

La fabbrica dei robot

*Bambini che costruiscono
e programmano con un ambiente vivo*

■ **Augusto Chiocciariello, Stefania Manca, Luigi Sarti**
ITD-CNR, Genova
{augusto, manca, sarti}@itd.ge.cnr.it

INTRODUZIONE

L'importanza ed il ruolo fondamentale che gli oggetti materiali rivestono nel complesso percorso della costruzione della conoscenza sono stati sottolineati soprattutto dal Costruzionismo [Papert, 1980; Harel e Papert, 1991]. Questa teoria accetta la tesi di fondo del Costruttivismo, secondo cui le conoscenze non sono il riflesso del mondo esterno né la proiezione sulla realtà delle strutture innate della mente, ma derivano da costruzioni successive con costante elaborazione di strutture nuove [Piaget, 1975]. Il costruzionismo, inoltre, rivaluta il pensiero *concreto*, visto non come una versione "inferiore" del ragionamento astratto, ma posto sul suo stesso piano [Turkle e Papert, 1992]. La tradizionale impostazione operativo-esperienziale secondo cui si impara facendo [Dewey, 1910], attraverso una costante interazione dialettica con strumenti che "veicolano" gli oggetti della conoscenza, è stata, quindi, produttivamente integrata con la valenza imprescindibile che la *costruzione* di oggetti ha nell'apprendimento: la conoscenza è il risultato di un impegno attivo col mondo attraverso la creazione e manipolazione di artefatti (tangibili e no), siano essi castelli di sabbia, programmi di computer, costruzioni LEGO, composizioni musicali, ecc., che rivestano un particolare significato personale e che siano soprattutto oggetti su cui riflettere [Papert, 1993]¹. In sintonia con gli orientamenti prevalenti del costruttivismo sociale (cfr. ad esempio [Salomon, 1993; Pontecorvo et al, 1995]), un'altra componente sottolineata dal co-

struzionismo è quella dell'importanza della *negoiazione* nel mondo sociale come parte cruciale dello sviluppo cognitivo del bambino. L'apprendimento e l'intelligenza non sono, quindi, titolarità esclusiva del singolo individuo che apprende, ma emergono piuttosto dall'interazione sociale in cui gruppi di individui intrattengono rapporti di natura collaborativa finalizzati alla costruzione di conoscenze comuni e condivise [Resnick, 1996]. Questa dimensione sociale dell'apprendimento può ben essere rappresentata da quei contesti ludici ed educativi che vedono il bambino a stretto contatto con suoi pari e con soggetti più grandi di lui, in grado di fornire supporto e motivazione nell'affrontare compiti cognitivi nuovi e non precedentemente affrontati, che siano però alla sua portata (la cosiddetta "zona di sviluppo prossimale") [Vygotskij, 1978].

Tra le proposte di ambienti di apprendimento elaborate nel tempo alla luce delle indicazioni costruzioniste, va senz'altro ricordato il linguaggio Logo, finalizzato a incoraggiare lo sviluppo di forme di pensiero procedurale ed operativo attraverso l'uso di semplici programmi come blocchi di costruzioni per altri più complessi [Papert, 1980]. La programmazione al computer, vista come una delle forme più alte di strumenti attraverso cui "pensare sul pensare" (l'atteggiamento dell'epistemologo), assolve a questo compito in due modi importanti: "Primo, il computer permette, anzi obbliga, il bambino ad esternare le proprie aspettative intuitive. Quando l'intuizione è tradotta in un programma diventa più evi-

¹ Per un'analisi approfondita dei caratteri del costruzionismo papertiano soprattutto in relazione all'eredità piagetiana si veda [Ackermann, 2001].

² Tra le numerose evoluzioni di Logo che si sono succedute nel tempo ci interessa particolarmente ricordare LEGO/ Logo, un'estensione del Logo che consente di collegare sensori e motori LEGO ad un PC [Resnick e Ocko, 1991; Resnick, 1991]. LEGO/ Logo è stato il progenitore di una famiglia di linguaggi per il controllo di robot.

dente ed accessibile alla riflessione. Secondo, le idee computazionali possono essere adottate come materiali per rimodellare la conoscenza intuitiva” [Papert, 1980].

La programmazione è anche al centro delle proposte più recenti², i *robot* e i *mattoncini programmabili*, che occupano oggi uno spazio di particolare interesse. Modificando il modo di pensare al vivente e collocandosi di fatto al confine tra animato e inanimato, tra vivente e non vivente [Turkle, 1984, 1995], possono essere giocattoli con cui imparare nuovi modi di pensare [Resnick et al, 1996], che stimolano nuove riflessioni sul rapporto tra vita e tecnologia [Martin et al, 2000], tra la scienza e l'apparato strumentale di cui si serve per gli esperimenti [Resnick et al, 2000], tra le attività di progettazione di tipo robotico e i valori e l'identità [Bers e Urrea, 2000]. È anche grazie ad oggetti come questi che, sostengono i fautori del costruzionismo, concetti tradizionalmente considerati appannaggio di individui adulti, in grado di manipolare conoscenza simbolica ed astratta, possono diventare accessibili e comprensibili anche dai bambini.

I mattoncini programmabili, che storicamente si rifanno ai primi esperimenti di Papert con una concretizzazione delle idee esplorate con il Logo, sono approdati ad un prodotto commerciale, il LEGO® MINDSTORMS™ Robotic Invention System, un kit per realizzare robot e altre costruzioni cibernetiche³.

Questo articolo racconta alcune delle attività di ricerca condotte nell'ambito del progetto “Construction kits made of Atoms & Bits” (CAB)⁴ che, utilizzando il LEGO MindStorms, si è posto l'obiettivo di studiare le relazioni e gli atteggiamenti dei bambini della scuola dell'infanzia e delle prime classi della scuola elementare (da quattro a otto anni) nei confronti di oggetti in grado di esibire comportamenti autonomi e interattivi. In particolare CAB ha sperimentato una metodologia atta a favorire l'interazione tra bambini e computer attraverso l'uso di costruzioni cibernetiche (figura 1), con lo scopo di esplorare la dinamica dei processi di apprendimento che emergono e di ridisegnare il kit nell'ottica di adeguarlo a questa fascia d'età.

Ma perché il LEGO MindStorms? Che cosa accade quando un prodotto pensato per ragazzi dai dodici anni in su diventa una proposta educativa per bambini di cinque anni di età? Come legittimare una scelta pedagogica di questa natura? Quali cambia-



menti introdurre? Come intervenire sugli strumenti e sul contesto?

All'interno di CAB si è cercato di dare risposte a questi interrogativi, ad esempio attraverso l'ampliamento delle funzioni assolvibili dal robot, la ricerca di soluzioni destinate a ridurre la complessità meccanica degli artefatti e la definizione di un ambiente di programmazione fortemente orientato alle esigenze che emergono nei diversi contesti costruttivi.

Nel seguito dell'articolo descriveremo alcuni di questi aspetti, con particolare riguardo alle problematiche emerse nella progettazione dell'ambiente software e alle implicazioni che le scelte operate hanno avuto sulla progettualità dei bambini. La prossime sezioni illustrano, inoltre, la metodologia di ricerca adottata e le ragioni e modalità che hanno guidato la ridefinizione del kit. Le sezioni finali sono dedicate ad esporre alcune esperienze realizzate con i bambini e a prospettare possibili linee di sviluppo dell'approccio⁵.

PARTIRE DALL'ESISTENTE PER RIPROGETTARE IL NUOVO

«Cercare di fare un buon progetto significa soprattutto avere un'attesa già in qualche modo pertinente e preformata.»

[Malaguzzi, 1995: pag. 99]

Il progetto CAB ha privilegiato un approccio che ha inserito le attività dei bambini con i mattoncini programmabili nel contesto e prassi di lavoro quotidiani, affiancandosi spesso ad altri lavori già in corso. Questa scelta è stata fortemente ancorata alla

figura 1

Alcuni bambini giocano con un robot che disegna sul pavimento.

3

Per una ricostruzione storica del percorso evolutivo che ha condotto al LEGO MindStorms si veda [Martin et al, 2000].

4

CAB è stato un progetto finanziato dall'Unione Europea nell'ambito del programma ESPRIT, iniziativa i³-Experimental School Environments Programme, che ha visto la partecipazione del Comune di Reggio Emilia, della Jönköping University (Svezia), dell'ITD-CNR e della LEGO Dacta A/ S. Ulteriori informazioni possono essere rintracciate alla URL [http:// cab.itd.ge.cnr.it/](http://cab.itd.ge.cnr.it/)

5

Descrizioni complementari ed approfondite di alcuni aspetti qui trattati sinteticamente sono reperibili nella documentazione del progetto [Askildsen et al, 2001a; Askildsen et al, 2001b; CRE, 2001; Gustafsson e Lindh, 2001].

tradizione delle scuole dell'infanzia di Reggio Emilia, in cui l'attenzione è sempre rivolta alla predisposizione delle situazioni che facilitano il lavoro dei bambini, profondamente radicato in situazioni progettuali di ampio respiro (in termini di problematiche affrontate e di tempo da dedicare) e in cui l'adulto interviene il meno possibile ma sa ascoltare e documentare quello che accade cercando di mantenere alta la motivazione dei bambini [Malaguzzi, 1995]. L'ottica della *sperimentazione* è stata, quindi, da un lato, quella di guardare agli strumenti ludici e cognitivi come profondamente legati al contesto d'uso e alla cultura che danno loro significato. Dall'altro, quella di guardare all'apprendimento come a un evento sociale profondamente condiviso e contestualizzato in cui bambini ed adulti lavorano assieme e in cui i bambini ricevono sostegno e supporto (*scaffolding*) dagli adulti nel loro lavoro di esplorazione [Wood et al, 1976].

I requisiti per la riprogettazione del kit sono stati elicitati dai bambini attraverso il ruolo di mediazione degli adulti⁶. Il dialogo tra insegnanti, atelieristi, pedagogisti e progettisti hardware e software è stato ancorato alla *documentazione* del lavoro dei bambini che veniva progressivamente prodotta dalle scuole coinvolte nel progetto. Ogni attività è stata adeguatamente monitorata e documentata in una varietà di formati (testi, immagini, filmati, ecc.) dagli insegnanti impegnati nella quotidianità del loro lavoro⁷. Attraverso le considerazioni e le interpretazioni effettuate dagli adulti sulla base delle problematiche emerse nel lavoro coi bambini, è stato possibile prendere in seria considerazione le "teorie", seppur provvisorie,

che i bambini elaboravano sulla base dei progetti realizzati con il LEGO MindStorms e con le successive versioni modificate del kit.

RIPROGETTARE IL KIT

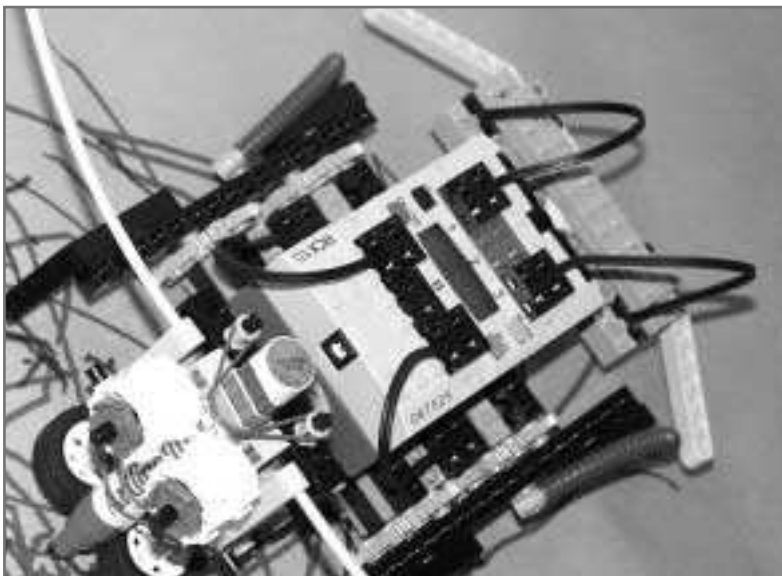
Una scatola di costruzioni, in funzione dei componenti offerti, privilegia certe tipologie di attività a scapito di altre. Il LEGO MindStorms è progettato per incoraggiare la costruzione di "Veicoli", robot mobili che si muovono e interagiscono con l'ambiente. Per favorire lo sviluppo di altri scenari d'uso sono state proposte ulteriori tipologie quali le "Costruzioni Animate" (che pur non esplorando l'ambiente possono prevedere parti che si muovono), i "Puppazzi Cibernetici" (una specializzazione delle costruzioni animate ispirate ai Furby⁸, ma smontabili e modificabili), i "Wearables" (dispositivi indossabili, quali distintivi interattivi)⁹.

Per i Veicoli e le Costruzioni Animate, che prevedono la presenza di parti meccaniche in movimento, abbiamo dovuto affrontare soprattutto la complessità di assemblaggio: il kit commerciale usa il LEGO Technic, un sistema che offre notevoli potenzialità costruttive ma risulta impegnativo anche per gli adulti. La soluzione adottata, privilegiando l'immediatezza d'uso a scapito della creatività costruttiva, è stata quella di predisporre moduli meccanici pronti per l'uso [vedi figura 2], quali uno chassis di veicolo con tre sistemi di locomozione modulari (ruote, cingoli, "zampe"), varie strutture di montaggio di sensori e attuatori¹⁰. Moduli per costruzioni animate che realizzano movimenti particolari includono un nastro trasportatore, una torretta rotante per orientare il mattoncino programmabile, un braccio meccanico, ecc.

L'uso di tipologie di costruzione diverse dai Veicoli consente l'esplorazione di comportamenti reattivi che si inquadrano, ad esempio, nel contesto della narrazione di storie, dove abbiamo osservato che i bambini tendono a combinare diversi materiali costruttivi (LEGO, carta, stoffa, creta). In questi scenari la riduzione di complessità meccanica permette ai bambini di esprimere la propria creatività e di affrontare da subito la definizione dei comportamenti. Ne è conseguita la necessità di affrontare la progettazione di nuove componenti per ampliare le tipologie di costruzioni possibili: un registratore di suoni, una catena di luci, un sensore di piegamento, un sensore di suono e un trasmettitore a raggi infrarossi. Nel se-

figura 2

Un veicolo, realizzato con moduli pre-assemblati, che reagisce agli ostacoli e lascia traccia del proprio percorso.



guito descriviamo in dettaglio uno di questi componenti.

Ai bambini piace aggiungere voci e suoni registrati alle loro costruzioni realizzate al computer. È emersa pertanto la necessità di un piccolo registratore digitale in grado di dare “voce” ai robot costruiti. Dal punto di vista del rapporto design-funzionalità, abbiamo ritenuto opportuno prevedere un’interfaccia operativa indipendente dal mattoncino programmabile: un pulsante per registrare e uno per riprodurre il suono posti sul registratore stesso. Il prototipo realizzato permette di registrare due brevi messaggi e può essere controllato sia manualmente che da software. L’esperienza con i bambini ha, però, mostrato l’esigenza di aumentare sia il numero che la durata dei messaggi. Un’alternativa possibile è quella di privilegiare la trasparenza dell’interfaccia associando un solo componente ad ogni messaggio. Se si vogliono messaggi più lunghi è sufficiente collegare i registratori tra di loro in sequenza [Ananny, 2001], consentendo la costruzione di una storia attraverso la manipolazione dell’ordine dei registratori. In questo modo sono possibili progetti con complesse interazioni tra sensori e sequenze sonore: per esempio, i bambini possono progettare un “muro dei suoni” che riproduca sequenze diverse se fa caldo, se si tocca un punto sensibile, se due mattoncini si scambiano un segnale, ecc. Proporre le stesse possibilità per lo scenario dei Veicoli pone vincoli costruttivi sulle dimensioni e peso dei componenti, che portano non a moltiplicare i registratori, ma a progettarne uno piccolo, complesso e flessibile anche se forse più opaco nel suo design.

DEFINIRE GLI STRUMENTI DELLA PROGRAMMAZIONE

Rendere accessibili ai bambini gli strumenti di controllo del comportamento di una costruzione cibernetica è un problema aperto. Programmare non è facile e molti pensano che sia un’attività da delegare a specialisti: se questo fosse vero dovremmo abbandonare l’idea di un kit di costruzioni e limitarci alla progettazione di giocattoli cibernetici con un grado forse elevato di interattività, ma sostanzialmente non modificabili dai bambini. Noi pensiamo invece che una versione di programmabilità, purché supportata da strumenti (ambiente, linguaggio, ecc.) specializzati nella direzione delle caratteristiche del problema da risolvere, sia possibile anche per i non programmatori. Sviluppare strumenti che consentano a per-

sone non particolarmente esperte né interessate ad approfondire la disciplina dell’informatica di istruire il computer a risolvere specifici problemi è un’area di ricerca fiorente. In particolare, gli studi di Bonnie Nardi (1993) indicano che utenti esperti in un dominio, o motivati a praticarlo, riescono ad apprendere e gestire linguaggi formali adeguati a quel dominio. Ne sono esempio una classe di strumenti come i fogli elettronici o gli ambienti di analisi statistica che forniscono un linguaggio di programmazione usabile per estendere le funzionalità del sistema. Strumenti di questo tipo hanno consentito lo sviluppo di una popolazione di utenti che, a diversi livelli di competenza, usano le funzionalità di programmazione disponibili.

Trasferire queste considerazioni nel mondo dei bambini implica, innanzitutto, ipotizzare che per loro sia possibile gestire il livello di complessità insito nel controllo del comportamento di un robot. Se questa ipotesi è verificata, ha senso porsi il problema di costruire un ambiente di programmazione per bambini. Nell’ambito del progetto CAB abbiamo verificato che i bambini riescono a interagire con costruzioni robotiche laddove il contesto sia ben strutturato (approfondiremo queste considerazioni nella sezione dedicata ad alcune esperienze realizzate).

Ma qual è il modello concettuale più adeguato alla definizione dei comportamenti di un robot? Vediamo un esempio. Vogliamo che un veicolo giri intorno a un baule a base quadrata. Un neofita tenderebbe a impostare la soluzione come se avesse a disposizione un’automobile controllata con un telecomando, e dal punto di vista della programmazione ad adottare uno stile imperativo, proponendo la seguente soluzione: “Vai avanti di un lato”, “Gira a sinistra di 90 gradi” e “Ripeti queste due istruzioni per gli altri tre lati”, nello stile della tartaruga Logo. Ma i robot sono dotati di sensori che consentono di “percepire” l’ambiente circostante e reagire di conseguenza: programmare un robot implica così gestire più sensori contemporaneamente in tempo reale. Il modello imperativo di programmazione, che è adeguato ad un ampio spettro di situazioni (calcolo scientifico, gestionale, ecc.), mostra così la corda nel contesto della programmazione di robot [Resnick, 1991; Papert, 1993].

Se aggiungiamo, ad esempio, al nostro veicolo un sensore di contatto possiamo affrontare il problema in modo radicalmente diverso, simulando il comportamento di

6

Nell’ambito della ricerca sul design esistono tradizioni, quali il Cooperative Design [Greenbaum e Kyng, 1991] o il Participatory Design [Schuler e Namioka, 1993], che coinvolgono gli utenti fin dalle fasi iniziali di progettazione degli artefatti. Nel Participatory Design esistono attualmente alcuni approcci metodologici, quali quello del *Children as Informant* [Scaife e Rogers, 1999] o del *Cooperative Inquiry* [Druin, 1999, 2002], che vedono la partecipazione attiva e paritaria dei bambini (di età 7-11 anni) nelle fasi di ideazione e progettazione.

7

Sulla metodologia di documentazione utilizzata si veda [Rinaldi, 1995].

8

Furby™ è un peluche tecnologico che richiede l’attenzione e le cure di un bambino. Se accudito, si comporta bene e lentamente passa dalla sua lingua (il furbese) a quella del bambino.

9

Non tutte le proposte hanno incontrato lo stesso favore e gradimento tra i bambini e gli educatori, incoraggiando quindi una revisione profonda o in certi casi l’accantonamento. Quest’ultimo è stato il caso dei Pupazzi Cibernetici che hanno suscitato più perplessità che non adesione tra gli insegnanti ed educatori, convinti di fornire una proposta pedagogicamente discutibile.

10

Gli attuatori sono dispositivi di output, quali i motori.

11

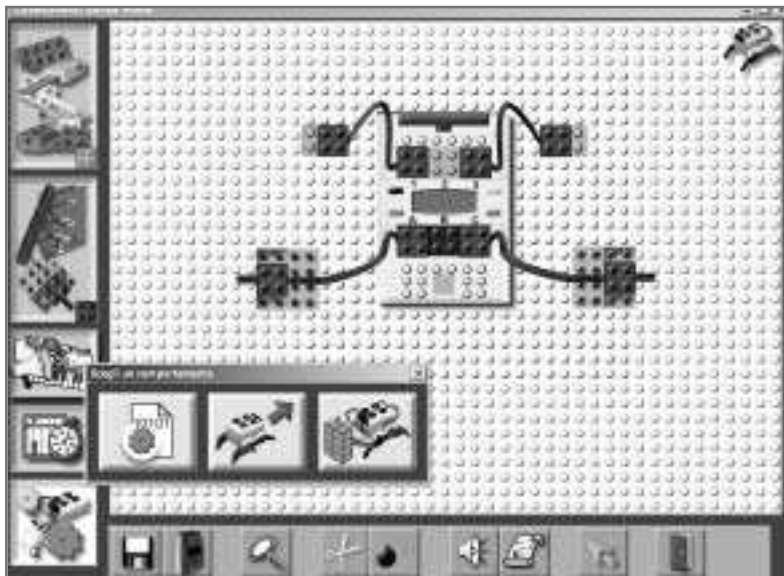
Per una descrizione dettagliata sul funzionamento del sensore di luce, si veda [Giordano, 2002] in questo numero.

chi, trovandosi al buio, debba circumnavigare un ostacolo seguendone il contorno con la mano. Il programma viene costruito mettendo in relazione le informazioni che pervengono dai sensori con i comandi da far eseguire ai motori. Costruire con pezzi LEGO un robot che “tasta” il muro mentre avanza è complesso; risulta più semplice far oscillare tutto il veicolo in modo che vada avanti a zig-zag: il robot si allontana dal muro quando il sensore lo tocca e si riavvicina quando perde il contatto. La relazione tra input e output sarà quindi: “se il sensore tocca, accendi il motore dal suo lato e spegni quello dal lato opposto; se non tocca, spegni il motore dal suo lato e accendi quello dal lato opposto”.

Questo tipo di soluzione offre una serie di vantaggi su quella imperativa: il comportamento emerge dall'interazione tra il robot e l'ostacolo indipendentemente dalla sua forma o dalle sue dimensioni. Inoltre, questo approccio consente, con minime modifiche morfologiche, di risolvere anche altri problemi: ad esempio, se volessimo far seguire al robot una linea sul pavimento¹¹, sarebbe sufficiente sostituire il sensore di contatto con uno di luce mantenendo inalterata la struttura del programma. Il robot seguirà a zig-zag il contorno della linea. In tutti questi casi, piuttosto che rappresentare nel programma la mappa della realtà, è lo stesso “campo di gioco” che funziona da mappa di sé stesso [Brooks, 1991]. In tutte le situazioni in cui non sono note a priori le caratteristiche dell'ambiente (la forma e le dimensioni dell'ostacolo, il tracciato della linea, ecc.) al robot è richiesta una dimensione di adattività conseguibile grazie all'uso di sensori.

figura 3

Una rappresentazione schematica di un veicolo con due sensori di contatto. Il menu dei comportamenti è selezionato, permettendo di scegliere tra i comportamenti disponibili appropriati alla configurazione corrente di input e output o la definizione di uno nuovo.



L'orientamento al dominio

Si è scelto di rappresentare i comportamenti dei robot per mezzo di regole composte da condizioni per i sensori e da azioni per gli attuatori. Una regola associa una condizione (un test sullo stato di un sensore) a una sequenza di azioni (comandi per gli attuatori): ad esempio, “Se il sensore di luce rileva un valore elevato di luminosità allora accendi il motore”. La facilità d'uso di questo sistema a regole dipende dalla disponibilità di condizioni e azioni che incapsolino i dettagli del funzionamento dell'hardware e siano direttamente operativi. A sua volta, l'usabilità di condizioni e azioni dipende da assunzioni sul tipo di costruzione: ad esempio, un veicolo con due motori può muoversi avanti e indietro, girare a destra e a sinistra. Ciò permette di definire comandi per il veicolo analoghi a quelli della tartaruga del Logo. Quindi, per ogni tipologia di costruzione, l'ambiente di programmazione mette a disposizione una serie di primitive (condizioni e azioni) specializzate.

Il comportamento complessivo di una costruzione emerge dalla composizione di comportamenti elementari che agiscono in maniera indipendente, ma concomitante. Per esempio, un veicolo con sensori di contatto che si muove reagendo ad eventuali ostacoli può essere controllato da due comportamenti: il primo dice al veicolo di muoversi in avanti; se si incontra un ostacolo, interviene un secondo comportamento per dire al robot di indietreggiare e girare nella direzione opposta al lato dell'urto. Il comportamento può essere costruito e provato in maniera incrementale ed indipendente nelle sue componenti. Nel caso in cui più comportamenti governino contemporaneamente uno stesso attuttore un meccanismo di priorità decide chi detiene il controllo delle azioni. Nell'esempio del veicolo che si muove tra gli ostacoli, il comportamento che gestisce gli urti è prioritario su quello che controlla il movimento in avanti del veicolo [figura 3].

L'ambiente di programmazione si presenta innanzitutto come un laboratorio di progetti, che ospita sia la galleria di quelli già esistenti che la possibilità di definirne uno nuovo. Un progetto, a sua volta, contiene sia i programmi che la documentazione multimediale del lavoro (foto, filmati, commenti sonori, testi) ed è composto da una o più costruzioni. L'ambiente permette di definire tipologie diverse di costruzioni per sostenere la specializzazione delle componenti (comportamenti, condizioni e azio-

ni). La scelta del tipo di costruzione permette di fare delle assunzioni sulle sue componenti meccaniche. Ad esempio, un veicolo è dotato di uno chassis con due motori e quindi può muoversi e girare. Quando lo si dota di sensori appropriati, il veicolo può seguire una serie di comportamenti predefiniti quali “Segui una linea”, “Cerca la luce”, “Segui un muro”. L’ambiente è capace di suggerire comportamenti appropriati a seconda dei sensori usati in una data costruzione [vedi figura 3]. Quando si definisce un nuovo comportamento, solo le condizioni e azioni appropriate alla configurazione hardware corrente sono presentate [vedi figura 4]. Grazie a questo meccanismo di specializzazione è possibile far evolvere l’ambiente secondo le necessità dettate dagli specifici progetti dei bambini.

L’interfaccia software cerca di rendere evidente cosa è possibile fare con il mattoncino programmabile organizzando le funzionalità in scatole che contengono componenti dello stesso tipo [vedi figura 3]. La struttura in scatole riflette, inoltre, una distinzione tra concreto e virtuale: sensori e attuatori fisici; funzionalità presenti nel mattoncino (ad esempio, suoni e messaggi) che non sono associate a componenti fisiche collegabili; strumenti virtuali, quelli cioè costruiti con il software, quali orologi e contatori. Questa tassonomia, riflessa nella struttura operativa dell’interfaccia, permette ai bambini di scoprire le funzionalità disponibili secondo una logica che va dal concreto all’astratto.

Il supporto metacognitivo e sociale

Formalizzare mediante regole il comportamento di un robot ha importanti implicazioni sul piano cognitivo e metacognitivo. Da un lato, la regola reifica la relazione di causa-effetto e fornisce ai bambini e ai loro insegnanti un importante strumento linguistico per astrarre i comportamenti reattivi (“Se la temperatura aumenta, il robot accende il motore della ventola”).

Dall’altro, l’immediatezza di interpretazione e la leggibilità delle regole consentono non solo di parlare facilmente del problema, ma anche, in retrospettiva, di ripercorrere i processi di soluzione messi in atto (“... poi abbiamo aggiunto questa regola, per insegnare al robot ad accendere il ventilatore quando fa caldo ...”), specie quando le scelte di programmazione effettuate non producono i comportamenti attesi. Caratteristica tipica del lavoro dei bambini è, infatti, quella di prefigurare un contesto ampio ed

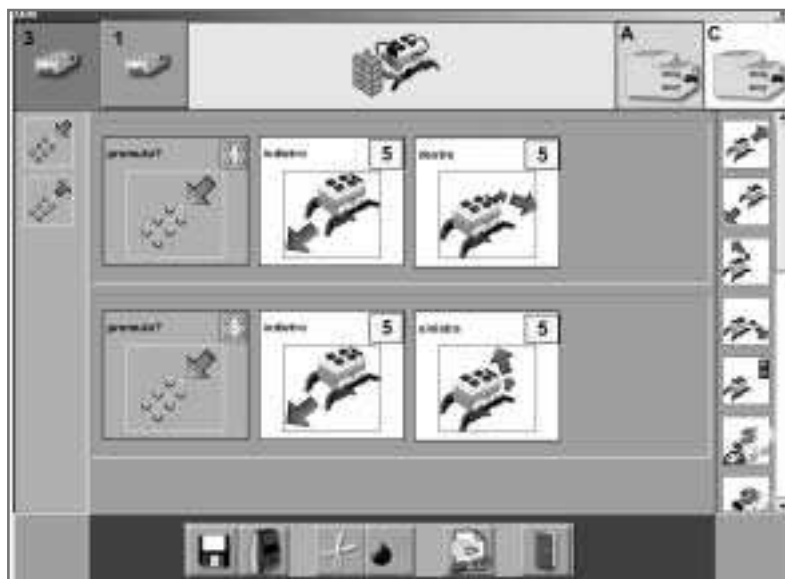
articolato dove le fantasie prendono corpo ed evolvono coinvolgendo più attori (si veda l’esempio riportato più avanti dei “Mostri e Difensori”). Un progetto in evoluzione ha quindi bisogno di forme atte a sostenere la memoria del lavoro compiuto, sia in termini di documentazione che di storia delle scelte costruttive e programmatiche, eventualmente ripercorribile.

L’ambiente prefigura, inoltre, un contesto sociale d’uso articolato su tre ruoli: i bambini, gli insegnanti, gli esperti. I *bambini* collaborano tra loro e con gli insegnanti in tutte le fasi del lavoro, dall’identificazione del problema all’invenzione di una soluzione appropriata e alla sua realizzazione: discutono e si confrontano sulle alternative possibili, ispezionano esempi e semilavorati, li modificano nei parametri e nella struttura; esplorano le potenzialità e i limiti della tecnologia; si impegnano in un processo iterativo di costruzione socialmente condivisa in cui le ipotesi che emergono vanno soggette sia al vaglio del gruppo, sia alla verifica sperimentale.

Gli *insegnanti* mediano tra bambini e tecnologia quel tanto che basta per rendere fluido il processo, senza deprimere la creatività e la motivazione dei bambini. Alcune funzioni di configurazione dell’ambiente di programmazione sono finalizzate a renderlo adattivo rispetto alle esigenze del gruppo e del particolare problema in esame: un insegnante può, ad esempio, configurare facilmente l’interfaccia, cambiando le icone e i nomi associati agli oggetti manipolati dal programma (azioni, condizioni, comportamenti). Ma è anche possibile (e operativamente facile) per un insegnante modificare

figura 4

Le due regole che definiscono il comportamento “urto”. N.B.: l’interfaccia visualizza solo le condizioni e azioni associate ai dispositivi di input e output selezionati.



diversi parametri di configurazione di azioni e condizioni predefinite: ad esempio, i fattori di scala associati alle durate temporali di azioni quali “Avanti” o le soglie di sensibilità sui vari sensori.

Gli *esperti* nella programmazione di robot possono estendere ulteriormente l'ambiente per integrare la definizione di nuove tipologie di costruzioni.

ALCUNE ESPERIENZE REALIZZATE CON I BAMBINI

Nel progetto CAB la sperimentazione si è articolata nell'arco due anni e ha coinvolto tre scuole dell'infanzia ed una scuola elementare del Comune di Reggio Emilia e tre scuole elementari in Svezia. Per una descrizione approfondita rimandiamo ai rapporti finali dei partner educativi [CRE, 2001; Gustafsson e Lindh, 2001]. Qui ci limitiamo a proporre la rilettura di tre esperienze realizzate nelle scuole dell'infanzia, con l'obiettivo di esplicitare le relazioni tra ricerca in classe e progettazione e sviluppo dell'ambiente di programmazione usabile anche dai bambini di cinque anni.

Anche e soprattutto per questa fascia di età, le esperienze riportate ci permettono di argomentare che: 1) non ci sono ostacoli cognitivi alla programmazione di costruzioni cibernetiche da parte dei bambini in presenza di un contesto ben definito e di strumenti specializzati; 2) per sostenere la complessità presente nei progetti dei bambini bisogna fornire un modello che incapsuli i dettagli implementativi; 3) l'ambiente proposto riveste carattere di usabilità, in quanto dotato di caratteristiche atte a renderlo appetibile, ispezionabile secondo una data granularità e “trattabile” (oggetto di discussione e di riflessione anche metacognitiva).

Specifico o generico?

Un gruppo di bambini della scuola “La Villetta” ha usato il kit specializzato RoboSports, sviluppato dalla LEGO per permettere ai visitatori del LEGOLAND park di partecipare a gare di robot. Il kit comprende un campo di gara per due squadre che competono nella realizzazione di un veicolo che porti il maggior numero possibile di palline in buca. Il campo è costituito da un tavolo con due piste ognuna composta da una linea nera e una buca retroilluminata. Le componenti meccaniche sono specializzate per costruire un set limitato di veicoli capaci di trasportare e spingere le palline in buca. L'ambiente software fornisce primitive quali il comportamento di “Segui una li-

nea”, interrompibile quando il sensore di luce “vede” la buca retroilluminata, e comandi di traslazione e rotazione per spingere le palline in buca. Il kit è completato da video esplicativi per guidare gli utenti nella costruzione e programmazione dei veicoli. Alla scuola “La Villetta” i genitori e gli insegnanti hanno ricreato il campo da gioco e i bambini hanno costruito e programmato i veicoli per la gara.

L'esperienza ha dimostrato che i bambini riescono a gestire il compito e, cosa più importante, ad appropriarsi del linguaggio formale utilizzato per descrivere i loro tentativi e ragionare in gruppo sul problema. Infatti, la specializzazione delle componenti ha reso semplice la costruzione di un veicolo adeguato al compito; l'ambiente di programmazione visivo, che fornisce un set limitato di primitive, ha consentito ai bambini di comporre autonomamente il programma. Inoltre, i bambini hanno usato le icone dei comandi per annotare il campo di gara e discutere sugli effetti delle istruzioni impartite al robot quando queste non sortivano l'effetto desiderato [CRE, 2000].

RoboSports ha dimostrato di essere un buon esempio delle potenzialità del mattoncino programmabile, in quanto propone un contesto specifico e rende disponibili costrutti linguistici ad hoc per la soluzione del problema presentato. Il suo limite sta però nell'essere troppo specializzato: le componenti hardware e software si possono usare solo per questa gara, o per gare di questo tipo, ma non possono essere trasferite ad una gamma più ampia di situazioni, finendo con l'ostacolare la creatività e lo sviluppo di progetti originati direttamente dai bambini.

Sostenere la complessità

Restiamo nella tipologia di gare di robot, questa volta con un progetto che è il risultato dell'immaginazione e del lavoro dei bambini. Nella scuola “Neruda” l'interesse che i bambini di una sezione avevano coltivato negli anni per i mostri ha fornito lo spunto per un progetto che esplorasse forme di vita artificiale. Questa “vita” si sviluppa ed evolve secondo la frequenza e la qualità delle relazioni tra gli “attori”, i Mostri e i Difensori della città [Barchi et al, 2001]. I mostri attaccano la città, gli abitanti costruiscono mura e trappole per difendersi e organizzano una squadra di difensori per tener lontani i mostri. I mostri e i difensori sono stati costruiti cercando di definire comportamenti che rendessero conto di una dinamica di scontro in cui non sono chiari

dall'inizio né l'evoluzione né l'esito. Su un grande tavolo i bambini hanno realizzato lo scenario e aggiunto quattro veicoli (due difensori e due mostri). I mostri hanno due sensori di contatto programmati per aggirare eventuali ostacoli e un sensore di luce puntato sul tavolo che arresta il veicolo se incontra una zona di colore nero. Il nero identifica sia le trappole che l'ingresso della città. Inoltre, i mostri hanno una pila montata sul dorso che li rende riconoscibili dai difensori. Questi ultimi hanno un sensore di luce che permette loro di muoversi nella direzione di attacco.

Questa prima definizione dei comportamenti ha fatto sì che i mostri e i difensori, dopo un primo urto, avanzassero a caso sul campo di gara senza un chiaro obiettivo. Soltanto casualmente un mostro finiva in trappola o riusciva a entrare in città; i difensori vagavano senza una chiara strategia per bloccare i mostri, con cui a volte si scontravano. I bambini si sono resi conto dei limiti e hanno proposto alcune alternative, ma gli insegnanti hanno incontrato un ostacolo nel realizzare i meccanismi di "attrazione" tra i difensori e i mostri, e tra i mostri e la città. È stato a questo punto fondamentale il contributo degli esperti che hanno suggerito due possibili modifiche al progetto per realizzare l'attrattiva: introdurre delle piste di altro colore che indicassero ai mostri la direzione per arrivare alle porte della città e modificare il programma dei difensori attivando un meccanismo di ricerca dei mostri, consistente nel girarsi intorno per rilevare la direzione della luce. Queste proposte sono state discusse dagli insegnanti con i bambini, che hanno modificato lo scenario e i robot ottenendo comportamenti infine conformi ai desiderata.

Questo lavoro ha coperto un arco temporale piuttosto lungo e articolato in varie fasi: progettazione e realizzazione dello scenario; realizzazione, programmazione e sperimentazione; modifica dei comportamenti. La partecipazione in qualità di esperti a questa attività ha consentito agli autori di questo articolo, impegnati nella progettazione dell'ambiente software descritto in precedenza, di appurare i seguenti aspetti: è possibile catturare la complessità di progetti di questo tipo nel modello proposto; comportamenti complessi quali quelli evidenziati da questa esperienza difficilmente possono essere sviluppati in autonomia dai bambini, ma possono far parte di un repertorio di componenti specializzati applicabili e ispezionabili direttamente dai bambini.

Questo progetto è risultato essere un buon esempio dello stile di interazione con oggetti cibernetici che Edith Ackermann definisce come "Giocare allo psicologo" [Ackermann, 1991, 2002]. "Giocare allo psicologo" e "Giocare all'ingegnere" costituiscono i due estremi di ruoli potenziali che i bambini assumono quando interagiscono con kit di costruzioni cibernetiche. Nel primo, prevale l'atteggiamento di osservazione in cui i bambini tipicamente si interrogano sulla natura dell'oggetto che hanno di fronte (sulle sue intenzioni, "intelligenza", ecc.), nel tentativo di capirne l'intima natura; nel secondo, prevale l'atteggiamento costruttivo o di modifica degli oggetti e dei loro comportamenti. Ma i bambini, sostiene Ackermann, oscillano continuamente tra questi due estremi, esistendo infatti gradi diversi di breakdown e di situazioni in cui la componente dello psicologo e quella dell'ingegnere vengono calibrate e combinate diversamente.

Diventare programmatori

«Siamo diventati dei programmatori di robot!

È vero! Questa è una scuola di programmatori! Possiamo fare tante cose!

Abbiamo scoperto tre segreti:

il primo è che due pezzettini del metro fanno una mattonella;

il secondo è che se l'uccellino tocca il sensore di piegamento il registratore fa cip cip;

il terzo è che i robot con la busta e la cassetta delle lettere si possono parlare.»

[CRE, 2001]

Nella scuola "La Villetta" alcuni bambini di cinque anni dell'anno scolastico 1999/2000 avevano cercato di ridare vita ad un ramo di un albero del giardino che si era spezzato durante una nevicata. Il ramo era stato portato dentro la scuola e "fornito di nuova vita" con l'aggiunta di sensori e attuatori che gli permettevano di reagire alla luce, al contatto con le foglie, ecc.

Nel successivo anno scolastico altri bambini, che nell'anno precedente avevano assistito all'inserimento del ramo nella vita della scuola, hanno deciso di estendere il progetto aggiungendo un uccellino che nel duro inverno chiede aiuto ad un amico robot per procurarsi del cibo. Il robot porta le briciole di pane sotto l'albero e chiama il suo amico uccellino che scende a raccogliere.

Questi bambini, che sono cresciuti in una realtà in cui le costruzioni cibernetiche erano già presenti e l'idea di programmazione

era stata sperimentata con RoboSports, hanno trovato naturale realizzare e programmare le loro idee con un prototipo di ambiente messo loro a disposizione.

Le frasi riportate all'inizio della sezione testimoniano il loro livello di consapevolezza raggiunto alla fine dell'esperienza. Durante il progetto i bambini hanno dovuto risolvere una serie di problemi. Il primo era quello di determinare il valore da dare al comando "Avanti" per far sì che il robot si muovesse di sei mattonelle sul pavimento. "Due pezzettini del metro fanno una mattonella" esemplifica il percorso che i bambini hanno seguito per correlare il parametro del comando "Avanti", reso disponibile dal nostro ambiente per i Veicoli, con l'effettivo spazio percorso dal robot. I bambini hanno costruito uno strumento di misura, che hanno chiamato "metro", per associare il valore del parametro ad un elemento del contesto, le mattonelle del pavimento. Poiché di quanto si muove un veicolo è funzione delle scelte fatte nella sua costruzione, il software permette di modificare un coefficiente di scala per rendere significativi i valori del parametro. Questa operazione di taratura è stata eseguita dagli insegnanti; i bambini hanno eseguito le misure e, verificato che bisognava muoversi in avanti di 12, hanno scomposto questo risultato inserendo due comandi "Avanti" ($10 + 2$), visto che l'interfaccia forniva uno slider con valori da 0 a 10 [figura 5].

Quest'esperienza ci ricorda che quando i bambini sono motivati riescono a mantenere a lungo la concentrazione e dimostrano competenze che in genere non ci si aspetta a quest'età. Poiché abbiamo scelto di rap-

presentare il parametro di "Avanti" solo con un numero, senza specificarne l'unità di misura, abbiamo favorito l'interpretazione dei bambini secondo cui il parametro indica una distanza. In realtà, il parametro controlla il tempo di accensione dei motori, ma portare in primo piano questo dettaglio costringerebbe a tener conto della corrispondenza tra tempi e distanze. Ciò è stato risolto dall'intervento degli insegnanti che hanno opportunamente tarato il fattore di scala per far corrispondere "Avanti 2" ad una mattonella, permettendo così ai bambini di controllare la situazione.

I bambini hanno realizzato il loro progetto in modo da poter raccontare la storia mentre veniva rappresentata dall'uccellino e dal robot. I punti di sincronizzazione sono essenziali al buon risultato della messa in scena: il robot deve iniziare a muoversi quando l'uccellino lo chiama, e l'uccellino deve scendere quando il robot arriva sotto l'albero. Una prima soluzione è stata quella di usare un sensore di suono per ascoltare la voce dell'uccellino ("se l'uccellino tocca il sensore di piegamento il registratore fa cip cip"). Questa soluzione, però, non ha retto alla verifica sperimentale. Il sensore di suono non distingue tra cinguettio e altri rumori attivandosi quindi in presenza di qualunque tipo di suono. L'organizzazione per scatole dell'interfaccia iconica ha aiutato i bambini a esplorare possibili alternative e scoprire autonomamente che "i robot con la busta e la cassetta delle lettere si possono parlare". Lo scambio di messaggi ha fornito un robusto meccanismo di sincronizzazione permettendo ai bambini di riprogrammare il robot portapane (figura 5) e completare il loro progetto adottando un meccanismo analogo per dire all'uccellino di scendere a mangiare quando il robot raggiunge l'albero.

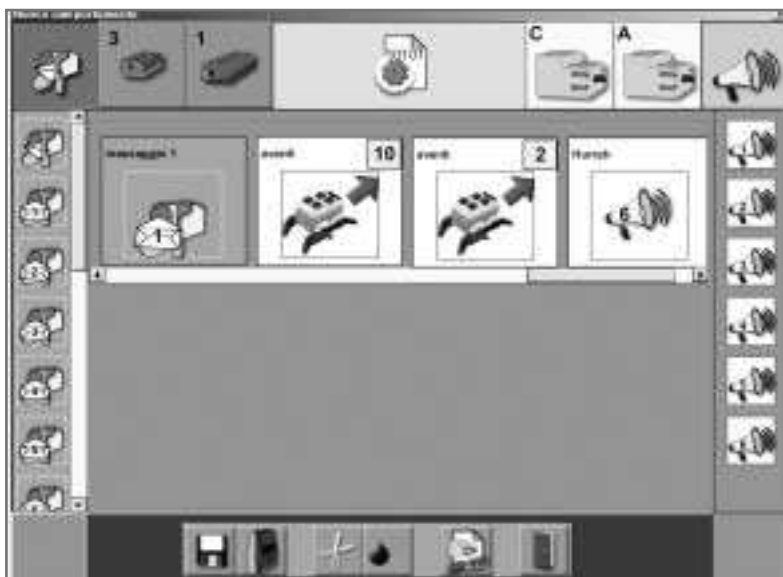
L'evoluzione del progetto e il modo in cui i bambini sono riusciti a gestire i problemi sorti durante le fasi di programmazione sottolinea l'appropriabilità e trattabilità dell'interfaccia proposta.

CONCLUSIONI

Il progetto CAB ha risposto in modo affermativo alla domanda: è possibile imparare costruendo robot nella scuola dell'infanzia? Non sembrano infatti esserci ostacoli cognitivi alla programmazione di costruzioni cibernetiche da parte dei bambini in presenza di un contesto ben definito e di strumenti specializzati. L'esplorazione della sensorialità, reattività, controllo e autonomia e lo

figura 5

La definizione del comportamento che dice al robot di muoversi in avanti quando riceve un messaggio. Nell'angolo in alto a sinistra sono mostrati tre sensori: quello dei messaggi (selezionato), uno di suono e uno di luce. Per far muovere il robot "Avanti 12" i bambini hanno usato due comandi ($10 + 2$), poiché l'interfaccia prevede uno slider (non mostrato in figura) che limita la scelta del valore del parametro all'intervallo [0 ... 10].



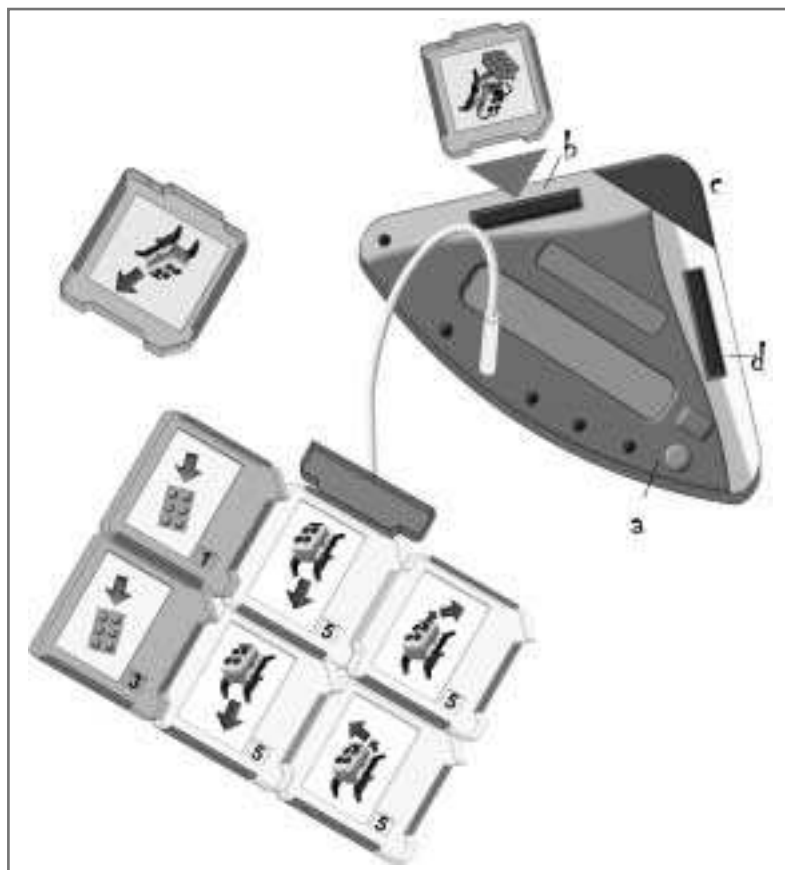
sviluppo delle teorie associate sono favoriti nei bambini dal confronto con la costruzione di oggetti cibernetici che le reificano.

L'ambiente di programmazione proposto permette di adattare le primitive messe a disposizione alle tipologie di costruzioni usate dai bambini nei loro progetti. Le scelte progettuali per l'interfaccia si sono rivelate accessibili ed usabili anche dai bambini di cinque anni, consentendo loro di programmare i comportamenti delle costruzioni e soprattutto di risolvere i problemi sorti durante l'evoluzione dei progetti.

I risultati raggiunti vanno comunque riferiti ad un contesto di ricerca dove la mediazione degli adulti poteva avvalersi di risorse, competenze e tempi di lavoro difficilmente rintracciabili nei normali contesti educativi. Nell'ambito del progetto abbiamo sviluppato alcune idee su possibili evoluzioni del materiale costruttivo che potrebbero permettere un uso creativo e più autonomo del materiale per i bambini e forse aprire la strada a sperimentazioni per una fascia di età di quattro anni [Ackermann, 2002]. Una prima indicazione riguarda un'ulteriore riduzione della complessità meccanica delle costruzioni: il kit, ad esempio, dovrebbe prevedere componenti pre-assemblate che permettano trasformazioni semplici di movimenti (da rotatorio orizzontale a rotatorio verticale, da veloce ma debole a lento ma potente, ecc.) senza scendere al livello dei singoli ingranaggi; oppure introdurre nuovi sensori e attuatori per aprire la strada ad attività costruttive che prevedano una pluralità di materiali, non solo LEGO, in cui emerga da subito il ruolo della programmazione, come nello scenario del "muro dei suoni".

Un'altra indicazione riguarda il design dei singoli componenti per migliorarne la leggibilità. Le componenti attive del kit, i sensori e gli attuatori, dovrebbero infatti rivelare la loro natura sia proponendo una plastica trasparente che permetta di vedere le parti interne, sia aggiungendo una risposta visiva: per esempio, l'aggiunta di una barra di LED sui sensori darebbe un'indicazione dell'intensità dell'input.

Infine, ci siamo chiesti: perché i bambini non dovrebbero poter costruire i programmi con componenti tangibili così come fanno con i LEGO? Questo approccio, che va sotto il nome di "programmazione tangibile" [Suzuki e Kato, 1995], è un settore di ricerca fiorente in cui sono presenti molti progetti indirizzati ai bambini [McNerney, 1999; Wyeth & Wyeth, 2001; Montemayor



et al, 2002]. Una versione tangibile dell'ambiente di programmazione descritto precedentemente potrebbe realizzare la visione di mischiare Atomi e Bit in maniera concreta e facile da usare [figura 6]. Comportamenti, condizioni e azioni potrebbero essere dei tasselli fisici che contengono l'elettronica necessaria a riconoscere automaticamente la topologia del puzzle [Gorbet et al, 1998] e a generare il programma corrispondente. I bambini potrebbero lavorare in modo cooperativo manipolando le componenti della programmazione così come fanno con le parti fisiche della costruzione. Ciò non sostituirebbe il computer, ma lo integrerebbe portando il suo ruolo sullo sfondo nelle fasi costruttive e in primo piano in quelle di riflessione e documentazione.

RINGRAZIAMENTI

Il lavoro descritto in questo articolo non sarebbe stato possibile senza l'impegno, l'entusiasmo e la creatività dei bambini e degli insegnanti delle scuole svedesi e italiane che hanno partecipato al progetto CAB. In particolare siamo debitori ai bambini della scuola "La Villetta" del titolo di questo articolo.

I nostri partner sono stati portatori di istanze e punti di riferimento diversi, cia-

figura 6

Una versione tangibile della definizione del comportamento "urto". Le etichette sulla stazione a cui si connettono i tasselli ne indicano le funzionalità: a) per definire una regola, connettere condizioni e azioni su questo lato; b) la regola corrente può essere usata per definire un comportamento inserito in questo slot; c) la porta di comunicazione con il PC; d) uno slot aggiuntivo per ridefinire il significato di un tassello di condizione o azione.

scuno dei quali ha concorso alla realizzazione di un approccio multidisciplinare e multiculturale che si è profondamente avvantaggiato della varietà e delle differenze preesistenti. Vogliamo ricordare in particolare Paola Barchi, Gino Ferri, Giovanni Piazza e Maura Rovacchi delle scuole dell'infanzia del Comune di Reggio Emilia per aver mantenuto sempre presente nel nostro lavoro il punto di vista dei bambini; Gaute Munch e Martin Rausch della LE-

GO A/S per il loro contributo all'ambiente di programmazione.

Siamo grati a Edith Ackermann per aver accettato di scrivere un articolo insieme fornendo un punto di vista complementare sul significato della programmazione per i bambini e aiutandoci a riflettere sul nostro lavoro. Vincenza Benigno e Donatella Persico hanno rivisto versioni preliminari di questo articolo fornendo preziose indicazioni per migliorarne la leggibilità.

PER SAPERNE DI PIÙ

Chi vuole approfondire le tematiche della robotica educativa ha a disposizione diverse fonti di informazione. Nel seguito è riportata una selezione di riferimenti bibliografici e siti web¹ che privilegia materiali relativi al LEGO MindStorms. Infine si propone un elenco di altri prodotti per la robotica educativa.

Una bibliografia su LEGO MindStorms

Una raccolta di progetti per ragazzi della scuola media:

Erwin B. (2001), *Creative Projects with LEGO Mindstorms*, Addison-Wesley Pub Co, Boston, MA.

Un libro di testo per un corso introduttivo a livello universitario:
Martin F. G. (2001), *Robotic Explorations: An Introduction to Engineering Through Design*, Prentice Hall, Upper Saddle River, NJ.

Libri che trattano la robotica a livello hobbistico e presuppongono competenze ingegneristiche e di programmazione:

Bagnall B. (2002), *Core LEGO MINDSTORMS Programming: Unleash the Power of the Java Platform*, Prentice Hall, Upper Saddle River, NJ.

Baum D. (2002), *Dave Baum's Definitive Guide to LEGO Mindstorms, 2nd edition*, Apress, Berkeley, CA.

Baum D., Gasperi M., Hempel R., Villa L. (2000), *Extreme Mindstorms: an Advanced Guide to LEGO Mindstorms*, Apress, Berkeley, CA.

Ferrari M., Ferrari F. (2001), *Building Robots With LEGO Mindstorms: The Ultimate Tool for Mindstorms Maniacs*, Syngress Media Inc., Rockland, MA.

Laverde D. (ed.) (2002), *Programming Lego Mindstorms with Java*, Syngress Media Inc., Rockland, MA.

Materiali didattici

Il Center for Engineering Educational Outreach della Tufts University mette a disposizione:

The LEGO Curriculum Site:

<http://www.ceeo.tufts.edu/curriculum/>

The Science and Engineering NASA Site of Remote Sensing:

<http://www.ceeo.tufts.edu/sensors/>

Il MediaLab del MIT offre un'enciclopedia di costruzioni robotiche (*constructopedia*):

<http://constructopedia.media.mit.edu/>

Il sito della Playful Invention and Exploration (PIE) Network, una rete di musei americani che in collaborazione con il MIT Media Lab sviluppa progetti e attività per integrare scienza, tecnologie e arte:

<http://www.smm.org/pie/>

Materiali informativi e software

I *LEGO User Groups* dedicati alla robotica costituiscono la più vasta e aggiornata fonte di informazioni su questo tema:

<http://news.lugnet.com/robotics/>

Ulteriori linguaggi e ambienti per programmare il LEGO MindStorms:

Not Quite C (NQC) è un linguaggio *Open Source* in sintassi C. <http://www.baumfamily.org/nqc/>

Bricx Command Center è un ambiente di sviluppo *Open Source* per utilizzare il linguaggio NQC in ambiente Windows.

<http://hometown.aol.com/johnbinder/bricxcc.htm>

LeJOS è un linguaggio *Open Source* in sintassi JAVA.

<http://lejos.sourceforge.net/>

MindOver è un videogioco dove uno o più giocatori costruiscono e programmano robot virtuali che competono in missioni varie. Ne esiste una versione che permette sia di simulare il comportamento di un robot fisico all'interno del videogioco, che di scaricare il programma creato sul robot reale.

<http://www.mindrover.com/>

Prodotti

Il *LEGO MindStorms* è il prodotto di maggior successo in questo ambito. Esistono però altre offerte significative che coprono un ampio spettro di esigenze a cui corrispondono livelli di usabilità diversi. Per esempio *TechnoK'NEX* è un mattoncino programmabile compatibile con vari kit meccanici, tra cui *LEGO Technic*; gli altri kit di costruzione richiedono competenze ingegneristiche poiché permettono di integrare componenti elettronici esterni. In alcuni casi il prodotto di base è lo stesso ma la veste commerciale ne propone specializzazioni diverse: *Il tuo Robot De Agostini* è una versione personalizzata del *Basic Stamps* di Parallax. Infine *Khepera* e *AIBO* sono due robot mobili complessi e potenti che privilegiano gli aspetti di programmazione fornendo una struttura meccanica e apparati di input integrati.

Base per robot mobili

Khepera, <http://www.k-team.com/robots/khepera/>

Il tuo Robot (De Agostini), <http://robot.deagostini.it/>

Kit di costruzione

Basic Stamps versione education (Parallax),

<http://www.stampsinclass.com/>

Handy Board e Handy Cricket,

<http://www.handyboard.com>

TechnoK'NEX Computer Control System, <http://www.knexus-education.com/leonardo.html>

Robot mobili programmabili

AIBO della Sony si può programmare grazie al kit software disponibile in <http://www.aibo.com/open/>

¹ Tutti i riferimenti sono aggiornati a ottobre 2002.

- Ackermann E (1991), The "Agency" Model of Transactions: Toward an Understanding of Children's Theory of Control, in Harel I. & Papert S. (eds), *Constructionism*, Ablex Publisher, Norwood, NJ.
- Ackermann E (2001), Constructivisme et constructionnisme: quelle différence?, in *Actes du colloque "Constructivisme: usages et perspectives en éducation"*, 4-8 Settembre 2000, Genève, Suisse, vol. 1, pp. 85-97.
- Ackermann E (2002), Ambienti di gioco programmabili: cos'è possibile per un bambino di quattro anni?, *TD - Tecnologie Didattiche*, n. 27, pp. 48-55.
- Ananny M. (2001), *Telling Tales: Supporting written literacy with computational toys*, Unpublished Master Thesis, The MIT Media Laboratory, Cambridge, MA.
- Askildsen T., Barchi P., Cagliari P., Chiocciariello A., Giacomini E., Gustafsson B., Lindh J., Manca S., Pausch M., Sarti L. (2001a), *Construction kits made of Atoms and Bits. Research findings & perspectives*, CAB Del. 25, <http://cab.itd.ge.cnr.it/public/papers.htm>
- Askildsen T., Chiocciariello A., Manca S., Munch G., Pausch M., Sarti L. (2001b), *A Cybernetic Construction Kit for Young Children*, CAB Del. 22, <http://cab.itd.ge.cnr.it/public/deliverables/del22.pdf>
- Barchi P., Cagliari P., Giacomini E. (2001), Encounters between children and robotics, in Askildsen T., Barchi P., Cagliari P., Chiocciariello A., Giacomini E., Gustafsson B., Lindh J., Manca S., Pausch M., Sarti L., *Construction kits made of Atoms and Bits. Research findings & perspectives*, CAB Del. 25, <http://cab.itd.ge.cnr.it/public/deliverables/booklet/booklet-CPE.pdf>
- Bers M., Urrea C. (2000), Technological Prayers: Parents and Children Working with Robotics and Values, in Druin A. & Hendler J. (eds), *Robots for Kids: Exploring New Technologies for Learning Experiences*, Morgan Kaufman Academic Press, San Francisco.
- Brooks R. A. (1991), Intelligence without representation, *Artificial Intelligence*, vol. 47, pp. 139-159.
- CRE (2000), *Pedagogical results 2*, CAB Del. 17.
- CRE (2001), *Final pedagogical report*, CAB Del. 23, <http://cab.itd.ge.cnr.it/public/deliverables/del23.pdf>
- Dewey J. (1910), *How we think*, Health, Boston (tr. it., *Come pensiamo*, La Nuova Italia, Firenze, 1973).
- Druin A. (1999), Cooperative Inquiry: Developing New Technologies for Children with Children, in *Proceedings of CHI '99*, ACM Press, New York, pp. 223-230.
- Druin A. (2002), The Role of Children in the Design of New Technology, *Behaviour and Information Technology*, 21(1), pp. 1-25.
- Giordano E. (2002), Percorsi di apprendimento, *TD - Tecnologie Didattiche*, n. 27, pp. 21-28.
- Gorbet M., Orth M., Ishii H. (1998), Triangles: Tangible Interface for Manipulation and Exploration of Digital Information Topography, in *Proceedings on Human factors in Computing Systems Conference (CHI '98)*, April 18-23, 1998, Los Angeles, USA, pp. 49-56.
- Greenbaum J., Kyng M. (eds) (1991), *Design at work: Cooperative design of computer systems*, Lawrence Erlbaum Associates, Hillsdale, NJ.
- Gustafsson B., Lindh J. (2001), *Swedish field test. Final report*, CAB Del. 24, <http://cab.itd.ge.cnr.it/public/deliverables/del24.pdf>
- Harel I., Papert S. (eds) (1991), *Constructionism*, Ablex Publishing, Norwood, NJ.
- Malaguzzi L. (1995), La storia, le idee, la cultura, in Edwards C., Gandini L., Forman G. (eds), *I cento linguaggi dei bambini. L'approccio di Reggio Emilia all'educazione dell'infanzia*, Edizioni Junior, Bergamo.
- Martin F., Mikhak B., Resnick M., Silverman B., Berg R. (2000), To Mindstorms and Beyond: Evolution of a Construction Kit for Magical Machines, in Druin A. & Hendler J. (eds), *Robots for Kids: Exploring New Technologies for Learning Experiences*, Morgan Kaufman Academic Press, San Francisco.
- McNerney T. S. (1999), *Tangible Programming Bricks: An Approach to Making Programming Accessible to Everyone*, Master's thesis, MIT, Cambridge, MA.
- Montemayor J., Druin A., Farber A., Simms S., Churaman W., D'Amour A. (2002), Physical programming: Designing tools for children to create physical interactive environments, in *Proceedings of the SIGCHI conference on Human factors in computing systems: Changing our world, changing ourselves (CHI 2002)*, ACM Press, pp. 299-306.
- Nardi B. (1993), *A Small Matter of Programming: Perspectives on End User Computing*, The MIT Press, Cambridge, MA.
- Papert S. (1980), *Mindstorms: Children, Computers, and Powerful Ideas*, Basic Books, New York (tr. it., *Mindstorms. Bambini, computers e creatività*, Emme, Milano, 1984).
- Papert S. (1993), *The Children's Machine. Rethinking School in the Age of the Computer*, Basic Books, New York (tr. it., *I bambini e il computer*, Fizzoli, Milano, 1994).
- Piaget J. (1975), *L'Equilibration des structures cognitives*, Puf, Paris (tr. it., *L'equilibratura delle strutture cognitive. Problema centrale dello sviluppo*, Boringhieri, Torino, 1981).
- Pontecorvo C., Ajello A. M., Zucchermaglio C. (eds) (1995), *I contesti sociali dell'apprendimento*, LED - Edizioni Universitarie di Lettere Economia Diritto, Milano.
- Resnick M. (1991), MultiLogo: A Study of Children and Concurrent Programming, *Interactive Learning Environments*, vol. 1, n. 3, pp. 153-170.
- Resnick M. (1996), Distributed Constructionism, in *Proceedings of the International Conference on the Learning Sciences*, July 1996, Evanston, USA.
- Resnick M., Berg R., Eisenberg M. (2000), Beyond Black Boxes: Bringing Transparency and Aesthetics Back to Scientific Investigation, *Journal of the Learning Sciences*, vol. 9, n. 1, pp. 7-30 (tr. it. in questo numero).
- Resnick M., Martin F., Sargent R., Silverman B. (1996), Programmable Bricks: Toys to think with, *IBM Systems Journal*, vol. 35, n. 3/4.
- Resnick M., Ocko S. (1991), LEGO/Logo: Learning Thorough and About Design, in Harel I. & Papert S. (eds), *Constructionism*, Ablex Publisher, Norwood, NJ.
- Rinaldi C. (1995), La costruzione del progetto educativo, in Edwards C., Gandini L., Forman G. (eds), *I cento linguaggi dei bambini. L'approccio di Reggio Emilia all'educazione dell'infanzia*, Edizioni Junior, Bergamo.
- Salomon G. (ed) (1993), *Distributed cognition: Psychological and educational considerations*, Cambridge University Press, Cambridge, UK.
- Scaife M., Rogers Y. (1999), Kids As Informants: Telling Us What We Didn't Know or Confirming What We Knew Already, in Druin A. (ed), *The Design of Children's Technology*, Morgan Kaufman Academic Press, San Francisco.
- Schuler D., Namioka A. (eds) (1993), *Participatory design: Principles and practices*, Lawrence Erlbaum Associates, Hillsdale, NJ.
- Suzuki H., Kato H. (1995), Interaction-Level Support for Collaborative Learning: AlgoBlock - An Open Programming Language, in *Proceedings of CSCL'95*, pp. 349-355.
- Turkle S. (1984), *The Second Self: Computers and the Human Spirit*, Simon and Schuster, New York.
- Turkle S. (1995), *Life on the Screen: Identity in the Age of the Internet*, Simon and Schuster, New York (tr. it. *La vita sullo schermo. Nuove identità e relazioni sociali nell'epoca di Internet*, Apogeo, Milano, 1997).
- Turkle S., Papert S. (1992), Epistemological pluralism and the revaluation of the concrete, *Journal of Mathematical Behaviour*, vol. 11, n. 1, pp. 3-33.
- Vygotskij L. S. (1978), *Mind in Society*, Harvard University Press, Cambridge, MA (tr. it., *Il processo cognitivo*, Boringhieri, Torino, 1980).
- Wood D., Bruner J., Foss G. (1976), The role of tutoring in problem solving, *Journal of Child Psychology and Psychiatry*, 17, pp. 89-100.
- Wyeth P., Wyeth G. (2001), Electronic Blocks: Tangible Programming Elements for Preschoolers, in *Proceedings of the Eighth IFIP TC13 Conference on Human-Computer Interaction (Interact 2001)*, Tokyo, Japan, July 9-13, 2001.