

Frank, A. U. "Reise in Die Vereinigten Staaten Von Nordamerika." 47.
Zürich: Eidesgenössische Technische Hochschule Zürich, Institut für
Geodäsie und Photogrammetrie, 1982.

REISE IN DIE
VEREINIGTEN STAATEN VON NORDAMERIKA

25. JULI BIS 14. AUGUST 1982

MIT ANHANG

EUROPÄISCHE KONFERENZ ÜBER
INTEGRIERTE, INTERAKTIVE COMPUTER-SYSTEME (EICSC)

STRESA 1. BIS 3. SEPT. 1982

ANDRÉ FRANK

VORWORT

Herr Dipl. Ing. A. Frank unternahm im Sommer 1982 eine Reise in die Vereinigten Staaten von Nordamerika, um dort drei Konferenzen zu besuchen und Vorträge zu halten. Im folgenden Bericht schildert er seine Eindrücke von diesem Aufenthalt.

Dabei stehen die Entwicklungstendenzen in der EDV - besonders im Bereich der Interaktivität zwischen Mensch und Maschine - im Vordergrund. Weil diese Interaktivität stark von den Möglichkeiten der Hardware abhängig ist, ist es wichtig, dass wir unsern Forschungs- und Entwicklungsarbeiten langfristig zutreffende Vorstellungen über die zu erwartende Entwicklung der Hardware und der Betriebssoftware zugrunde legen.

Die neu auf den Markt kommenden Arbeitsplatz-Rechner, die den Zugriff zu gemeinsam verwalteten Daten erlauben, verbessern und vereinfachen die erwähnten Interaktionsmethoden. Das EDV-System wird damit 'benutzerfreundlicher' und leichter zu bedienen sein..

Ich hoffe, den Leser mit diesem Bericht überzeugen zu können, dass sich die Reise von Herrn Frank in die USA gelohnt hat.

R. Conzett

INHALTSVERZEICHNIS

VORWORT	2
INHALTSVERZEICHNIS	3
A REISE IN DIE VEREINIGTEN STAATEN VON NORDAMERIKA	5
1. Reiseziele	5
2. Reiseroute	6
3. Allgemeine Eindrücke	7
4. SIGGRAPH	8
5. SIGSMALL	17
6. Schlumberger-Doll Research	20
7. Land Information at the Local Level	21
8. Schlussbemerkung	26
B EUROPÄISCHE KONFERENZ ÜBER INTEGRIERTE, INTERAKTIVE COMPUTER-SYSTEME	27
1. Thema	27
2. Gliederung des Berichtes	28
3. Xerox Star	29
3.1 Prinzipien der Benützerschnittstelle	29
3.2 Modus - ohne Modus	31
4. Object-oriented approach (SMALLTALK)	33
4.1 Objekte und Nachrichten	33
4.2 Vererbung (Inheritance)	34
4.3 Andere Ansätze	35
4.4 Beurteilung	35

5. Unterstützung der Programmierung	36
5.1 Strukturierte Editor	36
5.2 Programmieren durch Beispiel	36
5.3 Automatische Programmerstellung	37
5.4 Schrittweise (inkrementale) Kompilation und Unterstützung des Testens	37
5.5 Versionskontrolle	37
5.6 Anmerkungen zu Programmen	38
5.7 Ingetration	38
6. Anwendungen von AI	39
7. Synchrone oder anychrone Benützerschnittstelle	40
8. Folgerungen für die Arbeit am IGP	42
8.1 Bildschirmverwaltung	42
8.2 Benützerschnittstelle	42
8.3 Asynchrone Benützerschnittstelle	42
8.4 Objekt-orientierte Programmierung	43
8.5 Programmierumgebung	43
LITERATURVERZEICHNIS	44

A REISE IN DIE VEREINIGTEN STAATEN VON NORDAMERIKA

1. Reiseziele

Ziel dieser dreiwöchigen Reise war der Besuch der folgenden drei Konferenzen:

- Association for Computer Machinery,
Special Interest Group for Graphics (SIGGRAPH)
Konferenz in Boston, Massachusetts, 26. bis 30. Juli 1982
- Association for Computer Machinery,
Special Interest Group for Small Systems (SIGSMALL)
Konferenz in Colorado Springs, Colorado, 2. bis 4. Aug. 1982
- University of Maine at Orono,
International Symposium on Land Information at
the Local Level, Orono, Maine, 9. bis 12. August 1982

An jeder Konferenz hatte ich einen Vortrag zu halten:

- MAPQUERY Data Base Query Language for Retrieval of Geometric
Data and its Graphical Representation (SIGGRAPH)
- PANDA A Pascal Network Database Management System
(SIGSMALL)
- Management of Spatially Referenced Data (Orono)

2. Reiseroute

Mit einem Swissair-Flug erreichte ich Boston von Zürich aus direkt. Nach Abschluss von SIGGRAPH flog ich nach Denver, Colorado, und mietete einen Wagen, um in einer grossen Schleife durch die Rocky Mountains, Colorado Springs zu erreichen.

Nach der SIGSMALL Konferenz flog ich nach New York. Für den Besuch der Stadt stand mir nur wenig Zeit zur Verfügung, da ich noch das Forschungslaboratorium der Firma Schlumberger in Ridgefield, Connecticut, besuchen wollte. Wiederum mietete ich einen Wagen, mit dem ich von New York nach Ridgefield und von dort nach Boston fuhr. Die Strecke Boston - Orono, Maine legte ich in beiden Richtungen in Privatwagen von Vermessern aus Boston zurück. Schliesslich flog ich wiederum mit Swissair von Boston direkt nach Zürich zurück.

Ich hatte mit dieser Reiseroute Gelegenheit, einerseits die traditionsreiche New England Region relativ eingehend zu erleben und andererseits einen Blick auf die grossartige Bergwelt der Rocky Mountains zu erhaschen. Beeindruckt hat mich dabei die fast überall fühlbare Weite des Landes.

3. Allgemeine Eindrücke

Neben vielen Beobachtungen über den amerikanischen Lebensstil, die für die Arbeit des IGP wohl kaum wesentlich sind, war für mich die Grösse der Unterschiede innerhalb der USA überraschend. Besonders fiel mir der enorme Unterschied im Stand der Kenntnisse zwischen den bei uns bekannten, besten amerikanischen Universitäten und den unbekannteren auf, die sehr deutlich gegenüber der ETH zurückstehen. Das zeigte sich am deutlichsten im Vergleich der Beiträge zu SIGGRAPH (Autoren von besten Universitäten und Forschungszentren) und SIGSMALL (Autoren von kleineren Universitäten und kleineren Firmen).

Aehnliches gilt wohl auch für die vorhandenen Ausrüstungen. Für mich wurde dies offenkundig im Vergleich der Ausstattung von Schlumberger-Doll Research (wesentlich besser und moderner ausgerüstet als z.B. das Institut für Informatik der ETH Zürich) und der University of Maine at Orono, wo die geodätischen Instrumente deutlich altmodischer als die des IGP sind.

Die folgenden Bemerkungen beschreiben demnach nicht den Stand in der USA allgemein, sondern orientieren sich im allgemeinen an der Front der Entwicklung.

4. SIGGRAPH

Diese riesige (18'000 Teilnehmer, davon über 3'000 für das technische Vortrags-Programm und über 3'000 für einführende Lehrveranstaltungen, sogenannte Tutorials), von Profis hervorragend organisierte Konferenz und Ausstellung zeigte Spitzenleistungen in der Behandlung geometrischer Sachverhalte mit Hilfe von Computern.

Kommunikation Mensch - Maschine

Die Kommunikation zwischen Benutzer und Maschine ist als wichtigstes Problem erkannt; graphische Hilfsmittel sollen die Lösung erleichtern. Technische Grundlage dazu bilden Bildschirme, deren Bildpunkte (im allgemeinen 1000 x 1000 pixel) vom Computer einzeln gesteuert werden können, sogenannte 'bit mapped screen' (wie Lilith-Computer [Wirth81]). Solche Bildschirme sind einerseits als Terminal (BBN Computer Corp. ca. SFr. 10'000.--) erhältlich, meist aber in Kleinsysteme integriert (siehe Punkt 'Hardware').

Dem Benutzer präsentiert sich ein solcher Bildschirm wie sein (Papier-) Arbeitsplatz: verschiedene Dokumente sind gleichzeitig sichtbar und können sich gegenseitig überdecken. Er kann ein einzelnes Dokument hervorziehen und vergrössern, um es genauer zu betrachten, wodurch eventuell andere Dokumente überdeckt werden [Lipkie82] Mit Hilfe von Signeten (icons) werden auch abstrakte Begriffe, wie directory, file etc. dargestellt und durch graphisch sichtbare Aktionen des Benützers manipuliert - die sonst schwierig zu erreichende Vorstellung über die unsichtbaren Vorgänge im Innern des Computers werden damit durch Graphik sichtbar und damit anschaulich und verständlich gemacht. Diese Art von Interaktion mit dem Benutzer scheint zumindest für bestimmte Anwendungen sehr vorteilhaft; sie ist über das Forschungsstadium hinausgelangt und wird als Produkt im Xerox Star Personal Computer kommerziell angeboten.

In Tutorial 'The Design and Implementation of user-computer Interfaces' von James D. Foley wurden systematisch einzelne Punkte der Gestaltung von Benutzer-Schnittstellen behandelt. Ohne wesentlich Neues zu sagen, zeigen die abgegebenen Unterlagen im Ueberblick was zu beachten ist: "einheitlich und klar" ist die wichtigste Forderung.

Die Verwendung von formaler Hilfsmittel für die Sprachdefinition (Backus-Naur-Form, Syntax-Diagramm, State-Transition-Diagrams) lässt diese Ziele leichter erreichen. Zunehmend werden auch Compiler-Compiler eingesetzt, die ein maschinelles Erzeugen der Interaktionsprogramme erlauben. Eine strikte Trennung zwischen Interaktionssteuerung und Verarbeitungsroutinen wird empfohlen, weil sie Anpassungen erleichtert. Forschungsgruppen haben auch schon Sprachen zur einfachen Definition von Benutzerinterfaces entwickelt [Wong82] [Kasik82].

Meines Erachtens helfen auch die Bemühungen um eine klare Definition der Semantik der Input Devices zur Vereinfachung und Vereinheitlichung von interaktiv-graphischen Programmen mit [Anson 82] [Rosenthal82].

Auf verschiedene Benutzer-Anforderungen muss Rücksicht genommen werden ('user profile' als Lösung).

Für die Gestaltung von Programmen für die Parzellarvermessung ist die Warnung, geometrische und sprachliche Operationen nicht zu vermischen, wichtig. Beispielsweise muss vermieden werden, das visuell-optische Konzept '+' in einen sprachlichen Befehl 'HINAUF' umzusetzen, sondern ein direkter geometrischer Befehl (z.B. durch Bewegen des Cursors) ist angebracht.

Operating Systems

Betriebssysteme haben einen bedeutenden Einfluss auf die Struktur der Interaktion mit dem Benutzer. Alle neueren Systeme bieten UNIX oder ein UNIX-ähnliches Betriebssystem an.

Neu erscheint die Objekt-orientierte Sprach-Architektur (z.B. Smalltalk von Xerox [Goldberg81] oder Thoth [Cheriton79], siehe auch den folgenden Bericht über die ICES), die sich besonders für die einfache Gestaltung graphischer Interfaces eignet. Zum Teil wird angenommen, dass Smalltalk, als am weitest verbreiteten Vertreter der Objekt-orientierten Programmiersprache, UNIX-artige Betriebssysteme in absehbarer Zukunft (vielleicht in 3 - 5 Jahren) ablösen wird [CM82].

Eine Umstellung auf objekt-orientierte Architektur sollte uns wenig Probleme schaffen, sofern die Modularisierung schon heute konsequent mit abstrakten Datentypen [Parnas72] [Guttag77] erfolgt.

Hardware, die besonders auf diese Bedürfnisse zugeschnitten ist, wird bereits angeboten (XEROX Dolphin, SFr. 90'000.--, Symbolics 3600, ca.SFr.160'000.-pro Arbeitsplatz!) und dürfte rasch billiger werden. Der neue Prozessor Chip 286 von Intel scheint geeignet, solche Betriebssysteme zu unterstützen.

Graphische Darstellung

Die Qualität der graphischen Darstellung wird durch verschiedene Verfahren verbessert.

- Die Auflösung des Bildschirms wird erhöht (Farbbildschirme mit 1000 x 1000 pixel sind verschiedentlich erhältlich z. B. Chromatics für ca. SFr. 30'000.--, der Schirm allein ca. SFr. 10'000.--).
- Farbbildschirme werden mit mehr (billigem) Speicher ausgestattet. Es kann dann für jeden Bildpunkt die Farbe individuell durch den Anteil von Rot, Gelb und Blau (RGB) auf z.B. einer 32-teiligen Skala (6 bit) ausgewählt werden, was natürliche Farbwiedergabe erlaubt. Billigere Systeme reservieren pro Bildpunkt insgesamt nur 4 .. 8 bit, mit denen aus einer Farbtabelle eine von 8 .. 256 Farben ausgewählt wird; die Farbe der Tabelle kann jeweils pro Bild durch Rot-, Gelb- und Blau-Anteile festgelegt werden. Das ganze Verfahren wird 'pseudo colors' mit 'color look up table' genannt.
- Anti-aliasing vermindert den störenden Effekt der Treppentufen bei schrägen Linien. Verschiedene Methoden werden diskutiert und sind z.T. schon in Produkten implementiert (z.B. Jupiter 7, 768 x 775 pixel (europäische TV Norm) mit 8 bit per pixel ca. SFr. 30'000.--).

- Zeichengeräte sind durch vermehrten Einsatz von Prozessoren schneller und genauer geworden (Nicolet Zeta 5400 137 cm breit, Auflösung 0.0125 mm ca. SFr. 100'000.--).

Neuerdings können in elektrostatischen Plottern auch masshaltige Zeichenfolien verarbeitet werden (Versatec 106 cm breit, Auflösung 0.4 mm ca. SFr. 110'000.--). Elektrostatische Plotter produzieren ca. 30 cm Plan voller Breite pro Minute, unabhängig von der Dichte der Zeichnung. Die Qualität ist einer Heliographie durchaus vergleichbar. Bemerkenswert ist der Computer-gesteuerte Xerox Drucker 9700, der Schrift und Zeichnung mit einer Auflösung von ca. 0.25 mm druckt (Preis ca. SFr. 800'000.--, Leistung 120 Kopien A4 pro Minute!).

Raster - Vektor Display

Die erhöhte Qualität der Rasterbildschirme und der sinkende Preis für den notwendigen Bildspeicher machen Rasterdisplays zunehmend attraktiv. Dennoch bleibt der Vektor-Display für einfarbige Darstellungen mit grossem Detailreichtum im Moment noch notwendig. Komplexe, rasch bewegte Bilder lassen sich nach wie vor am besten auf refresh vector displays zeigen.

Rasterverarbeitung

Neu scheinen mir Ansätze zu einer systematischen Behandlung von Rasteroperationen, die Raster als eigenständig und nicht als Annäherung versteht [Guibas82]. Verschiedene Grundoperationen werden von den Hardware-Herstellern bereits angeboten. Was im allgemeinen noch fehlt, sind Software-Pakete für die Verarbeitung mehrfacher Windows auf einem Schirm [McKeown 82] und die zugehörigen Konzepte für das Benutzer-Interface.

Input-Devices

Neben Tastatur und Buchstaben-Bildschirm stehen verschiedene neue Hilfsmittel für die Gestaltung der Kommunikation zwischen Benutzer und Maschine zur Verfügung.

Interessant (und billig) ist Touch-screen, eine Folie auf dem Bildschirm, die z.B. Neu-Auswahl durch antippen mit dem blossen Finger erlaubt (Preis ca. SFr. 2000.--).

Spracherkennung ist wesentlich billiger geworden (von ca. SFr. 4000.- aufwärts) und scheint für gewisse Anwendungen ausserordentlich geeignet (z.B. Digitalisieren), weil dabei beide Hände frei bleiben.

Untersuchungen über verschiedene geometrische Inputtechniken zeigen, dass

- Maus, light pen und tablet etwa gleich schnell sind,
- der joystick akzeptable Resultate bringt,
- hingegen die Steuerung eines Cursors über Tasten nicht befriedigt.

Hardware

Auf dem Motorola 68000 Prozessor aufgebaute Systeme beherrschen den Markt. Dabei wird die Leistung eines M 68000 Chip als etwa einem Viertel einer VAX 780 entsprechend eingeschätzt! In den meisten Fällen wird UNIX oder ein ähnliches Betriebssystem angeboten. Als Programmiersprachen dominieren Pascal und C.

Im allgemeinen bewegen sich die Preise um ca. SFr. 40'000.-- (komplett mit Winchester-Platte, Bildschirm etc.).

Das billigste System scheint Charles River Data Systems anzubieten: 256 kbytes Hauptspeicher, 10 MBytes Winchester-Platte und Floppy-disk mit Software etwa SFr. 30'000.--.

Zusätzlich erscheint spezielle Hardware für Mikroprozessorsysteme auf dem Markt:

- Graphik-Karten bis 1000 x 1000 Auflösung (z.B. für 8 Farben ca. SFr. 10'000.-- oder pro 2 Farben SFr. 5'000.--),
- Vektor-Prozessor-Karten, welche z.B. zwei 40 x 40 Matrizen in 0.7 sec. multipliziert (ca. Fr. 12'000.--).

Unentschieden scheint im Moment, ob der Versabus (Motorola) oder der Multibus (Intel) allgemeinere Verbreitung finden wird.

Schliesslich werden LSI und VLSI chips entwickelt, um die bei der graphischen Darstellung häufigen Operationen zu beschleunigen. Davon dürften sowohl Vektor als auch Rasterdisplays profitieren.

Interaktiv-graphische Systeme für die Konstruktion

Viele Anbieter zeigen allgemein anwendbare interaktiv-graphische Systeme. Hauptsächlichste Stossrichtung sind Elektrotechnik (Chip und board design) und mechanische Konstruktion. In vielen Fällen werden Farbdisplays eingesetzt (besonders für Anwendungen der Elektrotechnik). Ich habe den Eindruck, dass sich die Tendenz weg vom zentralen Rechner, z.B. VAX 780 für wenige Arbeitsplätze und hin zum lokalen Prozessor pro Arbeitsplatz (häufig Motorola 68000) verstärkt. Voraussetzung ist eine Verbindung zwischen den Prozessoren über ein Netz (z.B. Ethernet).

Die wenigsten Hersteller erlauben aber bereits volle Programmierung der Arbeitsstation. Die Steuerung der Graphik erfolgt fast überall über lineare display Lists [Newman79]; allenfalls sind Rekursion oder Sprünge möglich. Dadurch müssen dann die auf dem Hauptrechner ablaufenden spezifischen Anwenderprogramme alle schwierigen Aufgaben übernehmen.

Business Graphics

Unter diesem Schlagwort werden leicht anzuwendende und komfortable Systeme zur Erzeugung von einfachen graphischen Darstellungen, wie sie in der Wirtschaft auftauchen, verstanden. Sie werden in grosser Zahl, mit unterschiedlichen Möglichkeiten und zu unterschiedlichen Preisen angeboten, dürften aber für uns kaum Bedeutung haben.

Geographische Datenverarbeitung

Geographic Systems Inc. bietet ein komplettes Hardware-Software-Grunddaten Paket an, um verschiedenartige wirtschaftliche Daten graphisch sichtbar zu machen (vgl. [Frank82]).

Scitex hat ihr Response-280 System für die Redaktion von Zeichnungen, speziell Landkarten, erneut vorgestellt. Die Unterstützung des Bearbeiters für die graphische Ueberarbeitung von Kartenentwürfen scheint beachtlich, hingegen fehlt die Unterstützung für die Ueberarbeitung der inhaltlichen Dimension vollständig. Ein solches System wird in Bern bei Kümmerli & Frei AG eingesetzt.

Abfragesprachen

Das Bedürfnis für Abfragesprachen, die gezieltere Operationen als rein graphisches 'Zoom' und 'Pan' erlauben, ist auch ausserhalb kartographischer Anwendungen, z.B. in der medizinischen Datenverarbeitung, fühlbar. Klar ist, dass eine Abfragesprache für geometrische Objekte sprachlich-syntaktisch zwar komplexe Ausdrücke ebenbürtig den Abfragesprachen für kommerzielle Datenbanken erlauben muss, dass aber die Elemente der Sprache nicht zwingend Buchstabenketten sein müssen, sondern dass diese auch aus einer Auswahl graphisch ausgewählt werden können [Senko77] [Friedel82].

Die von mir vorgetragenen Ideen über MAPQUERY stiessen auf einiges Interesse. Direkt führten sie zu längeren Gesprächen mit Mitgliedern von Arbeitsgruppen, die LIS-ähnliche Probleme behandeln (Tokyo, Calgary).

Generalisierung

Generalisieren als typisch kartographische Aktion muss bei der Präsentation geometrischer Objekte in unterschiedlicher Grösse generell angewandt werden; einzelne Ansätze lassen sich erkennen [Crow82] [Friedel82], aber eine bewusste, einheitliche und umfassende Bearbeitung steht noch aus.

Datenbank

Ein grosser Teil der Probleme der graphischen Datenverarbeitung sind Datenstrukturierungs- und Datenverwaltungsaufgaben. In zunehmendem Mass wird erkannt, dass die Datenbank-Methodik zur Lösung dieser Aufgaben beitragen kann - nur in Ausnahmefällen werden hingegen schon konkrete Anwendungen sichtbar oder sind Forschungsprojekte bekanntgeworden (am wichtigsten wohl bei Computer Corp. of America).

Graphik-Standard

Ogleich immer noch stark diskutiert, scheint doch das Interesse an den vorgeschlagenen Graphik-Standards abzunehmen. Verschiedene Anbieter behaupten, den amerikanischen Core Standard liefern zu wollen, wogegen der (europäische) GKS Standard nächstens internationaler ISO Standard werden soll. Gesprächsweise stellt man aber auch bei anderen Anwendern fest, dass diese Vorschläge als zu komplex und dennoch zu wenig genau auf ihre spezielle Aufgabe abgestimmt empfunden werden. Ueberraschend häufig werden sehr einfache Pakete verwendet. Besonders unbefriedigend scheinen die Datenspeicherungs- und Strukturierungsvorschläge.

Zusammenfassung

Die SIGGRAPH Konferenz ist in wenigen Jahren aus unbedeutenden Anfängen zu einer der weltweit grössten Computer-Konferenzen geworden. Die diesjährige Schau umfasste, besonders in der Ausstellung, praktisch alle Gebiete der Informatik.

Erstaunlich ist höchstens das fast völlige Fehlen der grossen Namen der EDV-Branche (Univac, NCR, Honeywell etc.); einzig IBM war mit den 'neuen' Farbterminals für Business-Graphics vertreten, die aber kaum beachtet wurden. Bei allen technisch hochstehenden Lösungen werden Mini- und Mikrocomputer verwendet, am besten ein Prozessor pro Arbeitsplatz. Grosscomputer lassen sich höchstens für einfache, wenig Interaktivität benötigende Business Graphics einsetzen.

Sollen Computer-Anwendungen vermehrt direkt dem 'naiven' Benutzer zugänglich sein, so werden graphische Darstellungen zunehmend Verbreitung finden. Kommt eine Zeit, in der alphanumerisch-interaktive Programme, wie wir sie heute verwenden, ebenso veraltet sind, wie heute Batch-Programme und Terminals ohne mehrfarbige graphische Darstellungen zum alten Eisen gehören?

5. SIGSMALL

Diese sehr kleine (total ca. 60 Teilnehmer), bescheiden organisierte Konferenz zeigte vor allem den Unterschied im Stand des Wissens zwischen Spitze und grossem Haufen.

Beeinträchtigt wurde die Konferenz - wie übrigens auch SIGGRAPH, aber dort in umgekehrter Richtung - durch die Unsicherheit über ihr Thema. Was ist ein kleines System? Hobby Computer, Small Business Systems, physisch kleiner, aber höchst leistungsfähiger Computer für militärische Anwendungen? Gibt es typische Small Systems Probleme?

Einige der systematisch diskutierten Themen:

Sicherheit

Verschiedene Autoren berichteten über Anstrengungen, sichere Betriebssysteme für militärische Anwendungen zu erstellen. Ziel ist, garantieren zu können, dass bestimmte Daten vor unberechtigtem Zugriff, auch vor Benützern, die Zugang zu Hard- und Software haben, geschützt werden können. Heutige Systeme schützen höchstens vor andern Benützern, sofern diese nicht privilegiert sind oder physischen Zugang zur Anlage haben; zusätzlich muss damit gerechnet werden, dass irgendwelche vorhandenen Fehler der Betriebssysteme einem sehr gut orientierten, böswilligen Benutzer erlauben, die einfachen, oberflächlichen Sicherheitsvorkehrungen zu überspielen. Sichere Betriebssysteme siedeln die Sicherheitsvorkehrungen sehr weit unten nahe der Hardware an, so dass sie ohne Änderungen der Hardware nicht übergangen werden können; so sind sie direkt im Chip eingebaut, sind sie nach heutiger Auffassung unzugänglich und damit vor Veränderung sicher. (Das wird bereits bei Verschlüsselungssystemen angewandt, bei denen der Code in einem Chip eingebaut ist und jeder Versuch, den Code zu lesen, diesen löscht.)

Diese Aufgabe zerfällt in zwei Teile: einerseits die Formulierung einer Geheimhaltungs-Politik und andererseits die Erstellung und Programmierung des Computersystems. Für die vom U.S. Verteidigungsministerium (Department of Defense = DoD) gewählte militärische Geheimhaltungspolitik ist die Grundlage gegeben, hingegen scheint die Formulierung einer Geheimhaltungspolitik für den zivilen Sektor noch Schwierigkeiten zu bereiten.

In diesem Sinne sichere Betriebssysteme bauen auf einer Ring-Struktur von zunehmend geschützteren Operationen auf (Multics, Honeywell, Intel 286) und haben Ähnlichkeit mit einer object-oriented architecture (Nebula). Aufmerksamkeit erfordert die Kontrolle des Input/Output-Datenflusses.

Die Entwürfe zu sicheren Betriebssystemen werden heute formal überprüft; angestrebt, aber noch nicht erreicht ist die formale Kontrolle des Codes.

Die vorgeschlagenen Verfahren lassen erwarten, dass sichere Betriebssysteme, wenn sie durch die Hardware angemessen unterstützt werden, kaum viel teurer und langsamer sein werden, als die heute üblichen. Weil diese Art von Sicherheit für die heute angestrebte Programmiermethodik Vorteile bietet, kann angenommen werden, dass Betriebssysteme allgemein 'sicher' werden.

Intel Prozessor 286

Von einem Konstrukteur von Intel wurde der 286 (16-bit) Chip vorgestellt, der mit den 8088 (8-bit) und 8086 (16-bit) kompatibel ist. Bemerkenswert ist die starke Integration von Betriebssystemaufgaben - z.B. die Verwaltung von virtuellen Adressräumen der Benutzer. Zusammen mit einem speziellen Chip für numerische Operationen (64-bit) ergibt sich eine ausgesprochen leistungsfähige und schnelle Kombination, die möglicherweise für bestimmte Anwendungen billiger sein dürfte als der heute dominierende Motorola 68000.

Der seit langem angekündigte 32-bit Chip 432 hingegen wird noch auf sich warten lassen. Er funktioniert zwar, aber die Arbeitsgeschwindigkeit ist noch wesentlich zu klein, um konkurrenzfähig zu sein.

Anpassung des Benutzerinterfaces an unterschiedliche Bedürfnisse

Brigham Young, University in Provo (Utah) hat ein Benutzerinterface, das den Wechsel zwischen Keyword und Menu erlaubt (ähnlich wie das ursprüngliche INTRA Konzept?).

Hard- und Software

Schliesslich ist ein LISP Interpreter und Compiler für den Zilog Z80 Mikroprozessor und ein Ada (Subset) für eine spezielle Version des LSI-11 16-bit Mikroprozessors (Super Micro von Western Digital, Preis von SFr. 15'000.-- aufwärts) erwähnenswert..

Datenbanken

Verschiedene Vorträge befassten sich mit Datenbanken auf Kleinsystemen, wobei unter Datenbank im allgemeinen eher ein Record Manager (wie beispielsweise RMS auf den SWT 09) verstanden wurde, als eine echte Datenbank mit den Strukturierungsmöglichkeiten und dem Funktionsumfang wie in [Zehnder81] beschrieben.

Für PANDA wurde lebhaftes Interesse bekundet; ein vergleichbares System scheint auch in den USA im Moment nicht verfügbar.

Intel arbeitet an einer back-end Datenbankmaschine. Sie soll ohne spezielle Hardwarekomponenten auskommen und auf dem relationalen Datenmodell beruhen.

Netzwerke

Ein Tutorial über Local Area Networks schloss die Tagung ab. Das verteilte Material gibt einen guten Ueberblick über die heute untersuchten Techniken, insbesondere werden die Ring-Netzwerke ausführlich dargestellt.

6. Schlumberger-Doll Research

Die Firma Schlumberger beschäftigt sich weltweit mit der Untersuchung von Oel-Bohrlöchern. Verschiedene Sonden messen unterschiedlichste physikalische Parameter der Gesteinsschichten, deren Auswertung nachher Hinweise auf Art, Umfang und Ausbeutungsmöglichkeit von Oelvorkommen geben.

Zur Verbesserung ihrer Dienstleistung besteht in Ridgefield, Connecticut, ein Forschungsinstitut, das sich mit Grundlagenforschung im Bereich Gesteinsmechanik, Strömungslehre in porösen Körpern, Nuklearphysik, Akustik und Informatik beschäftigt. Ungefähr 250 Personen (davon ca. 150 Wissenschaftler) und die ausserordentlich reichhaltige Ausrüstung sind in einem sehr geschmackvollen Gebäude in einer New England Parklandschaft untergebracht.

Ich besichtigte Teile der Informatik-Gruppe, die mit einer grösseren CDC-Anlage, 3 VAX 780, 2 Evans & Sutherland Picture Systems, 2 Farbdisplays und einer grösseren Zahl von Xerox Dolphin (eine Anzahl Perq's sind bestellt) ausserordentlich gut und modern ausgerüstet ist.

Die bearbeiteten Probleme umfassen den Einsatz neuer Erkenntnisse der Artificial Intelligence Forschung für die Bildung von Expert-Systems zur Auswertung der Bohrlochdaten. Insbesondere sollen Programme für die numerische Auswertung automatisch, nach bestimmten Parametern, erstellt werden. Wichtig ist schliesslich auch die graphische Darstellung der Resultate (geologische Profile, dreidimensionale Darstellungen etc.). Im Moment wird auch an der Einführung eines Datenbanksystems gearbeitet (INGRES), das gemeinsame Daten für mehrere Programme verfügbar machen sollte.

7. Land Information at the Local Level

Die Universität des Staates Maine in Orono ist eine der wenigen amerikanischen Universitäten, die ein Vermessungsprogramm anbietet, (daneben: Ohio State University, University of Wisconsin und Washington State University in Seattle). Die Ausbildung auf undergraduate level (4 Jahre) führt etwas weiter als bei uns die HTL Ausbildung (z.T. beinahe Niveau eines Kult.-Ing. ETH), wobei aber die Ausbildung fast ausschliesslich auf Vermessung ausgerichtet ist. Ein anschliessender zweijähriger Kurs, der zum Master führen soll, ist geplant; er soll vor allem Landinformationssysteme und Landnutzungsplanung umfassen. Die vier Assistenzprofessoren betreuen neben den ca. 10 Vermessungsstudenten pro Kurs (dieses Jahr 20 Anfänger) auch Vermessungsübungen für Bauingenieure und Förster; die Ausrüstung ist im Vergleich zum IGP bescheidener. Die Absolventen sind gesucht und fanden bisher problemlos eine Stelle, was den USA nicht ganz selbstverständlich.

Die Konferenz gab durch zahlreiche eingeladene Vorträge einen guten Ueberblick über die heutige Praxis im amerikanischen Vermessungswesen. Bevor auf Einzelheiten eingegangen werden kann, will ich versuchen, das amerikanische Vermessungssystem stark verallgemeinernd zu schildern [Thurnherr80].

Parzellar-Vermessung ist Aufgabe der County (entspricht ungefähr der Schweizerischen Gemeinde, meist grösser und viel mehr Fläche umfassend). Die Gemeindeverwaltung, d.h. der meist gewählte, nicht speziell ausgebildete County Surveyor, beschränkt sich auf die Registrierung von Parzellenbeschreibungen, ohne deren geometrische Richtigkeit zu prüfen. Die von privaten Vermessungsbüros erstellten Vermessungsunterlagen beschreiben nur eine Parzelle, meist mit minimaler Anknüpfung an Fixpunkte und ohne Bezug zur allgemeinen Situation oder zur Nachbarparzelle.

Extrem ist dies im Osten (New England), wo ausschliesslich Grenzbeschreibungen (Metes and Boundaries) verwendet werden. Im Westen wurde das Land durch den Staat in rechteckige Flächen aufgeteilt, deren Eckpunkte geodätisch (mehr oder weniger genau) bestimmt wurden. Diese Rechtecke wurden meist auf einmal weiter aufgeteilt, so dass ziemlich zusammenhängende Vermessungs-Inseln entstanden.

Der County Surveyor oder eine Firma, die sich auf die Versicherung von Land-Eigentums-Titeln spezialisiert hat, erstellt Index-Maps, welche erlauben herauszufinden, welche Dokumente Parzellen in einem bestimmten Gebiet beschreiben.

Die Grenzen werden durch die Parzellen-Beschreibung des Vermessers nur ungefähr angegeben, die exakte Lage geht nur aus der örtlichen physischen Versicherung der Grenzen durch Bolzen, Stein, Hecke, Mauer etc. hervor (Prinzip der 'general boundary').

Anstelle des Grundbuches für den Eigentumsnachweis hilft eine vollständige Kette von Eigentumsübertragungen, die durch eine sog. title search nachgewiesen werden. Für 'clear titles' ist eine Versicherung für ca. Fr. 600.-- erhältlich, die den Titel (nicht aber die Grenzen) garantiert.

Die meisten Counties erstellen für die Erhebung der Grundsteuer eine tax map, die eine übersichtliche Darstellung gibt, aber keine rechtliche Bedeutung hat. Aus dieser tax map und bestimmter Bewertungskriterien folgt die Bewertung der Liegenschaft für Steuerzwecke.

Weil die Baubewilligungsbehörde der County meist ebenfalls nur über gewählte 'Amateure' und keinen geschulten Angestelltenstab verfügen, ist es im allgemeinen ebenfalls Aufgabe der privaten Vermessungsbüros, die Unterlagen für das Baubewilligungsverfahren vorzubereiten. Insbesondere müssen sie zeigen, dass die verschiedenen (Federal, State und County) Baubeschränkungen nicht verletzt werden - heute eine sehr aufwendige Abklärung.

Im folgenden sollen nun einige der in den Vorträgen erwähnten Probleme kurz erwähnt werden.

Parzellar-Vermessung wird in USA nur so gut und so umfassend gemacht, wie es im Moment erforderlich ist; alte Vermessungsunterlagen sind deshalb nur selten wiederverwendbar und die Vermessungsarbeit beginnt in praktisch allen Fällen mit einer Neuaufnahme (und nicht wie bei uns mit dem Zusammenstellen der vorhandenen Pläne).

Vordergründig scheint das amerikanische Verfahren für die Parzellarvermessung kostengünstig: die Vermessung einer Einfamilienparzelle ist eventuell schon für Fr. 100.-- zu haben. Ob das Verfahren insgesamt, d.h. inklusive des aufwendigen title search und anderer notwendiger Dienstleistungen wirtschaftlich ist, wird bezweifelt. Welchen Wert soll dabei der Rechtssicherheit, die beim amerikanischen System nicht in gleichem Masse wie bei uns gegeben ist, zugeschrieben werden? Werden zusätzliche Nutzen einbezogen, die durch das Vermessungswesen erzeugt werden, so scheint das US-System nicht wirtschaftlicher zu sein. Zu oft werden gleiche grundlegende Vermessungsvorgänge für unterschiedliche Zwecke wiederholt.

In Gegenden, wo qualitativ bessere Vermessungswerke erstellt und nachgeführt werden, werden diese jedenfalls intensiv genutzt.

Hindernisse für eine Besserung sind:

- die stark wählerfolgs-orientierten County Surveyors, die selten längerfristig planen und meist momentane Kosten minimieren,
- die Benutzer sind nicht die Wähler, sondern andere technische Abteilungen der Behörden, allenfalls beauftragte Vermessungsbüros,
- das Fehlen von vermessungstechnisch ausgebildeten Fachleuten und von Lehrgängen für die übrigen mit der Registrierung von Landtiteln beschäftigte Personen,
- die verschiedenen, beim heutigen Verfahren notwendigen Dienstleistungsunternehmen fördern Verbesserungen natürlich nicht, da sie dadurch lukrative Beschäftigungen verlieren könnten,
- die Gewerkschaften wehren sich manchmal gegen Veränderungen im Berufsbild.

Die Steuerbewertung (allenfalls die Title Insurance Companies) scheint mir die heute beste Grundlage für Landinformationssysteme in den USA abzugeben. In den Maritime Provinces von Kanada, die eine Reform ihres Katasters durchführen, weisen die Schätzungen des Verkehrswertes eine Fehlermarge von nur noch 5 bis 10 % auf.

Wenig Bedenken scheinen zu bestehen, die Daten eines Grundstücks-informationssystems inklusive Eigentümer, weiterzugeben. Unser Konzept der 'beschränkten Oeffentlichkeit des Grundbuches' wird abgelehnt.

Die National Science Foundation hat eine Untersuchung über Mehrzweckkataster gestartet. Ich erhielt den ersten Bericht [Jones80] und ein zweiter sei praktisch abgeschlossen. Verschiedene Studien an Universitäten wurden angeregt, insbesondere eine Untersuchung über Kosten-Nutzen, die auch für uns interessant sein wird. Diese wird an der Universität Maine in Orono durchgeführt.

Vorläufig ergeben sich folgende Prioritäten:

- Geodätische Grundlagen (Fixpunkte),
- grossmassstäbliche Grundkarte,
- Katasterpläne als Deckfolie,
- Verbindung zum Land Register durch eindeutige Parzellenkennzeichen.

Interessant waren die Ausführungen über die Behandlung von Landrutschungen im Gebiet von Los Angeles. Nach der Maxime, dass den physischen Grenzzeichen Priorität zukomme, werden die Grenzen zwischen den verschobenen Grenzzeichen rekonstruiert - d.h. das Eigentum rutscht mit dem Land mit! Das scheint vom praktischen Standpunkt aus durchaus befriedigend.

Beeindruckt hat schliesslich die Präsentation über neue technische Systeme:

Am Massachusetts Institute for Technology, Cambridge, Massachusetts, wurde ein kleines elektronisches Vermessungssystem entwickelt, das die Vermessungssatelliten benützt, ohne auf deren Ephemeriden angewiesen zu sein. Mit zwei gleichzeitig eingesetzten Geräten können ohne Sichtverbindung Koordinatendifferenzen mit einer Genauigkeit von

$$\sigma = 0.5 \text{ cm} + 10^{-5} d$$

bestimmt werden (bis ca. 100 km!). Das Gerät lässt sich leicht in einem kleinen Personenwagen unterbringen und soll etwa Fr. 50'000.- kosten (je nach Stückzahl). Eine Messung dauert etwa 2 Stunden. Eine Anzahl Bestellungen liegt vor und die Fabrikation hat begonnen.

Daneben wird an der Verbesserung des Global Positioning System (GPS) des U.S. Departement of Defense gearbeitet. Bei der NASA bestehen auch Projekte, mit einem in einem Flugzeug eingebauten Laser Distanzen zu reflektierenden Zielen auf der Erde zu messen und so deren Relativposition zu bestimmen. Auch hier scheint cm-Genauigkeit erreichbar. Antrieb zu diesen Entwicklungen ist vor allem der Wunsch, geotektonische Verschiebungszonen zu überwachen und drohende Erdbebengefahren frühzeitig zu erkennen.

8. Schlussbemerkung

Diese ausserordentlich instruktive aber auch sehr anstrengende Reise hat mir verschiedene Einblicke in die amerikanische Vermessung und Informatik erlaubt.

Erstaunlich ist im Nachhinein doch, dass Vermessung als Disziplin fast vollständig fehlt. Müssen Vermessungsprobleme gelöst werden, so werden teilweise recht primitive Methoden eingesetzt; manchmal aber auch sehr überraschende Lösungen gefunden. Das Fehlen von Fachleuten dürfte wohl vor allem auf das Fehlen entsprechender Ausbildungsstätten zurückzuführen sein, was inzwischen etwas verbessert wurde.

Mein besonderer Dank gilt der Eidgenössischen Vermessungsdirektion und der Schweizerischen Geodätischen Kommission, die neben der ETH je einen Teil meiner Reisekosten grosszügig übernommen haben.

B EUROPÄISCHE KONFERENZ ÜBER
 INTEGRIERTE, INTERAKTIVE COMPUTER-SYSTEME

1. Thema

Computersysteme werden zunehmend Bestandteil eines Arbeitsplatzes. Zuerst für Programmierer u.ä., neuerdings auch zur Unterstützung von Büroarbeitsplätzen (Schlagwort: office automation). In solchen Fällen genügt eine Sammlung von - zwar interaktiven - Programmen nicht mehr; die einzelnen Programme müssen aufeinander abgestimmt sein und zusammenarbeiten (integrierte Systeme).

Damit die Benutzer die Bedienung solcher Systeme leicht erlernen können, muss die Schnittstelle zum Benutzer - d.h. was der Benutzer sieht und was er eingibt - besser gestaltet werden, als dies in der Vergangenheit üblich war.

Die Konferenz in Stresa, die erste ihrer Art, konzentrierte sich auf diese Fragen. Eine grosse Zahl der Referenten kamen aus den USA und berichteten über die dort laufenden Forschungsarbeiten. Inhaltlich bestand eine überraschende Uebereinstimmung mit den Eindrücken, die ich von der SIGGRAPH-Konferenz gewonnen hatte. In Stresa konnten aber einige Fragen vertieft behandelt werden.

2. Gliederung des Berichtes

Statt einzelne der insgesamt 'sehr guten Referate herauszugreifen, wird eine Synthese versucht. Für zusätzliche Details ist der Leser auf die Vortragsunterlagen verwiesen, die bei North-Holland erscheinen sollen.

Ausgehend vom m.E. heute besten praktisch anwendbaren System (Xerox Star) werden eine Anzahl Prinzipien, die für die Gestaltung integrierter, interaktiver Systeme wichtig sind, herausgestellt.

Daran anschliessend werden Einzelfragen der Interaktivität besprochen.

Die heute wichtigste Anwendung integrierter Systeme ist die Entwicklung von Programmen; hier kann nur auf groben Linien der Forschung hingewiesen werden, da die Ergebnisse leider meist noch nicht als anwendbare Produkte erhältlich sind.

Der Schluss bildet eine persönlich gefärbte Beurteilung des Gebotenen im Hinblick auf die Anwendung bei uns.

3. Xerox Star

Xerox Star ist ein Arbeitsplatz Computer der oberen Preisklasse mit speziellen Programmen für die Arbeit im Büro, die ja vor allem die Verarbeitung von Texten im weitesten Sinne umfasst. Vertrieben wird er in der Schweiz unter der Bezeichnung Xerox 8010 zum Preise von Fr. 40'000.--. Bemerkenswert am Xerox Star ist die Schnittstelle zum Benutzer, die ausserordentlich einfach und klar aussieht. [Newman82] hat über ein Vorgängersystem, OFFICETALK, berichtet, das mit geringerem Hardwareaufwand Beachtliches erreicht. Die hier erklärten Aspekte können niemals den gleichen überzeugenden Eindruck erwecken, wie eine Präsentation. Meines Erachtens ist mit der Benützerschnittstelle hier ein wichtiger Durchbruch gelungen, Vergleichbares ist mir nicht bekannt.

3.1 Prinzipien der Benützerschnittstelle

Die Benützerschnittstelle umfasst alles, was der Benutzer von einem EDV-System sieht, also die Art der Darstellung von Anforderungen zum Handeln und die Präsentation der Ergebnisse, aber auch die Art der Eingabe vom Benutzer. Im Xerox Star wurde die Benützerschnittstelle sehr sorgfältig, und bevor das Programm geschrieben wurde, entworfen. Dabei waren folgende Feststellungen wegleitend:

für Menschen

<u>leicht</u>	<u>schwierig</u>
konkret	abstrakt
sichtbar	unsichtbar
auswählen	neu schaffen
erkennen	nennen
kopieren und ändern	generieren
verändern	programmieren
interaktiv	Stapelverarbeitung

Man bemühte sich, dem Benutzer ein sichtbares elektronisches Büro zu präsentieren, das möglichst genau mit seiner bisherigen konkreten Erfahrung übereinstimmt (bis und mit den Begriffen: Dokument, Mäppchen, Ablage-Schrank). Alle Objekte sind sichtbar, wie auf einem Schreibtisch liegend.

Der Benutzer kann die sichtbaren Objekte auswählen, so dass ihr Inhalt sichtbar wird. Er kann aber auch die normalerweise unsichtbaren Eigenschaften der Objekte (Schriftgrösse, Schrifttyp etc.) sichtbar machen und dann ändern - was sofort auch zur Aenderung des Objektes führt.

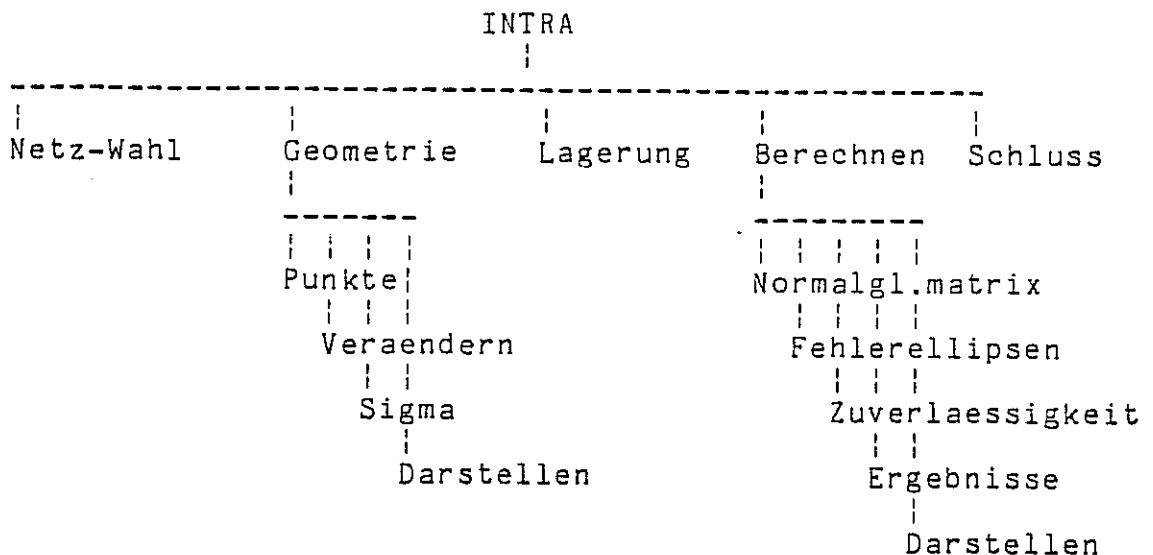
Statt von grundauf Neues zu schaffen, wählt der Benutzer besser ein irgendwie ähnliches, schon existierendes Objekt (z.B. ein leeres Formular vom Vorratsstapel), das er kopiert und dann verändert, - alles Aktionen mit sofort sichtbarem Ergebnis.

Nach der Aehnlichkeit mit der bekannten konkreten Umgebung und dem Sichtbarkeitsprinzip ('sie erhalten was sie sehen'), ist die Gleichartigkeit (engl. uniformity und consistency) entscheidend. Zwei Beispiele: Die Maus, die für die Bewegung des Fadenkreuzes auf dem Bildschirm verwendet wird, hat zwei Knöpfe, die überall die gleiche Bedeutung haben. Die generellen Befehle MOVE, COPY, DELETE (spezielle Tasten) etc. können in verschiedenem Zusammenhang verwendet werden und ersetzen so ein Dutzend Kommandos, die der Benutzer in traditionellen Systemen lernen muss, nämlich, welcher Befehl führt die MOVE-Funktion im Editor aus, welcher beim Senden von Nachrichten, welcher beim Uebertragen von Dateien etc. Gleichartigkeit verlangt aber auch, dass bestimmte Methoden der Arbeit von einer Arbeitsumgebung in eine andere übertragen werden können - der Benutzer soll möglichst nicht überrascht werden.

Gleichartigkeit bezieht sich schliesslich auch auf den Aufbau der Befehle, die immer als Substantiv (Objekt, das mit dem Cursor bezeichnet wird) und Verb (gewünschte Aktion durch Knopfdruck ausgewählt) in dieser Reihenfolge bestehen.

3.2 Modus - ohne Modus

Für die Gestaltung von Benützerschnittstellen ist der Begriff Modus (mode) [Nievergelt80] sehr wichtig. Ein Benutzer - bzw. das von ihm benützte System - befindet sich in einem Modus, sofern er nur bestimmte der insgesamt angebotenen Möglichkeiten zu Aktionen benutzen kann. Die Selektion von Aktionen durch Hierarchien von Befehlen (z.B. in INTRA) ist ein typisches



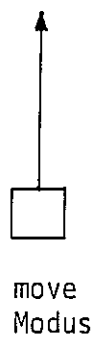
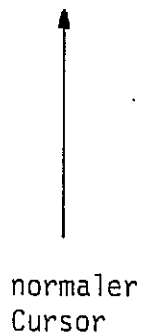
Kennzeichen von Modus (jede Verzweigungsstelle bildet einen Modus). Aber auch in Editors treten Modi auf (TECO: Nach I ist man im insert-mode - alles was man eintippt, wird in den Text eingefügt, bis man mit ESCAPE wieder in den Befehlsmodus zurückkehrt).

Modi sind für das Verständnis der Vorgänge durch den Benutzer bekanntermassen problematisch. Soweit sie vermeidbar sind, sollte man darauf verzichten.

Xerox Star benützt folgende Methoden: Die Operationen beziehen sich immer auf die sichtbaren Objekte - je nach Position des Cursors verändert sich also der Modus. Ist man im Textfenster, so wirken die Befehle zum editieren des Textes, ist man bei den übrigen Objekten, z.B. bei den Dateien, so haben die gleichen Befehle andere aber ähnliche Wirkungen auf diese.

Das scheint nicht zu verwirren, weil es unserer konkreten Anschauung entspricht.

Für gewisse Operationen sind Modi nicht zu vermeiden. Nach dem Befehl 'file x' 'MOVE' muss unbedingt ein Ort bezeichnet werden - eine andere Aktion kann nicht vorgenommen werden, was typisch für einen Modus ist. Dieser Modus wird dem Benutzer sichtbar gemacht, indem die Form des Cursors ändert (er zeigt ein am Cursor-Pfeil angehängtes Objekt) und ihn daran erinnert, dass er ein Ziel für den MOVE angeben muss.



4. Object-oriented approach (SMALLTALK

Object-oriented approach bezeichnet eine bestimmte Auffassung: Programme (in einem sehr weiten Sinne) sind Objekte, die miteinander Nachrichten austauschen - damit kann man menschlichen Organisationsformen (antropomorphie [Beach82]) extrem nachgebildete Programmstrukturen beschreiben.

Damit ist m.E. nicht eine Lösung für bisher unlösbare Probleme gefunden - man kann schliesslich alles in Assembler programmieren - sondern es werden dem Programmierer bessere, einfacher vorstellbare Primitive angeboten. Programme können schneller erstellt werden, was oft erst die Lösung schwieriger Probleme erlaubt.

Die folgende Diskussion stützt sich auf die bekannteste object-oriented Programmiersprache, SMALLTALK von XEROX PARC [Goldberg81]. Erst zum Schluss werden andere ähnliche Systeme erwähnt.

4.1 Objekte und Nachrichten

Alles was in einem SMALLTALK-System existiert, sind Objekte.

Objekte können als Implementierung von abstrakten Datentypen aufgefasst werden; das ist eine Uebertragung des Konzepts der abstrakten Datentypen (ADT) vom Entwurf in die Implementierungsphase, was einen menschlichen, fehleranfälligen Uebersetzungsvorgang vermeiden.

Objekte bestehen aus Informations - Paketen (Daten) und Methoden zu deren Verarbeitung. Sie enthalten Variablen (es gibt keine globalen Variablen ausserhalb von Objekten) und bieten den Klienten (benützendes Programm im Gegensatz zu einem menschlichen Benutzer [Lampson 81]) Anlaufpunkte für Operationen (Methoden). Jeder Methode ist eine Nachricht (message) zugeordnet, auf die das Objekt antwortet.

Objekte verkehren untereinander ausschliesslich durch den Austausch von Nachrichten. Muss ein Objekt den Wert einer Variablen in einem anderen Objekt erfahren, so sendet es diesem eine entsprechende Nachricht und erhält den Wert als Antwort zurück. Gleich erfolgt aber auch das Auslösen einer Aktion eines anderen Objektes (z.B. um etwas auf den Schirm zu schreiben).

Der Austausch von Nachrichten übernimmt hier die Aufgabe des Aufrufes von Subroutinen in PASCAL oder FORTRAN.

Aber auch Daten -Strukturen können durch Objekte und Nachrichten aufgebaut werden und schliesslich können Nachrichten auch zwischen parallel laufenden Prozessen ausgetauscht werden.

4.2 Vererbung (Inheritance)

Objekte sind Ausprägungen (instantiations) von Klassen, genauso wie PASCAL Variablen Ausprägungen von Typen sind.

Der Programmierer erstellt in SMALLTALK eigentlich Klassen, von denen er nachher beliebig viele Objekte erzeugen kann (was man z.B. mit PASCAL-Prozeduren nicht ohne weiteres kann, ganz abgesehen davon, dass PASCAL-Prozeduren keine statischen (oder own) Variablen kennen).

Der Programmierer erstellt aber nie eine Klasse ganz neu, sondern die Klassen sind (in SMALLTALK) in einer Hierarchie angeordnet. Ist eine neue Klasse, ein neuer abstrakter Datentyp, für eine spezielle Art von Zahlen (z.B. die Restklasse 400 in Real, um Azimute zu verarbeiten) erforderlich, so erzeugt er diese als Subklasse der Klasse 'Real', die bereits existiert; die neue Klasse hat damit schon alle Eigenschaften der Klasse 'Real' und aller Oberklassen davon geerbt. Das bedeutet, dass beispielsweise Methoden für das Drucken dieser neuen Objekte nicht mehr spezifiziert werden müssen (sondern eben eben geerbt, inherited, sind).

Dieser Mechanismus erleichtert die Arbeit in verschiedenster Hinsicht:

- Programmieren ist 'refinement' von existierenden Klassen, alles Brauchbare wird automatisch übernommen.
- Einheitlichkeit der Programmierung in einem Team ergibt sich durch die mehrfache Verwendung des gleichen Codes.
- Die Fehler suchen ist vereinfacht, in dem alles eine Ausprägung der Superklasse 'Klasse' ist, die bestimmte Methoden (z.B. um ein Objekt sichtbar zu machen) enthält und allen Objekten weitergibt.

- Jede Prozedur wird einzeln übersetzt und geladen;
Prozeduren können mit minimaler Umgebung geprüft und
Programme während des Ablaufes geändert werden.

Am wichtigsten scheint mir aber, dass die stufenweise Abstraktion, wie sie in dieser Klassen-Struktur zum Ausdruck kommt, dem menschlichen Denken entgegenkommt (ein Möbel → ein Tisch → ein Schreibtisch).

4.3 Andere Ansätze

Neben SMALLTALK wurde ein auf LISP basierendes Programmsystem ORBIT mit ähnlichen Eigenschaften vorgestellt [Steels 82]; dieses erlaubt, dass für eine Klasse Eigenschaften selektiv von verschiedenen Klassen geerbt werden (Beispiel Spielzeuglokomotive erbt von Spielzeug Grösse, Gewicht, von Lokomotive die Form).

Die Universität Waterloo benützt ein ähnliches System, das auf der Programmiersprache B (Vorgänger von C) beruht [Cheriton79]; es wird u.a. erfolgreich für die Programmierung interaktiv-graphischer Systeme eingesetzt.

4.4 Beurteilung

Die objekt-orientierten Sprachen scheinen vielversprechend, weil sie direkte Implementierungen von abstrakten Datentypen erlauben. Besonders wichtig ist die Möglichkeit, Hierarchien, von Objekten zu bilden, die gemeinsame Eigenschaften aufweisen.

Schliesslich ist die Möglichkeit des parallelen Ablaufes für die angepasste Gestaltung von Benützerschnittstellen (siehe Abschnitt 7) in gewissen Fällen Voraussetzung.

Nachteilig scheint die grosse Belastung der Hardware (Rechenleistung); es ist unklar, ob mit SMALLTALK heute bereits Systeme für produktive Aufgaben traditioneller Art erstellt werden können.

5. Unterstützung der Programmierung

Die Unterstützung der Programmierarbeit ist naturgemäss ein erstes Einsatzgebiet für viele dieser Ideen; manchmal erstaunt es fast, wie wenig Programme auf diesem Gebiet vorhanden sind! Programmierer haben offensichtlich gelernt, mit dem Vorhandenen auszukommen und wehren sich, wie viele andere potentielle EDV-Benützer, gegen neue Systeme (PASCAL, ADA, ...).

Die grundsätzliche Idee ist eine Programmierungsumgebung zu schaffen, die ein problemloses und integriertes Erstellen, Testen und Aendern von Programmen erlaubt. Verschiedene erfolgreiche Beispiele existieren: BASIC, LISP, SMALLTALK. Hier werden zuerst einige Einzelfragen untersucht, um nachher die Integration zu diskutieren.

5.1 Strukturierte Editor

Die von uns zur Erstellung von Programmen verwendeten Editoren (TECO, SED, etc.) sind normale Text-Editoren, mit denen beliebige Texte und damit auch Programme eingegeben und verändert werden können. Es wird nun versucht, sprachspezifische Editoren zu entwickeln, die das Eintippen von syntaktisch unkorrekten Programmen verunmöglichen (z.B. nach BEGIN wird automatisch END gesetzt) und gleichzeitig das entstehende Programm sauber darstellen (sog. prettyprinter - z.B. unser PRETTY für PASCAL) [Teitelbaum81].

5.2 Programmieren durch Beispiel

Programmieren erfordert ein beträchtliches Abstraktionsvermögen. Es wird nun versucht, den Benutzer einige Beispiele eintippen zu lassen und dann die abstrakte Regel automatisch finden zu lassen, allenfalls muss der Benutzer nachher noch den Unterschied zwischen den Fällen angeben [Liebermann82].

Diese Systeme sind erst im experimentellen Stadium. Anwendbare Systeme für einfache, beschränkte Anwendungsbereiche hingegen scheinen einsatzbereit [Zloof77].

5.3 Automatische Programmerstellung

Von [Smoliar82] wurde von einer Anwendung berichtet, bei der es für ein eng begrenztes Anwendungsgebiet (Interpretation von Bohrlochmessungen bei Schlumberger) gelang, grosse numerische Analyseprogramme automatisch auf Grund mathematisch formaler Spezifikation zu erstellen.

5.4 Schrittweise (inkrementale) Kompilation und Unterstützung des Testens

Zusammen mit dem strukturierten Editor, der einen Teil der Arbeit eines Compilers ausführen muss, wird versucht, den Programmierer beim Testen so zu unterstützen, dass er:

- Prozeduren ohne neues Laden ändern kann (möglichst während dem Lauf des Programmes).
- Code, Ausführungsmittelungen und interne Daten auf dem gleichen Schirm sieht und gleichzeitig manipulieren kann (vgl. Modula-2 auf Lilith [Wirth82]).
- Programme auch laufen lassen kann, wenn sie noch nicht vollständig sind.
- Programme Schritt um Schritt (möglichst auch rückwärts) ausführen kann.

5.5 Versionskontrolle

Wird ein grösseres Softwarepaket weiterbearbeitet, so besteht eine ausserordentlich aufwendige Aufgabe darin, die verschiedenen, nacheinander entstehenden und nebeneinander benutzten Versionen zu verwalten und in der richtigen Kombination zusammenzustellen. Programme sollen dies erleichtern [Tichy82].

Einerseits werden Differenzen zwischen verschiedenen Versionen automatisch bestimmt und diese verwaltet, so dass beliebige Versionen erstellt werden können. Andererseits muss festgehalten werden, welche Versionen welcher Moduln zusammengehören.

5.6 Anmerkungen zu Programmen

Die Kommentare zu Programmen könnten stärker strukturiert werden und auch Invarianten und Beispiele umfassen, die bei der Ausführung auf Wunsch berücksichtigt werden. Es könnte also bei der Ausführung automatisch geprüft werden, ob das Beispiel vom Programm richtig verarbeitet wird [Wert82].

5.7 Integration

Verschiedene der vorgeschlagenen Hilfsmittel existieren schon längst, verschiedene sind erst im Experimentierstadium. Nicht unterschätzt werden darf die Schwierigkeit, solche Teile zu einem integrierten System zusammenzubinden. Einerseits sind die internen Probleme zu lösen (Datenstrukturen, Betriebssysteme usw.), andererseits muss ein solch komplexes System eine einfache, einheitliche Benützerschnittstelle aufweisen (siehe Xerox Star).

Prinzipiell scheint dies für Programmiersprachen mit einer einfachen Struktur (SMALLTALK - alles ist ein Objekt; LISP - alles ist eine Funktion oder eine Liste) einfacher.

6. Anwendungen von AI

AI (artificial intelligence, automatisches Schliessen) beschäftigt sich im Moment stark mit der Erfassung von Beziehungen zwischen Begriffen, wie sie in natürlichen Sprachen vorkommen. Ziel ist ja nicht ein Volk von BASIC-sprechenden Benützern, sondern eher ein Computer, der die Sprache des Anwenders versteht. Verschiedene Beiträge haben sich mit Anwendungen solcher Methoden auf Probleme der Interaktion zwischen Benutzer und Maschine beschäftigt.

[Vittal82] zielt auf Informationssysteme, die anhand einer Darstellung der Beziehungen zwischen den Begriffen des Anwenderprogrammes einerseits und den Begriffen des Benützers andererseits Gemeinsamkeiten feststellen und gestützt darauf sehr gezielte Hilfe anbieten.

[Bates82] berichtet über eine Untersuchung über typisches Verhalten von Menschen beim Dialog mit einem Computer. Das soll zu einer klaren Liste von Anforderungen an einfachen natürlichen sprachlichen Schnittstellen zwischen Mensch und Maschine führen, die sich leichter implementieren lässt.

7. Synchrone oder asynchrone Benützerschnittstelle

Benützerschnittstellen können nach verschiedenen Prinzipien aufgebaut werden:

- Dialog (Frage - Antwort). Jeder Benützereingabe geht eine entsprechende Aufforderung voraus; der Benutzer wird geführt und kann jeweils nur auf die gestellte Frage antworten. Dieses Verfahren ist geeignet für Abläufe, die in fester Reihenfolge durchgespielt werden müssen und bei denen der Benutzer von dieser Reihenfolge nicht abweichen soll. Solche Dialoge können je nach Antwort verzweigen und sich damit der bearbeiteten Aufgabe anpassen, der Benutzer bleibt aber prinzipiell passiv - die Aktionen gehen vom Programm aus.
- Menu. Der Benutzer kann den Ablauf frei aus einer Anzahl angebotenen nächsten Schritten auswählen. Nach Ablauf des Schrittes kann erneut gewählt werden. Damit die Auswahl überblickbar bleibt, müssen die vielen möglichen Schritte in einer Hierarchie gegliedert werden, was zu Modi führt. Die Menus können je nach Bearbeitungsfortschritt angepasst werden [Frank79] [Conzett80] und allgemeine Methoden zur Verwaltung wurden entwickelt [Meyer82].
- Befehlssprache. Statt einem sichtbaren Menu kennt der Benutzer eine Anzahl Kommandi, die er auf die Aufforderung des Systemes eingeben kann.

Allen diesen bekannten Verfahren ist gemeinsam, dass Programm und Benutzer synchron arbeiten. Das Programm fragt (evtl. ohne ausführliche Aufforderung), der Benutzer antwortet und das Programm führt etwas aus, worauf sich das Spiel wiederholt. Eingaben von Benutzern sind nur möglich, wenn das Programm dazu bereit ist. Dies ist eine fühlbare Einschränkung, sobald einzelne Arbeitsschritte (Modi) länger dauern und der Benutzer diese unterbrechen möchte, um zwischendurch eine andere Tätigkeit auszuführen (was dem normalen menschlichen Arbeitsablauf eher entspricht).

Wünschbar ist in solchen Fällen eine asynchrone Arbeitsweise, bei der der Benutzer zu einem beliebigen Zeitpunkt ein laufendes Programm vorübergehend unterbrechen kann, um etwas anderes zu tun, worauf das erste Programm weiterläuft.

Unumgänglich wird diese Möglichkeit zur Steuerung, wenn verschiedene Eingabemittel (Tablet, Tastatur usw.) zur Verfügung stehen, die in beliebiger Reihenfolge betätigt werden können.

Statt zu versuchen, diese asynchronen Elemente in eine an sich synchrone Benützerschnittstelle zu integrieren (mit entsprechenden Schwierigkeiten bei der Programmstruktur) kann, wenn man das Prinzip einmal erkannt hat, ein Monitor gebaut werden, der anwendungsunabhängig eine entsprechende Steuerung übernimmt.

8. Folgerungen für die Arbeit am IGP

8.1 Bildschirmverwaltung

Die Verwendung verschiedener gleichzeitig sichtbarer Informationsfenster bietet gegenüber dem üblichen über den Bildschirm wandernden Strom vom Zeichen mehr Komfort für den Benutzer: er weiss, was er wo sehen kann. Informationsfenster, die im Moment nicht benötigt werden, können auf Erinnerungsfelder minimaler Grösse reduziert werden, die jederzeit geöffnet werden können [Newman82].

Die Anwendung dieser Verfahren für DATAUF und LIS-Prototyp ist zu prüfen. Allenfalls könnte uns die Uni Stuttgart ein Programmpaket für unsere Bildschirme (BYTEMAP) zur Verfügung stellen.

8.2 Benützerschnittstelle

Bei der Gestaltung neuer Benützerschnittstellen sollte man sich von ähnlichen Ueberlegungen, wie Xerox beim Star lenken lassen. Insbesondere scheint mir die Sichtbarkeit aller Objekte und Operationen wichtig.

8.3 Asynchrone Benützerschnittstelle

Für die Integration von Grafik (insbesondere Tablet und Cursor) ist es notwendig, dass der Benutzer handeln kann, ohne auf die Aufforderung durch das Programm zu warten (z.B. Bewegungen des Cursors müssen ständig nachgeführt werden). Der dazu notwendige Monitor sollte am IGP weiter untersucht werden.

8.4 Objekt-orientierte Programmierung

Programme, bei uns vornehmlich in PASCAL erstellt, sollten abstrakte Datentypen (Objekte) darstellen; die Modularisierung ist auf dieses Ziel auszurichten.

Schwierig ist die Entscheidung, wie weit dazu Elemente einer Pascal-Implementation, die nicht dem Standard entspricht, benützt werden dürfen, ohne dass bei der Uebertragung auf eine andere Anlage Schwierigkeiten entstehen.

Programmierrichtlinien, die diese Gesichtspunkte betonen, sollten erarbeitet werden.

8.5 Programmierungsumgebung

Bei uns sind schon eine Anzahl Hilfsmittel im Einsatz (screen editor, prettyprinter, compiler und debugger), die sehr wertvoll sind. Ein weiterer Ausbau (MACRO Prozessor nach [Kernighan81]) scheint sinnvoll. Eine Integration ist für uns leider mit vernünftigem Aufwand nicht zu erreichen.

LITERATURVERZEICHNIS

- Anson, E. The device model of interaction
SIGGRAPH '82 proceedings
Computer Graphics, Vol 16 (3) July 1982
- Bates, M.
Sidner, C.L. A Case Study of a Method for Determining
the Necessary Characteristics of a Natural
Language Interface
in: ECICS 82
- Beach, R.J.
Beatty, J.C.
Booth, K.S.
Plebon, D.A.
Finme, E.L. The message is the medium:
Multiprocess Structure of an interactive
paint program
SIGGRAPH proceedings '82
Computer Graphics, 16 (3) July 1982
- Bolt, R.A. Put-That-There:
Voice and gesture at the Graphics Interface
SIGGRAPH '80 Proceedings
Computer Graphics 14 (3) July 1980
- Cheriton, D.R.
Malcolm, M.A.
Melen, L.S.
Sager, G.R. Thoth a portable real-time operating system
Comm. ACM, Vol 22, No 2, 1979
- Clark, J.H. The Geometry engine:
a VLSI geometry system for Graphics
SIGGRAPH '82 proceedings
Computer Graphics, Vol 16 (3) July 1982
-
- CM82 Eroberungstaktiken in der Schlacht um das
PC-Areal
PC-Marktstudie von Strategie Inc.
Computer Magazin 7/82
-
- Conzett, R.
Frank, A.
Misslin, C. Interaktive Triangulation
VIII Internat. Kurs für
Ingenieurvermessung Zürich 1980
- Crow, F.C. A more flexible image
Generation environment
SIGGRAPH '82 proceedings
Computer Graphics, Vol 16 (3) July 1982
- ECICS82
ACM Italian Chapter European Conference on Integrated
Interactive Computing Systems
- Frank, A. Gestaltung interaktiver Programme
RZ Bulletin der
ETH Zürich, April 1979

- Frank, A. MAPQUERY
Data base query language for retrieval of
geometric data and it's graphical representation
SIGGRAPH 1982 Proceedings
Computer Graphics Vol.16 No.3 July 1982
- Frank, A. PANDA A Pascal Network Database Management
System
SIGSMALL conference 1982
Colorado Springs August 1982
- Friedel, M. Context-sensitive,
Barnett, J. Graphic presentation of information
Kramlich, D. SIGGRAPH '82 proceedings
Computer Graphics Vol.16 (3) July 1982
- Goldberg, A. The Smalltalk-80 System
Ingalls, D.H.H. BYTE, Vol. 6, No. 8, Aug. 1981
- Guttag, J. Abstract Data Types and the Development of
Data Structures
Comm. ACM Vol. 20 No. 6 June 1977
- Guibas, Leo, J. A Language for bitmap manipulation
Stolfi, Jorge ACM, Transactions on Graphics
Vol. 1, No. 3, July 1982
- Herot, C.F. Graphical Input through Machine
Recognition of Sketches
Computer Graphics 10 (2), Summer 1976
- Jones, G.E. Need for a Multipurpose Cadastre
et al. Panel on a multipurpose Cadastre
National Research Council
National Academy Press
Washington D.C. 1980
- Kasik, D.J. A user interface management system
SIGGRAPH '82 proceedings
Computer Graphics, Vol. 16 (3) July 1982
- McKeown, D.M. Graphical Tools for Interactive Image
Denlinger, J.L. Interpretation
SIGGRAPH '82 proceedings
Computer Graphics, Vol. 16 (3) July 1982
- Kernighan, B.W. Software Tools in Pascal
Plauger, P.J. Addison-Wesley, 1981
- Lampson, B.W. Distributed Systems
Paul, M. Architecture and Implementation
Siegert, H.J. Lectures Notes in Computer Science 105
Springer Verlag, Berlin, 1981

- Liebermann, H. Designing Interactive Systems
 From the User's Viewpoint
 in ECICS 82
- Lipkie, D.E.
Evans, S.R.
Newlin, J.K.
Weissmann, R.L. Star Graphics
 An object-oriented implementation
 SIGGRAPH '82 proceedings
 Computer Graphics, Vol. 16 (3) July 1982
- Meyer, B. Towards a Two-Dimensional
 Programming Environment
 in: ECICS 82
- Newman, W.M.
Mott, T. OFFICETALK-Zero
 An Experimental Integrated Office System
 in: ECICS 82
- Nievergelt, J.
Weydert, J. Sites, modes and traits
 telling the user of an interactive system
 where he is, what he can do and how to
 get to places
 in Guedj
- Parnas, D.L. On the Criteria to be used in
 Decomposing Systems into Modules
 Comm. ACM, Vol. 15, No. 12, Dec. 1972
- Rosenthal, D.S.H.
et al. The detailed semantics of graphics
 input devices
 SIGGRAPH '82 proceedings
 Computer Graphics, Vol. 16 (3) July 1982
- Senko, M.E. DIAM II with FORAL LP: Making Pointed
 Queries with Light Pen, Information
 Processing 77, North-Holland Publ. Co.,
 Amsterdam, 1977, p. 635
- Smoliar, S.W. An Interactive Approach
 to Software Specification
 in: ECICS 82
- Steels, L. An Applicative View of object-oriented
 Programming
 in: ECICS 82
- Teitelbaum, T. The Cornell Program Synthesizer
 A Syntax Directed Programming Environment
 Comm. ACM, Sept. 1981, Vol. 24, No. 9
- Thurnherr, F. Die Besteuerung von Schweizerischen
 Investitionen in Amerikanisches Grundeigentum
 Zürcher Studien zum öffentlichen Recht 17
 Schulthess Polygraphischer Verlag Zürich 1980

- Tichy, W.F. RCS
A Revision Control System
in: ECICS 82
- Vittal, J.
Sidner, C.L. Knowledge Representation Tools for the
Design, Training and Use of Information Systems
in: ECICS 82
- Wertz, H. An Integrated, interactive and incremental
Programming Environment for the Development
of Complex Systems
in: ECICS 82
- Wirth, N. The Personal Computer Lilith
Institut für Informatik, ETH Zürich,
Bericht Nr. 40, Zürich 1981
- Wirth, N. Programming in Modula-2
Springer-Verlag Berlin 1982
- Wong, P.C.S. FLAIR - User Interface dialog design tool
SIGGRAPH '82 proceedings
Computer Graphics, Vol. 16 (3) July 1982
- Zehnder, C.A. -Informationssysteme und Datenbanken
Verlag der Fachvereine, Zürich 1981
- Zloof, M. Query by example
AFIPS Conference Proceedings, 1975 National
Computer Conference, Vol. 44, AFIPS Press,
1975, p. 431