

L'estetica del software raccontata in questa intervista a Salvatore "Antirez" Sanfilippo



Da oltre mezzo secolo, con **l'arte generativa**, la **net art** o le **pratiche della software art**, **artisti e teorici riconoscono nel codice un medium espressivo**. Più raro, invece, è che questa riflessione emerga dall'interno della cultura open source, dove il software viene generalmente discusso in termini di architettura, prestazioni ed efficienza piuttosto che come una pratica progettuale dotata di una propria dimensione estetica. È proprio qui che si colloca **lo sviluppatore siciliano Salvatore "Antirez" Sanfilippo** (Campobello di Licata, 1977), noto nella comunità informatica come Antirez. Creatore di Redis, uno dei software open source più influenti degli ultimi vent'anni, Sanfilippo descrive spesso il proprio lavoro ricorrendo a un lessico vicino a quello dell'arte. Per lui programmare non significa soltanto scrivere codice efficiente, ma **costruire un sistema coerente di idee, in cui la ricerca**

della forma entra in relazione con la letteratura.

Salvatore Sanfilippo e il software Redis

Creato nel 2009, **Redis (REmote DIctionary Server)** è un database in-memory progettato per gestire dati che devono essere letti e scritti con estrema rapidità. Grazie alla sua struttura chiave-valore e al supporto di differenti tipi di dati, viene utilizzato come sistema di cache, motore per sessioni utente, contatori in tempo reale e strumento di comunicazione tra servizi. Oggi è uno dei software open source più diffusi al mondo e costituisce un'infrastruttura fondamentale per milioni di applicazioni.

Come nasce il software Redis di Salvatore Sanfilippo

Redis nasce seguendo **una filosofia fondata sulla sottrazione più che sull'accumulo**. Ispirandosi al principio del *Less is more*, Sanfilippo ha sviluppato il software al di fuori delle logiche imposte dalle roadmap commerciali. Una scelta che si riflette anche nel suo stile di vita: vivere tra Catania e Campobello di Licata significa rivendicare una distanza dalle dinamiche dei grandi hub tecnologici. In questa prospettiva, **il software esprime una visione nella quale semplicità, chiarezza e coerenza costituiscono valori estetici oltre che funzionali**.

Il codice come gesto artistico e politico secondo Salvatore Sanfilippo

Sanfilippo ha affermato in diverse occasioni di considerare il codice non soltanto uno strumento funzionale, ma una forma espressiva. Nella sua visione hacker del software, **la possibilità di sperimentare senza essere vincolati da finalità immediatamente utilitaristiche assume una valenza politica**, perché mette in discussione una concezione dello sviluppo

guidata esclusivamente dall'efficienza, dalla standardizzazione e dalla produttività. Emblematico è il comando *Lolwut*, introdotto in Redis 5 come una parentesi ironica all'interno di un software concepito per massimizzare le prestazioni. Ispirato a *Schotter* di Georg Nees, una delle opere fondative dell'arte computazionale, *Lolwut* genera nel terminale un'immagine composta da caratteri Unicode Braille. Una delle versioni di *Lolwut* richiama l'esperienza di Nanni Balestrini con la poesia generata da computer del 1961, mettendo in relazione il codice con una pratica creativa basata su regole, algoritmi e processi automatici. Queste scelte hanno suscitato critiche proprio perché introducevano un elemento apparentemente inutile in un software orientato all'efficienza.

Intervista a Salvatore Sanfilippo

Redis ha introdotto un approccio completamente nuovo nel mondo dei database. Qual è stata l'innovazione più significativa?

Tradizionalmente, i database usavano il disco per salvare i dati. Redis ha rotto questo paradigma iniziando a usare la memoria RAM. Questa, secondo me, è la parte artistica dell'informatica: lavorare fuori dagli schemi e sperimentare soluzioni che, a prima vista, sembrano prive di senso.

Lavori spesso da solo. Quanto conta l'autorialità nello sviluppo di un software?

Vedo la progettazione e la scrittura di un software come un processo artistico e, in quanto tale, come l'espressione di una prospettiva personale. È raro scrivere un romanzo o dipingere un quadro a quattro mani. Nella programmazione esistono progetti collaborativi, ma quelli costruiti con troppe mediazioni rischiano di perdere brillantezza. Il libro "The Mythical Man-Month" di Fred Brooks spiega bene questo concetto. Non basta mettere più persone su un progetto per finirlo prima, perché se manca una visione chiara, il

risultato è deludente. Tuttavia, trovo importante il confronto con la comunità.

I tuoi commenti nel codice sembrano rivolti a un lettore. C'è un intento letterario in questo gesto?

La programmazione è una forma di letteratura. Nell'informatica, infatti, questo modo di programmare si chiama Literate Programming. L'approccio risale agli anni '80 ed è stato inventato da Donald Knuth. L'idea è che il codice non sia solo qualcosa che deve essere eseguito, ma anche qualcosa che viene letto e interpretato da altri programmatori.

Come nasce la possibilità di un software che possa aspirare anche a una forma di bellezza, non solo alla funzionalità?

L'informatica si è sempre confrontata con la forma, proprio come l'architettura. È una disciplina progettuale e creativa allo stesso tempo. Ogni programmatore, guardando il codice di qualcun altro, si chiede se è bello o non lo è.

Quali sono gli elementi che definiscono l'estetica di un codice?

La cosa più bella di un programma è che il codice sia brillante ma semplice, così che da ogni parte emerga un'idea unitaria, i moduli abbiano senso e comunichino bene tra loro.

Cosa si perde nel software quando diventa solo business?

Molti software dovrebbero arrivare a un certo punto a essere considerati compiuti, con una manutenzione necessaria ma senza continuare ad aggiungere funzioni. Invece l'industria tende a far crescere continuamente i prodotti, aggiungendo complessità. Più un sistema diventa complesso, più diventa fragile e perde quella bellezza interna che aveva all'inizio.

Pensando all'etica del "Less is More", di cui tu sei sostenitore, qual è il valore di togliere invece che

aggiungere?

Less is More è un mantra nell'informatica. Rinunciare alla complicazione è fondamentale, ma purtroppo questo principio si è perso. Oggi vediamo programmi semplici che occupano quantità enormi di memoria e sistemi in cui anche le funzioni più basilari diventano difficili da usare proprio a causa della complessità accumulata.

Redis può essere considerato un sistema di idee oltre che un prodotto tecnico?

Quello che resta di Redis non è solo il codice sorgente, ma le idee contenute in quel codice. Molti programmatori hanno assimilato alcuni suoi principi concettuali, trasformandoli in modelli da riutilizzare nei propri software. In Cina, alcuni programmatori lo associano alla scrittura cinese per la sua semplicità, equilibrio ed eleganza.

Se il software è modificabile e replicabile, ha senso parlare di unicità?

La cultura del software è naturalmente contraria all'idea di unicità, perché il codice nasce per essere modificato e riutilizzato, anche se conserva la sua storia. Io pubblico quasi tutto il mio software con licenze molto permissive, che permettono a chiunque di modificarlo; questo favorisce la diffusione delle idee. Allo stesso tempo, un frammento di codice può acquisire un valore particolare. Se un autore come John Carmack stampasse, firmasse ed esponesse un frammento significativo del codice di Doom, quel pezzo potrebbe diventare un'opera unica.

Non ti interessa l'idea di essere riconosciuto come unico autore di quel codice?

No, non mi interessa particolarmente. Il software è sempre stato una sottocultura. Il vero problema è che in Italia manca il riconoscimento sociale di questa disciplina; se sei un

avvocato, ti riconoscono il ruolo. Se sviluppi software, anche a livelli molto alti, pensano che aggiusti le stampanti o che installi Windows.

Si può quantificare l'impatto che Redis ha avuto nel software?

Pochissimi sistemi non usano Redis. L'impatto si misura anche economicamente: Redis muove miliardi di dollari all'anno ed è uno dei software più scaricati al mondo. Su Docker Hub è stato per molto tempo il primo per numero di download.

A cosa stai lavorando ora?

Sto lavorando su Redis in ambito intelligenza artificiale, in particolare sulla ricerca semantica: associare immagini simili, suoni simili, o cercare immagini tramite descrizioni testuali. Si chiama "ricerca vettoriale per similarità".

Hai scelto di vivere a Catania, lontano dai grandi hub tecnologici. Che libertà hai trovato in Sicilia?

La scelta più logica sarebbe stata andare negli Stati Uniti per fondare un'azienda, ma per me l'Europa offre un sistema più equo, con ammortizzatori sociali e cure gratuite, anche se tende a scoraggiare un po' la creazione di aziende. Ho scelto di vivere a Catania anche per la socialità, il clima e la cultura, non vivo bene in contesti troppo rigidi culturalmente. Inoltre, è stata la patria di una delle prime sperimentazioni di arte digitale dentro un centro sociale chiamato Lauro, da cui è nato Freaknet, con esperienze di controcultura digitale, hacking e arte elettronica.

Vivere in Sicilia ha influenzato il tuo lavoro?

Secondo me esiste un modo siciliano di pensare alle cose: c'è molta attenzione ai sentimenti, alle relazioni umane, alle contraddizioni e anche a quel rapporto costante con la morte, ma quando si tratta di fare le cose emerge un pragmatismo che porta a cercare soluzioni semplici e immediate. È un

atteggiamento che riconosco anche nel mio modo di progettare il software.