

So funktioniert

Hinter dem Kürzel OLE verbirgt sich eine komplexe Technologie, die in Windows 95 und Windows NT enthalten ist. CHIP sagt Ihnen, was Object Linking and Embedding bedeutet und was Sie damit machen können.

Der Begriff „Objekte“ prägt seit den letzten Jahren die gesamte Softwarebranche. Da gibt es objektorientierte Programmiersprachen, objektorientierte Betriebssysteme und Programme, die Objekte benutzen. Selbst die Dokumente – also die Dateien, mit denen der Benutzer arbeitet – bestehen zum Teil schon aus Objekten.

Warum ist eine Orientierung auf Objekte so attraktiv für Programmierer, und welche Konsequenzen hat der Trend für den normalen Benutzer? Der Vorteil für den Programmierer liegt in der Abgeschlossenheit seines Codes, wodurch ein leichtere Handhabung in großen Projekten erzielt wird.

Auch der Benutzer hat Vorteile: Er braucht sich nicht mehr durch immer wieder neugestaltete Fenster mit gleicher Funktionalität zu kämpfen, was die Einarbeitung in unterschiedliche Programme vereinfacht. Aber auch der Datenaustausch zwischen Programmen geht mit Objekten besser und wird zudem flexibler – OLE macht's möglich.

Nur Vorteile also? Nein, es gibt auch Probleme. Je umfassender nämlich ein objektorientiertes Modell ist, desto höher werden die Ansprüche an Arbeitsspeicher und Rechenleistung. Konkret: Wer unter Windows 95 das OLE 2.0 mit Word und Excel flüssig nutzen möchte, sollte mindestens 16 Megabyte Arbeitsspeicher zur Verfügung haben.

Außerdem darf nicht vergessen werden, daß alle verwendeten Applikationen verfügbar sein müssen, wenn das Dokument vom Kollegen auf einem anderen Rechner bearbeitet werden soll. Dieses Problem wird durch die Viewer entschärft, die inzwischen für die oft benötigten Applikationen erhältlich sind.

Die Schwierigkeiten mit verteilten Objekten auf unterschiedlichen Rechnern ist eine weitere Hürde für die Objektsysteme, die jedoch mit Network OLE behoben werden soll (siehe Kasten S. 210).

Wenn das Thema Object Linking and Embedding (OLE) in einem Atemzug mit Windows 95 genannt wird, liegt es daran, daß vorher OLE stets eine Erweite-

rung war, die von verschiedenen Anwendungsprogrammen über das Betriebssystem gestülpt wurde. Probleme mit unterschiedlichen Versionen führten oft zu Fehlern und gelegentlich zu nicht mehr benutzbaren Programmen. Jetzt ist OLE 2.0 ein Bestandteil des Win-95-Pakets und gehört auch zu Windows NT.

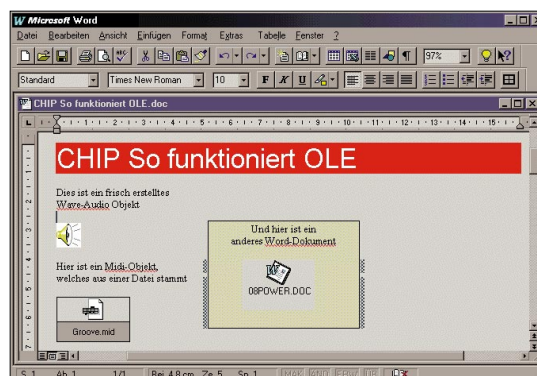
○ OLE – was man damit machen kann

Doch was bringt OLE 2.0 eigentlich? Mit OLE lassen sich zum Beispiel Word-Dokumente erzeugen, die eine Excel-Tabelle enthalten. Das Besondere daran ist, daß diese Tabelle in Word aktiviert und bearbeitet werden kann. Außerdem ist es möglich, eine Verknüpfung mit der Excel-Tabelle so einzurichten, daß beim Ändern der Tabelle automatisch auch das Word-Dokument nebst eingelagerten Grafiken angepaßt wird.

Noch sinnvoller wird diese Anwendung in Verbindung mit Datenbanken. So können zum Beispiel die aus dem Lager entnommenen Waren bei der Erstellung des Lieferscheins automatisch eingebucht werden, weil die Positionen der Lieferscheintabelle eine Verbindung mit der Datenbank besitzen.

Beide Szenarien machen auch deutlich, worin der hauptsächlich Gewinn für den Anwender besteht: Das Bearbeiten der Daten erfolgt nicht mehr mit vielen unterschiedlichen Programmen, sondern in einer gewohnten Hauptumgebung, zum Beispiel der Textverarbeitung. Die Verbindung zu den anderen, spezifischen Programmen erfolgt durch einzelne Module. Die Demonstration des „Office der Zukunft“ zeigt, wie es weitergeht: Es wird nicht einmal mehr spezi-

Innig verknüpft:
Diese Icons
repräsentieren
Objekte. Die
Sounddatei wird
per Doppelklick
aufgerufen, dann
startet sofort
die Medien-
wiedergabe.



elle Programme für Texte, Tabellen und Datenbanken geben, sondern nur Module, welche die Funktionen enthalten.

○ Die Funktionen im Detail

Was darf man von OLE 2.0 erwarten? Ein Blick auf den Funktionsumfang hilft da weiter. Im wesentlichen enthält OLE 2.0 die folgenden Merkmale:

- Aktivierung im Dokument
- Drag & Drop
- geschachtelte Objekte
- Vererbung von Eigenschaften
- Programmierbarkeit
- Konvertierung und Nachbildung von Objekten
- erweitertes Objektlayout
- Suchen und Rechtschreibprüfung
- Management von Versionen

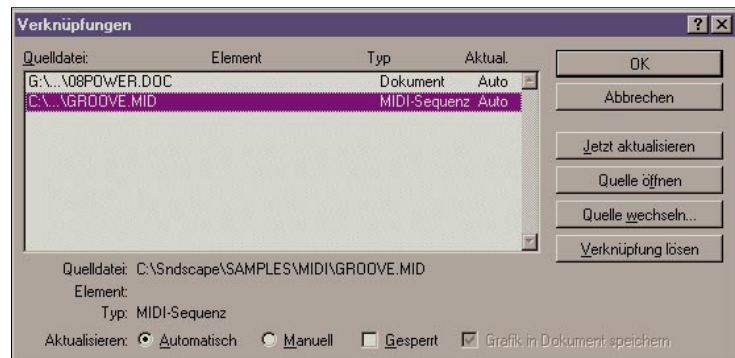
Die *Aktivierung im Dokument* erweist sich für den Anwender als bequemer Weg, das Objekt zu bearbeiten. Ein Doppelklick genügt, um beispielsweise eine Paint-Zeichnung in einem Word-Dokument zu retuschieren. Die Menüleiste von Word wird dabei gegen die des Grafikprogramms ausgetauscht.

Drag & Drop ist ein wesentlicher Bestandteil in Windows 95. Diese Eigenschaft funktioniert bei OLE auch zwischen Applikationen. Beispielsweise kann so eine Excel-Grafik sehr schnell in ein Word-Dokument eingebettet werden. Eine Verknüpfung per Drag & Drop herzustellen, funktioniert allerdings nicht.

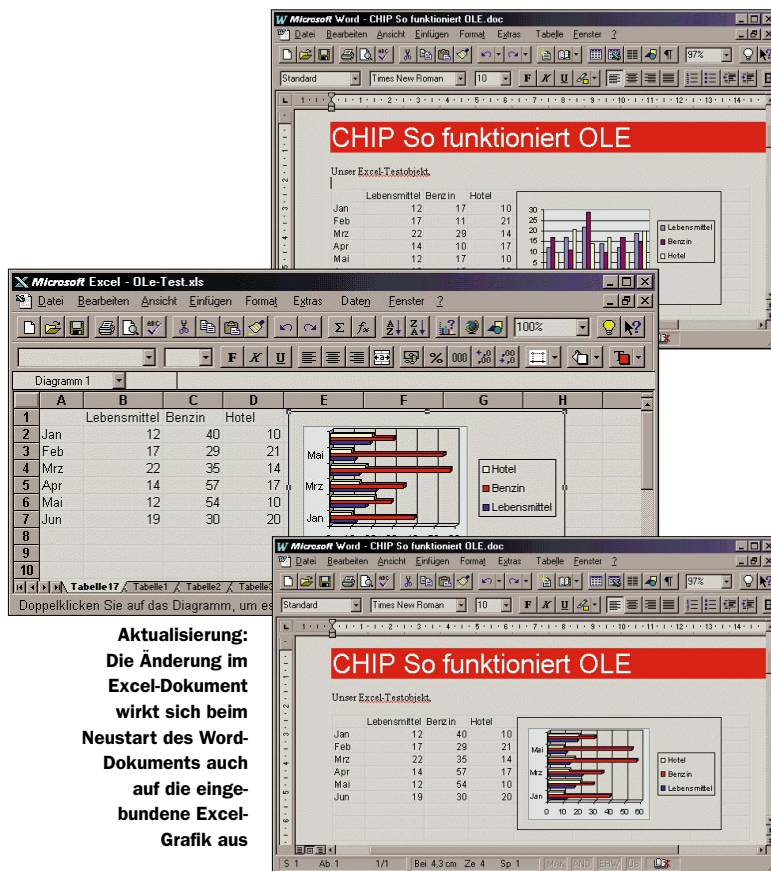
OLE läßt auch *geschachtelte Objekte* zu. Ein Beispiel: Ein Textdokument kann eine Excel-Tabelle enthalten, die wiederum ein grafisches Objekt in sich einge-

bunden hat. Trotzdem kann das grafische Objekt von der Textverarbeitung aus editiert werden.

Mit der *Vererbung von Eigenschaften* ist eine Einflußname des Rahmenobjekts auf das eingebettete Objekt möglich.



Eng verbunden: Wenn für das Objekt der Auto-Knopf aktiviert wird, dann werden Veränderungen am Originalobjekt beim nächsten Start des Dokumentes selbstständig aktualisiert



Dies ist dann sinnvoll, wenn ein einheitliches Aussehen innerhalb eines Dokumentes gewünscht wird. Die vererbten Eigenschaften hängen von der Objektart ab. So kann ein Text die Schriftart, den Stil, die Farbe und Größe erben, während ein Bild Eigenschaften wie Rahmen- oder Hintergrundfarbe übernehmen kann.

Auch unter dem Stichwort *Programmierbarkeit* oder *OLE-Automation* findet sich die Fähigkeit von OLE wieder, auf andere Programme Einfluß zu nehmen. Eine OLE-fähige Anwendung kann darüber Befehle und Makros definieren und so eine andere Anwendung steuern. Die Konvertierung und Nachbildung von Objekten ist ein Mechanismus zum Austausch von Objekten auf verschiedenen Plattformen. So können Dokumente mit OLE-Objekten auch auf einem Apple Macintosh verarbeitet werden.

Hinter dem Begriff *Erweitertes Objektlayout* verbirgt sich die automatische Integration von OLE-Objekten in das bearbeitende Programm. So kann eine lange Tabelle an das Seitenlayout einer Textverarbeitung angepaßt und entsprechend umbrochen werden.

Suchen und Rechtschreibprüfung ist eine angenehme Eigenschaft von OLE 2.0: Eingebaute Objekte lassen sich mit den Werkzeugen des Hauptprogramms bearbeiten. Man kann also die eingefügte Tabelle eines Textdokuments nach Begriffen durchsuchen. Die Sache hat freilich einen Schönheitsfehler: Unter Windows 95 und Office 95 funktioniert dieser Mechanismus noch nicht.

Mit dem *Management von Versionen* soll OLE 2.0 Konflikte ausbügeln, die durch unterschiedliche Programmversionen und Inkompatibilitäten entstehen. So können einem eingebetteten Objekt auch der Name und die Versionsnummer des Erzeugerprogramms mitgegeben werden.

○ Das OLE-Modell

Jetzt ist klar, was OLE kann. Doch wie funktioniert dieses verflixte Gebilde eigentlich? Die Grundlage von OLE ist das Component Object Model (COM), welches die Aufrufe von Objekten unter Windows regelt. COM ist sehr eng mit der Windows-Architektur verbunden und dient als Basis für viele Entwick-

Blick in die Zukunft: Network OLE

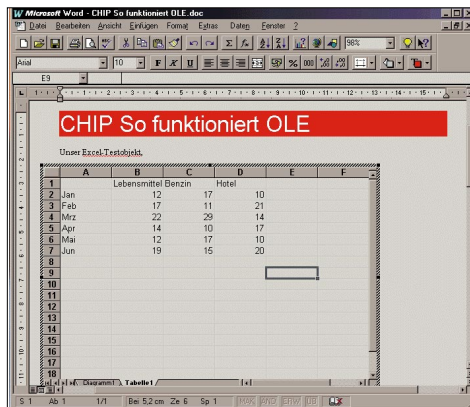
Mit der angekündigten Version von Windows NT 4.0 wird das OLE-Modell erweitert. Die Ausbaustufe läuft unter dem Stichwort „Distributed Component Object Model“ (DCOM), bisher „Network OLE“ genannt. Mit dieser Erweiterung können Applikationen die bekannten OLE-Funktionen auch über eine Netzwerkverbindung oder das Internet nutzen; dadurch ist es möglich, Objekte anderer Benutzer in Dokumente einzubauen, die automatisch aktualisiert werden. Prinzipiell ist DCOM ein Netzwerkprotokoll, das den Datenaustausch von Objekten in zwei Richtungen unterstützt. Nachvollziehbar war diese Funktion bisher auch schon per Remote-Procedure-Call (RPC)-Mechanismus; allerdings erforderte das einen erhöhten Programmieraufwand, weil die Netzwerkumgebung im Programm berücksichtigt werden mußte. DCOM hingegen setzt auf den bislang benutzten Werkzeugen für COM (Component Object Model) auf, die in einer Reihe von Programmiersprachen enthalten sind.

lungskits (C++, Delphi). Im COM ist auch die grundlegende Struktur für OLE-Objekte definiert: Jedes Objekt muß eine definierte Schnittstelle namens *IUnknown* haben, mit der die Lebensdauer des Objektes festgelegt und dessen Basis-eigenschaften abgefragt werden können. Natürlich reicht diese Schnittstelle für komplexe Objekte nicht aus, weshalb mit der Funktion *QueryInterface* weitere Schnittstellen erzeugt werden können.

OLE benutzt und verwaltet diese Schnittstellen. Hierüber läuft der ganze Transfer der Informationen, mit denen sich die Objekte in den Dokumenten ihre Inhalte zuschieben.

Letztendlich passiert nämlich nichts anderes als Datenaustausch: Schon beim gewohnten bequemen Drag-and-Drop-Mechanismus sagt die Anwendung, aus der ein Objekt gezogen wird, welche Struktur die Daten haben (etwa Bilddaten). Außerdem gibt sie die Dateninhalte mit. Diese Startanwendung wird „Server“ genannt, während das Empfangsdokument des Ziels „Container“ heißt.

Dazwischen sitzt OLE, überwacht Mausbewegungen und Tastatureingaben



Schaltautomatik:
Mittels der OLE-Technik
In-Place Editing können
die Objekte fremder
Programme gleich im
Dokument bearbeitet
werden. Das Umschal-
ten entfällt also.

In der Praxis ist die Funktionsvielfalt von OLE 2.0 unter Windows 95 nur teilweise umgesetzt. OLE erkennt, wenn die Datei auf dem gleichen logischen Laufwerk in ein anderes Verzeichnis gestellt wird, und führt die Verbindung nach. Der Vorgang mißlingt jedoch, wenn die Datei auf eine andere Festplatte oder auf ein Netzlaufwerk verschoben wird.

In diesem Zusammenhang darf man auf die nächste Version von OLE und die Umsetzung in den Betriebssystemen gespannt sein. Damit sollen auch beliebig geformte Objekte möglich sein – die bisherige Beschränkung auf Rechtecke entfällt also.

Jörg Lorenz

und teilt der Anwendung mit, was da gerade überbracht wird. Die Zielapplikation hat zwei Möglichkeiten: Entweder übernimmt sie die Daten und Methoden (Embedding). Hierbei dient das Dokument der Zieldanwendung nur als Speicher, und der Server wird automatisch aktiviert, wenn das Objekt bearbeitet wird. Wenn jedoch die Daten umfangreich sind oder von anderen Applikatio-

nen aktualisiert werden, dann bietet sich die zweite Möglichkeit: Die Daten verbleiben in einer anderen Quelle, und nur ein Verweis (Moniker) wird in das Dokument eingebunden (Linking). Der Moniker enthält eine Reihe Informationen, da die Quelle wiederum ein Dokument oder sogar ein Dokument innerhalb eines Dokuments sein kann. Die Nachführung des Monikers ist Aufgabe von OLE.



Adressen: Microsoft GmbH,
Edisonstr. 1, 85716 Unterschleißheim, Tel.: 089/3176-0

Literatur: Microsoft System Journal, Heft 1,
Jan/Feb 94, Darf ich vorstellen: OLE 2.0,
Seiten 6ff., Kraig Brockschmidt

OLE 2.0, A View From Orbit, Paper Microsoft
PDC 93, Vortrag, Kraig Brockschmidt

CHIP-Sonderheft Windows 95, OLE 2.0 –
Objektiv verbunden, Seite 46ff., Jörg Lorenz