

BOLLETTINO

GIATED

GRUPPO INTELLIGENZA ARTIFICIALE - ATED  
ASSOCIAZIONE TICINESE ELABORAZIONE DATI

NUMERO 1  
FEBBRAIO 1988

ATED  
casella postale 949  
6830 Chiasso

## PRESENTAZIONE

Nel giugno 1987, per iniziativa dell'ATED su spinta di un attivo gruppo di soci, è stata indetta una riunione preliminare per costituire un "Gruppo Intelligenza artificiale" che, riunendo i soci interessati a questa disciplina ed operando nell'ambito degli scopi istituzionali della associazione, potesse essere un valido centro di interesse, di informazione e crescita professionale. La proposta è stata accolta entusiasticamente da un gran numero di iscritti e si è proceduto subito alla costituzione di una commissione di coordinamento per definirne ed organizzarne le attività.

L'idea di pubblicare un bollettino relativo del Gruppo Intelligenza Artificiale dell'ATED è sorta quasi contemporaneamente al consolidamento dell'attività della commissione per tenere informati tutti coloro che hanno manifestato interesse per l'iniziativa sulle attività del gruppo e sulle novità di rilievo del settore. Quasi contemporaneamente è maturata l'idea di gestire la enorme mole di informazioni e di riferimenti che si accumulava (centinaia di libri, prodotti, corsi, migliaia di articoli) mediante una banca dati.

Ed hanno così avuto origine i due principali strumenti di comunicazione tra gli aderenti al gruppo. Il nostro foglio e la banca dati si propongono, unitamente alle altre iniziative, di fornire una panoramica il più possibile aggiornata sulla evoluzione del settore, raccogliendo e distribuendo tutti gli elementi utili per una visione panoramica e fornendo i riferimenti per approfondimenti più specialistici. Si viene in tal modo incontro ad uno dei principali problemi con cui quotidianamente ci troviamo confrontati: la continua necessità di aggiornamenti in un settore che, oltre che essere in continua evoluzione, è anche in violenta espansione. Agendo singolarmente siamo infatti costretti ad operare continui "tagli" sugli argomenti proposti restringendo, a vantaggio di una necessaria specializzazione, la visione generale dei problemi.

E' chiaro che questo bollettino nasce per nostra iniziativa e deve rimanere vivo con i contributi di tutti. E' auspicabile che diventi, unitamente alla banca dati del gruppo, una centrale di scambio non solo di notizie ma anche di idee e di esperienze contribuendo così alla continua crescita professionale cui tutti tendiamo.

Questo primo numero ha un carattere introduttivo. Pertanto alcuni articoli sono, più che trattazioni tecniche avanzate, note per avvicinare alla materia chi, pur avendo dimostrato interesse per questa disciplina, si deve costruire le basi per gli studi successivi.

Il nostro bollettino inizia con un incoraggiante saluto del prof. Rolf Pfeifer presidente dello SGAICO (Swiss Group for Artificial Intelligence and Cognitive

### INDICE:

- Presentazione ..... pag. 1
- Saluto del presidente dell' ATED ..... pag. 2
- Benvenuto al gruppo intelligenza artificiale in Ticino ..... pag. 2
- Bollettini elettronici ..... pag. 4
- Biblioteca di I.A. : PROLOG ..... pag. 6
- Informazioni sulla Banca Dati ..... pag. 7
- Attività della commissione ..... pag. 7
- Sistemi esperti a regole ..... pag. 8
- I.A. Prodotti per Apple Macintosh ..... pag. 10
- Attività LOGO nelle classi Speciali ..... pag. 12

## PRESENTAZIONE

Science), che mostra chiaramente quanto la nostra iniziativa abbia suscitato interessi non limitati al solo ambito ticinese e debba essere vista come un elemento in un contesto più vasto che comprende altre analoghe associazioni con cui sarà oltre che interessante, estremamente utile collaborare.

Il gruppo ritiene di fondamentale importanza per poter operare con efficienza e scioltezza, l'utilizzo di strumenti avanzati. Per questo, contemporaneamente alla pubblicazione di questo bollettino, è stato messo in funzione un "electronic bulletin board" per la raccolta e la distribuzione di informazioni, notizie e comunicazioni. In questo primo numero Roberto Fisch introduce il concetto di bollettino elettronico spiegandone efficacemente con parole chiare la sua struttura ed il suo utilizzo.

E' prevista una segnalazione periodica di riferimenti bibliografici. Tendenzialmente testi di carattere generale saranno presentati con un commento, mentre si segnaleranno semplicemente testi più specialisti, di interesse limitato ad un ristretto numero di lettori. Per questi ultimi si rimanda alle apposite pagi-

ne della banca dati. La didattica, in tutti i suoi aspetti, deve essere considerata una parte fondamentale del processo conoscitivo in una società evoluta. Affrontare i problemi della didattica in generale e quelli della didattica dell'informatica in particolare, rientra quindi nel campo di attività del nostro gruppo. Gabriele Scascignhini presenta un'approfondita riflessione sull'utilizzo del linguaggio Logo nella pratica educativa corrente.

Chi è interessato ai sistemi esperti troverà un interessante articolo di Silvano Marioni sui sistemi a regole di produzione e nella nostra rubrica "Biblioteca di I.A." un primo elenco di testi Prolog. Non poteva infine mancare una rubrica relativa ai prodotti per A.I. Iniziamo la nostra panoramica sul mercato con una prima presentazione di Carlo Lepori di ciò che è disponibile per Apple Macintosh.

Buona lettura dunque ed arriverci al prossimo numero.

Renzo Guglielmini

## SALUTO DEL PRESIDENTE DELL' ATED GIANCARLO ZÜGER - Centro Cantonale di Informatica, Bellinzona

Sono lieto di presentare questo bollettino che vuole essere nelle intenzioni dei promotori uno strumento di informazione e di aggiornamento sul tema dell'Intelligenza Artificiale. L'Associazione Ticinese Elaborazione Dati si è interessata a questo tema negli anni passati con alcune conferenze a livello informativo. Oggi l'interesse per le tecniche di Intelligenza Artificiale è cresciuto, ed è desiderio di molti il volere approfondire l'argomento ed esaminare le possibili applicazioni di questa tecnica nella professione informatica. Grazie all'impegno del Gruppo Intelligenza Artificiale, l'ATED vuole così andare oltre

la semplice divulgazione e dare la possibilità di indagare una disciplina che sta diventando oggi sempre più importante. La pubblicazione di questo bollettino è un ulteriore passo nella diffusione e nel perfezionamento delle conoscenze professionali, e ad esso auguro di raccogliere l'interesse dei soci e di tutta la realtà informatica ticinese.

Giancarlo Züger

Presidente dell'Associazione Ticinese Elaborazione Dati

## BENVENUTO AL GRUPPO INTELLIGENZA ARTIFICIALE IN TICINO PROF. ROLF PFEIFER - Ist. di Informatica dell'università di Zurigo

E' con piacere che ho appreso la notizia della creazione di un gruppo I.A. in Ticino. Nonostante esista da poco tempo si è già distinto in alcune attività. Una di queste è la collaborazione al seminario

"Aspetti dell'intelligenza artificiale", a cui ho avuto modo di partecipare come relatore, restando ben impressionato dell'interesse dei partecipanti.

Come voi tutti probabilmente saprete, Lugano ha una lunga tradizione nell'I.A. che risale a 15 anni fa (questo vale relativamente ai trent'anni della breve storia dell'I.A.). Risale agli anni settanta la creazione dell'ISSCO da parte della Fondazione Dalle Molle ed é di quegli anni la presenza a Lugano di ricercatori ad alto livello nel settore dell'Intelligenza Artificiale quali Roger Schank, Eugene Charniak, Margaret King (che é oggi direttrice dell'ISSCO), Yorick Wilks e Manfred Wettler (che era il direttore ai quei tempi). Personalmente il mio primo contatto con l'I.A. fu a un tutorial sulla "Semantica computazionale" che ebbe luogo a Villa Heleneum a Castagnola nel 1975. Ne fui talmente colpito che tornai al mio lavoro pieno di idee e con molto entusiasmo. Il tutorial fu in effetti una pietra miliare e molti dei partecipanti sono ora degli affermati ricercatori nel settore dell'I.A.

In tempi piú recenti, nel 1985, lo SGAICO, il Gruppo Svizzero per l'Intelligenza Artificiale e le Scienze Cognitive, ha organizzato a Lugano un tutorial sui "Sistemi basati sulla conoscenza" con la partecipazione di famosi relatori provenienti dall'Europa e dagli Stati Uniti. Il successo ottenuto é venuto non solo dal livello qualitativo della manifestazione, ma anche dal fatto che Lugano é una citta adatta a questo tipo di avvenimenti. Piccola a sufficienza per non sentirsi persi ma grande abbastanza da disporre di tutte le infrastrutture necessarie, la popolazione é molto cordiale e le autorità si sono sempre mostrate sensibili alle idee dell'Intelligenza Artificiale sostenendo le manifestazioni in questo settore.

Vorrei a questo punto anticipare che lo SGAICO ha in programma altre due manifestazioni a Lugano: una é un simposio nel giugno 1988 su "Applicazioni dell'Intelligenza Artificiale nella finanza, nelle banche, e nelle assicurazioni", l'altro sar  un corso avanzato di due settimane sull'I.A. che avr  luogo nell'estate del 1989. Questo corso ACAI (Advanced Course in Artificial Intelligence) é organizzato ogni due anni da una delle societ  membre

dell'ECCAI, il Comitato di Coordinamento Europeo per l'Intelligenza Artificiale.

Ho inteso che un istituto sull'Intelligenza Artificiale é stato creato da poco a Lugano. Sono completamente d'accordo con questa iniziativa e spero che l'istituto possa diventare completamente operativo nel piú breve tempo possibile. Ritengo che Lugano sia il posto giusto per questa attivit . In nessun'altra parte in Svizzera ho trovato cos  tanto interesse per l'I.A. tra i professionisti del settore dell'elaborazione dati.

Permettetemi di concludere con un paio di considerazioni sullo SGAICO. Anche se come associazione ha raggiunto una dimensione di rispetto nei confronti della realt  svizzera (circa 300 soci) é ancora piccola in termini assoluti. Come voi tutti sapete c'  un "problema di massa critica" per cui é necessario che un certo numero di persone lavori sullo stesso problema se si vuole ottenere un risultato. Questo credo che valga anche a livello politico: é necessario che ci sia un certo numero di persone che promuova l'interesse per l'Intelligenza Artificiale e le scienze cognitive in Svizzera. Per questo ritengo che tutte le organizzazioni interessate all'argomento debbano riunire le forze e lavorare insieme. Spero quindi in una buona cooperazione tra il nuovo gruppo dell'Associazione Ticinese Elaborazione Dati e lo SGAICO in modo che possano esserci benefici per entrambi le parti. Desidero quindi fare al Gruppo Intelligenza Artificiale i miei migliori auguri di successo e ancora una volta mi congratulo con i promotori per l'ottimo lavoro svolto.

Rolf Pfeifer

Presidente dello SGAICO, Swiss Group for Artificial Intelligence and Cognitive Science

## Introduzione

Questo articolo persegue due scopi: introdurre e spiegare in cosa consiste un bollettino elettronico e contemporaneamente gettare le basi per la definizione della banca dati GIA-ATED, che è attualmente a disposizione dei nostri associati.

In funzione di questi obiettivi tratteremo principalmente gli aspetti rilevanti per questo tipo di utilizzo e parleremo solo marginalmente delle altre possibili applicazioni, per esempio quelle in campo commerciale.

## Definizione

Il termine bollettino elettronico deriva dall'inglese "electronic bulletin board", che può essere tradotto con "albo elettronico". In realtà questo tipo di servizio va oltre la funzione di un albo, ma alle origini questa era la funzione principale.

Scopo di un bollettino elettronico è fornire tre tipi di prestazioni:

- 1) Permettere a chi lo gestisce di trasmettere informazioni, notizie e comunicazioni agli utenti.
- 2) Offrire agli utenti la possibilità di scambiarsi messaggi, sia privati (destinati a una sola persona), sia pubblici (destinati a tutti gli utenti). Questi messaggi possono contenere informazioni di carattere generale, domande relative a problemi incontrati o prese di posizione su temi di attualità nell'ambito degli argomenti trattati.
- 3) Servire come serbatoi di materiale: lunghi testi, raccolte di dati, programmi. Gli utenti autorizzati possono inviare (trasmettere) questo materiale al bollettino e, in un secondo tempo, altri possono riceverlo (trasferirlo) sui loro sistemi.

Tutte queste funzioni sono svolte da una serie di programmi operanti su un elaboratore, spesso un semplice Personal Computer, collegato alla rete telefonica. Gli utenti si collegano ad esso tramite un terminale o un elaboratore e interagiscono dialogando con il sistema.

## Materiale necessario

Chi volesse dotarsi in proprio di un bollettino elettronico necessita di tre componenti: un elaboratore, un programma particolare e un MODEM. Come abbia-

mo già anticipato, per applicazioni di piccola-media scala ci si può servire di un personal computer: ciò comporta il vantaggio di contenere i costi e inoltre di poter disporre di programmi ampiamente collaudati. Questi programmi sono di norma ottenibili gratuitamente presso bollettini già in funzione. L'installazione e l'esercizio richiedono un minimo di esperienza sul tipo di elaboratore che si utilizza.

Il MODEM (Modulatore-Demodulatore) è l'apparecchio che permette al computer di collegarsi alla rete telefonica e quindi di rispondere alle chiamate degli utenti. Tramite due cavi, uno inserito nell'elaboratore e uno nell'attacco del telefono, viene creato un ponte fra i segnali acustici della linea telefonica e quelli digitali del computer.

Anche chi vuole essere utente di questi bollettini deve dotarsi (o essere già in possesso) di un elaboratore (dall'home computer in su), di un MODEM e di un programma per le telecomunicazioni.

I MODEM possono essere acquistati o affittati presso le PTT (informazioni tramite il no. 113), rispettivamente acquistati presso fornitori autorizzati. Il loro costo varia dalle poche centinaia al migliaio di franchi a seconda delle prestazioni. Un tipo particolare di MODEM sono gli accoppiatori acustici: si tratta di apparecchi ai quali si applica direttamente la cornetta del telefono dopo avere ottenuto risposta dal sistema al quale ci si vuole collegare. I loro vantaggi sono il basso costo e la portabilità, gli svantaggi la bassa velocità di trasmissione e la qualità, influenzata da fattori esterni come rumori o la forma del telefono.

I programmi per le telecomunicazioni permettono all'elaboratore di inviare e ricevere comandi o informazioni a un altro sistema al quale si è collegati. I costi sono molto contenuti: se ne trovano di gratuiti con buone prestazioni, rispettivamente si deve pagare qualche centinaio di franchi per ottenere quelli di tipo professionale. Da notare che tutti i programmi "integrati" (tipo Framework, Symphony, Open Access, Enable, ecc.) contengono già moduli per le comunicazioni.

## Modalità di collegamento

Il collegamento con un bollettino elettronico avviene in modo molto semplice: si compone il numero telefonico e, quando il sistema risponde, ci si annuncia tramite il proprio nome. Se è la prima volta che ci si collega si viene pregati di fornire alcune indicazioni

che permettano agli operatori di verificare l'identità di chi chiama e prevenire in questo modo usi illeciti.

Segue la richiesta di una "password" (parola chiave) che l'utente stesso sceglie e che lo identifica presso il sistema, eliminando la possibilità che altri usino il suo nome e possano accedere alle informazioni a lui riservate.

Terminate queste operazioni preliminari viene eventualmente trasmesso un testo informativo di carattere generale o su di un tema di attualità (comunicazioni di servizio, novità, annunci). A partire da questo punto l'utente può interagire liberamente con i vari servizi del bollettino, fino al momento in cui decide di terminare il collegamento.

### Struttura

I bollettini elettronici sono normalmente organizzati in aree: esse corrispondono alle tematiche offerte agli utenti.

Nel caso dell'intelligenza artificiale si potrebbero avere, oltre ad un'area generale, aree dedicate ai linguaggi (PROLOG, LISP, SMALLTALK ecc.), ai sistemi esperti, alla robotica, alla visione, alla lingua naturale, ai prodotti, alla bibliografia e così via.

Si differenziano due tipi di area: l'area messaggi e l'area archivi.

In un'area messaggi gli utenti possono leggere le comunicazioni introdotte dagli altri, fornire risposte o introdurre contributi originali. Queste aree servono quindi sia come sistemi di posta elettronica, sia come conferenze elettroniche. Con queste si intende lo scambio di opinioni e esperienze fra più persone, senza che queste debbano trovarsi fisicamente nello stesso luogo; ognuno potrà infatti aggiornarsi periodicamente su quanto detto dagli altri e intervenire in merito.

Nelle aree archivi si trovano testi, programmi o dati che possono essere trasferiti dall'utente sul proprio elaboratore utilizzando appropriati comandi del programma di comunicazioni. Grazie a questo sistema diventa possibile in tempi molto brevi mettere a disposizione di una vasta cerchia di interessati nuovi programmi di dominio pubblico, articoli interessanti o raccolte di informazioni e dati. Questi archivi vengono introdotti dall'operatore del bollettino oppure dagli utenti stessi.

### Modalità di utilizzo

L'utilizzo dei bollettini è semplice in quanto l'utente viene sempre guidato da menu di opzioni fra le quali

egli può scegliere. In ogni momento egli può richiedere una breve spiegazione dei comandi disponibili.

Una sessione con il bollettino si svolge di norma secondo lo schema seguente: una volta allestito il collegamento l'utente consulta l'albo vero e proprio, che consiste in testi e comunicati di interesse generale preparati dai curatori del bollettino e richiamabili a partire dal menu di entrata.

In una seconda fase può interessare andare a vedere alcune delle aree messaggi per verificare se sono presenti invii indirizzati direttamente all'utente o per seguire lo svolgimento di conferenze.

In una terza fase avviene una verifica delle aree archivi, per immettere del materiale che si vuole mettere a disposizione degli altri utenti o per trasferirne sul proprio sistema.

### Sicurezza

Pur avendo una vocazione pubblica e quindi di apertura, è ovvio che esistono dei meccanismi di controllo per evitare che l'uso del bollettino venga disturbato da elementi esterni alla cerchia degli interessati.

Oltre alla password personale di ogni utente, i curatori del bollettino (in inglese "system operators" o "sysops") hanno a disposizione una serie di opzioni per assicurare un corretto esercizio del sistema. Gli utenti vengono suddivisi in categorie aventi differenti diritti di utilizzo dei comandi ed è inoltre possibile riservare l'accesso a singole aree solo a certi gruppi. Si può quindi affermare che questi sistemi offrono ogni garanzia di serietà e di sicurezza sia a chi li gestisce sia a chi li utilizza.

### Conclusione

Come detto nell'introduzione, l'obiettivo di questo articolo e quindi la sua ragione di essere all'interno del primo numero del nostro giornale, era di creare le premesse e stuzzicare l'appetito per l'utilizzo della banca dati GIA-ATED. Riteniamo che la nuova frontiera dell'intelligenza artificiale debba poggiare sulle nuove tecnologie e farne il massimo uso. In questo senso vogliamo sfruttare questo mezzo per accelerare e semplificare i collegamenti fra chi nella Svizzera Italiana mostra un interesse per queste tematiche. Affiancando all'informazione su carta quella elettronica, i promotori intendono dotarsi di uno strumento moderno per svolgere meglio il proprio compito.

**W.D. Burnham, A.R. Hall, Prolog Programming and Applications, Macmillian, Londra 1985**

Un'introduzione completa al Prolog in un libro di poco più di cento pagine. Per motivi di semplicità non sono trattate le basi teoriche del Prolog ma non per questo il libro perde di rigore. Una presentazione sintetica ma completa, ideale per chi in poco tempo vuole conoscere le possibilità di questo linguaggio.

**J.B. Rogers, A Prolog Primer, Addison Wesley, Reading 1986**

Concepito per l'insegnamento, il libro ha un'impostazione che privilegia la pratica sulla teoria. Numerosi e chiari esempi aiutano a comprendere lo stile di programmazione del linguaggio Prolog. Ideale per chi vuole acquisire le nozioni di base del linguaggio.

**I. Bratko, Prolog programming for Artificial Intelligence, Addison Wesley, Reading 1986**

Il libro comprende nella prima parte un'introduzione al linguaggio e nella seconda parte la dimostrazione delle possibilità del Prolog nelle principali aree dell'Intelligenza Artificiale. E' un buon compromesso di presentazione delle tecniche pratiche di programmazione e delle idee della programmazione logica alla base del Prolog.

**L. Sterling, E. Shapiro, The Art of Prolog, MIT Press, Cambridge 1986**

A tutt'oggi forse il libro più completo sul Prolog. Suddiviso in quattro parti, programmazione logica, Prolog, programmazione avanzata e applicazioni, il libro è caratterizzato da grande chiarezza e rigore. Concepito per un corso avanzato sul Prolog potrebbe essere utilizzato anche da principianti per approfondire alcuni concetti di base.

**C. Markus, Prolog programming, Addison Wesley, Reading 1986**

Il libro è un'introduzione pratica al linguaggio e dà indicazioni su come costruire applicazioni in Prolog. Argomenti quali i sistemi esperti, l'elaborazione del linguaggio naturale, i database relazionali sono trattati con esempi di programmi. Viene preso come riferimento il linguaggio Arity Prolog.

**G. Casadei, A. Teolis, Prolog, dalla programmazione all'Intelligenza Artificiale, Zanichelli, Bologna 1986**

Introduzione al linguaggio concepito sia come testo per l'insegnamento che come manuale di autoistruzione. Nella seconda parte esamina in modo chiaro e didattico la risoluzione di semplici applicazioni.

**W. Clocksin, C. Mellish, Programmare in Prolog, Franco Angeli, Milano 1986**

Traduzione italiana di quello che è stato per lungo tempo l'unico libro sul Prolog. Presenta il Prolog e la sua teoria di base con un'impostazione di tipo accademico. Recentemente lo stesso Mellish ha affermato che se avesse immaginato il successo del libro lo avrebbe scritto in un altro modo. Spesso citato, non dovrebbe mancare nella libreria del programmatore Prolog.

**P. Cianciarini, Prolog linguaggio e applicazioni, Jackson, Milano 1987**

Il libro è una esauriente introduzione al linguaggio e contiene numerosi esempi di programmi e di tecniche di base. In un'appendice sono confrontate tra loro diverse versioni commerciali di Prolog.

**G. Adorni, S. Gaglio, L. Massone, Manuale di Intelligenza Artificiale, La Nuova Italia Scientifica, Roma 1987**

L'obiettivo del libro va oltre la semplice divulgazione dei concetti di base del Prolog. Dopo un'introduzione essenziale al linguaggio, la seconda parte presenta applicazioni non banali nei settori della robotica e visione, comprensione del linguaggio naturale

fino alla riscrittura in Prolog del sistema esperto MYCIN.

**P. Robinson, Programmazione in Turbo Prolog, McGraw Hill, Milano 1987**

Libro interessante se si accettano i limiti del Turbo Prolog della Borland. Il libro, come la versione del linguaggio, non copre argomenti fondamentali del Prolog quale ad esempio la metaprogrammazione. Numerosi esercizi introducono il lettore all'uso di questa versione di Prolog.

## INFORMAZIONI SULLA BANCA DATI

Quando è stato deciso di preparare questo bollettino la banca dati era solo nelle previsioni. I tempi redazionali sono però abbastanza lunghi e al momento di andare in stampa ci siamo trovati con la banca dati già in funzione. Ci è quindi sembrato giusto dare una breve segnalazione del materiale che è possibile trovare sulla banca dati, almeno in riferimento agli articoli comparsi su questo numero.

### GE SE LN RO VI DI LI MO

Si possono scaricare, tra gli altri, i seguenti programmi:

- PDPROLOG.ARC Interprete PROLOG
- LOGO.ARC Interprete LOGO
- ESIE.ARC Expert System Shell
- XLISP.ARC Interprete LISP

Ogni gruppo comprende diversi files compressi, funzionanti nell'ambiente MS-DOS

## ATTIVITA' DELLA COMMISSIONE DI COORDINAMENTO

### ATTIVITA' DELLA COMMISSIONE DI COORDINAMENTO

Nel periodo da giugno a dicembre 87 la commissione si è riunita 5 volte per definire ed organizzare la attività del gruppo. Si è iniziato con la definizione dei settori di interesse (**GE**nerale, **L**inguaggi, **RO**botica, **L**inguaggi **N**aturali, **MO**delli, **S**istemi **E**sper-ti, **V**isione, **DI**dattica - le cui sigle compaiono in testa agli articoli pubblicati). In seguito sono maturate le idee relative alle attività da svolgere:

- raccolta e distribuzione di informazioni (bibliografia, corsi e seminari, prodotti, aziende e specialisti..)
- formazione (organizzazione di corsi)

- contatti con altre associazioni analoghe.

La prima attività si è concretizzata con:

- l'organizzazione di questo giornale
- l'organizzazione della banca dati

La seconda con l'organizzazione di un seminario introduttivo al linguaggio LISP e con la collaborazione con la fondazione Dalle Molle per la organizzazione del corso "Aspetti della Intelligenza Artificiale". Analoghe iniziative sono previste per i prossimi mesi. I verbali delle riunioni sono a disposizione di chi ne faccia richiesta presso la segreteria dell'ATED.

# SISTEMI ESPERTI A REGOLE, INTRODUZIONE

SILVANO MARIONI - Banca Solari & Blum, Lugano

GESE LN O VI DI LI MO

*"Difficile non è sapere, ma saper  
far uso di ciò che si sa"*

proverbio cinese

## Introduzione

Alla fine degli anni 60, Edward Feigenbaum aveva messo a punto presso l'Università di Stanford il programma DENDRAL, un sistema di aiuto per l'analisi spettrografica. La principale caratteristica di questo programma era la separazione tra la definizione della conoscenza, intesa come l'insieme dei dati con le regole operative sui dati stessi, e il modo di utilizzarla. Era l'idea che avrebbe portato ai Sistemi Esperti.

Ancora oggi la definizione classica di Sistema Esperto lo vede composto da tre elementi di base: la base di conoscenza, il motore di inferenza e l'interfaccia uomo-macchina. La base di conoscenza raccoglie le conoscenze di un esperto in un particolare settore, ed è organizzata secondo un appropriato metodo di rappresentazione della conoscenza. Il motore di inferenza è la parte di programma capace di dedurre o di indurre delle conclusioni a partire dalle informazioni della base di conoscenza ed estrarre le soluzioni ai quesiti posti dall'utente.

L'interfaccia uomo-macchina è la parte di programma che si occupa del dialogo con la macchina e permette all'esperto di istruire il sistema e all'utente di fare il lavoro di consultazione.

Per costruire un Sistema Esperto è necessario quindi avere a disposizione un esperto per raccogliere le sue conoscenze e il suo modo di ragionamento e trasferirlo nel programma.

## La conoscenza

La conoscenza che si può avere di un oggetto può essere molto differenziata ma in genere può essere classificata in tre tipi. Conoscenza superficiale, cioè le relazioni tra causa ed effetto ricavate spesso in modo empirico. Un esempio può essere la frase "Se l'auto non parte e il serbatoio è vuoto fai il pieno" che ci dà semplicemente una regola di comportamento. Conoscenza profonda, cioè il modello dell'oggetto, il suo comportamento, la sua struttura. Un esempio è la frase "L'auto funziona con la benzina contenuta nel serbatoio" che dà delle informazioni di base utilizzabili in diverse linee di ragionamento. Da

ultima un tipo particolare di conoscenza, la "metacoscienza" o "conoscenza sulla conoscenza" che tratta la risoluzione del problema da un punto di vista strategico. Un esempio è la frase, "Se l'auto non parte controlla la benzina nel serbatoio, eventualmente controlla se è un piccolo guasto e da ultimo chiama un meccanico", che da indicazioni sui metodi di risoluzione del problema in base alle conoscenze che si vengono via via a determinare.

## La base di conoscenza

Il primo problema che si pone quando si vuole costruire una base di conoscenza è la scelta del modo in cui esprimere la conoscenza nel programma. Tra i diversi formalismi, le regole sono state le prime ad essere usate forse perché più vicine al ragionamento umano. Una regola si compone di una o più condizioni e una o più conclusioni. In genere tutti i sistemi accettano più condizioni ma non tutti permettono più conclusioni.

Le regole si basano su un procedimento logico chiamato "modus ponens" che permette di affermare conclusioni a partire da una o più condizioni. Considerando la seguente regola:

*se il sole è coperto, allora il tempo è brutto*

utilizzando il "modus ponens" si ottiene che se vera la condizione "il sole è coperto" è vera anche la conclusione "il tempo è brutto". Esiste anche un procedimento logico chiamato "modus tollens" che permette di negare le condizioni a partire dalla negazione delle conclusioni. Se consideriamo la stessa regola, utilizzando il "modus tollens" si ottiene che se è falsa la conclusione "il tempo è brutto" è falsa anche la condizione "il sole è coperto". Mentre il "modus ponens" permette di lavorare sul concetto di verità il "modus tollens" opera sul concetto di falsità. Tutti i sistemi a regole utilizzano il "modus ponens" mentre pochi sistemi utilizzano entrambi i procedimenti.

Un vantaggio dei sistemi a regole è la modularità: le singole regole possono essere aggiunte o tolte dalla base di conoscenza senza interferire con le altre.

Un altro vantaggio è la facilità di scrittura e di comprensione poiché è abbastanza naturale espri-



mere i concetti sotto forma di regole. Lo svantaggio principale è il calo delle prestazioni quando il numero delle regole cresce. Inoltre il fatto che i sistemi a regole sono più adatti a rappresentare conoscenze superficiali piuttosto che conoscenze profonde, pone dei limiti nelle applicazioni di una certa complessità.

Oltre alla base di conoscenza esiste un elemento importante nel funzionamento di un sistema esperto che viene in genere chiamata memoria di lavoro e che serve per conservare i risultati temporanei durante l'evolvere di una consultazione.

## Il motore di inferenza

Il potere della conoscenza deriva dalla sua possibilità di utilizzazione. Nei Sistemi Esperti basati su regole il motore di inferenza è l'elemento che permette di dare vita in modo dinamico alle informazioni statiche della base di conoscenza. Se vogliamo fare una classificazione sulla base della teoria di funzionamento possiamo dire che i motori di inferenza si dividono in due gruppi: *motori d'ordine 0* e *motori d'ordine 1*. Questi nomi, che derivano dal calcolo simbolico, indicano la possibilità di lavorare con espressioni del calcolo delle proposizioni (motori di ordine 0) o la possibilità di lavorare con espressioni del calcolo dei predicati (motori di ordine 1). La grossa differenza pratica tra i due sistemi è la possibilità, presente solo nel calcolo dei predicati, di utilizzare le variabili. Vediamo di chiarire con un esempio. Nel caso di un motore di inferenza di ordine 0 (calcolo delle proposizioni) la seguente regola :

*se Socrate è uomo, e uomo è mortale, allora Socrate è mortale*

permette di dedurre che Socrate è mortale. Se si volesse renderla valida per altre persone bisognerebbe aggiungere tante regole quante sono le persone da considerare. Nel caso di un motore di inferenza di ordine 1 (calcolo dei predicati) la regola si presenta nel seguente modo :

*se X è uomo, e uomo è mortale, allora X è mortale*

L'identificatore X si comporta come una variabile e quindi una sola regola è valida per tutte le persone. Dal punto di vista dell'implementazione un motore di inferenza di ordine 0 è sostanzialmente basato sulla gestione di una tabella a più dimensioni. Un motore di inferenza di ordine 1 ha invece una logica

più complessa perchè si basa sull'algoritmo di unificazione.

Una ulteriore classificazione dei motori di inferenza riguarda il modo in cui analizzano ed utilizzano le regole. Esistono in effetti due strategie che corrispondono a due modi di ragionamento. Nel primo caso l'utente può fornire a priori tutti i fatti di cui è a conoscenza per ottenere dal programma una o più conclusioni. Se alcuni di questi fatti fanno rendere vera una regola questa aggiunge alla memoria di lavoro la sua conclusione e questo può far ricominciare un nuovo processo di deduzione fino a quando non è più possibile dedurre nuovi fatti. Questo modo di procedere, chiamato *forward chaining* o concatenamento in avanti, è guidato dai fatti ed è tipico del ragionamento deduttivo.

Nel secondo caso il meccanismo funziona all'inverso. A partire da una conclusione data come ipotesi, il sistema tenta di soddisfare le condizioni che portano alla sua verifica. Per fare questo il programma pone le domande per verificare ognuna delle condizioni all'utente. Può darsi per che una condizione sia la conclusione di un'altra regola e questo porta il motore di inferenza ad iniziare un nuovo processo all'indietro alla ricerca delle nuove condizioni da verificare.

Quando tutte le condizioni sono state soddisfatte, la conclusione definita come obiettivo è considerata vera. Questo modo di procedere, chiamato *backward chaining* o concatenamento all'indietro, è guidato dagli obiettivi ed è tipico del ragionamento induttivo o per ipotesi. In genere il *backward chaining* è usato nella risoluzione di problemi di diagnosi mentre il *forward chaining* è più usato nei problemi di classificazione.

Esistono motori di inferenza che sopportano solo il *backward chaining* o solo il *forward chaining* o che possono utilizzare entrambi i metodi all'interno della medesima consultazione.

## Interfaccia uomo-macchina

E' questo l'elemento indispensabile per la creazione e il funzionamento di un Sistema Esperto e si compone in genere di un modulo di acquisizione della conoscenza e un modulo di consultazione. Attraverso il modulo di acquisizione della conoscenza l'esperto può costruire la base di conoscenza introducendo le regole generalmente espresse con una sintassi simile al linguaggio naturale. Generalmente al momento dell'introduzione delle regole viene fatto

un controllo sintattico e in alcuni casi anche un controllo sulla coerenza della base di conoscenza con la verifica degli eventuali conflitti. In alcuni sistemi più semplici questo modulo può mancare e la base di conoscenza può essere costruita utilizzando un normale programma di elaborazione testi.

Il modulo di consultazione deve permettere all'utente di usare il Sistema Esperto e, senza avere nessuna conoscenza dell'argomento, ottenere le risposte alle sue richieste quasi come se stesse dialogando con l'esperto. Al momento della progettazione del Sistema Esperto devono essere previste tutte le domande da porre all'utente in modo da ottenere tutti i fatti e le condizioni che non sono deducibili da altre regole.

Un'altra prestazione del modulo di consultazione deriva dal fatto che la conoscenza è espressa in forma esplicita. Questa caratteristica permette di utilizzare le regole per dare spiegazioni sulla linea di ragionamento che il sistema sta seguendo. Possono essere quindi chieste spiegazioni sul "come" una certa conclusione è stata raggiunta o sul "perché" una domanda viene posta. La risposta sarà data con

la visualizzazione della regola che in quel preciso contesto viene presa in considerazione dal motore di inferenza o del testo esplicativo ad essa associata.

### Gestione dell'incertezza

Una delle caratteristiche dei Sistemi Esperti è la capacità di fornire dei risultati anche con informazioni incomplete. Infatti, a differenza dei programmi tradizionali, possono funzionare in situazioni dove l'utente non sa dare risposte a determinate domande o non le sa dare con certezza.

Per risolvere queste situazioni in cui non si può rispondere con un deciso sì o no si fa ricorso all'utilizzo di tecniche matematiche particolari per la gestione dell'incertezza. I metodi più usati sono i fattori di certezza, il calcolo probabilistico con l'uso del teorema di Bayes e la logica sfumata o fuzzy logic.

La scelta della tecnica più appropriata dipende dal problema che si deve risolvere e dal modo in cui l'incertezza può propagarsi nella catene di regole. L'approfondimento di queste tecniche sarà il tema di uno dei prossimi articoli.

## INTELLIGENZA ARTIFICIALE: PRODOTTI PER APPLE MACINTOSH

CARLO LEPORI - Scuola Tecnica Superiore, Trevano

E' evidente l'utilità di conoscere i prodotti che riguardano l'intelligenza artificiale, a disposizione sui calcolatori più diffusi. Iniziamo con il materiale disponibile per il Macintosh della Apple (i prodotti citati dovrebbero funzionare su Mac +, Mac SE e Mac II). Si conta di tener aggiornata questa lista con la presentazione di nuovi prodotti e di estenderla anche ai calcolatori MS-DOS, ecc. Le fonti utilizzate sono: MAC up 3/87, MACWORLD Oct. '87, ecc.

### 1.- Linguaggi funzionali: LISP

#### XLisp 1.5 Public Domain

Interprete Lisp che permette anche la programmazione a oggetti, disponibile anche per altre macchine. Non dispone di Editor incorporato il che rende complicato l'uso. Non dispone di compiler. Dynamical binding.

Coral Lisp Coral Software - \$ 100 (introductory)

#### GE SE LN RO VI DI LI MO

Versione ridotta di Allegro Common Lisp, comprendente compilazione incrementale, programmazione a oggetti, editore, ecc.

#### MacScheme 1.11 Semantic Microsystem - \$ 395

Implementazione del dialetto Scheme, sviluppato al MIT (usato p.es. in Abelson e.a., Structure and interpretation of Computer programs, MIT Press 1985): precursore del Common Lisp con sintassi propria. Ambiente confortevole; accesso al ToolBox, compilatore, applicazioni indipendenti, ecc. (Solo l'interprete: \$ 125)

#### Allegro Common Lisp 1.1 Coral Software & Franz Inc. - \$ 600

Implementazione completa di Common Lisp (basata sul testo: Guy Steele, Common Lisp the Language, Digital Press 1984, fornito quale manuale!), con programmazione a oggetti; compilazione incre-

mentale, disattivabile per il debugging; editore incorporato (programmabile) che lavora Mac-Style e EMACS-Style. Ambiente molto confortevole. Accesso a parte del ToolBox (menu, finestre, dialoghi, ecc.) come oggetti; a tutto il resto con funzioni low-level. Disponibile anche l'implementazione dei "Flavors"; previsti l'interfaccia per funzioni in altri linguaggi e un compilatore per applicazioni indipendenti.

**Le\_Lisp 15.1 Act Informatique - FF 4800**

Implementazione francese a interprete di buona portabilità tra workstations. Tendenza Common Lisp; accesso al ToolBox; programmazione a oggetti e compiler annunciati.

**ExperLisp 1.5 ExperTelligence - \$ 790**

Prodotto più conosciuto: dispone di interessanti possibilità grafiche (le "turtles" ereditate da ExperLogo); lavora a compilazione incrementale e permette anche di creare applicazioni indipendenti dall'ambiente Lisp. Editor incorporato. Tendenza Common Lisp (lexical binding).

**ExperCommonLisp 2.1 ExperTelligence - \$ 1390**

Compilazione come ExperLisp; implementazione quasi completa di Common Lisp (sono state citate alcune riserve!); programmazione a oggetti incorporata (il ToolBox implementato come sistema di classi).

**2.- Linguaggi funzionali: Logo**

**Object Logo 1.5 Coral Software - \$ 80**

Logo con programmazione a oggetti (anche le "turtles" sono oggetti), compilazione incrementale, editore integrato, capacità matematiche (interi grandi, frazioni e complessi) e trattamento di liste pari al Lisp. Ambiente molto confortevole. Accesso al ToolBox, oggetti Macintosh; compatibile con Apple-Logo; interfaccia MIDI disponibile.

**ExperLogo 1.1ExperTelligence \$ 150**

Logo a compilazione incrementale con editor incorporato; grafica ("turtles", dette "bunnies") sul piano, sulla sfera e 3D. Riprende anche molti elementi del ExperLisp.

**3.- Linguaggi dichiarativi: Prolog**

**Modula-Prolog BBC - Fr. 500**

Interprete nello stile di Edinburgo, scritto in Modula-2. L'applicazione è (era?) a disposizione gratuitamente a Zurigo. La licenza della BBC di Baden comprende la sorgente e permette di interfacciare l'interprete con procedure scritte in Modula-2. Manca un editor incorporato e l'accesso diretto al ToolBox, compresa la grafica.

**Personal Prolog Optimized Systems Software - \$ 75**

Interprete nello stile di Edinburgo, in ambiente con più finestre e possibilità grafiche.

**Wisdom Prolog MIT Press - \$ 109**

Interprete nello stile di Edinburgo: esistono anche versioni MS-DOS e UNIX. Shell definibile da utente, possibilità di usare editor e altre applicazioni senza lasciare l'interprete. Sono disponibili su dischetto (\$ 17) tutti gli esempi del libro "The Art of Prolog" di Sterling e Shapiro.

**AAIS Prolog Advanced A.I. Systems - \$ 150**

Interprete nello stile di Edinburgo, in ambiente con più finestre e possibilità grafiche. Compatibile con molti dialetti Prolog.

**ProLogo Plus 1.0 ExperTelligence - \$ 175**

Interprete scritto in ExperLogo, con la sorgente a disposizione!

**Prolog II 2.2 PrologIA - DM 1365**

Prodotto francese che non segue la sintassi di Edinburgo, di buona portabilità tra workstations. Comprende ampliamenti del linguaggio tra cui la possibilità di programmazione modulare. Accesso alla grafica e a procedure scritte in altri linguaggi.

**MacIF/Prolog 3.0 InterFace - DM 1710**

Interprete nello stile di Edinburgo, scritto in C, con accesso a procedure in C, diffuso tra le workstation. La versione Macintosh è ancora provvisoria, senza editore ma con la possibilità di accedere al ToolBox.

**MProlog Logic-Lab 2.1 Epsilon - DM 1998**

Ambiente di sviluppo diffuso in varie classi di calcolatori, segue la sintassi di Edinburgo e permette la programmazione modulare. Ha un sistema grafico proprio e compilatore. La versione per il Macintosh è ancora in preparazione.

**MacProlog 1.0a Logic Programming Associates  
- \$ 395**

Versione Macintosh del prodotto noto come Micro-Prolog in ambiente PC e come Sigma-Prolog nell'ambiente workstations. Permette la scelta tra la notazione di Edinburgo, la notazione stile LISP e una detta "simple notation". Implementazione al momento tra le più confortevoli, comprende anche due compiler (uno veloce per lo sviluppo e uno ottimizzato per la compilazione finale). Accesso al Toolbox, ma per ora non alla grafica!

**4.- Linguaggi a oggetti**

La programmazione a oggetti fa parte di molti sistemi LISP (e Logo) citati sopra; in questa categoria vengono descritti solo i due prodotti seguenti:

**Smalltalk-80 0.3 Apple Computer - \$ 50**

Versione provvisoria distribuita a scopo di studio, che realizza però una implementazione completa secondo Xerox-PARC. Il linguaggio ha uno stile di interfaccia grafico con l'utente che non corrisponde alle norme Macintosh.

**Neon 1.5 Kriya Systems - \$ 245**

Linguaggio a oggetti basato sia su Smalltalk (per i concetti) sia su FORTH (come stile di linguaggio). Il Toolbox è a disposizione come sistema di classi; ogni programma è di fatto un ampliamento del linguaggio. Assembler a disposizione.

**ATTIVITA' LOGO NELLE CLASSI SPECIALI DEL CANTONE TICINO  
GABRIELE SCASCIGHINI - Ispettorato delle Scuole Speciali, Lugano**

**GE SE LN RO VI DI LI MO**

**Il presente articolo affronterà i seguenti aspetti:**

1) Dapprima una breve discussione delle motivazioni e dei problemi di carattere fondamentale del nostro lavoro nell'ambito della microinformatica accompagnata da una sommaria descrizione delle attività svolte da alcuni insegnanti delle scuole Speciali del Cantone Ticino.

2) In seguito sottolineeremo l'importanza di riflettere piuttosto che sul linguaggio Logo, (le sue caratteristiche grammaticali e sintattiche) sul come esso viene usato nella pratica educativa corrente. Questo cambiamento di punto di vista ci permetterà una superficiale descrizione delle difficoltà di vario genere (logico-matematiche, spazio-temporali ecc.) incontrate dai nostri allievi e dei risultati della nostra attività. La lettura di questo testo presume qualche, (anche sommaria), informazione relativa ai linguaggi funzionali (Logo, Lisp) e alla geometria della tartaruga.

**Introduzione**

Perchè utilizzare un ambiente micro-informativo nell'ambito dell'educazione specializzata? Pos-

siamo sintetizzare le ragioni del nostro interesse in tre punti.

1) Gli elaboratori esercitano un fascino particolare. Questa capacità di indurre reazioni positive per attività "intellettuali" è una caratteristica importante, soprattutto se pensiamo alle difficoltà di motivazione che spesso insegnanti o terapisti che lavorano con soggetti ipodotati rilevano nei loro allievi.

2) La passività o il "vuoto relativo" dell'ordinatore, l'inerzia dell'oggetto elaboratore implica, per contrapposizione e valorizza indirettamente l'attività del soggetto. La microinformatica può favorire un radicale capovolgimento del ruolo; chi non conosce diventa improvvisamente un attore indispensabile. I nostri soggetti, sovente dipendenti, incapaci di sentirsi autori di soluzioni creative si trasformano, provano le loro possibilità di pensiero davanti ad una macchina passiva, dipendente dal programma del soggetto, alla quale è necessario insegnare.

3) Infine, la capacità di memorizzazione e di ripetizione della macchina può essere di aiuto nel processo di apprendimento.

Queste ipotesi, certamente ancora molto generiche ( il problema per tutti resta appunto quello dell'approfondimento con metodo scientifico, chiaren-

do limiti, precisando condizioni ecc. ) ci sono comunque sembrate appassionanti per dirigere il nostro interesse su questo tema. Le ragioni più profonde si iscrivono anche nell'appassionante dibattito sulla modificabilità, educabilità dell'intelligenza. In quale misura e in quali condizioni un ambiente informatico può favorire una modifica, beninteso stabile, dell'intelligenza di un soggetto con ritardo mentale? Questa complessa domanda, è centrale per tutti quei ricercatori interessati ai problemi dello sviluppo del pensiero. I limiti descrittivi della nostra riflessione non ci consentono tuttavia di entrare nel merito di questo appassionante dibattito. Citeremo un punto di riferimento obbligato, a proposito di questo tema, Jean Paour, collaboratore al Laboratorio di "Psychologie" de l' "enfant" dell'université di Alex-en-Provence. Questo ricercatore, nei suoi studi evidenzia le prove empiriche sulla plasticità mentale del soggetto con ritardo evolutivo di grado medio o leggero. Citiamo da un suo recente articolo "è possibile provocare progressi cognitivi importanti e stabili a lungo termine corrispondenti ad un accesso allo stadio del pensiero operatorio concreto", più oltre "l'osservazione longitudinale ci ha portato a considerare che la mancanza di motivazioni appropriate e l'inadeguatezza delle strategie esploratorie e d'organizzazione dell'ambiente costituiscono un determinante cognitivo importante della genesi e del mantenimento del ritardo mentale." Paour sta ora lavorando cercando di determinare in quale precise condizioni un ambiente informatico può favorire in soggetti insufficienti mentali questo genere di dinamismo evolutivo.

### **Il progetto Logo nelle scuole specialistiche ticinesi**

A partire dal settembre 1984 un gruppo di insegnanti delle scuole speciali cantonali ticinesi inizia la propria formazione nell'ambito dell'informatica frequentando alcuni corsi di carattere introduttivo e di programmazione con il linguaggio Logo. Nello stesso tempo ognuno di questi insegnanti ha la possibilità concreta di sperimentare con i propri allievi quanto apprende nel corso di questa formazione. Una decina di classi (circa 50 allievi) con i loro rispettivi insegnanti sono coinvolti in questo progetto. Fondamentalmente vengono privilegiate le attività in ambiente Logo: alcune classi hanno invece compiuto passi interessanti nella direzione di imparare l'uso di programmi di elaborazione di testi che tra l'altro ora serve anche per la preparazione di una rivista gestita dagli stessi allievi di una classe speciale, "La gazzetta computerizzata". Si tratta di una rivista che raccoglie il frutto del lavoro di programmazione di tutti gli allievi partecipanti a questo progetto. La rivista riporta programmi, realizzazioni grafiche con la geometria della tartaruga, problemi aperti, suggerimenti sul

come si potrebbe migliorare un programma ecc... Un vero e proprio giornalino sulla falsariga delle raccolte di testi liberi della scuola frenetiana!

### **Dal Logo al come lo si utilizza**

#### **Il fenomeno dello "hacking"**

La tartaruga, entità fondamentale nella geometria Logo, diversamente dal punto della geometria cartesiana, è paragonabile ad un essere umano: ha una posizione e guarda in una certa direzione. I bambini possono servirsi della tartaruga, della sua immagine, come primo esempio di matematica, possono identificarsi in essa e ricorrere quindi alle loro precedenti esperienze psicomotorie per affrontare ed approfondire le loro conoscenze matematiche e geometriche. Tutto questo avviene nella misura in cui il bambino impara a parlare la lingua della tartaruga. Le prime esperienze avvengono quando il bambino impara a descrivere il suo modo di muoversi nello spazio e cerca l'opportuna traduzione nel linguaggio tartaruga. Di fronte ad un problema l'aiuto dell'insegnante consiste spesso, o dovrebbe consistere, soprattutto nel favorire questa identificazione (apprendimento sintonico). L'insegnante ricorre spesso di fronte alla domanda dell'allievo ad incitazioni euristiche del tipo: "gioca alla tartaruga!", "mettiti al suo posto!", ecc... In questo ordine di idee, con questo ottimismo, è stato detto e scritto molto. Poco invece è stato esplicitato nella direzione di capire o di superare le difficoltà emergenti nelle attività informatiche in ambiente Logo. Non tutto fila liscio; il bambino quando è immerso nelle attività di programmazione con il linguaggio Logo si trova di fronte a ostacoli, conflitti di natura varia. Tutti sono d'accordo nel sostenere il grande potenziale educativo del linguaggio Logo. Troppo pochi invece sono gli studi sulle applicazioni concrete, sul come, sul metodo quindi, si procede in ambiente educativo. In altre parole si tratta di cominciare a mettere in discussione non tanto il linguaggio Logo, non tanto ciò che ingegneri ed informatici hanno mirabilmente costruito ma l'applicazione concreta, i metodi con il quale si usa questo interessante strumento. Gli informatici sottolineano spesso che un programma complesso non può funzionare immediatamente, è sempre necessario passare attraverso un'importante fase di messa a punto, di scoperta dei "bugs". Si impara dall'errore, affermano d'altro canto i "logofili" entusiasti! Con questo spirito è necessario, a nostro avviso procedere nella valutazione di questa innovazione educativa. Si tratta, esattamente come quando cerchiamo il motivo per cui una procedura Logo non funziona come desideriamo, di scoprire i "bugs" nascosti nel nostro programma educativo. Molti applicatori del sistema Logo interpretano gli scritti di Papert (inse-

gnante al MIT, già collaboratore di Jean Piaget e creatore del linguaggio in collaborazione con altri eminenti informatici) sostanzialmente in termini intesi a scoraggiare ogni intervento diretto (a es. prescrizione di un compito da raggiungere, proposte di attività curriculari particolari ecc...). L'educatore crede che le acquisizioni delle cosiddette idee potenti implicite nel mondo Logo vengano spontaneamente scoperte. Osservazioni più precise mettono però in risalto come in condizioni di questo genere molti bambini tendono a marciare sul posto: un fenomeno che un ricercatore molto importante di questi problemi, Uri Leeron dell'Università di Haifa definisce con il termine di "Hacking". Poco spazio viene riservato alla riflessione e alla pianificazione. Molti invece i tentativi sovente infruttuosi e dispersivi per prove ed errori. In questo scenario i bambini smettono spesso l'attività di programmazione nel bel mezzo di un programma. L'errore non è preso in considerazione, le difficoltà semplicemente eliminate. Alcuni educatori assolvono parzialmente questo approccio metodologico ritenendolo importante in una prima fase di decondizionamento dell'allievo precedentemente confrontato con gravi insuccessi scolastici.

#### La curiosità spontanea ?

Il modello bambino proposto negli scritti di Papert sottintende una curiosità innata, impetuosa. Nel bambino sarebbe particolarmente potente la presenza del desiderio di conoscere la natura del "bugs" che l'elaboratore segnala spesso all'utente. L'osservazione diretta mette in luce un'altra realtà: parecchi bambini, se lasciati soli di fronte all'elaboratore si accontentano di provare il funzionamento delle procedure senza interessarsi di capirne il loro funzionamento. Il risultato spesso viene prediletto dai bambini... Apatia e superficialità connotano i risultati di questo stile educativo.

#### Un insegnamento più guidato ?

Se tutto quanto esposto precedentemente è vero diventa molto importante, una vera e propria sfida educativa come qualcuno l'ha definita, la ricerca di mezzi più attivi di aiuto ai bambini evitando, certamente, di sacrificare le attività più esploratorie.

Uri Leeron, seguendo questo tipo di ragionamento ha coniato una sigla particolarmente significativa: QPL. Quasi-Piagetian-Learning. Questo ricercatore propone, tra l'altro l'introduzione di guide di studio per facilitare, organizzare la ricerca del bambino e per stimolare la riflessione su contenuti precisi e delimitati. L'uso di una guida in uno stile educativo più direttivo mostra paradossalmente un significativo miglioramento dell'autonomia dell'allievo. L'adulto possiede un mezzo euristico supplementare; " hai

guardato sulla guida ? ". Proporre queste riflessioni non significa affatto la reintroduzione, il ritorno a stili verbosi o nozionistici ! Insegnare non vuole dire proporre conferenze ai bambini; dirigere, orientare l'apprendimento è ben diverso dal forzarlo o dal costringerlo.

### Analisi di qualche difficoltà pedagogica.

#### Girare, spostarsi a destra ...

Un problema incontrato all'inizio della nostra attività con i bambini consisteva nell'osservare la loro difficoltà di separare la nozione di spostamento e di rotazione. Molti bambini cercano con un comando del tipo destra 55 di raggiungere un punto situato a destra della posizione della tartaruga. Essi immaginano che il comando destra, ad esempio, rappresenti una sintesi fra un movimento continuo ed omogeneo ed una rotazione costante passo per passo. Questa rappresentazione, in effetti più vicina al linguaggio naturale ( nel movimento spontaneo, corporeo come pure quando svoltiamo con l'automobile o con la bicicletta in realtà non dissociamo questi due elementi ) crea molti conflitti cognitivi. Edoardo Marti, ricercatore all'università di Ginevra, analizzando questo problema in un suo recente articolo sostiene più in generale che queste difficoltà sono rivelatrici di un problema più importante: " le unità significative ", citiamo, " che il bambino utilizza per analizzare il movimento della tartaruga non sono quelle del linguaggio Logo. " Se è vero che il riferimento alle proprie esperienze corporee può essere di grande aiuto resta pure anche importante non dimenticare che il bambino si trova pur sempre di fronte ad un problema di comunicazione con un robot dotato di un sistema linguistico proprio e soprattutto diverso da quello del bambino anche se certe forme lessicali potrebbero trarre in inganno.

#### Referenziale della tartaruga e del bambino

Sempre osservando i ragazzi all'opera con i loro primi progetti grafici vengono alla luce frequenti confusioni fra quello che i piagetiani chiamano referenziale intrinseco della tartaruga e referenziale assoluto del bambino. Se la tartaruga, sullo schema è orientata verso il basso spesso il bambino trova difficoltà ad adattarsi alla sua direzione proprio perchè diversa ( opposta ) dall'orientamento del proprio corpo. Un conflitto in più, un motivo ,comunque per affrontare un problema nuovo.

#### Limiti nella costruzione di procedure

Anche a proposito della presunta " naturale facilità " della programmazione strutturata è necessario

sfatare qualche mito. La possibilità con il Logo di definire da parte dell'utente nuove procedure, di memorizzarle e di poterle riutilizzare in ogni momento rappresenta senza dubbio una interessante proprietà. Per sfruttare questa caratteristica ed uscire da uno stile di programmazione additivo e lineare (nel senso di continuamente e semplicemente giustapporre uno dopo l'altro i comandi della tartaruga senza intuire l'efficacia dell'uso di strutture più complesse) occorre per essere in grado tra l'altro, di poter scoprire quali relazioni esistono fra due sottoprocedure. Questo concetto, altrimenti chiamato "interfaccia" rappresenta un ostacolo, nella nostra attività anche con ragazzi di 12-15 anni con ritardo mentale leggero, difficilmente sormontabile. Le nostre osservazioni evidenziano un uso molto lineare delle primitive Logo e della loro concatenazione: i nostri ragazzi programmano passo per passo, non sono ancora in grado di mettere in relazione lo stato (cioè la posizione e l'orientamento) della tartaruga prima e dopo ogni sottoprocedura. Il concetto di interfaccia e quello di trasparenza di stato rappresentano un livello di astrazione superiore alle procedure stesse. In questo ordine di idee ci sembrano pure significative le osservazioni circa l'uso alquanto modesto di una procedura in un altro contesto oltre a quello per il quale è stata prodotta.

#### La ricerca dell'errore

Un elemento fondamentale della dinamica suscitata dal lavoro con il computer è la ricerca e la correzione dell'errore. Negli errori di battitura si assiste ad un'irritazione e insofferenza verso il computer. E' proprio attraverso l'errore di battitura che i bambini imparano ad analizzare ed isolare gli elementi del loro scritto.

Infatti in un primo tempo un risultato errato porta il bambino a credere che tutto il procedimento sia errato. Solo successivamente, ed è un passo importante, impara l'analisi. Gli errori di linguaggio sono più gestibili dal gruppo che lavora perchè risolvibili con discussioni di precisazione della comunicazione tra di loro. Alcuni errori assumono un carattere semantico; l'uso, ad esempio della primitiva I ( indietro ) con lo scopo di tornare globalmente al punto di partenza indica una " surgeneralizzazione " del significato della primitiva stessa. Altri sono invece errori di coordinazione, evidenziano le difficoltà del soggetto di coordinare azioni o insiemi di azioni. Non vengono, in questo tipo di errore, rispettate le relazioni fra le istruzioni, in particolare:

- le relazioni fra un'istruzione ed un'altra (errori " inter-instructions " )
- le relazioni fra sequenze di istruzioni altre (errori " inter-sequences " )

Queste difficoltà emergono chiaramente quando il soggetto, abbandonato il modo diretto di programmare ( un'istruzione alla volta con verifica immediata ) e cercando invece di raggruppare più istruzioni alla volta, cerca di fare il " debugging " del suo programma.

#### Progetto individuale e progetto collettivo

L'attività Logo svolta in gruppo se da un lato favorisce la cooperazione, la spiegazione fra compagni d'altro canto manifesta anche dei limiti:

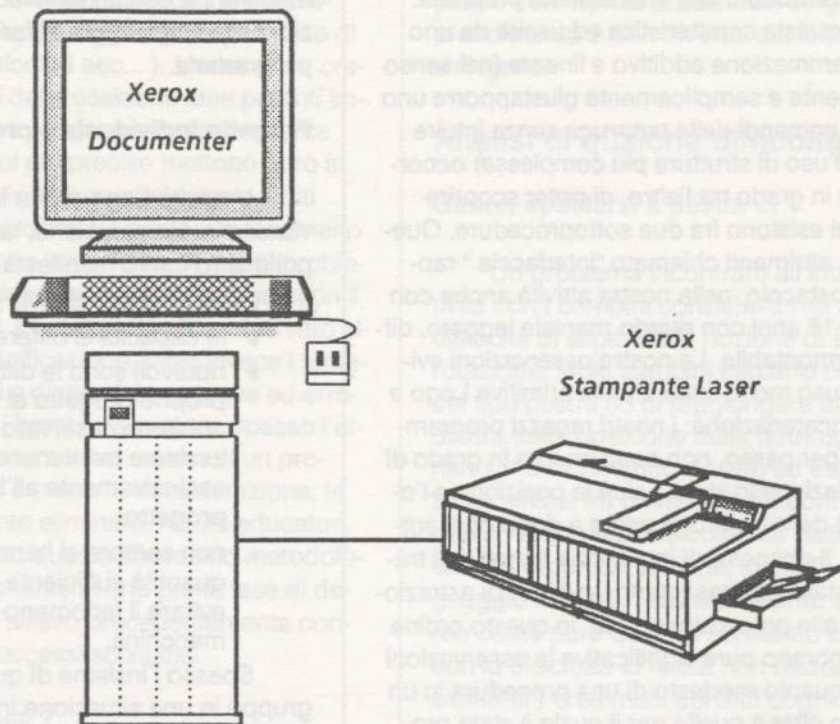
- gli interessi si diversificano rapidamente
- le capacità si differenziano
- notevoli sono le difficoltà di comunicare il proprio progetto al compagno. Sovente abbiamo osservato più allievi progredire assieme mantenendosi ognuno esclusivamente all'interno del proprio progetto.
- non sempre si hanno a disposizione una quantità sufficiente di elaboratori tale da evitare il fenomeno dell'accappararsi la macchina.

Spesso l'insieme di questi problemi porta il gruppo in una situazione in cui è molto difficile la gestione della frustrazione, la tolleranza dell'errore. In alcune nostre classi abbiamo però osservato situazioni e scelte diverse. L'attività individuale del ragazzo davanti allo schermo, in queste classi è stata abbandonata perchè il gruppo era ritenuto un importante mezzo per discutere, affrontare difficoltà, trovare errori. In queste classi c'è chi scrive, chi detta, chi coordina, chi fa finta di essere la tartaruga. Predomina un linguaggio scarso ed egocentrico, la comprensione del punto di vista dell'altro resta... una conquista. Frequente, in questi gruppi è l'uso della tartaruga da pavimento.

Queste alcune sommarie osservazioni di carattere psicopedagogico relative alle nostre attività con soggetti con handicap mentale di grado leggero. Il lettore attento si sarà accorto che le difficoltà non mancano e soprattutto avrà avuto un'ulteriore conferma della estrema complessità di scoprire o più semplicemente di descrivere qualsivoglia legame fra intelligenza umana e intelligenza artificiale

Si consiglia la lettura dei seguenti testi introduttivi: Seymour Papert, Mindstorms, Emme edizioni Abelson Harold, Logo for the Apple II, Mc Graw-Hill Abelson, Disessa, La geometria della tartaruga, Muzio editore.

# Xerox Documenter



Rank Xerox SA  
Via Zurigo, 5  
6900 Lugano  
Tel. : 091 / 22.73.21  
Tc. : 091 / 23.84.28

**TeamXerox**

## CEDOLA DI RISPOSTA

Se volete avere maggiori informazioni sull'attività dell'Associazione Ticinese Elaborazione Dati vi invitiamo a compilare il presente tagliando e a spedirlo a: ATED, casella postale 949, 6830 Chiasso.

NOME : .....

.....

.....

- Desidero ricevere informazioni sull'Associazione Ticinese Elaborazione Dati
- Desidero ricevere informazioni sul Gruppo Intelligenza Artificiale dell'ATED
- Desidero ricevere informazioni sul Bollettino GIA-ATED
- Desidero ricevere informazioni sulla Banca Dati GIA-ATED