hershey E.A.: PSL/PSA; a Computer Aided Technique for Structured Documentation and Analysis of Information Processing Systems, IEEE Trans. on SE, SE-3(1), 1977
- <Wamier J.D.: Logical Construction of Programs, Van Nostrand Reinhold, 1974
- <Wasserman79> Wasserman A.I., Stinson S.K.: A Specification Method for Interactive Information Systems, Proc.: Specifications of reliable sw, IEEE, 1979
- <Wasserman80> Wasserman A.I.: Information System Design Methodology, J. of the American Society for Information Science, Jan. 1980
- <Wasserman82> Wasserman A.I.: The User Engineering Methodology: An Overview, aus "Info Systems Design Methodologies: A Comparative View, Olle T.W., Sol H.G. Verrijn Stuart A.A. (editors), North-Holland", IFIP 1982
- <Wasserman86> Wasserman A.I., Pircher P.A.: A Graphical, Extensible Integrated Environment for SW-Development, ACM 0-89791-212-8/86/0012/131
- <Wesseler88> Wesseler B.: Was bringt uns CASE ?, online 11/88
- <Yoder78> Yoder C.M., Schrag M.L.: Nassi-Shneiderman Charts, An Alternative to Flowcharts for Design, Proc. of ACM SIGSOFT/SIGMETRICS SQ and Assurance Workshop, Nov. 1978
- <Young88> Young M., Taylor R.N., Troup D.B.: Software Environment Architectures and User Interface Facilities, ACM-98-5589/88/600/697
- <Yourdon79> Yourdon E., Constantine L.L.: Structured Design: Fundamentals of a Discipline of Computer Program and System Design, Prentice Hall, 1979
- <Zave82> Zave P.: An Operational Approach to Requirements Specification for Embedded Systems, IEEE Trans. on SW-Eng., May 1982

Zur Bearbeitung nichtlinearer Gleichungssysteme mit MAPLE

Gerhard Kranner *)

In der theoretischen Physik treten zuweilen mathematische Probleme auf, deren exakte, symbolische Lösungen qualitativ neue Erkenntnisse im jeweiligen Bereich ermöglichen. Ein typischer Fall hierfür sind die Endlichkeitsbedingungen in der Quantenfeldtheorie, deren exakte Lösungen in 1. Ordnung zur Klassifizierung aller divergenzfreen Feldtheorien, die über kompakten Lie-Gruppen formulierbar und renormierbar sind, führen könnten. Im Zentrum steht dabei ein nichtlineares Gleichungssystem 3. Ordnung, welches als Unbekannte die komplexen Variablen

$$Y_{ij}^a = Y_{ji}^a, \quad a=1..m, \quad i,j=1..n$$

enthält, das sind wegen der i-j- Symmetrie $2*m*n(n+1)/2$ reelle Unbekannte. Das System hat die Gestalt

$$\sum (4 Y^a (Y^b)^* Y^b + Y^a (Y^b)^* Y^b + Y^b (Y^b)^* Y^a \quad (*)$$

$$+ Y^b * \text{trace}(Y^a (Y^b)^* + Y^b (Y^a)^*), \quad b=1..m)_{ij} = Y_{ij}^a (c_i + c_j),$$

wobei $(Y^a Y^b)_{ij} := \sum (Y_{ik}^a Y_{kj}^b, k=1..n)$ bedeutet, c_i = reelle Parameter

und $(Y^a)^* :=$ komplex konjugiert(Y^a).

Ergänzend zu analytischen und topologischen Betrachtungen über die Struktur der Lösungen von (*) ist es mit Hilfe von MAPLE möglich, sich für kleine Dimensionen von m und n einen Überblick über deren Vielfalt und Gestalt zu verschaffen.

Schon im einfachsten Fall, $n=1$, m = beliebig, wo (*) zu

$$5 (Y^a)^* \sum ((Y^b)^2, b=1..m) + 3 Y^a \sum (Y^b (Y^b)^*, b=1..m) - 2 Y^a = 0, \quad (**)$$

mit $Y_{ik}^a = Y^a$, $c_i = c = 1$ gesetzt,

reduziert, stellt sich als erstes die Frage: Wie sind die komplex konjugierten Variablen $(Y^a)^*$ sinnvoll zu behandeln? Es zeigt sich hier - wie auch in anderen Fällen - folgendes: Sowohl eine Zerlegung von Y in Real- und Imaginärteil,

$$Y^a = X^a + i Z^a, \quad (Y^a)^* = X^a - i Z^a,$$

als auch die Verwendung seiner Polarform

$$Y^a = R^a \exp(i P^a), \quad (Y^a)^* = R^a \exp(-i P^a),$$

erweisen sich als nachteilig, weil dann

1. alle Gleichungen in allen Variablen nichtlinear werden (im Unterschied zu (*) oder (**), die immerhin linear in $(Y^a)^*$ sind) und

2. man für jede Lösung deren Realität überprüfen sowie für den Fall, daß Wurzeln auftreten, unterscheiden muß, ob sie überhaupt existieren, etc.

Bei Verwendung der Polarform kommt zudem noch die Erschwernis, e- Potenzen umzuformen oder zu vereinfachen. Da das System ohnehin komplexe Wurzeln erlaubt, ist es also hier am einfachsten, Y^a und $(Y^a)^*$ selbst als Variablen beizubehalten.

Welche Vorgangsweise ist nun weiter sinnvoll? In MAPLE- Notation lautet (**) etwa:

```
SYSTEM:= proc(m) for a to m do
```

```
S.a := 5 * CY[a] * sum('Y[b]^2', 'b'= 1..m) + 3 * Y[a] * sum('Y[b] * CY[b]', 'b'= 1..m) - 2 * Y[a];
```

```
CS.a := subs( [Y=CY, CY=Y], S.a )
```

```
od end;
```

wobei $Y[a] = Y^a$ und $CY[a] = (Y^a)^*$ entsprechen. Zu jedem Index a in (**) gehört also eine Gleichung S.a und daraus durch Vertauschung $Y \leftrightarrow CY$ ihre komplex konjugierte CS.a. Man könnte nun versuchen, alle Gleichungen direkt nach allen Variablen aufzulösen, etwa für $m=2$ mittels

*) Dipl. Ing. Gerhard Kranner ist Dissertant am Institut für Theoretische Physik

```
solve( {S1, S2, CS1, CS2}, {Y[1], Y[2], CY[1], CY[2]} );
```

Dies ist für kleine Dimensionen m noch praktikabel, stößt aber wegen der Nichtlinearität für wachsendes m sehr bald auf Rechenzeit- und memory- Grenzen. Viel besser ist es, vorerst das in $CY[a]$ lineare System

```
{ S1, S2, ... Sm } nach { CY[1], CY[2], ... CY[m] }
```

aufzulösen, woraus man für variiertes m schnell erkennt, daß

$$CY[a] = Y[a] / (4 * \sum('Y[b]^2', 'b'=1..m)) .$$

Die Lösung des komplex konj. Systems ist dann einfach die komplex konj. Lösung

$$Y[a] = CY[a] / (4 * \sum('CY[b]^2', 'b'=1..m)) .$$

Im allgemeinen würde man nun weiter so vorgehen, daß man dieses $Y[a]$ in obige Lösung für $CY[a]$ einsetzt und das so erhaltene, nichtlineare System nach $CY[a]$ auflöst. Hier liefert diese Substitution zufällig die Trivialität

$$CY[a] = CY[a] / (4 * \sum(CY[b]^2) / (4 * \sum(CY[c]^2))) = CY[a] .$$

Die Ursache dafür ist, daß mit $CY[a]$ eigentlich schon die gesamte Lösung vorliegt, denn es folgt ja

$$Y[a] \quad Y[d]$$

$$CY[a] \quad CY[d]$$

d.h. wenn man setzt: $Y[a] = R[a] \exp(i P[a])$ etc., so gilt

$$\exp(2i P[a]) = \exp(2i P[d]) = \dots ,$$

$$\text{also } P[d] = P[a] \pm i P_i \quad \text{für alle } a, d, \dots$$

was offenbar höchstens für 2 verschiedene Indizes a und d erfüllbar ist; es sind also höchstens 2 verschiedene $Y[a]$ ungleich Null, usw. Man erkennt an der Lösung für $CY[a]$ vor allem auch, daß von vornherein der Fall $\sum(Y[b]^2) = 0$ gesondert zu untersuchen ist.

An diesem einfachen - und tatsächlich auch "am Papier" sehr leicht zu lösenden - Fall (**) sieht man exemplarisch, daß man

1. mit komplexen Variablen vorteilhaft umgeht, indem man zuerst ihre komplex Konjugierten als "Parameter" im System stehen läßt, dann nach den unkonjugierten Variablen auflöst, diese Lösungen konjugiert und als "Parameter" in die Lösungen selbst substituiert und

2. nach Möglichkeit lineare Subsysteme abspalten und vorrangig lösen soll.

Die Beachtung dieser Vorgangsweise wird bei der Behandlung des "vollen" Systems (*) schon im Fall $n = 2$, $m = 2$ essentiell; dieser enthält die 6 Unbekannten

$$Y_{ij}^a = Y_{ji}^a, \quad a, i, j = 1..2$$

und ihre Konjugierten $(Y_{ij}^a)^*$. Selbst der Versuch, das System (*) nach letzteren linear aufzulösen, stößt bei Verwendung der MAPLE- Funktion 'solve' auf Probleme; die Ursachen hierfür sind genereller Natur:

Bei linearen Gleichungssystemen mit "großen" Koeffizienten (hier Polynome in Y) kann 'solve' leicht die vorgesehenen Speichergrenzen sprengen; auch die package- procedure 'linalg[linsolve]' führt über involvierte Determinanten- Berechnungen oft zu übergroßen Zwischenausdrücken. Es ist für diese Fälle günstig, ein für allemal etwa 4×4 -, 5×5 -, 6×6 - und 7×7 - Determinanten $D := \det a[i,j]$ mit allgemeinem $a[i,j]$ zu berechnen und zu speichern, um dann in einer modifizierten 'linsolve'- procedure die Determinanten zu konkreten Koeffizienten $K[i,j]$ des Systems durch Substitution

$$a[i,j] \rightarrow K[i,j]$$

direkt aus D (mit anschließendem 'expand') zu erhalten.

Zur Manipulation sehr großer Ausdrücke mit 'expand', 'normal', 'factor' und 'collect':

Bei rationalen Ausdrücken A ab einer "MAPLE- Länge" von etwa $\text{length}(A) = 40000$ ist es im Falle von Polynom- Brüchen empfehlenswert, mittels $\text{normal}(A)$ etwaige gemeinsame Faktoren (aus Zähler und Nenner) zu beseitigen, ehe man Teile von A genauer untersucht. $\text{expand}(A)$ und $\text{collect}(A)$ ergeben im Fall involvierter Brüche meist sehr lange, unüberschaubare Ausdrücke; jedoch kann insbesondere 'collect', auf Zähler und Nenner gesondert angewandt, zu beträchtlichen Verkleinerungen führen. Die hervorragende MAPLE- Funktion 'factor' bewältigt Faktorisierungen multivariater Polynome bis zu Größen von $\text{length}(A) = 300\,000$ (was mehrere Stunden CPU- Zeit kosten kann!). Bei großen Ausdrücken ist häufig die Verwendung selbst konzipierter, problemadäquater Prozeduren von Vorteil; z.B. ist es oft günstig, alle Faktoren eines Ausdrucks mittels 'collect' nach einer Variable zu ordnen und dann deren Koeffizienten wieder zu faktorisieren (und evtl. mit diesen Faktoren nochmals dasselbe). Dies führt in vielen Fällen nicht nur zu drastischen Verkleinerungen und besserer Überschaubarkeit, sondern macht auch allfällige algebraische Strukturen oft erst sichtbar.

Mit $m = n = 2$ erreicht die direkte Berechenbarkeit des Systems (*) bereits ihre Grenze. Die eigentliche Vorgangsweise zu dessen Lösung stützt sich auf ein induktives Verfahren, das (bei beliebiger Dimension n) eine schrittweise Erhöhung der Dimension m von m auf $m+1$ erlaubt. Auch hier ist zur Gewinnung induktiver Ansätze und ihrer Verifikation die Lösung von Studienfällen angebracht. Die dabei gewonnenen und oben ausgeführten Erfahrungen dürften sich auch im Umgang mit anderen großen oder komplexen Systemen als bedenkenswert erweisen.



Herrn
Dipl. Ing. F. Blöser
EDV-Zentrum
Abt. Hybridrechenanlage

DVR 5886

I m H a u s e