OpenStep im Detail

Durch die Fusion von Apple und NeXT und die Entscheidung, OpenStep als Basis f r ihr zuk nftiges Betriebssystem zu verwenden, ist OpenStep pl tzlich in aller Munde. Steve Jobs ist damit seinem Ziel, NEXTSTEP bzw. OPENSTEP als das Betriebssystem der Zukunft zu machen, einen großen Schritt n her gekommen, wenn auch anders als erwartet. Doch was genau verbirgt sich hinter diesem Standard und was ist der Unterschied zu OPENSTEP und NEXTSTEP? Dies und einiges mehr soll dieser Artikel beantworten.

von Gerald Erdmann

Der Kreis schließt sich. Als nunmehr vor ber zehn Jahren Steve Jobs Apple im Zorn verlassen und mit dem Geld, daß er durch den Verkauf seiner Apple-Aktien bekommen hat, seine neue Firma NeXT gegr ndet hat, wollte er ein Betriebssystem (damals noch in Verbindung mit einem eigenen, speziell daf r designten Computer) entwickeln, das alles bisher Dagewesene in den Schatten stellt. Er hat es geschafft, zumindest aus technologischer Sicht. NEXTSTEP (oder NextStep oder NeXTstep oder NeXTSTEP, wie es zwischenzeitlich mal hieß) war und ist immer noch fortschrittlicher als fast alles, was auf dem Markt derzeit erh ltlich ist.

Was NeXT stets fehlte, waren entsprechende Ums tze, NEXTSTEP im Markt zu etablieren. Zuerst wurde die Hardware abgestoßen, nicht weil sie zu schlecht war, im Gegenteil, sondern vielmehr weil es sich f r eine kleine Firma nicht rechnet, eine eigene Hardware-Entwicklung zu betreiben, speziell auf einem Markt, auf dem sich alle 18 Monate die Leistung verdoppelt bzw. verdoppeln muß.

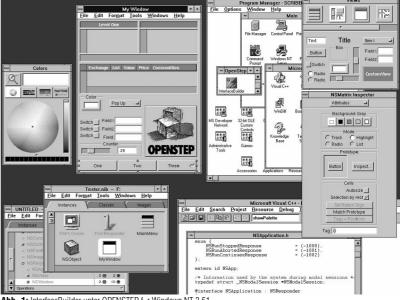


Abb. 1: InterfaceBuilder unter OPENSTEP f r Windows NT 3.51

Der Weg, nur noch das Betriebssystem und die dazugeh rigen Entwicklungstools zu verkaufen, war der Richtige, denn schnelle und solide Hardware k nnen andere besser und billiger produzieren. Ob auch sch ner, sei dahingestellt. NEXTSTEP konnte dadurch die Anzahl der Installation mehr als vervierfachen, doch der weiterhin hohe Preis und ein gleichbleibend schlechtes Marketing halfen nicht, Steves Vision in die Realit t umzusetzen.

NeXT war zuletzt in der Gewinnzone, jedoch haupts chlich durch Speziall sungen f r Großkunden und durch WebObjects. Die Betriebssystem-Fraktion bei NeXT stand kurz vor dem Aus, da inzwischen OPENSTEP f r NT erh ltlich war und dies die Zukunft - laut mehrerer NeXT Manager sein sollte. NeXT hatte den Kampf gegen Windows aufgegeben.

Der Standard

Als sich auch NEXTSTEP nicht im Markt bemerkbar etablieren konnte, beschloß man, das Produkt noch weiter zu beschneiden, in dem man sich auf die objektorientierte Entwicklungsumgebung konzentrierte. Mit Open-Step hat man zusammen mit Sun einen Standard beschlossen, der im Wesentlichen auf der API von NEXTSTEP 3.2 beruht. Mit diesem offenen Standard sollte es m glich sein, plattformunabh ngige objektorientierte Anwendungen zu entwicklen, ohne dabei auf den Komfort von NEXTSTEP zu verzichten.

Der Standard selbst besteht haupts chlich aus den objektorientierten "Kits" von NEXTSTEP und der "Darstellungsmaschine" Display PostScript. Genau dies waren und sind die St rken und genau diese reichen auch aus, um Applikationen zu entwicklen, denn im Prinzip ist es egal, auf welchem Betriebssystem die Applikationen laufen.

Die Architektur

Der OpenStep-Standard besteht aus mehreren Schichten, die bei der Implementierung zwischen der Betriebssystemebene und der Anwendungsebene angesiedelt sind. Das darunliegende System muß jedoch gewisse Merkmale vorweisen. (Preemptives) Multitasking oder virtuelle Speicherverwaltung sind f r eine Umsetzung "lebensnotwendig". Hieraus ergeben sich aber keine gr ßeren Probleme, da heutzutage fast iedes moderne Betriebssystem diese Features zumindest ansatzweise - zur Verf gung stellt.

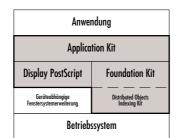


Abb. 2: Die OpenStep-Architektur

OpenStep besteht im wesentlichen aus drei Teilen (Abb. 2):

Application Kit

Das Application Kit stellt dem Entwickler alle notwendigen Objektklassen zur Verf gung, um Programme mit einem grafischen User-Interface (z. B. die Klassen NSWindow oder NSButton) zu erzeugen. Neben einer ganzen Reihe von User-Interface-Objekten, die bei den bisherigen Implementierungen im bew hrten NEXTSTEP "Look & Feel" daherkommen (aber nat rlich nicht m ssen), hilft das Application Kit auch bei der Behandlung von Events (Mausklicks, Tastatureingaben u. .) und bei der Kommunikation zwischen verschiedenen Applikationen, wie "Copy & Paste" oder "Drag & Drop".

Jedes OpenStep-Programm besitzt genau ein Objekt der NSApplication-Objektklasse. Dieses Objekt verwaltet das Men bzw. die Men leiste (beide geh ren zu Klasse NSMenu), f hrt Buch ber alle Fenster (der Klasse NSWindow und deren Unterklassen) dieser Applikation, empf ngt alle Tastatur- und Maus-Events (als NSEvent-Objekt) und leitet diese an die entsprechenden Objekte weiter.

Display PostScript

Wie schon unter NEXTSTEP ist auch unter OpenStep die Grafikausgabe via Display PostScript von Adobe standardisiert. Mit diesem Beschreibungsmodell ist ein echtes "What You See Is What You Get" m glich, denn f r Bildschirm und Drucker werden in der Regel die gleichen PostScript-Kommandos ausgef hrt! Dank zahlreicher Opti-

mierungen steht die Darstellungsperformance auf dem Bildschirm der der anderen Systeme in nichts nach.

Der normale Benutzer merkt von der Verwendung dieser Seitenbeschreibungssprache jedoch nichts, da Display PostScript bei allen Implementierunvollkommen gen transparent ins System eingearbeitet wurde. Auch der Entwickler braucht keine Erfahrung von PostScript zu haben, solange er die grafischen Objekte des Application Kits verwendet.

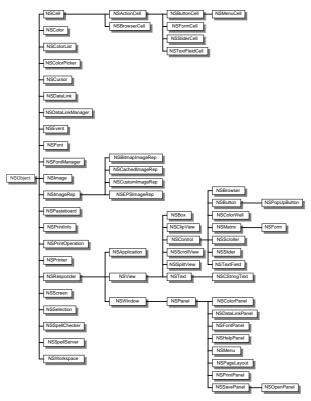


Abb. 3: Ein Ausschnitt der Objekthierarchie des Applikation Kits

Erst wenn er selbst Grafiken (z.B. ein Kuchendiagramm) in einem Objekt einer Unterklasse des NS-Views darstellen will, muß er dies via PostScript machen.

Foundation Kit

Das Foundation Kit ist der Unterbau von OpenStep und dient zum hardwareunabh ngigen Umgang mit Daten und Resourcen, wobei die grundlegenden Klassen zur Verwaltung von Feldern, Listen, multinationalen Zeichens tzen (in Unicode), Threads oder verschiedener Zahlenformate dienen.

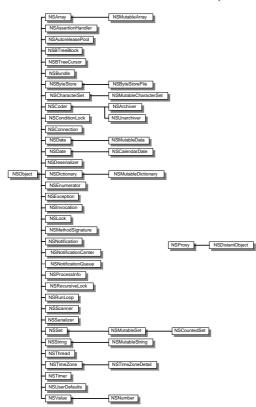


Abb. 4: Ein Ausschnitt der Objekthierarchie des Foundation Kits

Neben dem Hauptobjekt NSObject, von dem alle Objektklassen von OpenStep abstammen, dient die Klasse NSTimer dazu, bestimmte Objekte mit einer Zeitverz gerung – auch mehrmals – aufzurufen. Objekte der Klasse NSNotification werden "verschickt", sobald ein bestimmtes Ereignis eingetreten ist oder eintreten wird, wie z. B. daß ein Fenster vergr ßert oder verschoben werden soll oder daß die gesamte

Applikation beendet wird; um solche Nachrichten zu empfangen, muß man sich im NSNotificationCenter "anmelden." Whrend man mit Objekten der Klasse NSThread Programmteilst cke parallel – als Thread – ablaufen lassen kann, dient die Klasse NSBundle dazu, um Programmteile bei Bedarf dynamisch nachzuladen.

Das Foundation Kit selbst beinhaltet noch zwei weitere Teile, die eher im Foundation Kit selbst angesiedelt sind: Distributed Objects und das Indexing Kit.

Distributed Objects & Indexing Kit

Distributed Mit dem Objects Framework (DO) lassen sich beliebige Objekte transparent einem heterogenen Netzwerk weltweit verteilen. Die hierbei angebotenen Objekterweiterungen, die in der "Mutter aller OpenStep-Objekte" NSObject angesiedelt sind, definieren implementieren den Transportmechanismus auf der Basis der Interprozeßkommunikation des Unix-Derivats Mach. Dieser Mechanismus wird gegen ber den zu verteilenden Objekten verborgen, so daß f r die Applikation kein Unterschied besteht, ob

ein Objekt lokal oder verteilt arbeitet; die Objektschnittstelle ist f r beide F lle identisch; es ist ebenfalls egal, auf welcher Plattform das entfernte Objekt beheimatet ist.

Um auch eine Kommunikation mit anderen objektorientierten Umgebungen zu erm glichen, k nnen Objekte bereits heute mit Microsofts OLE/COM-Objekten zusammenarbeiten (DOLE); auch hier wird die "bersetzung" der Objektkommunikation verborgen. CORBA 2.0 Objekte werden in naher Zukunft ebenfalls unterst tzt werden.

F r andere UNIX-Derivate wie z.B. HP-UX, Digital Unix, SunOS und Solaris gibt es von NeXT selbst bzw. jetzt von Apple ein Framework, um Distributed Objects auch ohne OpenStep auf den Server-Plattformen zur Verf gung zu stellen; mit Portable Distributed Objects (PDO) kann man z. B. auf einem Datenbank-Server, der unter einem der oben genannten Betriebssysteme l uft, per DO von einem OpenStep-Rechner zugreifen.

Das Indexing Kit war unter NEXTSTEP eine eigene Objektsammlung verschiedener Klassen zur effizienten Verwaltung gr ßerer Datenmengen. Unter Open-Step dient es dazu, Objekte und unstrukturierte Daten - wie Texte, Bilder aber auch andere multimediale Informationen - effizient abzuspeichern, zu indexieren und schnell wiederzufinden. Unter OpenStep wird das Indexing Kit nur intern f r einige Objektklassen verwendet; f r den "Endentwickler" steht es selbst nicht zur Verf gung.

Die Implementierung

W hrend ein Standard selbst nur auf dem Papier oder auf einem Datentr ger steht, ben tigt man vielmehr eine Implementierung von OpenStep. Hierbei unterscheidet man Laufzeitumgebungen, um OpenStep-Programme ablaufen zu lassen, und Entwicklungsumgebungen, um OpenStep-Programme zu entwickeln.

Die Laufzeitumgebung besteht aus diversen Funktionsbibliotheken, die auf dem jeweiligen Rechner installiert werden m ssen. Zu diesen Bibliotheken gesellen sich noch ein Display-PostScript-D



mon, der f r die Umsetzung der PostScript-Kommandos zust ndig ist, ein Kommunikations-D mon, der vor allem f r die Kommunikation von verteilten Objekten ben tigt wird und aus einem Pasteboard-D mon, der f r den Datenaustausch zwischen einzelnen Programmen – nicht nur zwischen OpenStep-Kompatiblen – zust ndig ist.

Die Laufzeitumgebung, einmal installiert, 1 uft vollkommen unsichtbar im Hintergrund des jeweiligen Betriebssystems ab. OpenStep-Programme und Nicht-OpenStep-Programme sind dabei vom Benutzer nicht zu unterscheiden. Erst beim Start einer Open-Step-Applikation werden vom Betriebssystem die entsprechenden D mons automatisch mitgestartet, so daß auch weiterhin kein Unterschied zu merken ist.

Eine Entwicklungsumgebung hingegen, besitzt dar berhinaus noch die von NEXTSTEP her bekannten Klassiker wie den Interface-Builder, den ProjectBuilder und nat rlich einen Compiler, der auf GNU gcc basiert.

Die ersten Implementierungen brachte NeXT selbst vor gut einem Jahr mit OPENSTEP/Mach und OPENSTEP/NT auf den W hrend OPEN-Markt. STEP/Mach eigentlich nichts anderes als NEXTSTEP 4.0 mit OpenStep-kompatibler Programmierschnittstelle ist, bietet OPEN-STEP/NT nur die Umsetzung der OpenStep-Spezifikation f r Windows NT, um entsprechende Programme zu entwickeln und ablaufen zu lassen.

OpenStep? OPENSTEP? Die Verwirrung ist fr viele an dieser Stelle komplett. OpenStep (gemischte Schreibweise) definiert genau, welche Objekte mit welchen Methoden dem Programmierer – theoretisch – zur Verfgung stehen, ist also der Name

der Spezifikation. OPENSTEP (komplett große Schreibweise) hingegen stellt dem Entwickler eine Implementierung von Open-Step zur Verf gung, um entsprechende Programme zu entwickeln. Da OpenStep nur vorschreibt, was mindestens implementiert sein muß, steht es jedem Unternehmen frei, weitere Klassen, Methoden oder ganze Kits bzw. Frameworks zur Verf gung zu stellen; da diese nicht standardisiert sind, kann hierbei nat rlich keine Garantie der problemlosen Portierbarkeit gegeben werden, falls diese verwendet wer-

W hrend OPENSTEP/Mach das komplette NEXTSTEP-Betriebssystem zur Verf gung stellt, kann man unter Windows NT und ab OPENSTEP/NT 4.2 auch unter Windows 95 nur auf die im Standard festgeschriebenen Klassen zur ckgreifen. Diese Klassen sind Pflicht, denn nur dann ist OPEN-STEP auch OpenStep. Um also den verschiedenen zwischen OPENSTEP-Implementierungen Source-Code-kompatibel zu bleiben, darf nur auf die im Open-Step-Papier festgeschriebenen Objektsammlungen zur ckgegriffen werden.

Nobody is Perfect

Nat rlich hat auch OpenStep gewisse Grenzen und Einschr nkungen. Der Source-Code ist nur kompatibel soweit er sich an die Vorlagen der Spezifikation h lt. Dar berhinaus herrscht eine gewisse Grauzone, denn es gibt mehrere Objektklassen, die z. B. in OPENSTEP/Mach und OPEN-STEP/NT enthalten sind, jedoch OPENSTEP/Solaris fehlen. Auch die Spezifikation selbst ben tigt noch an einigen Stellen ein paar Nachbesserungen, die aber mit Sicherheit in geraumer Zeit beseitigt werden.

berhaupt nicht portabel sind derzeit die fertig bersetzten Programme, d.h. f r jede Plattform muß eine eigene Bin rversion erzeugt werden. OpenStep ist eine reine API (Application Programming Interface) und keine virtuelle Maschine wie etwa Java. Zwar wird man in Zukunft neben Objective C auch OpenStep-Programme in Java und C++ schreiben k nnen, doch bleibt das Resultat architekturgebunden. Zur Zeit jedenfalls.

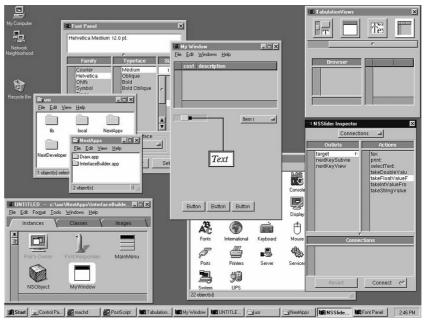


Abb. 5: OPENSTEP f r Windows 95



Die Zukunft

Die Zukunft heißt in jeglicher Hinsicht Apple, da der MacOS-Nachfolger "Rhapsody" da aufsetzt, wo OPENSTEP f r Mach-OS aufgeh rt hat. Apple wird – in guter alter NeXT-Manier – Rhapsody nicht nur f r Power-PC-Prozessoren, sondern (vermutlich zeitgleich) auch f r Intel-Prozessoren auf den Markt bringen; "Fat Binaries", also eine Programmversion, die auf beiden Plattformen ausf hrbar ist, werden ebenfalls unterst tzt.

Dar berhinaus wird Apple weiterhin OPENSTEP f r Windows NT, Windows 95 und demn chst auch OPENSTEP f r MacOS herausbringen. Gerade letzteres ist f r den Erfolg von Rhapsody sehr wichtig, denn damit k nnen alle jetzigen MacOS 7.x und demn chst MacOS 8.x Besitzer ebenfalls OpenStep-Programme laufen lassen, die schon speziell f r Rhapsody entwickelt wurden; ein fließender bergang zwischen den beiden Betriebssystemen ist somit gew hrleistet. Mit Sicherheit wird diese sogenannte "Yellow Box" f r MacOS kostenlos sein.

Ebenfalls kostenlos wird demn chst die Laufzeitumgebung f r Windows NT und Windows 95 zu haben sein. Mit dieser Entscheidung hat Apple genau den richtigen Schritt in Richtung Massenverbreitung von Open-Step gemacht, denn wer heute Programme nach dem OpenStep-Standard entwickelt, kann nicht nur dem gesamten Apple-Markt, sondern auch dem Windows-Markt (unter Intel-Hardware) verkaufen! Die Programme selbst m ssen nicht modifiziert, sondern mittels eines einzigen Mausklicks einfach neu bersetzt werden.

Weiterhin hat Sun noch OPEN-STEP f r Solaris im Programm, so daß selbst dieser große Workstationmarkt nun pl tzlich ebenfalls nur einen Mausklick entfernt ist. Wer dies nicht glaubt, frage bitte bei Stone Design (http://www.stone.com) nach, das bereits zur Mac-Entwicklerkonferenz, auf der Rhapsody zum ersten Mal der ffentlichkeit pr sentiert wurde, sein Grafikprogramm Create – neben OPENSTEP/Mach, OPENSTEP/NT und OPENSTEP/Solaris – f r Rhapsody portiert hat: Kopiert. Kompiliert. Fertig.

Last but not least arbeitet die Free Software Foundation an einer freien OPENSTEP-Implementierung, genannt GNUstep. Neben einer kompletten Eigenentwicklung aller Bibliotheken wird das Darstellungssystem auch auf Display PostScript beruhen, wobei der D mon auf GhostScript basiert. Da GNUstep als Sourcecode verf gbar ist, kann es theoretisch f r jede Plattform bersetzt werden, wie z.B. Linux, Windows oder OS/2. Die Entwicklung ist aber noch lange nicht abgeschlossen, da alle Entwickler ehrenamtlich "nebenbei" arbeiten; N heres zu GNUstep und zum Stand der Entwicklung finden sie unter http://www. GNUstep.org.

OPENSTEP/Mach alias NEXT-STEP wird hingegen vermutlich mit der im Juni erscheinenden Version 4.2 wohl die letzte Version berhaupt sein - Rhapsody ist ja der designierte Nachfolger -, jedoch k nnen nun auch viele Betriebssysteme - vor allem diverse Unix-Derivate - nun Ihr eigenes System aufwerten. Durch Anbieten einer OpenStep-Laufzeitumgebung wird ber kurz oder lang Großteil der Macintosh/NEXTSTEP-Programme f r die eigene Plattform verf gbar sein. Kein Softwarehersteller muß " berredet" werden, ein Vierteljahr Arbeit in eine Portierung zu stecken, denn der Betriebssystemanbieter bernimmt selbst

die "Portierung". Einmal Open-Step, immer (wieder) OpenStep.

Mehrere Betriebssysteme sind schon im Gespr ch. Neben Digital und Hewlett Packard, die offziell schon seit ber einem Jahr an einer OpenStep-Implentierung fr Ihre Unix-Systeme arbeiten, kursiert seit geraumer Zeit auch der Name Silicon Graphics.

Sollten die letzten drei Hersteller mit auf den OpenStep-Zug springen, ist durchaus zu erwarten, daß OpenStep einen hnlichen Erfolg wie NFS, das ebenfalls als Standard begann, haben wird.

