



Per un Museo del Codice Sorgente

S. Penge

Sommario

Codexpo è il primo Museo del Codice Sorgente, quel testo magico che vive dentro computer e cellulari, controlla i satelliti e fa funzionare il Web. Un testo scritto in linguaggi diversi, da persone diverse, con stili diversi.

Codexpo non è un museo dell'hardware, ma del software. Inoltre, non mira a raccogliere e esibire i programmi già pronti all'esecuzione, bensì i programmi nel loro stato originale.

Un museo che per la prima volta mostra l'anima nascosta dei computer: il codice sorgente.

Abstract

Codexpo is the first Museum fully dedicated to the Source Code, the magical text that lives inside computers and mobile phones, that controls satellites and makes the Web work. A text written in one from many different languages, by different authors and with different styles.

Codexpo is not an hardware museum, it is about software. Moreover, it doesn't aim to collect and show ready-to-run programs, but rather programs in their original state. A museum that for the first time shows the hidden soul of computers: the source code.

Keywords: Source code, Museum, Programming languages, Digital humanities, Open source, Linguistics, Critical code



1. Introduzione

Siamo tutti abituati a usare in maniera trasparente i computer e i programmi che li fanno funzionare; così finiamo per dare per scontato che i computer si comportino sempre in un certo modo, e che la colpa – o il merito - di questi comportamenti siano della macchina. Raramente pensiamo che dietro ogni programma ci sia un *codice sorgente*, e che questo codice sia *stato scritto da qualcuno, in qualche momento, per un certo scopo*.

Mostrare, spiegare, raccontare la storia di questo tipo particolare di testo servirebbe allora a renderci più consapevoli dei processi di scrittura e lettura che ci sono dietro, a restituire valore all'opera creativa – e alla responsabilità – dell'uomo che li ha creati [1].

Programmare non è mettere in sequenza 0 e 1, ma *raccontare una storia, inventare attori e scenari, immaginare finali alternativi*. Certo vanno usati strumenti formali adeguati – i linguaggi di programmazione, che vanno appresi seriamente e praticati per anni - ma ciò che è importante è la visione che lega le parti con il tutto in una narrazione coerente e convincente. Senza questa visione, si rischia di limitarsi a copiare e incollare frammenti di codice altrui.

E senza una comprensione dell'importanza di questa parte creativa del lavoro di chi scrive codice si rischia di fraintendere la complessità della professione del programmatore e di ridurne l'attrattiva per i giovani. Peggio ancora, si rischia di crescere una generazione di cattivi programmatori da batteria, seguaci del metodo copia-e-incolla, capaci solo di allineare righe di codice senza farsi domande.

Ma che si voglia intraprendere un percorso professionale da sviluppatore di codice oppure limitarsi ad usare dispositivi digitali in maniera attiva e non passiva, la consapevolezza della storia, delle scelte, delle sfide, delle differenze che costituiscono la sostanza di questo oggetto culturale così importante è oggi fondamentale per tutti.

Dalla constatazione di queste esigenze nasce il progetto dell'associazione di promozione sociale Codexpo.org.

In estrema sintesi, noi pensiamo che:

1. Il codice sorgente sia una parte importante della cultura della fine del secondo millennio e presumibilmente di tutto quello appena iniziato. Per questo va raccolto, conservato, selezionato ed esposto [2]

2. Progettare e realizzare codice sorgente sia un'attività umana piuttosto nuova, ma che assomigli alle altre attività di scrittura di testi che pratichiamo da millenni. Per questo va studiata, capita e insegnata

3. Contro chi agita lo spauracchio degli 'algoritmi padroni' e della 'cattiva intelligenza artificiale', occorra costruire un umanesimo e un rinascimento digitali. Per questo proponiamo di ripartire dall'umano: quello che sta dietro e intorno alle macchine.

2. Il progetto

Codexpo.org (<http://codexpo.org>) è il nome di un'Associazione di Promozione Sociale che ha come missione principale quella di costruire il primo museo non dell'hardware, ma del software. Un museo che per la prima volta porta alla luce l'anima nascosta dei computer: il codice sorgente. Un museo reale, fisico, in cui il visitatore possa toccare con mano il codice sorgente – accompagnato da un'anteprima su web, che serve a preparare alla visita e a diffondere il progetto stesso.

Un museo della *letteratura digitale*, cioè dei programmi nel loro stato originale. Il termine "letteratura digitale" è usato qui in maniera provocatoria in tempi di e-book e AI: non la letteratura tradizionale resa digitale, né la letteratura scritta da software, ma il campo infinito di *testi scritti da umani negli innumerevoli linguaggi di programmazione*. Una sconfinata letteratura di cui non si intuisce nemmeno l'esistenza, di cui poco si sa e che raramente è studiata come tale.

In una prospettiva che vede i sorgenti come *testi scritti da autori in determinate lingue storiche e in determinati contesti*, l'esposizione vuole fornire un'informazione ricca e rigorosa, innanzitutto sui linguaggi stessi: ne esistono migliaia, da quelli di uso generico a quelli specifici di un dominio, da quelli che ricalcano il lessico e la sintassi di lingue naturali a quelli che fanno uso esclusivamente di segni di punteggiatura [3]. Linguaggi e paradigmi, nella loro dipendenza ed evoluzione storica, formano un patrimonio che è spesso sconosciuto persino agli studenti di Informatica.

C'è poi una connessione stretta tra i linguaggi di programmazione e la lingua madre dell'autore. Ogni codice sorgente è connotato diversamente per il lessico scelto nel nominare variabili, funzioni e procedure, per lo stile e la densità dei commenti. Non a caso, Donald Knuth, uno dei più grandi teorici della programmazione, proponeva negli anni '80 la *Literate Programming* [4] cioè una pratica di scrittura/lettura del codice sorgente a partire dalla sua controparte in linguaggio naturale. Proprio per questa centralità della lettura [5], l'esposizione è fortemente legata al concetto di *Open Source* [6] come precondizione: per verificare, modificare, imparare dagli esempi è necessario poter leggere liberamente il codice.

Un aspetto sfuggente e spesso frainteso è quello della *qualità* del codice sorgente, e in generale della valutazione anche estetica. Se anche il codice sorgente viene *letto*, indipendentemente dalla sua esecuzione da parte della macchina, è naturale che vengano poste in continuazione questioni di qualità, di stile, di estetica, oltre a quelle di efficienza e correttezza.

Per tutti questi motivi, accanto alla schede informative che spiegano origine, significato e contesto storico, verranno installate opere multimediali, digitali e non, che puntano a rendere evidenti le parentele del codice sorgente con le altre tipologie di testi più conosciute, come sceneggiature teatrali e brani musicali. Verranno esposti codici che nascono come poesie [7] o opere artistiche [8] e codici che portano l'efficienza ai più alti livelli a prezzo dell'illeggibilità, fino alle gare di codice offuscato, che fanno riferimento ad un'estetica alternativa a quella classica [9].

Il Museo partirà dunque dagli *oggetti* (i programmi e i linguaggi), ma presenterà anche le *azioni* della scrittura e lettura di un testo e richiamerà l'attenzione sulle *persone* che scrivono codice e sugli inventori dei linguaggi, da Grace Hopper a Linus Torvalds, da Niklaus Wirth a Larry Wall.

Museo inteso non solo come spazio di conservazione e/o catalogazione di frammenti significati di codice sorgente (come l'ambizioso progetto Software Heritage [10] dell'INRIA di Parigi), ma come spazio di continua *riorganizzazione*, come spazio di *formazione* e di *comunicazione*.

In attesa di realizzare il Museo, stiamo cominciando a organizzare delle mostre temporanee in giro per l'Italia, in cui l'esposizione del codice si unisce ai seminari, alle performance e ai laboratori in cui i partecipanti potranno vedere dal vivo come viene creato il codice sorgente.

3. I destinatari

A chi può essere rivolta un Museo così specialistico? Non rischia di interessare solo un piccolo manipolo di addetti ai lavori? Pensiamo di no.

Il Museo è pensato per soddisfare esigenze diverse:

- curiosità generica per un mondo sotterraneo e sconosciuto, ma di cui a volte i media fanno balenare sprazzi (Matrix, gli algoritmi cattivi dell'IA);
- interesse per il background teorico e per l'humus che ha consentito la nascita di migliaia di linguaggi diversi;
- passione da collezionista per i reperti storici, usati come lente per ricostruire la storia della cultura digitale [11];
- bisogno di ripensare una didattica del coding e del pensiero computazionale da un punto di vista storico e più ampio;
- attenzione per un'operazione culturale che estende il campo degli studi della linguistica testuale ad un campo totalmente nuovo.

Interessi che possono convivere o motivare persone diverse.

La promozione del Museo farà anche leva su una motivazione più forte: il numero di programmatori usciti dalla scuola e dall'Università (comprendendo in questa voce tutti i mestieri della programmazione, che sono tanti e diversi) è insufficiente rispetto ai bisogni del mercato. Contemporaneamente, e questa purtroppo non è una buona notizia, quelli che mancano sono i *buoni* programmatori: creativi, flessibili, capaci di apprendere, con una visione estesa, con una cultura vasta, consapevoli. E mancano le *ragazze*, che – quando ci sono – si dimostrano invece sempre all'altezza se non più creative dei colleghi maschi. Il Museo si pone anche l'obiettivo di *orientare e motivare i giovani* diplomati/e e laureati/e a seguire questa strada professionale con maggior

consapevolezza degli orizzonti di una disciplina che dopo cinquant'anni dalla sua creazione è ancora in fasce.

4. Conclusione

In definitiva, in nostro obiettivo è costruire un Museo se non popolare, sufficientemente aperto e flessibile da poter catturare utenti che non sanno nulla di programmazione insieme a professionisti, studenti e docenti, ricercatori ed education policy maker etc.

Il visitatore, entrato nel Museo con un immaginario pieno di circuiti stampati e codici binari, deve passare attraverso una sorpresa dopo l'altra e uscire con il bisogno di ripensare le sue categorie duali (scientifico/umanistico, meccanico/umano, logico/artistico) e con un desiderio di approfondimento ulteriore che potrebbe portarlo lontano.

E questo è l'augurio che facciamo a tutti.

Bibliografia

1. Lévy, P. (1982) *De la programmation considérée comme un des beaux-arts*, Paris, Editions La Découverte
2. Unesco Paris Call <https://unesdoc.unesco.org/ark:/48223/pf0000366715.locale=fr> ultimo accesso febbraio 2019)
3. Esoteric Languages list https://esolangs.org/wiki/Language_list (ultimo accesso febbraio 2019)
4. Knuth, D.E. (1984) "*Literate programming*", *The Computer Journal*, 27, Jan 1984, <http://www.literateprogramming.com/knuthweb.pdf> (ultimo accesso febbraio 2019)
5. Deimel, L. E., Naveda, J. F. (1990) *Reading computer programs*, Carnegie Mellon University, https://resources.sei.cmu.edu/asset_files/EducationalMaterial/1990_011_001_15824.pdf (ultimo accesso febbraio 2019)
6. Spinellis, D., (2003) *Code Reading: The Open Source Perspective*. Addison Wesley.
7. Hopkins, S. (1992), *Camels and needles: computer poetry meets the Perl programming language*, Usenix Winter 1992 Technical Conference, http://www.digitalcraft.org/iloveyou/images/Sh.Hopkins_Perl_Poetry.pdf (ultimo accesso febbraio 2019)
8. Cox, G., McLean, A., Ward, A. (2001) "The Aesthetics of Generative Code", *Proceedings of Generative Art*, <http://generative.net/papers/aesthetics/> (ultimo accesso febbraio 2019)
9. Mateas, M., Monfort, N. (2005) "*A Box, Darkly: Obfuscation, Weird Languages, and Code Aesthetics*", *Proceedings of the 6th Digital Arts and Culture Conference*, IT University of Copenhagen, 1-3 Dec 2005, pp. 144-153 http://www.nickm.com/cis/a_box_darkly.pdf (ultimo accesso febbraio 2019)

10. Software Heritage Project, <https://www.softwareheritage.org/> (ultimo accesso febbraio 2019)
11. Montfort, N., Baudoin, P., Bell, J., Bogost, I., Douglass, J., Marino, M.C., Mateas, M., Reas, C., Sample, M., Vawter, (2012) *10 PRINT CHR\$(205.5+RND(1)); : GOTO 10*, MIT Press, <https://10print.org/> (ultimo accesso febbraio 2019)

Biografia

Stefano Penge, laureato nell'86 in Filosofia della Scienza, si è sempre occupato della ricerca e dello sviluppo nel campo del software e dei linguaggi educativi, delle piattaforme e-learning e degli strumenti AI. Su questi temi ha tenuto corsi in presenza e a distanza presso diverse Università e ha scritto alcuni volumi.

Dal 2000 ha aderito al modello Open Source e dal 2011 si occupa di Open Data.

Nel dicembre 2017 ha creato l'Associazione di Promozione Sociale Codexpo.org, con l'obiettivo di realizzare il primo Museo del Codice Sorgente e di creare e diffondere una cultura interdisciplinare intorno alle attività di progettazione e scrittura del software.

Codexpo.org <http://codexpo.org>

Email: stefano.penge@gmail.com