

Verso una teoria delle “architecture machines”

NICHOLAS NEGROPONTE

Fonte: Journal of Architectural Education , Vol. 23, No. 2 (Mar., 1969), pp. 9-12

URL: <http://www.jstor.org/stable/1423828> .

Quando un progettista alimenta una macchina con istruzioni step by step per risolvere un problema , la soluzione che risulta è senza dubbio da attribuire all’ingegno e al lavoro del progettista.

Non appena il progettista fornisce alla macchina le istruzioni per trovare un metodo di soluzione, la paternità dei risultati diventa ambiguo.

Ogni volta che un meccanismo è dotato di un processore in grado di trovare un metodo *per trovare un metodo di soluzione*, la paternità della risposta probabilmente appartiene alla macchina.

Se prescindiamo da quest’ultimo argomento e separiamo la creatività della macchina dall’iniziativa del progettista, degradingo ad una visione pedagogica dei nostri progetti e delle nostre azioni risalente ai nostri nonni.

La macchina evolutiva

Questa discussione non riguarda macchine che per forza producono architettura, è una prefazione a macchine che possono imparare l'architettura e forse anche imparare a come imparare l'architettura. Possiamo chiamare queste macchine “architecture machine”; la collaborazione di un architetto con tali dispositivi è un dialogo tra due sistemi intelligenti - l'uomo e la macchina - che sono grado di produrre un sistema evolutivo¹.

I computer sono sicuramente impiegati formidabili; essi eseguono bene quando gli è stato detto con esattezza come fare qualcosa e possono sollevarci dalla fatica e dalla noia delle fasi più ripetitive del progettare.

Non è abbastanza? Perché chiedere a una macchina di imparare, di associare il fare con obiettivi, di essere auto-migliorativa e di essere etica²? La risposta è implicita nella domanda.

Se una macchina è in grado di essere auto-migliorativa ed evolutiva, avrà più opportunità di rendere più rilevanti le sue capacità di calcolo e di informazione.

La maggior parte dei software di progettazione sono irrilevanti, poiché sono strumenti per fare quello che già fanno i progettisti, solo più alla moda e più velocemente (anche se raramente in modo più economico). E poiché ciò che i progettisti già fanno non sembra eccellente, correremmo il rischio di stimolare un modus operandi che potrebbe rendere ancora più prolifica la cattiva architettura.

La motivazione dell’architettura computerizzata (machine assisted architecture) è duplice: in primo luogo, gli architetti hanno difficoltà nel gestire problemi a larga scala, perché sono troppo complessi, in secondo luogo, gli architetti non considerano i problemi a piccola scala, perché li considerano troppo particolari e individuali (e, per loro, banali). Come risultato di entrambi gli

¹ This issue will be discussed at length in Nicholas Negroponte's *The Architecture Machine*, Cambridge: MIT Press, late 1969. The preparation of the manuscript has been sponsored by Joint Center for Urban Studies of Harvard University and MIT: The reader should also refer to: Warren M. Brodey and Nilo Lindgren, "Human Enhancement Through Evolutionary Technology," *IEEE Spectrum*, September 1967, page 87.

² Warren McCulloch, *Embodiments of Mind*, Cambridge: MIT Press, 1965. The reader should particularly look at "Towards Some Circuitry of Ethical Robots," in that volume.

aspetti, "meno del 5% delle residenze costruite negli Stati Uniti e meno dell'1% dell'ambiente urbano è il risultato del lavoro di professionisti del progetto³. "

Nel tentativo di combattere queste carenze, i ricercatori stanno sviluppando sistemi informatici, computer grafica e servizi informatici che liberano il progettista e gli permettono più tempo per fare ciò che ama davvero.

Tali sforzi hanno esito significativo solo in un contesto dove le macchine possono imparare a essere adattabili e imparare a essere rilevanti.

Ironia della sorte, un umanesimo ambientale potrebbe essere raggiungibile solo grazie alla cooperazione fra uomo e macchine che sono ritenute dispositivi non umani - dispositivi in grado di rispondere in modo intelligente all'invisibile, individuale, costante cambiamento dei flussi di bit, che riflettono l'identità di ogni comunità, così come la coerenza della città. Se questo è vero, allora il primo problema è: può una macchina dedurre risposte da una serie di dati?

La macchina che impara

Un teorema del 1943 di McCulloch e Pitts stabilisce che un robot costruito con cicli ripetitivi di un certo carattere formale, è in grado di dedurre ogni legittima conclusione da un insieme finito di premesse⁴. Un approccio a tale facoltà è quella di aumentare la probabilità di significatività degli output (il progetto) generata da un sistema casuale o disordinato di input (i criteri). Ross Ashby ha dichiarato: "E' stato spesso osservato che qualsiasi sequenza casuale, se abbastanza lunga, conterrà tutte le risposte; nulla impedisce a un bambino di scarabocchiare: $\cos^2 x + \sin^2 x = 1$ ".⁵ Con lo stesso spirito, per parafrasare il dibattito su British Museum /scimpanzé, un gruppo di scimmie, mentre scarabocchia casualmente, può disegnare piante, sezioni e prospetti di tutte le grandi opere di architettura e farlo in un periodo finito di tempo.

Sicuramente è necessario qualche vincolo e discriminante ex ante se la macchina deve convergere a soluzioni in tempi ragionevoli. Ad esempio, un'architectural machine dovrebbe avere queste cinque componenti:

1) un meccanismo euristico, 2) un apparato di memoria, 3) un dispositivo di trattamento delle informazioni abituali, 4) un selettore di valutazione e 5) l'opportunità di dimenticare.

1) Un metodo euristico è basato su regole empiriche (o strategie) che limitano drasticamente il campo delle soluzioni. Un metodo euristico non garantisce una soluzione, tanto meno quella ottimale, ma il vantaggio è nel risparmio di tempo e nella riduzione della ricerca di alternative. L'apprendimento euristico è particolarmente rilevante per le macchine evolutive, dal momento che si presta alla personalizzazione attraverso il dialogo con l'architetto. In un'architecture machine, l'elemento euristico è probabilmente vuoto di indicazioni quando la macchina arriva nel vostro ufficio. Attraverso le indicazioni dell'architetto, il sistema operativo acquisisce un'ampia gamma di regole per gestire le informazioni. La prima volta che si pone un problema, la macchina tenta di applicare procedure attingendo a casi (o contesti) apparentemente simili. L'osservazione euristica di situazioni analoghe sarebbe il primo contributo alla soluzione di un nuovo problema.

2) Per gestire le osservazioni è necessario disporre di un meccanismo di memoria.

³ John Eberhard, "A Humanist Case for the Systems Approach," AIA JOURNAL, July 1968.

⁴ Warren McCulloch and Walter Pitts, "A Logical Calculus for Ideas Immanent in Nervous Activity," Bulletin of Mathematical Biophysics, Vol. 5, Chicago University Press, 1943, pages 115-133.

⁵ Ross Ashby. "The Design of an Intelligence Amplifier," Automata Studies, edited by Claude Shannon and I. McCarthy, Princeton University Press, 1956.

L'apprendimento a memoria è la forma più elementare di archiviazione di problemi da associare ad una risposta. Quando una situazione si ripete in continuazione, il meccanismo di memoria può ricordare la circostanza per usarla quando si incontrano situazioni simili in futuro. In architettura, tale ripetizione di sottoproblemi è estremamente frequente: progettazione di parcheggi, ascensori, impianti idraulici, ecc. Anche il meccanismo di memoria si presta alla personalizzazione.

3) A differenza del meccanismo euristico, da personalizzare, sarebbe opportuno disporre sin dall'origine di un dispositivo con un repertorio di situazioni che sono affrontate abitualmente. Negli esseri umani le risposte ripetitive diventano abitudini (alcune buone e alcune cattive), grazie a un meccanismo di condizionamento, localizzato nella memoria, che ha il compito di gestire tutte le informazioni non eccezionali, e di assisterli nel superare gli ostacoli di ogni giorno. Allo stesso modo, in un computer la codifica delle abitudini del progettista permette di rispondere agli eventi standard generati da un problema, dal meccanismo euristico o dall'apparato mnemonico. Ogni robot dovrebbe sviluppare i propri riflessi condizionati⁶. Come il cane di Pavlov, la presenza di eventi abituali innesca risposte predefinite con poco sforzo (senza richiamo a una memoria cosciente) fino al momento in cui la risposta è cancellata dalla frustrazione (o dall'evoluzione) e sarà gestita da un'altra componente del sistema.

4) Un selettore di valutazione di ogni azione. Nel modello di Skinner, il meccanismo di valutazione seleziona ogni azione in relazione a quello che il "maestro" desidera⁷. Occorre fornire a questo meccanismo l'impostazione, cioè le tecniche di simulazione così evolute da essere in grado di testare qualsiasi ambiente a prescindere dalle conoscenze del progettista. Il progetto di questo dispositivo è di fondamentale importanza perché senza selettore la cattiva architettura potrebbe farsi largo con la stessa facilità del buon design. Una selettore di valutazione deve essere in grado di contrastare la cattiva progettazione; deve valutare, o almeno osservare, confrontando obiettivi con risultati.

5) Infine, disimparare è importante quanto imparare. Il detto "la sua (del computer) incapacità di dimenticare tutto ciò che è stato messo in esso⁸ ..." è falso. Le informazioni possono assumere minore importanza nel corso del tempo fino a scomparire ed essere dimenticate. L'obsolescenza si verifica nel tempo o per irrilevanza. Una innovazione tecnologica nel settore delle costruzioni, per esempio, può rendere interi campi di conoscenze obsoleti (e, gli esseri umani, spesso odiano arrendersi all'innovazione). O le passate procedure potrebbero non soddisfare le condizioni ambientali che sono cambiate nel tempo, così da invalidare la risposta euristica, o memorizzata, o come riflesso condizionato.

Questi cinque elementi sono solo pezzi; l'intero corpo sarà un gruppo sempre mutevole di meccanismi che saranno sottoposti a mutazioni strutturali⁹, che si evolvono, il tutto sotto la

⁶ Albert Uttley, "Conditional Probability and Conditioned Reflexes," Automata Studies, edited by Claude Shannon and J. McCarthy, Princeton University Press, 1956.

⁷ B. F. Skinner, Science of Human Behavior, New York: Macmillan Co., 1953.

⁸ Arthur R. Miller, "The National Data Center and Personal Privacy," The Atlantic Monthly, Vol. 220, November 1967.

⁹ The concept of simulated evolution through bearing offspring is covered at great length in Lawrence J. Fogel, Alvin J. Owens and Michael J. Wash, Artificial Intelligence through Simulated Evolution,

direzione di un timoniere. Anche se questo non è il luogo per descrivere i dispositivi di controllo o le configurazioni hardware in dettaglio, è importante capire la disposizione generale delle parti. Architecture machine con i suoi cinque elementi è pensata per stare a stretto contatto con l'architetto, con cui deve lavorare 24 ore su 24 con capacità di calcolo e memoria locale. In una postazione remota un grande computer sarebbe collegato a tante architecture machine via telefono. Questa postazione avrebbe potenti processori e memoria estesa (per raccogliere tutti i regolamenti edilizi, le leggi i trend demografici di tutto il mondo...). L'architecture machine è connessa a questa periferica principalmente per tre ragioni:

- 1) acquisire grandi picchi di potenza di calcolo,
- 2) acquisire informazioni memorizzate,
- 3) comunicare con altri architetti e altri computer.

In altre parole, la configurazione è quella in cui molte parti, umane e meccaniche, sono comunicanti tra loro e con gli altri, mentre il consorzio nel suo insieme è in qualche modo comunicante con il mondo reale.

La macchina che vede

La comunicazione è la risposta discriminatoria di un organismo ad uno stimolo¹⁰. Se dobbiamo fare i conti con la comunicazione, oltre la retorica formale o la sintassi, dobbiamo affrontare noi stessi la versatilità di un meccanismo discriminante, l'interfaccia¹¹.

In questo caso, l'interfaccia è il punto di contatto e l'interazione tra una macchina e l' " ambiente delle informazioni ". Il punto di osservazione che ci interessa è dove i processori diventano tangenti al mondo reale in modo diretto o con la mediazione di un essere umano (che provvede al rilevamento).

Per una macchina sono necessarie tre proprietà per ricavare l'immagine di un progetto, o affrontare un problema di progettazione, o anche una cosiddetta soluzione progettuale: un evento, una manifestazione, una rappresentazione.

L'evento può essere visivo, uditivo, olfattivo, tattile, extrasensoriale o un comando.

La manifestazione misura l'evento con appropriati parametri (luminanza, frequenza, cervello-onda-lunghezza, angolo di rotazione, ecc.).

La rappresentazione è l'atto di mappare le informazioni in un contenitore compatibile con le caratteristiche di elaborazione dell'organismo ricevente¹².

Queste tre proprietà-evento, manifestazione, rappresentazione-formano l'interfaccia.

In una collaborazione con l'architectural-machine forse l'interfaccia sensoriale più rilevante è quella visiva.

Le tecniche di computer grafica sono divenute il paradigma per i sistemi di architettura assistita da

New York: John Wiley & Sons, 1956.

¹⁰ Colin Cherry, *On Human Communication*, Cambridge: MIT Press, 1957.

¹¹ Extensive research has been undertaken to establish congenial architect-machine interfaces. URBAN5 is a computer system that illustrates some of the conveniences of graphical and natural language discourse. This has been reported in Nicholas Negroponte and Leon Groisser, "URBAN5," *Ekistics*, Vol. 24, September 1967. URBAN5 is discussed at greater length in a forthcoming publication, Gary Moore (editor), *Proceedings of the First International DMG Conference* (held June 1968).

¹² Stuart M. Silverstone, *Information Manipulation for the Evolution of Physical Environments*, Cambridge: Urban Systems Laboratory, MIT, Research Report (forthcoming).

computer¹³, ma al di là di linee di input e output, punti, caratteri, l'architecture machine deve avere occhi (e orecchie e ...).

Riteniamo fantasmagorica l'idea del robot-designer, che prefigura una macchina parlante che ti vede; nella nostra cultura attuale questa idea appare sciocca o spaventosa, per i nostri figli sarà un fatto ordinario. Per Mortimer Taube è offensiva¹⁴, per Marvin Minsky è ovvia¹⁵. Oliver Selfridge sta studiando una machine vision¹⁶. La sua macchina, "Pandemonium", è in grado di osservare molte caratteristiche di un luogo e di commentarle con voci (da cui pandemonio).

Il lavoro più recente di Seymour Papert e Marvin Minsky ha ampiamente dimostrato che tali informazioni a livello locale non sono sufficienti; un buon numero di ossezioni locali (e globali) è necessario per raggiungere un adeguato livello di giudizio¹⁷.

Applicando l'osservazione di Minsky-Papert, è possibile costruire un' architectural machine visiva sviluppando un semplice dispositivo che osserverà semplici modelli¹⁸. Tale meccanismo è il preludio a macchine che un giorno si aggireranno per la città per catturare le immagini della città. In tal modo, l'architecture machine potrebbe acquisire informazioni al di là di quelle che sono memorizzate.

Inoltre, tale acquisizione dati evita la perdita di informazioni nel trasferimento dal mondo reale agli apparati del progettista, al suo cervello. Tale ricerca è un esercizio di apprendimento attraverso il vedere (apprendendo solo quegli aspetti che sono davvero visivamente rappresentabili). La macchina è finalizzata a riconoscere ciò che ha visto (usando molti strati euristici) per poi estrapolare alcune caratteristiche (probabilità, elementi comuni, scopi, modelli, ecc.) Dopo che il primo modello è riconosciuto, la macchina ne chiede un secondo e poi un terzo, fino a quando non ha considerato ad esempio 10 modelli in risposta a 10 semplici questioni. Dopo la soluzione 10 alla macchina sarà data una 11^a opzione e la macchina chiederà di generare una propria soluzione. In questo esperimento, la soluzione sarà nella forma vernacolare presentata nell'originale 10. Anche se una macchina del genere è più manierista di uno studente, l'esercizio è rilevante in quanto si inverte il ruolo abituale del computer. Attualmente, lo sforzo della ricerca è concentrato sulla macchina generatrice di forme da un dato set di criteri; per una machine architecture il problema è il

¹³ Murray Milne (editor), *Proceedings of Computer Graphics in Architecture Conference (held Spring 1968)*, New Haven (forthcoming).

¹⁴ Views that oppose the concept of machine-intelligence are extensively presented by Mortimer Taube, *Computers and Common Sense*, New York: McGraw-Hill, 1961. For further material, Hubert L. Dreyfus, *Alchemy and Artificial Intelligence*, Rand Corporation Paper, 1966, page 3244.

¹⁵ Marvin Minsky, "Artificial Intelligence," *Scientific American*, Vol. 215, September 1966. The entire issue provides material on the use of computers.

¹⁶ Oliver Selfridge and Ulric Neisser, "Pattern Recognition by Machine," *Computers and Thought*, edited by Edward A. Feigenbaum and Julian Feldman, New York: McGraw-Hill, 1963.

¹⁷ Marvin Minsky and Seymour Papert, *The Perceptron*, Cambridge: MIT Press, 1969. The book further expands on some of the myths of parallel processing.

¹⁸ Such work is being carried out by Anthony Platt, in cooperation with Seymour Papert, Leon Groisser and the author. The research is being conducted in Project MAC's Artificial Intelligence Laboratory under Ford Foundation sponsorship through MIT's Urban Systems Laboratory. The work is one of four experiments directed toward the actual construction of an architecture machine.

contrario; data una forma, generare i criteri. . . imparare dai criteri e un giorno generare nuove forme.