

Zunächst als ein etwas teures „Spielzeug“ belächelt, hat seit 1984 der kleine Apple Macintosh Computer durch das Setzen neuer Paradigmen die gesamte Computerwelt grundsätzlich zum Besseren verändert – und wie [1–2].

Viele Benutzer von Personal Computern (PCs) wissen überhaupt nicht, wem sie die heutige Ausstattung ihres PCs [3] verdanken. Zwar gilt der Macintosh Computer als die Antwort Apples auf den am

12. August 1981 von IBM als ein „easy-to-use System“ für 1.565 \$ eingeführten PC, dennoch bot für 2.495 \$ der Macintosh bereits bei seiner Einführung am 24. Januar 1984 wesentlich mehr „Computing“ als der IBM PC. Vor allem, er war (und ist) ein *easy-to-use System*.

Es wird immer wieder vom „PC-Standard“ gesprochen. Eine solche Norm gab es aber eigentlich nie. Was es gibt, ist eine sehr unvollständige Beschreibung der IBM

von 1984 und ein Sammelsurium von Ergänzungsvorschlägen verschiedener Firmen, die PC-Clones bauten [6]. Also so ein ziemliches Chaos bei der Hard- und Software, was sich bis heute auswirkt [4].

Hingegen ist der Macintosh bis hin zum aktuellen PowerMac [5] ein, zwar firmeneigenes, aber technisch sehr klar und sauber definiertes Standardprodukt, dessen wesentlichsten Merkmale und Schnittstellen publiziert sind.

Die 20 Macintosh Paradigmen

Ein kurzer Rückblick auf all die innovativen Pionierleistungen der Firma Apple

Es war der Macintosh Computer, der den Weg zum modernen, benutzerfreundlichen Arbeitsplatzcomputer bahnte. Viele Merkmale eines Macintosh werden bei den PCs heute nun auch als selbstverständlich angesehen. Bei anderen Merkmalen – z. B. dem „Plug & Play“ – ist der PC allerdings noch immer Meilen vom gegenwärtigen Stand des Macintosh entfernt.

1 Graphical User Interface

Zwar hat die Firma Apple Computer die grafische Benutzeroberfläche eines Computers (GUI) nicht selbst erfunden, dennoch war es Apple, die 1983 erstmals dieses vom Xerox Palo Alto Research Center (PARC) entwickelte benutzerfreundliche Konzept im noch 10.000 \$ teuren Lisa Computer – dem Vorläufer des Macintosh – kommerziell anwandte.

Zum GUI gehört, daß *alle* Bildschirmausgaben als hochauflösende „bit-mapped“ Grafik erfolgen, auch beim Text. Erst damit wurde das WYSIWYG-Prinzip (What-You-See-Is-What-You-Get) bei Anwendungsprogrammen möglich. Das war die Voraussetzung zur Entwicklung von intelligenten, leistungsfähigen Grafik- und Desktop-Publishing-Programmen.

2 Point & Click

Das GUI-orientierte Macintosh-Betriebssystem (MacOS) arbeitet ausschließlich mit Objekten, wie Fenstern (windows) mit Rollbalken (scroll bars), Symbolen (icons) und Menüs, die vom Computerbenutzer per Mauszeiger ausgewählt und per Maustaste angeklickt oder bewegt werden können.

Auch alle Anwendungsprogramme für GUI-Systeme sind immer reine *ereignisgesteuerte* (event-driven) Software, die eine völlig andere Vorgehensweise beim Programmieren sowie moderne Programmiersprachen (z. B. Pascal, Modula-2, C++) erfordern. Vom Programmierer werden nur Versatzstücke (Tasks) zur Verfügung gestellt. Erst der Computerbenutzer verbindet diese Stücke durch seine Handlungen (Point & Click) zum Gesamtprogramm. Der Programmierer muß sich daher bei der Entwicklung in all die möglichen Vorgehensweisen von Benutzern hineinendenken.

3 Point & Shoot

Das MacOS beruht u. a. darauf, daß die Auswahl der Menü-Befehle, aber auch der Informationsfluß zu den Rollbalken und anderen Objekten durch die Triggerung von Geräten (z. B. durch den Mausklick) irgendwann, nicht im Programm vorher bestimmbar – also absolut *asynchron* ausgelöst wird.

Dieses Prinzip des „Point & Shoot“ ermöglichte es, eine wirklich benutzerfreundliche Software zu entwickeln, die ein vollkommen intuitives Arbeiten mit einem Computer erlaubt. Erst dadurch wurde der Computer für den Menschen zu einem echten Werkzeug.

Die programmtechnische Behandlung derartiger asynchroner Systeme ist bereits seit den 60er-Jahren (!), vor allem durch die Arbeiten des Finnen Dijkstra [7], theoretisch geklärt. Zwar wurden diese wichtigen Ideen auch schon bei der Behandlung von Interrupts durch Peripheriegeräten in Gerätetreibern (Handlern) früherer Computer verwendet, aber es ist das große Verdienst eines Steve Jobs erstmals all diese Prinzipien konsequent bei einem kommerziellen Produkt – dem Apple Macintosh Computer – eingeführt zu haben.

4 The Clipboard

Eine der besten Ideen, die Apple in verbesserter Form vom Xerox PARC übernahm, war die Einführung einer Zwischenablage (clipboard) für Text- und Grafikobjekte. Diese Zwischenablage wurde als Datei mit spezieller Struktur realisiert. Sie erlaubt auch das automatische Konvertieren zwischen Grafikformaten (PICT, TIFF). Da die Zwischenablage aber nur ein Objekt speichern kann, stellt das MacOS außerdem das *Album*, ebenfalls eine Platten-

datei, zum Speichern mehrere Objekte per „Cut & Paste“ zur Verfügung.

5 Cut & Paste

Auf der Basis dieser Zwischenablage wurde es nun möglich, in einem Fenster eine Grafik auszuschneiden (cut), d. h. in die Zwischenablage zu kopieren, und dann diese Grafik in einem anderen Fenster (Dokument), z. B. in einer Textseite wieder einzufügen (paste).

6 Undo

Das MacOS führte die Möglichkeit des Rückgängigmachens (undo) des zuletzt ausgeführten Befehls ein. Viele Anwendungsprogramme erlauben heute sogar das „Undo“ von mehreren Befehlen.

7 Mac Toolbox – The first API

Das zentrale Geheimnis eines jeden Macintosh ist seine „Toolbox“. Das sind System-ROMs mit einer sehr umfangreichen Bibliothek von optimierten Basisroutinen für alle Macintosh-Funktionen, die jedem Anwendungsprogrammierer zur Verfügung stehen [8], nur Microsoft mag sie nicht so recht! Erst durch die konsequente Verwendung dieser Toolbox-Routinen erhalten alle Macintosh-Anwendungen ihr gleichartiges Erscheinungsbild.

In der Mac Toolbox sind die Routinen in Gruppen, den Managern zusammengefaßt, z. B. QuickDraw (erzeugt alle Grafik), der Device Manager, der File Manager, der Window Manager, der Event Manager, der Text Manager und der Font Manager (weitere siehe in [9]). Beim ersten Macintosh, dem Mac 128 war die Toolbox noch 64 kByte groß, inzwischen ist sie bei den neueren Macs auf 1 MByte (bei den AV-Macs auf 2 MByte) angewachsen [10].

Es muß einmal sehr deutlich gesagt werden: Ohne die von Niklaus Wirth 1971 entworfene wegweisende höhere Programmiersprache *Pascal* [11], mit der erstmals Datentypen und -strukturen vom Programmierer frei definiert werden konnten, ist der Macintosh Computer nicht vorstellbar. Die ersten PCs enthielten hingegen in ihren ROMs lediglich BASIC! Mit BASIC oder aber mit FORTRAN konnte man niemals die raffinierten Datenstrukturen, wie sie für die Toolbox benötigt wurden, beschreiben.

Die Mac Toolbox ist aus heutiger Sicht das erste realisierte „Application Programmiers Interface“ (API) eines Computers überhaupt. Der Zugang zu den Toolbox-Routinen erfolgt über Sprungadressen im Hauptspeicher (RAM), so daß es mit RAM-Patches leicht möglich ist, Verbesserungen an den Toolbox-Routinen vorzunehmen.

8 Dynamic Memory Allocation

Um dem Memory Manager der Toolbox das Laden von Programmen in beliebige Bereiche des Hauptspeichers zu ermöglichen, verwenden alle Macintosh-Programme von der Stunde Null an die *relative Adressierung* (beim PC wurde die klassische absolute Adressierung verwendet).

Das war 1983 eine sehr weitsichtige Entscheidung von Apple, da solche Programme unverändert in einer Multitasking-Umgebung oder bei einer virtuellen Speicher-verwaltung lauffähig bleiben. Auch benötigen solche Programme wesentlich weniger Speicherplatz als z. B. die Windows-Anwendungen in der PC-Welt.

9 Plug & Play Design

Das Design des Macintosh verwendet seit der Einführung des „offenen“ Mac II im März 1987 konsequent das benutzerfreundliche Prinzip des „Plug & Play“, mit dessen Entwicklung sich die PC-Welt noch heute herumquält und dafür noch immer keine überzeugende Lösung anbieten kann [6]. Dabei ist die Idee ganz einfach und sehr kostensparend. Bei einer Systemerweiterung werden neue Hardware-Komponenten einfach in den Computer eingesteckt, die Zusatzsoftware mit einem intelligenten Installierprogramm hinzugefügt, und alles „spielt“ sofort, wie versprochen.

10 Built-in Network

Anfang 1985 wurde das 230 kBit/s schnelle AppleTalk-Netzwerk (später in *LocalTalk* umbenannt) eingeführt. Jeder Mac ist seitdem mit diesem „Plug & Play“ LocalTalk nach dem *AppleTalk*-Protokoll ausgestattet, das ein einfaches Vernetzen mehrerer Macs sowie von Druckern erlaubt.

11 PostScript Printing

Ebenfalls Anfang 1985 wurde der netzwerkfähige *LaserWriter*, der erste bezahlbare PostScript-Drucker [12] eingeführt. Er kostete 6.995 \$, andere kosteten damals noch weit über 50.000 \$, und die Benutzer von mehreren Macs konnten sich den Drucker via LocalTalk-Netz teilen.

Wir wissen heute, daß dieses den breiten Durchbruch des grafisch hochwertigen PostScript-Drucks, insbesondere im Zusammenhang mit dem im Sommer 1985 von Aldus eingeführten Desktop-Publishing (PageMaker) brachte. Ab 1988 konnten Macs dann auch auf PostScript-Druckern in Farbe drucken (QMS ColorScript 100).

12 Built-in SCSI

Bereits ab 1986 wurden alle Macs mit dem im gleichen Jahr genormten schnellen

SCSI-Bus zum Anschluß von Geräten (Disketten-Laufwerke, Platten, Scanner, CD-ROM u. a.) versehen. Dieser leicht zu konfigurierende und preiswerte Peripheriebus war die natürliche Ergänzung zum „Plug & Play“ Design des Macs.

13 24-Bit Colour (QuickDraw)

Im März 1987 wurden die ersten Macs „farbig“. Die Toolbox des Mac II enthielt in ihren 256 kByte zusätzlich das *Color QuickDraw* für eine 8-Bit Farbdarstellung mit 256 Farben. Dem folgte 1989 das *32-Bit QuickDraw*, womit eine 3x8-Bit „True-Colour“-Farbdarstellung und Verarbeitung möglich wurde. Der Mac wurde damit auch zum Farbbild-Editor, was viele Künstler und Verlage zu nutzen wußten.

Diese 24-Bit Farbtechnik war zwar nicht ganz neu, nur war sie auf dem Macintosh erstmals bezahlbar. Workstations mit ähnlichen Möglichkeiten kosteten zu dieser Zeit 30.000 \$ und mehr, und sie waren auch längst nicht so benutzerfreundlich.

14 Cooperative Multitasking

Zunächst konnte auf einem Mac – wie beim PC – nur immer ein Anwendungsprogramm *nach* dem anderen laufen, aber nicht gleichzeitig. Mit der Einführung des genialen *MultiFinder* (später im System 7 integriert) änderte sich das im August 1987 grundlegend. Es konnten nun – ohne zusätzliche Hardware (!) – mehrere Programme gleichzeitig im kooperativen Multitasking laufen. Zwischen den Programmen (Fenstern) konnte per „Cut & Paste“ Texte oder Grafiken kopiert werden, was eine erhebliche Arbeitserleichterung bedeutete.

15 Built-in Video

Mußte 1987 für den Monitor eines Mac II noch extra ein Video-Controller gekauft werden, wurden bereits ab 1989 alle Macs (beginnend mit dem Mac SE/30 und dem Mac IICI) mit eingebauten Video-Controllern und gesondertem Video-RAM versehen.

16 HyperText (HyperCard)

War die Programmierung des Macintosh anfangs nur etwas für erfahrene Profis, die sich in den modernen Programmiersprachen auskannten, so wurde das mit der Einführung von *HyperCard* im August 1987 etwas anders. Damit konnte nun auch ein normaler Benutzer mit Hilfe der natürlichen Skriptsprache *HyperTalk* Informationen nach seinen Wünschen bearbeiten, organisieren, sortieren und in den „HyperCard-Stacks“ abspeichern.

Mit HyperCard wurde auch erstmals das völlig neuartige Konzept von „HyperText-Links“ zu weiterführenden Informationen eingeführt, dem heute bei der Informationssuche im Internet eine große Bedeutung zukommt (World Wide Web).

17 System Enabler

Mit dem System 7.1 führte Apple das Konzept des „System Enablers“ ein. Das ist eine Datei mit Code, der das Betriebssystem

beim Einschalten des Macs entsprechend der verfügbaren Hardware *automatisch* einrichtet. Damit ist es möglich, das MacOS in jeder Hardwareumgebung, auch mit einem anderen Mikroprozessor (z. B. PowerPC) laufen kann. Es wird dann nur ein neuer System Enabler benötigt.

Eine solche Lösung wünscht sich jeder PC-Besitzer, wenn er z. B. versucht, Windows auf einem PC mit einer nur etwas abweichenden Hardware zu installieren.

18 QuickTime

Zusammen mit dem 32-Bit System 7 wurde im Mai 1991 das *QuickTime* eingeführt, mit dem ein Mac echtzeitbezogene Daten (z. B. digitale Videos mit Ton) synchronisiert darstellen und bearbeiten kann. Die QuickTime-Software ist auch in einer Windows-Version für den PC erhältlich.

19 Voice Control

Zur Standardausrüstung der Audio-Video-Macs (AV-Macs) gehört die Möglichkeit der Sprachsteuerung aller Computer-Funktionen per DSP, was u. a. für Behinderte eine deutliche Erleichterung bedeutet. Bisher funktioniert das nur in Englischer Sprache.

20 Integrated AV

Seit 1993 gibt es die AV-Macs, die auch mit Fernsehsignalen in den Normen PAL, SECAM und NTSC umgehen können, und die Fernsehbilder auf dem Monitor in einem Fenster darstellen können. Mit entsprechender Software können auch Bilder (frames) ausgeschnitten werden, in einen QuickTime Video-Clip eingefügt und auf einem Videorecorder gespeichert werden.

Mit der multimedialen AV-Technologie wird aus dem persönlichen Computer auch ein Kommunikationsgerät, wenn er mit entsprechenden Netzwerken und TV-Satellitenempfangsanlagen verbunden ist.

Literatur

- [1] Levy, Steven: *Insanely Great – The Life and Times of Macintosh, the Computer that Changed Everything*. San Francisco (USA): Viking Press 1993. ISBN: 0-670-85244-9. Preis: 19,95 \$.
- [2] Thompson, Tom: *The Macintosh at 10*. BYTE 19, No. 2, 47–54, Februar 1994.
- [3] Dittberner, K.-H.: Zur Auswahl eines leistungsfähigen PC's. FU Berlin (IfP): wdv-notes Nr. 262, 1993–1994.
- [4] Dittberner, K.-H. (Ed.): *Wird die PC-Welt veräppelt?* FU Berlin (IfP): wdv-notes Nr. 202, 1991–1994.
- [5] Duntemann, Jeff and Pronk, Ron: *Inside the PowerPC Revolution*. Palo Alto (USA): Coriolis Group 1994. ISBN: 1-883577-04-7. Preis: 24,95 \$.
- [6] Halfhill, Tom R.: *Transforming the PC: Plug and Play*. BYTE 19, No. 9, 78–94, September 1994.
- [7] Dijkstra, E. W.: *Cooperating sequential processes*. //z. Genuys, E. (Ed.): *Programming Languages*. London: Academic Press 1968, 43–112.
- [8] Apple Computer Inc.: *Inside Macintosh – Volume I–VI*. Reading (USA): Addison-Wesley 1988.
- [9] Dittberner, K.-H.: *MacMETH: Das Modul »MacTools«*. FU Berlin (IfP): wdv-notes Nr. 294, 1993–1994.
- [10] Dittberner, K.-H.: *Macintosh Computer: Modell-Übersicht*. FU Berlin (IfP): wdv-notes Nr. 236, 1985–1994.
- [11] Wirth, Niklaus: *The programming language Pascal*. Acta Informatica 1, 35–63, 1971.
- [12] Adobe Systems Inc.: *PostScript Language Reference Manual*. Reading (USA): Addison-Wesley 1985. ISBN: 0-201-10174-2. Preis: 22,95 \$.