

OSSERVATORIO

a cura di
Lionello Cantoni

IL FUTURO DEL SOFTWARE POGGIA TUTTO SUL PASSATO

Presentare Ottorino Beltrami, capo storico dell'informatica italiana e attuale presidente della Assolombarda, è per me impossibile e me ne scuso con i lettori. Oltre alla grande ammirazione che ho per lui per quanto ha fatto per la nostra informatica, me l'impedisce infatti il consueto affettuoso terrore che provo per questa imponente personalità italiana: Beltrami infatti secondo me riuscirebbe a licenziarmi, anche se mi rendo conto che questo è oggi tecnicamente impossibile, dovunque io sia e qualunque cosa io faccia, se la mia presentazione non riuscirà di suo gusto.

Tutti gli interventi che mi hanno preceduto hanno cercato di immaginare il futuro del software.

Poiché il futuro è sempre, in qualche modo, figlio del passato, penso che sia interessante esaminare che cosa e quanto di quello che si era pensato nell'epoca pionieristica dell'informatica italiana, che ho vissuto in prima persona, è stato successivamente realizzato nel campo del software.

Prima di tutto, dobbiamo ricordarci che «all'inizio era l'hardware»; ossia, nella ingenuità di quei tempi (stiamo parlando della fine degli anni Cinquanta, primi anni Sessanta) si assumeva che realizzata la macchina si fosse fatta la maggior parte del lavoro. Questa illusione durò poco; ben presto ci si accorse infatti che, con un tale approccio, ogni installazione richiedeva uno sviluppo autonomo di programmi con una mobilitazione di risorse assolutamente non trascurabili. E si verificò quella che poi sarebbe stata la regola del settore, almeno per i grandi utilizzatori: la stretta integrazione, cioè, tra fornitore ed utente nella realizzazione di un progetto applicativo.

Mi sembra che oggi questa impostazione, peraltro mai abbandonata, stia vivendo un vero e proprio revival con l'avvento dei sistemi esperti, dove l'utente ha riacquisito una funzione di leadership in quanto depositario della conoscenza del problema.

Vale la pena di ricordare che a quell'epoca i programmi si scrivevano ancora nei linguaggi assembler, con una logica simile a quella del linguaggio macchina (se non addirittura in linguaggio macchina). Un primo orientamento di carattere molto generale fu quindi di uscire

dalla logica della macchina per avvicinarsi a quella dell'utente, introducendo linguaggi *general purpose* di alto livello o procedurali.

Come conseguenza, si costituì in Italia, negli anni Sessanta, una vera scuola di concezione e sviluppo di linguaggi di questo genere con caratteristiche che, superando le soluzioni poi diventate classiche (*Cobol* e *Fortran*), anticipavano i linguaggi più evoluti tipici della seconda metà degli anni Settanta (*Pascal*, ad esempio).

Il *Palgo*, per chi se lo ricorda, era stato sviluppato nei laboratori di Pregnana per la linea Elea 6.000 ed incorporava già allora molte delle principali metodologie di tipo strutturato.

Parallelamente allo sviluppo dei linguaggi *general purpose*, negli stessi anni Sessanta si ebbe anche l'intuizione di affrontare il problema con linguaggi orientati alla specifica area di applicazione. Anche in questa direzione furono mobilitate notevoli risorse, sia in termini quantitativi sia qualitativi e, tra gli sviluppi effettuati, ricordo il *Sigma* per il dimensionamento delle strutture statiche, in acciaio o in cemento armato, ed il *Lambda* per i problemi di produzione e distribuzione; ambedue furono realizzati per l'Elea 6.000 e, limitatamente al secondo, per la successiva linea 100.

Vale la pena di sottolineare che questi linguaggi orientati erano, in effetti, dei veri generatori automatici di applicazioni del tipo di quelli che oggi si affacciano sul mercato, ma si basano su una tecnologia ben altrimenti sviluppata.

In realtà l'hardware degli anni Sessanta non era in grado di sostenere delle soluzioni così avanzate e questa fu la causa del loro scarso successo e, quindi, del loro successivo accantonamento. Oggi la situazione è evidentemente diversa e questo stesso approccio si presenta con

prospettive assai più promettenti.

Quanto detto riguarda il mondo delle applicazioni che gli informatici chiamano «batch». Ma neanche allora avevamo trascurato le cosiddette applicazioni «real time» che già a quei tempi venivano richieste dai clienti più innovativi e, in particolare, dalle banche.

Oggi è ovvio ritenere che qualsiasi elaboratore disponga di *interrupt* per poter ripartire la sua capacità di calcolo su diversi lavori secondo criteri di priorità. Allora, anche su macchine di elevate prestazioni, com'era ad esempio il GE 400, questi meccanismi non esistevano ed il loro funzionamento doveva essere simulato da software.

Diventava perciò necessario per ogni cliente, costruire un sistema operativo in real time, dotato di rilevanti capacità di riconfigurazione dinamica in caso di guasto non bloccante, con notevole anticipo sui tempi e sulle tecnologie.

E non è con questo che avessimo trascurato gli aspetti del software che più propriamente riguardano le capacità del computer nel supporto alle decisioni manageriali. Il responsabile di questa rubrica, Cantoni, sa benissimo quanto sia stato curato a suo tempo il filone delle ricerche e delle applicazioni della allora nascente Ricerca Operativa.

Anche questa è un'anticipazione di ciò che, nella prospettiva degli anni Novanta, si presenta come un modo fondamentale di utilizzo dell'informatica; quella che viene oggi chiamata «informatica strategica» non è infatti niente altro che un rinnovato impegno ad avvalersi del computer come strumento di competitività dell'impresa.

Infine, proprio per tutte le attività di sviluppo software nelle quali ci trovavamo impegnati, ci rendevamo conto già a quei tempi che il software doveva uscire dalla fase artigianale di produzione allora imperante ed

acquisire i connotati di una vera disciplina ingegneristica da gestire con criteri industriali.

Un approccio di questo genere urtava contro due ordini di difficoltà: uno, obiettivo, dovuto alla complessità del problema ed alla scarsa esperienza disponibile sul ciclo di vita del software; l'altro, legato all'ambiente umano allora coinvolto in questo settore, costituito cioè da persone di rilevanti capacità intellettuali, che vedevano le attività di programmazione come un fatto creativo non imbrigliabile in schemi e vincoli industriali. E bisogna aggiungere, a onor del vero, che la situazione del settore giustificava allora ampiamente questa impostazione di principio.

Tuttavia i tempi stavano già cambiando e, se pur faticosamente, si riuscirono a individuare le direzioni lungo cui muoverci per conferire al prodotto software quelle caratteristiche di prevedibilità, pianificazione, affidabilità, verificabilità che rappresentarono poi i prerequisiti dei successivi sviluppi. Giusto per sottolineare il cammino percorso, basti dire che la documentazione del software era in pratica inesistente anche per i progetti più complessi, in quanto ritenuta un inutile spreco di tempo; in effetti, mancavano completamente il concetto di manutenzione e l'idea del suo peso economico nel ciclo di sviluppo del prodotto.

Se vogliamo trarre una conclusione da questo excursus storico su alcuni dei temi della ricerca e dell'applicazione informatica italiana di oltre 20 anni fa, possiamo constatare come, una volta di più, anche le idee più brillanti debbano attendere il loro tempo per potersi veramente affermare e diffondere.

Questa verità si applica probabilmente in tutti i campi della attività umana, ma è particolarmente valida in campo industriale.

Ottorino Beltrami