

Professionelle Entwicklung mit NEXTSTEP/OPENSTEP/PDO/EOF

Peter Pichler

NeXT™ ist der einzige Anbieter, der eine komplett integrierte objektorientierte Client/Server Umgebung für die unternehmensweite Software Entwicklung liefert. NEXTSTEP bietet somit eine ausgezeichnete Entwicklungsplattform zur Erstellung von unternehmensspezifischen Client-Server-Lösungen und zur Integration führender Standardsoftware in allen Netzwerkumgebungen.

NEXTSTEP läuft heute auf Intel-PCs (Marken wie Dell, Compaq, NEC, HP, Digital, NCR, Epson...), NeXT Hardware, HP-PA RISC, Sun SPARC sowie DEC ALPHA Workstations und zu seinen Kunden zählt NeXT internationale Konzerne wie:

Disney, Mitsubishi, Reebok, Motorola, MCI, Fannie Mae, Meryll Lynch, Schweizer Bankverein, ...

Eine große Stärke der NEXTSTEP Entwicklungsumgebung liegt in der Realisierung sogenannter *Mission Critical Applications*, wo es weniger darauf ankommt, große Stückzahlen zu verkaufen, sondern darum, in kurzer Zeit verlässliche, für den Erfolg des Unternehmens dringend benötigte Software zu erstellen, bzw. Anpassungen vorzunehmen (Gesundheitswesen, Banken, etc.).

Als zum Beispiel das Berliner Rudolf-Virchow-Klinikum - eines der größten deutschen Krankenhäuser - seine Software an die neuen Anforderungen durch das am 1.1.1996 in Kraft getretene, neue Gesundheitsstrukturgesetz (GSG) anpassen mußte, startete man ein Entwicklungsprojekt unter MS-Windows mit den dazugehörigen Tools. Das Projekt mußte aber erfolglos abgebrochen werden, da sich herausstellte, daß ein derart komplexes Client/Server System unter Windows nicht realisierbar war.

Die Entwicklungsgesellschaft EFI bot daraufhin die Entwicklung eines Prototypen unter NEXTSTEP an, und nach nur 6 Monaten Entwicklungszeit stand ein vollständig dem neuen Gesetz gerechtes System zur Verfügung. Nach der Testphase auf 20 Arbeitsplätzen läuft es jetzt auf 80 Stationen (geplant ist der Ausbau auf alle 700 Plätze). Kommentar des EFI Entwicklungsleiters: *„Unsere Erfahrung zeigt, daß die Entwicklung auf OS/2 oder Windows vier bis 10 mal so lange dauert...“*

VME, eine der größten Einkaufsvereinigungen im Möbelhandel, mit 180 angeschlossenen Möbelhäusern in Deutschland, Österreich und der Schweiz, mit einem Gesamtumsatz im Vorjahr von 4.3 Mrd. DM, setzte bei der Umstellung vom "kostenintensiven Austausch von Informationen zwischen VME und seinen Mitgliedsunternehmen" auf NEXTSTEP, und betreibt zwei SUN Server für Kommunikation und Datenbank, sowie mehrere 486er und Pentiums als NeXT-Clients.

Was macht die Überlegenheit von NEXTSTEP aus?

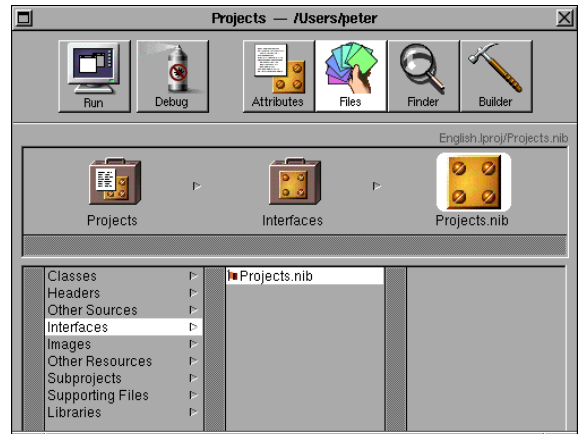
Schon vor 7 Jahren setzte man bei NeXT auf *objektorientierte Technologien*, und so ist NEXTSTEP heute das einzige ausgereifte und vollständig objektorientierte offene Betriebssystem.

Außerdem setzte man konsequent auf bewährte Standards wie *RPC, TCP/IP, NFS, GOSIP, POSIX, Novell, SMTP*. Der *NEXTSTEP Window-Server* basiert wie z.B. X-Windows auf dem *Client/Server Modell*. Zur Darstellung wurde *Display PostScript* gewählt, woraus sich die vollkommene Übereinstimmung von Bildschirm-inhalt und Ausdruck ergibt. PostScript Objekte können mittels Drag&Drop beliebig zwischen Applikationen ausgetauscht werden.

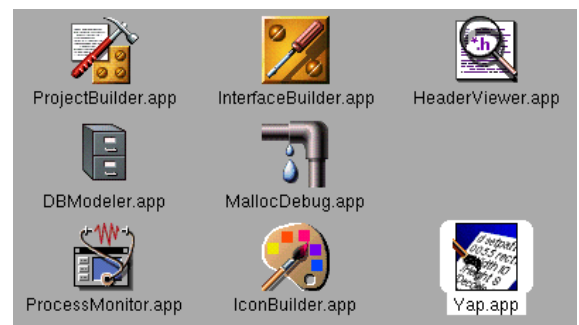
Multi Architektur Binaries ermöglichen die Speicherung von Code für verschiedene Hardware-Plattformen in einer Applikation. Sie können also eine Anwendung einmal auf ihrem File-Server installieren und (in ihrem heterogenen Netzwerk) sowohl auf PCs als auch auf Workstations verschiedener Hersteller im ganzen Unternehmen nutzen. Auch die Verbreitung von bereits übersetzten Programmen im Internet wird so unterstützt...

Die integrierte Entwicklungsumgebung besteht aus:

- **ProjectBuilder** (organisiert Files, Ressourcen und Parameter, ermöglicht integrierten Zugriff auf die anderen Entwicklungswerkzeuge, steuert den Entwicklungsprozeß).



- **InterfaceBuilder** (konstruiert interaktiv die Benutzerschnittstelle durch die Erzeugung und Konfiguration von entsprechenden Objekten; es kann ein Skelett der Anwendung erstellt (und auch getestet) werden, ohne eine einzige Zeile Code zu generieren; wird nur das Interface verändert, muß die Applikation nicht neu übersetzt werden - Zeitersparnis).
- **DBModeler** (erstellt ein OO Datenmodell auf der Basis relationaler Datenbanken).

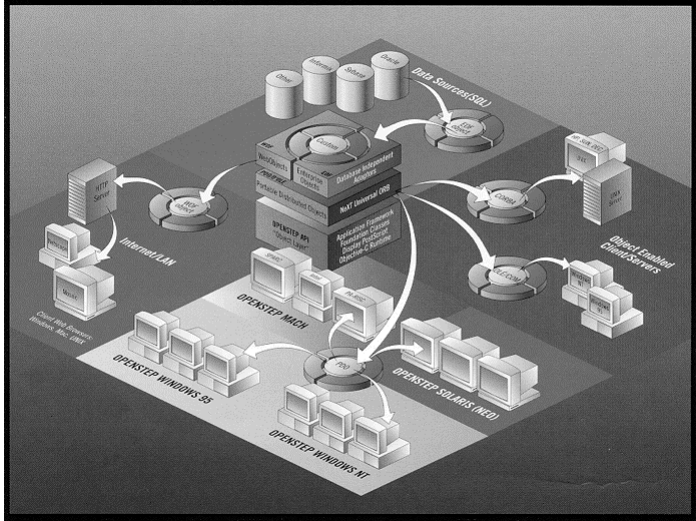


Zusätzlich stehen dem Entwickler zur Verfügung:

- **Application Kit** (gewährleistet die Konsistenz von Applikationen verschiedener Hersteller durch konsequente Umsetzung der "User Interface Guidelines"),
- **Database Kit** (Client/Server-basierter Zugriff auf Datenbanken über Adapter, Erstellung von Entity/Relationship Modellen, Objekte zur Darstellung),
- **Indexing Kit** (Verwaltung von Ressourcen für Multimedia Anwendungen),
- **Foundation Kit** (Hardware-unabhängiger Umgang mit Daten und Ressourcen),
- **Enterprise Objects Framework** (EOF; Anbindung an relationale Datenbanken),
- **Portable Distributed Objects Framework** (PDO; transparente Verteilung von Objekten im heterogenen Netzwerk).

OPENSTEP

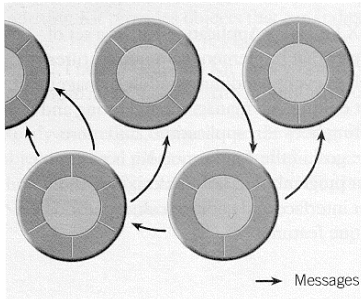
- wird seit 1994 mit SunSoft (Software Abteilung von SUN) entwickelt,
- ist eine Plattform für Entwickler zur Erstellung von objektorientierter Software,
- ist ein offener Standard für portable objektorientierte Anwendungen.
- basiert auf den Programmierschnittstellen der aktuellen NEXTSTEP Version.



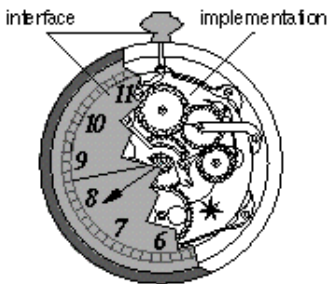
Durch OPENSTEP werden viele neue Plattformen erschlossen (Windows-95 / NT/ SUN-NEO / Solaris). Unter Windows arbeiten so OPENSTEP- und Windows-Anwendungen am selben Computer zusammen. Die Entwicklungsumgebung von NEXTSTEP wurde portiert.

Objekt orientierte Programmierung:

Das OO-Programmiermodell ist heute allgemein anerkannt. Das *Objektmodell* von NeXT ist durch 7 Jahre Entwicklung und drei „Major Releases“ sehr ausgereift und zuverlässig.

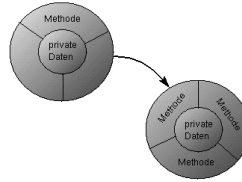


Die Idee ist Applikationen als Netze von Objekten aufzubauen, welche die kleinsten funktionalen Einheiten bilden. Die Objekte entsprechen jenen aus der realen Welt, die sie repräsentieren, und beinhalten Daten (*Attribute*) sowie Operationen (*Methoden*), um auf die Daten zuzugreifen, bzw. Funktionalität zu aktivieren. Die Kommunikation zwischen Objekten geschieht über *Messages* - Nachrichten die Daten übermitteln und Methoden starten.



Durch die strikte Trennung in der *Schnittstelle* zum Objekt (Beschreibung der zur Verfügung gestellten Funktionalität) und der *Implementierung* derselben, kann die Implementierung eines Objekts jederzeit durch eine gleichwertige ersetzt werden, ohne daß Programme die dieses Objekt verwenden, betroffen werden. Außerdem sind solche Objekte viel leichter wiederzuverwenden, als konventionelle Unterprogramme.

Ein wesentlicher Vorteil kommt zum Tragen, wenn Objekte über das Netzwerk operieren - dadurch werden Anwendungen wie PDO und WebObjects, die weiter unten behandelt werden, erst möglich.



Durch Abstraktion und Kapselung wird zudem der Entwickler im Designprozess unterstützt. Die mitgelieferte Objekt-Bibliothek umfaßt circa 80% der üblichen Standardfunktionen.

Objective-C

Objective-C, die Standard-Programmiersprache der NEXTSTEP-Entwicklungsumgebung, ist klarer gehalten als C++. Die Verknüpfung der Objekte erfolgt erst zur Laufzeit wodurch eine Änderung/Erweiterung im Betrieb möglich wird (ohne neue Übersetzung und einen Neustart der Applikation == Dynamic Binding).

Es wurden zu ANSI-C lediglich ein neuer Datentyp, eine Notation zum Versenden von Messages (plus Runtime-Modul), Syntax zur Definition und Implementierung von Klassen hinzugefügt.

Beispiel für die Programmierung mit Objective-C

```
// Headerfile "RateServer.h"
// #import verhindert das mehrfache includieren von Headerfiles

#import <appkit/appkit.h>

@interface RateServer: Object
{
    float rate; // private Daten
}
- init;
- (float) rate; // Lesezugriff auf private Daten
- setRate: (float)newRate; // Schreibzugriff
- addToRate: (float)increment; // Operation auf privaten Daten
@end

// -----
// Implementationsfile "RateServer.m"

#import "RateServer.h"

@implementation RateServer

- init
{
    [super init]; // initialisieren der Superklasse
                  // von der abgeleitet wurde
    rate = 0.0;
    return self;
}

- (float)rate // Implementierung der Zugriffsmethode
{
    return rate;
}

- setRate: (float)newRate // Setzen eines neuen Wertes
{
    rate = newRate;
    return self; // der Default-Returnwert ist vom
                  // Typ id, self zeigt auf uns
}

- addToRate: (float)increment
{
    if ( increment > 1000.0 )
    {
        id sound;
        if ( sound=[ Sound findSoundFor: "WowThatsMuch" ] )
        {
            [sound play]; // Sound abspielen...
            [sound free]; // und Ressource freigeben
        }
    }

    rate += newRate; // Erhöhen des Wertes
    return self;
}

@end
```

Portable Distributed Objects (PDO)

PDO ermöglicht die Kommunikation zwischen Applikationen durch Objective-C Objekte (realisiert so auf komfortable Weise Interprozess-Kommunikation), basierend auf dem *RPC Standard* (Remote Procedure Calls).

Server Dienste werden über ihren Namen verfügbar gemacht (*"vendind"*), Clients bauen eine Verbindung über den Namen des Dienstes auf und können dann vollkommen transparent auf die Server-Objekte zugreifen.

PDO kann mit Microsoft's *OLE/COM* und *CORBA* (Common Object Request Broker Architecture) zusammenarbeiten.

Beispiel für die Programmierung von PDO

```
// -----
//                               PDO Bei spi el
// -----
// (Objective-C unterstützt „C++“ Kommentare)

// Server:
id vendObj = [ [ServerClass alloc] init ] ; // Object erzeugen,
//                               // initialisieren,

id serverConn = [ NXConnection // und bekannt machen
                  registerRoot:vendObj
                  withName:"ServerObject" ] ;

if ( serverConn )
{
    [ serverConn runFromAppKit ] ; // Event Loop starten,
} // der Server ist bereit
// für Requests.

// -----
// Client:
id proxyObj = [ NXConnection // connect zum Server über Namen
                  connectToName:"ServerObject"
                  onHost:"ServerHost" ] ;

[ proxyObj doWorkForMe ] ; // Services konsumieren
```

Enterprise Objects Framework (EOF)

Das Enterprise Objects Framework (EOF) ist plattformübergreifend und stellt einen Bausteinkasten zur Entwicklung mehrstufiger Client/Server Anwendungen bereit. Es verbindet relationale Datenbanken mit Objekten in der NEXTSTEP-Applikationsumgebung und macht so aus Firmen-Daten Firmen-Objekte. Unterstützt werden relationale Datenbanken wie Oracle, Sybase und andere, sowie auch SAP R/3 Anbindungen.

EOF arbeitet mit dem ProjectBuilder, InterfaceBuilder und PDO zusammen und erweitert somit die Entwicklungsumgebung.

WebObjects, dynamische Web Seiten

WebObjects liefert die Werkzeuge zur Entwicklung von dynamischen, Server-basierenden Anwendungen für das World Wide Web (WWW wurde durch Tim Berners-Lee am CERN 1990 auf NeXT entwickelt).

Durch die Möglichkeit, Instanzvariablen in HTML-Seiten und Objektmethoden durch Benutzeraktivitäten starten zu lassen, können *dynamische Web Seiten* realisiert werden. Dabei kapselt WebObjects HTML/HTTP Interna für Programmierer, und stellt ein eigenes *Session Management* für das zustandslose HTTP zur Verfügung.

WebObjects beseitigt einige Nachteile von *HTML* (Hypertext Markup Language):

- schlechte Performance durch seiten-orientierte Transaktionen (Bearbeitung nur auf dem Server möglich)
- eingeschränkte Möglichkeiten für das Benutzer-Interface, nur einfache „Widgets“.
- nur statische Web Seiten.
- keine Zusammenarbeit mit existierenden Applikationen und Daten.
- fehlende Skalierbarkeit.

WebObjects unterstützt Java, arbeitet mit jeden beliebigen HTTP-Server und Browser zusammen, unterstützt gleichzeitig mehrere Datenbanken/Datenquellen und bietet folgende Vorteile :

- verbesserte Performance beim Client.
- dynamische, Echtzeit Web Anwendungen.
- große Auswahl an Benutzerschnittstellen-Komponenten.
- Verteilung von Applikationen über mehrere Server + „load-balancing“ (Lastausgleich).
- Plattform-Unabhängigkeit.
- Integration von Web Anwendungen in existierenden Umgebungen.

Kombination Java und NeXT-Objektmodell

Die offene Technologie unterstützt alle Web-Standards wie Browser, HTTP Server, Skriptsprachen (Java, Perl). Eine eigene Foundation existiert zur Beschleunigung von Entwicklungen. Innerhalb von WebObjects kann Java sowohl am Client (Applets) als auch am Server laufen.

WebObjects ist unabhängig von Browsern (Netscape, Spry, Microsoft, Mosaic, OmniWeb..) und arbeitet mit allen Standard-HTTP Servern. Zusätzlich werden *Vended Java Applets*, HTML3.0, SHTTP, Netscape Erweiterungen zu HTML und Netscape SSL (secured links) zu *Netscape Commerce* Servern unterstützt.

Weitere Informationen zu NeXT

<http://www.next.com>



Die Mann-Jahr-Theorie: Ein Projekt für das ein Programmierer einen Monat braucht, brauchen zwei Programmierer zwei Monate.

Kommentiere jede Zeile

```
#include <stdio.h> /* include-Datei für io-Funktionen */
#include <di eses.h> /* include-Datei für dieses Programm */
/* Leerzeile zur Verbesserung der Lesbarkeit */
void main(void) /* Beginn des Hauptprogramms */
{ /* Blockbeginn */
    int a=0; /* der Variablen a den Wert 1 zuweisen */
    a++; /* a um Eins erhöhen */
    ... /* Hier könnte weiterer Code folgen */
} /* Blockende */
```

Kommentiere nichts

```
...
p+=>(*a++);
a+++++;
p=(a-b)?*a;*b;
...
```