

Le origini del software industriale in Italia

La Cenerentola software: un tecnodramma italiano.

Il lavoro mette in forma di sceneggiatura diverso materiale documentario sulle origini del software industriale italiano, dalla fase pionieristica presso il Laboratorio di Barbaricina della Olivetti, fino agli inizi dell'esperienza della Olivetti-General Electric ormai in mano agli americani. Si tratta di estratti di messaggi di posta elettronica, di estratti di libri o di altri scritti degli autori e di altri protagonisti e osservatori, che si sono occupati di queste vicende con specifici riferimenti al

software. Il quadro che viene rappresentato rivela lo stato di improvvisazione e di precarietà che caratterizza la concezione e lo sviluppo del software sui primi computer commerciali, le condizioni di lavoro approssimative e primitive e un drammatico contrasto di strategie tra i responsabili del Laboratorio di ricerca, orientati verso strategie di differenziazione, ma poco ancorati alla realtà del mercato, e i responsabili dell'attività commerciale, propensi all'imitazione dei concorrenti e

alla visione a breve termine. Questa divaricazione di strategie fu ulteriormente aggravata dalla sottovalutazione della complessità del processo produttivo di software e dall'insufficiente presidio manageriale delle attività da parte dei vertici. Tutto ciò accade mentre l'azienda è segnata dalla duplice scomparsa di Adriano Olivetti e del responsabile del Laboratorio, Mario Tchou, e dalla forte pressione esterna per la liquidazione delle attività nel settore elettronico.

Gli autori

- *Mario Bolognani (mabol@fastwebnet.it) è componente del Nucleo di valutazione strategica dell'Università "Sapienza" e svolge attività professionale su management e strategie. È docente di Gestione aziendale presso la Facoltà di Ingegneria dell'Università "Sapienza". È stato dirigente di ricerca presso l'Euratom e ricercatore e progettista presso Olivetti, General Electric e in altre imprese private e pubbliche. Ha fondato e diretto imprese impegnate nei servizi di informatica, di consulenza direzionale e trasferimento tecnologico. È iscritto dal 1988 all'Albo nazionale dei giornalisti.*
- *Luigi Logrippio (luigi@uqo.ca) è professore all'Università di Québec nell'Outaouais, a Gatineau. Dopo essersi laureato in Giurisprudenza all'Università di Roma, ha lavorato in Olivetti ed è stato in Siemens Data. Successivamente è emigrato in Canada, dove ha iniziato come ricercatore all'Università del Manitoba. Ha poi insegnato come professore all'University of Ottawa, dove è stato direttore di dipartimento per sei anni.*

Prologo

L'anno zero dell'informatica italiana risale al 1954-55 con la Costituzione del Cscce (Centro Studi Calcolatrici Elettroniche) e l'inizio del progetto Cep (Calcolatrice Elettronica Pisana) presso l'Università di Pisa. Nello stesso periodo l'Olivetti costituisce a Pisa-Barbaricina un'unità di ricerca e sviluppo denominata Laboratorio Ricerche Elettroniche (Lre). Il Laboratorio si occupò subito della progettazione e dello sviluppo delle macchine Elea. La Cep fu pronta in versione ridotta nel 1958 e la macchina fu completata nel 1960.

Gli Elea 9001, 9002 e 9003, il primo a valvole e il terzo transistorizzato, divennero disponibili in rapida sequenza nel 1957. Dice Filippazzi, uno dei progettisti dell'Elea: "Possiamo affermare che l'Elea 9003 era un sistema assolutamente all'avanguardia sotto tutti gli aspetti: la concezione logico-sistemistica, la tecnologia costruttiva, il design". [5]

Però vedremo che il software Elea non era all'avanguardia; e infatti la storia industriale del software italiano inizia solo nel 1958 nel Laboratorio di Barbaricina, con l'arrivo di Mauro Pacelli dall'Università di Pisa nel gruppo Lre.

È utile ricordare l'organizzazione della ricerca e sviluppo software nell'Olivetti nei cinque o sei anni cruciali che ci interessano: 1958-64, cioè fino allo scorporo della Divisione Elettronica e alla costituzione della Olivetti-General Electric (Oge). Questa organizzazione non rimase stabile in quegli anni, ma ai fini di questo lavoro prenderemo in considerazione solo le sue grandi linee. Il primo direttore del Lre fu Mario Tchou. Alla sua scomparsa nel 1961, il successore fu Giorgio Sacerdoti. All'arrivo di Pacelli, in Lre fu costituita un'unità da lui diretta, chiamata "Programmazione logico-matematica" (Plm). Dopo la breve stagione di Barbaricina, le due unità furono trasferite nel 1959 prima a Borgolombaro, poco a sud di Milano, e poi, nel 1961, a Pregnana Milanese, a est di Milano. A Milano, in via Pirelli, operava l'unità organizzativa che si occupava del mercato delle macchine elettroniche, denominata Direzione Commerciale Elettronica (Dce), guidata da Elserino Piol. Nella Dce fu costituito un altro gruppo di sviluppo software, diretto da Lionello Cantoni, che aveva il compito di sviluppare applicazioni in supporto alla vendita. La specialità principale di questo gruppo fu la ricerca operativa ma, come vedremo, esso finì per occuparsi anche di software di base. Quest'ultima attività fu diretta per qualche tempo da Marisa Bellisario, altro attore principale di quel periodo. Il Lre e la Dce erano coordinati solo all'apice aziendale dall'Amministratore delegato della Olivetti, che per una buona parte del periodo che ci interessa (1962-64) fu Roberto Olivetti. Solo nel 1963 fu creata la Divisione elettronica con l'obiettivo di integrare le diverse attività, ma era tardi. Vedremo che in questo difetto di integrazione organizzativa si celava una debolezza strutturale che fu fatale. L'Elea 9003 fu annunciato commercialmente nel tardo 1959 e i primi esemplari furono installati qualche mese dopo. L'Elea 6001, un più piccolo calcolatore scientifico, fu annunciato nel 1961, e l'Elea 4001, poi convertito nel GE115, nel 1965. Altri eventi che contribuirono agli sviluppi che ci interessano furono: l'acquisto della Underwood da parte della Olivetti nel 1959 e la morte di Adriano Olivetti nel 1960. Il primo evento causò considerabili difficoltà finanziarie per la Olivetti, che dovette ristrutturare e quasi ricostruire la Underwood. Il secondo evento causò una perdita di continuità e di visione nella gestione dell'azienda, aggravati nel campo dell'informatica dalla menzionata perdita di Mario Tchou l'anno dopo. Qui di seguito abbiamo raccolto e sistemato parte del materiale documentario in nostro possesso su questa fase pionieristica dell'informatica italiana. Si tratta di estratti di messaggi di posta elettronica a noi indirizzati e, in qualche caso, di estratti di libri o di altri scritti nostri e di altri protagonisti e osservatori, che si sono occupati di queste vicende con specifici riferimenti al software. Ringraziamo gli autori che ci hanno concesso di usare i loro messaggi personali.

I. Lo sviluppo software alla Olivetti negli anni '50-'60

Franco Filippazzi (già progettista hardware dell'Elea, poi dirigente industriale in Olivetti-General Electric e in altre aziende):

Gli sforzi dei costruttori erano stati concentrati sulla macchina fisica, mentre il software era visto come un fattore ancillare dell'hardware. Ciò era dovuto anche al fatto che, mentre la macchina era tangibile, visibile a tutti, il software era invece una cosa invisibile, difficile da capire e valutare. Gli sforzi in questo campo erano limitati in tutto il mondo; basta esaminare i dati statistici del tempo per vedere come il software fosse allora una piccola frazione dell'investimento totale in ricerca e sviluppo nel settore dei calcolatori. Le macchine erano consegnate all'utente praticamente "nude": i programmi venivano sviluppati poi, in base alle esigenze specifiche, con il concorso di utente e fornitore. [6]

Luigi Logrippo (sviluppatore di software applicativo sull'Elea 9003 presso la Direzione commerciale elettronica Olivetti, dove ebbe anche l'opportunità di collaborare con il gruppo Plm; ora professore all'Université du Québec in Canada):

Ricordo che una notte, passeggiando nei sotterranei della banca dove stava il nostro enorme Elea 9003, trovai un compatto Ibm 1401 con un amichevole collega programmatore accanto (tenuta Ibm di quei tempi: giacca e cravatta scure). Questi mi mostrò un bel tabulato di una compilazione Cobol. E noi della Olivetti programmavamo ancora in linguaggio macchina! Più importante ancora, questa macchinetta, di capacità circa dieci volte inferiore al 9003, funzionava senza problemi nei suoi componenti hardware e software. La stampante e il lettore di schede sembravano volare! Era un altro mondo... Il 1401 fu annunciato verso la fine del 1959, forse poche settimane dopo il 9003. Aveva fin dall'inizio un assembler, Report Program Generator e altro software. Il Cobol e il Fortran seguirono poco dopo, nel 1960. Immagino che già alla fine degli anni '50 l'Ibm avesse molte decine di persone per sviluppare il software della serie 1400, in modo da poterlo consegnare insieme alla macchina fisica.

Renato Obermito (programmatore dell'Elea 9003 nel gruppo software di base presso la Divisione elettronica Olivetti, vive ora nei pressi di Torino):

Il gruppo guidato da Pacelli, alla fine del 1959, non aveva più di dieci persone che si occupavano di routine scientifiche, *sort&merge*, calcolo vettoriale e matriciale e linguaggi assembler e compilatori. È interessante notare che nessuno si occupava di ciò che più tardi venne chiamato 'Sistema operativo'. All'inizio del 1962 entrò nel gruppo la prima squadra di giovani programmatori (non più di 16 persone) tutti diplomati e addestrati da zero con un corso di cinque settimane. Con qualche altro nuovo ingresso alla vigilia della

cessione agli americani (estate del 1964) il gruppo del software non arrivava a 35 persone, addestrate sommariamente, ben pochi dei quali erano laureati e potevano vantare un significativo curriculum di studi. In parallelo a questo gruppo, presso la Divisione commerciale operava un altro gruppo, coordinato da Lionello Cantoni, che si occupava di applicazioni per i clienti; il gruppo era costituito da non più di 10-15 tecnici, inclusa Marisa Bellisario che, con una decina di programmatori in cantina, sviluppava il software di base ma più orientato ai clienti.

Logrippio:

I programmatori Olivetti, per lo più diplomati dalle scuole tecniche, venivano addestrati con un corso di cinque-dieci settimane sul linguaggio macchina e il *flow-chart*, senza mai vedere le macchine che avrebbero dovuto usare. Non c'era alcun principio di progettazione o test di programmi. L'analisi del problema era trasmessa in vari modi, tra cui i prevalenti erano il raccontino parziale e intuitivo (si andava talora dalla bocca del cliente al linguaggio macchina), i *flow-chart* (che spesso i programmatori erano costretti a codificare senza capirci niente), e raramente un solido documento di base. Anche lo stile di programmazione era estremamente primitivo. C'erano solo le sequenze di istruzioni e i salti, condizionali o incondizionali, che erano usati abbondantemente e andavano in tutte le direzioni. Quando un programma doveva essere modificato, i programmatori non volevano modificare le parti del codice che già funzionavano, ma piuttosto inserivano un salto alla fine del programma dove veniva scritto il nuovo codice, dal quale poi con un altro salto si tornava indietro. Più tardi il gruppo della Bellisario rese disponibile un linguaggio assembler (di nome Psico) che non cambiò di molto il quadro, perché il compilatore era lento (il programma sorgente era memorizzato su nastro magnetico e ogni compilazione richiedeva diverse 'passate' del semilavorato fino al codice oggetto), cosicché i programmatori assemblavano il programma solo all'inizio e poi facevano la messa a punto in linguaggio macchina.

Marisa Bellisario (programmatrice Elea 9003, poi alto dirigente in Olivetti General Electric e in altre aziende, scomparsa nel 1988):

A Valdarno (dove, presso la Marzotto, fu installato uno dei primissimi Elea, *n.d.a.*) i guardiani e gli operai sono convinti che io abbia una sorella gemella: mi vedono spesso uscire all'alba e rientrare alle 8:30 del mattino. [2]

Logrippio:

Più tardi lei, quando divenne capo dell'unità di sviluppo, chiese spiegazioni quando scopri che 'la macchina' era stata spenta tra le 00:00 e le 8:00 di una bella giornata di Natale!

Poiché mancavano sistemi di *debugging*, la messa a punto dei programmi poteva essere svolta solo dai programmatori stessi quando il calcolatore era disponibile. Ciò poteva essere fatto quasi solo di notte, perché durante il giorno la macchina era usata per eseguire programmi produttivi. Inizialmente, il metodo principale per il *debugging* era l'esecuzione istruzione per istruzione controllata dalla *console*, osservando il risultato nei registri dell'unità centrale, dove apparivano in codice binario come sequenze di luci. Si potevano pure usare le stampe del contenuto della memoria dalla scrivente della console e dalla stampante parallela. Abbastanza presto il gruppo Bellisario rese disponibile un programma, chiamato "Monitor", per il test passo a passo: simulava l'esecuzione di un programma stampando le istruzioni una a una, insieme al contenuto delle zone di memoria interessate dall'istruzione. Fu un grande aiuto, ma consumava chili di carta e aiutava poco per errori complessi.

In quegli anni l'Elea 9003 era affetto da seri problemi di manutenzione. Ogni mattina il gruppo di manutenzione faceva il suo giro, ma già nel pomeriggio il sistema cominciava a guastarsi, specialmente nelle apparecchiature periferiche. Nei primi tempi i programmi venivano perforati su banda di carta, carattere dopo carattere e quindi erano letti dal lettore di banda (l'Le privilegiò la banda come metodo di input, un altro errore storico). Spesso il lettore si inceppava rompendo la banda. Tuttavia, prima che il nastro si rompesse, una porzione del programma era già stata letta in memoria, così i programmatori effettuavano l'input del programma un pezzo di banda alla volta. Il programma veniva quindi riappiccicato in memoria con comandi di console e copiato sui nastri magnetici, a loro volta fonte di errori. Ricordo soprattutto l'errore di nastro che si manifestava con una lucetta su cui c'era scritto PE (chissà cosa volesse dire, noi lo chiamavamo "Porca Eva!"). Esso sopravveniva regolarmente, talvolta buttando a mare ore di lavoro, perché fermava tutta la macchina. Quando finalmente arrivarono i lettori di schede, quelli Bull erano lenti e inaffidabili; si sentiva troppo spesso gridare: 'bourrage' (cioè, intasamento), e varie schede rimanevano distrutte. Il trattamento degli errori non era neanche quello un punto forte dei primi Elea.

Lo stato dell'arte

Logrippio:

La parola "software" appare per la prima volta nel 1958 nell'edizione a stampa di un saggio dello statistico John W. Tukey in *1958 in the American Mathematical Monthly*. Tukey è anche accreditato dell'invenzione del termine "bit".

Mario Bolognani (programmatore dell'Elea 4001/GE115 a Pregnanza, nella Olivetti-General Electric, oggi consulente di management

e docente di 'Gestione aziendale' all'Università Sapienza di Roma): Lo stesso termine 'informatica' fu coniato più tardi, nel 1962, dal francese Dreyfus, fondatore della Cap Gemini. Insomma, non c'era a disposizione neppure un vocabolario adatto per trasmettere quel tipo di conoscenze.

Logripo:

Alla fine degli anni '50, però, si sapeva già molto sul software. Ho aperto le Communications of the ACM del 1958 (Anno I), e vi ho trovato i seguenti articoli:

- *Automatic programming systems;*
- *The problem of programming communication with changing machines: a proposed solution;*
- *Proposal for an Uncol;*
- *On the equivalence and transformation of program schemes.*

Molto di più si sarebbe potuto scoprire con uno sguardo a quello che succedeva nell'industria americana e inglese (quest'ultima molto creativa a quel tempo).

Bolognani:

Vi sono altre testimonianze importanti del livello di conoscenze sul software raggiunto in Italia negli anni Cinquanta in ambito scientifico. Per limitare l'attenzione alla questione cruciale dei compilatori, si sa che nel 1952 Corrado Bohm, dottorando al Politecnico di Zurigo, definì un linguaggio di programmazione e ne descrisse il compilatore completo con lo stesso linguaggio. Fu una pietra miliare per lo sviluppo dei linguaggi di programmazione. Questo successo fu menzionato con onore in un noto articolo di Knuth e Pardo. [7]

Logripo:

Conservo ancora i Proceedings del congresso internazionale "Symbolic Languages in Data Processing", che si tenne a Roma nel marzo del 1962. Accanto ai grandi nomi internazionali dell'epoca (Dijkstra, Gorn, Naur, Wegner ecc.), ci sono tre articoli del gruppo Plm, di cui gli autori erano: Gavioli, Pacelli, Palermo, Picciafuoco. Trattavano di vari problemi di compilazione di linguaggi della famiglia dell'Algol. [8]

Luigi Petrone (progettista di linguaggi presso la Olivetti, poi professore all'Università di Torino):

Io ho raggiunto l'Olivetti quando il laboratorio da Borgolombardo stava traslocando a Pregnana (perciò tra il 1961 e il 1962) e quindi non ho partecipato all'esperienza di Barbaricina. Venivo da un'esperienza in Fiat alla Centrale Calcoli Elettronici, dove si cercava di concretare il tentativo della Direzione generale di seguire le novità della nascente industria (mi avevano mandato nel 1955-56 per 3-4 mesi all'Eth di Zurigo, dove era attivo Speiser ed era installato lo Zuse) e poi

ci eravamo occupati di costruire un interprete e un compilatore per un miniAlgol per la Univac installata in Fiat. Dalla Fiat ero passato all'Euratom di Ispra, dove il centro diretto da Braffort era molto attivo come insemminatore di idee (avevamo avuto come ospiti MacCarthy, M.P. Schutzenberger e altri) e io ebbi l'occasione di formarmi delle solide basi di teoria dei linguaggi. A Pregnana mi aveva voluto Pacelli per poi affidarmi l'incarico di costruire tutti i compilatori della nuova linea su basi *syntax-directed*. Si trattava di una tecnologia nuovissima e promettente che ci vedeva tra i primi a livello mondiale. Certo, il gruppo era esiguo (avevo come collaboratore Vandoni e poi dovevano raggiungerci Lusso e qualche altro) e qualche mese dopo (dopo la scomparsa di Mario Tchou, *n.d.a.*) si ebbe la lunga pausa che accompagnò le trattative per la vendita alla General Electric, trattative che precedettero lo smembramento di Pregnana. La Olivetti-General Electric si sarebbe occupata solo della linea 100 (precedentemente chiamata Elea 4001) e noi fummo presi prigionieri dalla Direzione commerciale di via Pirelli.

Bolognani:

Ricordo anche che Petrone, autore di un lavoro precursore sui compilatori guidati dalla sintassi, [14] passeggiando nei corridoi dei passi perduti di via Pirelli, si lamentava che le sue idee originali e anticipatrici fossero state sviluppate successivamente da Lewis e Stearns della General Electric, con cui Petrone aveva avuto dei rapporti e che si presero buona parte dei riconoscimenti a lui dovuti. Dunque le conoscenze in Italia, e nel gruppo Plm in particolare, erano di prim'ordine a livello mondiale. Purtroppo, la dissoluzione del gruppo disperse persone e conoscenze. Attorno al 1965, con la Ge in pieno controllo dell'azienda, a Pregnana Milanese la cultura del software di base era tornata al livello degli smanettatori individuali, la biblioteca era deserta e io, con Misasi, ne eravamo gli unici frequentatori (almeno del gruppo software). Certo, si era diffusa una maggiore consapevolezza dell'importanza dei linguaggi di sviluppo, dall'assembler ai linguaggi per lo sviluppo di applicazioni, fino ai primi embrionali sistemi operativi. Però io mi trovai a sviluppare un *Tape operating system*, mentre Ibm aveva già il *Disc operating system*, quindi: si annaspava sempre all'inseguimento e in ritardo. Gli ingegneri elettronici continuavano a dominare: Fubini, il capo del Laboratorio e tutti i dirigenti più influenti (Filippazzi, Raviola, Cesa Bianchi, lo stesso Perotto che lavorava per Olivetti, ma aveva lì il suo laboratorio) venivano dalla progettazione hardware. Ogni lavoro nel software sembrava la riscoperta della ruota, come se la diaspora della Divisione elettronica, seguita alla cessione, avesse lasciato lo zero assoluto. In realtà, oltre alla dispersione dell'esperienza fatta vi era anche un altro fenomeno. La trasmissione delle cono-

scenze era bloccata perché chi sapeva qualcosa se lo teneva per sé come un elemento distintivo per fare carriera e avere aumenti di stipendio. Io ho lavorato con Pacelli a partire dal 1968 sull'idea del "Common Compiler Design" per la nuova linea della General Electric (e anche con Petrone che in quel periodo era a Phoenix nel gruppo di Pacelli), e a quel tempo non solo era una miniera di conoscenze (aveva già alle spalle 4 anni di Usa con un passaggio molto fruttuoso nel gruppo che presso il Mit sviluppava Multics) ma, fatto per me nuovo e sorprendente, ci educava e ci faceva ragionare sulle tecnologie software.

Un contrasto di strategie

Mauro Pacelli (Docente di meccanica razionale all'Università di Pisa, poi capo del software presso il laboratorio Olivetti di Barbaricina, quindi negli Usa in General Electric, Honeywell, Ncr. Vive ora in California, nei pressi di San Diego): Non avevamo a nessun livello un'idea chiara dei processi industriali necessari allo sviluppo, implementazione e lancio di un sistema di elaborazione. Non credo che esistessero neppure i finanziamenti necessari per fare un buon lavoro, ammesso che sapessimo come farlo. Forse fu un errore incominciare in quelle condizioni. Ma non erano

errori. Era un processo di apprendimento che terminò prima che avessimo imparato il mestiere. Questo accadeva a tutti i livelli dello sviluppo, produzione, marketing e vendite. Forse potevamo avere un'opportunità se la nostra attività fosse stata coordinata a livello europeo con altre imprese. Avevamo un potenziale innovativo notevole ed eravamo stati scelti per le capacità innovative; ma dopo la morte di Adriano Olivetti e di Tchou, gli obiettivi cambiarono a livello più modesto. Comunque è stato un periodo breve, ma molto intenso e piacevole.

Elserino Piol (Capo della Direzione commerciale elettronica, poi alto dirigente in Olivetti, oggi attivo come imprenditore nel settore del venture capital):

La confusione nel software non era solo in Olivetti, ma in tutta l'industria del tempo. Ho conosciuto bene la realtà Bull e non era certo un esempio da imitare. E quando nel luglio 1964 (in relazione alla trattativa in corso tra Olivetti e General Electric) il team dirigenziale Olivetti capitanato da Roberto Olivetti (e di cui io facevo parte) visitò la sede della Computer Division della General Electric a Phoenix in Arizona, risultava che nel software erano anche più confusi di

Il software fra mercato e innovazione

commento

di Giorgio De Michelis

La vivace ricostruzione di Mario Bolognani e di Luigi Logrippio della storia del software italiano (in verità è sostanzialmente la storia del software in Olivetti) negli anni a cavallo tra il '50 e il '60 è un regalo prezioso, non solo per il fatto che documenta un periodo assai interessante dell'informatica italiana, ma anche perché lo fa attraverso le voci dei suoi protagonisti, che a distanza di anni riflettono sulle scelte che caratterizzarono lo sviluppo del software in anni assai lontani, in cui bisognava prendere decisioni sulla base di intuizioni non suffragate da alcuna evidenza né tecnica, né commerciale. Il quadro che emerge dalle pagine di Bolognani mi pare sostanzialmente corretto (il dubbio palesato da quel 'mi pare' è do-

vuto al fatto che io entravo nell'informatica italiana quando quella fase era già finita, per cui non posso aggiungere la mia testimonianza a quelle riportate da Bolognani) tutto pervaso dal conflitto tra chi voleva seguire le priorità del mercato e chi voleva seguire quelle della ricerca e dell'innovazione e dalla sottovalutazione del software rispetto all'hardware. È interessante e bello che a distanza di quasi cinquant'anni Elserino Piol ammetta oggi che forse si era sbagliato nel non capire l'importanza dell'approccio pionieristico di Pacelli, ed è significativo che le attività degli innovatori, guidati da Pacelli, fossero pienamente inserite in un contesto di ricerca internazionale dove erano guardate con interesse e attenzione.

Ma il quadro presentato da Bolognani risulta sostanzialmente incomprensibile, se non teniamo conto del contesto in cui si svolsero gli eventi da lui narrati. Certo, c'era in quegli anni una gran confusione in Olivetti sul terreno delle strategie e dei piani di sviluppo del software di base. Ma era una situazione comune a tutta l'informatica mondiale: negli altri paesi d'Europa e negli Stati Uniti, il dibattito sulla possibilità di sviluppare l'informatica per passi incrementali, come stava avvenendo con RPG, Fortran e Cobol, oppure seguire la strada dell'innovazione radicale e puntare ai linguaggi Algol-like era lo stesso in ogni parte del mondo e quello che più conta è che le valutazioni delle due parti erano sostanzialmente

noi. Tutta l'industria informatica richiese tempi lunghi per definire una strategia per il software e la corretta bilancia tra hardware e software.

Bolognani:

Non era così all'Ibm, come testimonia efficacemente Logrippe e come si sa dalla letteratura tecnica. Avevano le idee chiare e le perseguivano con determinazione e con una superiore organizzazione industriale. Certo, avevano dalla loro anche la forza economica e la dimensione del mercato, ma senza strategia e organizzazione non avrebbero vinto la partita, in quella fase storica.

Logrippe:

In effetti, rivediamo un momento quali erano le migliori pratiche di sviluppo software di quell'epoca, seguite dalle ditte più esperte negli Stati Uniti e in Inghilterra, forse anche in Germania. Contemporaneamente all'hardware, si sviluppava un assemblatore, che doveva essere veramente efficace. Su questo si costruivano *linker and loader*. Venivano poi i sottoprogrammi di entrata/uscita, i sottoprogrammi scientifici, utilitarie di fusione e ordinamento nastri, eccetera. Su questa

impalcatura si costruivano i compilatori per i linguaggi ad alto livello, come Fortran, Algol e Cobol. È facile oggi capire che ogni scorciatoia da questa via maestra si perde nei boschi! È facile anche capire che, per seguire questa via, la Olivetti avrebbe dovuto cominciare lo sviluppo del software un paio di anni prima, sotto la guida di persone già esperte. Forse la società fu erroneamente incoraggiata dall'architettura di macchina particolarmente conviviale degli Elea (indirizzi e rappresentazione dati decimali anziché binari: negli Elea era possibile lavorare in linguaggio macchina, mentre in molte altre architetture questo era quasi impossibile) e anche dalla filosofia, discussa più sotto, per cui si sarebbero dovuti sviluppare il più rapidamente possibile i linguaggi ad alto livello. Come vedremo, si cercò di fare tutto in parallelo e con vario successo. Fu così che i programmatori scrissero (o cercarono di scrivere) compilatori e programmi di utilità sulla macchina "nuda". L'Olivetti fu molto fiera di avere messo in produzione quello che credeva essere il primo computer commerciale transistorizzato, l'Elea 9003. A parte il fatto che ce ne furono un paio d'altri prima, sfortunatamente si è visto e si vedrà che tanto il 9003 quanto il 6001 non erano dei prodotti completi; erano anzi difficilmente utilizzabili,

auto-referenziali: chi era per il mercato sosteneva Fortran e Cobol, perché essi erano richiesti dal mercato; chi era per i nuovi linguaggi ad alto livello li sosteneva perché erano innovativi. Mancavano strumenti per verificare se le prospettive di mercato erano credibili e/o stabili e mancavano strumenti per valutare quando i prodotti innovativi sarebbero stati disponibili e avrebbero portato a nuove applicazioni e quindi a nuovi clienti. Salta agli occhi, con il senno di poi, la stranezza di un dibattito tutto centrato sui linguaggi di programmazione piuttosto che sui sistemi operativi, in cui quindi non si era ancora capito che cosa avrebbe fatto la differenza negli anni seguenti. Oggi in un sistema si rendono disponibili tutti i linguaggi richiesti dai clienti, ma l'innovazione non sta lì, quanto piuttosto nel sistema operativo in cui questi linguaggi vengono implementati.

Ma soprattutto il fatto era che il gruppo Olivetti era drammaticamente solo in Italia (non c'erano ancora gruppi d'informatica significativi nelle Università, mancavano i finanziamenti pubblici,

non c'era un'industria del software) e isolato in Europa e nel mondo (non che mancassero altre iniziative imprenditoriali in Europa, ma ciascuna di esse faceva come se fosse l'unica e guardava agli USA per capire dove si doveva andare e non si fece nessuno sforzo per far discutere le piccole imprese europee tra di loro, e per far sì che condividessero parte degli investimenti ecc.). L'America era ancora lontana: ci si andava raramente e senza poter stabilire legami saldi (la situazione sarebbe mutata negli anni immediatamente successivi, ma perché le imprese d'oltre Oceano seppero richiamare alcune delle figure più significative del software italiano, che da allora divennero dei punti di riferimento per chi lavorava al software italiano, ma senza progetti comuni). Tenendo conto di questi due fattori di contesto che delineano una tecnologia nel suo farsi (non era l'Olivetti che era una start up: era l'intero mercato dell'informatica ad esserlo, a livello mondiale) e una posizione marginale dell'Italia a cui non si è saputo far fronte con azioni

che ne limitassero la fragilità, il quadro è molto più lusinghiero di quanto non appaia nel testo di Bolognani. Nonostante tutto, c'eravamo anche noi e con pieno merito: si può forse dire lo stesso del software italiano oggi?

È vero comunque che in quegli anni non c'era alcun elemento per scegliere tra mercato e innovazione: bisognava esserci nell'uno e nell'altra, sviluppando tutto ciò che poteva facilitare il passaggio dell'innovazione al mercato ed è questo intreccio di mercato e innovazione che le ridotte dimensioni dell'iniziativa italiana nel software non ha saputo sviluppare.

Ecco, come dicono alcuni dei protagonisti dell'epoca, sono stati anni in cui si andava a tentoni ma si facevano cose nuove e di qualità. Non dimentichiamolo, ma non cadiamo neanche nella nostalgia di un passato che ci onora, ma che è definitivamente passato: oggi i problemi sono altri e andrebbero affrontati con la stessa determinazione con cui si mossero i nostri precursori in quegli anni lontani.

a causa della mancanza di software di sistema e applicativo. Alcuni anni dopo, sia pure in ritardo e con difficoltà, come racconta Bolognani, migliori pratiche furono seguite nella Oge per il Ge 115.

Logrippio:

Esisteva una minima strategia o filosofia aziendale per lo sviluppo del software degli Elea?

Pacelli:

Non credo che ci fosse una filosofia Tchou o Sacerdoti per il software, ma mi ricordo chiaramente che Tchou spingeva la mia iniziativa su linguaggi ad alto livello derivati dall'Algol e sull'architettura di computer per facilitare la compilazione e l'esecuzione degli stessi. Queste idee condussero al Palgo e alla concezione dell'Elea 9004 (il Palgo era un dialetto dell'Algol inventato nel gruppo Plm. L'Elea 9004 era una macchina con la logica "a stack", quindi ideale per l'esecuzione dell'Algol, simile alle macchine della serie Burroughs 5000. Ne fu prodotto solo un prototipo, *n.d.a.*).

Il Palgo (un linguaggio sviluppato anzitutto per l'Elea 6001) doveva essere un linguaggio di programmazione per la 9004 e anche un laboratorio per sperimentare nuove idee sui linguaggi di programmazione. In particolare ero affascinato dalle caratteristiche del Simula e prima che me ne andassi in Usa stavo pensando di introdurre nel Palgo strutture derivate dal Simula, che in seguito furono introdotte nel C language e sono ancora nel C++.

Quando il 9004 fu "ucciso" in favore di un "interior decor" alla Ibm, l'interesse per il Palgo svanì.

Il Pac (Programmazione automatica commerciale) era un linguaggio di programmazione per applicazioni commerciali sulla falsariga di un analogo prodotto Ibm (di cui non ricordo il nome), prima che nascesse il Cobol. Io, Palermo e Vandoni scrivemmo le specifiche del compilatore per l'Elea 9003, che fu implementato da un gruppo di programmatori con diploma di istituto tecnico, di cui faceva parte Obermito. Il compilatore fu anche usato per alcune applicazioni, ma non so quante volte. Quando cominciai a interessarmi di tecniche di ordinamento su nastro, chiesi a Obermito di implementare un generatore di programmi di ordinamento in cui si usavano proprietà dei numeri di Fibonacci (proprietà che permettono di ridurre il numero delle passate su nastro nell'ordinamento polifase, *n.d.a.*).

Obermito:

Io, appena terminato il corso, sviluppai da solo l'Aps (Assembler di programmi simbolici) per Elea 6001. Questo prodotto, molto malfunzionante purtroppo (ho ancora adesso il rimorso per non averlo fatto meglio, ma era il mio

primo lavoro dopo le famose 5 settimane di corso), fu distribuito ai pochi clienti 6001 (specie università, ricordo Padova e Bologna; ma anche qualche ente pubblico, come l'Istituto Superiore delle Poste e delle Telecomunicazioni). Voglio raccontare un aneddoto per dare un'idea del clima in cui si lavorava. Al termine del corso, Pacelli mi chiamò e mi disse che avrei dovuto sviluppare il Palgo per l'Elea 6001. Io gli replicai che forse avremmo dovuto sviluppare prima l'Aps. Al che Pacelli, che dell'Aps si era dimenticato, rispose: 'è vero, allora faccia prima l'Aps e dopo il Palgo (abbiamo visto che questo era l'ordine giusto, *n.d.a.*). Entrambi pensavamo che sviluppare un assembler da solo sarebbe stata questione di giorni, o al massimo di settimane. Ci lavorai un anno con i risultati discutibili di cui ho detto. Per mesi girai l'Italia a rattoppare il prodotto che era un vero esempio di programma spaghetti.

Piol:

Il software costituì una grande area di discordia tra me, che rappresentavo il mondo commerciale, e Giorgio Sacerdoti, il capo del Laboratorio. Va detto che nessuno di noi aveva le idee chiare. Mentre sul software applicativo ritenevo, a torto, di non avere problemi (era comunque di mia competenza), sul software cosiddetto di base la confusione era grande e il dissidio tra noi drammatico. Sacerdoti aveva affidato a Pacelli lo sviluppo di tutto il software di base. Pacelli impostò la sua attività con un approccio da ricercatore, senza considerare gli standard emergenti e comunque con una tempistica inaccettabile. I processi di pianificazione e controllo per tutta l'attività elettronica erano inesistenti. In particolare ritenevo che i linguaggi di programmazione che Pacelli voleva realizzare non fossero coerenti con le limitate possibilità dell'Elea 9003; così, invece di aspettare Pacelli, feci mettere a punto autonomamente software di base che, con funzionalità minime, mi permettessero di operare. Vi era insomma uno scontro di strategia che non fu mai risolto. D'altra parte io stavo sul mercato e dovevo rispondere ai clienti. Il problema principale dell'Elea era la completezza e l'affidabilità del sistema offerto, nel quale il software era praticamente assente. In condizioni normali avremmo dovuto concludere che l'Elea 9003 non era vendibile. Luigi Dadda (professore al Politecnico di Milano, poi rettore dello stesso ateneo, *n.d.a.*) mi manifestò la forte convinzione che nessuno avrebbe mai comprato l'Elea 6001 con il Palgo ma senza un compilatore Fortran, linguaggio ormai diventato lingua madre per gli utenti. Creai allora un gruppo di lavoro gestito dalla Bellisario, che lavorando giorno e notte, presentò il compilatore alla scadenza prevista. Salvai il cliente, ma fui quasi licenziato da Roberto Olivetti per questa invasione di campo [16]

Bellisario:

L'Elea 6001 mi mette... in una situazione competitiva per portare a compimento un altro *crash program* voluto da Piol (era il 1962, *n.d.a.*). Si tratta di realizzare, in un mese o poco più, un compilatore per il linguaggio scientifico Fortran, che il cliente, l'Istituto Superiore Tecnico delle Poste e Telecomunicazioni, pone come condizione per rispondere al bando di concorso. Al laboratorio di Borgolombardo (nel Plm, *n.d.a.*) non si vede ancora nulla di funzionante, anche se i ricercatori stanno lavorando da tempo a un altro linguaggio scientifico, poi rifiutato dal mercato, l'Algol. [2]

Logrippo:

La validità a lunga scadenza delle decisioni di Tchou e Pacelli fu confermata dalla storia. Essi si resero conto che il costo del software era tale che quest'ultimo doveva sopravvivere all'evoluzione dell'hardware. Puntarono quindi sui linguaggi ad alto livello, e soprattutto sui linguaggi della famiglia dell'Algol, per raggiungere questa compatibilità. Ma furono i Bell Labs che riuscirono una decina di anni dopo a creare una specie di esperanto informatico con il C, che peraltro dovette attendere molti anni prima di assumere questo ruolo. Molti dei linguaggi di programmazione attuali sono stati influenzati dall'Algol, attraverso Pascal, Ada, C e fino a Java, mentre il Fortran è rimasto senza discendenti. La volontà di Pacelli di includere nel Palgo caratteristiche del Simula era anch'essa profetica, anticipava evidentemente l'Object Oriented. Chi ha detto che l'informatica si sviluppa rapidamente? Bisogna riconoscere che la dialettica tra ricerca a lunga scadenza e bisogni di commercializzazione è una costante nell'industria informatica. Conosciamo le storie dei gruppi di ricerca industriale che persero i contatti con i bisogni pratici, e sappiamo che queste situazioni hanno condotto a espansioni e ridimensionamenti ciclici. Ci vuole una direzione aziendale forte ed esperta per mantenere il giusto compromesso; ma nel nostro caso il coordinamento non era al punto giusto, e non aveva le conoscenze necessarie. Come dice Piol, il coordinamento era il compito di Roberto Olivetti, l'amministratore delegato, una persona troppo in alto, con molte altre responsabilità e senza conoscenze specifiche nell'industria informatica. Molte delle disfunzioni di cui stiamo parlando possono essere ricondotte a questo difetto strutturale.

Bolognani:

Questo sgradevole e irrisolto confronto competitivo tra le due diverse anime dell'Olivetti – da una parte, la ricerca lungimirante, ma poco ancorata alla realtà del mercato e, dall'altra, la componente commerciale appiattita sul breve termine – era ulteriormente aggravato, come abbiamo visto,

dalla sottovalutazione della complessità del processo produttivo di software da parte di tutti e dall'insufficiente presidio manageriale delle attività da parte dei responsabili del Laboratorio.

Pacelli:

Prima dell'acquisto da parte della General Electric non esisteva alcuna forma di pianificazione integrata tra marketing e sviluppo. Non esistevano obiettivi, non dico a lungo termine, ma anche a pochi mesi, che riflettessero le necessità del mercato. Si lavorava con lo stato mentale di 'quando sarà finito qualcuno lo venderà', e si dava solo (sbagliando) un peso notevole all'innovazione (giusto). Erano due mondi separati, quello dei venditori che all'improvviso avevano trovato un cliente che voleva un compilatore Fortran su sistemi di 10 kbyte a un nastro magnetico, e quello degli sviluppatori che cercavano di sorpassare Ibm. La mia opinione? L'approccio innovativo incoraggiato da Tchou era giusto perché, se volevamo diventare qualcuno, non si poteva competere con l'Ibm vendendo gli stessi prodotti della Ibm (approccio Bellisario e Piol). Ma al tempo stesso occorreva mettere in piedi una sotto-organizzazione per fare delle cose tradizionali, tanto per agganciare qualche cliente in attesa dei prodotti "eccellenti". Oggi si direbbe "eravamo una start-up e i finanziamenti non sono mai arrivati".

Piol:

Non dimentichiamo che a fine 1964 (quando io lasciai la Direzione commerciale elettronica) la Olivetti aveva raggiunto circa il 25% del mercato italiano dei computer. E poiché tali computer gestivano banche (come Monte dei Paschi di Siena e San Paolo), industrie (come Marzotto, Fiat, Eni), amministrazioni pubbliche ecc., significa che in pochi anni si era creata una buona base di software applicativo.

Bolognani:

Il 25% non è male per una start-up.

Logrippo:

L'Olivetti non era una start-up; era un'azienda multinazionale di grande reputazione in Italia e anche all'estero, quanto la Fiat. Aveva macchine da scrivere, calcolatrici, telescriventi, macchine utensili dappertutto ed era naturale che un'azienda che avesse trattato con l'Olivetti per decenni fosse interessata a continuare con lo stesso fornitore per i nuovi prodotti. Questo fu il vantaggio che fu dilapidato. Ho l'impressione che si sarebbe potuto recuperare dal "processo di apprendimento" di cui parla Pacelli solo con uno sforzo enorme, che forse neanche Adriano Olivetti o Mario Tchou avrebbero potuto fare. A media scadenza, con un prodotto di prima

classe e con una direzione aziendale molto determinata, la Divisione elettronica avrebbe potuto operare più a lungo (si veda il caso della Siemens). L'Elea 4001/GE 115 fu finalmente un successo, ma a quel punto la volontà aziendale, già esitante prima, aveva ceduto. Ci si potrebbe chiedere se il 4001 fu un successo perché c'era anche la General Electric. Però a lunga scadenza si lottava contro forze schiaccianti, di certo nel campo dei mainframe. L'intera industria hardware europea fu quasi distrutta nel giro di circa una generazione. Se Vercingetorige avesse fatto meno errori, la storia sarebbe cambiata?

Bolognani:

Io penso che la strada da intraprendere in Italia, ma anche in Europa, fosse allora, come oggi, quella delle tec-

nologie software. Nell'interessante libro scritto di recente da Piol sulle vicende Olivetti di quel tempo e successive, [16] le tecnologie software non sono mai abbastanza a fuoco, se non come accessorio per vendere le macchine. Perché forse pensa, nonostante i tanti esempi contrari nel mondo, che in Italia non si possa fare industria con i prodotti software e i servizi di informatica? Ma forse la chiave di tutto sta nel passaggio in cui Piol accenna di sfuggita al ruolo avuto da Mauro Pacelli nello sviluppo del software per Elea: aveva un approccio da ricercatore universitario! Ma non è così che nascono quelle che lui stesso chiama "tecnologie destabilizzanti"? Piol sostiene che non era il momento e forse aveva ragione; tuttavia è vero che Pacelli e uomini come lui si sono impegnati a lungo nell'industria informatica italiana, senza mai tro-

Il software come snodo per l'innovazione

commento

di Alfonso Fuggetta

Mario Bolognani e Luigi Logrippo ripercorrono una vicenda per certi versi emozionante e per altri dolorosa. È la storia dell'informatica italiana a cavallo tra gli anni '50 e '60, anni nei quali l'Italia ebbe la possibilità di divenire un player a livello mondiale nel settore. È una storia di successi, per certi versi clamorosi, ma anche di paure, di errori complessivi di un "sistema paese" che non fu in grado o non volle sostenere lo sviluppo di un'industria, l'informatica, che è oggi il cuore della società moderna, anche in un momento di crisi e di profonda trasformazione strutturale come quello che stiamo vivendo. Ma quali sono le principali osservazioni che mi sento di sottolineare nella puntuale e appassionata ricostruzione proposta da Bolognani e Logrippo? Un primo punto essenziale riguarda la centralità del software. È il software che determina sempre più il successo di prodotti complessi e sofisticati. Lo era nei primi anni '60 quando, come osservano diver-

si protagonisti menzionati nella ricostruzione di Bolognani e Logrippo, Olivetti diede più importanza all'hardware che al software, al contrario di quello che fece Ibm. Lo è oggi, dove un prodotto come l'iPhone, e il suo Application Store, hanno rivoluzionato il mondo delle telecomunicazioni mobili e dell'elettronica di consumo. Ma lo è più in generale per tanti prodotti moderni, come una lavatrice o un'auto o un radiatore elettrico, che senza software non potrebbero più esistere e funzionare con le prestazioni che oggi essi garantiscono. Un secondo aspetto essenziale che emerge dal "tecnodramma" di Bolognani e Logrippo è la constatazione che l'Italia disponeva di skill, capacità e potenzialità per entrare da protagonisti nel mondo dell'informatica. Fu una colpa grave, non solo di Olivetti, ma dell'intero sistema industriale e politico italiano, non aver sostenuto l'azione di quei manager. Mentre in USA l'industria informatica fu considerata da subito strategica, in

Italia fu sostanzialmente incompresa e snobbata dai più, in primo luogo dalla politica. Ma allo stesso tempo, le vicende accadute nel seguito ci hanno mostrato che i colossi dell'informatica sono nati, si sono sviluppati e si sono anche estinti. Il caso della Digital è per certi versi emblematico. Fu l'azienda che sviluppò i minicomputer, che li portò in massa nelle imprese e nelle università. Fu considerata da molti l'antagonista di Ibm. Poi però, non seppe continuare il suo percorso di innovazione. Fu prima acquisita da un'azienda nata pochi anni prima e sviluppatasi in modo travolgente, Compaq. In seguito, fu la stessa Compaq ad essere acquisita da Hp. Dello storico marchio Digital non è rimasto nulla. Eppure, tutte queste storie riportano non solo casi di "morte di aziende", come la Digital, ma anche di "nascita" di nuovi attori come la Dell o la stessa Apple, rinata dalle sue ceneri nella seconda metà degli anni '90. Quindi non si tratta solo di rimpiangere

vare il credito corrispondente alle capacità. Ricordo che egli, di ritorno dall'Arizona in Italsiel, dove mi chiamò, assieme ad altri ex-olivettiani, ci mise subito a sviluppare un sistema di gestione di basi di dati (era il 1969). Allora non c'era nulla del genere sul mercato, se non qualche cosa di sperimentale di Charles Bachman e, se fossimo stati tempestivi, avremmo anche potuto farcela. Ma Santacroce, l'amministratore delegato della società del gruppo Iri, tagliò l'investimento e ci mise "sul mercato". Forse anche lui pensava che l'approccio fosse troppo da ricercatore universitario?

Piol:

Prima di rispondere a questo interrogativo sul ruolo del software nell'industria informatica, devo fare una precisazione.

Ho cercato, per correttezza e nei limiti del possibile, di esporre le mie idee al momento in cui i fatti avvenivano, anche se poi ho cambiato idea. Per quanto riguarda Mauro Pacelli, è perfettamente vero che al tempo lo consideravamo non solo un 'professore', ma anche un grande teorico che buttava via i soldi dell'azienda per creare linguaggi (Pac e Palgo) che non servivano a niente. Ovviamente mi sbagliavo e nel tempo ho riconosciuto i miei errori di valutazione. Si deve però tener presente, a parziale giustificazione delle mie errate valutazioni, che queste risultavano dalla sommatoria di ignoranza (mia), della pressione commerciale per installare i prodotti, delle limitazioni dell'hardware Elea ecc.

Rimane anche il fatto che si operava molto isolati (a quel tempo, un viaggio in Usa era considerato un grande evento), per cui quando ebbi la visibilità delle attività che hanno

ciò che poteva essere e alla fine non è stato, ma di riconoscere che se anche allora non fummo avveduti come Paese, ciò non significa che non ci siano ora opportunità che non si possano cogliere: i treni partono sempre, bisogna saperci saltare sopra al momento giusto; le competenze e i cervelli li abbiamo anche ora, vanno valorizzati e sostenuti. Un terzo aspetto importante è il bisogno di vision e di una prospettiva di lungo termine. È un concetto che emerge in diversi punti della ricostruzione di Bolognani e Logrippo. In particolare, in un passaggio Logrippo sottolinea il difficile ruolo del management, che deve saper bilanciare i bisogni e le sfide sul breve-medio periodo con la vision di lungo termine. Anche su questo abbiamo molto da imparare oggi.

Quante volte iniziative promettenti vengono soffocate sul nascere, alla prima difficoltà o per mancanza di pazienza, commitment, coraggio? Abbiamo un bisogno disperato di investire sul futuro, con la consapevolezza che innovare è rischioso, può non aver successo. Ma è solo così che si può arrivare a produrre valore, una prospettiva di sviluppo, la capacità di competere e affermarsi in un mercato sempre più globale e difficile. Infine, è molto importante riflettere su alcune considerazioni di Elserino Piol, uno dei protagonisti, nel bene e nel

male, di quegli anni. Nei suoi commenti, ripensando criticamente alla sua esperienza, Piol riprende un tema da sempre caro a Mario Bolognani: il bisogno di costruire e sviluppare strategie originali e vincenti nel campo del software. Il software è un prodotto complesso, una tecnologia difficilmente confrontabile con gli altri settori industriali moderni. Ha peculiarità proprie, come la sua invisibilità e immaterialità; il software può essere facilmente modificato; ha costo marginale nullo; vive a lungo, trasformandosi e adattandosi alle esigenze dell'utenza; può essere facilmente distribuito in qualunque parte del mondo in pochi secondi. Come si gestisce lo sviluppo e la commercializzazione di una tecnologia che ha queste caratteristiche così uniche e distintive?

È un tema che era valido allora e lo è tuttora. Basta guardare alle mosse di giganti come Apple, Microsoft e Google, o al mondo dell'open source. Vi sono trasformazioni radicali in atto. Apple vende applicazioni per l'iPhone a bassissimo costo, valorizzando il concetto della *long tail*. L'open source punta sui servizi e sulla personalizzazione. Google ha sviluppato un'efficacissima combinazione tra ricerca su Internet, pubblicità, "software as a service" e *cloud computing*. Microsoft stessa si sta riposizionando, con un proprio mix di strategie e opzio-

ni, in parte coincidenti con quelle degli attori descritti in precedenza. La stessa Ibm, più volte citata nelle pagine di Bolognani e Logrippo, non è più la stessa azienda degli anni '60. Si è riposizionata su software e servizi.

L'avvento di Internet anche come "mercato del software" ha fatto sì che nascessero centinaia se non migliaia di software house medio-piccole che sviluppano innovazione, creando nuove tipologie di prodotti per bisogni che fino a poco tempo fa erano latenti o di nicchia. I casi di *tripit.com* (gestione dei viaggi) e *evernote.com* (raccolta e gestione di note e informazioni) sono da questo punto di vista emblematici. Il mondo del software è in movimento. Nuove strategie emergono e vecchie convinzioni tramontano. C'è spazio per nuove idee, per invenzioni sia nel prodotto, sia nel modello di business. Più che il tempo della rassegnazione o del rimpianto, è il tempo delle opportunità e delle sfide.

È importante guardare a quanto accade 40 anni fa non per recriminare in modo sterile su ciò che non fu, ma per imparare dagli errori del passato e per ritrovare quelle energie e quelle motivazioni che possano rilanciare l'industria dell'informatica italiana. È una sfida che il nostro paese non può permettersi di sottovalutare o perdere.

poi portato al Cobol, trovai Sacerdoti, Pacelli e compagni impreparati mentalmente a utilizzare gli standard nascenti, rispetto alle invenzioni domestiche. Comunque, è vero, è con persone come Pacelli che si innova e si creano tecnologie competitive. Ma a condizione di operare in un contesto culturale preparato: non era così nella Olivetti degli anni '60 (anche per colpa di persone come me) e certamente non lo era in Italsiel, con una strategia orientata solo alla clientela governativa o quasi-governativa e a creare legami con i potenti di allora.

La situazione in Olivetti è cambiata a partire dal 1970 (grazie anche a Marisa Bellisario) e poi è diventata "industry standard" negli anni '80. Purtroppo, anche qui (anni '80) venne fatto un grande errore: si capì l'enorme importanza di avere una significativa attività software, e io feci alcune acquisizioni significative, ma si affrontò il problema in modo sbagliato e conflittuale con la Olivetti. E poi tutto finì.

Però non è vero che le tecnologie software non sono a fuoco nel mio orizzonte. Se l'Italia avrà un futuro nell'IT, lo potrà avere solo attraverso il software. Ma ultimamente ho visto l'orizzonte offuscato, in quanto tentativi di convincere operatori software ad aggregarsi per creare strutture più efficienti, o incoraggiare nuove iniziative di sviluppo, non hanno avuto successo. Conosco una decina di operatori software con aziende valide, in profitto, ma che non sanno cosa fare da grandi: resistono a ogni tentativo di nuove strategie. E bisogna aggiungere che sarebbe opportuno avere del capitale (per accelerare il processo di integrazione e rischiare su innovazioni di prodotto), ma far rinascere il *venture capital* in Italia sembra una missione impossibile, e lo sto sperimentando. Ma qui il discorso richiederebbe lo scrivere un altro libro.

Concludendo, ritengo che sia giusto fare il possibile per focalizzare l'innovazione italiana sulla tecnologia software, e spero che il mio attuale pessimismo, dovuto a recenti esperienze, venga a cadere.

Epilogo, quasi una conclusione

È interessante notare molti elementi di unanimità tra gli interventi precedenti. Il divario tra l'Olivetti e i suoi concorrenti nel settore informatico era legato ai seguenti fattori: investimenti insufficienti, scarsa sensibilità del valore della componente software dei sistemi con ritardi alla partenza, insufficiente competenza ingegneristica nel software, scarsa esperienza manageriale con seri difetti di pianificazione e di integrazione della attività R&S con quelle della Direzione commerciale, sproporzione nella dimensione dei mercati, morte improvvisa dei due capi creativi. Il disaccordo determinante fu il contrasto di stra-

tegie nel software. Malgrado i vantaggi a lungo termine di una strategia orientata verso l'utilizzazione della migliore ricerca nella teoria dei linguaggi e dei compilatori, ricerca abbondante in Italia a quell'epoca, vinse la linea pragmatica del "facciamo come l'Ibm" e se ne videro i risultati.

Chi si occupa di strategie aziendali sa bene che, all'osso, le strategie possibili sono due: la strategia che punta al contenimento dei costi, oppure la strategia che fa leva sull'unicità. Nel primo caso, l'impresa non si distingue dai concorrenti per l'originalità dei suoi prodotti, ma per la capacità di metterli sul mercato sostenendo costi inferiori. Nel secondo, caso l'obiettivo è quello dell'essere differente dalle altre imprese nella gamma dei prodotti/servizi venduti. Nel primo caso l'innovazione è prevalentemente di processo; nel secondo di prodotto.

Chi si occupa di innovazione sa bene che ogni impresa che opera su un mercato emergente inizia la sua attività con l'innovazione di prodotto e, successivamente, quando i prodotti si sono consolidati, fa prevalere l'innovazione di processo, finalizzata a produrre le stesse cose a costi inferiori.

Anche nelle scelte di politica economica dei governi si manifesta questa dialettica dei due tempi, il tempo dell'innovazione e il tempo del miglioramento della produttività. Sappiamo anche che in Italia i governi successivi alla stagione del boom economico e la maggior parte delle imprese hanno privilegiato il tempo dell'ottimizzazione, del contenimento dei costi, le strategie di costo insomma, rinviando sempre a un dopo asintoticamente lontano la ricerca dell'unicità, la ricerca e l'innovazione di prodotto, finendo per escludere l'Italia dal gruppo dei protagonisti dei mercati dell'alta tecnologia.

Questo profondo contrasto di linee esisteva in seno al gruppo dirigente della divisione elettronica e si manifesta ancora oggi, nonostante i distinguo e l'ammissione di errori compiuti, nelle argomentazioni di Mauro Pacelli e di Elserino Piol. È una differenza di analisi e di obiettivi che non appare di facile composizione.

La scelta tra software "industry-standard" e software di nuova concezione con robuste basi teoriche era certamente non facile. L'indicazione dell'Algol come base per il linguaggio di sviluppo non era campata in aria, come la storia ha poi dimostrato.

L'altra possibilità era di fare come gli altri (copiare l'Ibm) per poi tentare di avere successo con l'aggressività commerciale e con il contenimento dei costi. Questo era il dilemma che non è stato risolto allora e che oggi solo sul piano teorico ci fa dire che la linea dell'innovazione di prodotto doveva essere perseguita per prima, anche nel

software e non solo nelle architetture hardware. Magari con l'approccio da "terza via", suggerito qui da Pacelli con il senno di poi. Purtroppo, non è bastato il fiato per completare l'esperienza e verificare l'assunto teorico. Con la cessione alla General Electric, nel 1964, Pacelli emigra negli Usa e le decisioni sul software vengono dal *Product planning* dove opera, vincente, Marisa Bellisario, mentre il gruppo di sviluppo software, ancora non più di cinquanta persone, rimane decapitato, senza maestri né leader. Né la breve coda General Electric, pur con grandi risorse, ha sciolto la contraddizione. Il software della "nuova linea" GE (abortita con l'uscita dell'azienda dal mercato dei computer nel 1969), affidato per la parte linguaggi al talento di Pacelli, ancora una volta incorporava l'Algol come base per il linguaggio di sviluppo (che, guarda caso, si chiamava C-language). Ma questa volta non solo il software era declinato sull'Algol, ma anche l'architettura della macchina ne era condizionata, sul modello del contemporaneo Burroughs B5000 (infatti, tra i tecnici alle dipendenze di Pacelli in Arizona operava Jack Merner, uno dei progettisti del B5000). Purtroppo, anche questo progetto non ha visto la luce e quindi i nostri dubbi rimangono irrisolti.

E la storia della tecnologia non è un esperimento scientifico, dove normalmente alla fine prevale la migliore soluzione tecnica. I capitali e il campo dei concorrenti hanno enorme peso.

Un secondo punto di ambiguità non risolta riguarda la natura della divisione elettronica: era o non era una start-up? Se si guarda la rapida successione degli eventi, si può affermare che le prime mosse (collaborazione con la Cep, creazione del Laboratorio vicino all'Università, affidamento a ricercatori universitari di compiti rilevanti) erano tipiche di una start-up, ma che ben presto, già nel 1958, con il rientro a Borgolombardo e a Pregnana si capisce che la cultura organizzativa dell'impresa madre, orientata al controllo dei costi, alle economie di scala e al mercato, tende a prevalere, ingabbiando gli slanci innovativi. Invece, su un punto cruciale per le strategie dell'oggi, tutti i nostri interlocutori convergono: la centralità e il primato del software nelle scelte che riguardano i sistemi, primato che ieri era misconosciuto e che ancora oggi stenta a fare breccia in chi dirige le imprese e l'economia in Italia. Il fatto che il software sia oggi incorporato in modelli di business originali, come nel caso Google o nel caso Apple, in cui i ricavi vengono dalla pubblicità o dalla vendita di musica online, o che sia addirittura reso disponibile a tutti come un bene pubblico non rivale (come nel caso del software *open source*) non deve indurre in errore. Nessuna macchina contemporanea e nessun servizio può funzionare senza una ricca

dotazione di software di sistema e applicativo e senza funzionalità ardite e complesse. Anche i produttori storici sopravvissuti, come Ibm e Hp, tendono a qualificarsi ormai come imprese di servizi in cui la componente prodotti software e servizi professionali di informatica è maggioritaria nella struttura dei ricavi.



RIFERIMENTI

- [1] Amodio, Rosario, *Olivetti nella grande sfida internazionale dell'informatica*, intervento tenuto durante il convegno "Olivetti Cento Anni – Olivetti nella grande sfida internazionale dell'informatica", Milano, 15 novembre 2008.
- [2] Bellisario, Marisa, *Donna e top manager*, Rizzoli, Milano 1987.
- [3] Bolognani, Mario, *Bit Generation*, Editori Riuniti, Roma 2004.
- [4] Bonfanti, Corrado, *L'informatica italiana compie cinquant'anni*, Pristem – Note di Matematica, Storia, Cultura, n. 12-13, Università Bocconi, 2005.
- [5] Filippazzi, Franco, *Elea: storia di una sfida industriale*, in Luigi Dadda: "La nascita dell'informatica in Italia", Polipress, Milano, 2006.
- [6] Filippazzi, Franco, *Elea 9003: storia di una sfida industriale*, Università di Udine, 21 marzo 2008, http://nid.dimi.uniud.it/history/papers/filipazzi_08.
- [7] Knuth, Donald E., Pardo, Louis T., *The Early Development of Programming Languages in A History of Computing in the Twentieth Century*, New York, Academic Press, 1980. (Reprinted in Knuth, Donald E., et al., *Selected Papers on Computer Languages*, Stanford, CA, Csl, 2003).
- [8] International Computation Centre, *Symbolic Languages in Data Processing*, Gordon and Beach, 1962.
- [9] Logrippo, Luigi, *My first two computers: Elea 9003 and Elea 6001*, <http://www.csi.uottawa.ca/~luigi/papers/elea.htm>, 2005.
- [10] Logrippo, Luigi, e-mail, 6-2, 7-2, 21-2-2007, 3-3-2009.
- [11] Obermito, Renato, e-mail, 19-2-2007.
- [12] Pacelli, Mauro, e-mail, 11-1-2005, 20-3-2009.
- [13] Petrone, Luigi, e-mail, 11-3-2009.
- [14] Petrone, Luigi, *Syntactic mappings of context-free languages*, Proceedings Ifip '65, New York.
- [15] Piol, Elserino, e-mail, 11-1-2005 e 3-3-2009.
- [16] Piol, Elserino, *Il sogno di un'impresa*, Il Sole 24 Ore, Milano, 2004.
- [17] Restivo, Antonio, *Gli esordi scientifici*, in Storia dell'Informatica in Italia, a cura di Fabrizio Luccio, Edifir per InfoCamere.