

f/oss, considerazioni sulle motivazioni

A.Cammozzo
Rev.1 18.02.06
Rev.2 18.03.06

Reflexive identity

Il concetto di *reflexive identity*¹ esprime la reciproca influenza esercitata dai soggetti di una comunità nei confronti delle proprie istituzioni collettive e viceversa. Una sorta di circolo in cui l'intervento del singolo nello spazio negoziale costituito dalla comunità induce negli altri una diversa visione del mondo e quindi dello spazio comunitario stesso, e quindi agisce sulle regole stesse che la collettività impone al singolo, innescando ulteriori retroazioni.

Il concetto di *reflexive identity* è stato proposto come sintesi delle motivazioni che spingono al produrre *free software*. In particolare rappresenta bene gli aspetti di senso di identità e di incentivi sociali, assorbendo gli aspetti efficaci dei modelli sociali rappresentati dalle *community of practice*, le *epistemic communities* e di *gift economy*. Trascurati dal modello restano la creatività e l'apprendimento, che però sono considerati tra i primi motivi per partecipare a progetti f/oss secondo alcune indagini².

Questi aspetti possono acquistare una propria dimensione nell'ambito della *reflexive identity* se si vorranno inserire i seguenti elementi nel ciclo di produzione di identità tra il soggetto e il suo ambiente sociale:

- 1) l'esistenza di uno *spazio* all'interno del quale il software svolge un ruolo esplicito di regolamentazione, e che incorpora in questo la conoscenza che i programmatori vi trasmettono.
- 2) l'interazione con questo *ambiente* induce necessariamente un *apprendimento*, in quanto l'ambiente cristallizza conoscenza. La programmazione è quindi il modo in cui la conoscenza *aderisce* all'ambiente.
- 3) la produzione di software, così come tanti altri tipi di opere più o meno esplicitamente cooperative, è espressione creativa di una collettività e non solo l'opera di un singolo.

Code is law

Il codice come legge è una lettura proposta da Lessig³ nella sua analisi delle regolamentazioni del comportamento nel cosiddetto *cyberspace*. Vengono identificati quattro vincoli nel processo di *behaviour regulation*:

1. la legge (*law*) che agisce attraverso la minaccia di sanzioni
2. le norme sociali (*social norms*), ovvero le aspettative su come ci si deve comportare
3. il mercato (*market*) il cui strumento di regolamentazione è il prezzo
4. l'"architettura" (*architecture*) ovvero il modo in cui il mondo è costruito, e i limiti vincoli che

1 - Francesco Rullani, *The debate and the community: the 'reflexive identity' concept and the FLOSS community case*, Working Paper No. 2005-18, Laboratory of Economics & Management - Sant'Anna School of Advanced Studies, Pisa, Italy: (July 5) 2005.

2 - Karim R. Lakhani, Robert G Wolf "Why Hackers Do What They Do: Understanding Motivation and Effort in Free/Open Source Software Projects" 2003

- Benno Luthiger, Fun and Software Development,

https://www.foss.ethz.ch/people/lbenno/BLuthiger_Fun_SoftwareDevel_OSS2005.pdf

3 - Lawrence Lessig, "The Laws of Cyberspace" Draft 3, 1998

ciò impone necessariamente al comportamento umano.

Nell'applicare il concetto di architettura a una realtà immateriale, Lessig le attribuisce necessariamente le qualità di uno *spazio*.

Le caratteristiche di questo spazio sono:

1. elevatissima *plasticità*, con pochi vincoli tecnici
2. estrema rilevanza *dell'architettura* rispetto agli altri mezzi di regolazione
3. potenziale *socialità*
4. percettivamente *reale*
5. spesso si confonde con lo spazio reale e lo pervade

Per il programmatore che lo frequenta buona parte delle ore della giornata (e forse anche della notte) ciò che si apre dietro a una tastiera e un video è uno spazio non meno concreto di quello dietro alla porta di casa.

L'aspetto più fertile della metafora si svela proprio secondo la chiave di lettura di Lessig della *regulation*: lo spazio digitale offre possibilità di interazione limitato unicamente da pochi vincoli tecnici e dalla creatività di chi lo progetta. La possibilità di *programmarlo* significa agire non meno di quanto potremmo fare sul paesaggio urbano con fondi illimitati e formidabili mezzi tecnici.

Nello spazio di Internet l'architettura ha particolare rilevanza, visto che, ad eccezione dalle limitazioni tecnologiche più estreme, l'ambiente è integralmente definito dal codice (*code*) che lo costituisce: protocolli, software, configurazioni che implementano le politiche.

Ma chi può intervenire nella regolamentazione nel ciberspazio? La maggior parte dei soggetti che ne fanno uso possono intervenire sul suo contenuto ma non direttamente sull'architettura: qualcuno può consapevolmente partecipare al processo di formazione delle *leggi* (es. pedopornografia in Internet) partecipando alla vita politica; altri possono senz'altro contribuire all'elaborazione delle *norme* sociali di comunità ristrette cui appartengono (chat, mailing lists); tutti partecipano al *mercato* con le regole sue proprie, ma ben pochi sono nelle condizioni di poterlo influenzare anche minimamente, specie data una sua spiccata tendenza al monopolio.

I programmatori, in quanto soggetti in grado di regolamentare *l'architettura*, sono relativamente pochi, caratterizzati dal disporre di conoscenze tecniche, di capacità innovativa e di incontrare il consenso⁴ altrui. Scrivendo programmi è possibile incidere in modo anche drastico nell'architettura anche senza essere un soggetto influente del mercato o della politica. Possiamo immaginare cosa rappresenti l'attrattiva della potenziale interazione con l'architettura dello spazio in cui si "vive"⁵.

La creazione di questo ambiente negoziato condiviso si ripercuote sull'identità di chi vi partecipa con il meccanismo della *reflexive identity*.

Come nello spazio fisico, lo spazio digitale ospita l'oggettivizzazione della capacità di influire sull'ambiente attraverso la creatività individuale e le relazioni che intercorrono tra individui: diviene uno *spazio performativo* teso alla produzione di realtà. La produzione di programmi f/oss è una di queste *performances* creative più o meno collettive, in cui il dialogo e la formazione del consenso è parte fondamentale per il successo della *performance*.

La presenza pervasiva del software negli oggetti che popolano lo spazio reale e che ne condizionano il comportamento fa cadere la barriera tra *cyberspazio* e spazio convenzionale, per cui il potere di agire sul codice è un potere che riguarda sempre più aspetti del vivere quotidiano, anche per chi è inconsapevole che questo è letteralmente imbottito di codice.

4 il consenso non è sempre richiesto: le architetture possono essere imposte attraverso il mercato.

5 Ted Nelson [computer lib] parla di un programmatore che costruì nel modo più confortevole ed accogliente possibile l'ambiente nel quale era convinto si sarebbe trasferito dopo la morte.

Apprendimento

L'aspetto che sembra di più sfuggire alla sintesi sembra quello dell'apprendimento: è vero che scrivere codice per imparare a farlo ben rientra nella concezione della programmazione come artigianato, che richiede un apprendistato, un tirocinio. Tuttavia questo argomento non pare così forte da fondare una motivazione così sentita, specie da parte di programmatori non principianti.

Una chiave di lettura possibile è quella dell'apprendimento che viene dall'interazione con uno spazio organico considerato come ambiente *intelligente* nel quale il soggetto è immerso (fino a confondersi) attraverso il quale l'apprendimento si esplica. O meglio, è l'intelligenza stessa una proprietà dell'ambiente. Come spiega il geografo Thift⁶

[...] *intelligence is not a property of an organism but of the organism and its environment.*

[...] *I want to argue that the world consists of a series of 'intelligencings', to use a rather clumsy phrase, intelligencings which vary substantially in their reach and understanding and interaction,*

[...] *Environments, in turn, can be thought of as a myriad of 'external physiologies' that have been adapted to act in roles as different as substitute or accessory organs, means of communication, or even microclimates. We now know that this process of constant bioturbation is a key element of evolution. Second, and following on, such a conception of organism has an explicit spatiality. Intelligence is a dynamic map of the way in which particular bodies are constructed. Different entities construct their bodies differently using different means of becoming and different locational anchors: for example, 'animals' can be foraging herds, or migrating flocks or hunting carnivores, each of which has their own distinctive geographies which are a part of what they are, including at what level of aggregation it becomes sensible to talk about a definable entity (Lulka 2004). Then, third, intelligence is about the capacity to lay out territories of intelligibility, environments which are predictable but which can also compel knowledge, can instruct, can teach, can make all manner of requests for significance. Environments are more than means of testing therefore. They are means of learning, of in-forming, if you like.*

Programmare un ambiente digitale diviene appunto plasmare un ambiente complesso e ricco di contenuti cognitivi in continuo interscambio con il contesto sociale che vi si immerge. Condividere codice significa condividere conoscenza, mezzi di trasmetterla, stimoli per suscitare.

*The reason that these distinctively human differences are so important is because it becomes possible to 'learn not just from the other but through the other' (Tomasello 1999, 6) with the result that cognitive resources can be pooled and elaborated in ways that other species are not able to achieve. In other words, through a special kind of intelligencing, **learning sticks** and is able to be projected forwards in time.*

Il modo in cui la conoscenza e l'apprendimento *aderisce* all'ambiente è appunto la programmazione, che ad esso trasferisce il suo comportamento. Saper leggere il codice significa conoscere il comportamento *a priori* e quindi esercitare nei suoi confronti un potere.

Creatività e identità

Un paper di Fischer /Giaccardi / Eden / Sugimoto / Ye⁷ ci apre una chiave di lettura su riflessività e creatività sociale, collocando il free software nel contesto della produzione creativa come frutto di una collettività che interagisce e che continua a riplasmare un contesto cognitivo in modo originale,

6 - Nigel Thrift, *From born to made: technology, biology and space*, 2005

7 - Gerhard Fischer¹, Elisa Giaccardi, Hal Eden, Masanori Sugimoto, and Yunwen Ye: *Beyond Binary Choices: Integrating Individual and Social Creativity*, Contribution to the International Journal of Human-Computer Studies (IJHCS) Special Issue on Creativity (eds: Linda Candy and Ernest Edmond)

espressivo e soggetto a valutazione/apprezzamento sociale. La creatività in definitiva è un fatto sociale.

*To support social creativity, situations need to be sufficiently **open-ended** and **complex** that users will encounter new, unpredictable conditions, and will eventually experience breakdowns [Schön, 1983]. As any professional designer knows, breakdowns—although at times costly and painful—offer unique opportunities for reflection and learning, underscoring the importance of the **back-talk** of situations [Fischer et al., 1998].*

Il codice del software f/oss, secondo gli autori, rappresenta quel tipo particolare di oggetto liminale (*boundary object*) che consente e catalizza l'esternalizzazione della conoscenza, diviene *knowledge broker* e in questo modo stimola la creatività collettiva.

Boundary objects should be conceptualized as evolving artifacts that become understandable and meaningful as they are used, discussed, and refined [Ostwald, 1996]. For this reason, boundary objects can act as “reminders” that trigger relevant knowledge, or as “conversation pieces” that ground shared understanding, rather than as containers of knowledge. It is the interaction around a boundary object, not the object itself, that creates and communicates knowledge.

La conclusione è quella di suscitare delle proprietà riflessive ad intere comunità:

The efforts to create alliances between information technology and creative practices are examples for creating reflective communities.

Altrove⁸ si considera come non solo il free software sia un vettore di creatività collettiva, e nemmeno solo altri oggetti esplicitamente e volutamente cooperativi come Wikipedia⁹. La naura della rete tende a confondere sempre più il confine tra gli oggetti, e, benchè si tenti di continuare ad attribuire loro i tradizionali attributi dell'opera d'ingegno, la proprietà si diluisce sempre più e ciò che acquista maggior rilevanza è il contenuto, anche se valorizzato in nuovi contesti.

Esempio emblematico è la pagina di risultati del motore di ricerca, i cui contenuti non appartengono al motore stesso, e nella loro aggregazione offrono un indice di una conoscenza che nessuno ha previsto, organizzato né aggregato.

I contenuti attribuiti a una persona da un motore di ricerca sono un esempio di conoscenza individuale riflessa con una tale forza nell'ambiente da contribuire a costuire l'identità dell'autore. I risultati del motore di ricerca di fatto *sono* l'identità della persona nello spazio della rete.

Ad ulteriore conferma di questa sovrapposizione tra identità e la sua rappresentazione nei motori di ricerca può essere il desiderio di proiettare una identità diversa su fatti o persone attraverso le cosiddette *google bombs*¹⁰, meccanismi che hanno come scopo l'alterazione dell'ordine “naturale” di *ranking* delle pagine. Emblematico l'esempio della celebre google bomb che ha associato la *biografia* del presidente George W. Bush con le parole “miserable failure”. E' stato usato in questo caso il potere di alterare l'architettura, il comportamento previsto, per proiettare degli attributi negativi *sull'identità* di qualcuno.

Software e corpo

Il rapporto tra identità e codice va oltre lo spazio della rete e si svolge quindi in due ambiti: attraverso l'interazione con un ambiente densamente popolato di oggetti attivati da software (la pervasività dei microprocessori ha investito ogni sfera della vita) e, per i programmatori, la scrittura di codice.

8 - Alberto Cammozzo, *Pirateria e naufragio dei media*, 06/05 <http://homes.stat.unipd.it/mmzz/Papers/media-0605.pdf>

9 Enciclopedia cooperativa: <http://www.wikipedia.org>

10 Wikipedia: http://en.wikipedia.org/wiki/Miserable_failure

L'attività di scrittura del codice va vista, in questa prospettiva, come una attività di scrittura e di evoluzione stessa della scrittura. Come detto da Nigel Thrift e Shaun French¹¹

The expansion of humanity beyond bodies has taken place in two ways, as a result of the invention of writing and then print, and as a result of the invention of various machines; line-by-line instructions and rude mechanicals. 'Software' and 'hardware'.

Possiamo quindi vedere l'attività di scrittura del codice come la prosecuzione dell'attività di espansione delle possibilità di *azione* del corpo iniziata attraverso la costruzione di semplici attrezzi, poi attraverso la progettazione di macchine complesse.

Questa estensione del *corpo* si fonde con l'estensione delle possibilità di *comunicazione*, passata attraverso la parola, la scrittura, la stampa, per giungere alla rete potenziata dai motori di ricerca

L'interazione con il software riguarda *azioni* compiute da *oggetti* che sono immersi nello dello spazio reale: il software, nella sua pervasività, diviene così l'oggettivizzazione di relazioni:

Software is more like a kind of traffic between beings, wherein one sees, so to speak, the effects of the relationship. What transpires becomes reified in actions, body stances, general anticipations.

Anche se spesso non percepita direttamente, la possibilità di interazione con l'ambiente popolato di oggetti interattivi ha delle conseguenze sull'identità e sull'immagine di se. Questo risulta immediatamente percepibile quando l'oggetto tecnologico si rompe o il software non funziona come ci attendiamo e la relazione con l'oggetto diventa fonte di rabbia e frustrazione, per la nostra inettitudine a interagire con l'oggetto o per l'errore di chi ne ha programmato il comportamento.

Creative output e identità

Alex Golub, in *Copyright and taboo*¹², partendo da un episodio di “anima rubata” mette in relazione le leggi sul copyright statunitensi e la letteratura tradizionale sui taboo melanesiani:

*Here, I'll argue that copyright and taboo are alike because they are both responses to the same existential predicament: **the fact that our identities and senses of self are always already rooted in the inevitableness of our bodies even as they exceed our corporeality.** In realm of taboo, this troubling confusion is figured in terms of the body while copyright figures this dilemma in terms of an individual's creative output.*

Poco oltre,

And just as the bodies of others have become separated from them so as to become part of ours, bits of our bodies such as fingernails, hair, feces, urine, blood can be separated from us and come into the possession of other people. We have issues about all of this—and it is these issues which Valeri¹³: takes as central to notions of taboo.

While Valeri sees the flow of identity through the body as a source of anxiety, Marilyn Strathern sees it as the building block of a distinct Melanesian sociality. She has famously argued that where “the west” sees individuals whose interaction creates social relationships, “Melanesians” see social relationships whose interactions create individuals.

Prosegue Golub confrontando questa visione del mondo con quella dei giocatori dei MMOGS (*massively multiplayer on-line videogames*), realtà virtuali altamente articolate che hanno l'aspetto di veri e propri mondi paralleli alla cui costruzione i partecipanti contribuiscono attivamente.

A Melanesian viewpoint helps us see the ways in which self-making involves a steady continuum of

11 - Nigel Thrift, Shaun French *The automatic production of space*, Transactions of the Institute of British Geographers Volume 27, Issue 3, Page 309-335, Sep 2002

12 - Alex Golub, *Copyright and Taboo*, in Kely, Christopher M. "Culture's Open Sources: Software, Copyright, and Cultural Critique", Anthropological Quarterly - Volume 77, Number 3, Summer 2004

13 Valeri, Valerio. 1999. *The forest of Taboos*, Madison: University of Wisconsin Press

both physical and non-physical concerns rather than a bright and clear boundary between “the real world” and “an imaginary one.”

Il programmatore che contribuisce a costruire l'architettura del cyberspazio lega molta importanza al codice che scrive perchè è in qualche modo parte di se, specie (per il f/oss) se il sorgente è visibile. Inoltre questo *flusso di identità* contribuisce alla costruzione del se. Compiere un'opera è in parte completare se stessi.

Perchè si scrive

In conclusione, dato che abbiamo associato continuamente l'attività di programmazione con quella creativa tipicamente autorale, può essere interessante affidarci per un attimo all'analisi che un autore illustre ha fatto dell'attività di scrittura.

Primo Levi, nell'*Altrui mestiere*¹⁴, si pone il problema di *perchè si scrive*.

I motivi che delinea non siano lontani da quelli tradizionalmente attribuiti ai programmatori, in particolare di software libero.

[...] Mi sembra che si possano configurare almeno nove motivazioni, e proverò a descriverle; ma il lettore, sia egli del mestiere o no, non avrà difficoltà a scovarne delle altre. Perché, dunque, si scrive?

1) *Perché se ne sente l'impulso o il bisogno. [...]*

Questa motivazione risponde alla necessità creativa più profondamente artistica, e difficilmente in senso stretto si può applicare all'attività di programmazione. Un programma ha pur sempre uno scopo. Se però per *bisogno* si intende la necessità di avere un programma che svolga quello scopo, vale altrettanto bene per la programmazione.

2) *Per divertire o divertirsi. [...]*

E uno dei motivi principali anche per un programmatore, come accertato da apposite indagini.

3) *Per insegnare qualcosa a qualcuno. [...]*

Vale quanto detto sugli spazi intelligenti e la codifica delle relazioni. Tuttavia raramente l'attività di programmazione viene svolta espressamente per insegnare come un programma va fatto

Nel caso di programmi scritti con sotterranei intenti didattici può valere quanto precisato da Levi:

A meno di rare eccezioni, come il Virgilio delle Georgiche, l'intento didattico corrode la tela narrativa dal di sotto, la degrada e la inquina: il lettore che cerca il racconto deve trovare il racconto, e non una lezione che non desidera.[...]

Ai programmatori di una certa scuola non può non venire in mente il paragone con “The Art of Computer Programming” di Don Knuth.

4) *Per migliorare il mondo. [...]*

Suona subito affine lo sforzo di Stallman ”to make the world a better place”. Levi tuttavia diffida degli eccessi di idealismo, specie se abbinati a forti volontà.

5) *Per far conoscere le proprie idee. Chi scrive per questo motivo rappresenta soltanto una variante più ridotta, e quindi meno pericolosa, del caso precedente. [...]*

Un programma non rappresenta di per se un'idea, ma non sono rari i programmi *proof of concept* scritti per dimostrare che una data cosa si può fare, e per lanciare un'idea che se considerata valida può essere estesa e ampliata. I programmi più significativi rappresentano una certa *visione del*

14 Levi, Primo. *L'altrui mestiere*. Einaudi, Torino 1985

mondo, se non altro per il modo in ci sono fatti.

6) *Per liberarsi da un'angoscia. spesso lo scrivere rappresenta un equivalente della confessione o del divano di Freud. [...]*

Nessuno ha mai indagato il valore terapeutico della programmazione. La cosa potrebbe riservare delle sorprese. Dall'autobiografia di Linus Torvalds si evince che scrivere Linux lo abbia guarito dall'asocialità. Qualcuno può sostenere che in certe interfacce è possibile ravvisare una qualche angoscia...

7) *Per diventare famosi. Credo che solo un folle possa accingersi a scrivere unicamente per diventare famoso; ma credo anche che nessuno scrittore, neppure il più modesto, neppure il meno presuntuoso, neppure l'angelico Carroll sopra ricordato, sia stato immune da questa motivazione. Aver fama, leggere di sé sui giornali, sentire parlare di sé, è dolce, non c'è dubbio; ma poche fra le gioie che la vita può dare costano altrettanta fatica, e poche fatiche hanno risultato così incerto.*

Quello che Raymond¹⁵ chiama “egoboo” o l'esaltazione della reputazione individuale.

8) *Per diventare ricchi. Non capisco perché alcuni si sdegnino o si stupiscano quando vengono a sapere che Collodi, Balzac e Dostoevskij scrivevano per guadagnare, o per pagare i debiti di gioco, o per tappare i buchi di imprese commerciali fallimentari. Mi apre giusto che lo scrivere, come qualsiasi altra attività utile, venga ricompensato. Ma credo che scrivere solo per denaro sia pericoloso, perché conduce quasi sempre ad una maniera facile, troppo ossequente al gusto del pubblico più vasto e alla moda del momento.*

Risulta evidente il confronto di questa franca opinione con chi si stupisce dei programmatori che si fanno pagare per scrivere free software.

9) *Per abitudine. [...]*

Non sorprende che dall'indagine su *Fun and Software Development* di Benno Luthiger risulti che *Software development is significantly more fun when it takes place in the context of an open source project than when it is carried out under commercial conditions.*

L'attività di programmazione, se lavorativa e routinaria, spogliata degli aspetti di creatività e innovazione diviene abitudinaria e noiosa. Ciò che compone il *fun* è proprio l'aspetto creativo.