

USO DEL COMPUTER NELLA DIDATTICA

(si riportano contributi della prof.ssa Bianca Maria Varisco, tratti o adattati dal testo "Le tecnologie didattiche", ed. Pensa Multimedia)

- **Cambiamenti nei contenuti e nella gestione dell'utente**
- **L'evoluzione del software didattico**
- **Dall'insegnamento all'apprendimento**

Cambiamenti nei contenuti e nella gestione dell'utente

La storia dei calcolatori ci indica come l'evoluzione della loro architettura, della potenza di elaborazione dei loro processori, della capacità delle loro memorie, abbia fatto sì che si sia passati dall'esclusivo uso matematico dei sistemi informatici (considerati macchine logico-matematiche) ad una loro utilizzazione sempre più flessibile e generalizzata (*general purpose*: g.p.). Tale uso si è concretizzato non solo nella possibilità di fruire, con la stessa macchina, sulla base di un unico sistema operativo, di più programmi, a seconda delle esigenze del momento, ma anche di gestire programmi di natura estremamente eterogenea, che al primitivo ed esclusivo trattamento di numeri e testi, ha via via aggiunto la possibilità di elaborare, in maniera sempre più sofisticata ma anche facile, ovvero "amichevole" per l'utente, contenuti diversi e multimediali. Ciò permette l'estensione dell'operatività dei sistemi software a domini di conoscenza inizialmente inimmaginabili, come l'area umanistica (storia, archeologia, economia, sociologia, giurisprudenza, letteratura, lingue), quella artistica (musica, grafica, architettura e design in genere, animazione ecc.), oltre quella scientifica (fisica, biologia, astronomia, medicina, ecc.)

Le funzioni operabili non solo su numeri e testi, ma anche su suoni, su immagini fisse e in movimento, nonché sul montaggio digitale di immagini e suoni, sono notevolmente aumentate, diventando sempre più sofisticate ed offrendo prodotti finali di alta qualità.

Molte delle funzioni (apri un documento, crea un documento nuovo, chiudi, salva, taglia, copia, incolla, inserisci parole, testi, suoni, immagini) sono inoltre diventate "trasversali" ai programmi g.p., esistendo una parziale omogeneità nelle interfacce utente (cioè nel modo di presentarsi) adottate dai vari sistemi applicativi di tale genere. Questo fatto permette a ciascuno, una volta familiarizzato con un sistema software, di passare con facilità ad esplorare ed impraticarsi delle funzioni di applicativi di altro genere. La natura grafica delle interfacce, fortemente intuitiva (in gergo "amichevole"), molte volte corredata da help ad avvio automatico su opzione dell'utente (ad esempio rappresentato da nuvolette a mo' fumetto che si aprono sfiorando con il mouse le icone per l'attivazione delle funzioni) facilita l'interazione con la macchina e incentiva la libera iniziativa dell'utente nel cercare e trovare le funzioni e le strategie migliori alla realizzazione del documento desiderato.

La multiutilizzabilità è un'ulteriore caratteristica di molti applicativi che prevedono, al loro interno, funzioni differenziate a seconda dell'uso che se ne vuole fare, del livello di competenza od expertise degli utenti (livello avanzato o iniziale -vedi in quest'ultimo caso la possibilità di attivare gli help automatici sopra menzionati), del loro stato operativo (uso professionale o didattico, utenza normodotata o disabile).

L'evoluzione del software didattico

.....

a) Il computer come tutor

Seguendo una cronologia storica, in continuo sviluppo, egli parlò innanzitutto di computer usato come tutor, cioè insegnante, tutore, riferendosi all'utilizzazione di software didattico che potremmo raggruppare tra gli applicativi dedicati. Tale categoria di programmi include quelli che negli Stati Uniti vengono denominati CAI (Computer Assisted Instruction) e CAL (Computer Assisted Learning).

I primi, CAI, che nel campo della formazione sono normalmente identificati con l'acronimo CBT (Computer Based training), sono programmi orientati decisamente a supportare o sostituire

(nel caso dell'autoistruzione) l'insegnante in alcune sue tradizionali funzioni. Tutti i seguenti programmi presentano una struttura di presentazione dei contenuti (lineare o più o meno ramificata) sempre e comunque preordinata dal progettista ed implementata dallo sviluppatore in modo definitivo:

- **Dimostrazioni**, di teoremi, fenomeni fisici, esecuzione automatica di compiti pratici. Esempi di demo sono quei programmi che a volte le case produttrici di software offrono ai commercianti o a probabili acquirenti per dare loro una "dimostrazione" delle funzioni specifiche di un nuovo applicativo o della sua ultima release (versione). L'esecuzione di questo tipo di programma è completamente o parzialmente automatica, ciò vuol dire che all'utente viene lasciato un minimo spazio di iniziativa, normalmente quello di selezionare la funzione che desidera veder automaticamente applicata, attraverso l'attivazione di tutte le sue possibili opzioni.
- **Esercitazioni** o testing, cioè somministrazione di quesiti (normalmente a risposta multipla ma anche del tipo vero-falso) su argomenti specifici, il cui livello può essere predefinito dall'insegnante, determinato dal sistema sulla base delle risposte offerte dallo studente ad una serie di item iniziali, o selezionato direttamente dall'allievo. Nell'esercitazione il sistema può offrire all'utente la spiegazione dell'errore commesso o la motivazione della giustezza della risposta data, offrendo la possibilità, in caso di errata risposta, di ritentare sino al raggiungimento del successo, mentre alla fine della sessione può fornire immediatamente il risultato del punteggio ottenuto e il livello di competenza dimostrata; quest'ultima funzione è inclusa nel testing che rappresenta un'automazione delle procedure di somministrazione ed elaborazione di test.
- **Tutoriali**. Questa categoria di CAI è l'esemplificazione applicativa dell'Istruzione Programmata (IP) proposta negli anni '50 dallo psicologo comportamentista Frederic Skinner (come quell'insieme di tecniche e metodi che cercano di offrire regole atte a determinare con precisione la struttura, l'ordine e i tempi di presentazione dei contenuti, in modo da garantire agli allievi un apprendimento qualitativamente ottimale) e del suo corollario, il Mastery Learning (apprendimento per la padronanza, teorizzato da Block come una metodologia intesa a far giungere tutti gli studenti a padroneggiare adeguatamente conoscenze e abilità fondamentali, facendo leva sull'attitudine, la perseveranza e la qualità dell'insegnamento). Nei tutoriali la somministrazione, è rigorosamente individuale, anzi l'individualizzazione dell'insegnamento è un obiettivo primario di questo tipo di programma a matrice comportamentista (fondato sul trinomio skinneriano s-r-r ovvero stimolo-risposta-rinforzo), che adotta sempre modalità di verifica delle informazioni proposte e acquisite che comprendono tendenzialmente domande a risposta multipla (secondo la logica espressa da Norman Crowder), prevedendo, per ciascuna risposta, un intervento (ulteriore informazione-verifica) differenziato. I programmi per computer di tipo tutoriale vengono sovente inclusi nel "pacchetto" software che accompagna il sistema di base all'acquisto di un computer. Un tutorial presenta delle sequenze, ormai multimediali, di informazioni su parti del sistema e sul loro funzionamento (quest'ultime ora inclusive di brevi animazioni), invitandoci, alla fine dell'unità informativa, ad esercitarci per dimostrare, con il nostro comportamento, l'avvenuta comprensione e apprendimento di quanto precedentemente illustrato e dimostrato dal sistema (ad esempio le tecniche di puntamento, selezione e trascinamento del mouse), ottenendone un feedback positivo o negativo. Anche in questo caso, il livello di istruzione può venir selezionato con le modalità proprie delle esercitazioni.

Sebbene molti autori ritengano CAI e CAL sinonimi, noi preferiamo identificare i secondi con quei programmi di tipo "tutoriale" più centrati sull'apprendimento che sull'insegnamento. Infatti, mentre in tutti i sistemi CAI precedentemente illustrati l'iniziativa è sempre gestita dal sistema, che individualizza gli interventi in base al feedback ricevuto dall'allievo e al ricorso ad una molteplicità di itinerari predeterminati, nei CAL, sebbene gli scenari siano predefiniti, viene lasciato all'utente un sufficiente spazio di iniziativa, all'interno del quale potersi autonomamente orientare, prendere decisioni, seguire addirittura percorsi autodeterminati: in questo caso si parla di "personalizzazione" dei percorsi. I tradizionali programmi CAL (giochi, e simulazioni) sono stati etichettati in passato come "software creativo", per distinguerli da quelli CAI e sottolineare nel loro uso l'emergenza dell'iniziativa.

tiva dello studente. Inoltre è venuto in essi a cadere il concetto stesso di “programma didattico” considerato come strumento ben definito, adattivo ma chiuso ed autogestito, sostituito da quello più ampio e flessibile di "ambiente d'apprendimento", aperto a ricevere l'intervento attivo e partecipante dello studente:

- I prototipi di questa specie sono i **game didattici**, nei quali la presa di decisione, a volte obbligatoriamente immediata, altre volte sufficientemente ponderata, è un obbligo, per il giocatore. La motivazione attivata da questo tipo di software ha fatto sì che numerosi autori si siano interessati del rapporto gioco-motivazione, al fine di creare ambienti didattici ludiformi, che conservano le caratteristiche altamente accattivanti delle attività ludiche. Malone e Lepper hanno ad esempio individuato sette aspetti caratterizzanti la motivazione intrinseca tipica dei videogame: quattro individuali (sfida, curiosità -perceptiva e cognitiva-, controllo della situazione, fantasia del contesto d'azione) e tre interpersonali (competizione, cooperazione, riconoscimento dell'attività svolta). Tale tassonomia dovrebbe guidare gli insegnanti nell'allestimento di ambienti d'apprendimento (più o meno tecnologici) intrinsecamente motivanti, game didattici inclusi.
- A questi ultimi (che non devono perdere la motivazione intrinseca propria dell'attività ludica per essere didatticamente efficaci) si aggiungono le **simulazioni**, che possono avere funzioni addestrative (pensiamo ai piloti o ai macchinisti dei treni che vengono addestrati il più delle volte con simulazioni), o essere utilizzati per l'analisi e lo studio dei fenomeni fisici, la formulazione di ipotesi circa le variabili che li regolano e i loro reciproci rapporti, nonché per la verifica, simulata, di tali ipotesi.

b) Il computer come tool

E' questo il campo degli **applicativi general purpose** o g.p. (word processor, data base management system, fogli elettronici, editor grafici, musicali ecc.). In questo settore, in campo didattico, a tutti i livelli scolari e formativi, sicuramente gli editor testo e quelli grafici sono i più sfruttati nella quotidiana attività didattica che prevede l'utilizzazione degli strumenti informatici. Sono sfruttati non solo all'interno di attività specifiche di scrittura o disegno, ma in quelle più trasversali rappresentate dall'editare testi per descrivere esperienze vissute, relazionare su ricerche condotte, produrre newsletters (bollettini informativi), scolastici o aziendali, anche utilizzando i tools offerti dell'editoria elettronica. Proprio per l'ampia diffusione degli editor grafici e testuali a tutti i livelli scolari (dalla scuola dell'infanzia a quella primaria, secondaria e superiore) e in tutti i campi della formazione, ne sono state prodotte numerose versioni comprendenti, oltre le normali funzioni di base proprie dei programmi professionali, una serie di funzioni didattiche specifiche. In questo caso si parla di editor "vestiti", cioè attrezzati ad un uso educativo: si va dalle "tastiere parlanti" rivolte alla scuola dell'infanzia e alle prime classi della scuola primaria, a strumenti che avviano a comporre testi secondo, intenti, stili, figure retoriche differenziate, a riassumere brani, a progettare e revisionare prodotti testuali di vario genere.

Ultima generazione di editor sono quelli che permettono l'**authoring multi-ipermediale**, e che classifichiamo sotto questa categoria perché non prevedono, da parte dell'utente, un'effettiva programmazione delle funzioni assolvibili dai vari oggetti inclusi (testi, suoni, immagini) e delle loro concatenazioni, ma, attraverso un'interfaccia amichevole (facilitativa) di tipo grafico (con icone e menu), gli permettono, selezionato un oggetto ipermediale precedentemente creato o importato attraverso un'amichevole interfaccia grafica, di limitarsi alla pura scelta della funzione che su di esso vuole attivare: sarà il sistema che automaticamente "incollerà" il programma o script (predisposto in maniera adattabile dallo sviluppatore del sistema) ad essa corrispondente all'oggetto selezionato. Tali strumenti sono sempre più diffusi sul mercato, e vengono rivolti ai principianti o ai livelli di scuola dell'infanzia e primaria. Esiste inoltre una serie di programmi di presentazione, di tipo multi-ipermediale, che permettono di preparare pagine di testo, corredate da immagini statiche, da animazioni e da suoni, che implicano un minimo lavoro di "impostazione" delle pagine, servendosi di semplici comandi di formattazione: anche in questo caso non si tratta di vera e propria programmazione, ma di un'attività compresa tra l'editing e lo sviluppo.

Un'altra categoria che, secondo noi va aggiunta ad una concezione di computer inteso come strumento di lavoro ormai sempre più insostituibile, è rappresentata da numerose **funzioni telematiche**. Dagli strumenti per la comunicazione a distanza attraverso modalità in tempo differito o reale -asincrona o sincrona (posta elettronica, forum, BBS -Bulletin Board System-, liste di interesse, conferencing e videoconferencing) alla ricerca di informazioni su base dati remote o sui milioni di siti compresi nella rete delle reti, internet, magari servendosi di utili motori di ricerca per la loro selezione. A proposito di Internet, ricordiamo che è possibile, in esso, non solo ricercare e fruire informazioni, ma anche diventarne erogatori. In questo caso, dopo aver acquisito uno spazio su un server, si provvederà al lavoro di editing che, oltre alla stesura dei puri contenuti (testuali e grafici), prevede, come nei sistemi di presentazione, l'impostazione delle pagine e la definizione delle parole, frasi, immagini "calde" (cliccando sulle quali si passerà ad altre pagine con procedura ipertestuale), stabilendo dei links (legami). Anche in questo caso ci si servirà di comandi di formattazione. Oggi ci sono due tipi di "linguaggi" di formattazione di pagine Internet: HTML o HyperText Markup Language e il più recente VRML o Virtual Reality Modelling Language.

c) Il computer come tutee

Il computer viene considerato come qualcuno che esegue "a cottimo" quello che gli si dice di fare, senza dare altra risposta autonoma se non il segnale della possibilità o impossibilità di eseguire il compito assegnatogli, offrendo l'eventuale segnale della presenza di un errore sintattico nell'istruzione impartita.

In questo caso si fa esplicito riferimento ai sistemi di sviluppo, ed in particolare ai classici **linguaggi di programmazione**. Tali linguaggi (di alto livello, prossimi all'utente ovvero rivolti al problema da risolvere) sono formati, grosso modo da due componenti: un'interfaccia utente (sempre presente in ogni tipo di programma e più o meno amichevole) ed un traduttore, cioè un programma che traduce lessico e sintassi del linguaggio di programmazione usato, ovvero i suoi comandi, le sue funzioni ed gli altri suoi segni convenzionali (comprese eventuali parentesi quadre, virgolette, punti e virgola ecc.), in un codice comprensibile alla macchina, che come abbiamo già detto, essendo digitale, è in grado di riconoscere solo due stati o situazioni: la presenza (on) o l'assenza (off) di un segnale, rappresentabili dai simboli 1 e 0, sui quali svolgere, attraverso i suoi circuiti, solo operazioni logiche, le tre operazioni booleane di *and*, *or* e *not*.

L'unico codice comprensibile all'elaboratore è perciò un linguaggio di tipo binario (il linguaggio macchina) composto da istruzioni formate ciascuna da stringhe (serie) di 1 e 0. La rappresentazione finale di tutte le operazioni che saranno effettivamente eseguite dalla macchina (istruzioni macchina) deve impiegare tale linguaggio; per questo motivo i traduttori le codificano in modo da essere immagazzinate ed elaborate come stringhe di bit (Binary digiT), unità minima di informazione. Dati e testi, che verranno eventualmente immessi nel programma o digitati da tastiera al suo avvio, verranno codificati esclusivamente attraverso il codice ASCII (American Standard Code for Information Interchange) che trasforma ciascun carattere della tastiera in stringhe univoche formate da 8 cifre binarie.

I traduttori possono essere di due tipi: compilatori ed interpreti. Mentre i primi traducono l'intero programma sorgente, scritto in linguaggio di alto livello, quello scelto ed usato dall'utente (ad esempio il linguaggio Pascal) in programma oggetto (scritto in linguaggio macchina) per poi permettere di eseguirlo interamente e verificarne la funzionalità, i secondi traducono istruzione per istruzione dando feedback continui sulla loro correttezza. Ogni linguaggio di programmazione adotta, normalmente, una modalità di traduzione. Va precisato inoltre che i traduttori sono presenti non solo nei linguaggi di programmazione, ma anche in tutti i programmi o file eseguibili (applicativi g.p. e dedicati) mentre sono assenti nei semplici documenti (file dati), che richiedono, per l'appunto, l'apertura dell'applicativo g.p. che li ha generati, per essere letti e manipolati.

L'unico linguaggio pensato e sviluppato per la didattica (o meglio per costruire, suo tramite, un ambiente per apprendere in maniera attiva e costruttiva) LOGO, offre, allo studente, la duplice possibilità: quella di impartire comando dopo comando verificando, ad ogni step, la funzionalità dell'o-

perazione eseguita attraverso l'output grafico (nel caso del micromondo della geometria della Tartaruga) di ogni singola istruzione impartita, ed eventualmente di correggerla con un successivo comando di "aggiustamento", e la possibilità, una volta stabilita l'efficacia dell'intera sequenza di comandi, di passare in ambiente editor, dove compilare, in maniera il più possibile compatta e strutturata, l'intero programma (magari usando variabili al posto di dati) per poi mandarlo in esecuzione digitando semplicemente il suo nome ed, eventualmente, il valore che si vuole assegnare, di volta in volta, alle variabili incluse nel programma e dichiarate nel suo titolo (costruendo, ad esempio più quadrati dai lati di lunghezza diversa).

In questa sede riteniamo pedagogicamente importante sottolineare qual è stata la filosofia di fondo che ha spinto Papert (inventore del LOGO) a proporre la programmazione informatica come strumento per apprendere, condividendo l'idea che tale filosofia abbia contribuito, in maniera determinante, a rivoluzionare il modo di concepire l'uso delle tecnologie informatiche nell'apprendimento e nella formazione. L'obiettivo di Papert non è mai stato quello di formare dei "programmatori" più o meno in erba (pensiamo che con LOGO possono lavorare, con tastiere adeguatamente predisposte, anche bambini della scuola dell'infanzia, ma anche, nel trattamento delle liste e attraverso applicazioni complesse dei suoi vari ambienti, studenti della scuola secondaria) ma quello di utilizzare il computer e la programmazione come uno strumento potente per concepire ed esprimere progetti personali, carichi di significato, con finalità diverse a seconda delle versioni via via elaborate dal linguaggio di base: progetti essenzialmente grafici, pagine di testi scritti e disegnati, progetti di robotica, animazioni grafiche sino a prodotti ipermediali che usano come linguaggio di sviluppo LOGO.

La *potenza*, concetto che in Papert riassume tale idea, è uno dei tre pilastri di quello che lui definisce "apprendimento sintonico" o significativo per il soggetto. L'altro principio è quello della *continuità*, continuità tra quella che è la conoscenza e l'esperienza più o meno ingenua elaborata dagli studenti intorno ad un dominio di conoscenza, ed il progetto che andranno a sviluppare in quel campo: la sua elaborazione potrà mettere in crisi le ipotesi, formulate sulla base delle preconoscenze, circa la soluzione da dare ai problemi emergenti nello sviluppo dello stesso progetto. Il terzo concetto è quello di *risonanza* culturale delle tematiche scelte, che, per noi europei significa soprattutto collegamento delle attività svolte con l'elaboratore a progetti di tipo disciplinari o, ancor meglio, multi-inter-trans-disciplinari, cioè rapporto tra uso delle tecnologie informatiche e programmazione curricolare.

LOGO, i suoi vari ambienti e le sue svariate versioni, sempre più multi-ipermediali, sono stati concepiti da Papert (e dai suoi collaboratori) come **ambienti per l'apprendimento**, una serie di "micromondi" dove concepire progetti e costruire saperi utili, condivisi, che si adeguano ai molteplici stili dei soggetti, permettendo la sinergica collaborazione delle loro reciproche complementarietà, un sapere pratico, incorporato in contesti di utilizzo. E ciò attraverso l'elaborazione di alcune idee geniali che, alla fine degli anni '70, rivoluzionarono l'uso del computer nella didattica, considerandolo uno strumento la cui gestione doveva essere completamente affidata all'iniziativa dell'utente, valorizzando l'autonomia operativa di quest'ultimo e la pura reattività (esecutiva) della macchina.

.....

Quello dei sistemi con interfacce grafiche facilitative, che sopra abbiamo classificato tra gli editor, che permettono di diventare "autori" di prodotti multi-ipermediali, senza far uso esplicito dell'attività programmatoria, ma servendosi di appositi bottoni e menu che selezionano programmi predisposti, i quali "automaticamente" si adattano agli oggetti selezionati (immagini, parole, ecc.), facendo in modo che, allo sfioramento, puntamento o clic del mouse su di essi, si odano suoni, si aprano finestre di testo, si salti da una pagina ad un'altra (non sequenziale) dell'ipertesto, non è l'unico ed esclusivo campo dei sistemi di authoring multi-ipermediale, anche se sicuramente il più recente e probabilmente quello che avrà più sviluppo nell'immediato futuro. Esiste infatti tutta un'altra categoria di **sistemi di sviluppo multi-ipermediali**, che permette di realizzare prodotti di alta qualità, anche in 3D, elaborando necessariamente programmi, o script, che definiscono e gestiscono le funzionalità degli oggetti creati. In questo caso l'attività di authoring non è immediata (come nel ca-

so dei sistemi con interfacce facilitative) ma implica un periodo più o meno lungo di familiarizzazione con il sistema e l'acquisizione di un vero e proprio linguaggio di programmazione (normalmente del tipo ad oggetti).

Dall'insegnamento all'apprendimento

A differenza dei programmi centrati sull'insegnamento, gli ambienti via via centrati sull'apprendimento, tendono a distribuire la responsabilità dell'iniziativa tra i protagonisti dell'interazione - software e allievo- (vedi videogame, simulativi e ambienti di esplorazione navigazione ipermediale da un lato, applicativi g.p. dall'altro) o a lasciarla completamente all'utente (vedi sistemi di sviluppo).

All'approccio *tecnocentrico* (di natura deterministica e tecnicistica) si è andato così, via via, sostituendosi quello *antropocentrico*, attraverso una fase di informatizzazione della didattica che ha visto l'ingresso e il radicamento degli **applicativi g.p. nella quotidiana pratica didattica** di molte scuole e istituti di formazione, sistemi software che sono stati paragonati a "protesi umane" le quali, sostituendosi all'uomo nello svolgimento di funzioni di basso livello, potenziano, sviluppando e a volte trasformando profondamente, quelle di alto livello, anche attraverso l'interiorizzazione, da parte dell'utente, di strategie e funzioni tipiche di certi programmi. Un esempio eclatante è il cambiamento delle modalità di scrittura in un utente esperto nell'uso di un word processor, rispetto alle modalità prima praticate dallo stesso attraverso carta e penna: la concentrazione è tutta rivolta alla generazione e allo sviluppo delle idee, che si andranno via via strutturando prima nella mente, e poi sullo schermo, attraverso una serie di funzioni informatiche divenute, per l'utente, utili strategie di pensiero (copia-taglia-incolla).

Un altro cambiamento notevole, riscontrato nell'allestimento di strumenti o ambienti informatici per l'apprendimento-insegnamento, è consistito nell'aver spostato il focus dell'attenzione da un rapporto uno a uno tra macchina ed utente, ad un rapporto uno a più (normalmente piccoli gruppi di 2, 4 studenti) o (più raramente) di uno a molti. Le nuove tecnologie telematiche permettono inoltre l'interazioni a distanza (mediata da computer, modem e telefono) non solo di due individui remoti, ma anche di uno a molti e di molti a molti, con distribuzione anche planetaria dei soggetti in interazione, e attraverso modalità di comunicazione in rete sincrona o asincrona. Quella della **comunicazione telematica** è diventata la seconda generazione della prospettiva antropocentrica.

Questo cambiamento ha favorito sempre più le attività di studio e di ricerca cooperativa e collaborativa (si parla, di attività didattiche organizzate in comunità o circoli d'apprendimento che fanno spesso uso di tecnologie telematiche), e contribuirà sempre più a mutare anche la collocazione e **distribuzione logistica dei computer** all'interno delle scuole o degli istituti di formazione: si ridurranno i laboratori centralizzati, con macchine fisse a disposizione frontale, e sarà favorito l'allestimento di laboratori con macchine mobili e di piccoli ambienti informatici, con circa tre-quattro computer, comuni ad alcune aule o attigi, se non addirittura interni, ad esse.

Gli ambienti di **authoring ipermediali**, con interfacce più o meno facilitative, permettono inoltre di rappresentare (e comunicare) le plurime strade (processo) percorse e vissute nello sviluppo di una ricerca, mentre quelli multimediali offrono un'occasione in più alla possibilità di raccontare a 360 gradi (utilizzando una pluralità di codici e linguaggi) ciò che vogliamo descrivere o esprimere del nostro vissuto esteriore od interiore.

Il passaggio dalla visione *tecnocentrica* a quella *antropocentrica*, che ha favorito l'informatizzazione e la comunicazione, ha visto un'evoluzione e un'alternanza di "**paradigmi**" nella progettazione e implementazione di sistemi didattici, e nell'allestimento di ambienti d'apprendimento supportati dalle tecnologie, che possiamo reinterpretare nel modo seguente: dal paradigma dell'istruzione, tipico dei programmi CAI e ICAI, si è passati a quello dell'**espressione** (comprendente tutti gli

editor, inclusi quelli multimediali); della **costruzione**, attraverso sistemi di sviluppo (ambienti di programmazione tipo LOGO e relativi micromondi, nonché sistemi di authoring multi-ipermediale, da quelli a interfaccia facilitativa ai veri e propri sistemi di sviluppo); dell'**esplorazione** (degli scenari presenti nei videogiochi, nonché di quelli ipermediali esplorati attraverso la navigazione di ambienti a volte molto articolati); della **comunicazione** (attraverso l'*authoring* multi-ipermediale ma soprattutto le tecnologie telematiche). A tali paradigmi desideriamo aggiungere un ulteriore, quello della **ricerca** che, oltre ai sistemi di simulazione, o di costruzione di modelli, più o meno dinamici di fenomeni fisici, etologici ecc., dovrebbe far riferimento a tutti i tools telematici di IR (*Information retrieval*), di cui abbiamo parlato sopra, e, sempre a livello di ricerca informativa, ai prodotti o ambienti multi-ipermediali considerati come *tools* dedicati.