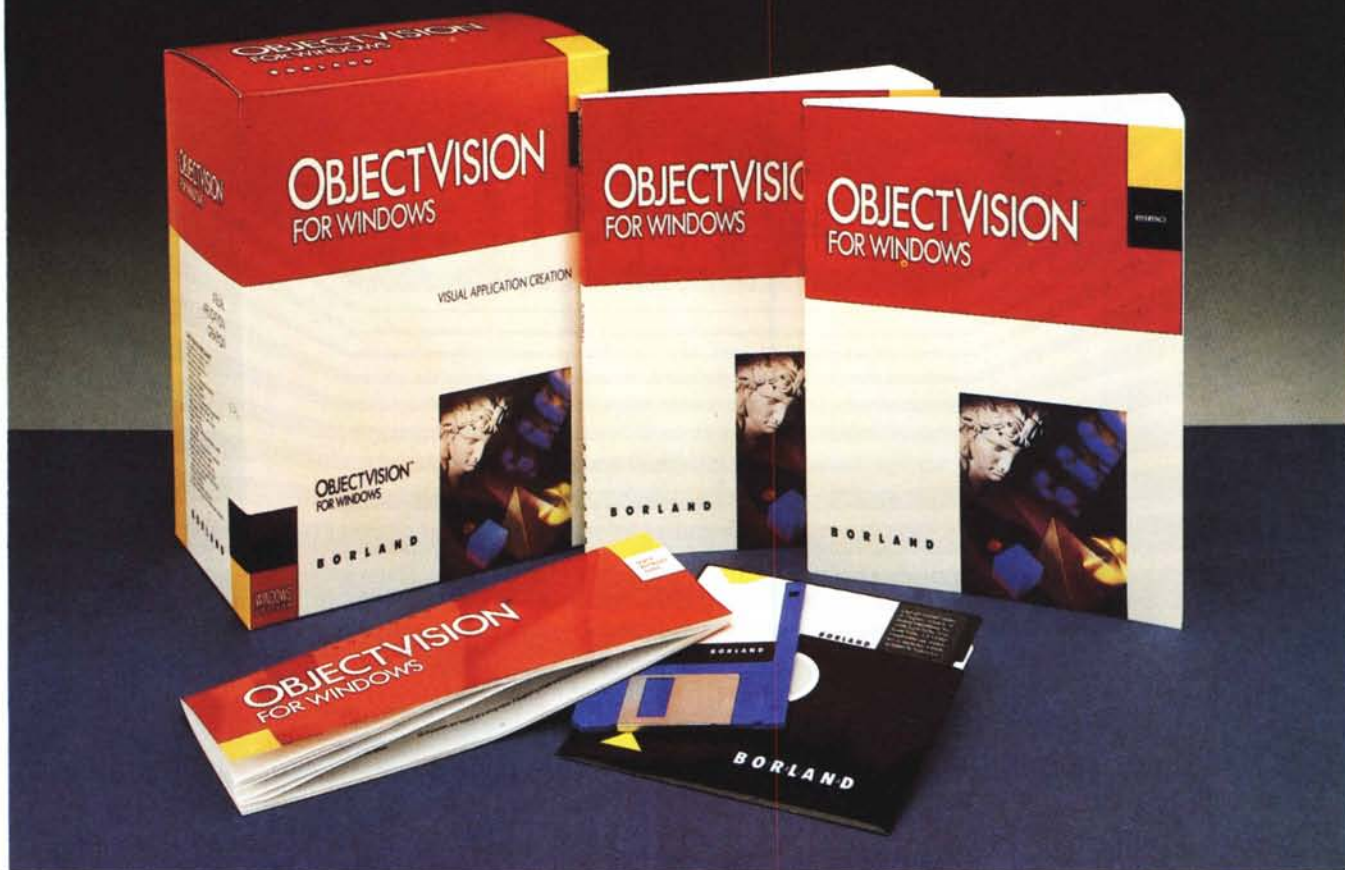


PROVA



# Borland Object Vision

di Francesco Petroni

**U**no dei ricordi più vivi legati all'appena trascorso 1990 sarà sicuramente la videocassetta che ha accompagnato il lancio, eseguito a livello mondiale, di Windows 3.0.

Tra le numerose «partecipazioni speciali» è stata significativa quella di Philippe Kahn della Borland che ha... recitato una parte brillante fingendo dapprima un certa perplessità di fronte alla novità, poi, più seriamente, nell'ultimo intervento, dichiarando l'adesione della sua Borland allo standard Windows e citando come prodotti che avrebbero seguito lo standard il Quattro Professional e il Paradox.

Invece il primo prodotto Borland per Windows è questo Object Vision che ci era pervenuto, inaspettatamente, almeno per me, in una versione Beta Con-

fidenziale già alla fine di dicembre. Borland Object Vision (lo chiameremo OV) rappresenta una novità interessante per numerosi motivi.

Il primo è quello, già citato, di essere il primo prodotto per Windows realizzato dalla Borland a conferma delle affermazioni di Kahn.

Il secondo è che si tratta di un prodotto di cui non si era assolutamente sentito parlare in giro, neanche a livello di indiscrezione e questo, nel mondo del software, è abbastanza inusuale.

Il terzo motivo di interesse è che si tratta di un prodotto originale, nel senso che non si può incasellare in nessuno dei grandi filoni in cui si possono categorizzare i prodotti dell'Informatica Individuale.

Il quarto motivo è che Object Vision

si avvicina ulteriormente al concetto, forse utopistico, di prodotto fondato totalmente su tecniche dichiarative, in cui non si programma e non si può programmare (la tecnica contrapposta è quella procedurale) ma si interagisce con l'ambiente e si utilizzano gli strumenti che questo mette via via a disposizione.

In questa ottica è anche corretto che Object Vision non disponga di un proprio formato per gli archivi dei dati.

Object Vision si mette infatti «a servizio» dei due DBMS più diffusi per MS DOS (dBASE e Paradox) e di tre altri formati, per incaricarsi di una serie di compiti che tali prodotti non possono compiere se non attraverso «procedure» tradizionali.

In concreto con dBASE o Paradox o

Basic, ecc. è facile costruire le strutture degli archivi. Non è altrettanto facile realizzare delle procedure di acquisizione, precalcolo, controllo e validazione dei dati, che garantiscano la correttezza dei dati inseriti. Object Vision può coprire questi vuoti.

### Che cosa è Object Vision

OV serve per realizzare Forms, ovvero Maschere di acquisizione dati, che da una parte abbiano sul video le caratteristiche estetiche di un Modulo Cartaceo, che possa anche essere stampato con qualità tipografica, e dall'altra alimentino direttamente un archivio «tradizionale» impostato con un comune prodotto di tipo DBMS.

La costruzione del Form avviene esclusivamente maneggiando Oggetti, che corrispondono alle differenti tipologie di elementi utilizzati ed ognuno dei quali ha proprie specifiche caratteristiche.

Per costruire un Form si opera in quattro ambienti differenti, chiamati Tools, ognuno dei quali dispone di un proprio specifico menu, in cui però sono sempre presenti alcune voci in comune con gli altri tre.

Il primo è il Tool FORM con il quale si realizza, attraverso un sofisticato Editor Grafico, perfettamente integrato nell'ambiente Windows, il Modulo sul video.

Il secondo è il Tool TREE, che serve per sviluppare in forma grafica e sul video il processo di calcolo e/o decisionale eventualmente sottostante il riempimento di ciascun campo.

Il terzo Tool è lo STACK che permette di organizzare il flusso dei dati nel caso questo debba avvenire su più Form successive e collegate logicamente tra di loro.

Infine l'ultimo Tool, il LINK, che realizza il collegamento tra il Form disegnato con Object Vision ed un archivio esterno di formato proprio. Object Vision può sia agganciarsi ad un archivio preesistente, sia creare direttamente la struttura esterna.

Insomma all'Object Vision si può delegare la funzione di interfaccia logica, di calcolo e grafica per archivi dati che sono creati e rimangono nel loro formato originario, che può essere un ASCII puro, Paradox, dBase, Btrieve o, non dimentichiamoci che stiamo lavorando sotto Windows, il Dynamic Data Exchange di Windows.

Più in là nell'articolo approfondiremo questo aspetto che è il nocciolo tecnico del prodotto.

Object Vision può essere utilizzato a

### Borland Object Vision

**Produttore:**  
Borland International, Inc. 1800 Green Hills Road P.O. Box 660001 Scotts Valley, CA 95067-0001

**Distributore:**  
Borland Italia S.r.l.  
Via Cavalcanti, 5 - 20127 Milano  
Tel. 02-2610102

**Prezzi (IVA esclusa):**  
Borland Object Vision L. 949.000  
(offerta valida fino al 31/05/91, L. 299.000).

### Il pacchetto e la sua installazione

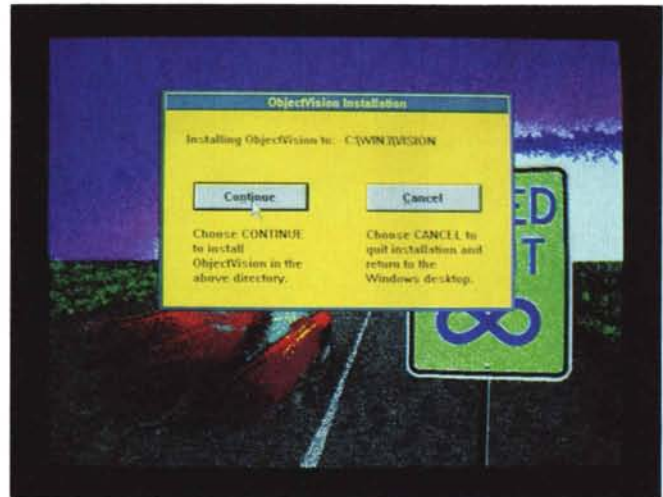
Il package contiene due manuali. Il TUTORIAL, di circa 160 pagine, che serve ad introdurre il prodotto, ad insegnare ad usarlo, tramite degli esercizi condotti con dei file caricati in fase di installazione, e il REFERENCE, più corposo (260 pagine) che tratta in maniera sistematica le varie funzionalità.

L'installazione è molto semplice in quanto l'intero prodotto è ospitato su un unico dischetto da 1,44. Si lancia dalla finestra Run di Windows e comporta l'unica richiesta del nome della directory nella quale scaricare il materiale. La fase d'installazione è allietata da una videata molto spettacolare (fig. 1).

### I quattro Tools

Passiamo ora in rassegna i quattro Tools a disposizione del programmatore. Questo è sicuramente il miglior modo per descrivere il prodotto.

Figura 1 - Borland Object Vision - Installazione da Windows. OV occupa un solo dischetto, che contiene, oltre al programma, anche una serie di esempi interessanti (che vedete in parte in alcune foto). L'installazione, durante la quale appare una suggestiva immagine di genere fantastico, si lancia dalla finestra RUN di Windows 3, ed è del tutto automatica, richiede solo il nome desiderato per la sub-directory.



Prova Object Vision Microcomputer (Goal)		
Le varie tipologie di Campi		
Tipo generale	testo o numero libero	<input type="text" value=""/>
Tipo Fissa	12233555,00	Selection List
Tipo Percentuale	12,342	Grande
Tipo Finanziario	123.000.000	Grandissimo
Tipo Currency	L. 300.000.000,000	Medio
Tipo Date/Time 1 2	07 April 1990 7.40.48 AM	Piccolo
Tipo Scrolling	<input type="checkbox"/> Tipologia True/False	Tipo Check Boxes
questa e' una Finestra scorrevole che contiene un testo e dispone di	Tipo Picture es. (&&&&&&##&&&&##&&&&) per il codice fiscale	<input checked="" type="checkbox"/> Maschio <input type="checkbox"/> Femmina
		ABCDEF-01A01-H5012

Figura 2 - Borland Object Vision - Tipologie di Campi. Nel definire un campo, successivamente alla definizione del nome, vanno indicati la tipologia, oltre ai tipi tradizionali possono essere scelti i classici tipi Windows (Scrolling List, Check Box, ecc.), i formati di visualizzazione, gli eventuali messaggi di Help, ecc.

## Il Form Tool

È lo strumento che serve per disegnare a video la Maschera, ovvero il Form. Si lavora in un ambiente editor misto, che dispone sia di funzionalità testuali che di funzionalità grafiche.

Gli Object che si possono inserire in un Form sono di sei tipi e per ciascuno è possibile determinare una serie di Proprieties:

TEXT, testi liberi, per i quali si può determinare una serie di caratteristiche estetiche, quali font, allineamento, bordatura.

RECTANGLES, di tipo Filled e di tipo Rounded, per i quali si può determinare retino di riempimento e tipo di linea di bordo.

LINES, comunque posizionabili e con quattro tipi di spessori.

GRAPHICS. Un elemento grafico deve essere realizzato con un prodotto esterno, ad esempio PowerPoint e Paintbrush e trasferito attraverso il Clip-

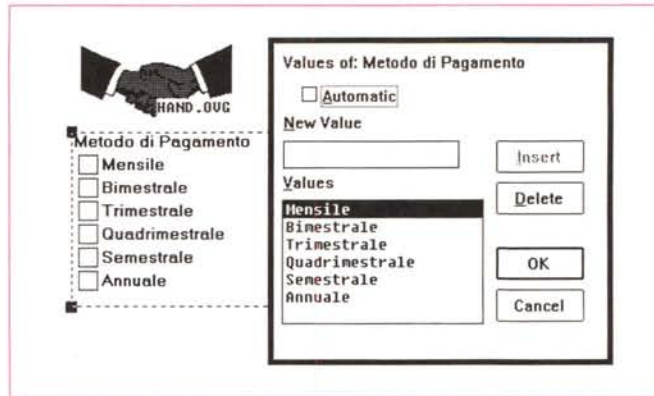


Figura 3 - Borland Object Vision — Campo di tipo Check Box. Vediamo un «collage» che mostra il funzionamento di un campo di tipo Check Box. A destra la Dialog Box attraverso la quale lo si confeziona, a sinistra l'effetto sul Form. La Check Box è una tipica struttura Windows.

board nella Form. Object Vision chiede quale nome si voglia attribuire al grafico in quanto questo rimane memorizzato comunque in un file esterno di tipo bit-mapped con desinenza OVG.

Il dimensionamento della finestra con il Form, il posizionamento e il dimensionamento al suo interno dei vari Objects avvengono ovviamente sempre con operazioni mouse. Nella finestra agisce anche un griglia (strumento tipico dei prodotti di Drawing) dimensionata auto-

maticamente sulla base del carattere di Default. Gli Objects, di qualsiasi tipo essi siano, possono subire oltre alle citate operazioni di movimento e di scaling, anche operazioni di Cut, Copy e Paste, per velocizzare le operazioni di composizione. Conviene in pratica definire un campo, copiarlo più volte e lavorare in modifica sulle copie.

Ovviamente l'elemento più importante è il FIELD, di cui si può determinare: Le tipologie. Sono undici e le descriveremo tra un po'.

I vari attributi estetici del contenuto del campo e della casellina che lo contiene.

Altri attributi come l'impossibilità di sovrascrittura in caso di campi calcolati, oppure il diniego all'accesso, da parte dell'utente, al sottostante Decision Tree.

Interessante è la possibilità di scrivere un messaggio di Help collegato al campo in questione.

Le tipologie dei campi, evidenziate anche nella figura 2, sono:

**General**, vale sia per numerici che per testuali. È il tipo di default.

**Fixed**, valori numerici a lunghezza fissa e con un numero di decimali da indicare.

**Percent**, (0.125 equivale 12.50 %) anche in questo caso va indicato il numero di decimali.

**Financial**. Contiene i separatori delle migliaia.

**Currency**. Visualizza, prima del numero che appare in formato Financial, il simbolo Currency, che è quello scelto nella configurazione di Windows.

**Date e Time**. La gestione delle date e degli orari è presa (pari pari come si dice a Roma) dal Quattro. Ad esempio la funzione @NOW riguarda la data corrente in giorni a partire dal 1/1/1900 e l'orario in frazioni di giornata. In ogni caso va anche indicato il formato di visualizzazione, scelto tra dieci.

**Scrolling**. Serve per testi lunghi per i quali Object Vision dispone una finestra con tanto di barra di scorrimento.

**Selection List**. In fase di immissione, quando il cursore è posizionato sul campo, appare la lista delle opzioni.

Figura 4 - Borland Object Vision — Operatività lato Utente. Object-Vision è un prodotto «double face». L'utente che opera su un'applicazione accede solo ad alcune funzionalità. L'autore invece ha piena operatività su tutte le opzioni di menu ed in particolare accede ai quattro Tools Form, Stack, Tree e Link.



Figura 5 - Borland Object Vision — Maschera con elementi Grafici.

Nel Form si possono includere facilmente elementi grafici prelevati dal mondo Windows tramite il Clipboard. Giunti in OV gli oggetti grafici vengono salvati con desinenza OVG (mentre l'applicazione vera e propria ha desinenza OVD) ma rimangono file esterni. Nell'esempio si notano anche sfondi colorati, non possibili con OV «nativo», e realizzabili come elementi grafici.





del contenuto del campo in maiuscolo. Comunque tale effetto è realizzabile più semplicemente con il campo di tipo Picture.

### Lo Stack

La traduzione «classica» di Stack è Pila. Una applicazione può richiedere più Forms, che l'operatore dovrà compilare seguendo una certa sequenza logica (fig. 8).

OV gestisce un unico elenco di campi, questi possono apparire in una o più Form. Digitati nel primo Form appaiono già riempiti nei successivi.

Volendo sintetizzare quanto detto fino ad ora diremo che OV gestisce all'interno dell'applicazione una serie di Campi. Questi appaiono, verso l'utente, in una o più Form.

C'è ora da chiarire come questi campi, ancora interni a OV, si colleghino agli archivi esterni nei quali, in definitiva, i dati in essi contenuti, vengono memorizzati.

### La creazione del Link

Una volta definita l'estetica del Form e una volta definiti tutti i suoi campi con tutte le rispettive caratteristiche estetiche e contenutistiche, si passa al Tool Link mediante il quale si attiva un collegamento tra i dati gestiti con il Form e l'archivio o gli archivi esterni in cui i dati vengono alla fine immagazzinati.

Esistono alcune varianti del problema dipendenti da due fattori, scollegati tra di loro. Il primo è che l'archivio esterno ancora non esista, il secondo è che ci si voglia collegare a più archivi. In totale quindi le combinazioni possibili sono quattro.

Citeremo i due casi estremi, precisando che nei casi reali si lavorerà sempre su archivi preesistenti definiti e consolidati, verso i quali il Form di OV svolge la funzione di interfaccia lato utente.

Nel primo caso (un solo archivio, generato da OV) l'operazione è automatizzata al massimo, in quanto basta digitare un nome di file che non esiste per indurre OV a crearlo. Object Vision, elaborando i nomi di campo presenti nei Form, crea addirittura i nuovi nomi dei campi nel file esterno e li collega direttamente (fig. 9).

Supponendo che ci si voglia collegare a più file e supponendo, per semplicità, che questi siano già stati creati, siano già pieni e dispongano di un indice (nel caso si tratti di file Paradox o dBASE) i passi operativi diventano i seguenti (fig. 10).

Si apre il Tool Link.

Si indica il nome che si vuol far as-

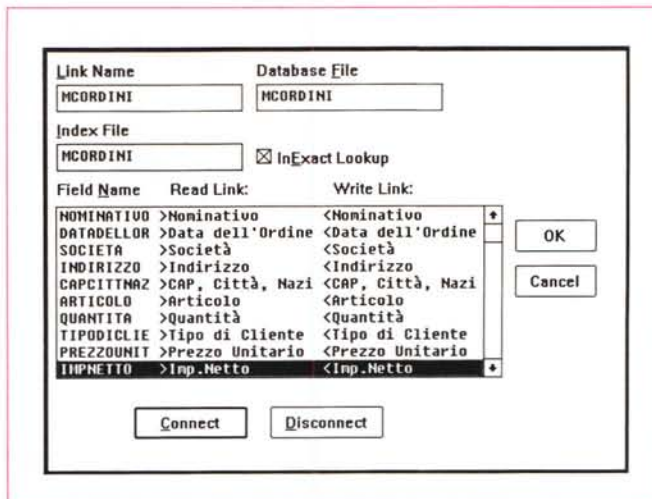


Figura 10 - Borland Object Vision - Link verso tre Archivi. Abbiamo verificato la possibilità di creare, da una stessa Form, tre Link su tre archivi dBASE generati in precedenza ed indicizzati. In tal caso occorre indicare per ciascun campo l'archivio di destinazione. Se il nome del campo è lo stesso (in OV e in dBASE) l'aggancio avviene in modo automatico. Questa possibilità più evoluta viene descritta a fondo nel testo dell'articolo.

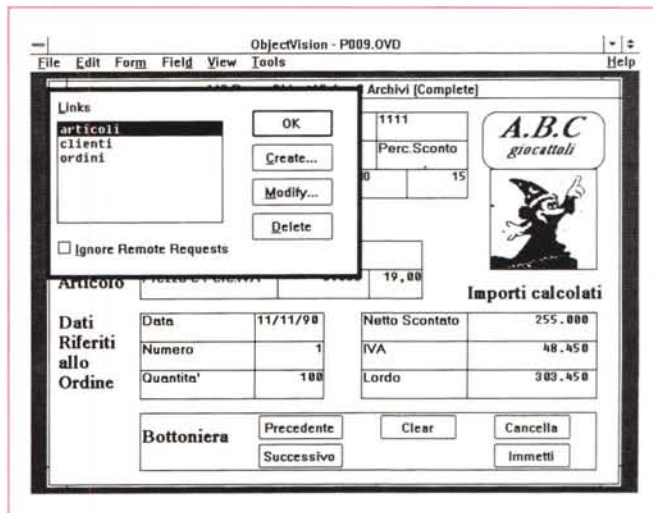


Fig. 9 - Borland Object Vision - Creazione del Link. Una volta generata la maschera, o l'insieme delle maschere, occorre agganciarla al file che contiene i dati, cosa che si fa con il Tool Link. Questo è in grado sia di realizzare il collegamento con file, di diverse tipologie, preesistenti, sia di creare il file con i dati nel formato desiderato. In questo ultimo caso, che vediamo nella figura, OV genera dei nuovi nomi dei campi per adattarli alle specifiche volute dal formato prescelto.

umere al Link stesso e che va utilizzato nelle funzioni di dialogo con gli archivi esterni.

Si indica il nome del file esterno, il nome del suo eventuale indice. Nel caso di collegamento con file dBASE, si può impostare il criterio di esattezza nella ricerca (il comando Set Exact On/Off del dBASE).

Appare una finestra di dialogo a tre colonne. Nella prima sono riportati i nomi dei campi lato OV. Nella seconda i nomi dei campi in lettura dall'archivio esterno e nella terza i nomi dei campi in scrittura.

Nel caso in cui i nomi siano uguali il collegamento viene creato automaticamente, altrimenti è l'operatore che clicca sui vari nomi di campo.

L'operazione si esegue tante volte quante sono gli archivi esterni da collegare.

Alla fine di ciascuna creazione di Link OV chiede se generare automaticamente nella maschera i bottoni che dialogano con l'archivio. Va risposto Sì a seconda dei casi. I bottoni disponibili, che variano un poco a seconda del tipo di

file esterno, e che possono comunque essere inseriti manualmente come oggetti del Form, sono:

- @OPEN «linkname» apre il collegamento
- @CLOSE «linkname» lo chiude
- @TOP «linkname» va al primo record
- @BOTTOM «linkname» all'ultimo
- @PREVIOUS «linkname» ritorna sul record precedente
- @NEXT «linkname» sul successivo
- @CLEAR «linkname» ripulisce i campi di OV
- @UPDATE «linkname» riversa i campi di OV nel record corrente
- @STORE «linkname» riversa i campi di OV in un nuovo Record

È evidente che in ogni Form andrà inserita una «Bottoniera» attraverso la quale gestire il dialogo con l'archivio. In questo modo tutte le normali operazioni individuali (su un solo record) possono essere delegate all'Object Vision.

Per scendere nel concreto abbiamo creato una Form di gestione Ordini (si vede in qualche figura), collegata con tre archivi, due in lettura (ARTICOLI e CLIENTI) e uno in lettura/scrittura (ORDINI). Esistono tra di essi due campi di

Figura 11 - Borland Object Vision — Help in stile Windows.

In qualità di perfetto esponente del mondo Windows 3.0, Object Vision utilizza il sistema di Help messo a disposizione dall'ambiente. Inoltre, all'interno delle Forms delle Applicazioni realizzate con Object Vision, è possibile inserire finestre di dialogo con l'help relativo al singolo campo.

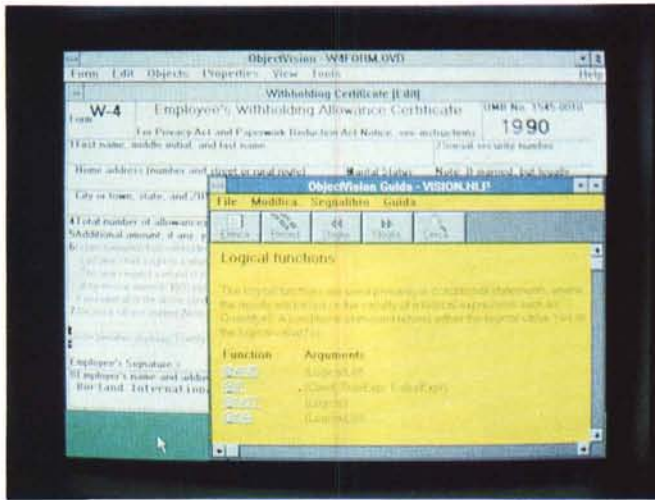


Figura 12 - Borland Object Vision — Stampa di un Modulo.

Nei vari menu disponibili per ognuno dei quattro Tools è presente sia l'Opzione View Screen/Printer sia l'opzione File Print. La prima permette di vedere un'anteprima a video del Form, prima della stampa, la seconda di eseguire la stampa vera e propria del Form.

APEX Life Insurance Company			
Assicurato MARIO ROSSI		Data 04/01/80	
Indirizzo VIA DEL GATTAMELATA 123		Città FIRENZE	Provincia FI
Sesso <input checked="" type="checkbox"/> M <input type="checkbox"/> F		Data di Nascita 11/11/54	Luogo di Nascita FIRENZE
Nome del Beneficiario MARIA VERDI		Relazione con l'Assicurato MOGLIE	
Indirizzo del Beneficiario VIA DEL GATTAMELATA 123			
Premio Annuale Rate L. 2.000.000,00 L. 300.000,00			
L'assicurato ha i prerequisiti <input checked="" type="checkbox"/> No <input type="checkbox"/> Si		Possibile Ass.ne Temporanea <input checked="" type="checkbox"/> No <input type="checkbox"/> Si	
L'ass.to è soggetto al calcolo standard <input checked="" type="checkbox"/> No <input type="checkbox"/> Si		Inserite Clausole di non Copertura <input checked="" type="checkbox"/> No <input type="checkbox"/> Si	
E' richiesto l'esame medico <input checked="" type="checkbox"/> No <input type="checkbox"/> Si		Deposito richiesto L. 500.000,00	
		Deposito incassato Non	

collegamento, presenti in ambedue gli archivi collegati e che permettono di associare l'Articolo e il Cliente al singolo Ordine.

In pratica digitando il codice dell'Articolo vengono immediatamente visualizzati i campi del relativo archivio, oppure digitando il numero d'Ordine, oltre ai dati dell'Ordine stesso vengono visualizzati anche i dati relativi all'Articolo e al Cliente interessato allo stesso ordine.

Non abbiamo sperimentato, trattandosi di tematiche più spinte che esulano un po' dall'ambito di una prova, Maschere più complesse, come quella relativa all'alimentazione contemporanea e controllata di più archivi relazionati tra di loro.

È probabile che lavorando con il Decision Tree e con le funzioni chiocciolina per il dialogo con gli archivi ci si possa riuscire abbastanza facilmente. Vedremo.

I tipi di file collegabili sono cinque:

**ASCII FILE.** In questo caso occorre indicare l'ordine dei campi e il tipo di collegamento (Read, Write o Append).

In pratica le tre opzioni del comando Open del Basic.

**PARADOX.** Con Paradox si «gioca in casa». Può essere indicato anche un campo con un indice secondario. Poi va riempita una tabellina (ma come detto il processo è automatizzabile) in cui al nome del campo per OV va fatto corrispondere il campo in lettura e il campo in scrittura (generalmente coincidono) della struttura Paradox.

**dBASE.** Simile al Paradox, solo che in questo caso può essere indicato anche l'archivio Indice, che, come noto, il dBASE tratta in maniera differente rispetto a Paradox.

**Btrieve.** È un formato di file non legato a prodotti specifici e quindi usato da molti programmatori con linguaggi più tradizionali.

**DDE.** Il Dynamic Data Exchange permette il collegamento caldo con le altre applicazioni sotto Windows.

## Conclusioni

Tutto quello che si può fare con Object Vision lo si fa interagendo con i

menu e con le finestre di dialogo. Non esiste non solo un ambiente «tradizionale» di programmazione, ma neanche un ambiente Macro.

Questa caratteristica produce due conseguenze importantissime.

La prima è che, utilizzandolo, si raggiunge pressoché immediatamente una totale padronanza degli strumenti operativi.

La seconda è che l'utilizzatore è sempre molto guidato in quanto OV gli mostra, volta per volta, solo gli strumenti che al momento possono essere utilizzati. Il rischio di sbagliare viene ridotto al minimo e viene sempre indotta la sperimentazione.

Anche la funzione Help, delegata a Windows 3, è molto puntuale (fig. 11).

Il problema diventa quindi il seguente: quanto può essere complessa una applicazione sviluppata con un prodotto con il quale non si può programmare?

Può esserlo abbastanza se si rientra nei precisi limiti applicativi dichiarati da Object Vision stesso.

L'uso di Object Vision diventa invece consigliabile quando sia importante per l'applicazione la produzione di un modulo cartaceo, pieno di dati, anche calcolati al momento, e che presenti caratteristiche tipografiche (fig. 12).

Deve essere chiaro che comunque OV non è fatto per vivere da «single». Oltre a non gestire un proprio formato di file dati, il che potrebbe non essere un problema perché OV è in grado di generare e di utilizzare un file di altro formato, non dispone di altre funzionalità sempre necessarie in un'applicazione gestionale.

Mancano del tutto funzionalità di manipolazione «insiemiche» sull'archivio, come funzionalità che permettano l'aggiornamento batch di più campi, oppure funzionalità per la generazione ed esecuzione di Report.

Insomma Object Vision non potendo vivere, ancora, di vita autonoma è uno strumento al quale delegare parte delle funzioni di gestione archivi presenti in qualsiasi applicazione di tipo DBMS. In particolare risulta molto indovinato l'accoppiamento con Paradox 3.5, che può fare da motore, sofisticato e potente, alla carrozzeria disegnata con OV.

Usarlo, oltre a essere molto facile è anche divertente in quanto Object Vision rispetta tutte le regole dei prodotti Windows, sia quelle più esteriori, per esempio il Taglia e Cuci degli elementi grafici, che quelle più evolute e promettenti, come il rispetto delle specifiche DDE, per la comunicazione con le altre applicazioni, presenti e future (Paradox ad esempio) sotto Windows.

ME