di Francesco Petroni

DUE O TRE COSE SU OLE AUTOMATION

Questo articolo costituisce la continuazione di quello, intitolato Ragionare per Oggetti, pubblicato nel numero 166 di MC. Parleremo un po' più a fondo della tecnologia OLE Automation che rappresenta il gradino più alto nella programmazione Object Based. Chiariamo subito che si tratta di un argomento complesso, soprattutto per due motivi: il primo è che, riguardando due o più prodotti utilizzati contemporaneamente, pretende un'ottima conoscenza dei vari applicativi coinvolti. Il secondo motivo è che sia le librerie, che servono per far funzionare OLE Automation, così come la documentazione, anche quella in forma di file Help, sono difficilmente reperibili, in quanto non sono quasi mai presenti nei prodotti di base.

OLE e OLE Automation

OLE, Object Linking and Embedding, è una tecnologia che permette di integrare informazioni provenienti da applicazioni differenti. Il suo utilizzo più immediato e semplice consiste nella realizzazione dei Compound Document, documenti unici realizzati con un applicativo (OLE Container) in cui vengono appunto assemblati oggetti provenienti da altri applicativi (OLE Objects).

Questo è l'utilizzo più semplice, alla portata di qualsiasi utente, e si basa sullo sfruttamento del comando di menu Inserisci Oggetto, presente in tutti i prodotti "contenitori". Rientra però nel concetto, più generale, del Component Object Model (COM), che raccoglie, oltre ai Compound Document, anche OLE Automation e OLE Controls.

OLE è anche la modalità standard utilizzata dalle applicazioni che "espongono" propri oggetti e che permettono la loro manipolazione da parte di procedure scritte con i linguaggi presenti in altre applicazioni. Tali oggetti si chiamano OLE Components e la loro tipologia varia in funzione del tipo di applicazione e all'interno della stessa applicazione, che può esporre più tipi di oggetti.

Ad esempio in un Word Processor ci può essere l'oggetto Documento, l'oggetto Paragrafo, l'oggetto Frase, in uno Spreadsheet, l'oggetto Foglio, l'oggetto Grafico, l'oggetto Intervallo, l'oggetto Cella. In OLE Automation esistono quindi da una parte degli OLE Components e dall'altra non un Container, ma una procedura scritta con un linguaggio di programmazione adeguato (lo standard è il Visual Basic for Application) presente in un'altra applicazione, e che svolge il compito di OLE Controller.

Esercizi di OLE Automation

Per fare i nostri esercizi servono una serie di file, che, per comodità, abbiamo

elencato in figura 2. I file più importanti sono le librerie che svolgono le funzioni di intermediario tra l'applicazione Controller e l'applicazione che espone i propri Components. Queste librerie non si trovano necessariamente nel prodotto. Quelle di Excel e di PowerPoint si trovano nei rispettivi pacchetti, quelle di Word e di Access no. Altra complicazione è l'esistenza di ulteriori librerie, specifiche di una coppia di prodotti. Ad esempio la libreria Wr95acc.Tlb serve solo per gestire, da Access, oggetti Word.

Oltre alle Librerie è necessaria la loro documentazione. Se si trovano le librerie è probabile che si trovino anche i file di Help, nei quali trovare perlomeno notizie sugli Oggetti "esposti" dall'applicazione controllata e sulle loro Proprietà, Eventi e Metodi (in figura 1 vediamo l'Help di MS Graph 5.0, accessorio che espone propri Objects).

La documentazione è indispensabile in quanto la sintassi di OLE Automation non è affatto intuitiva, anzi, nel caso di utilizzo di un oggetto e/o una proprietà gerarchicamente complessi, risulta anch'essa complessa.

Altro materiale che è bene avere a disposizione è una collezione di file con i dati, realizzati con i vari prodotti da usare come Components, ben consolidati e ben conosciuti, da utilizzare nelle prove che non lasciano margini di incertezza ed il risultato delle quali è abbastanza certo.

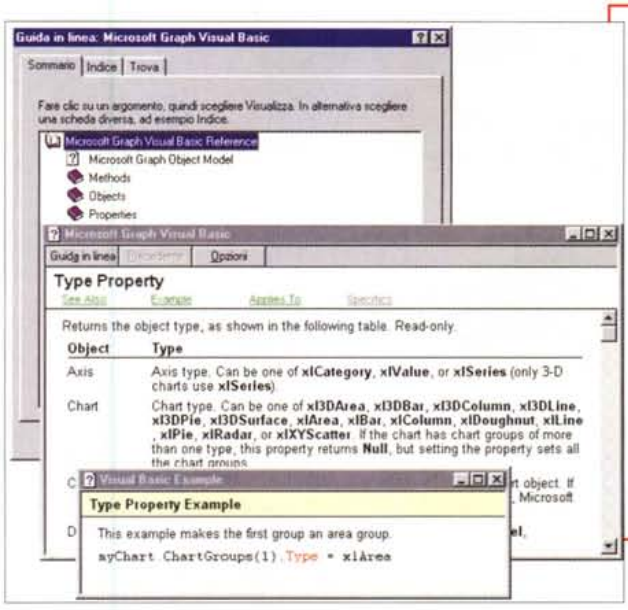


Figura 1 - OLE Automation - Un esempio di Help per OLE Automation.
Per praticare OLE Automation occorre disporre delle Librerie che consentono ad un prodotto, OLE Automation compatibile, di essere gestito per oggetti (OLE Objects o OLE Components) da un'applicazione esterna (OLE Controller). Occorre disporre anche della documentazione che illustra quali siano gli oggetti gestibili, le loro proprietà, i loro eventi e i loro metodi. Qui vediamo l'Help dell'accessorio MS Graph 5.0 (il file si chiama VBA_GRP.HLP) in cui sono documentati, in maniera fin troppo sintetica, tutti questi elementi.

Il contenitore OLE

Capito il concetto di Compound Document e di OLE Object (basta fare un po' di esperimenti con il comando Inserisci Oggetto di WordPad), il passo successivo può essere quello di sperimentare il controllo OLE Container, presente in Visual Basic ed in Access. Vediamone le principali proprietà, anche per capire i vari aspetti coinvolti in OLE:

Esercizi

Cartella	File	
\NEWOLE	Es01.Vbp	Primo esercizio VB
	Es01.Frm	
	Es02.Vbp	Secondo esercizio VB
	Es02.Frm	
	Es03.Xls	Terzo esercizio Xls
	Es04.Vbp	Quarto esercizio VB
	Es04.Frm	

Materiale

Cartella	File	
\DATI	Inferno.Doc	Documento Word
	Ole.Xls	Foglio Excel
	Ole.Ppt	Presentazione PowerPoint
	DbOle.Mdb	Database Access
\LIBRERIE	Wb70en.Tlb	Word 7.0
	Wd95acc.Tlb	Word da Access
	Wd95ppt.Tlb	Word da PowerPoint
	Ex5en32.Olb	Excel 7.0
	Powerpnt.Tlb	PowerPoint 7.0
	Msaccess.Tlb	Access 7.0
	Gren50.Olb	MS Graph 5.0
	Pj4en32.Olb	Project 7.0
	Sqlole32.Tlb	SQL
	Binder.Tbl	Office Binder
\HELP	Wrbasic.Hlp	Word 7.0
	Vba_xl.Hlp	Excel 7.0
	Vba_pp.Hlp	PowerPoint 7.0
	Vba_acc.Hlp	Access 7.0
	Vba_grp.Hlp	MS Graph 5.0
	Vba_pj.Hlp	Project 7.0
	Sqlole.Hlp	SQL
	Vba_bin.Hlp	Office Binder
	Vba_off.Hlp	Office
	Vbait2.Hlp	Generale

Figura 2 - Esercizi di OLE Automation - Aspetti organizzativi.
*Per poter svolgere i nostri esercizi occorre preparare una serie di file con i dati e occorre disporre di alcune librerie (file con desinenze *.DLL, *.TLB, *.OLB). È opportuno separare, in cartelle differenti, i file Programmi dai file Dati, da quelli con le Librerie e dai file di Help che useremo per documentarci. Lavoreremo con i prodotti della Microsoft, OLE Automation e Visual Basic for Application compatibili, che sono Visual Basic, Excel, Word, Access e PowerPoint. Da notare il fatto che alcuni prodotti sono solo OLE Controller, come Visual Basic, altri solo OLE Objects, come PowerPoint, altri, come Excel, possono svolgere ambedue le funzioni.*

Action - determina l'operazione da eseguire su un oggetto, ad esempio Embed, o Link, o Insert Object, ecc.;
Class - identifica l'applicazione che supporta l'oggetto e il tipo di oggetto utilizzato (esempio Excel.Sheet, esempio Word.Document);
Display Type - si può visualizzare il contenuto dell'oggetto oppure un'icona che ne specifica il tipo. Ad esem
pio un oggetto con un'Immagine può essere visualizzato come Figura o come Icona, un oggetto Suono può essere solo visto come Icona;
OLETypeAllowedil - tipo di OLE (Link, Embed o entrambi) possibile nel contenitore;
SourceDoc - determina il nome del file in caso di Link;
SourceItem - determina l'insieme dei dati, o il dato, che costituisce il Link (es. un Intervallo di uno Spread
sheet, un Segnalibro in un Documento di testo);
Size Mod - indica in quale modo un oggetto viene visto all'interno della sua frame (ritagliato, stirato, zoomato);
Update Option - manuale o automatica (in caso di Link, quando cambia il file dell'oggetto);
Verb - quale metodo viene richiamato sull'oggetto al momento della sua attivazione.

Un esercizio molto semplice da eseguire consiste nell'inserire in un Form un OLE Container e nel "riempirlo" con un oggetto che intendiamo studiare a fondo. Poi si può realizzare una piccola procedura che evidenzia, in qualche ma-

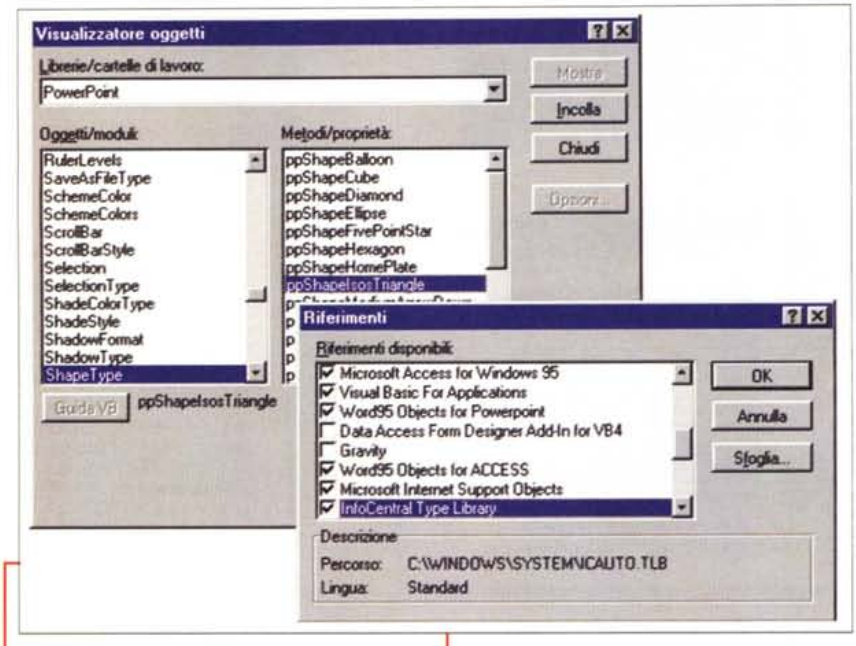


Figura 3 - OLE Automation - Caricamento delle Librerie e Visualizzazione degli Oggetti.
Queste sono le due finestre, presenti in qualsiasi ambiente di programmazione (Macro di Excel, Codice Visual Basic, Procedure di Access, ecc.), che servono la prima a verificare la presenza delle librerie (attraverso le quali si agisce sugli OLE Objects messi a disposizione dagli altri applicativi) e a caricarle e la seconda a selezionare Oggetti, Proprietà e Metodi dei vari oggetti.
Purtroppo questa seconda finestra non permette di confezionare in modo automatico istruzioni "lunghe", in cui, ad esempio, occorra impostare una gerarchia complessa per identificare l'oggetto. Attenzione, non tutte le librerie vengono installate con i prodotti "normali". Alcune sono presenti solo nel Microsoft Solutions Development Kit versione 2.0.

OLE Objects creati tramite OLE Automation

I pulsanti inseriti nel Form Visual Basic (in alto in figura 7) servono per creare dal nulla nuovi oggetti OLE. Sul Form abbiamo inserito anche due Caselle di Testo nelle quali scriviamo due parole o frasi che vengono, tramite OLE Automation, passate al documento creato con la procedura.

Passiamo in rassegna i pulsanti:

Word Open (legato al pulsante P1). In pratica provoca l'apertura di un file esistente, così come visto nell'esercizio precedente,

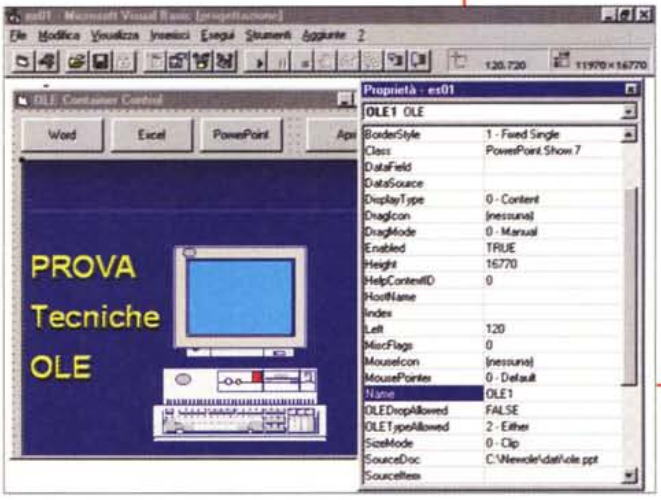


Figura 4 - OLE Automation - Con Visual Basic automatizziamo Word, Excel e PowerPoint - (1). Per il nostro primo esercizio utilizziamo il controllo OLE Container di Visual Basic, che permette non solo di inserire "al volo" un OLE Object (lo si può fare sia in fase di sviluppo che in fase di esecuzione), ma anche di gestirlo dinamicamente da programma, per modificarne alcune caratteristiche. In questo caso è necessario conoscere alla perfezione le proprietà tipiche del controllo OLE Container e i metodi che agiscono su di esso.

niera, tutte le proprietà assunte da controllo.

Riempimento di un OLE Control

Il primo esercizio "ufficiale" lo realizziamo con il Visual Basic. In un Form inseriamo un Controllo OLE Container, che riempiamo, a seconda del pulsante clickato, con un documento Word, con un foglio Excel, con una presentazione PowerPoint, i cui file siano già stati realizzati e quindi siano già disponibili.

In pratica eseguiamo, via programma, la stessa operazione che eseguiamo quando, lavorando in comandi diretti, lanciamo il comando Inserisci Oggetto, Crea da File, presente in tutte le applicazioni OLE Client.

In figura 4 vediamo la presentazione PowerPoint e, sulla destra, il Pannello delle Proprietà nel quale si possono verificare tutte le proprietà dell'oggetto OLE Container. In figura 5 vediamo l'oggetto Word, si tratta di una nostra composizione... poetica. Vediamo sia il Form Visual Basic che l'applicazione Client Word con il documento aperto. Il documento è stato aperto sfruttando il Metodo Action = 7 eseguito sull'OLE Container.

Il listato, composto da quattro brevi routine, è in figura 6. Tre servono per l'apertura dei tre file, e il quarto serve per aprire l'applicazione Component.

Word Insert (pulsante P2). In questo caso viene creato un nuovo documento che contiene come primi due paragrafi le frasi scritte nelle due Caselle di Testo, T1 e T2.

Excel (pulsante P3) fa la stessa cosa su un foglio Excel. Le due frasi vengono scritte in due celle.

Il pulsante P4 serve per aprire un file MDB di **Access**, preesistente, e per aprire una sua tabella.

Con il pulsante P5, infine, si crea una Slide in una Presentazione **PowerPoint**, della tipologia scelta tra quelle disponibili, e con le due frasi scritte, sempre via OLE Automation, sulla Slide.

Tra le varie istruzioni riferibili ad un

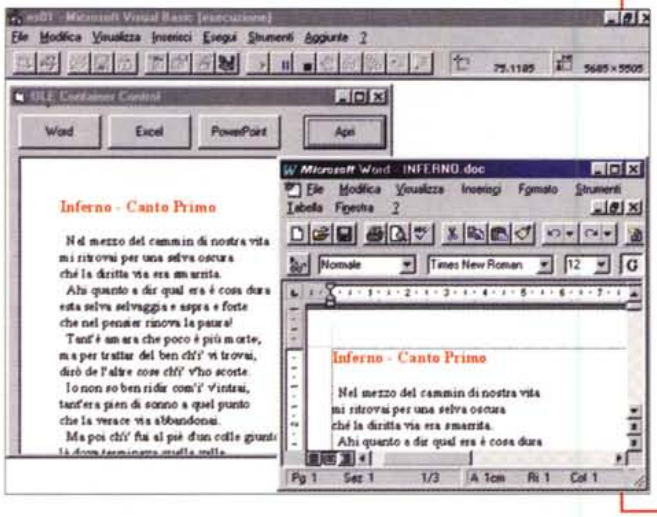


Figura 5 - OLE Automation - Con Visual Basic automatizziamo Word, Excel e PowerPoint - (2). Il Metodo più interessante, tra quelli tipici del controllo OLE, è Action, che serve a specificare quale azione debba essere attivata sul controllo OLE. Ad esempio si può provocare l'apertura dell'applicazione Server alla quale appartiene l'oggetto. Altre azioni servono per creare ex novo l'oggetto, attivando la finestra Inserisci Oggetto, altre ancora, altrettanto interessanti, per salvare l'oggetto OLE, sia come file binario, sia come oggetto OLE.

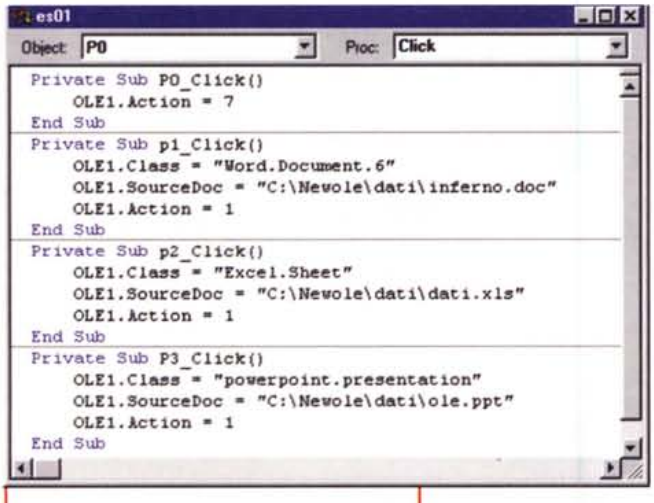


Figura 6 - OLE Automation - Con Visual Basic automatizziamo Word, Excel e PowerPoint - Listato.
Una volta inserito nel Form il controllo OLE, bastano tre istruzioni per "riempirlo" con un OLE Object. Occorre definire la proprietà Class, che identifica l'applicazione che supporta l'oggetto, es. Excel.Sheet, e la proprietà SourceDoc, che identifica il nome del file. Il metodo Action = 7 è quello che, in pratica, dà vita all'oggetto.

generico oggetto OG (già creato con il comando CreateObject) citiamo quelle che servono per portare in primo piano la sua finestra, oppure per nasconderla, per massimizzarla, per chiuderla:

```
OG.AppMaximixe
OG.AppClose
OG.Visible = True
ecc.
```

Insomma tutti i prodotti si comportano nella stessa maniera, nel senso che una volta definito un loro Object, molti dei comandi OLE Automation sono in comune, e quindi indipendenti dal tipo di oggetto. Gli altri dipendono dal tipo specifico di Oggetto.

In pratica con questo sistema è possibile creare un Documento Word, un Foglio Excel, una Presentazione PowerPoint, direttamente da fuori.

Access si comporta differentemente. Nell'esempio non abbiamo fatto altro che attivare un database MDB ed aprire una sua tabella. In pratica non abbiamo creato nulla.

Relativamente a OLE, Access presenta altri aspetti interessanti, come il famoso campo di tipo OLE, e la possibilità di essere gestito tramite oggetti OLE. Stiamo preparando un articolo di approfondimento su Access e OLE, in quanto si tratta di un ambito di utilizzo più particolare.

Tornando al nostro esercizio va detto, per completezza di esposizione, che c'è un altro sistema, più rudimentale e pericoloso, per comandare da un programma esterno una qualsiasi applicazione, anche non programmabile, anche non OLE Automation Compatibile. Si tratta del comando SendKeys, con il quale si può, a proprio rischio e pericolo (in quanto senza nessuna forma di controllo e di intercettazione degli errori), simulare l'esecuzione di comandi ed operazioni da tastiera, all'interno dell'applicazione controllata.

Figura 7 - OLE Automation - Piccolo campionario di procedure OLE Automation "esterne".
Nel secondo esercizio sperimentiamo le poche istruzioni fondamentali con le quali si realizza "dal nulla" un lavoro con un qualsiasi prodotto che "espone" i propri oggetti. Anche in questo caso usiamo il Visual Basic come Controller e vari altri applicativi come OLE Objects. Nella figura vediamo il risultato raggiunto creando un oggetto Access.Application nel quale abbiamo aperto una tabella con i Dati.

OLE Automation, tecnologia ben più nobile, permette di utilizzare più comandi, non solo quelli di tastiera, e di controllare passo passo l'esecuzione del programma.

PowerPoint un po' più a fondo

Guardate la figura 9. Rappresenta una slide PowerPoint pesantemente riempita di oggetti grafici, ognuno dei quali ha una sua forma, un suo colore e un suo motivo di riempimento. Questa

```
Dim WB As WordBasic
Dim XL As Excel.Application
Dim AC As Access.Application
Dim PPa As PowerPoint.Application, PPp As Presentation, PPs As Slide
Private Sub P1_Click()
    Set WB = CreateObject("Word.Basic")
    WB.FileOpen Name:="c:\newole\dati\inferno.doc"
    WB.AppMaximize
    WB.FilePrintPreview: WB.ClosePreview
    WB.AppClose: MsgBox "Eccoci"
End Sub
Private Sub P2_Click()
    Set WB = CreateObject("Word.Basic")
    WB.AppMaximize: WB.FileNewDefault
    WB.Insert T1: WB.InsertPara: WB.Insert T2: WB.InsertPara
End Sub
Private Sub P3_Click()
    Set XL = CreateObject("Excel.Application")
    XL.Workbooks.Add: XL.Visible = True
    XL.Cells(1, 1).Value = T1.Text: XL.Cells(2, 1).Value = T2.Text
    XL.Application.Quit
End Sub
Private Sub P4_Click()
    Set AC = CreateObject("Access.Application")
    AC.OpenCurrentDatabase ("c:\newole\dati\dbole.mdb")
    AC.Visible = True: AC.DoCmd.Maximize
    AC.DoCmd.OpenTable "Dati"
End Sub
Private Sub P5_Click()
    Set PPa = CreateObject("Powerpoint.Application")
    PPa.AppWindow.Visible = True
    Set PPp = PPa.Presentations.Add
    Set PPs = PPp.Slides.Add(1, 9)
    PPs.Objects(1).Text = T1: PPs.Objects(2).Text = T2
    PPs.Objects(1).Text.CharFormat.Shadow = True
    PPs.Objects.AddShape ppShapeDiamond, 2500, 5000, 3000, 3000
    PPs.Objects(3).GraphicFormat.Fill.BackColor.RGB = RGB(255, 255, 0)
    Quit
End Sub
```

Figura 8 - OLE Automation - Piccolo campionario di procedure OLE Automation - Listato.
Uno dei problemi di OLE Automation è il fatto che si lavora "a cavallo" tra due applicazioni. Occorre quindi stare molto attenti ad "indirizzare" correttamente le istruzioni, quelle che vanno all'applicazione controllata e quelle che agiscono all'interno dell'applicazione controllante. Ci saranno poi delle istruzioni per il passaggio dei dati da una parte all'altra. Rientra in questo discorso la gestione delle finestre, ad esempio la decisione su cosa far vedere durante l'esecuzione del programma.

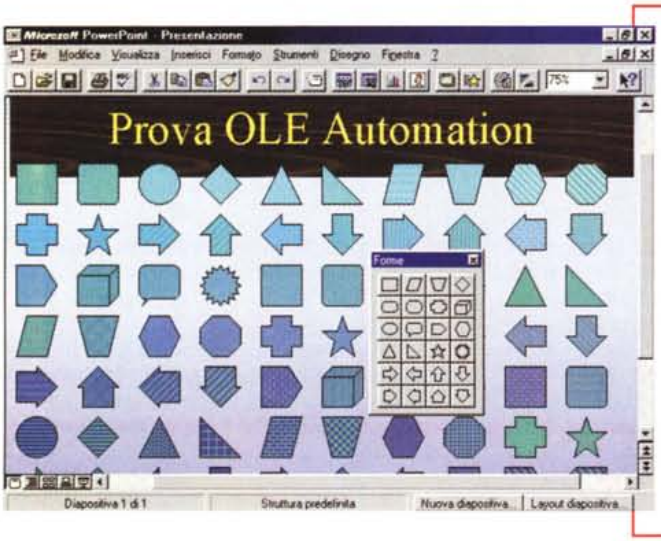


Figura 9 - OLE Automation - Divertimento con PowerPoint, da Excel.
Questa bruttissima Slide, realizzata con PowerPoint, è stata completamente costruita via OLE Automation con una Macro di Excel. Una caratteristica di PowerPoint è quella di "esporre" propri OLE Object, ma di non essere un OLE Controller. Buona parte delle istruzioni OLE Automation corrispondono ai normali comandi diretti di PowerPoint. Ad esempio il metodo AddSlide accetta come parametro un numero che corrisponde ad una precisa tipologia di Slide. Le tipologie sono le stesse proposte quando, usando normalmente il prodotto, si aggiunge una nuova Slide.

slide è stata totalmente realizzata attraverso OLE Automation con una Macro VBA di Excel.

In realtà il programma è stato inizialmente realizzato con Visual Basic. Il listato è stato poi completamente copiato in una Macro VBA di Excel. L'unica modifica necessaria è stata quella dell'istruzione che serve a caricare il titolo da una cella Excel.

Si può notare come buona parte delle istruzioni con le quali vengono creati e maneggiati oggetti di PowerPoint corrispondono ai suoi comandi interni. Ad esempio quando, usandolo normalmente, si crea una nuova Slide, PowerPoint propone un campionario di tipi. Il comando VBA che crea una nuova slide dispone di un parametro che indica proprio il tipo di slide. Lo stesso discorso vale per tutti gli altri comandi.

Va notata l'estrema (per non dire eccessiva) complessità dei comandi che servono per definire le proprietà di un oggetto. La complessità deriva dalla complessità insita nella definizione degli oggetti e delle loro proprietà.

Ad esempio per indicare che il motivo del riempimento di un certo oggetto (l'oggetto numero x nella slide) è pieno o solido, il comando è:

NomeSlide.Ojects(x).GraphicFormat.Fill.BackColor.SchemeColor=ppFillColor

Un comando del genere non presenta nessuna difficoltà concettuale, ma è difficile da confezionare senza un buon Help in linea o un manuale ben documentato.

Il listato completo dell'esercizio, che si esegue come Macro VBA da Excel, è in figura 10.

Azioni eseguibili mediante un OLE Control

Nell'ultimo esempio applichiamo OLE Automation ad un oggetto inserito in un OLE Container di Visual Basic. L'OLE Container contiene un oggetto Graph, realizzato con MS Graph 5.0, l'OLE Server condiviso tra Word, PowerPoint, Access e Project, che potete trovare nelle cartelle condivise (Shared) installate con MS Office 95.

MS Graph 5.0 espone propri oggetti che possono essere manipolati dal di fuori, anche se l'oggetto MS Graph 5.0 risiede in un OLE Container di Visual Basic.

Detto in altre parole è possibile gestire da fuori, via OLE Automation, tutte le caratteristiche estetiche e contenutisti-

che del diagramma inserito nell'applicazione VB.

Nel nostro esempio (vediamo in figura 11 l'output e in figura 12 il listato) tramite una serie di Pulsanti scorriamo i vari tipi disponibili di Grafico, ne definiamo, scrivendolo in una Casella di Testo, il Titolo, facciamo sparire o apparire la Legenda. Infine, se il grafico è di tipo 3D, lo facciamo ruotare di 360 gradi realizzando una specie di animazione.

Insomma un significativo campionario di istruzioni OLE Automation.

I reali utilizzi di OLE Automation

VBA ed OLE Automation costituiscono una tecnologia ambiziosissima grazie alla quale Windows diventa un unico ambiente operativo, nel quale possono agire programmi trasversali, che sfruttano oggetti forniti dai vari prodotti, a seconda delle necessità dell'applicazione e di cosa il singolo prodotto può dare in termini di oggetti.

OLE Automation è, una volta rispettati i prerequisiti in termini di disponibilità di librerie e di documentazione, praticabile con successo.

Da qui a dire che tale tecnologia può essere usata produttivamente e in modo affidabile in un'applicazione "chiusa" ce ne passa.

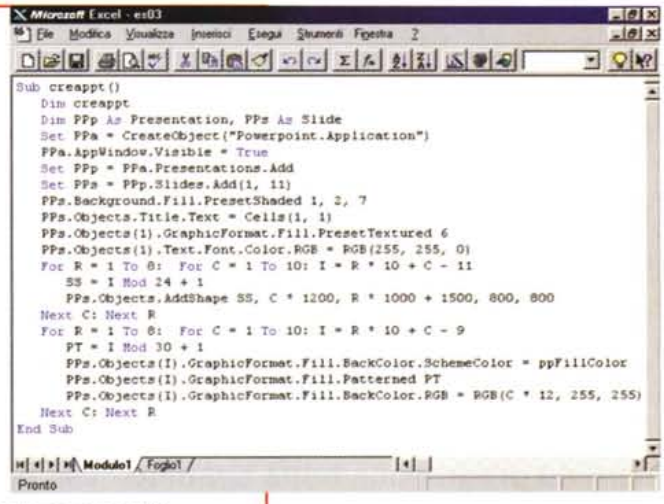
Il pericolo maggiore consiste nella complessa struttura (fatta di programmi e librerie) sottostante l'applicazione, che deve essere comunque presente e stabile quando la si usa, soprattutto su macchine che non siano quella su cui l'applicazione stessa è stata sviluppata.

È però probabile che tra pochi mesi, quando, anche sotto la spinta poderosa delle necessità di Internet, sarà ulteriormente messa a punto la tecnologia COM (Components), OLE Automation sarà più sicura e quindi sfruttabile in applicazioni alle quali si chieda anche produttività ed affidabilità.

Nel prossimo numero parleremo di Access e OLE, o meglio dei vari modi di sfruttare OLE in un Database realizzato in formato MDB di Access.

MC

Figura 10 - OLE Automation - Divertimento con PowerPoint, da Excel - Listato della Macro.
Per il Primo Principio Fondamentale dell'OLE Automation (chissà se esiste veramente) il listato di un programma OLE Automation è indipendente dal prodotto Controller con il quale è stato scritto. Questo programma inizialmente lo abbiamo scritto con il Visual Basic, poi lo abbiamo copiato in una Macro di Excel. L'unica modifica apportata dopo la copia è stata quella che serve per leggere il Titolo della Slide da una cella del foglio di Excel anziché da una variabile VB.



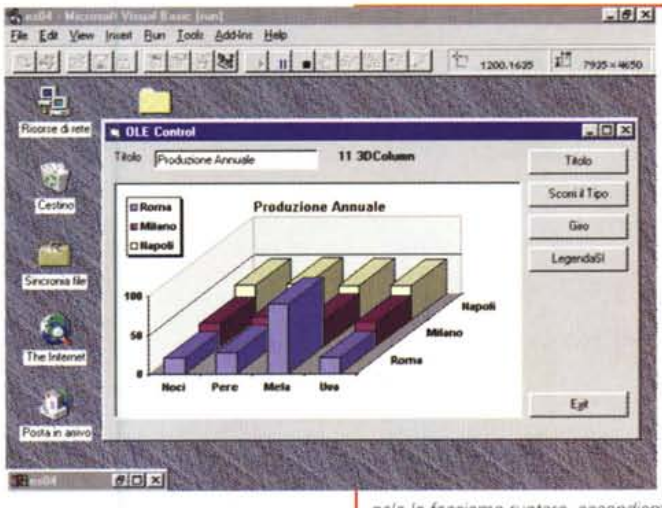


Figura 11 - OLE Automation - Azioni sul contenitore OLE - Output.
In questo ultimo esercizio utilizziamo un OLE Object costituito da un grafico realizzato con MS Graph 5.0 (il cui Help, ricordiamolo, è visualizzato in figura 1). Sul Form abbiamo inserito una serie di pulsanti con i quali vengono lanciate delle piccole "routinette" OLE Automation che modificano alcune delle proprietà del grafico. Inseriamo il Titolo, sperimentiamo, scorrendoli tutti, i vari tipi di diagramma, se il grafico è tridimensionale lo facciamo ruotare, accendiamo o spegniamo le legende.

Figura 12 - OLE Automation - Azioni sul contenitore OLE - Listato.
Questi sono i listati associati ai vari eventi Click sui vari pulsanti. Anche in questo caso la sintassi non è intuitiva. Le istruzioni si possono scrivere solo se si dispone di un Help o di una documentazione adeguata. L'Help del Visual Basic del Graph 5.0 è, a mio parere, troppo semplificato e del tutto privo di esempi significativi. OLE Automation ci fa faticare.