

# Il computer come calcolatore – 3

## Storia dell'Informatica e della Comunicazione Digitale

**Federico Gobbo**

federico.gobbo@uninsubria.it

Università dell'Insubria, Varese

© Alcuni diritti riservati.

A.A. 2009-10

# L'informatica diventa piú complessa

In questa sezione vediamo **la terza generazione** dei calcolatori, costruiti mediante la tecnologia dei circuiti integrati. Inoltre vediamo l'atmosfera del MIT nei primi anni 1960, dove nasce l'hacking e come risultato inatteso dell'Intelligenza Artificiale. Dalla parte opposta la pratica di scrivere software diventa piú razionale, e nasce l'ingegneria del software, e viene fondata ARPANET, l'antenata di Internet.

# L'informatica diventa piú complessa

In questa sezione vediamo **la terza generazione** dei calcolatori, costruiti mediante la tecnologia dei circuiti integrati. Inoltre vediamo l'atmosfera del MIT nei primi anni 1960, dove nasce l'hacking e come risultato inatteso dell'Intelligenza Artificiale. Dalla parte opposta la pratica di scrivere software diventa piú razionale, e nasce l'ingegneria del software, e viene fondata ARPANET, l'antenata di Internet.

**Nota Benissimo:** quanto esposto qui è *molto parziale*, e va integrato con il libro di testo, il volume di Paul E. Ceruzzi!

# La legge di Moore della potenza di calcolo

Il transistor è stato inventato nel 1947, e già nel **1958** esce il suo raffinamento, il **circuito integrato**, che permette di scalare rapidamente la potenza di calcolo.

Nell'Aprile **1965** Gordon E. Moore scrive un articolo intitolato *Cramming More Components Onto Integrated Circuits*, dove sostiene che il numero di transistor componibili in un chip raddoppierà ogni anno. Questa previsione, con qualche ritocco, si è rivelata vera, e ha influenzato profondamente la visione sociale del progresso tecnologico dell'informatica.

## Il Tech Model Railroad Club (Tmrc)

Nel 1958 Peter Samson entra al MIT come matricola e si iscrive a un club studentesco dedito al modellismo ferroviario, il *Tech Model Railroad Club*, in cui gli aderenti costruiscono ferrovie comandate elettricamente tramite relé.

Per i membri di quel club, il termine **hack** non significava ‘goliardata’, come per gli altri, ma veniva inteso gergalmente come “innovazione, stile, virtuosismo tecnico”, così come il termine **to log** non era “fare a pezzi”, ma smontare un sistema (di qui, Levy 1994: cap. 1).

## Il corso 641 del MIT, anno 1959

Nel 1959 viene aperto un corso nuovo, dal titolo *Artificial Intelligence*. Lo tiene il professor John McCarthy. Il corso afferisce all'istituto di ingegneria elettrica: nessuno considera i calcolatori un oggetto degno di studio autonomo (“roba da militari”). I membri del Tmrc si iscrivono in massa.

John McCarthy lavora a un linguaggio di programmazione per poter operare sui calcolatori astruendo dal linguaggio macchina. La macchina a disposizione era un IBM 704, che era proibito anche avvicinare. Gli operatori in camice bianco, chiamati spregiativamente *clercks* ‘chierici’ dai membri del Tmrc.

# John McCarty fonda l'Intelligenza Artificiale

John McCarthy (classe 1927) ottiene il PhD in matematica a Princeton nel 1951 e pensa di applicare i risultati della matematica costruttiva di Church ai calcolatori. Con l'aiuto degli studenti del corso 641 disegnerà il linguaggio di programmazione **LISP**, basato in parte sul lambda calcolo, che diventa il primo linguaggio di riferimento per l'I.A. (le applicazioni piú note scritte in LISP oggi sono AutoCAD e Emacs.)

Nella relazione invitata per il centenario del MIT (1961) sostiene che grazie al *time sharing* il software potrà **scorrere come l'acqua o l'elettricit ** nella societ  (**metafora della rete**). Nel 1971 otterr  il Turing Award per i suoi risultati di I.A., termine da lui coniato.

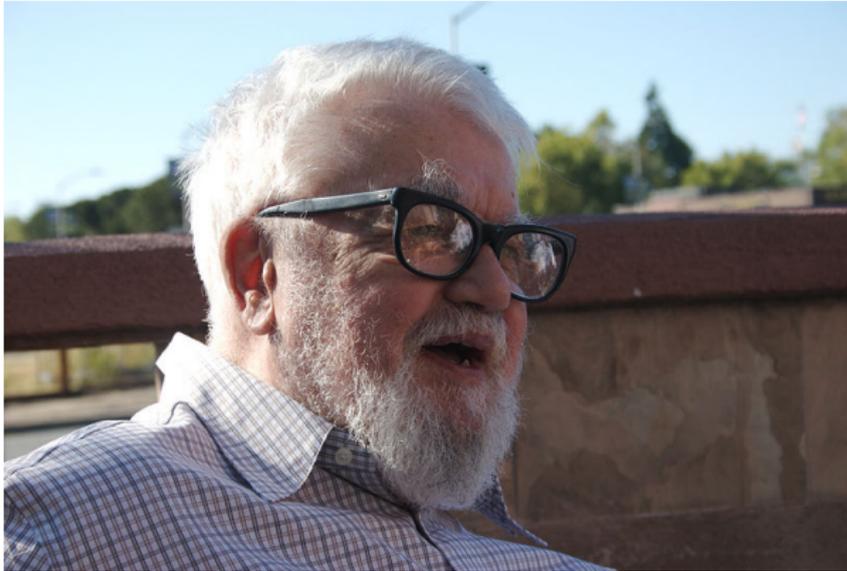


Figura: John McCarthy nel 2006

## La metafora della rete è nell'aria

Lo psicologo Joseph C.R. Licklider, già membro del SAGE, pubblica nel 1960 un articolo dal titolo *Man-Computer Symbiosis*, in cui postula l'interazione tra la “logica artificiale” basata sulla nozione di algoritmo e la “logica naturale” basata sulla nozione di **euristica**.

La sua idea, per certi versi analoga a McCarthy, è usare la rete dei centri SAGE per scopi **cognitivi** anziché militari: ogni centro ha un “terminale interattivo” dove l'operatore si interfaccia con i computer, che dialogano tra loro per fornire informazioni elaborate computazionalmente.

## Al MIT arriva il Tx-0, detto 'Tixo'

Il Lincoln Lab del MIT nel 1956 costruisce il **Transistorized eXperimental computer Zero (Tx-0)**, uno dei primi calcolatori completamente a transistor, avente 64.000 parole a 18 bit, basato su una reingegnerizzazione del Whirlwind. Occupava una sola stanza (anziché un intero piano...) e aveva un video di 7x7 pollici con una risoluzione di 512x512 pixel, inserito in un oscilloscopio.

La macchina disponeva di quattro istruzioni, che permettevano di registrare in memoria, aggiungere e fare i jump. Questa "astronave stile H.G. Wells" (Levy 2000:25) cambia la vita agli studenti di McCarty: l'input è una Flexowriter, che inserisce un nastro perforato anche lungo, che nell'elaborazione produce dei suoni simili a un organo elettronico.



Figura: Il Tx-0

# Hacker: non camici bianchi ma scarpe da tennis

Con Tixo niente piú chierici: gli studenti di McCarty cominciano a giocare a scacchi con Tixo, con cui vivono in simbiosi, in una specie di comunità, i cui membri si autonominano **hacker**. Questi i fondamenti della loro etica:

- L'accesso ai computer dev'essere illimitato e completo.
- Tutta l'informazione dev'essere libera.
- Dubitare dell'autorità. Promuovere il decentramento.
- Gli hacker devono essere giudicati per il loro operato, razza o posizione sociale.
- Con un computer puoi creare arte.
- I computer possono cambiare la vita in meglio.

## Arriva il PDP-I, il primo minicomputer

Nel 1961 i ragazzi del Tmrc mettono le mani sul primo minicomputer del mondo, il PDP-I della neonata DEC. Grosso come circa tre frigoriferi, con un forte impianto di raffreddamento, l'esemplare del MIT è il secondo in ordine di costruzione, ed era costato solo \$ 120.000.

Mentre con Tixo si scriveva in FORTRAN, con PDP-I si scrive in LISP, che però occupa molta memoria – non era stata ancora inventata l'idea del *garbage collection*. Gli studenti in blocco seguono un altro corso al MIT, dal titolo *Introduzione all'I.A. 6.544*, tenuto da Marvin Minsky, il quale incoraggia i nuovi hacker e la loro relazione con le macchine in maniera esplicita: “gli esseri umani sono macchine di carne”, esclama.



Figura: PDP-1

# Marvin Minsky

Marvin Lee Minsky (classe 1927) consegue il PhD in matematica nel 1954 a Princeton, dal 1958 lavora al MIT. Turing Award nel 1969, registra diversi brevetti. Assieme a Seymour Papert inventa il linguaggio di programmazione **LOGO** e scrive il controverso libro *Perceptrons*, che negli anni 1970 è concausa della caduta d'interesse nel paradigma connessionista. Minsky e Papert formulano la teoria della **Società delle Menti**, oggi titolo del suo corso al MIT.

Nel 1986 esce *Robotics*, un libro dal taglio piú divulgativo. Nel 2006 ha pubblicato **The Emotion Machine**.

Tra l'altro è stato consulente per *2001: A Space Odyssey* (libro di Arthur C. Clarke, film di Stanley Kubrick). Il suo sito è <http://web.media.mit.edu/~minsky/>.



Figura: Marvin Minsky nel 2006

# Il linguaggio di programmazione LOGO

Il LOGO è un buon esempio di applicazione del paradigma dell'I.A. nel campo della didattica: il programmatore muove una tartaruga sul piano, dandogli dei comandi per disegnare o semplicemente spostarsi. In questo modo si imparano i principi della trigonometria e si possono disegnare figure anche molto complesse (Abelson-Disessa 1986).

```
PER XXX  
  RIPETI SEMPRE  
    AVANTI 1  
    DESTRA 1
```

# Il linguaggio di programmazione LOGO

Il LOGO è un buon esempio di applicazione del paradigma dell'I.A. nel campo della didattica: il programmatore muove una tartaruga sul piano, dandogli dei comandi per disegnare o semplicemente spostarsi. In questo modo si imparano i principi della trigonometria e si possono disegnare figure anche molto complesse (Abelson-Disessa 1986).

```
PER XXX  
  RIPETI SEMPRE  
    AVANTI 1  
    DESTRA 1
```

Che cos'è XXX?

# Il linguaggio di programmazione LOGO

Il LOGO è un buon esempio di applicazione del paradigma dell'I.A. nel campo della didattica: il programmatore muove una tartaruga sul piano, dandogli dei comandi per disegnare o semplicemente spostarsi. In questo modo si imparano i principi della trigonometria e si possono disegnare figure anche molto complesse (Abelson-Disessa 1986).

```
PER XXX
  RIPETI SEMPRE
    AVANTI 1
    DESTRA 1
```

Che cos'è XXX?

Un cerchio!

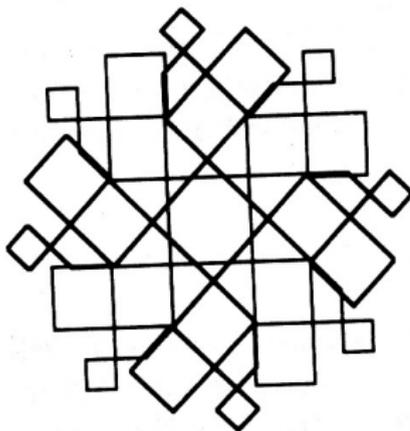


Figura: Semplici disegni fatti con LOGO (Abelson-Disessa 1986:33)

# La Società delle Macchine delle Emozioni

Nel 1986 Minsky postula che l'intelligenza sia una **proprietà emergente** dall'interazione tra agenti non intelligenti e porta argomenti filosofici a sostegno della sua tesi. Il libro uscì doppio, con la versione ipertestuale su CD-ROM.

Vent'anni dopo, Minsky postula una nuova I.A. dove parcellizzare i processi mentali in eventi cerebrali al fine di riprodurli artificialmente (non robot che puliscono la casa, ma che possono ridere a una barzelletta). Esempio: il Sé non è un Ente ma una *decentralized cloud*. Estratti del libro sulla Macchina delle Emozioni sono disponibili sul suo sito.

# Project MAC e derivati

Nel 1963 il DARPA (organo militare Usa) finanzia il MIT con 2 milioni di dollari per un progetto di ricerca, a cui prende parte tra gli altri anche Minsky. L'obiettivo è rendere disponibile la potenza di calcolo degli elaboratori come l'acqua potabile: per la prima volta **la metafora caratterizzante dei computer è la Rete.**

Tra i primi risultati del **Project MAC** annoveriamo il concetto di **time-sharing**, sulla base del quale fu costruito il **Multics**, un elaboratore di nuova concezione, che per la prima volta aveva il concetto di **sistema operativo (SO)**: la macchina sostituisce (sic!) l'*operator* umano.

Dal 2007 il MIT ha rilasciato il sorgente del Multics a fini di studio: <http://web.mit.edu/multics-history/>.

## J.C.R. Licklider

Joseph Carl Robnett Licklider (1915–1990), dopo aver conseguito il PhD in psicoacustica nel 1942, si interessa delle tecnologie dell'informazione ad Harvard negli anni della guerra e nel 1950 diventa professore associato al MIT.

Dopo aver lavorato al SAGE, nel **1960** scrive l'articolo *Man-Computer Symbiosis*, dove postula la **grafica come interfaccia** mediante computazione interattiva.

Nel 1968 diventa il direttore del Project MAC e scrive l'articolo *The Computer as a Communication Device*, che dà il via ad **ARPAnet**.



Figura: J.C.R. Licklider

# ARPAnet, l'antenata di Internet

Il dipartimento della difesa americano, per limitare i danni di un attacco nucleare americano, intende mettere gli elaboratori in rete così le informazioni non sono centralizzate, disassemblando le informazioni in pacchetti (**packet switching**).

Grazie soprattutto a Licklider, vennero coinvolti quattro elaboratori: alla UCLA, all'*Augmentation Research Center* di Stanford, al centro IBM dell'UC di Santa Barbara, e l'Università dello Utah.

Il 29 ottobre **1969** viene trasmesso il primo messaggio.

29 OCT 69	2100	LOADED OP. PROGRAM FOR BEN BARKER BBW	CSK
	22:30	Talked to SRF Host to Host	CSK
		Left op. program running after sending a host dead message to imp.	CSK

Figura: Il log del primo messaggio su ARPANET (il "lo" [gin])

# Douglas Engelbart

Douglas C. Engelbart (classe 1925), tecnico radar durante la guerra, consegue il PhD in ingegneria elettrica ed informatica (EECS) a Berkeley nel 1955. Influenzato dalle idee di Vannevar Bush sulle potenzialità della tecnologia sulla mente umana, inizia la sua carriera di ricerca a Stanford. Nel 1962 scrive il suo manifesto: *Augmenting Human Intellect: a Conceptual Framework*, poi partecipa ad ARPAnet.

Costruisce il primo sistema interattivo ad interfaccia grafica, inventando uno strumento di puntamento, chiamato **mouse**. Sviluppa le idee di Bush concependo il concetto di **applicativo**: programmi per scrivere o disegnare semplici da usare per **gruppi di utenti** in collaborazione, secondo l'idea di ipertesto. Nel 1968 fa quella che fu chiamata la "madre di tutte le demo".



Figura: Il primo mouse

# La teoria dell'accrescimento cognitivo

Engelbart introduce una micropolitica per le interfacce, vale a dire tutti i programmi dovevano usare gli stessi meccanismi di rappresentazione cognitiva, così da poterli condividere: nasce il concetto di **GUI standard**.

L'obiettivo di Engelbart era di rendere il computer uno strumento adeguato ad "accrescere" (*augment*) il funzionamento dei gruppi, per accompagnare e dirigere in modo morbido la co-evoluzione degli uomini e degli strumenti.

Il sistema Augment finale prevedeva tra le altre cose un laboratorio di disegno, di programmazione, e diversi cataloghi di documenti, oltre a un aiuto in linea incluso nel programma.

## Cybersyn: esperimento di democrazia elettronica

Le idee di Bush, Licklider ed Engelbart hanno un potenziale di trasformazione della società enorme. Il governo cileno di Salvador Allende nell'autunno del **1971** chiama Antony Stafford Beer, cibernetico britannico, per assumere la direzione scientifica del progetto **Cybersyn**: cybernetics + synergy. La cibernetica dava i modelli per analizzare il sistema, la sinergia sottende che gli elementi del progetto hanno ampi margini di autonomia.

Con una rete basata sul telex governo, aziende, scuole, accedevano a banche dati comuni mediante dei Datafeed. Il colpo di stato militare di Pinochet (1973) mette fine a Cybersyn instaurando la dittatura.

## Cybersyn: esperimento di democrazia elettronica

Le idee di Bush, Licklider ed Engelbart hanno un potenziale di trasformazione della società enorme. Il governo cileno di Salvador Allende nell'autunno del **1971** chiama Antony Stafford Beer, cibernetico britannico, per assumere la direzione scientifica del progetto **Cybersyn**: cybernetics + synergy. La cibernetica dava i modelli per analizzare il sistema, la sinergia sottende che gli elementi del progetto hanno ampi margini di autonomia.

Con una rete basata sul telex governo, aziende, scuole, accedevano a banche dati comuni mediante dei Datafeed. Il colpo di stato militare di Pinochet (1973) mette fine a Cybersyn instaurando la dittatura.

In Italia l'unico articolo su Cybersyn fu scritto nel 1977 dal professor Gaetano Aurelio Lanzarone (De Cindio-De Michelis 1980:29).



Figura: Una stanza di controllo operativo di Cybersyn con i Datafeed

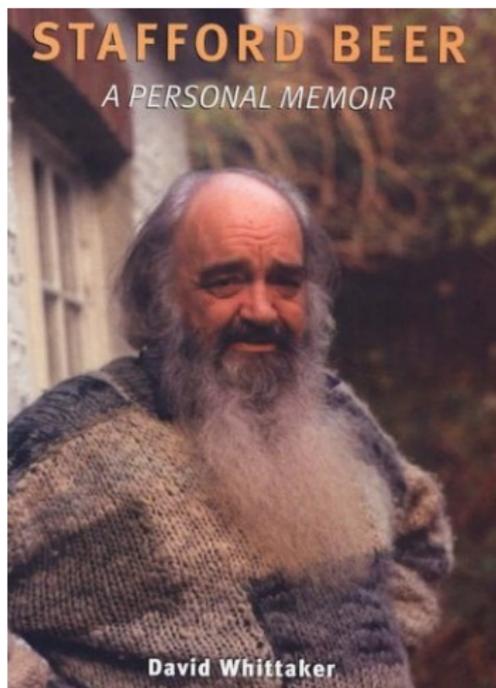


Figura: Copertina della biografia di Stafford Beer (2003)

## Quei lunghissimi 20 minuti, anno 1960

Il 5 ottobre **1960** Thomas J. Watson Jr., presidente IBM, era al quartiere generale della difesa a Colorado Springs: il computer aveva intercettato degli oggetti non identificati entrare nel cielo americano. Il livello di allarme era 5: 99,9 per cento che fosse un missile nucleare sovietico.

Non c'era tempo per un **controllo visivo**: che fare? Il colonnello canadese Slemon, a 18 mila piedi di quota, fa la domanda fondamentale: "dov'è Chruščëv?" Il presidente URSS, il rinnovatore del dopo Stalin, era a New York. **L'operatore decide che l'URSS non stava attaccando, contrariamente a quanto diceva il computer.**

## Quei lunghissimi 20 minuti, anno 1960

Il 5 ottobre **1960** Thomas J. Watson Jr., presidente IBM, era al quartiere generale della difesa a Colorado Springs: il computer aveva intercettato degli oggetti non identificati entrare nel cielo americano. Il livello di allarme era 5: 99,9 per cento che fosse un missile nucleare sovietico.

Non c'era tempo per un **controllo visivo**: che fare? Il colonnello canadese Slemon, a 18 mila piedi di quota, fa la domanda fondamentale: "dov'è Chruščëv?" Il presidente URSS, il rinnovatore del dopo Stalin, era a New York. **L'operatore decide che l'URSS non stava attaccando, contrariamente a quanto diceva il computer.**

Il computer aveva erroneamente visto la luna sorgere dietro la Norvegia e l'aveva classificata come un missile intercontinentale (McKenzie 2001)

# Il software non è piú comprensibile dal singolo

Con il progetto SAGE e con le missioni Apollo degli anni 1960 il ruolo del software diventa sempre piú importante: non esiste piú nessun singolo essere umano che conosca tutto il software di un sistema complesso. La crisi del 5 ottobre 1960 fa sorgere la domanda: che succede se il software di un sistema ad alta criticità contiene un errore?

Uno degli ingegneri IBM leader del progetto System/360, scritto in ALGOL, Frederick P. Brooks, Jr., sostiene che “a ogni aggiornamento OS/360 introduce in media 1000 errori nel sorgente” (in McKenzie 2001:31).

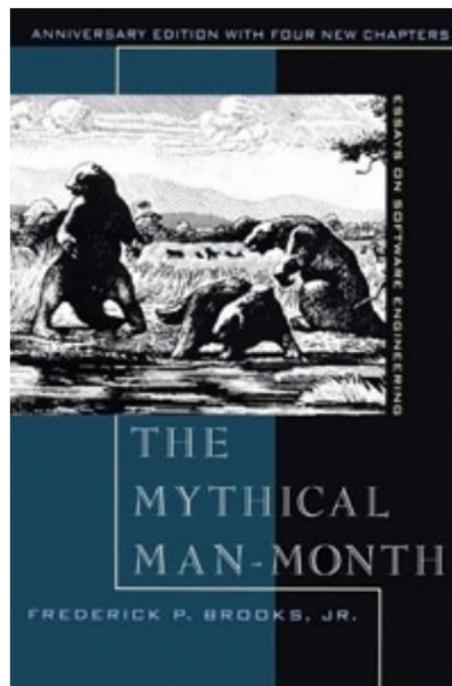


Figura: Sulla difficoltà di amministrare progetti software grossi

## Il software costa sempre di piú

Negli anni 1960 ci si accorge che, mentre l'hardware costa sempre meno, il numero di livelli di astrazione nella programmazione diventa sempre piú alto e quindi la scrittura di software viene a costare, in percentuale, sempre di piú. Occorre dunque che la programmazione diventi qualcosa di non piú artigianale ma professionale.

Esempio: nel 1955 il programmatore Barry Boehm è pagato \$2 l'ora, la macchina viene noleggiata a \$600 l'ora. Nel 1966 il costo di sviluppare software per il System/360 dell'IBM era salito a circa \$600 l'ora (McKenzie 2001: 34–35).

## Nasce la programmazione strutturata

Nel **1968** viene organizzata la prima conferenza per parlare di queste cose, a Garmisch-Partenkirchen (Baviera). Emerge come figura Edsger W. **Dijkstra** (1930–2002), teorico dell'informatica, il cui sistema operativo scritto per la Technische Hoogeschool Eindhoven era stato lodato per la semplicità e l'eleganza, ricordato di solito per una citazione (1969): “program testing can be used to show the presence of bugs, but never to show their absence!”.

Dijkstra descrive l'**ingegneria del software** in termini di **programmazione strutturata**: analogamente alle dimostrazioni matematiche, la programmazione va scomposta in sottoproblemi da risolvere uno per volta: **divide et impera** (McKenzie 2001:37).

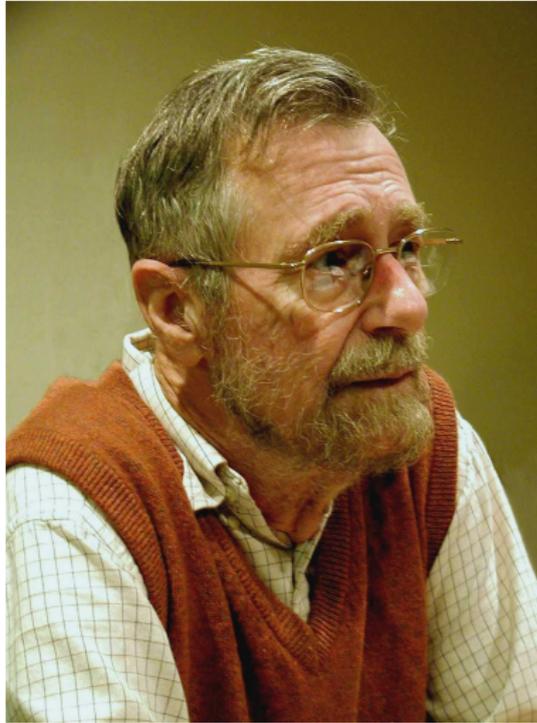


Figura: Edsger Dijkstra

# Classificazione della conoscenze data dal software

McKenzie (2001:2) ci ricorda le fonti della conoscenza scientifica:

- **induzione:** apprendimento mediante osservazione, esperimenti, test;
- **autorità:** persone degne di fiducia ci dicono quali sono le conoscenze;
- **deduzione:** inferiamo le proprietà a partire da credenze o postulati, mediante teorie o modelli.

L'ingegneria del software introduce la deduzione nella programmazione, che fino ad allora era dovuta all'induzione (se il programma lo facevi da te) o dall'autorità (se il programma ti veniva passato da qualcun altro).

# Classificazione dei modelli di produzione del software

Campbell-Kelly (2004:9) classifica la produzione del software in tre modelli:

- **software nel contratto**: il software viene dato come parte del progetto – dal 1956 al 1968 circa, con progetti come SAGE o MAC;
- **software come prodotto corporate**: il software viene valutato come prodotto di mercato, e viene venduto con contratti di assistenza, anche senza l'hardware – dal 1968 al 1977 circa, con compagnie nuove come SAP o Oracle;
- **software come prodotto di massa**: il software viene venduto come un prodotto per il (nascente) mercato dei consumatori – dal 1977 circa, con compagnie che non producono hardware, quali Microsoft o Lotus.

# Corrado Böhm

Corrado Böhm (classe 1923) durante la guerra studia ingegneria elettrica a Losanna. Nel 1954 ottiene il dottorato in matematica con una tesi in cui scrive un compilatore metacircolare, cioè scritto nel linguaggio del linguaggio programmazione. Ricercatore allo IAC-CNR di Roma negli anni 1950, si occupa soprattutto di lambda calcolo, MdT, architettura von Neumann, semantica dei linguaggi di programmazione: nel 1966 enuncia il teorema di Böhm-Jacopini.

Nel **1970** ottiene la prima cattedra di informatica d'Italia, presso l'Università di Torino. Dal 1974 insegna alla Sapienza di Roma. La sua pagina web: <http://www.dsi.uniroma1.it/~boehm/>.



Figura: Corrado Böhm (sinistra), con Ida, Vestergaard, Antoy, Lucas al workshop WRS01 tenuto a Utrecht (foto di Francisco Correa)

## Primo sabotaggio dell'Olivetti (Bolognani 2004)

Nel **1960** muore Adriano Olivetti in circostanze tragiche, e l'anno dopo muore anche il progettista Mario Tchou. La miopia del gruppo di intervento finanziario per sostenere l'azienda (Fiat, Pirelli, Mediobanca, Imi, Centrale) cerca di estirpare il “neo” informatico dall'azienda.

Solo un piccolo gruppo di ricerca e sviluppo, capitanato da Pier Giorgio Perotto, si oppone ai “contafagioli” (nel gergo olivettiano, gli amministrativi).

## L'Olivetti negli anni 1960 (de Witt 2005)

L'Olivetti del dopo Adriano nel **1956** fa uscire, grazie al progettista Natale Capellaro, la **Divisumma**: una macchina calcolatrice elettrica capace di fare le quattro operazioni su cifre di 12 caratteri, dotata di **stampante** incorporata. Nel 1967 ne vengono prodotte un milione di pezzi, imponendo l'Olivetti come leader mondiale.

Negli anni 1960 l'Olivetti inventa le **telescriventi**, macchine da scrivere che trasmettono sulla linea telefonica direttamente, e le vende al Ministero delle Poste.

La fatturatrice **Mercator** fa un salto ulteriore: fatta a transistor, è dotata di registri di memoria e velocizza i conti. Per la prima volta al mondo una macchina calcolatrice viene venduta per la scrivania degli uffici.



Figura: Olivetti Divisumma 18 (fonte: Freaknet Computer Museum)

## Programma 101, il primo desktop computer al mondo

A Pregnana Milanese nel 1962 si era costituita la Divisione Elettronica Olivetti, con finalità di ricerca applicata, ceduta nel 1964 alla General Electric americana. Grazie all'interesse di Roberto Olivetti e Natale Capellaro, Pier Giorgio Perotto (1930–2002), “progettista riottoso”, già membro del gruppo di Barbaricina con Mario Tchou, continua a lavorare con il suo piccolo a una macchina di tipo nuovo. Nel **1965** l'Olivetti presenta il risultato di Perotto e del suo gruppo a New York: la **Programma 101**, che riscuote un grande successo negli Usa, dove vengono vendute il 90% delle 40.000 prodotte.

Piccola, usabile da una segretaria, P101 ha una stampante incorporata e permette di fare calcoli scientifici ed è programmabile con un set di 10 registri di memoria e fino a 120 istruzioni su una cartolina magnetica esterna (De Witt 2005:32).

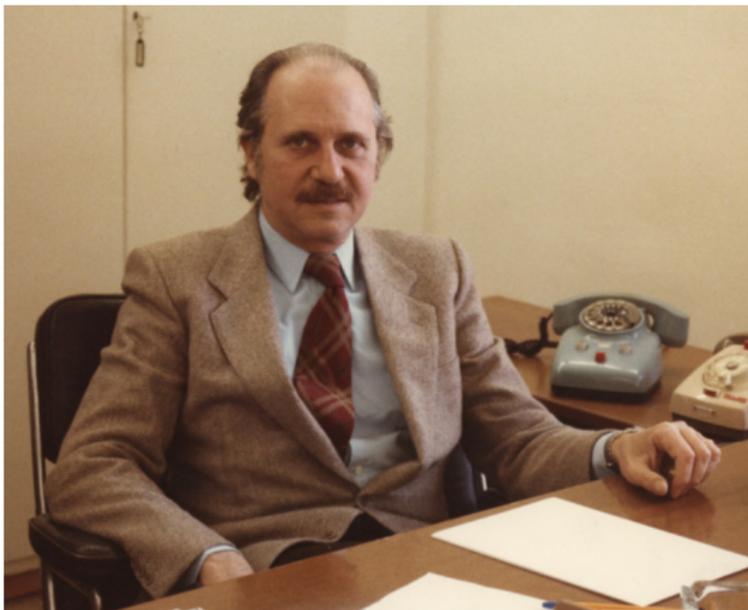


Figura: Pier Giorgio Perotto (archivio storico Olivetti)

# Ma?!? Perché Olivetti non è diventata la Apple?

Con le parole dello stesso Perotto (Bolognani 2004:13):

*Il guaio fu che, dopo l'exploit del Programma 101, non si riuscì a "controllare" lo sviluppo delle architetture nel campo dell'informatica distribuita. Si sarebbe dovuto, dopo il primo prodotto, far uscire con grande rapidità nuove versioni aggiornate e allargare subito la gamma dei prodotti, in modo da occupare tutti gli spazi, dettando gli standard di fatto del nuovo immenso mercato che si apriva. Ma le risorse mancavano e si diede tutto il tempo ai concorrenti di occuparlo.*

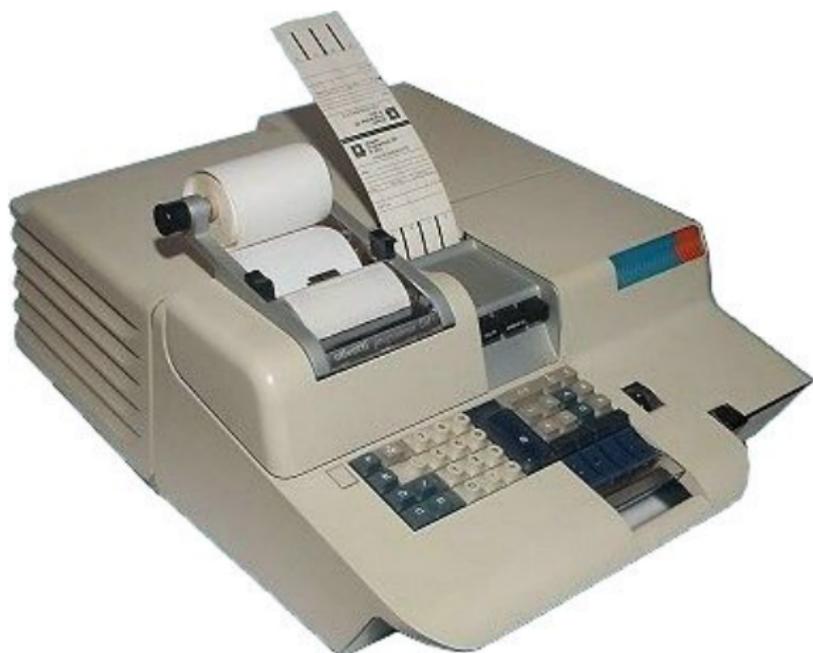


Figura: Olivetti Programma 101 (archivio storico Olivetti)



Figura: Interno dell'Olivetti Programma 101 (archivio storico Olivetti)

# Grazie. Domande?



Potete scaricare questa presentazione qui:

<http://www.slideshare.net/goberiko/>

© (CC) (BY) (NC) (SA) Federico Gobbo 2009 di tutti i testi. Pubblicato in Italia.  
Attribuzione – Non commerciale – Condividi allo stesso modo 2.5

© delle figure degli aventi diritto. In caso di violazione, scrivere a: [federico.gobbo@uninsubria.it](mailto:federico.gobbo@uninsubria.it).