

---

---

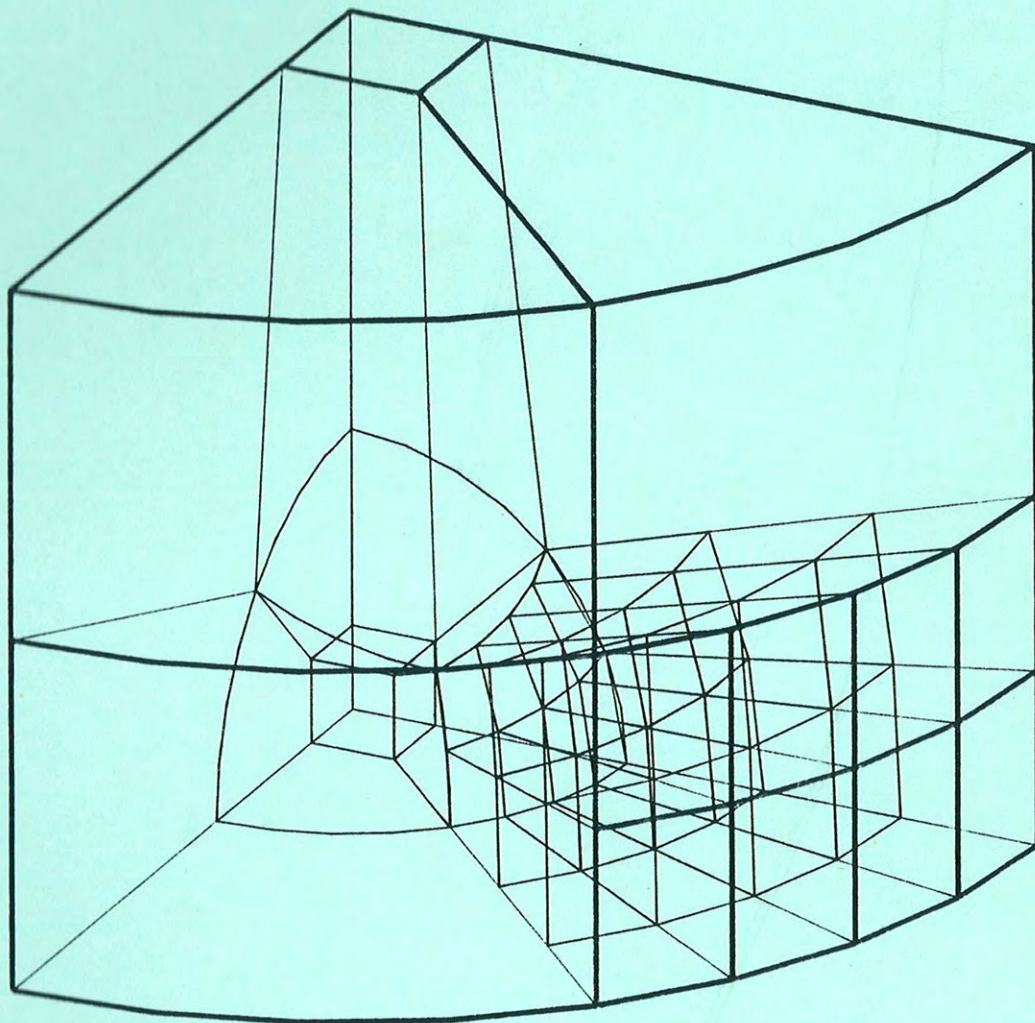
W I D E R B A C K

---

---

HEFT 16

1987



Zerlegung in Finite-Elemente mit EUCLID/EDAS

---

---

**TU**-PRA

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



**INHALTSVERZEICHNIS**

Seite der Redaktion ..... 3

Overheadfolien-Erstellung ..... 4

Ausgabe von Graphik und Text auf dem Laserdrucker ..... 9

Übersicht über die unterstützten Graphik-Geräte ..... 11

Neues von der Calma Anlage ..... 13

3D-Finite-Elemente Preprocessing mit EDAS ..... 14

Der neue österreichische EUNET/USENET Backbone ..... 18

A Computernetwork with DECNET ..... 26

Artificial Intelligence in Software Engineering ..... 34

IMPRESSUM Feedback Nr. 16

Redaktion: Dr. G. Wehrberger, I. Poremba  
Layout: Dipl. Ing. A. Sprinzi  
Herausgeber, Verleger, Hersteller:  
EDV-Zentrum der Technischen Universität Wien,  
Abt. Prozeßrechenanlage, Leitung Dr. M. Paul  
alle: A-1040 Wien, Gußhausstraße 25

Bilderwitze: aus 'Computer-Cartoons', Wang Edition Nr. 1

*Seite der Redaktion*

Die vorliegende Ausgabe des "Feedback" verfolgt wie immer das Ziel, wesentliche Neuigkeiten sowie Schwerpunkte aus der Arbeit der Abt. Prozeßrechenanlage den Benützern unserer Anlagen in Form einzelner Beiträge vorzustellen. Somit, und auch rein äußerlich unterscheidet sich das neueste "Feedback" daher kaum von den bisherigen Ausgaben. Erst im Inneren des Blattes werden Sie feststellen, daß die Gestaltung erstmals mit einem hochwertigen Textverarbeitungssystem (TEX) und unserem Laserdrucker erfolgte. Zwei weitere Möglichkeiten, Text und Graphik auf einem Blatt bzw. einer transparenten Folie zu vereinen, sind unsere erste — mit der vorhandenen Hard- und Software realisierte — Antwort auf das in der Computer-Branche kursierende Schlagwort "Desk Top Publishing".

Aus den Schwerpunktgebieten Datennetze und Graphik-Software (CAD) wird diesmal über Projekte berichtet, die das experimentelle Stadium bereits verlassen haben, und sich im täglichen Betrieb bewähren. Der Beitrag über neue Methoden des Software-Engineering in einer AI-Programm-Entwicklungsumgebung ist hingegen wieder zukunftsorientiert und soll dazu anregen, über die Verwendung zeitgemäßer Software-Entwicklungswerkzeuge nachzudenken.

Wir glauben, mit dieser Nummer des "Feedback" wieder einen Einblick in die verschiedenen Arbeitsgebiete der Abt. Prozeßrechenanlage geben zu können, und hoffen, daß auch für Sie etwas Interessantes dabei ist.

G.W.

## OVERHEADFOLIEN - Erstellung Ein neues Dienstleistungsangebot der PRA

Günther Wehrberger

### **EINLEITUNG**

Der Overheadprojektor wird als billiges und einfach zu bedienendes Visualisierungshilfsmittel heute fast bei jedem Vortrag, und auch immer mehr im Lehrbetrieb eingesetzt. Voraussetzung für die Projektion sind allerdings gut vorbereitete Folien, wobei die händische Erstellung entweder sehr zeitaufwendig ist, oder meistens schlecht lesbare und wenig ansprechende Folien zur Folge hat.

Die Abt. Prozeßrechenanlage hat diesem Umstand Rechnung getragen, und versucht, mit den zur Verfügung stehenden Mitteln eine neue Dienstleistung - die Erstellung von Overheadfolien mit dem Computer - den Instituten der TU-Wien anzubieten. Aufgrund der vorhandenen Hard- und Software stehen zwei verschiedene Möglichkeiten zur Generierung von Overheadfolien zur Verfügung.

### **RASTERGRAPHIK - VERFAHREN**

Zu den großen Vorteilen der Rastergraphik zählt vor allem die einfache Flächenfüllung sowie die zur Verfügung stehende breite Farbpalette. Die UNIRAS - Rastergraphik - Software in Kombination mit einem Inkjet - Plotter sollte vor allem dann verwendet werden, wenn

- einzelne Folien besonders hervorgehoben werden sollen, oder
- Ergebnisse (Meßdaten), die mit der UNIRAS-Grundsoftware aufbereitet wurden, dargestellt werden.

Neben der "unbegrenzten" Farbvielfalt ist auch die große Zahl an Zeichensätzen (Abbildung 1) ein wertvolles Hilfsmittel für die graphisch anschauliche Gestaltung von Folien. Da jeder UNIRAS-Programmlauf zunächst ein Rasterbild als Ergebnis liefert, welches erst in einem zweiten Schritt auf ein geeignetes Ausgabegerät ausgegeben werden kann, ist ein interaktives Arbeiten nicht möglich, was zweifellos als ein gewisser Nachteil angesehen werden muß.

# Verschiedene UNIRAS Zeichensätze \*)

Century bold style (filled)

SWISS BOLD style

Das ist Altdeutsch,

oder Complex Script bzw. Triplex Italic

\*) Siehe auch FEEDBACK Nr.15 (1986)

*Abb.1 : Auswahl aus 31 verschiedenen Zeichensätzen*

## Erstellung von Folien mit beliebigen graphischen Elementen

Um ein Rasterbild, welches dann sowohl auf Papier als auch auf einer Spezialfolie am Inkjet-Plotter ausgegeben werden kann, zu erzeugen, muß ein FORTRAN - Programm erstellt werden, welches die entsprechenden Unterprogramme aus der UNIRAS - Bibliothek enthält. Der Kreativität sind dabei keine Grenzen gesetzt, und die Ergebnisse rechtfertigen im allgemeinen den Programmieraufwand. (Abbildung 2)

## Erstellung von Text - Folien

Da in vielen Fällen der Inhalt einer Folie aus reinem Text besteht, wurde an der Abt. Prozeßrechenanlage ein auf den UNIRAS - Routinen aufbauendes Anwenderprogramm entwickelt, welches auch ohne Programmierkenntnisse das Erstellen von Overheadfolien ermöglicht.

Der Benutzer des Programmes 'INKTXT' kann durch Einfügen von Darstellungsparametern in ein normales Textfile daraus ein UNIRAS-Rasterbild erzeugen. Texthöhe, Farbe der Zeichen, Zeichensatz, Linienbreite, Zeilenabstand, Hintergrundfarbe u.s.w. werden dabei durch die eingefügten Darstellungsparameter festgelegt. Das auf diese Weise entstandene Rasterbild kann dann in einem zweiten Schritt mit dem Treiber-Programm 'INKPLOT' auf dem Inkjet-Plotter ausgegeben werden. Ein Preview der graphischen Darstellung ist in diesem Falle auf einem 'Regis' - Graphikterminal (VT241, VT125) möglich, womit die Ausgabe auf den Inkjet-Plotter erst erfolgen muß, wenn der Text die endgültige Erscheinungsform hat. Die Abbildungen 3 u. 4 zeigen in anschaulicher Weise, wie durch das Einfügen von Kommandosequenzen in das Textfile der Text gestaltet werden kann.

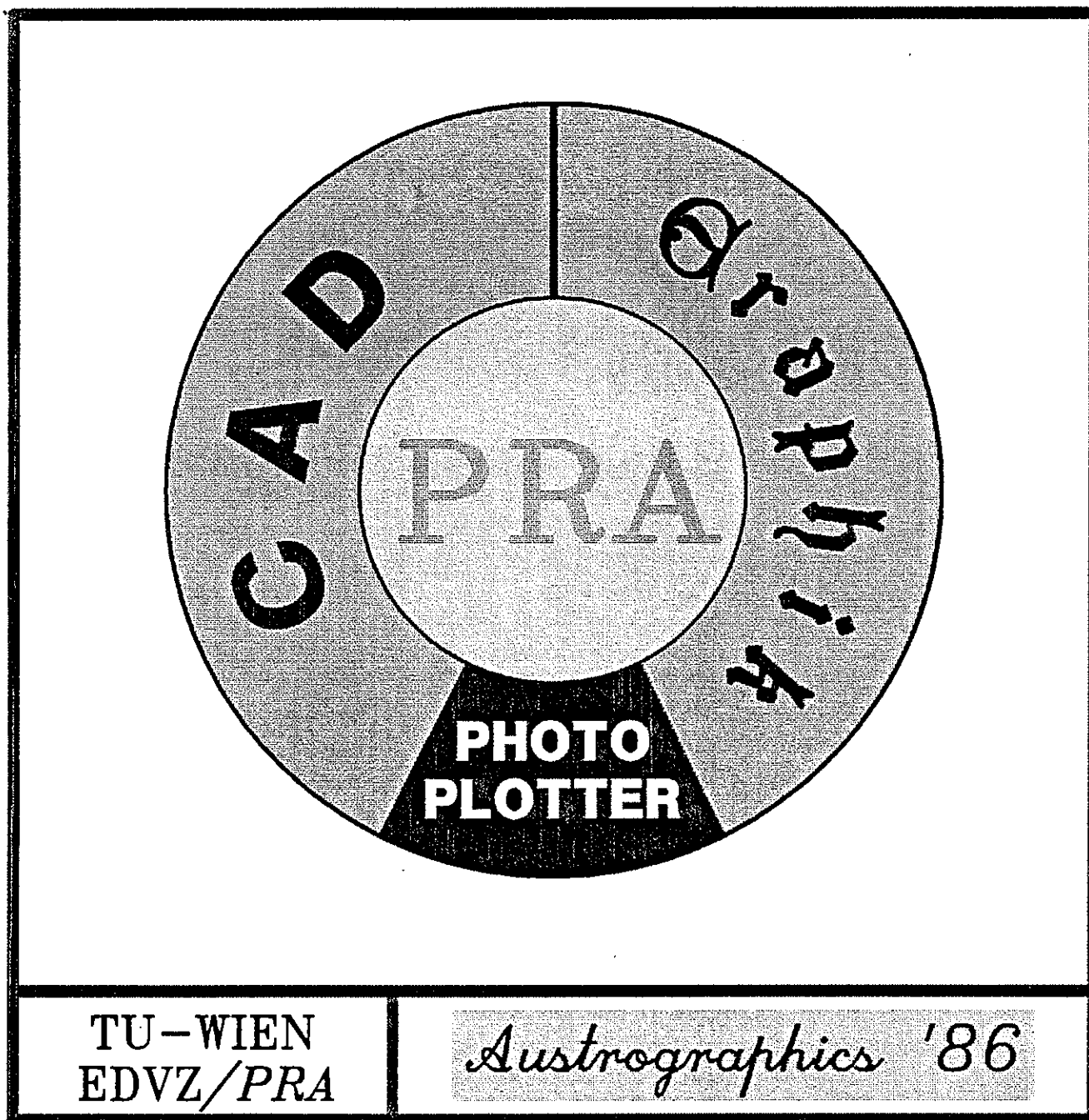


Abb.2 : Die UNIRAS-Software ermöglicht die Erstellung graphisch anspruchsvoller Folien

### VEKTORGRAPHIK - VERFAHREN

Seit kurzem kann die Abt. Prozeßrechenanlage den Benützern noch eine weitere Methode zur Erstellung von Overheadfolien anbieten. Das weltweit meistverkaufte CAD-System für Personalcomputer - AutoCAD - welches an der PRA zur Verfügung steht, bietet die Möglichkeit, interaktiv mit Hilfe des Texteditors und einer Maus als graphischem Eingabegerät, Text und Zeichnungen auf einem einzigen Blatt (Folie) zu vereinen.

OVERHEADFOLIEN - Erstellung



```
#DEF 0 height=6 font=SHA2, center
Beispiele der Textausgabe
mit dem Inkjet-Plotter #DEF 1, nocenter
```

Dieser Text wurde mit einem Programm zur Erstellung von hochwertigen Overheadfolien erzeugt. Man kann damit Textfiles in verschiedenen Schriftarten, Farben und Größen darstellen:

```
#DEF 3 font=ITALIC height=6, 2 font=GOTE height=6, space=11
Text in Gotischen Lettern,<ESC>[13m in Italic,
#DEF 5 font=SHA2 col=green, 4 font=SCRI hei=4 col=blue spac=9
in Script und blau,<ESC>[15m oder in Century in gruener Farbe;
#DEF 7 font=SHA5 height=5 color=white background=red, center
Zentrierter weisser Text vor rotem Hintergrund. #DEF 1, nocenter
```

Darüberhinaus können im Text Indices und Exponenten, wie z.B.  $X_{i,j}$ ,  $x^2$ ,  $A_m^*$  vorkommen. Außerdem steht der komplette Hershey-Zeichensatz mit einer Vielzahl an Sonderzeichen und fremdsprachigen Symbolen zur Verfügung, die an jeder Stelle des Textes eingefügt werden können:

```
#DEF 1 height=6 space=12
z.B. ^638(^638), ^2330<^2325 ^1405^2289^2277^1406>*
^2268<0_>^1270 ^741^744^746^871 ^3801^3302^3503^1030^2805 ...
```

Abb.3 : Textfile mit Darstellungsparametern

### Beispiele der Textausgabe mit dem Inkjet-Plotter

Dieser Text wurde mit einem Programm zur Erstellung von hochwertigen Overheadfolien erzeugt. Man kann damit Textfiles in verschiedenen Schriftarten, Farben und Größen darstellen:

*z.B. Text in Gotischen Lettern, in Italic, in Script und blauer, oder in Century und gruener Farbe;*

**Zentrierter weisser Text vor rotem Hintergrund:**

Darüberhinaus können im Text Indices und Exponenten, wie z.B.  $X_{i,j}$ ,  $x^2$ ,  $A_m^*$  vorkommen. Außerdem steht der komplette Hershey-Zeichensatz mit einer Vielzahl an Sonderzeichen und fremdsprachigen Symbolen zur Verfügung, die an jeder Stelle des Textes eingefügt werden können:

z.B.  $\mu(\uparrow 638)$   $\text{G}_6$   $[\Psi]^*$   $\int_0^\infty$   $\text{Q}\text{Q}\text{Q}\text{Q}\text{Q}\text{Q}$   $\text{H}\text{B}\text{C}\text{D}\text{E}$

Abb.4 : Ausgabe des Textfiles von Abb.3 am Inkjet-Plotter

Ein weiterer Vorteil der interaktiven Arbeitsweise eines CAD-Systems ist die sofortige Darstellung der Eingaben am Bildschirm, sowie die rasche und einfache Änderbarkeit. Um den Folien professionelles Aussehen zu verleihen, kann jedes Blatt außerdem automatisch mit einem Rahmen bzw. institutsspezifischen Kennzeichen versehen werden (Abb. 5). Die Ausgabe kann mit Hilfe eines Gould-Plotters (6120, DIN A3/A4) in sieben verschiedenen Farben auch direkt auf eine Spezialfolie erfolgen.

Für alle jene Benützer, die AutoCAD ausschließlich zur Erstellung von Unterlagen für Lehrzwecke, Vorträge und Veröffentlichungen verwenden wollen, haben wir ein spezielles Menü 'PRATEXT' zusammengestellt. Es befindet sich als vorletztes Untermenü im Hauptmenü und besteht selbst wieder aus einer Anzahl von Untermenüs, mit deren Hilfe in einfachster Weise ein "Blatt" im DIN A4-Format bearbeitet werden kann, wobei für die Auswahl der Farben, der Textfonts sowie die Plotterausgabe eigene, selbsterklärende Untermenüs existieren. Für nähere Auskünfte steht der Autor jederzeit gerne zur Verfügung (Kl. 3617).

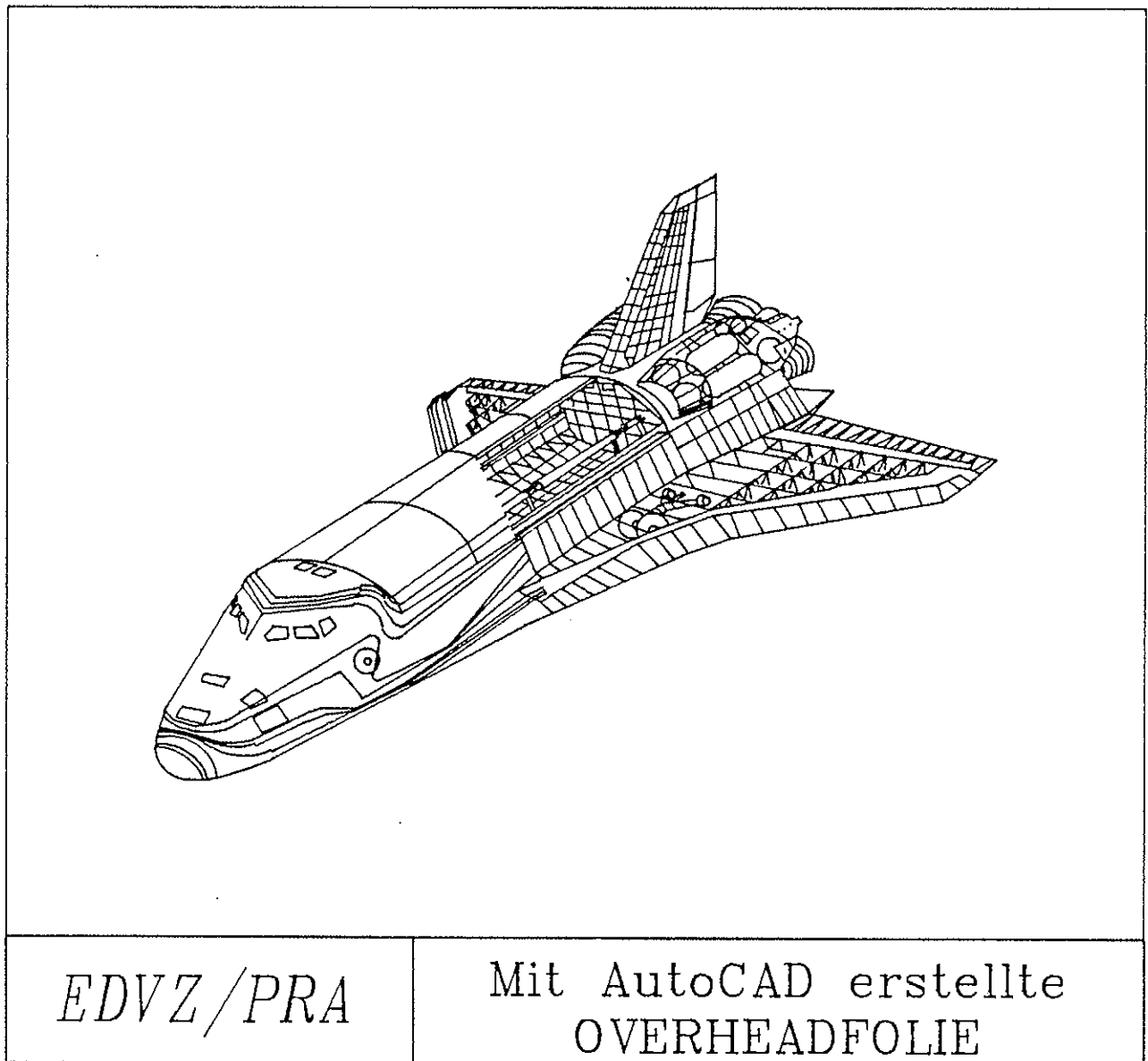


Abb.5 : Beispiel eines Folienrahmens für Vorträge

## Ausgabe von Graphik und Text auf dem Laserdrucker

Rudi Sedlaczek

Nach der Aufrüstung der beiden Laserdrucker der Prozeßrechenanlage mit 1MB lokaler Speicherkapazität und neuer Firmware zu graphikfähigen Modellen LN03-PLUS steht nunmehr sowohl am Karlsplatz als auch in der Gußhausstraße ein Gerät zur Erstellung hochwertiger Graphiken mit einer Auflösung von maximal 300 Punkten/Zoll zur Verfügung.

Die Drucker erkennen zwei Graphik-Protokolle:

- TEKTRONIX-4014 Modus

In diesem Modus, der durch Escape-Sequenzen ein- bzw. ausgeschaltet wird (<ESC>[?38h bzw. <ESC>[?38l), verhält sich der Drucker wie ein Tektronix 4014 Graphik-Terminal. Das Blatt wird dabei immer im Querformat bedruckt, die X-Richtung ist also parallel zur längeren Blattkante, die maximale Zeichenfläche beträgt 25.64 x 19.3 cm.

Wird der Drucker in den Tektronix-Modus geschaltet, wird die aktuelle Seite ausgegeben, wenn sie druckbare Zeichen enthält. Bei der Rückschaltung in den normalen ANSI-Modus wird die Seite nicht automatisch ausgeworfen, so daß nachfolgende Textdaten auf einer Seite mit graphischen Daten gemischt werden können.

- Sixel Graphik

Dieses Graphik-Protokoll, das auch vom VT241 Graphik-Terminal und LA210-Matrixdrucker verstanden wird, dient zur Übertragung von Rasterbildern. Die maximale Zeichenfläche beträgt 28.79 x 20.32 cm, das sind 3400 x 2400 Pixel bei einer Auflösung von 300 Punkten/Zoll.

Die Graphik-Software der Prozeßrechenanlage unterstützt diese neue Funktionalität bereits in vollem Umfang:

### GOSY

Zur Online-Ansteuerung des Laserdruckers steht eine leicht modifizierte Version des Tektronix 4014-Treibers im File APPL\$GOSY:LASLIB.OLB zur Verfügung. Wie bei allen Tektronix-Geräten ist in INITP der Initialisierungscode ICODE = 3 zu wählen, als Filename kann der systemweit definierte logische Name 'LN03' verwendet werden.

Geräteunabhängige Graphik-Files werden durch folgenden Aufruf ausgegeben:

```
PLT filename LAS
```

Soll das Bild im Hochformat ausgegeben werden, kann dies bei der Online-Ausgabe durch die GOSY-Subroutine TURN (14.85, 14.85, 90.) nach der Initialisierung erfolgen. Bei der Ausgabe von geräteunabhängigen Graphik-Files kann die Drehung (und Skalierung) des

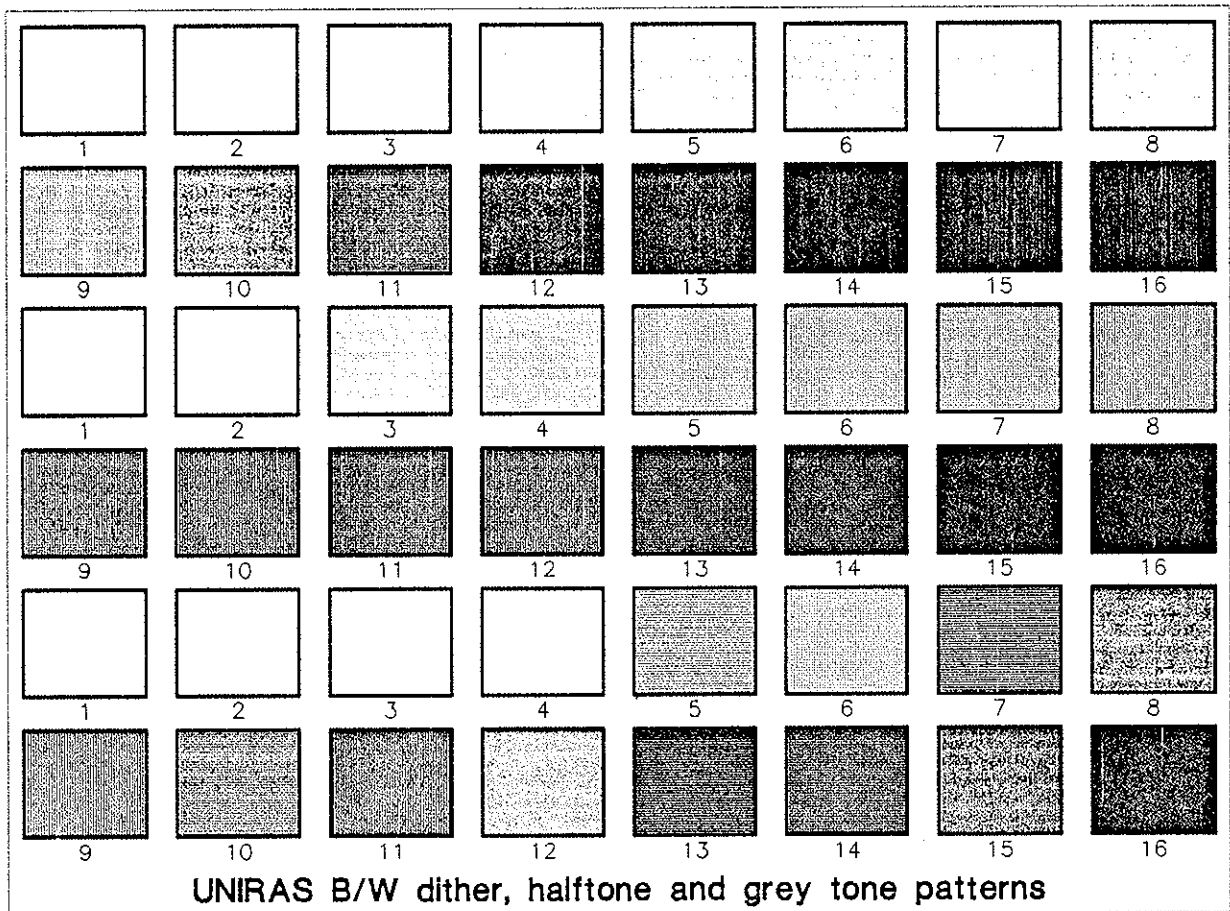
*Ausgabe von Graphik und Text auf dem Laserdrucker*

Bildes durch das File PLT.PAR im aktuellen Default-Directory festgelegt werden (siehe GOSY Benutzerhandbuch, Anhang D).

Text und Graphik können nicht gemeinsam auf einer Seite ausgegeben werden, da beim Abschließen der Graphikausgabe (CALL PLOT (0.,0.,999)) das Blatt immer ausgeworfen wird. Auch bei jedem Aufruf von PFEED im Programm wird eine neue Seite begonnen.

**UNIRAS**

Die Ausgabe von UNIRAS Rasterbildern auf den Laserdrucker, die im Sixel-Graphik Modus erfolgt, wurde bereits im TU-PRA INFO 86/10 beschrieben. Doch erst mit der Aufrüstung der Laserdrucker können nun beliebig komplexe Graphiken ausgegeben werden, was vorher auf Grund der beschränkten Speicherkapazität noch nicht möglich war. Auch dieser Treiber verwendet standardmäßig das Querformat und erlaubt eine maximale Zeichenfläche von 26.8 x 19.47 cm, entsprechend 3262 x 2300 Pixel bei 300 Punkten pro Zoll. Soll die Graphik im Hochformat erzeugt werden, kann durch die UNIRAS-Subroutine GROTA das Bild um 90 Grad gedreht werden.



*Abb.1 : Verschiedene Grauwertstufen*

*Ausgabe von Graphik und Text auf dem Laserdrucker*

Um Text und Graphik auf einem Blatt gemeinsam ausgeben zu können, muß der Text und das vom UNIRAS-Lasertreiber erstellte Sixel-File zu einem File zusammengefügt werden. Die Graphik wird immer von der aktuellen Position beginnend ausgegeben, daher ist die Positionierung des Bildes auf dem Blatt auch vom angegebenen FORM-Typ abhängig. Die maximale Zeichenfläche steht nur bei Verwendung des FORM-Typs LN03\_NOFEED zur Verfügung.

In obiger Abbildung sind als Beispiel mit UNIRAS erzeugte Grauwertstufen dargestellt, die zur Flächenfüllung verwendet werden können.

## Übersicht über die unterstützten Graphik-Geräte

Rudi Sedlaczek

Auf Grund der ständig steigenden Anzahl von graphikfähigen Ausgabegeräten soll im folgenden ein kurzer Überblick gegeben werden, welche Geräte derzeit von den Graphik-Paketen GOSY (Vektorgraphik) und UNIRAS (Raster- und Vektorgraphik) unterstützt werden.

### GOSY

Im Betriebssystem VAX/VMS können derzeit folgende graphische Geräte mit dem Graphischen Output System GOSY verwendet werden:

Beschreibung	Library	ICODE	Treibercode	Device(Default)
Goerz BBC-281 Plotter	281LIB	2	281	PL281
HP 7550 Plotter (HP-GL)	755LIB	2	755	HP7550
Epson FX80 Matrixdrucker	FX8LIB	2	FX8	FX80
Epson FX100 Matrixdrucker	FX1LIB	2	FX1	FX100
Watanabe Digi-Plot Plotter	WATLIB	2	WAT	TT
LN03 PLUS Laserdrucker	LASLIB	3	LAS	LN03
Tektronix 4010 Terminal	TEKLIB	3	TEK	TT
Tektronix 4014 Terminal	T14LIB	3	T14	TT
Tektronix 4025 Terminal	T25LIB	3	T25	TT
VT241 Graphik Terminal	241LIB	4	241	TT
VT125 Graphik Terminal	125LIB	4	125	TT
HP 7221 Plotter	721LIB	5	721	HP7221
Calcomp 960 Plotter	CALLIB	5	CAL	CALCOMP
Goerz 211 Plotter	211LIB	5	211	PL211

Die Treiberbibliotheken und Gerätetreiber sind im Directory APPL\$GOSY: gespeichert. Auf alle Geräte können die Graphikinformationen direkt vom Anwendungsprogramm aus ausgegeben werden. Hier müssen die entsprechenden Libraries beim Linken angegeben werden, wobei bis zu 4 verschiedene Geräte angesprochen werden können, die sich allerdings im Initialisierungs-Parameter ICODE unterscheiden müssen.

*Übersicht über die unterstützten Graphik-Geräte*

Alternativ oder zusätzlich zur direkten Ausgabe kann ein geräteunabhängiges Graphik-Metafile erstellt werden, das mit Hilfe der Steuerprozedur PLT von spezifischen Treiber-routinen in einem zweiten Schritt auf beliebige Ausgabegeräte ausgegeben werden kann (siehe auch GOSY Benutzerhandbuch, Anhang D).

### UNIRAS

Im sogenannten 'Host Raster Mode', bei dem das Rasterbild zuerst im Rechner angelegt wird, bevor es an das Ausgabegerät gesendet werden kann, stehen Treiber für folgende Geräte zur Verfügung:

Gerät	Initialisierungsroutine
Tektronix Inkjet Plotter 4695	G4695
Digital LN03 PLUS Laserdrucker	GLN03
Digital LA210 Matrixdrucker	GLA210A
Mannesmann Tally MT490 Matrixdrucker (Epson kompatibel)	GMT490A
Tektronix 4105	G4105D
Tektronix 4107 (Dither bzw. Intensity)	G4107D G4107I
Tektronix 4109 (Dither bzw. Intensity)	G4109D G4109I
Tektronix 4113 (Dither bzw. Intensity)	G4113D G4113I
Tektronix 4115 (Dither) mit 4, 6 oder 8 Ebenen	G4115D G41154 G41156 G41158

Im interaktiven 'Local Raster Mode', bei dem die Graphik-Primitiven direkt an das Ausgabegerät gesendet werden und auch graphische Eingaben möglich sind, können folgende Geräte angesprochen werden:

Gerät	Initialisierungsroutine
Terminals mit Regis Graphik-Protokoll	LREGIS
Digital VT240 Graphik Terminals	LVT240
Digital VT125 Graphik Terminal	LVT125
Digital Rainbow mit Regis Emulator	LRB240
Tektronix 4105, 4107, 4109, 4111, 4113, 4115 Graphik Terminals	LT41XX
GKS Metafile Output	LGKSM LGKSMD

Im 'Vector Mode' können folgende Geräte angesprochen werden:

Gerät	Initialisierungsroutine
Tektronix 4010 Graphik Terminal	PT4010
Tektronix 4014 Graphik Terminal	PT4014
Calcomp Plotter 960	PCPLOT

(bzw. jeder Plotter, der mit den Calcomp Standard-Routinen PLOT, PLOTS und NEW-PEN angesteuert werden kann).

## Neues von der Calma Anlage

Günter Houdek

### NEUE MENÜFELDER FÜR GDSII

Das NOVA-System mit der GDSII-Software von CALMA, welches hauptsächlich für den Leiterplattenentwurf eingesetzt wird, hat nach acht Jahren intensiver Verwendung die Grenzen der Rentabilität schon lange überschritten, und den Ruhestand wohl verdient. Da die Firma CALMA in Zukunft sowohl Hardware als auch Software nicht mehr warten wird können, ist der Ersatz der NOVA-Anlage durch ein dem neuesten Stand der Technik entsprechendes Leiterplatten- Entwicklungssystem geplant.

Als Zwischenlösung wurden an der ECLIPSE, dem zweiten Rechner des CALMA- Systems neue Menü-Felder für die GDSII-Software erstellt, die denen von GDSII ähnlich sind. Damit wird einerseits der Übergang zum GDSII - Softwarepaket erleichtert, andererseits neuen Benützern ein schneller Einstieg ermöglicht.

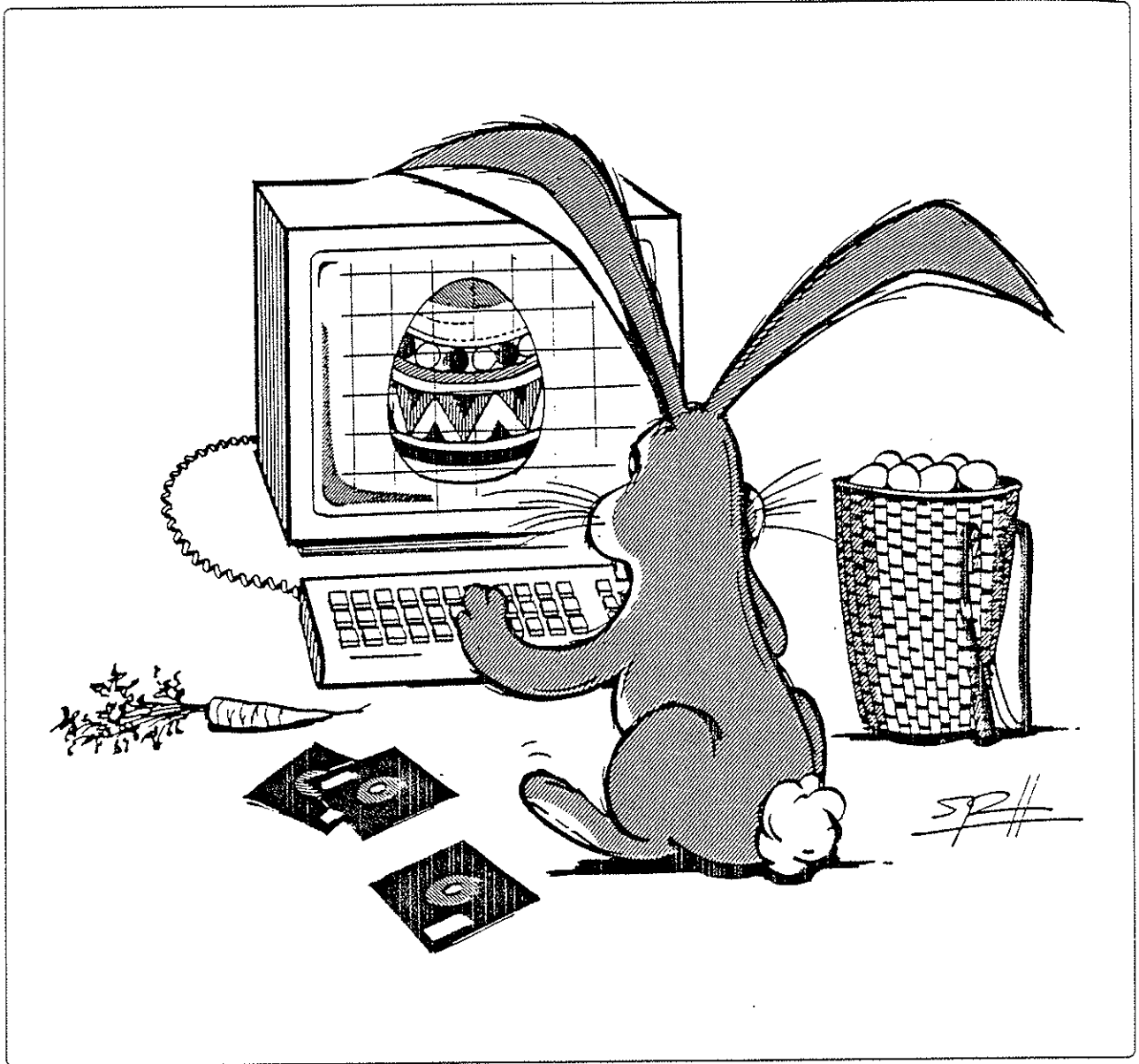
Zusätzlich zu den MENÜS A + B, sind jetzt auch die MENÜS C + D verfügbar, die speziell für die Printplattenerstellung konzipiert sind. Die Bauteilbibliothek (TUW\$PARTS.DB) muß dazu mit dem Befehl 'OPENREF1' od. '2' zur eigenen Bibliothek dazugelinkt werden. Weiters muß der Textfont C\$PARTS mit 'BINDFONTS' zum FONT 1 gebunden werden. Das D-MENÜ zeigt rechts oben einen Button 'SETUP' in weißer Schrift, bei dessen Aufruf eine Initialisierungsroutine aufgerufen wird, die alle GPL-Programme lädt, sowie die externen Variablen definiert. Beim Hochfahren der Station (Farbstation) wird diese Initialisierung automatisch aufgerufen.

Das D-MENÜ ist speziell für die Bauteilaufrufe konzipiert, und das C-MENÜ unterstützt insbesondere das Editieren.

Im C-MENÜ sind auch die Pen- sowie Photoplotterroutinen untergebracht, die eine einfache Ausgabe erlauben. Für die Filmherstellung auf dem Photoplotter wird ab April kein Magtape mehr verwendet werden müssen, da dann die Daten automatisch von der CALMA zur VAX seriell überspielt werden, wobei sich der Benutzer auch nicht mehr um das RENAMEN der Daten (PHCODES.) zu kümmern braucht.

Eine ausführliche Dokumentation wird Anfang April erhältlich sein, mit deren Hilfe es einem neuen Benutzer möglich sein soll, problemspezifische Arbeiten in GDSII nach kürzester Einarbeitungszeit (ca. 2 Stunden) durchzuführen.

Wir hoffen, daß mit diesen neuen MENÜS ein schnelles Erstellen von Layouts für Printplatten (auch für Neueinsteiger in GDSII) ermöglicht wird.



Die Mitarbeiter der Abt. Prozeßrechenanlage  
wünschen allen Benutzern  
recht angenehme Ostertage.



**BERICHTE**  
aus der Arbeit der Institute  
und der Abt. Prozeßrechenanlage

## 3D-Finite-Elemente Preprocessing mit EDAS

F. Schmöllebeck\*

Um mit Hilfe der Finite-Elemente-Methode Berechnungen von mechanischen bzw. elektrotechnischen Problemen durchführen zu können ist es nötig, die zu untersuchenden Teile in geometrisch einfache Körper - die sogenannten Finiten-Elemente (FE) zu zerlegen. Unter Finite-Elemente Preprocessing versteht man das Erzeugen von solchen "FE-Netzen". Als Finite-Elemente eignen sich im dreidimensionalen Fall z.B. Tetraeder, Pentaeder und Hexaeder.

Am Institut für Grundlagen und Theorie der Elektrotechnik - Abteilung Berechnung elektromagnetischer Felder wird derzeit ein Programmsystem zur Berechnung dreidimensionaler elektromagnetischer Felder entwickelt wobei die 3D-FE-Netze mit Hilfe von EDAS erzeugt werden. Es steht bereits ein Software-Paket für 2D-FE-Berechnungen von elektromagnetischen Feldern zur Verfügung bei welchem neben eines am Institut entwickelten Preprocessors auch EDAS verwendet wird.

Speziell bei dreidimensionalen Problemen ist das Zerlegen des Grundgebietes (der zu untersuchenden Teile) extrem zeitaufwendig und unübersichtlich, so daß eine Reihe von Softwarepaketen zur Verfügung steht welche das Unterteilen in FE unterstützen. Die meisten dieser Preprocessing-Pakete sind in FE-Programmsysteme integriert und bieten demzufolge oft nur bescheidene Möglichkeiten zur interaktiven, graphischen Entwicklung von FE-Netzen. Vor allem die Möglichkeiten der Visualisierung und der Anwendung von geometrischen Transformationen sind bei solchen integrierten Softwarepaketen sehr eingeschränkt.

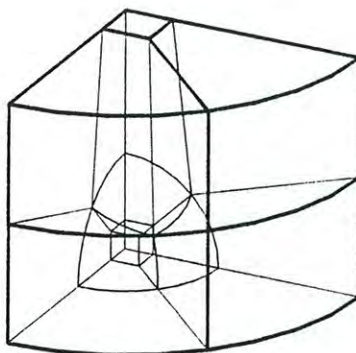
Mit dem an der PRA verfügbaren Preprocessor "EDAS", welcher voll im 3D- Volumen-CAD-System EUCLID eingebettet ist, können praktisch sämtliche Möglichkeiten eines extrem leistungsfähigen CAD-Systems für die Erstellung von FE-Netzen genutzt werden.

### *Unterteilung in Finite-Elemente*

Die zu unterteilende Geometrie wird vorerst in sog. "Makroelemente", das sind den Finiten-Elementen ähnliche, aber wesentlich größere Strukturen unterteilt. Dies kann unter Nutzung sämtlicher EUCLID-Eingabefunktionen wie z.B. graphische Eingabe, graphisches Identifizieren, numerische Eingabe von Punkten in verschiedenen Koordinatensystemen (auch Mischformen möglich) geschehen.

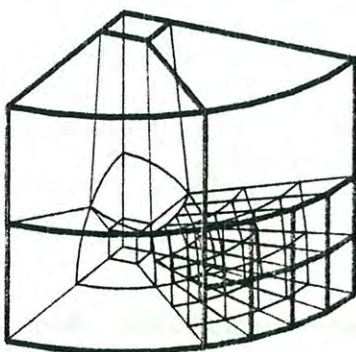
---

\* Dipl. Ing. Friedrich Schmöllebeck ist Assistent am Institut für Grundlagen u. Theorie der Elektrotechnik an der TU-Wien.



*Makrostruktur: Dielektrische Kugel (1/8 dargestellt)  
- umgebender Raum zylinderförmig begrenzt*

Die Makroelemente werden anschließend im interaktiven Dialog in Finite-Elemente unterteilt. Damit besteht die Möglichkeit, etwaige Eingabefehler schon während des Generierens der Finiten-Elemente festzustellen und ebenfalls im menügesteuerten Dialog zu beheben.



*Makrostruktur: Dielektrische Kugel  
- ein Makroelement in Finite-Elemente (Hexaeder) zerlegt*

Zur lokalen Netzverdichtung sind mehrere Arten von Übergangs- Makroelementen verfügbar. Auch die Möglichkeit, die Elementsgröße in jeweils einer Unterteilungsrichtung eines Makroelementes nach einer geometrischen Reihe zu stufen, kann bei der lokalen Netzverdichtung hilfreich sein.

Es steht folgende Palette von zwei und dreidimensionalen Elementen zur Auswahl :

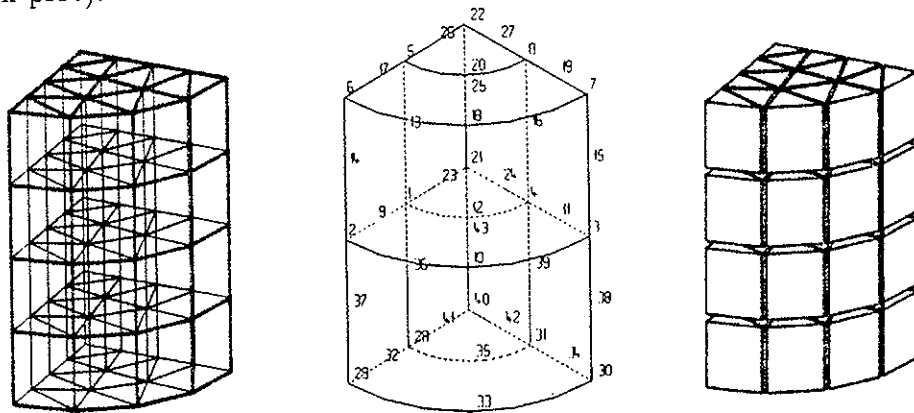
<u>ELEMENTSTYP</u>	<u>KNOTENZAHL</u>
STAB	2,3
DREIECK	3,6
VIERECK	4,8,9
PENTAEDER	6,15,18
TETRAEDER	4,10
HEXAEDER	8,20

Alle in der Tabelle enthaltenen Elemente, die Seitenmittenknoten besitzen können auch gekrümmt werden. Dadurch werden z.B. quadratische isoparametrische Elemente unterstützt.

### Verbindung EDAS - EUCLID

Die Datenstruktur des Modules EDAS ist voll in EUCLID integriert. Gerade dieser Punkt hebt den hier beschriebenen Preprocessor in eine Sonderstellung unter den 3D-Preprocessing Paketen. So sind z.B. sämtliche Visualisierungsmöglichkeiten und Darstellungsarten von EUCLID in EDAS voll wirksam. Auf Wunsch kann aus einem FE-Netz ein vollwertiges EUCLID- Objekt erzeugt werden welches dann verschoben, gespiegelt, verdreht, in jeder beliebigen Ansicht visualisiert und in der Datenbasis von EUCLID gespeichert werden kann (beim Holen von FE-Netzen aus der Datenbasis sind Restriktionen zu beachten). Bool'sche Operationen sind mit FE- Strukturen grundsätzlich nicht erlaubt, da eine eindeutige Zuordnung der Knotenpunkte des Netzes während einer solchen Operation nicht möglich ist.

Für FE-Strukturen ist zusätzlich das geschrumpfte Darstellen der Finiten-Elemente möglich (Shrink plot).



*Zylinderstruktur aus Pentaedern modelliert:  
Darstellung als "Drahtnetz" und "Shrink-Plot"*

Durch spezielle, in der EUCLID-Sprache vom Benutzer geschriebene Prozeduren kann EDAS den verschiedensten Anforderungen angepaßt werden (z.B. interaktive Eingabe von Randbedingungen etc.).

### Bearbeitung von Netzen durch externe Programme

Sollen ein einmal erzeugtes Netz oder Teile davon durch externe Programme bearbeitet werden, stehen dafür Interfacdateien zur Verfügung. Mit Hilfe dieser Files ist es möglich in jeder beliebigen, dem Problem angepaßten Sprache Programme zu entwickeln, welche das generierte Netz nach Belieben modifizieren. Das veränderte Netz kann anschließend von EDAS gelesen, dargestellt und weiterverarbeitet werden. Auch zum Erzeugen von Eingabedateien für selbst entwickelte FE-Programme sind die oben erwähnten Interfacdateien eine wertvolle Hilfe.

### ***Aufbau einer Netzbibliothek***

Einmal entwickelte Strukturen, oder auch nur Teile von schon bestehenden Netzen, können auf externen Dateien gesichert und später auch in anderen Netzen verwendet werden. Es besteht somit die Möglichkeit zum Anlegen von Sammlungen öfter benötigter Teilstrukturen welche im Bedarfsfall nicht nochmals erzeugt werden müssen.

Durch die Möglichkeit solche Netzteile mit Hilfe von EUCLID geometrischen Transformationen zu unterziehen wird eine "Netzbibliothek" auch unabhängig vom Maßstab des aktuellen Netzes.

### ***Randbedingungseingabe***

Da sich die Eingabe von Randbedingungen bei mechanischen und elektrotechnischen Problemen grundsätzlich unterscheidet und an unserem Institut ausschließlich elektrische Probleme behandelt werden, ist es mir hier nicht möglich die Eingabemöglichkeiten von mechanisch orientierten Randbedingungen im Detail zu beschreiben. EDAS beinhaltet standardmäßig die Möglichkeit, Eingabedateien für die FE-Pakete ANSYS und TPS10 (in Zukunft auch NASTRAN) zu erzeugen.

Die Eingabe der für diese Pakete nötigen Randbedingungen und Zusatzinformationen ist ebenfalls graphisch unterstützt und menügesteuert möglich. Für die Implementierung von Eingabemöglichkeiten für andere Arten von Randbedingungen und Zusatzinformationen wie dies z.B. in der Elektrotechnik der Fall ist, bietet sich wie schon oben erwähnt die EUCLID- Sprache an.

### ***Korrekturen an bereits bestehenden Netzen***

In bereits bestehenden Netzen können Makroelemente bzw. Finite-Elemente gelöscht und Knotenpunkte verschoben werden. Eine neue Unterteilung von Makroelementen ist jederzeit möglich. Doppelknoten können zur Entkopplung von Freiheitsgraden (Schlitze in der Mechanik, Unstetigkeit bestimmter Feldgrößen in der Elektrotechnik) definiert werden.

### ***Darstellung von Verschiebungen***

Für mechanische Probleme besteht die Möglichkeit, über eine spezielle Interfacedatei die vom FE-Programm errechneten Verschiebungen der Knotenpunkte einzulesen. Anschließend kann die verformte Struktur mit Hilfe von EDAS dargestellt werden. Mit EUCLID sind dann z.B. Kollisionsanalysen möglich.

## Der neue österreichische EUNET/USENET Backbone- Kommunikationsmöglichkeit mit der ganzen Welt

Walter Kunft

### *Einführung*

USENET ist ein Datennetz, das Computersysteme, die das Betriebssystem UNIX<sup>\*)</sup> verwenden, zusammenschließt. Es erstreckt sich über die ganze Welt, (4). Im Jahre 1985 waren bereits 3200 Knoten offiziell bekannt, davon entfielen 260 auf Europa (Stand März 85, siehe (1)). Inzwischen ist die Zahl der Knoten auf über 5000 angewachsen, in Europa erhöhte sich die Zahl auf über 400 (Stand Jänner 1986). Der europäische Teil von USENET wird EUNET genannt.

USENET hat im Laufe der letzten Jahre eine große Bedeutung im Bereich der Wissenschaft und Forschung erlangt. Fast alle Universitäten, in denen UNIX eingesetzt wird, haben Zugang zu diesem Netz. Sie bilden gleichsam das Rückgrat des Datennetzes und sorgen dafür, daß über dieses Datennetz aktuelle und wissenschaftlich relevante Informationen bezogen werden können (News-Dienst). Im USENET-Verbund sind aber nicht nur Universitäten sondern die Forschungsstätten nahezu aller großen Industrieunternehmen, die UNIX verwenden, inkludiert. Außerdem nehmen die größeren Softwarehäuser an USENET teil.

Die Ausbreitung von USENET geht Hand in Hand mit der Ausbreitung von UNIX, das heute in den verschiedensten Bereichen und auf den verschiedensten Computersystemen aller Größenordnungen eingesetzt wird. Mit dem Betriebssystem UNIX und seinen Dienstprogrammen bekommt der Teilnehmer auch automatisch die Softwareunterstützung mitgeliefert, die er braucht, um an USENET teilzunehmen.

USENET ist ein Nachrichtenvermittlungsnetz, das heißt die Nachrichten werden zusammen mit Adreßinformation vom Quellknoten weggeschickt und von Knoten zu Knoten weitergereicht, bis sie schließlich beim Zielknoten ankommen. Dabei werden stets ganze Nachrichten transportiert. Diese Technik wird auch "Store and Forward Technik" genannt. Die Weitergabe von Briefen, News usw. wird durch die einzelnen Knoten ohne gesetzliche Gewähr durchgeführt. USENET ist ein Netz ohne Netzzentrale, das beliebig wachsen kann, ohne daß das gesamte Netz davon informiert werden muß.

---

<sup>\*)</sup> UNIX ist ein Warenzeichen von AT&T Bell Laboratories

Die im USENET angebotenen Dienste sind:

- Dateitransfer
- Electronic Mail
- News

Die Kommunikationsprogramme von UNIX bieten auch die Möglichkeit, mit entfernten Systemen im Dialogbetrieb zu kommunizieren, diese ist jedoch in der Dienst- Palette von USENET von untergeordneter Bedeutung.

In den letzten Jahren wurden die internationalen Datennetze des Wissenschafts- und Forschungsbereiches in immer stärker werdendem Ausmaß miteinander gekoppelt. Diese Entwicklung hat es ermöglicht, daß USENET-Teilnehmer auch andere Netze über Gateways erreichen können, z.B. ARPANET, CSNET, BITNET, EARN, um nur die wichtigsten zu nennen.

### *Das Konzept von EUNET*

EUNET, der europäische Teil von USENET, wird von der European Unix User Group (EUUG) betrieben. Hauptanliegen der EUUG ist die Förderung der Anwendung und Ausbreitung von UNIX. Der Betrieb des Datennetzes EUNET ist ein Aspekt dieser Zielsetzung. In vielen Ländern Europas wurden in den letzten Jahren nationale Benützervereinigungen gegründet, die die Organisation von EUNET im jeweiligen Land durchführen und selbst wieder Mitglied der EUUG sind.

Für einen EUNET-Anschluß benötigt ein Teilnehmer einen Nachbarn, der bereits angeschlossen ist. Die Struktur von EUNET wird damit weitgehend dezentral bestimmt. Das Topologiekonzept von EUNET sieht drei Kategorien von Knoten vor, (2):

- Backbone Sites (Hauptknoten)
- Secondary Feeder (Zwischenknoten)
- Terminal Sites (Endknoten)

BACKBONE SITE ist diejenige Stelle, welche den Hauptknoten in jedem Lande betreibt. Über ihre Rechner laufen alle Fernverbindungen, sie geben technische Unterstützung und warten und versenden die Netz-Software (siehe auch Punkt 3.7). Ein spezieller Hauptknoten wickelt den Verkehr zwischen Europa und Übersee ab, nämlich der Backbone mcxav in Amsterdam. Für Deutschland ist die Universität Dortmund (Knoten unido) Backbone Site, für Österreich hat die Prozeßrechenabteilung des EDV-Zentrums der TU-Wien (PRA) diese Funktion übernommen. Die Unterscheidung zwischen SECONDARY FEEDER und Backbone Site ist fließend. Feeder sind für das reibungslose Weiterleiten von News und Mails von und zu ihren angeschlossenen Terminal Sites verantwortlich. TERMINAL SITES schließlich treten als reine Konsumenten im EUNET auf.

Die Verbindung zwischen Knoten kann über Standleitungen oder über öffentliche Wähl-Netze erfolgen. Hier ist es vor allem das öffentliche Telefonnetz, das wegen seiner Flächen-deckung häufig zur Kopplung zwischen Knoten herangezogen wird. In letzter Zeit wurden aber - vor allem im internationalen Bereich, also zur Kopplung von EUNET-Backbones - in immer stärkerem Ausmaß öffentliche Datenpaketvermittlungsnetze eingesetzt. Die Verwendung des Datenpaketvermittlungsnetzes erfolgt über einen PAD (Packet-Assembly-Disassembly-Unit), da das gesamte Kommunikationskonzept in UNIX an der Kommunikation von asynchronen Terminals mit Wirtsrechnern orientiert ist.

*Der neue österreichische EUNET/USENET Backbone*

Um den Teilnehmern die Bürde der Definition der Wege von Nachrichten abzunehmen - dies ist vor allem beim Verkehr mit anderen Datennetzen wichtig - bieten die Backbones automatische Wegesuche als zusätzliche Dienstleistung an. Sie unterhalten zu diesem Zweck Verzeichnisse aller Installationen und ihrer Verbindungen untereinander.

Es besteht keine Verpflichtung, den internationalen Verkehr über Backbones abzuwickeln. Im folgenden sind kurz die Vor- und Nachteile des Backbone-Konzeptes dargestellt, (3):

Nachteile:

- Electronic Mail wird teurer, wenn sie über die Backbones fließt. Ein direkter Austausch zwischen den beteiligten Knoten ist schneller und billiger.
- Das interaktive Arbeiten im USENET wird von den Backbones nicht unterstützt.

Vorteile:

- Backbones bieten ein automatisches Routing an, der Benutzer braucht sich nicht mehr um den günstigsten Weg zu kümmern, um zum gewünschten Zielknoten zu gelangen.
- Dem Benutzer wird es erspart, bei jedem gewünschten Zielknoten einen Account zu beantragen.
- Inkompatibilitäten können von Backbones überbrückt werden. So kann z.B. ein Teilnehmer, der nur einen Zugang zum Telefonnetz besitzt, über den Backbone mit einem Teilnehmer, der nur einen Zugang zum DATEX-P besitzt, kommunizieren.
- Backbones übernehmen Gatewayfunktionen zu anderen Datennetzen
- Beim News-Dienst müssen riesige Datenmengen vor allem aus den USA nach Europa transportiert werden. Hier ist es wesentlich billiger, die Information einmal in ein Land zu holen und dort zu verteilen, als jeden dieselbe Information nur für sich über den Ozean holen zu lassen.
- Backbone Sites bieten Unterstützung bei technischen und organisatorischen Problemen.

Die Vorteile des Backbone-Konzeptes rechtfertigen die Anstrengungen, die zur Realisierung eines EUNET/USENET-Backbones für Österreich unternommen wurden.

### *Der neue österreichische EUNET/USENET - Hauptknoten*

#### **Vorgeschichte**

Die ersten Aktivitäten, UNIX-Rechner in Österreich an EUNET/USENET anzuschließen, wurden von Herrn Dipl. Ing. Schwabl vom Institut für Praktische Informatik der Technischen Universität Wien unternommen (1). Die Realisierung eines österreichischen Hauptknotens erfolgte in mehreren Stufen:

- Realisierung eines provisorischen Backbones im Rahmen eines vom Bundesministerium für Wissenschaft und Forschung geförderten Forschungsprojektes. Als provisorischer Backbone wurde ein Rechner des Institutes für Praktische Informatik mitverwendet, seine Integration in EUNET/USENET wurde gemeinsam von Mitarbeitern des Institutes für Praktische Informatik, des Interuniversitären EDV Zentrums und der Abt. Prozeßrechenanlage der TU-Wien durchgeführt (5).

#### *Der neue österreichische EUNET/USENET Backbone*



- Am 1. August 1986 konnte dann der neue offizielle EUNET/USENET- Hauptknoten, namens "tuvie", in Betrieb genommen werden, dessen Betriebsführung die Abt. Prozeßrechenanlage durchführt.

### Konfiguration und Erreichbarkeit des neuen Hauptknotens

Der österreichische Hauptknoten *tuvie* besteht aus einer Micro-VAX-II, die mit dem Betriebssystem ULTRIX 32m ausgestattet ist (siehe Abb. 1). *Tuvie* ist über Ethernet mit einer VAX-11/780 der Abt. Prozeßrechenanlage verbunden und durch DECnet-ULTRIX in das Universitätsnetz Austria (UNA) integriert. Als Kommunikationssoftware wird die UUCP-Software der EUUG eingesetzt, das automatische Routing leitet das Programm *sendmail* von ULTRIX 32m ein.

*Tuvie* ist wie folgt erreichbar:

- Zugang über Telefon: (0222) 65 87 11 bis 13 (CCITT V.21, 300 Baud, originate mode), 65 87 14, 65 87 15 (CCITT V.21, V.25)
- Zugang über DATEX-P: Von Österreich: 2422109401, aus dem Ausland: 02322422109401.
- Zugang von der TU Wien aus über PACX, Geschwindigkeit: 300, 1200 oder 9600 Bd.
- Zugang über UNA: '\$ set host etuvie' und login im Backbone.

Die angegebenen Telefon- und DATEX - P - Nummern führen zum Vermittlungssystem PACX. Nach dem Verbindungsaufbau bzw. bei Benutzung von Standleitungen zum PACX muß noch der Anschluß zu "tuvie" hergestellt werden. Dies geschieht durch Eingabe des Klassennamens *tupra* auf die Eingabeaufforderung "enter class" von PACX. Nach Start der Klasse *tupra* ("class *tupra* start") kann durch Carriage-Return die Login-Aufforderung durch "tuvie" erreicht werden. Zu beachten ist, daß "tuvie" mit automatischer Erkennung der Übertragungsgeschwindigkeit arbeitet. Aus diesem Grund ist es notwendig, nach der ersten Login-Aufforderung durch eine neuerliche Eingabe von Carriage-Return die Erkennung der Baud-Rate zuverlässig zu ermöglichen. Es erfolgt darauf eine erneute Login-Aufforderung, worauf normal mit der Login-Prozedur fortgefahren werden kann.

*Tuvie* kann auch über das Universitätsnetz Austria (UNA) erreicht werden. UNA ist ein DECnet-Verbund über DATEX-P, in den Universitätsinstitutionen aus nahezu allen Universitäten Österreichs eingebunden sind. Eine Verbindungsmöglichkeit von UNA zu *tuvie* eröffnet daher fast allen Universitätsbereichen die Dienste von EUNET/USENET. Der Implementierung der Gatewayfunktion zu UNA in *tuvie* kommt noch eine zusätzliche Bedeutung zu. UNA ist zunächst ein geschlossener Verbund, in dem DECnet verwendet wird. Dieser Verbund wird aber schrittweise geöffnet werden, so daß schließlich ein offener Verbund nach den Regeln der Open Systems Interconnection entstehen wird, wie es der Zielsetzung des Projektes zur Schaffung eines offenen Akademischen Computernetzes (ACONET) entspricht. Der EUNET- USENET Backbone wird dann eine Gatewayfunktion des zukünftigen ACONET zum USENET ausüben.

### Die Dienstleistung von *tuvie*

In diesem Kapitel wird ein kurzer "Überblick" über die Dienstleistung gegeben, die der österreichische EUNET/USENET-Backbone *tuvie* bietet.

Im Normalfall können die Dienste von EUNET/USENET dadurch konsumiert werden, daß eines oder mehrere UNIX-Systeme einer Universitätsinstitution oder einer Firma an den

*Der neue österreichische EUNET/USENET Backbone*

Hauptknoten angeschlossen wird. Über *tuvie* können dann die in den Punkten 4.1 bis 4.4 ausgeführten Dienste in Anspruch genommen werden. Im EUNET gilt die Konvention, daß Verbindungen zwischen den Hauptknoten und ihren Benützern immer von letzteren aufgebaut werden.

### Dateitransfer

Der Dienst des Dateitransfers wird vom Programm UUCP (UNIX to UNIX COPY) im lokalen System erbracht. Dieses Programm führt jedoch noch keine Datenübertragung durch, sondern ist für das Spooling der Benutzeranforderungen für Dateitransfers zum Backbone und über diesen zum gewünschten Zielsystem verantwortlich. Der eigentliche Transfer wird durch den "Dämon" (UNIX-Bezeichnung für einen Programmprozeß) UUCICO durchgeführt.

Wenn alle Dateien vom lokalen Endknoten zum Backbone übertragen worden sind, beginnt UUCICO des Hauptknotens, die Spooling-Directories im Hauptknoten zu durchsuchen, ob Aufträge für Übertragungen zum lokalen System vorliegen. Wenn dies der Fall ist, werden die Dateien zum lokalen System übertragen. Erst wenn in keiner Richtung weitere Dateien zu übermitteln sind, wird die Verbindung zwischen dem Endknoten und dem Hauptknoten wieder abgebaut.

### Electronic Mail

Der Aufruf zum Absenden einer Mail an einen Benutzer in einem entfernten System, das auch in einem anderen Datennetz integriert sein kann, hat folgenden Aufbau:

```
$mail [options] tuvie!person@host[.domain]
```

*Tuvie* unterstützt, wie jeder EUNET-Backbone automatisches Routing. Der Benutzer braucht nach dem Namen des Backbone nur Angaben über das Zielnetz, das System im Zielnetz und den Benutzer im Zielsystem machen. Derzeit gibt es 24 Datennetze, die über EUNET durch entsprechende Spezifikationen des Feldes "domain" erreicht werden können, die bekanntesten davon sind: ARPANET, CSNET, BITNET und EARN. Wird .domain weggelassen wird automatisch das Zielnetz EUNET/USENET angenommen. Die Universität Linz, die an EARN angeschlossen ist und den Nodenamen "aearn" hat, kann z.B. mit folgender Adreßangabe erreicht werden:

```
tuvie!user.in.Linz@aearn.earn
```

### News

Der News-Dienst bietet dem Benutzer die Möglichkeit, Informationen über bestimmte Sachgebiete oder über elektronisch geführte Diskussionen, die in sogenannten "News-groups" angeboten werden, zu beziehen.

Im USENET gibt es zwei Kategorien von News:

- News, die nur in gewissen geographischen Bereichen oder nur innerhalb bestimmter Organisationen zur Verfügung stehen und
- News, die netzweit angeboten werden.

Die Einteilung der News in diese beiden Kategorien erfolgt nach Maßgabe des Interessentenkreises. Die netzweit verfügbaren News werden wieder in zwei Gruppen eingeteilt:

- net-groups werden nach dem Konzept des "Schwarzen Brettes" im Netz verteilt,
- mod-groups (moderated groups) behandeln moderierte Themen und werden nach eingeschränkten Mail-Listen verteilt.

*Der neue österreichische EUNET/USENET Backbone*

Das Programmsystem NEWS ermöglicht es dem Benutzer, den News-Dienst in Anspruch zu nehmen, das heißt Nachrichten einer gewünschten Newsgroup zu lesen oder eigene Informationen den anderen Bezieher der entsprechenden Newsgroup zu senden.

**Benützerverzeichnisse**

Für die Wahl des günstigsten Weges zwischen zwei Knoten, die in den Backbones beim Automatischen Routing getroffen wird, ist neben der Kenntnis aller vorhandenen Knoten und deren Verbindungen untereinander vor allem auch interessant, wann Nachrichten zu welchem Nachbarknoten weitergegeben werden. Diese Informationen werden in eigenen Verzeichnissen (maps) festgehalten und in den Backbones laufend aktualisiert. Für jedes im USENET bekannte System gibt es einen Eintrag, der vom Benutzer unter Verwendung des Directory-Dienstes "netdir" vom Hauptknoten abgerufen werden kann.

**Interaktive Benützung von tuvie**

Da es viele Benutzer gibt, die zwar die Dienste von EUNET/USENET in Anspruch nehmen wollen, selbst aber kein UNIX-System zur Verfügung oder in Reichweite haben, haben wir noch eine weitere Möglichkeit geschaffen, die Dienste von tuvie zu benützen, nämlich die Benützung von tuvie im Dialogbetrieb. Der Benutzer kann tuvie über Telefon, DATEX-P (X.3,X.28,X.29), PACX, TUNET oder auch über das Universitätsnetz Austria anwählen und die Login-Prozedur ausführen. Hierauf kann er sämtliche in den Punkten 4.1 bis 4.4 angeführten Dienste von tuvie direkt in Anspruch nehmen (in der Zieladressespezifikation kann dann tuvie! weggelassen werden).

**Gebühren**

Nach ersten Kalkulationen wurde für den österreichischen Backbone folgende Vergebüh- rung eingeführt (Stand März 1987):

- Grundgebühr:

	Universitäten	kommerzielle Anwender
	KEINE*	KEINE*

\*) Grundgebühren entfallen zunächst noch, um so dem Benutzer Unzulänglichkeiten bei der Betriebseinführung abzugelten.

- Verkehrsgebühren (Gebühr in öS/kByte):

Bereich	Universitäten	kommerzielle Anwender
Österreich	0.-	0.-
Europa	1.30	1.30
Übersee	7.60	7.60

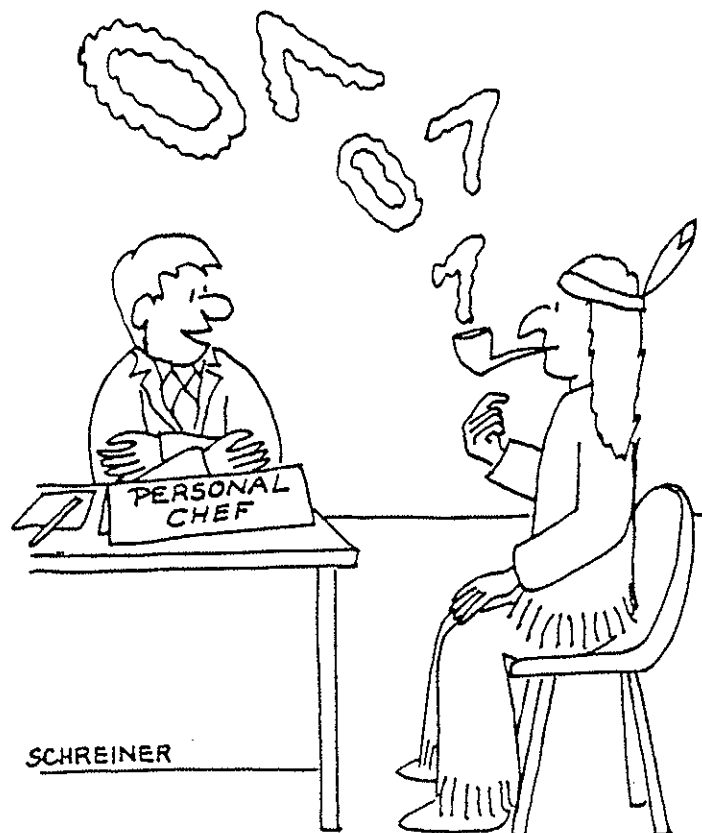
- Gebühren für die interaktive Benützung des Backbones (Gebühr in öS/Login-Stun- de):

	Universitäten	kommerzielle Anwender
	0,-	115,-**

\*\* ) Zusätzlich zur Verkehrsgebühr

**LITERATUR**

- (1) Schwabl W., Institut für Praktische Informatik, Abt. Softwaretechnologie und Echtzeitsysteme der Technischen Universität Wien: *UNIX Netzwerke*, MARS Report Nr. 6/85, Wien, April 1985.
- (2) Hebgen M., D. Karrenberg, M. Rotert, W. Zorn: *Neueste Netz-Nachrichten*, Sonderausgabe zur GI-NTG Tagung "Kommunikation in Verteilten Systemen", 11.-15.3.1985, Universität Karlsruhe, Hrsg.: Universität Dortmund, Informatikrechner - Betriebsgruppe, Universität Heidelberg, Rechenzentrum und Universität Karlsruhe, Informatik-Rechnerabteilung, März 1985.
- (3) Schwabl W.: EUNET bzw. USENET in: *UGA Nachrichtenblatt*, Nummer 1, Juli 1985, S. 8-14, Hrsg.: Unix User Group Austria, Wien, Juli 1985.
- (4) Emerson S. L.: *USENET - A Bulletin Board for UNIX Users* in: *BYTE*, S 219 ff., Oktober 1983.
- (5) Stimmer H., J. Beiglböck, P. Berger, W. Kunft, L. Mayerhofer, M. Paul, F. Plank, W. Schwabl: *USENET-Knoten, Stufe 0*, Endbericht über die Projektstufe 0 des Forschungsvorhabens: Realisierung eines nationalen Knotenrechners für USENET und dessen Integration in das experimentelle ACONET, Wien, März 1986.



... SIE SIND ALSO EXPERTE FÜR  
DATENFERNÜBERTRAGUNG ....

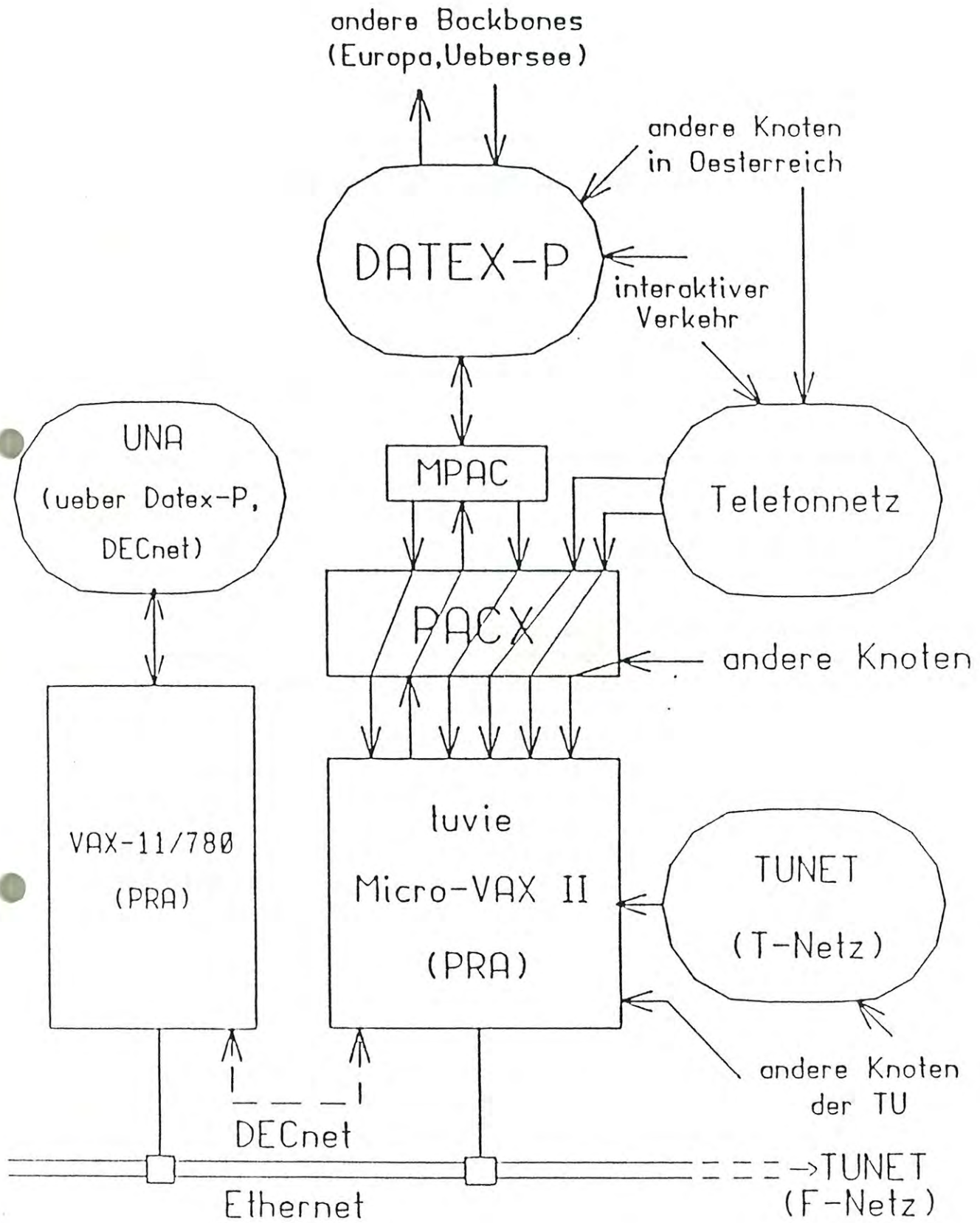


Abb.1: Konfiguration und Erreichbarkeit des österreichischen Backbones

Der neue österreichische EUNET/USENET Backbone

## A Computernetwork with DECNET Connecting Austrian Universities via DATEX-P<sup>\*)</sup>

Walter Kunft, Christian Panigl, Wilfried Wöber

### *Introduction*

Five years ago the Austrian Federal Ministry of Science and Research decided to establish an Academic Computernetwork (ACONET) in Austria which should use DATEX-P and which should be an open network according to the rules of ISO's Open Systems Interconnection.

In the years 1982 and 1983 an experimental interconnection was established which connected different systems of universities in Graz, Linz and Vienna. The work was done within the framework of three research projects supported by the Ministry of Science and Research. We implemented the OSI transport protocol class 0 on top of the X.25-services of the Austrian packetswitching data network DATEX-P to connect systems of different manufacturers.

The experimental interconnection gave us experience with the services and protocols of OSI and their implementation. But what the institutes wanted was a reliable network connecting all universities and providing stable, extensive and comfortable services. Of course there was a gap between such a network and our experimental interconnection. Therefore we adopted another approach to fulfill the requirements of the institutes. First we decided to create a manufacturer-dependent network providing all requested services with the necessary reliability. As a prerequisite the manufacturer concerned must have announced officially that he would introduce the OSI-rules in his communication architecture as soon as possible. In this case the manufacturer-dependent network will change to an open interconnection as soon as the manufacturer has implemented the OSI-protocols. This new approach was the beginning of the University Network Austria (UNA) in 1985 as described in this contribution.

### *What is UNA?*

UNA is an interconnection of Digital-computers of institutions of nearly all Austrian universities. The network has been established to provide a tool for our university-institutions to solve their wide-area communicationproblems. The carrier-network of UNA is DATEX - P, the applied software is the Packetnet System Interface Software (PSI) and DECnet. That means UNA is in its first realization-step a manufacturer-dependent, "closed" network which connects only DEC-systems.

---

<sup>\*)</sup> Manuskript eines Vortrages, erschienen in : 1986 DECUS EUROPE SYMPOSIUM PROCEEDINGS, Hamburg, Sept.22-26, 1986, S.323-329

The establishment of UNA was possible only because Digital Equipment Corporation sponsored the project by supplying communication-hardware and -software free of charge.

UNA will be established in three steps:

1. Initial interconnection of 15 university- and research-institutions. This step has nearly been completed today.
2. Expansion of the interconnection and access to international data networks. Access to EUNET/USENET is already possible.
3. Enlargement to an OSI-network and integration in ACONET. This is a very important step which will be realized if Digital's DECnet/OSI is available.

Institutions of the following universities or research-centres are participating in the first step of UNA:

Austrian Research Centre in Seibersdorf,  
University of Leoben,  
Technical University of Graz,  
University of Graz,  
Research Society Joanneum in Graz,  
University of Klagenfurt,  
University of Innsbruck,  
University of Salzburg,  
University of Linz,  
Institute of Astronomy in Vienna,  
University of Vienna,  
Technical University of Vienna,  
University of Veterinary Medicine of Vienna,  
Institute of Nuclear Research in Vienna and  
Academy of Sciences in Vienna.

### *Which services does UNA provide?*

The DECnet-Software, which is used in UNA, provides a powerful tool to universities to solve their local communication problems. As mentioned above only DEC-systems can be integrated in UNA currently.

To cover needs to exchange information between different universities or research centres, i.e. to perform wide-area communications, another SW-package is available in UNA, namely the Packetnet System Interface (PSI). PSI serves to connect VMS- or RSX-systems (and TOPS-20-systems but this operating system is not used in UNA today) to a public or private packetswitching data network (PSDN). PSI handles all procedures defined in the CCITT recommendation X.25 and offers a service which is practically equivalent to the service of the network layer in ISO's open systems interconnection (OSI). Additionally PSI offers the service of a Packet-Assembly-Disassembly unit (PAD) according to the CCITT recommendations X.3, X.28 and X.29 (more details found in paragraph 3.2.2). This service can be used for access to international data bases or to offer information to other users. It is important to note that the PSI service is manufacturer-independent, that means it allows communication with any manufacturer's system which is capable to handle the CCITT recommendations mentioned above.

PSI plus DECnet can be used to expand the local communication services of DECnet to the wide area via DATEX-P. This is the most important service UNA provides.

### *A Computernetwork with DECNET*

Another important service urgently requested by our university-institutions was the possibility of access to international data networks, especially EARN / BITNET, EUNET / USENET, DFN and so on. Access to EUNET / USENET from UNA is already possible, connections to other networks will take place in the future.

### *Communications-services provided by DECnet for a local region*

DECnet supports nearly all possible network-configurations which connect DEC-computers of all types and IBM-PCs. It is suitable for point to point connections and for multipoint-connections. You can use DECnet to operate a local area network of the ETHERNET-type. The following overview gives a very short description of the most important DECnet-services.

#### **Task-to-task communication**

This is the fundamental DECnet-service. It allows the user to establish a logical link between tasks in different systems and to exchange information between them via this logical link. Decnet supports two kinds of task-to-task communication:

- transparent communication: The other task is handled like a data file. Logical links are established and disconnected with OPEN and CLOSE calls, informations are exchanged using READ and WRITE commands.
- nontransparent communication: This kind of communication is more complicated. Additionally to the transparent communication it provides the means to react on asynchronous events by using the mailbox-mechanism. Incoming calls can be rejected and interrupt- and status-messages can be exchanged in parallel to the normal data flow.

#### **Remote file access using DCL**

Local file access is expanded to file manipulations in remote systems by the introduction of node-names and access control information in file-specifiers. Many DCL commands therefore cause file operations in other systems. Here are three following examples:

```
$ copy test.dat remsys"user passwd":test1.dat
```

causes the file test.dat to be transferred from the current directory in the local system to the remote system remsys and to be stored in the home-directory of "user" with the name test1.dat. The access control information "user passwd" enables DECnet to perform a login in the remote system.

```
$ submit/remote nodex::comproc.com
```

causes the execution of a commandfile comproc.com in the remote system nodex.

```
$ dir etuvie"user passwd":"/"
```

lists the names of all files which are contained in the root-directory of the node "etuvie".

#### **Remote file access using RMS**

Decnet provides file- record- and random-access to remote RMS-files and data bases. To do this RMS-calls can be used with file specifications which contain the name of a remote node.



**Remote system support**

Decnet contains facilities to load operating systems or tasks down to small remote systems without disks. This support of small remote systems is not so important for UNA.

**Support of remote PCs**

This DECnet-service is very important in UNA because it enables resources of a larger system in a region, e.g. disks or a printer to be used by PCs. IBM-PCs or compatible systems can be equipped with DECnet/DOS and connected to a VAX with VMS and DECnet/VMS, for instance, by an ETHERNET.

**Network Command Terminal**

The DCL-command:

```
$ set host remsys
```

connects the terminal concerned logically to the remote system "remsys". All commands entered at the local terminal, for instance, to perform a login-procedure, are executed in the remote system.

**Electronic Mail**

The mail-utility is expanded by DECnet so that E-mail can be sent to users in remote systems. For example:

```
MAIL> send
To: remsys::wilfried
Subj.: ...
.....
```

can be used to send electronic mail to user wilfried at the remote system "remsys".

**Interactive information exchange**

The phone-utility is also expanded by DECnet and allows the user to exchange information interactively in different systems.

**Remote network management**

A system manager can issue DECnet management commands for remote systems at his local terminal. These commands are transferred to the remote system concerned and executed there. This service is very important because it allows a central network management for the whole region.

***Communications-services provided by PSI***

The software-packages PSI and PSI-Access can connect the systems of a local region to a PSDN, e.g. DATEX-P. PSI must be used in a system which is directly connected to a network node of the PSDN. PSI-Access can be used in other systems of the same region to make use of the services of the PSDN though they are not directly connected to the PSDN. The PSI-access-systems communicate with the region's PSI-system connected to the PSDN by using DECnet. DECnet transforms the service from the PSI-host of the region to all PSI-Access-hosts.

***A Computernetwork with DECNET***

PSI supports permanent and switched virtual circuits according to CCITT X.25. Furthermore it provides a Packet-Assembly-Disassembly (PAD) service according to the CCITT recommendations X.3, X.28 and X.29. The following paragraphs give a short survey about the PSI-services.

#### **Task-to-task communication**

PSI provides a service which is nearly equivalent to the network service defined by ISO's Open Systems Interconnection. This service can be used by application-tasks to exchange informations not only with application-tasks in DEC-computers but also with application programs in any other manufacturer's system which communicates according to the rules of X.25.

#### **PAD-Service**

The command: `$ set host/x29 dte.number`  
connects the terminal of the local PSI- or PSI-Access-host to a remote system with the DATEX-P number "dte.number". The local host functions as a software- PAD and establishes a X.29 connection to the remote system. Because of the fact that X.29 is an international, manufacturer-independent standard, the target system may be any manufacturer's product which supports X.29. The user-interface of the PSI-PAD-service is analogous to CCITT X.28 but more convenient.

The PAD-service of PSI also allows a remote terminal connected to a real or a software-PAD to establish an X.29 connection to the local host. Therefore it provides the means to offer access to a local information-system.

#### **PSI-Mail**

Electronic mail can be sent via DATEX-P to a user in a remote system by using PSI. For example the following commands to the mail-utility:

```
MAIL> send
To: psi%dte_number::username
Subj.: .....
.....
```

cause the entered mail to be transferred to PSI (or PSI-Access) and sent to the user "username" in the remote system with the DATEX-P-number "dte.number".

#### ***Expansion of the local DECnet-services to a wider area by PSI***

PSI provides the means to expand the local communication services of DECnet to a wider area via DATEX-P. This is done by using the technique of the so-called "Data Link Mapping" in DECnet. There are special DECnet-management commands which cause an X.25-connection to be established between the calling and the called system with this connection then overlaid by a logical link of DECnet. Such a combination is called a Data-Link-Mapping-circuit (DLM-circuit).

The following example shows the DECnet-management commands which must be issued in NODEA (calling system) and NODEB (called system) to establish a DLM-circuit between them:

*A Computernetwork with DECNET*

NODEA:

NODEB:

```

NCP> set circ x25_outnodeb - NCP> set circ x25_in -
  _ type x25 -                _ type x25 -
  _ owner executor -          _ owner executor -
  _ number 2622101101 -       _ usage incoming -
  _ usage outgoing -         _ state on
  _ state on

```

At NODEB which has the DATEX-P number 26221011 an incoming circuit must be specified. In this example no `dte_number` is specified in the NCP-command at NODEB. This means that calls from all other dtes are accepted.

At NODEA which is the caller an outgoing circuit and the destination address (in this example the address of NODEB and a subaddress 01) must be specified. When the outgoing circuit is set to "state on" a DLM-circuit will be established between NODEA and NODEB. NODEB becomes reachable for NODEA and vice versa and all local DECnet-services can be used between the two regions via DATEX-P.

In the current version of PSI and DECnet an operator intervention is required to establish a DLM-circuit. Doubtless this is a disadvantage for the user but it makes network management and charging much easier.

### *Access from UNA to EUNET/USENET*

As mentioned before the information exchange between UNA-hosts and EUNET/USENET is already possible. Our department of the EDP-Centre operates the Austrian EUNET backbone which has the name "tuvie". Tuvie is a Micro-VAX II computer with the operating system ULTRIX 32m and DECnet/ULTRIX. Tuvie is also integrated in UNA (with the name "etuvie") and performs the gateway-functions between the two networks.

For the time being we have adopted a simple approach to offer EUNET/USENET- services to UNA-subscribers. All UNA-participants who want access to EUNET get a user-account in tuvie. They perform a login with the DECnet command "set host etuvie" and can then utilize all EUNET-services by using the EUNET- backbone interactively. Mail to be sent to EUNET can be prepared in the UNA- system concerned, transferred to the EUNET-backbone with DECnet and sent with the mail-utility of tuvie. Mail from EUNET or news can be read interactively and, if necessary, transferred to the UNA-system by DECnet. This approach makes the solution of accounting- and charging-problems between the two networks much easier.

The future solution will be that UNA is to be registered as a separate domain within EUNET and mail will be routed directly between the two networks.

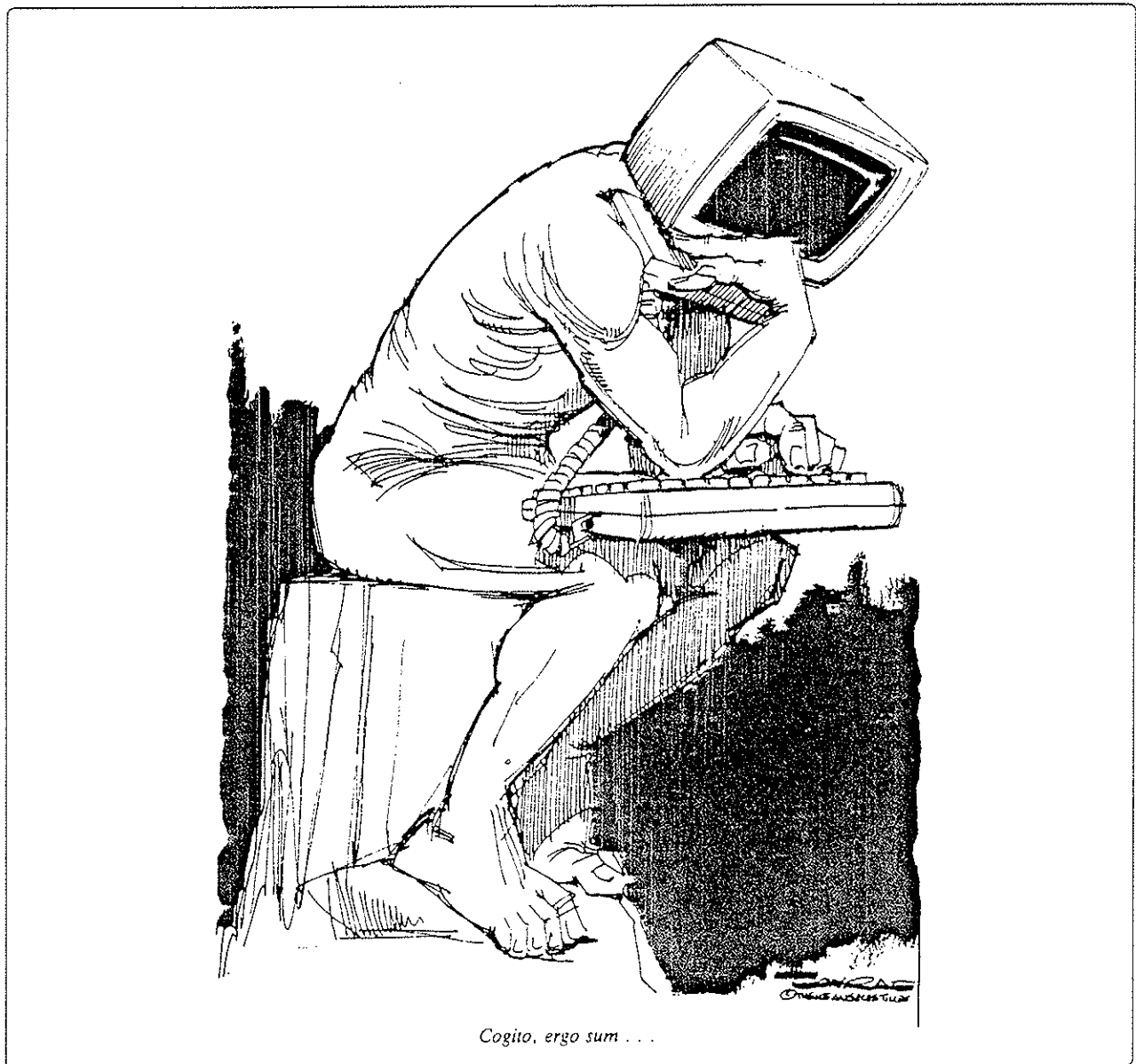
### *Future plans concerning UNA*

In the future the present UNA-interconnection will be expanded. Additional services, for instance access to EARN/BITNET, will be developed and offered to the participants. Great efforts will be necessary to organize and administrate UNA-applications together with the university- or research-institutions concerned. Existing central network services such as directory services or central user's guides will be expanded.

### *A Computernetwork with DECNET*

The most important future development will be the enlargement of UNA to an open systems interconnection. There are several steps planned in this direction:

- Contract with DFN to take over DFN-software which allows communication with non-DEC-systems
- Test-installation of DEC's X.400 gateway-software at our EDP-centre. The X.400 gateway uses OSI-protocols in the layers 4 and 5. DEC offers the products VOTS (VAX OSI-Transport Service) and OSAK (OSI Applications Kernel) to handle the protocols of these layers.
- Usage of DECnet/OSI as soon as it will be available. This step will make UNA to an open network and will integrate it in the future Academic Computer Network (ACONET) in Austria.



## Artificial Intelligence in Software Engineering

Antonin Sprinzl

Der Begriff Artificial Intelligence (AI) ist, wie einschlägiges Studium gegenwärtiger Veröffentlichungen zeigt, zu einem Lieblingsbegriff vieler geworden. Das Spektrum der Wortmeldungen ist sehr breit gefächert. Den interdisziplinär geprägten Ideenmarkt der AI frequentieren heute Informatiker, Psychologen, Linguisten, Soziologen, Philosophen und andere, unter denen hinsichtlich des die Gegenwart kennzeichnenden Phänomens **Komplexität** ein reger Kommunikationsaustausch stattfindet. Einerseits hört man vom Bestreben, Maschinen konstruieren zu wollen, die die menschliche Intelligenz, wenn nicht gleich übertreffen, so doch weitestgehend nachahmen sollen. Andererseits läßt man sich von *weltlicheren* Motiven, vor allem finanzieller Art, sehr stark motivieren ... und baut Expertensysteme, Robotersysteme, natürlichsprachige Systeme (NLU's) bzw. sonstige sog. AI-Systeme.

Ein gewisses Mißtrauen sowie zögernde Haltung gegenüber dieser neuen, sich seit längerem im Formungsstadium befindlichen Disziplin seitens mancher scheint daher aufgrund vorläufig ungenügend präzisierter Zielvorstellungen zum Teil verständlich. Nichtsdestotrotz, die Entwicklung auf dem AI-Gebiet bietet aber schon heute manche *Zwischenprodukte*, die von einem sehr hohen praktischen Nutzen sein können und daher eine erhöhte Beachtung wohl verdienen. Eines davon sind zum Beispiel **exploratory program construction environments** (programming environments, Programmierumgebungen, Programm-Entwicklungsumgebungen,...). Diese können als qualitative Ergänzung (Alternative ?) zu den bekannten Verfahren des Software Engineering (SE) zur Bewältigung komplexer Software-Projekte herangezogen werden. Es sei angemerkt, daß das Thema geeigneter SE-Methodik in den letzten Jahren an Aktualität beträchtlich zugenommen hat und zum Gegenstand intensiver Forschungsbestrebungen gehört.

### *Grundzüge einer Projektentwicklung, klassisches Vorgehen*

Dem klassischen Software Engineering liegt der Ideenansatz zugrunde, von klar umrissenen Vorstellungen und Vorgaben auszugehen. Zu Beginn steht die Durchführung einer umfassenden, erschöpfenden Projektanalyse, um anschließend über die Stufen Konzept, Implementierung und Testphase das so realisierte Projekt dem Endanwender zum Einsatz zu übergeben. Ein wesentliches Merkmal tritt dabei zum Vorschein. Der Projektumfang wird im wesentlichen während der Konzeptphase fixiert und in seiner Struktur eingefroren. Die Vorteile eines solchen Phasenmodell-Vorgehens (eine der häufigsten SE-Methoden) sind einleuchtend. Der Projektumfang läßt sich nämlich vergleichsmäßig leicht in kleinere Einheiten zerlegen und mehr oder weniger schablonenartig, unter Einsatz entsprechender Kontrollmechanismen, in einem Team *durchziehen*. Der Phasenmodell-Ansatz kann daher mit sicherem Vorteil dann Anwendung finden, wenn von vornherein

klar definierte Anforderungen vorliegen und bezüglich des zu realisierenden Vorhabens keine *Unklarheiten* bestehen (trifft dies oft zu?). Treten zum späteren Zeitpunkt, z.B. nach dem generellen Konzeptentwurf doch noch neue Aspekte auf, die einen strukturellen Charakter aufweisen und berücksichtigt werden müssen, führt dies in den meisten Fällen zu erheblichen Komplikationen (sehr oft kommen die Anwender erst während des System-Probelaufs aufgrund der realisierten Funktionalitäten darauf, was sie gerne hätten bzw. was ihnen eigentlich fehlt). Es ist nicht schwer zu erkennen, daß diese Vorgangsweise zum größten Teil statische Charakterzüge aufweist. Eine Idee wird realisiert, gefolgt von einer Pflege- bzw. Wartungsphase. Ein evolutionäres Vorgehen bzw. dynamische Anpassung der Implementierung an die neuen Gegebenheiten ist in der Regel fast unmöglich. Das Software-Produkt veraltet mit der Zeit und muß durch ein neues ersetzt werden. Deshalb werden substantielle Modifikationen bzw. Ergänzungen, die sich sehr oft erst durch die Benützung des Produktes als wünschenswert (notwendig) ergeben, insbesondere bei umfangreicheren Systemen, auch kaum gewagt. Es ist fraglich, ob dieser Weg zur Bewältigung der Software-Krise einen Beitrag zu leisten vermag.

### *Projektentwicklung in einer AI-Programmierungsumgebung*

Eines der zentralen Themen der AI-Forschung ist die Bewältigung des Komplexitätsproblems. Es wird davon ausgegangen, daß die meisten AI-orientierten Fragestellungen, die sich der AI-Gemeinschaft stellen, entweder ungenügend spezifiziert oder überhaupt nur zu einem geringen Teil verstanden werden. Es bedarf daher kaum weiterer Erklärung, warum hier die klassische Methodik des Software Engineering mit erheblichen Schwierigkeiten zu tun hat. Dies führte zur Konstruktion von adäquaten Programmierungsumgebungen, die eine schnelle, dynamische Ideenexploration und -realisierung ermöglichen, problemlos das Ausprobieren verschiedener Varianten gestatten sowie das *Durchspielen* alternativer Strategien zulassen (rapid prototyping). Sie fördern das Experimentieren mit Ideen, die Kreativität und den innovativen Geist.

Zu den hervorstechenden Merkmalen derartiger Programmierungsumgebungen gehören ihr modularer Aufbau; ferner die Möglichkeit, auch relevante Entwurfsentscheidungen zum spätestmöglichen Augenblick hinauszuschieben; ihre graphisch-orientierte, benutzerfreundliche Schnittstelle zur leichten Visualisierung der bereits realisierten Ideen. Eine AI-Programmierungsumgebung kennzeichnet insbesondere eine multiparadigmatische Programmiersprache (s. später), die verschiedene Wissensrepräsentationsarten ermöglicht, diverse tools wie Editor, Compiler, Interpreter, Linker, Librarian, sowie ein Benutzerschnittstelle mit Fenstertechnik (multiwindows, [21]), Dialogtechniken wie iconizing, Menus, toolboxes, Maus, trackball, Zugang zum Betriebssystem, u.d.m., die ein effizientes, **integrales Ganzes** bilden. Daß derartige Programmierungsumgebungen auch zur Lösung nicht AI-orientierter Probleme herangezogen werden können, liegt ja auf der Hand.

Derzeit gibt es eine Reihe eher teurer Knowledge Engineering Environments auf dem Markt ([12], [13], [14], [15], [16]), weiters im Forschungsstadium befindliche Environments wie etwa LOOPS ([1]), Modula-Prolog ([2]), manche *ausgebaute* LISP-Environments, u.a.m. Darüber hinaus sind bereits derartige Techniken und Eigenschaften in einem beachtenswerten Umfang in Personal Computers wie MACINTOSH, ATARI o.ä. vorzufinden.

### *Multiparadigmatische Programmiersprachen*

Ein kurzer Rückblick in die Entwicklungsgeschichte der Programmiersprachen läßt deutlich erkennen, daß bei ihrem Entwurf in den häufigsten Fällen die Vorstellung einer sequentiell ablaufenden, das Programm ausführenden Maschine zugrunde lag. Dieser Umstand prägte entscheidend einen in der Vergangenheit überwiegend angewandten imperativen Programmierstil. Die AI bevorzugt aufgrund spezifischer Aufgabenstellungen jedoch andere Programmierparadigmen, wie z.B. das funktionale-, logik- objekt- oder regelorientierte Paradigma. Weitere Forschungen letzter Zeit förderten Programmierparadigmen zutage, zu denen u.a. das relations-, plan-, operator-, data-structure-, oder zugriffsorientierte Paradigma gehören. Fast zwangsläufig ergibt sich nun die Frage, welche der erwähnten Paradigmen zur Lösung eines gestellten Problems am besten einzusetzen seien. Die Verfügbarkeit mehrerer Paradigmen innerhalb einer Programmeinheit, so scheinen bisherige Erfahrungen anzudeuten, bietet dem Programmierer größere Flexibilität in der Wahl eines adäquaten, der Problemstellung angepaßten Ausdrucksmittels. Dies kann wiederum zu eleganteren, übersichtlicheren, leichter überschaubaren und überprüfbareren, weniger fehleranfälligen Lösungen führen.

Natürlich bietet auch ein Betriebssystem mit seinen Sprachprozessoren mehrere Programmierparadigmen an. Deshalb kann ein gestelltes Problem, geeignet zerlegt, unter Zugrundelegung unterschiedlicher, in den jeweiligen Programmiersprachen vorgefundener Paradigmen realisiert werden. Ein wesentlicher Nachteil besteht allerdings in der strikten Trennung der einzelnen logischen Programmteile sowie in der statischen Art des Linkens. Gewünscht ist die koexistente Verfügbarkeit mehrerer Paradigmen innerhalb einer Programmiersprache.

Derzeitige Versuche laufen darauf hinaus, einerseits bestehende Programmiersprachen mit ihren inhärenten Paradigmen um neue *anzureichern*, wie z.B. das Kombinieren des assignment-, objekt-, logik-orientierten Paradigmas (Modula + Prolog), des funktional-, objekt-orientierten Paradigmas, des funktional-, objekt-, zugriffs-orientierten Paradigmas (LISP-Basis + Erweiterungen), u.d.m. Andererseits wird versucht, neue multiparadigmatische Programmiersprachen, die sog. Programmiersprachen der 4. Generation, zu entwickeln. Zum letzteren Versuch zählt auch LOOPS, eigentlich ein wissensbasiertes Programmiersystem, daß am Xerox Palo Alto Research Center in Kalifornien entwickelt wurde. Dieses Programmiersystem bietet das funktions-, objekt-, regel-, und seit kurzem auch das zugriffsorientierte Paradigma ([1]).

### *Zukunftstrends in Programm-Entwicklungsumgebungen*

Die wesentliche Aufgabe, die eine Programm-Entwicklungsumgebung (PEU) zu erfüllen hat, besteht darin, den Projektentwickler ("Designer") bei seiner kreativen Tätigkeit der Umsetzung einer Idee oder eines Konzepts in die Realität weitestgehend von allen "nebensächlichen", diesen Prozeß nicht betreffenden Tätigkeiten zu entlasten.

Die Benutzung des Werkzeuginstrumentariums innerhalb einer PEU sollte nicht von der intellektuellen Tätigkeit während der Ideenumsetzung ablenken. Vielmehr sollte sich der Designer ausschließlich auf die zu realisierenden Ideen sowie die "Ideenreifung" voll konzentrieren können.

Zur Entwicklungsgeschichte der PEU läßt sich folgendes anmerken. Zu den ersten sogn. "tools" der 1. Generation gehören Sprachprozessoren, Debugger, Profiler, usw. (manche

von ihnen erfreuen sich auch heute noch großer Beliebtheit). Das entscheidende Merkmal dieser tools : sie sind diskret und verrichten ihre Arbeit nicht im Kontext mit den anderen, d.h. sie operieren unabhängig voneinander. Sie arbeiten in einem "Edit-Compile-Run-Debug"-Zyklus, was zu einem fortwährenden Starten und Terminieren führt (in den meisten Betriebssystemen sind dies noch dazu sehr "teure" Operationen). Für den Designer kann sich daraus eine Fülle verschiedenst gearteter "Nebenbeschäftigungen mit den tools" ergeben, die von ihm nur als Ablenkung empfunden werden können, weil sie sein Problem nicht betreffen.

Zur 2. Generation der sog. "toolboxes" können PEU gezählt werden, bei denen die Eigenschaften und Funktionalitäten der diskreten Werkzeuge während des box-Entwurfes unter dem Aspekt der Integrität einheitlich berücksichtigt wurden. Tools der 2. Generation können einander "from inside" aufrufen. Z.B. ein Editor kann vom Debugger aufgerufen werden, das editierte File muß aber neu übersetzt und gestartet werden.

Die 3. Generation von PEU für "Life-cycle in Software Engineering" soll dem Designer während des ganzen Lebenszyklus des Projektes unterstützen und die Entwurfskomplexität in überschaubaren Grenzen zu halten helfen. (Es gibt derzeit nur wenige Systeme, die in diese Kategorie fallen).

Die 4. Generation im Gespräch befindlicher PEU trägt oft den Titel "Intelligent Tools". Dabei wird vom ausgiebigen Gebrauch gegenwärtig verfügbarer AI-Techniken (Expertensysteme) ausgegangen. Hierzu zählen insbesondere Forschungsprojekte wie z.B. "Programmer's Assistent" (MIT), Psi (Stanfort), die sich mit der Problematik des Programm-entwurfes, der Implementierung sowie Wartung in ganzheitlicher Form beschäftigen.

### *Zurück zur Gegenwart*

Der vorliegende Artikel möge unseren Kundenkreis auf einen Teilaspekt gegenwärtiger Entwicklung auf den Gebieten AI und Software Engineering aufmerksam machen, mit dem Hintergedanken, über die Effizienz mancher Projektrealisierungen sowie die Verwendung zeitgemäßer Werkzeuge nachzudenken. Die im Rahmen der Programmberatungstätigkeit gewonnenen Erfahrungen lassen erkennen, daß im Zusammenhang mit der Verwendung zeitgemäßer Programmiersprachen bzw. Werkzeuge seitens der Benutzer sehr oft zögernde Haltung zu verzeichnen ist. Als Beispiel soll hier lediglich angeführt werden, wie wenig Beachtung es findet, daß das VMS-Betriebssystem eine saubere Multisprach-Programm-entwicklung gestattet. (Kann dies die Bezeichnung 'zeitgemäß' noch verdienen?).

### *Literaturverzeichnis*

- [1] Stefik M.J., Bobrow D.G., Kahn K.M.: *Integrating Access-Oriented Programming into a Multiparadigm Environment*, IEEE Software, S.10 - 18, Jan. 1986
- [2] Muller C.: *Modula-Prolog: A Software Development Tool*, IEEE Software, S.39 - 45, Nov. 1986
- [3] Bobrow D. G., Stefik M.J.: *Perspectives On AI Programming*, Multiparadigmatisches Programmieren, Science Vol.231, S.951 - 957, Feb. 28, 1986
- [4] *Multiparadigm Research: A Survey Of Nine Projects*, IEEE Software, S.70 - 77, Jan. 1986
- [4a] Teitelman W.: *A Tour Through Cedar*, IEEE Trans. on Software Engineering, Vol. SE-11, No.3, S.285 - 302, March 1985



- [4b] Ramamoorthy C.V., Shekhar S., Garg V.: *Software Development Support for AI Programs*, Computer, S.30 - 40, Jan. 1987
- [5] Simon H.A.: *Whether Software Engineering Needs to Be Artificially Intelligent*, IEEE Trans. on Software Engineering, Vol. SE-12, No. 7, S.726 - 732, July 1986
- [6] Yau S.S., TSAI J.J.: *A Survey of Software Design Techniques*, IEEE Trans. on Software Engineering, Vol. SE-12, S.713 - 721, No. 6, June 1986
- [7] Basili V.R., Selby R.W., Hutchens D.H.: *Experimentation in Software Engineering*, IEEE Trans. on Software Engineering, Vol. SE-12, No. 7, S.733 - 743, July 1986
- [8] Tonhäuser V., Haas H.: *Prototyping zur Entwicklung von Software*, online 10, S. 28 - 32, 1986
- [8a] Sneed H.: *Software is vielfach sanierungsreif*, online 2, S.26 - 30, 1987
- [9] Inside Technology: *Bringing Order Out Of Chaos In Software Design*, Electronics, S.30 - 34, Nov. 4, 1985
- [10] Technology to watch: *A New Way To Move AI Into The Mainstream*, Electronics, S.90 - 92, Jan. 22, 1987
- [11] Swärtzel H.: *Informatik in der Praxis*, Springer, Okt. 1986
- [12] Rauch-Hindin, W.B.: *Artificial Intelligence in Business, Science and Industry*, Vol. I, Aufsätze über AI Trends and Applications, Knowledge Representation, Multiple paradigms, Environments, Programmierstile, Expertensystem-Werkzeuge, Prentice-Hall, 1986
- [13] O'Keefe R. M., Belton V., Ball T.: *Experiences With Using Expert Systems in Operational Research*, Journal of the Operational Research Society, Vol. 37, No. 7, S.657 - 668, 1986
- [14] Manuel T.: *What's Holding Back Expert Systems ?*, Electronics, S.59 - 63, Aug. 7, 1986
- [15] Technology to watch: *Inference's Strategy To Speed Things Up*, Electronics, S.66 - 69, Aug. 7, 1986
- [16] Holzbauer Ch., Pfahringer B.: *VIE-KET, ein hybrides Knowledge Engineering Tool*, Österreichisches Forschungsinstitut für AI, 1986
- [17] *Die Zukunft der Infosysteme*, Tagungsband, Springer 1986, Aufsätze über AI, 5. Comp. Generation, Comp. Architekturen, multiparadigmatische Sprachen
- [18] Duncan K.A., Harris D.I.: *Computers in Education*, Proceedings of the IFIP TC3, 4th World Conference on Computers in Education - WCCE 85, Norfolk, VA, USA, Aug. 1985, North-Holland
- [19] Rollinger C.R., Horn W.: *GWAI-86 und 2. Österreichische Artificial-Intelligence-Tagung*, Tagungsband, Springer 1986
- [20] Brachman R.J., Levesque H.J.: *Readings in Knowledge Representation*, 1985, Morgan Kaufman Pub's, Inc., California
- [21] Inside Technology: *Suddenly, work-station makers unite on a windowing standard*, Electronics, S.58 - 59, Jan. 22, 1987

Die folgenden, vom Autor besuchten Tagungen lieferten wertvolle Gedanken Anregungen:

*Die Zukunft der Informationssysteme* ( 16.-18. Sept. 1986, Uni Linz ), sowie  
 2. *Österreichische AI-Tagung* ( 22.-26. Sept. 1986, Ottenstein ).

