

# ProGraph 2.5

di Raffaello De Masi

Fin dall'inizio, la diffusa credenza della difficoltà di programmazione del Macintosh portò presto a considerare l'area linguaggi come la cenerentola dell'ambiente. Sebbene fin dall'84 fossero già disponibili idiomi di gran qualità (il Basic della Microsoft, già allora rivoluzionario per l'assenza dei numeri di linea e la presenza di tool di programmazione strutturata potenti e flessibili, uno strano Pascal della stessa Apple, solo interpretato, uno strano Neon, molto vicino al Forth, un eccezionale Basic Apple che non vide mai la luce, e così via), l'ambiente programmatico per così dire amatoriale si ridusse, dal mare magnum di Apple II, a una sparuta pattuglia, quasi una setta carbonara. Era comunque il periodo in cui già comparivano i primi segni dell'abbandono degli ambienti di programmazione da parte degli innumerevoli «softwaristi della domenica» che avevano infestato le riviste con indigesti listatoni da due o più pagine; l'ambiente si faceva evoluto e raffinato, destinato a veri professionisti, capaci di tirar fuori anche il midollo dal Toolbox e da QuickDraw. Lo stesso Basic si trasformava da rozza contadinotta capace di sfornare solo sformati di spaghetti a raffinata signora di classe, capace di generare rampolli come il True Basic, lo ZBasic e successivamente gioielli come il Future e il PG-PRO. Congiuntamente apparivano pezzi di grosso calibro, come l'Atzec C, capace di simulare anche un perfetto ambiente UNIX, i vari idiomi Symantec, alcuni Prolog e, addirittura, un ADA.

## Cosa è ProGraph

Oggi la situazione è abbastanza stabilizzata, con capostipiti ben affermati che si evolvono continuamente con regolarità ed efficienza. In questa ottica ci interessa presentare un pacchetto, che vede la luce circa cinque anni fa, che oggi rappresenta il non plus ultra nel campo della programmazione Object Oriented; ProGraph, appunto, giunto alla versione 2.5



Di questo pacchetto ebbi notizia quattro anni fa, sfogliando un numero di MacUser. Il recensore ne parlava come di una stella di prima grandezza, facile da usare, capace di generare applicazioni compatte ed efficienti. Incuriosito anche dall'insolita interfaccia su cui si basava, ne ordinai una copia a MacZone, che mi rifilò, per diverse centinaia di dollari in fondo di magazzino vecchio di un paio di upgrade. Di tale bidone mi resi conto solo qualche settimana dopo, quando vidi ancora una volta recensito, stavolta su MacWeek, la versione successiva del pacchetto, di due numeri (ancorché decimali) più aggiornata rispetto a quella in mio possesso. Scrisi immediatamente alla ProGraph che mi offrì di upgradare il package alla successiva versione. Di aggiornamento in aggiornamento siamo arrivati alla versione 2.5, di cui leggete in queste pagine.

ProGraph è un linguaggio completamente orientato all'oggetto, e fin qui niente di particolare. Il fatto è che offre

**ProGraph 2.5**  
**Very High level pictorial objected oriented programming environment for Mac**

TGS Systems Ltd - 2745 Dutch Village Road, Suite 200 - Halifax, Nova Scotia, Canada B3L4G7 - Tel. 902 455 4446

Prezzo: dollari can. 895,95

un approccio tanto inusuale alla programmazione da essere praticamente unico. Ma non corriamo!

Il package è costituito dal software e da due manuali; il primo è una combinazione di tutorial e di giro guidato, il secondo, molto più grosso, è una vera e propria guida alla programmazione, molto particolareggiata, e la cosa è senz'altro necessaria se si tiene conto che l'ambiente in cui si lavora è tanto alieno da risultare, senza l'aiuto dei manuali, assolutamente inattaccabile.

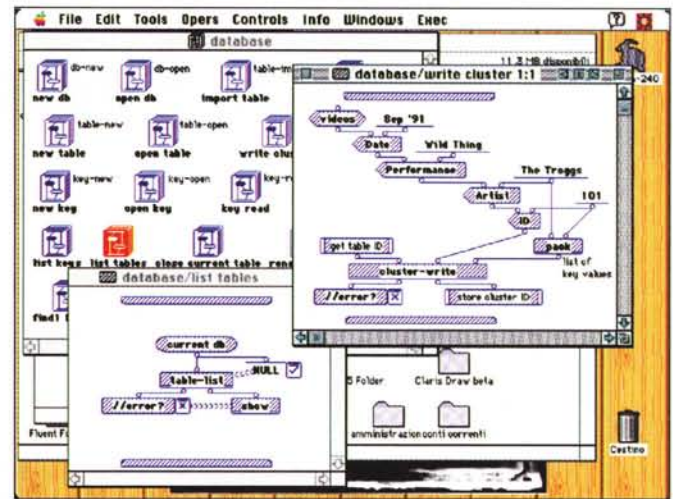
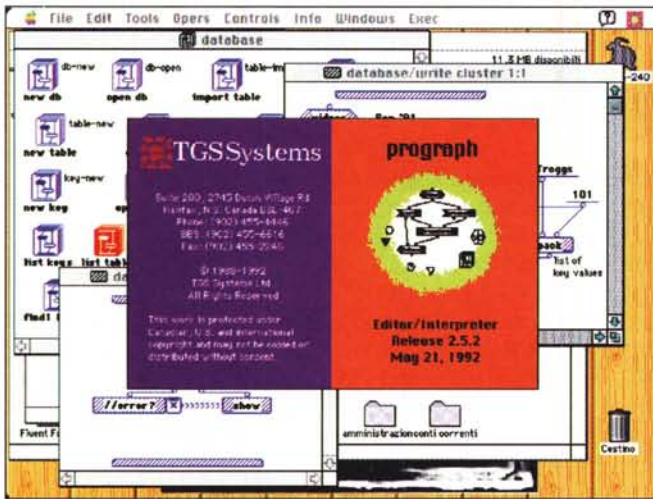
Come dicevamo poco fa, possiamo definire ProGraph interamente visuale e iconico; in

altri termini è completamente pittorico e rappresentabile come una flowchart, in cui i singoli elementi vengono sistemati direttamente dall'operatore, scegliendoli da una serie a disposizione, e da esso possono essere risistemati, ridiretti o completamente convertiti. Tutti gli elementi di ProGraph hanno la forma di un'icona, mentre il testo è solo usato per assegnare nomi alle icone o aggiungere eventuali commenti al foglio. Ricordiamo, per inciso, che ProGraph supporta interamente il modello SI (Single-Inheritance) della programmazione orientata all'oggetto. Un programma redatto in ProGraph consiste dei seguenti elementi:

- classi, con i loro associati metodi e attributi;
- metodi universali;
- operatori (predefiniti, definiti dall'utente, e con controlli associati);
- data object (istanze di classi e/o tipi di primitiva di dati);
- persistenze.

Un programma ProGraph, redatto nel suo editor, viene «scritto» sistemando su un foglio bianco, né più né meno come in un programma di grafica, gli elementi costituenti lo stesso. Ogni elemento può essere collegato ad un altro da «vie», e, cliccandolo, può essere definito nei suoi particolari e nelle sue componenti.

Non è questo il luogo per tenere una dissertazione sulle tecniche d'uso dell'e-



Viste degli ambienti di editor e interprete del linguaggio ProGraph.

ditor di ProGraph. Tanto per intenderci diremo che il suo uso assomiglia molto a quello di Authorware Pro o di Course Builder. Più di questi, che pure sono pregevoli, dispone di strumenti di trace e di debug a dir poco fantastici (basti pensare che si può, ad esempio, eseguire passo passo il programma anche all'indietro), inserire breakpoint anche gerarchici (se si supera quello di grado superiore quello inferiore, nel ramo, viene ignorato). Il linguaggio dispone di un numero veramente notevole di operatori; oltre quelli classici di ogni idioma ci sono tutte le chiamate al Toolbox e alle routine di QuickDraw. Una volta redatto il programma si passa all'ambiente interprete (editor e questo sono, semplicemente, coesistenti) e si adottano tutte le tecniche per il debug (come già detto estremamente sofisticato) e l'ottimizzazione del programma. Dopo di che si passa all'Application Builder, il vero cuore pulsante dell'applicazione. L'ultimo passo è rappresentato dal compilatore capace di generare codici differenziati a seconda del microprocessore cui l'applicazione sarà destinata. Come ogni buon linguaggio che si rispetti, è possibile agganciare codice esterno in un altro linguaggio, ad esempio C e Pascal

(ma abbiamo provato anche «materiale» prodotto con Future Basic senza problemi).

Il compilatore è molto veloce, anche se non ci siamo messi a fare le gare al cronometro per contare il numero di linee compilate al secondo (cosa che pare sia un punto d'onore per i C e i Pascal di MS-DOS). La compilazione avviene, ovviamente, in un solo passaggio, e crea un codice oggetto poco ingombrante, tenendo conto che il compilatore nell'analisi del sorgente «carica» solo le routine di cui ha bisogno.

### Gli accessori

Tanto per non guastare, ProGraph viene fornito dotato di un ProGraph Database Engine. Si tratta di un'applicazione già costruita (e interessantissima da esplorare) rappresentata da un set di primitive pronte per permettere la creazione di applicazioni di database. Utilizzando acconciamente queste primitive, un utente abbastanza scaltrito può realizzare programmi capaci di manipolare basi di dati, tabelle in basi di dati, blocchi (cluster) all'interno di tavole e chiavi per accesso indicizzato ai cluster; con l'aiuto di questo ambiente è possibile

accedere a basi di dati in «query» (sola lettura), e «update» (lettura/scrittura in single user e in sharing). In quest'ultimo modo, cluster particolari possono essere bloccati, prevenendo la modifica simultanea da parte di più utenti (vera croce e tagliola della gestione in multiutenza). Il database engine offre all'utente una buona serie di routine e funzioni già definite (circa cinquanta) utilizzabili in applicazioni personali.

### Conclusioni

ProGraph è davvero un passo nel futuro della programmazione del Macintosh. Completamente orientato all'oggetto, potentissimo, dotato di un disassembler dai muscoli d'acciaio (pronto a cavarci d'impaccio in caso di cancellazioni troppo affrettate), capace di generare applicazioni efficienti e compatte, facile da usare se si entra nella filosofia di scrittura del suo editor, dotato di una libreria di funzioni estesissima (circa cinquecento, tra funzioni e comandi, ma si ricordi che si tratta, nella maggior parte, di strutture già molto complesse), richiede solo un certo ambientamento alla sua tecnica di redazione, e, forse, proprio in questa ottica, avrebbe giovato di un manuale più particolareggiato (talvolta, certe tecniche della programmazione OO vengono date per sottintese). Comunque, di fronte a un listatone in C o Pascal, un programma scritto in ProGraph è un prato fiorito a paragone di una boscaglia tenebrosa. Così, ne consiglieri anche l'uso a chi si avvicina per la prima volta alla programmazione ed ignora le spine che si nascondono dietro anche al più facile (si fa per dire) dei linguaggi. MS

### Ultim'ora

Al momento di andare in stampa ho appreso che TGS sta per lanciare un suo nuovo linguaggio, CPS, che si affiancherà e non sostituirà Prograph. Dotato della stessa interfaccia, e basato sulle stesse tecniche, sarà dedicato a programmatori di solide basi e eccellente livello culturale. Già presentato come ambiente di sviluppo di alcune potenti applicazioni disponibili sul mercato (tra cui anche un eccellente pacchetto di grafica) avrà un costo superiore a quello di Prograph (che probabilmente godrà di uno sconto). Ne parleremo appena possibile!

TGS Systems Ltd - 2745 Dutch Village Road, Suite 200  
Halifax, Nova Scotia, Canada B3L4G7 - Tel. 902 455 4446  
Prezzo dollari can. 895,95