

Università Iuav di Venezia

SCUOLA DI DOTTORATO

dottorato in Architettura, città e design  
curriculum Scienze del design

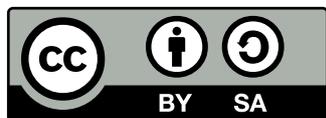
xxvii ciclo  
A.A. 2014/2015

## **Il design dei prodotti interattivi nel contesto delle pratiche open source**

relatore  
Davide Rocchesso

dottoranda  
Serena Cangiano

coordinatore  
Raimonda Riccini



2015, CC BY SA  
Quest'opera è distribuita con Licenza  
Creative Commons Attribuzione 4.0  
Internazionale.

© Autori e artisti per le immagini  
Dove possibile, sono state indicate  
le fonti delle immagini.

Per contattare l'autore:  
[serenacangiano@gmail.com](mailto:serenacangiano@gmail.com)

*font*  
Minion Pro (Adobe 1990)  
disegnato da Robert Slimbach  
Form (The Foundry 1999)  
disegnato da David Quay e Freda Sack

## sommario

7		Introduzione
13	1	L'Ecosistema open source. Definizioni e concetti di base
13	1.1	Il termine "Ecosistema open source"
15	1.2	Gli elementi dell'Ecosistema open source: open source hardware, open design, DIY, le tecnologie di produzione digitalizzata, le licenze open source
17	1.2.1	Open source hardware: piattaforma di innovazione e di semplificazione della tecnologia
30	1.2.2	Open design o del contributo del design ai Commons
38	1.2.3	Le tecnologie di produzione digitalizzata
46	1.2.4	Il fai da te tecnologico. Comprendere le motivazioni dei futuri utilizzatori.
52	1.2.5	Le licenze open source per il design
54	1.3	Open source nel design: nuovi paradigmi o attualizzazione di manifesti pioneristici del design?
67	2	I possibili livelli progettuali nell'open source
68	2.1	Definizione dei livelli progettuali nel design di artefatti interattivi open source
69	2.2	Il file sorgente originale come blueprint "aumentato"
83	2.3	Le istruzioni e la documentazione a supporto della riproducibilità

85	2.3.1	I tutorial generati dagli utenti
88	2.3.2	I tutorial redazionali
90	2.3.3	I tutorial integrati in “Learning system”
96	2.3.4	La documentazione dell’interazione
111	2.4	I modelli e le modalità di collaborazione
115	2.4.1	I modelli di collaborazione per lo sviluppo hardware open source
117	2.4.2	I modelli di collaborazione nello sviluppo del prodotto
129	2.5	I sistemi modulari e parametrici
130	2.5.1	La modularità
145	2.5.2	La parametricità
151	2.6	Le macchine e le interfacce di riproduzione
157	3	Due casi studio di artefatti interattivi open source
157	3.1	Open Mirror di Studio Habits
159	3.1.1	Modello di rilascio open source
160	3.1.2	Processo di progettazione e sviluppo
166	3.2	Primo di Solid Labs
168	3.2.1	Modello di rilascio open source
168	3.2.2	Processo di progettazione e sviluppo
173	4	“Progettare prodotti-piattaforma”. Proposta di un framework per il design di artefatti interattivi open source
173	4.1	Cosa sono i prodotti-piattaforma?
176	4.2	Descrizione del framework: dall’User Centered Design e Meta-design al design dei prodotti-piattaforma
180	4.2.1	I requisiti. La programmabilità, la riproducibilità, la generatività
183	4.2.2	Gli strumenti: le attitudini e le motivazioni dell’utente
188	4.2.3	Le attività: il processo di progettazione e il processo di sviluppo
193	4.3	“Progettare prodotti-piattaforma”. Applicazione del design framework in un workshop

194	4.3.1	La metodologia del workshop
197	4.3.2	I risultati del workshop
203	4.3.3	Alcune riflessioni sul workshop
207	5	Conclusioni: i punti critici nella progettazione di artefatti open source
211	6	Apparati
219	7	Bibliografia
237		Ringraziamenti



## Introduzione

Lo sviluppo del settore Open Hardware e Open Design, ovvero artefatti fisici e tecnologici le cui informazioni e specifiche d'implementazione sono rilasciate pubblicamente tramite licenze libere, combinato alle tecnologie di produzione digitalizzata e al crescente fenomeno del movimento dei maker, sta stimolando la diffusione di nuove pratiche di progettazione e produzione di artefatti e dispositivi tecnologici. Queste pratiche si basano, per esempio, sulla possibilità che progettisti ed esperti di tecnologia oppure amanti del fai da te, interagendo con le comunità in rete e con piattaforme e servizi collaborativi, possano utilizzare o condividere i file digitali (schemi hardware, codici software e file 2D e 3D) di un oggetto o dispositivo al fine di permettere ad altri di modificarlo o realizzarlo tramite tecnologie di prototipazione rapida a basso costo. Le pratiche intorno all'open source sono manifestazioni del passaggio da un'era postmoderna all'era della partecipazione (*peer to peer*) in cui la cultura del "fare insieme" (*Do it together*) è resa possibile dalle tecnologie della comunicazione, dell'informazione e, nel caso degli artefatti fisici, delle tecnologie della produzione. Se, da un lato, queste pratiche dimostrano la possibilità di implementare metodi, processi e business sostenibili, innovativi e compatibili con gli attuali e futuri scenari socio economici e ambientali (crisi, scarsità di risorse), dall'altro sono parte di un fenomeno in divenire la cui interazione con la disciplina del design è tanto poco definita quanto dirompente.

Il design dell'interazione, ovvero l'ambito della progettazione riferito a artefatti, ambienti e servizi interattivi e tecnologicamente avanzati, ha da sempre beneficiato dello sviluppo del settore delle tecnologie open source così come ha contribuito in modo

attivo alla diffusione di ambienti di programmazione e piattaforme di prototipazione open source di facile utilizzo indirizzata a designer e amatori (p.e. Processing, Arduino).

Con la diffusione delle pratiche open source anche nel design e la diffusione di progetti di macchine di prototipazione rapida a basso costo e open source inizia la sperimentazione sulla creazione di nuove tipologie di artefatti interattivi che ridefiniscono il “territorio” del design.

In questo nuovo scenario di uso e d’interazione con dispositivi interattivi “aperti”, nuove domande emergono in merito alla necessità di sviluppare approcci, metodi e teorie a supporto del design in un contesto open source così come la necessità di definire delle soluzioni che rendano più efficace e realistico il paradigma stesso dell’open source nel design.

Progettare nell’open source, progettare per l’open source è il tema alla base di questa ricerca di dottorato. La ricerca punta a definire gli elementi di un possibile *framework* di progettazione che definisce l’ambito di azione del designer, e in particolare, dell’interaction designer nel contesto delle pratiche open source.

Nello sviluppo del lavoro di ricerca si considerano come punti di riferimento le seguenti domande:

1. Quali sono i nuovi livelli di configurazione di un artefatto interattivo e le opportunità progettuali attivate dall’ecosistema Open Source, ovvero l’insieme di piattaforme software, hardware, servizi e comunità on-line?

2. Che tipo di informazioni e competenze devono essere trasferite agli utenti del mondo open source al fine di supportarli nel processo di riconfigurazione di un artefatto interattivo open source?

La tesi si sviluppa in quattro capitoli. Nel primo capitolo sono presentati i concetti di base dell’open source tramite la definizione di un modello di analisi olistico basato sulla definizione dell’“Ecosistema open source”, ovvero un modello analitico che enfatizza l’interazione tra i seguenti domini: Open Source Hardware, Open Design, DIY, le tecnologie di produzione digitalizzata e le licenze open source. Il capitolo punta a fornire le definizioni e i principi legati a questi domini al fine di chiarire la loro relazione

diretta o indiretta con l'ambito della progettazione di artefatti interattivi open source. Inoltre, questo capitolo presenta un *excursus* sulle esperienze legate all'arte moltiplicata e all'arte programmata degli inizi degli anni sessanta al fine di introdurre una relazione tra queste esperienze pionieristiche a cavallo tra attività artistica e progettuale e le correnti pratiche di progettazione in ambito open source.

Seppur l'intento della ricerca non sia dare un quadro storico alle pratiche di progettazione collaborative e aperte, questo *excursus* individua degli elementi di continuità tra i principi e i modi della progettazione open source e le esperienze di alcuni massimi esponenti del design italiano come Bruno Munari, in particolare, nel discorso intorno alla creazione di multipli d'arte o alla realizzazione di ambienti e opere interattive.

Nel secondo capitolo affronto la descrizione e l'analisi delle pratiche connesse in maniera diretta o indiretta all'ambito della progettazione e dello sviluppo di artefatti tecnologici o di design open source. L'obiettivo è definire e descrivere i livelli di intervento dal punto di vista progettuale nel processo di sviluppo di un artefatto interattivo open source. I livelli progettuali individuati sono cinque: la creazione di *blueprint* si trasforma nella realizzazione di un artefatto digitale aumentato che contiene le informazioni per riprodurre un prodotto e allo stesso tempo è la tecnologia che permette di produrre il prodotto tramite macchine di produzione digitale; la creazione di guide passo-passo e di piattaforme basate su contenuti generati dagli utenti sono attività che diventano parte del processo di progettazione che si apre alla comunità di persone in internet; la collaborazione tra i realizzatori di un prodotto e le persone che possono contribuire a migliorarlo o estenderlo dipende da un modello che è il risultato dell'attività progettuale che punta a stimolare il processo di design partecipativo; la modularità del prodotto tecnologico e il suo design basato su parametri si trasforma nell'oggetto della progettazione e nella strategie per assicurare la sua riproducibilità; infine, le attività di progettazione nel contesto open source si estendono alla realizzazione delle macchine di produzione e alle loro interfacce.

Nel terzo capitolo sono descritti e analizzati due casi studio di artefatti interattivi open source.

L'analisi di ogni caso studio si basa sulla descrizione degli aspetti salienti che riguardano le strategie di rilascio open source e il modello di sviluppo del prodotto. Il primo caso studio è Open Mirror, uno specchio che funziona da altoparlante per iPhone e la cui interfaccia si basa su gesti; il secondo caso studio è Primo, un giocattolo interattivo per insegnare la programmazione ai bambini.

Le specifiche e i file sorgente di entrambi i prodotti sono state rilasciati tramite licenze Creative Commons al fine di permettere ad altre persone di modificare, riprogrammare e riprodurre ogni elemento del prodotto, dal codice informatico all'hardware e al design.

Nel quarto capitolo è descritta la proposta del *framework* "Progettare prodotti-piattaforma" laddove un prodotto-piattaforma è un artefatto interattivo tecnologico che gli utenti possono modificare o sviluppare ulteriormente grazie ad un ecosistema di servizi digitali, documentazione condivisa e licenze libere. Il *framework* è descritto in relazione ai modelli proposti nell'ambito dell'User Centered Design e la Human Computer Interaction (Meta-design) e include la documentazione della metodologia e dei risultati del workshop "Progettare prodotti-piattaforma" tenutosi il 18 ottobre 2014 presso la sede del makerspace WeMake a Milano.

Dalla definizione di strumenti di analisi all'applicazione di un framework di progettazione, la tesi documenta un percorso e una proposta per interpretare il ruolo del designer e dell'attività progettuale nel contesto delle pratiche di sviluppo e progettazione open source. Queste pratiche supportano un nuovo modo di "fare le cose", valorizzando quelle qualità proprie del design di operare a cavallo fra tecnica, innovazione dei sistemi di produzione e bisogni dell'uomo (Botta, Cangiano 2014). Esse vivono dell'impatto fra l'attività di progetto e le tecnologie avanzate e delinea un contesto in cui il designer evolve le proprie competenze ed estende le sue sfere di azione e influenza. Dalla customizzazione, guidata da logiche di mercato orientate alla vendita di prodotti personalizzati, si passa alla produzione di artefatti 'aumentati che sono

predisposti per essere modificati, adattati o evoluti nella loro forma, nelle loro funzioni e nei loro comportamenti rispetto a nuove necessità che emergono da un contesto tecno-sociale rinnovato. Questo contesto è la 'piattaforma mondo' in cui il designer si trova oggi ad operare.



# 1. L'Ecosistema open source

## Definizioni e concetti di base

### 1.1 Il termine "Ecosistema open source"

Il punto di inizio di questo studio è la proposta di uno strumento concettuale che descrive il contesto della ricerca il cui obiettivo è indagare l'impatto dello sviluppo di tecnologie open source e le pratiche connesse ad esso sulla progettazione di artefatti fisici interattivi. Ho definito questo strumento "Ecosistema open source", ovvero un insieme di paradigmi, pratiche e fenomeni connessi allo sviluppo di artefatti fisici tecnologici, la cui interazione va a ridefinire i confini dell'ambito di azione della disciplina del design e, in particolare, le attività dei progettisti di sistemi e prodotti interattivi. Da un lato, utilizzo questo strumento concettuale per comunicare i risultati della ricerca, dall'altro esso è un elemento fondamentale per comprendere il framework progettuale che propongo nel capitolo quattro.

Perché utilizzo il termine ecosistema? Ecosistema è il termine utilizzato in un emergente approccio alla progettazione dell'esperienza utente (User Experience Design) che riflette sulle diverse modalità di interazione quando l'informazione diventa accessibile tramite una molteplicità di dispositivi (p.e. computer, tablet, smart phone) (Levin 2014). Questo approccio si appropria del concetto di ecosistema utilizzato in biologia, che vede il sistema degli organismi viventi e non viventi interconnessi tra di loro, e lo utilizza per descrivere l'attuale contesto tecnologico in cui le persone interagiscono con diversi dispositivi connessi alla rete: i dispositivi supportano un tipo esperienza continua e coordinata e per questo motivo il loro insieme creano appunto un ecosistema. Sulla base della definizione adottata nell'ambito del user ex-

perience design, utilizzo il termine ecosistema al fine di definire una relazione di interdipendenza tra i paradigmi, le pratiche e i fenomeni che sono direttamente connessi allo sviluppo di prodotti e tecnologie open source con altri non direttamente connessi a questo ambito come, per esempio, il paradigma della produzione digitalizzata (digital fabrication) o il movimento dei maker e del fai da te elettronico. Utilizzare il termine ecosistema mi permette di superare tramite un approccio olistico la difficoltà di definire una serie di fenomeni che sono di recente diffusione e altamente variabili poiché legati al ritmo veloce dell'evoluzione tecnologica.

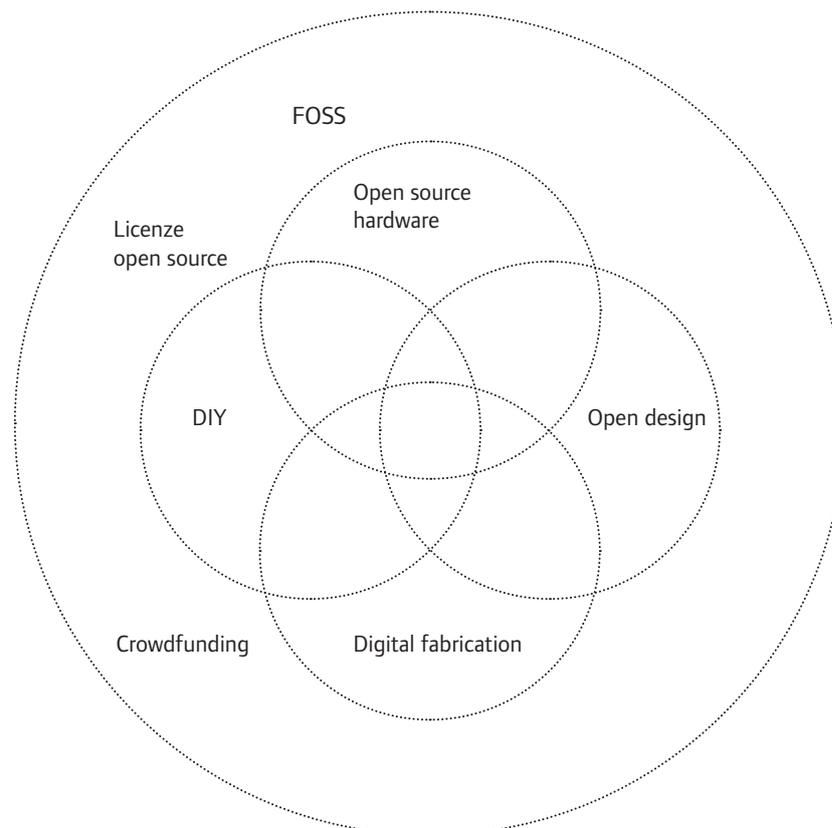
## 1.2 Gli elementi dell'“Ecosistema open source”: open source hardware, open design, DIY, le tecnologie di produzione digitalizzata, le licenze open source

Gli elementi dell'Ecosistema open source sono quattro. Il primo elemento riguarda l'open source hardware, ovvero l'ambito in cui sono progettati, sviluppati e resi pubblici sul web e tramite licenze libere gli schemi e i file sorgente per realizzare artefatti tecnologici quali microcontrollori, parti elettroniche e parti meccaniche. Questo elemento ha delle connessioni dirette con il tema della progettazione di artefatti interattivi open source poiché rappresenta l'infrastruttura principale che ha dato origine alla traslazione dei modelli dell'open source software al mondo degli artefatti fisici. Il secondo elemento è l'open design, ovvero l'ambito che si focalizza sullo sviluppo e il rilascio sul web e tramite licenze libere dei processi e dei prodotti risultanti dalle attività connesse alla discipline del progetto, dal design grafico, al design del prodotto, all'architettura al design delle interfacce e dei servizi. Questo secondo elemento ha una connessione diretta con la progettazione di prodotti interattivi open source e, data l'emergente configurazione delle sue pratiche e modelli, rappresenta un ambito di cui questo studio punta ad essere un contributo. Il terzo elemento è la produzione digitalizzata, ovvero l'insieme delle tecniche e dei processi tecnologici che permettono di produrre un artefatto a partire dalla conversione del modello tridimensionale o bidimensionale digitale delle sue parti in istruzioni interpretabili da macchine di manifattura additiva o sottrattiva come, per esempio, stampanti 3D, fresatrici a controllo numerico e macchine a taglio laser. Le tecnologie di produzione digitalizzata e della prototipazione rapida sono diventate accessibili al di fuori dell'ambito della produzione industriale e sono utilizzate, per esempio, in contesti di sperimentazione e formazione informale come i FabLab (fabrication laboratory). La diffusione di tecnologie di produzione digitalizzata favorisce l'implementazione di modelli open nella progettazione, sviluppo e produzione di artefatti interattivi su diversi livelli (p.e. la collaborazione a distanza e la produzione distribuita di artefatti). Allo stesso tempo il settore dello sviluppo di tecnologie

di produzione digitalizzata beneficia della diffusione di pratiche open source grazie alle quali si diffondono soluzioni espandibili, customizzabili e a costo ridotto. Il quarto elemento si riferisce alle pratiche legate al movimento del fai da te elettronico (DIY). Queste pratiche sono connesse all'ambito della progettazione di prodotti interattivi open source poiché sono la manifestazione concreta di pratiche basate sulla condivisione di conoscenza *peer to peer*, della collaborazione e partecipazione dal basso, della creazione di comunità di sviluppo di un prodotto. Infine, le licenze libere per la pubblicazione di artefatti tecnologici come le licenze Creative Commons sono il quinto e ultimo elemento dell'Ecosistema. Le licenze libere per il rilascio di un progetto open source sono l'infrastruttura legale che legittima i modelli di progettazione e sviluppo open source favorendo la creazione di modelli di business sostenibili.

Nella definizione di "Ecosistema open source" escludo vo-

Fig. 1. Diagramma dell'Ecosistema open source.



lontaneamente molti elementi. In particolare, due elementi importanti non sono approfonditi in questo studio. Il primo di questi è il FOSS Free e Open Source Software (Wikipedia 2014), al quale faccio riferimento nel secondo capitolo per presentare i modelli di collaborazione e di sviluppo di un progetto open source. Il FOSS è implicitamente parte dell'Ecosistema poiché rappresenta l'elemento che ha generato i successivi fenomeni legati alla progettazione e sviluppo open source di artefatti fisici. Il secondo elemento escluso dallo studio sono i modelli alternativi di finanziamento all'avviamento di business, il *crowdfunding*. Il *crowdfunding* è la pratica di finanziamento collaborativo di un progetto tramite il contributo di un largo numero di persone e l'utilizzo di piattaforme on-line come, per esempio, Kickstarter e Indiegogo. Questa pratica permette a chi intende sviluppare un prodotto di accedere a un capitale generato dai contributi di persone che acquistano il prodotto prima della sua reale produzione o semplicemente donano del denaro in favore della sua realizzazione (Howe 2009).

Questo tipo di modelli di finanziamento si contrappone ai modelli classici dell'industria del design poiché mette in diretto contatto il progettista e l'utilizzatore finale: sulla base di una domanda generata da un gruppo di *early adopters*, aumenta la possibilità per un nuovo tipo di prodotto di essere implementato e messo sul mercato. Seppur il modello *crowdfunding* faciliti il processo di diffusione di un prodotto altamente riprogrammabile e customizzabile non è esclusivamente indirizzato all'ambito della progettazione e sviluppo di prodotti interattivi open source.

### 1.2.1 Open source hardware: piattaforma di innovazione e di semplificazione della tecnologia

La versione 0.1 della definizione di Open source hardware riporta che:

il termine open source hardware si riferisce ad artefatti tangibili – macchine, dispositivi o altri prodotti fisici – i cui blueprint sono resi pubblici in modo che chiunque possa realizzare quegli artefatti, modificarli, distribuirli e usarli in generale per svariati scopi (OSHWA 2014)<sup>1</sup>.

---

1 OSHWA 2014, mia traduzione.

La diffusione di progetti di open source hardware si basa sulla traduzione delle pratiche e dei modelli di sviluppo proposti dal FOSS (Free and open source software) per l'ambito dello sviluppo di prodotti fisici, in particolare, di prodotti tecnologici quali microcontrollori, piattaforme di prototipazione hardware-software, parti meccaniche. A partire dal 1997 diverse iniziative sono state avviate per rendere possibile questa traduzione come, per esempio, il programma Open Hardware Certification Program che permette ai produttori di hardware di autocertificare i loro prodotti come prodotti open source. A questa iniziativa promossa da Bruce Perens, ideatore della Open Source Definition e co-fondatore della Open Source Initiative ([opensource.org](http://opensource.org)), sono successe altre iniziative con l'obiettivo di regolare e favorire lo sviluppo di progetti open source nel mondo dell'hardware. Nonostante questi progetti, lo sviluppo di hardware open source si è concretizzato solo a partire dall'anno 2000 grazie a tre fattori quali la diffusione di internet ad un livello che ha reso possibile lo scambio dei file sorgente di progetti hardware, la riduzione dei costi delle tecnologie di produzione di hardware e il successo commerciale di alcuni progetti che ne ha aumentato la visibilità e resi comunicabili i benefici in termini di business (Mota 2014).

In particolare, quest'ultimo fattore ha determinato il successo del modello open source per lo sviluppo e la produzione di hardware e può essere ricondotto a poche aziende e progetti quali RepRap, progetto di stampanti 3D auto-replicabili e fai da te, Arduino, la piattaforma di prototipazione hardware-software, e le aziende Adafruit e SparkFun, entrambe produttrici e rivenditrici di hardware e kit elettronici open source.

Le motivazioni che hanno portato il modello open source a funzionare oltre che per il software anche per l'hardware risiedono soprattutto nella stretta connessione con la capacità di innovare: gli inventori di hardware, rilasciando la loro tecnologia in open source, riescono ad essere supportati da una comunità estesa di sviluppatori, creando dei centri di conoscenza on-line e ottenendo un'enorme quantità di dati forniti da utenti-consumatori. Dal lato degli utenti consumatori, invece, il beneficio è riconosciuto nella possibilità di accedere a un ventaglio di prodotti molto diversi che riescono a emergere sul mercato grazie alla creazione di derivati di progetti esistenti.

Un esempio di questo fenomeno sono le stampanti 3D la cui varietà sul mercato dipende dalla riduzione del tempo di creazione di un derivato a partire da un progetto di stampante 3D open source. L'accesso ad un ventaglio di soluzioni più ampio è un argomento chiave nei modelli basati sul rilascio open source di hardware poiché il consumatore ha la possibilità di acquistare prodotti che prima non erano presenti sul mercato, perché troppo di nicchia per spingere un'azienda a finanziarne la produzione o perché troppo complicati da produrre.

Il gruppo principale di utenti di un progetto di hardware open source sono anche coloro che contribuiscono al suo sviluppo, ovvero persone che hanno competenze di elettronica e programmazione. Diversamente dal software, il beneficio più interessante per l'utente consumatore di hardware open source risiede nella possibilità di accedere a delle soluzioni tecnologiche di difficile reperibilità nel mercato *business to consumer* o totalmente inesistenti in qualsiasi mercato anche quello *business to business*. Open source hardware e innovazione tecnologica sono, infatti, connessi l'uno all'altra proprio perché il modello open source permette di produrre nuove tipologie di prodotti che soddisfano i bisogni di una nicchia di mercato piuttosto che quelli di un mercato di massa (von Hippel 2005).

Un altro aspetto fondamentale del modello open source è lo sviluppo di hardware che permette di creare soluzioni a problemi complessi che sarebbero altresì impossibili da risolvere a causa dell'enorme investimento di risorse economiche e umane. Oltre agli sviluppatori o agli utilizzatori finali di un prodotto open source, il terzo gruppo di interesse che beneficia di questo modello sono, quindi, le comunità di persone che condividono una problema complesso come, per esempio, l'assenza di infrastrutture di comunicazione, di produzione o di trasporto. Un esempio emblematico in questo senso è lo sviluppo di reti di trasporto di merci basate sull'utilizzo di quadricopteri, elicotteri telecomandati dotati di quattro motori e quattro sistemi di eliche: un primo caso è il progetto Dronenet (Frazer et al 2013) che punta a creare una rete di trasporto con quadricopteri realizzati con hardware software e elettronica open source. Il progetto nasce per risolvere il problema della creazione di infrastrutture stradali efficienti per mettere

in comunicazione villaggi africani e per semplificare il trasporto di medicine o altri beni durante campagne di aiuto umanitario (The Economist 2008). Un secondo caso d'interesse è Open Source Ecology, uno dei principali progetti e comunità di open source hardware che sviluppa la documentazione di cinquanta macchine industriali (dal trattore alla macchina per la stampa di circuiti) in modo da renderle accessibili a un costo ridotto rispetto alle macchine commerciali e da renderle modulari e altamente riparabili. Fondato nel 2003, da Marcin Jakubowski, Open Source Ecology ha la missione di creare “un'economia open source, libera che stimola l'innovazione a partire dalla collaborazione aperta” (Open Source Ecology 2014)<sup>2</sup>. Il legame tra hardware open source e innovazione si estende in questo caso alla creazione di modelli economici e di business in cui la collaborazione tra comunità di persone è uno dei fattori chiave.

Sviluppatori, utenti consumatori, comunità/società sono, quindi, i gruppi di utenze sulle quali ricadono i benefici pratici della diffusione di progetti di open source hardware. A partire da questa classificazione di gruppi di utenti e benefici è possibile individuare tre caratteristiche dell'open source hardware (OSHW). OSHW è un *innovation enabler*, ovvero un funziona da infrastruttura che supporta la realizzazione di prodotti non presenti sul mercato. OSHW è, inoltre, uno *user e community empowerment tool*, ovvero un' infrastruttura che abilita comunità di persone a collaborare sulla risoluzione di problemi complessi o problemi socialmente rilevanti. Infine, OSHW è un *technology mediator*, ovvero un'infrastruttura che propone soluzioni sistemiche in favore della semplificazione della tecnologia.

Riferendoci agli studi sulle comunità creative di Manzini, micro-controllori e piattaforme di prototipazione hardware open source possono essere considerate delle *community enabling platforms*, ovvero strumenti i cui modelli di sviluppo e business si basano sul coinvolgimento delle comunità di utilizzatori tramite la rete e generano valore per queste comunità che le utilizzano per realizzare progetti di natura profit o semplicemente amatoriale (Manzini 2005). Il coinvolgimento avviene su più livelli come, ad esempio,

---

2 Open Source Ecology 2014, mia traduzione.

le persone possono contribuire condividendo in rete la documentazione di un progetto interattivo o dei suoi derivati realizzati con una specifica piattaforma, componente o macchina open source. Questo tipo di coinvolgimento determina come conseguenza l'acquisizione di controllo e potere sulla tecnologia da parte delle persone che si manifesta nella forma di un controllo totale sulla tecnologia che utilizzano poiché la pubblicazione della documentazione fornisce le informazioni dettagliate su ogni componente e sul loro utilizzo e permette loro di interfacciare un prodotto con i dispositivi e i sistemi che le persone ritengono più giusti per soddisfare i loro bisogni. Infine, l'appropriazione tecnologica si trasforma in un potere economico perché abilita le persone a creare prodotti derivati che possono essere realizzati per l'avviamento di progetti di business profittevoli.

La terza caratteristica chiave che ho attribuito all'open source hardware riguarda il processo di apertura della tecnologia, nel senso di aumento dell'accessibilità e usabilità tramite la sua semplificazione. Utilizzo come caso studio di riferimento uno dei progetti di open source hardware più diffusi: la piattaforma di prototipazione elettronica Arduino.

Il progetto nasce a Ivrea, in Italia, come risultato di anni di sviluppo di uno strumento a supporto della realizzazione di prototipi di artefatti interattivi da parte degli studenti e dei ricercatori dell'Interaction Design Institute di Ivrea IDII. Arduino nasce, infatti, come un *ubiquitous computing user experience design toolkit* (Kuniavsky 2010).

Nato come derivato e evoluzione del progetto Wiring di Hernando Barragán, studente del IDII, Arduino consiste in un ambiente di programmazione – Arduino IDE e un microcontrollore. A questi due elementi di base si aggiungono un sito e una wiki che raccolgono la documentazione necessaria per riprodurre e usare la scheda e il software e un forum che abilita l'interazione tra i membri della comunità di utilizzatori di Arduino.

Progettato come strumento per la prototipazione di prodotti interattivi nel contesto della didattica del design dell'interazione, Arduino è diventato una piattaforma per l'insegnamento dell'elettronica e della programmazione hardware e parallelamente è stato

scelto come componente hardware per la realizzazione di prodotti finiti come, per esempio, le stampanti 3D open source e fai da te, Makerbot e Ultimaker (Banzi 2012).

Arduino è un progetto che esemplifica i tre aspetti chiave dell'open source hardware come indicati precedentemente: è un innovation enabler poiché è diventato l'hardware di riferimento per molti progetti che propongono nuove tipologie di prodotti altamente customizzabili, stimola la user empowerment poiché è uno strumento diffuso in ambito amatoriale soprattutto per l'insegnamento della programmazione e dell'elettronica, infine, il suo contributo più interessante è relativo alla semplificazione della tecnologia poiché il suo sviluppo si è allontanato dai requisiti puramente ingegneristici e si basa sui requisiti propri della user experience design (Banzi 2014).

#### a. Il software Arduino IDE

Arduino IDE (Integrated Development Environment) è un software open source scaricabile gratuitamente del sito di Arduino. L'interfaccia utente di Arduino IDE si basa sull'interfaccia dell'ambiente di programmazione Processing ([processing.org](http://processing.org)) che era diffuso già a partire dal 2001 come strumento di insegnamento della programmazione a designer e artisti. Come Processing, la IDE di Arduino presenta un numero limitato di funzionalità che sono visualizzati tramite pochi elementi di interfaccia grafica.

Il linguaggio di Arduino è basato su Java e propone metodi<sup>3</sup> (il modo in cui sono richiamate le funzioni) simili a quelli di Processing, ovvero si basa su una sintassi e interfacce software standard (Baafi 2014). Questa scelta deriva dalla volontà di supportare i principianti nell'apprendimento della programmazione hardware fornendo un'esperienza simile alla programmazione software basata su Processing.

---

<sup>3</sup> Metodo è un termine che viene usato nel contesto della programmazione orientata agli oggetti per indicare un sottoprogramma (subroutine) associato in modo esclusivo ad una classe e che rappresenta (in genere) un'operazione eseguibile sugli oggetti e istanze di quella classe, voce "Metodo (programmazione)", in [Wikipedia.org](http://it.wikipedia.org/wiki/Metodo_%28programmazione%29), [http://it.wikipedia.org/wiki/Metodo\\_%28programmazione%29](http://it.wikipedia.org/wiki/Metodo_%28programmazione%29) (ultima visita 26 novembre 2014).

## b. Il protocollo di comunicazione

L'interfaccia USB è l'interfaccia standard per programmare una scheda Arduino, ovvero per trasferire il codice sviluppato con la IDE sul microcontrollore. USB (Universal Serial Bus) è il protocollo di comunicazione standard diffuso su tutti i personal computer. Questa scelta ha permesso di aumentare l'accessibilità della piattaforma Arduino a chiunque fosse in possesso di un computer riducendo la complessità della programmazione hardware professionale che di solito si affida a strumenti e componenti esterni come i programmatori hardware.

## c. L'interfaccia fisica

Al fine di semplificare l'utilizzo della scheda Arduino, sono stati ridotti drasticamente il numero dei componenti elettronici visibili sulla scheda e sono stati introdotti degli elementi di interfaccia fisica come per esempio: il tasto di "reset" che permette di annullare il programma caricato sulla scheda Arduino, il LED che comunica che la scheda funziona in maniera ottimale, i connettori femmina che permettono di connettere dei cavi elettrici senza la necessità di saldarli e che permettono di riutilizzare la scheda molteplici volte. Infine, è possibile considerare parte fondamentale dell'interfaccia fisica della scheda Arduino, i livelli dei simboli grafici e delle etichette stampate sul fronte della scheda che supportano il riconoscimento delle diverse porte di input e output.

Un aspetto chiave dell'interfaccia fisica di Arduino, quello dei connettori, ha permesso di sviluppare il concetto di *shield*, ovvero schede hardware compatibili con Arduino che integrano e espandono le sue funzionalità (p.e. una *shield wifi* permette di connettere Arduino ad una rete wifi). Anche sul piano della progettazione delle *shield*, ci sono casi in cui sono implementate soluzioni di semplificazione: la motor shield di Arduino presenta dei connettori maschio che permettono di connettere i moduli del Tinkerkit<sup>4</sup>: i connettori di colore arancione sono riferiti agli ou-

---

4 Il Tinkerkit è una scheda compatibile con Arduino che permette di connettere attuatori e sensori in modo semplice senza la necessità di saldare e creare dei circuiti (Tinkerkit 2014).

tput analogici, mentre i bianchi agli input analogici. L'uso del colore permette di distinguere le diverse funzionalità dei connettori semplificando l'uso della shield per utenti non esperti.

#### d. La wiki e il forum di Arduino

La documentazione accessibile sul sito ufficiale di Arduino è una componente fondamentale del progetto di semplificazione dell'utilizzo della piattaforma. La documentazione di Arduino è di tipo redazionale e deriva dalle attività condotte dai fondatori del progetto<sup>5</sup>. I contenuti redazionali del sito sono accompagnati dai contenuti generati dagli utenti e pubblicati tramite un motore collaborativo di tipo wiki, Arduino Playground ([playground.arduino.cc](http://playground.arduino.cc)). La gestione delle interazioni tra gli utilizzatori di Arduino avviene tramite il forum on-line di Arduino che conta circa 220.000 utenti. All'interno del forum partecipano dei *super user* che in maniera volontaria moderano i contenuti e contribuiscono fornendo supporto. Le risorse utili a sviluppare progetti con la piattaforma Arduino non sono limitate a quelle disponibili sui siti ufficiali del progetto: molte guide sono pubblicate su altri siti e da altri progetti hardware che si interfacciano con Arduino (p.e. Adafruit Learning System).

#### e. Il negozio on-line di Arduino

Il negozio on-line di Arduino è il sito di recente pubblicazione ([store.arduino.cc](http://store.arduino.cc)) che implementa una soluzione a supporto dell'acquisto dei componenti elettronici e hardware necessari per sviluppare un progetto interattivo con Arduino. Questo sito esemplifica uno dei modelli di business adottati da aziende che producono open source hardware (p.e. Sparkfun e Adafruit), ma allo stesso tempo rappresenta uno strumento di semplificazione tecnologica perché permette con facilità di accedere ai cataloghi di parti e componenti elettronici che altrimenti sono difficilmente acquistabili tramite siti dedicati a esperti di elettronica.

---

<sup>5</sup> I fondatori del progetto Arduino sono cinque: Tom Igoe (US), Massimo Banzi (IT), David Mellis (US), David Cuartielles (ES), Gianluca Martino (IT).

## f. L'identità visiva e esperienza utente

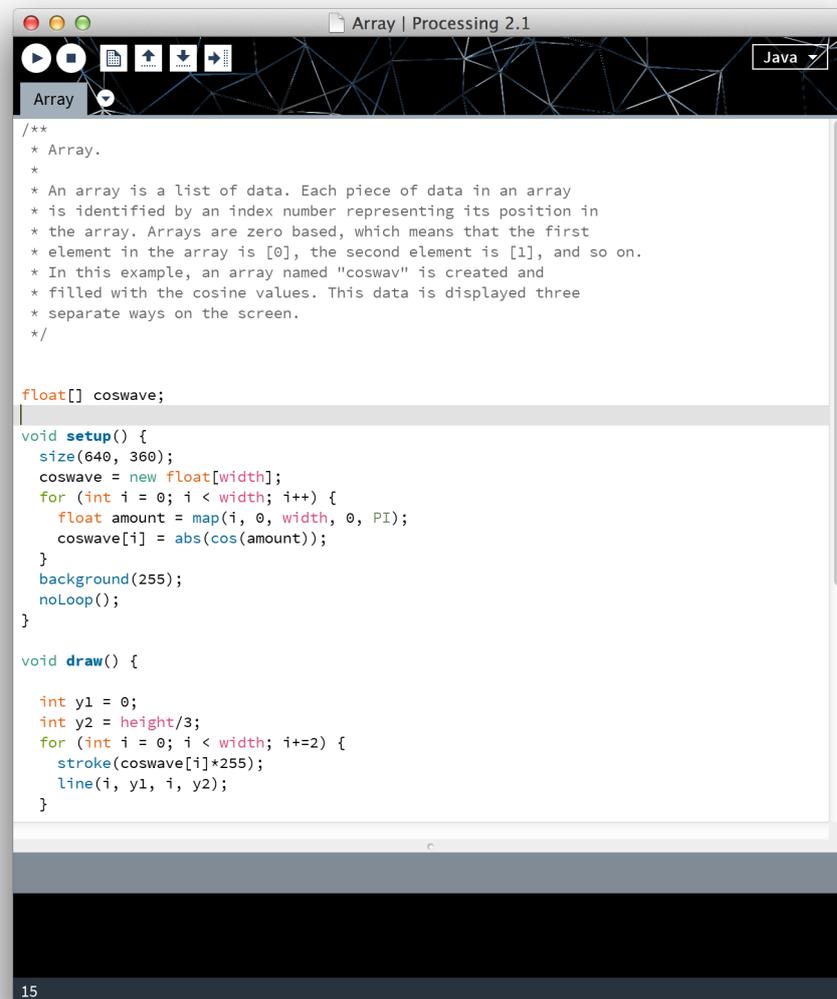
Nel processo di semplificazione tecnologica proposto da Arduino rientra anche il progetto della sua identità visiva che è finalizzata alla creazione di un'esperienza di acquisto legata alla fedeltà al marchio. Le soluzioni di immagine coordinata declinata sulle estensioni del progetto, quali per esempio il packaging, hanno l'obiettivo di risolvere i problemi legati alla contraffazione, ma puntano a fornire agli utilizzatori un livello ulteriore di comunicazione della semplicità di utilizzo della scheda. Un esempio è anche la sostituzione del kit per principianti: originariamente si trattava di una selezione di componenti elettronici di base come cavi, bottoni, LED. Nel 2013 il kit per principianti è stato sostituito dall'Arduino Starter kit, il kit in cui le parti fisiche, i componenti e l'Arduino sono selezionati per lo sviluppo di progetti di diversa complessità documentati in un manuale cartaceo progettato per utenti alle prime esperienze.

Come Arduino, le diverse piattaforme di prototipazione basate su open source hardware e indirizzate a utenti non esperti hanno contribuito all'inizio di un processo di semplificazione della tecnologia<sup>6</sup>. L'insieme delle scelte dipendenti dai requisiti dell'esperienza utente, dell'apprendimento da parte di non esperti, dell'accessibilità delle parti e dei componenti, è uno dei fattori chiave del successo di questo tipo di prodotti tecnologici. Il successo del modello open source applicato all'hardware si basa sull'implementazione di requisiti di *user experience design*, non dalla disponibilità gratuita dei file sorgente di un progetto hardware o della sua documentazione. David Cuartielles, co-fondatore del progetto, ribadisce questi concetti affermando che le persone non scelgono l'hardware perché è open source ma perché il prodotto è migliore degli altri (Cuartielles 2012).

---

<sup>6</sup> Presenta le medesime caratteristiche *Little Bits*, il kit modulare per sviluppare progetti elettronici e interattivi ([littlebits.cc](http://littlebits.cc)).

Fig. 2. La semplificazione dell'ambiente di programmazione in Arduino. Sulla sinistra un esempio di sketch (programma) scritto con l'ambiente di programmazione Processing. Sulla destra un esempio di sketch dell'Arduino IDE.



```
Array | Processing 2.1
Java
Array
/**
 * Array.
 *
 * An array is a list of data. Each piece of data in an array
 * is identified by an index number representing its position in
 * the array. Arrays are zero based, which means that the first
 * element in the array is [0], the second element is [1], and so on.
 * In this example, an array named "coswav" is created and
 * filled with the cosine values. This data is displayed three
 * separate ways on the screen.
 */

float[] coswave;

void setup() {
  size(640, 360);
  coswave = new float[width];
  for (int i = 0; i < width; i++) {
    float amount = map(i, 0, width, 0, PI);
    coswave[i] = abs(cos(amount));
  }
  background(255);
  noLoop();
}

void draw() {

  int y1 = 0;
  int y2 = height/3;
  for (int i = 0; i < width; i+=2) {
    stroke(coswave[i]*255);
    line(i, y1, i, y2);
  }
}
```

15

```
Blink | Arduino 1.5.8
Blink
/*
 * Blink
 * Turns on an LED on for one second, then off for one second, repeatedly.
 *
 * Most Arduinos have an on-board LED you can control. On the Uno and
 * Leonardo, it is attached to digital pin 13. If you're unsure what
 * pin the on-board LED is connected to on your Arduino model, check
 * the documentation at http://arduino.cc
 *
 * This example code is in the public domain.
 *
 * modified 8 May 2014
 * by Scott Fitzgerald
 */

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin 13 as an output.
  pinMode(13, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);           // wait for a second
  digitalWrite(13, LOW); // turn the LED off by making the voltage LOW
  delay(1000);          // wait for a second
}

1 Arduino Yún on /dev/tty.usbmodem1411
```

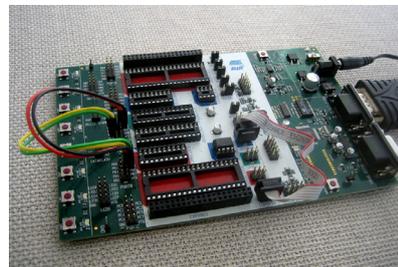
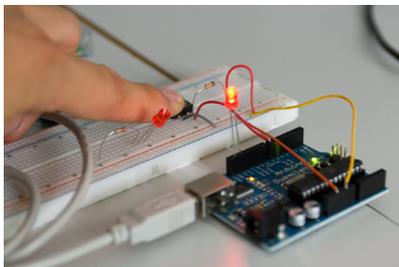
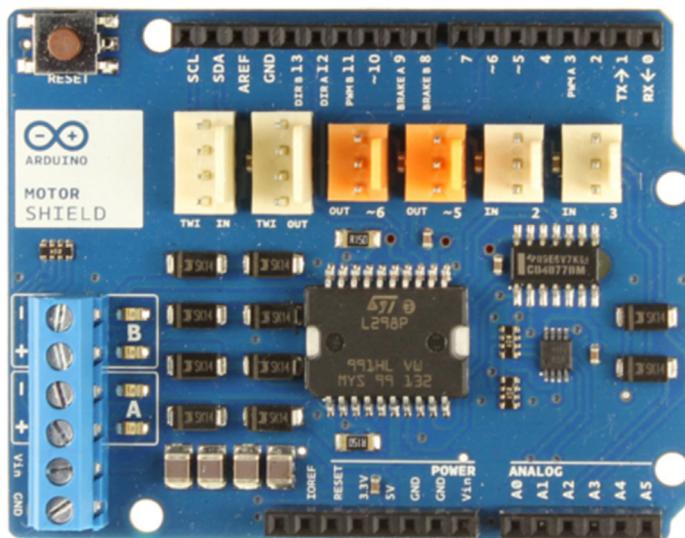
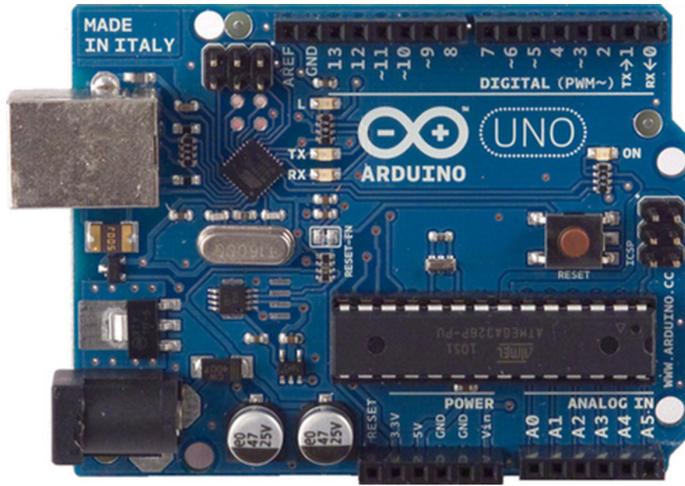
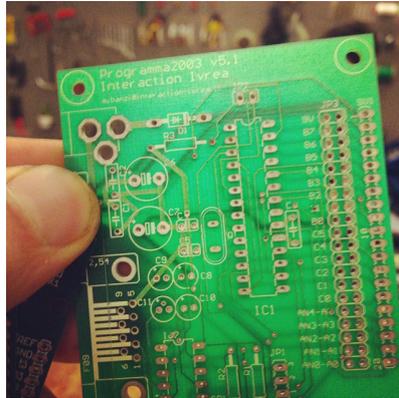


Fig. 3 I sistemi di interfacciamento standard. A sinistra una scheda Arduino con cavo USB e, a destra, il programmatore hardware Atmel AVR STK500

Fig. 4 La modularità dell'hardware: Programma 2003, Arduino UNO, Arduino Motor Shield



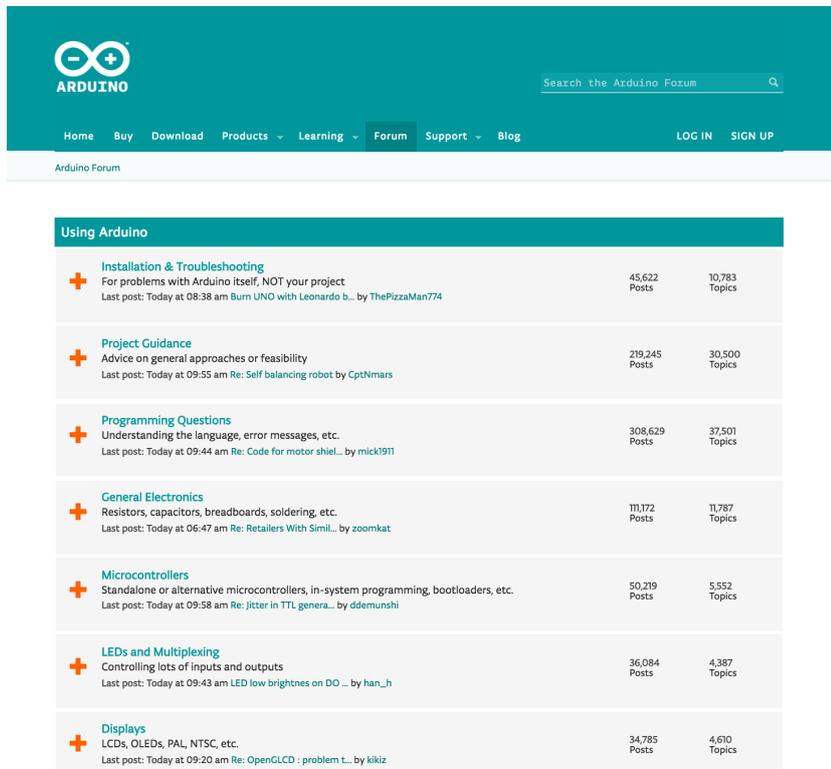
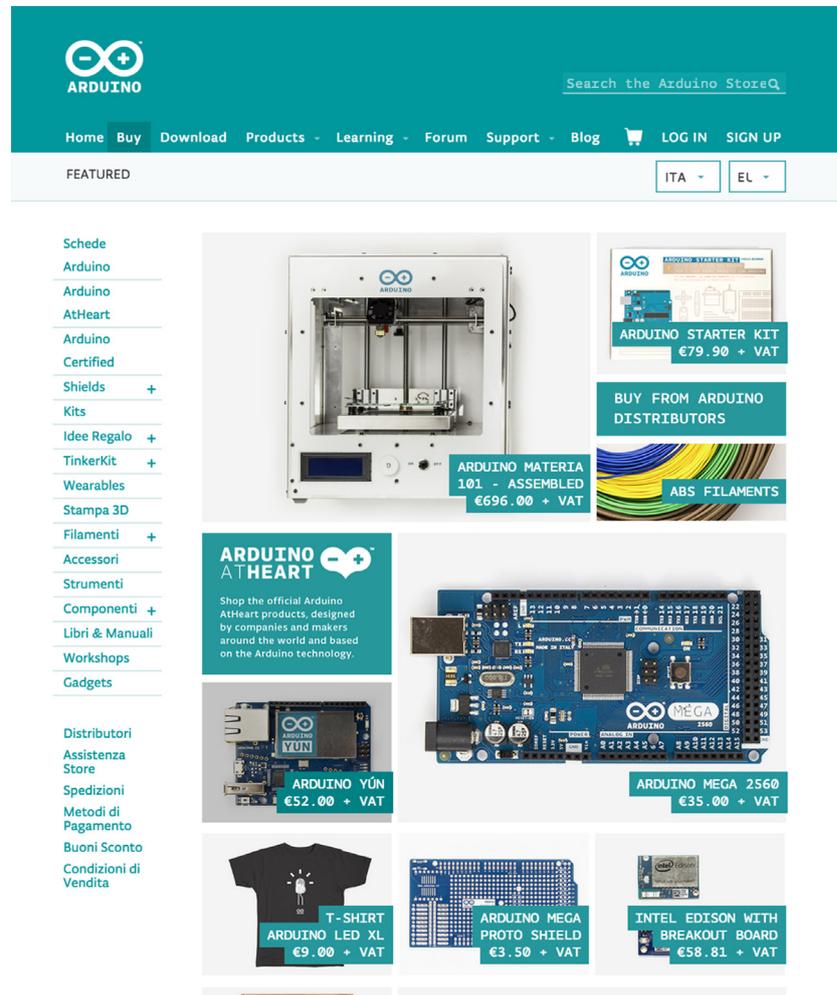


Fig. 5 La community on-line: Il forum di Arduino.



Fig. 6 Arduino Starter Kit

Fig. 7 Il negozio on-line di Arduino



### 1.2.2 Open design o del contributo del design ai Commons

Open sourcing design means to not make things twice (Parvin 2014).

Per presentare il secondo elemento dell'“Ecosistema open source”, l'open design, utilizzo la definizione del progetto collaborativo Open Design Definition, avviato nel 2011 dall'Open Design Working Group della Open Knowledge Foundation, organizzazione no-profit internazionale attiva nella promozione di tutte le forme

di conoscenza aperta e contenuti fondata nel 2004 a Cambridge (Open Knowledge 2014). La Open Design Definition è sviluppata in modo collaborativo da esperti e moderata dai membri dell'Open Design Working Group ([design.okfn.org](http://design.okfn.org)), tramite il software di sviluppo software collaborativo GitHub. La definizione di Open design è strettamente connessa alla definizione di Open Source così come proposta da Bruce Perens (Perens 1999). L'ultima versione della definizione è la seguente:

Open Design è un progetto di artefatto di design la cui documentazione sorgente è resa pubblicamente disponibile in modo che chiunque possa studiare, modificare, distribuire, fare, prototipare e vendere un artefatto sulla base di quel disegno. La documentazione dell'artefatto è disponibile nel formato adatto alla realizzazione dell'artefatto. Idealmente (ma non in maniera esclusivamente necessaria), l'Open Design usa componenti e materiali di facile lettura, processi standard, infrastrutture aperte, contenuti non soggetti a restrizioni, e strumenti di progettazione open source per massimizzare l'abilità delle persone di creare e usare l'hardware (Open Design Working Group 2014).<sup>7</sup>

Nonostante la definizione di Open design faccia riferimento al file sorgente di un artefatto fisico come, per esempio, il modello tridimensionale di un tavolo, il risultato della collaborazione ha portato a chiarire il significato del termine design nel contesto delle pratiche open source individuando tre significati diversi. Il primo significato è "progettare qualcosa" (design come processo), il secondo riguarda la documentazione, quindi, i disegni (design come blueprint), il terzo rimanda al risultato, ovvero il design nella sua versione utilizzabile (design come artefatto).

L'obiettivo della definizione non è quello di offrire una base teorica per comprendere cosa sia il design, ma uno dei punti di partenza per comprendere le implicazioni del paradigma open source nel design che possono essere interpretate solo se in relazione alla diffusione di pratiche di produzione *peer to peer* e all'attualizzazione del paradigma della produzione digitalizzata e distribuita. Alcuni studi cercano di individuare le caratteristiche peculiari dell'Open design e spostano il focus dalla questione ter-

---

<sup>7</sup> Open Design Working Group 2015, mia traduzione.

minologica alla relazione dell'Open design (Tonze et al 2014) con altri temi quali il copyright, la divisione del lavoro, l'accessibilità ai Commons laddove per Commons si intende l'insieme delle risorse culturali e naturali accessibili e di proprietà di tutti (Wikipedia 2014).

Bauwens (Bauwens 2009) spiega l'emergere dell'open design attraverso tre dimensioni: l'input, il processo e l'output. L'input è riferito al sistema dei contributi volontari nell'open design in cui nessuno deve chiedere il permesso per partecipare o usare del materiale grezzo che è libero dalla protezione del copyright in modo da migliorarlo e modificarlo. Il processo riguarda l'aspetto del design inclusivo, l'abbassamento delle soglie per la partecipazione, la suddivisione in compiti modulari piuttosto che lavori funzionali, la comune validazione della qualità e dell'eccellenza delle alternative proposte (peer governance). Infine, l'output è riferito al fatto che l'Open Design contribuisce a creare Commons usando licenze che assicurano che il valore risultante di un progetto sia disponibile per tutti, senza la necessità di richiedere permessi. Questo crea a sua volta un livello di risorse gratuite e open source che possono essere utilizzate per una successiva iterazione da parte di altre persone.

Studiando le implicazioni dell'apertura della produzione Seravalli le indica altri temi fondamentali per interpretare l'open source nel design. Le pratiche legate all'open source nel design e nella produzione generano una sovrapposizione tra i designer, gli utenti finali e i produttori e una trasformazione della relazione tra questi attori (*shuffling of roles*). La centralità della condivisione e della collaborazione implica che non sono generati solo dei prodotti e dei servizi che rispondono ai bisogni delle persone, ma anche delle connessioni sociali tra chi partecipa, e facilita lo scambio di conoscenza e tra loro (*beyond use value*). La crescente disponibilità di tecnologie e conoscenza riguardo al fare rinforza pratiche marginali che sono ai limiti della produzione di massa industriale (*alternative production forms*).

Infine, sono generate nuove forme di Commons che devono essere interpretate come i modi di organizzare e attuare la condivisione e la collaborazione e, di conseguenza questo implica nuovi modi di apertura (*diverse forms of Commons*) (Seravalli 2014).

L'Open design è associato al fenomeno del fai da te oppure alla possibilità di scaricare da internet un file 3D per riprodurlo con una macchina di manifattura additiva (p.e. le stampanti 3D). A questa immagine, si aggiunge lo scenario che vede l'amatore non esperto, il bricoleur, trasformarsi in un designer perché grazie a piattaforme on-line collaborative è abilitato a condividere in rete le sue creazioni. Lo scenario dell'open design come caricamento e scaricamento di file stampabili in 3D da internet è un'interpretazione non scorretta ma parziale: la digitalizzazione dei processi di progettazione e produzione di beni fisici e la distribuzione di tali beni in forma di codici è una delle infrastrutture tecnologiche che permette l'attualizzazione dell'open design, ma non ne rappresenta il suo valore distintivo. Avital individua i tratti distintivi dell'Open Design descrivendo le sue precondizioni che suddivide in quattro livelli al fine di ampliare la comprensione del paradigma open source applicato al design: il livello degli artefatti, dei processi, delle pratiche e delle infrastrutture, laddove "le infrastrutture sono i fondamenti istituzionali e tecnici che abilitano e limitano la vitalità delle pratiche del design" (Avital 2011).<sup>8</sup> A partire da questi livelli, Avital cerca di definire l'Open Design indicando cosa esattamente non è open design nello schema che segue.

Nel 2012 ho partecipato al primo simposio sull'Open design, organizzato dall'Università di Arte e Design di Linz. L'Open design Symposium proponeva una serie di presentazioni da parte di professionisti del mondo del design del prodotto, del design dell'interazione e delle arti multimediali per riflettere sulle emergenti pratiche legate al rilascio di progetti di design sulla rete tramite licenze Creative Commons. Tra i relatori c'era Ronen Kadushin, designer israeliano che ha iniziato a sperimentare e studiare le pratiche open source per applicarle alle sue creazioni di design a partire dalla sua tesi di master nel 2005 e poi con la pubblicazione di una serie di progetti di mobili rilasciati in rete tramite licenza Creative Commons. Nel suo Open Design Manifesto, Kadushin indica i due prerequisiti necessari per realizzare progetti di design open source, ovvero un file di design CAD, reso pubblico tramite licenze Creative Commons che può essere scaricato liberamente dalla rete, e un processo di produzione tramite

---

8 Mia traduzione.

Fig. 8. Le caratteristiche distintive dell'Open Design

THE DISTINCT FEATURES OF OPEN DESIGN	OPEN DESIGN IS...	OPEN DESIGN IS NOT...
	ACCESS	AVAILABLE, SHARABLE, LICENSED UNDER OPEN-ACCESS TERMS
BLUEPRINTS	SPECIFIED BY COMMON DIGITAL NOTATION LANGUAGE	SPECIFIED BY PROPRIETARY NOTATION LANGUAGE
DERIVATIVES	RECONFIGURABLE AND EXTENSIBLE	BLACK-BOXED AND FIXED
EXCLUSIVITY	REPRODUCIBLE	LIMITED TO A FINITE SERIES OR ONE-OFF
MEANS OF PRODUCTION	FABRICATED BY COMMERCIAL, OFF-THE-SHELF, MULTI-PURPOSE MACHINES	FABRICATED BY ARTISAN HANDWORK, CUSTOM-BUILT MACHINES OR MOULDS
MANUFACTURING PROCESS	SUBJECT TO DISTRIBUTED AND SCALABLE PRODUCTION	SUBJECT TO CENTRALLY CONTROLLED AND PRESET BATCH PRODUCTION
POTENTIAL	GENERATIVE	CLOSED-ENDED

l'utilizzo di macchine a controllo numerico e senza l'utilizzo di strumentazioni speciali. Queste precondizioni permettono da un punto di vista tecnico che i prodotti open source e i loro derivati possano essere costantemente disponibili per la produzione, in qualsiasi numero, senza investimento in strumentazioni speciali, dovunque e ad opera di chiunque (Kadushin 2010).

Le pre-condizioni descritte nel Manifesto dell'Open design di Kadushin sono diventate i fondamenti di successive operazioni e progetti di design open source nel mondo. Droog Design, noto studio di design olandese, presenta nel 2011 Design for Download, una mostra in cui sono proposti progetti di mobili fai da te e strumenti per la progettazione parametrica di artefatti di design realizzati da Studio Ludens ([studioludens.com](http://studioludens.com)). Nel 2012, durante il Salone Internazionale del Mobile, la rivista Domus presenta la mostra curata da Joseph Grima, *the Future in the Making*, al cui interno è proposta l'esposizione Open Design Archipelago (Domus web 2012), un'esposizione di progetti di design open source sviluppati con tecniche di produzione digitalizzata e elettronica fai da te. Le tre mostre raccontano l'open design in maniera programmatica: c'è l'invito a riflettere su come i designer possano ragionare in maniera indipendente sul modello di produzione e distribuzione del loro prodotto e su come le persone, gli utilizzatori finali, possano intervenire avendo accesso a strumenti semplificati di progettazione e customizzazione (Droog 2011). A partire da queste iniziative sviluppate all'interno del design e a opera dei designer, l'open design può essere considerato come un manifesto programmatico che stimola la riflessione su cosa vuol dire design nell'era di internet e delle tecnologie della comunicazione e dell'informazione, delle tecnologie di produzione digitalizzata. In quanto manifesto programmatico non manca la dimensione politica che è da considerare parte integrante di qualsiasi tentativo di definizione di open design.

Questa condizione culturale che noi definiamo come Adhocracy, ma che potrebbe essere definita in molti altri modi, non è originata all'interno della comunità del design: essa unisce le persone e le aziende, gli eserciti e gli artisti, i designer e i produttori, gli adolescenti e i filosofi. Essa porta un nuovo insieme di strumenti, artefatti, clienti e anche una nuova estetica (Grima 2012, p. 10).<sup>9</sup>

Nel maggio del 2012, ho partecipato al festival LiWoLi – Art Meets Radical Openness, un festival in cui ho incontrato artisti, studiosi e sviluppatori che usano e creano software libero open source

---

9 Mia traduzione.

(FOSS). L'edizione del 2012 ha avuto luogo a Linz (Austria) e il tema era "Swarm, perform, explore – How to sense the City?". Il programma del festival presentava varie attività come conferenze, workshop pratici e performance multimediali. Dopo aver seguito l'Open Design Symposium, ho avuto la possibilità di partecipare al workshop "Objects as Information", un workshop di un giorno riguardo alle possibilità offerte dall'attività di progettazione di oggetti nel mondo della post-fabbricazione CNC e computazionale ([radical-openness.org](http://radical-openness.org)).

I docenti del workshop erano Greg Saul, designer fondatore dello studio Diatom e creatore del progetto Sketchchair ([sketchchair.com](http://sketchchair.com)), un software di progettazione parametrica di una sedia customizzabile, e Ronen Kadushin, designer israeliano citato precedentemente per il suo Manifesto dell'Open design.

Obiettivo del workshop era esplorare il tema degli oggetti come informazione per comprendere le nuove modalità di interazione con un artefatto di design, i cui blueprint sono facilmente condivisibili tramite la rete con altri che possono modificare e adattare per diversi scopi. Lavorando in gruppi di due persone, abbiamo sviluppato dei concept di progetti di design open source cercando di specificare e documentare il processo di design e, in seguito, implementare e produrre parzialmente un artefatto tramite una macchina a taglio laser e infine esplorare come strumenti computazionali possono essere integrati per permettere agli utilizzatori finali di produrre l'oggetto desiderato.

Sviluppando il concept di una radio FM open source le cui manopole per il volume e le frequenze potevano essere customizzate tramite un'applicazione web, ho avuto modo di apprendere alcuni aspetti fondamentali dell'open design: quando si sviluppa un progetto di design open source, il processo di progettazione diventa una parte fondamentale della documentazione che dovrebbe favorire la riproducibilità di un oggetto. La documentazione del processo di progettazione dovrebbe essere connessa al file sorgente dell'oggetto e alle istruzioni per riprodurlo. Documentare il processo sottostante la creazione dell'artefatto è, quindi, fondamentale per permettere ad altri di modificarlo o produrlo.

Questo tipo di attività di documentazione del processo di progettazione open source è stata formalizzata in una proposta

metodologica da Massimo Menichinelli che nel 2011, propone un toolkit per lo sviluppo di *open peer to peer design* (Menichinelli 2011). Il toolkit presenta il termine metadesign, che ritorna molto spesso nella letteratura esistente sull'Open Design e, in particolare, in quella che cerca di mettere in evidenza le implicazioni dell'open source sul ruolo e la figura del designer. Parlando dei computer come macchine ontologiche che danno forma al nostro modo di vedere il mondo e all'emergere di un database culture (Manovich 1998), Jos De Mul suggerisce che il designer, pur non abbandonando il suo ruolo di professionista che definisce l'aspetto materiale e immateriale di un artefatto, si trasforma in una meta-designer (De Mul 2011): egli progetta un multidimensional design space che fornisce all'utente delle interfacce semplificate in modo che possa co-progettare senza avere competenze di design o il tempo necessario per acquisirle.

De Mul utilizza come esempio di metadesigner, il game designer, ma ritengo che tale definizione possa essere ricondotta all'attività dell'interaction designer che progetta artefatti digitali e fisici interattivi e le relazione che le persone hanno con questi artefatti.

Ritengo che l'introduzione di terminologie e definizioni, come il metadesigner, non facilita la comprensione del complesso fenomeno dell'open design. Diversamente è utile tentare di individuare quali sono i gruppi di persone che beneficiano di questo approccio alla progettazione che possono essere classificati in maniera olistica a:

— **chi non ha competenze in design del prodotto e di progettazione meccanica;** se il beneficio materiale portato dall'open design è la diffusione di molteplici soluzioni di design e prodotti nella rete che sono accessibili e modificabili, chi non ha competenze nel design del prodotto può facilmente accedere a librerie di soluzioni esistenti a supporto dello sviluppo di un artefatto fisico. Rientrano in questo gruppo sia persone che realizzano artefatti a livello amatoriale sia professionisti che hanno competenze tecniche. Il contributo del design ai Commons è proprio sul versante delle soluzioni di design funzionale come, per esempio, le librerie per realizzare giunti che offrono la possibilità di produrre degli oggetti basati su moduli.

— **chi non ha competenze di ingegnerizzazione del prodotto;** l'attività del designer riguarda il livello della progettazione di prodotti e servizi e necessita di competenze di ingegnerizzazione per trasformare un progetto in un prodotto adatto alla produzione industriale di massa. L'open design permette la diffusione di soluzioni che sono state già soggette a delle iterazioni sul piano della verifica della loro compatibilità con processi e sistemi di produzione e per questo, chi non ha competenza di ingegnerizzazione del prodotto, ha accesso a soluzioni che sono state precedentemente testate per essere prodotte, per esempio, con macchine a controllo numerico. Il contributo del design ai Commons è il rilascio di soluzioni che sono ad un livello di iterazione tale che possono facilmente essere ingegnerizzate o sono già pronte per essere integrate in uno specifico sistema di produzione.

— **chi ha competenze di design;** chi ha competenze di design e i designer che svolgono attività di progettazione a livello professionale, beneficiano del proliferare di prodotti open source poiché hanno accesso a librerie di soluzioni che possono riutilizzare innumerevoli volte e senza chiedere permessi. In questo senso, l'Open Design “permette di non progettare le cose due volte” (Parvin 2014) e di focalizzare l'attività progettuale sulla creazione di sistemi piuttosto che sull'implementazione di soluzioni funzionali o puramente di stile. Questa prospettiva implica che il contributo del design ai Commons si trasformi in un contributo autoreferenziale in cui la condivisione di risorse di design facilita la pratica del design e stimola la creazione e il rafforzamento della comunità del design stesso.

### 1.2.3 Le tecnologie di produzione digitalizzata

Il passaggio a un'era basata su concetti di collaborazione e *openness* è resa possibile principalmente dal cambiamento dei processi produttivi: la possibilità di usare tecnologie che permettono di progettare attraverso il codice e non attraverso processi analogici e l'esistere di un prodotto come un codice digitale intangibile sono i requisiti che permettono di condividere e collaborare sul progetto di design prima che venga prodotto realmente (Atkinson 2011).

La produzione digitalizzata (*digital manufacturing o fabrication*) è uno degli elementi fondamentali dell'Ecosistema open source. Per digital fabrication si intendono i processi di manifattura di artefatti fisici tramite macchine di produzione che sono controllate da computer (*CNC Computer Numerical Control*). Tramite questi processi è possibile realizzare oggetti di piccole dimensioni, prodotti e anche strutture architettoniche (Iwamoto 2009).

Le tecnologie CNC sono combinate alle tecnologie CAM (*Computer Aided Manufacturing*) che permettono di controllare la pianificazione di una produzione, la gestione del processo, il trasporto e lo stoccaggio dei beni. L'utilizzo di queste tecniche e tecnologie nel processo di produzione di beni ha il suo inizio con la diffusione dei computer stessi, ovvero negli anni 60. Mentre negli USA al MIT e a Cambridge sono sviluppati i primi strumenti di progettazione grafica come Sketchpad, nel 1968 alla Renault, azienda automobilistica francese, Pierre Bézier lavora allo sviluppo di UNISURF CAD CAM, la prima applicazione commerciale per gestire la progettazione delle scocche delle auto (Robben 2013). Bézier implementò l'algoritmo di Paul de Casteljau per generare e modellare curve (le curve di Bézier), un metodo che è rimasto nei comuni software di progettazione grafica contemporanea come per esempio Adobe Illustrator. Da queste applicazioni si è poi evoluto il settore delle tecnologie CAD (*Computer Aided Design*) che hanno permesso di progettare artefatti tramite simulazioni basate su modelli tridimensionali. Le simulazioni di artefatti fisici realizzate tramite software CAD combinati alle tecnologie CAM potevano essere tradotti in linguaggio macchina e riprodotti da una macchina a controllo numerico. La visione di una manifattura completamente digitalizzata, quindi, passa dall'utilizzo di computer, hardware e software, ma è in realtà legata all'idea di automazione della produzione. In ingegneria l'idea di industria automatizzata si concretizza nel paradigma CIM (*Computer Integrated Manufacturing*), ovvero l'automazione dell'intera produzione industriale tramite l'utilizzo di soluzioni robotiche.

Nel 2012 la rivista *The Economist* presenta un rapporto speciale dal titolo *The third industrial revolution* (AA.VV. 2012). Il

rapporto racconta nel dettaglio le implicazioni della crescente digitalizzazione della produzione e della sua robotizzazione. Emblematica è l'illustrazione usata nel rapporto per comunicare il paradigma della terza rivoluzione industriale: un uomo seduto ad una scrivania che da una tastiera di un computer controlla una fabbrica dal quale escono oggetti di diverso tipo, da un'auto, al martello all'aereo. Da questa immagine ne deriva che la *digital fabrication*, analizzata da una prospettiva economica, assume il valore di paradigma innovativo di produzione piuttosto che di un emergente paradigma tecnologico.

La produzione digitalizzata, infatti, è l'infrastruttura di base per invertire il modello di produzione delocalizzata in un modello di produzione distribuito: grazie all'abbattimento dei costi del lavoro dovuto all'integrazione di tecnologie di automazione e alla riduzione della necessità di competenze avanzate da parte dei lavoratori del settore industriale, è possibile produrre beni localmente e creare economie più sostenibili, dal punto di vista sia sociale sia ambientale. Diversamente da *The Economist*, l'economista Jeremy Rifkin mette in evidenza come il paradigma della terza rivoluzione industriale proponga degli scenari di produzione sostenibile che si basano principalmente su due fattori: l'energia e la connettività. La terza rivoluzione industriale secondo Rifkin propone un cambiamento verso un modello di produzione basato su energie totalmente rinnovabili e su internet come infrastruttura di distribuzione di questa energia. In *The zero marginal cost society*, Rifkin aggiunge una nuova dimensione al suo discorso sul cambiamento di paradigmi nell'economia contemporanea, parlando dell'affermarsi dei Social Commons (Rifkin 2014) come modello alternativo al modello economico capitalista.

Il concetto di creazione collaborativa di Commons nell'ambito dello sviluppo tecnologico e della produzione è strettamente connesso ai modelli e alle pratiche dello sviluppo open source dove, appunto, comunità di persone collaborano per creare un prodotto beneficiando di questo processo in maniera collettiva su diversi livelli, dall'accessibilità ad un prodotto altamente customizzato alla creazione di business competitivi su scala globale (p.e. il sistema operativo Linux). Le tecnologie e i modelli di produzione digitalizzata sono, quindi, considerati parte dell'Ecosistema open



Fig. 9 Illustrazione del rapporto speciale di The Economist dal titolo The third industrial revolution

source poiché supportano l'attualizzazione della creazione di *Collaborative Commons* nella produzione dei beni.

Ritorno di nuovo alla tecnologia per spiegare come il paradigma della *digital fabrication* sia strettamente connesso ai modelli di progettazione e sviluppo proposti dall'open source design e hardware. Alle tecnologie e sistemi CAD, CAM e CIM, si aggiungono le tecnologie di prototipazione rapida (*Rapid prototyping*). Queste tecnologie sono per esempio le macchine di manifattura additiva, ovvero le stampanti 3D. La tecnologia di manifattura additiva, specialmente quella di tipo FDM (*Fused Deposition Modeling*) si è largamente diffusa quando a partire dal 2009 i principali brevetti di tecnologie per la stampa 3D basati su Fusion Deposition Modeling iniziano a scadere (Royal Academy of Engineering 2013) e quando si diffondono progetti basati su hardware open source, in particolare Makerbot e RepRap, che hanno aperto il mercato dello sviluppo di stampanti 3D open source.

L'hardware open source è, infatti, uno dei fattori principali che ha favorito la crescente accessibilità delle macchine di prototipazione rapida a basso costo e open source, ma la democratizzazione dell'utilizzo di queste macchine non avrebbe potuto attuarsi senza la diffusione di nuovi concetti legati alla produzione di beni come la *personal fabrication* e degli spazi collaborativi come i fablab (*fabrication laboratory*) e i makerspace (Cavalcanti 2013).

*Personal fabrication* nasce dall'idea che come le persone comuni abbiano avuto accesso al personal computer, queste possano accedere anche a personal fabricator ovvero l'insieme della logica, della sensoristica, degli attuatori e dei display che permettono di creare un dispositivo funzionante.

L'idea di *personal fabricator*, presa in prestito da Neil Gershenfeld dalla fantascienza (i *personal fabricator* erano le macchine della serie Star Trek per riprodurre oggetti e dispositivi di qualsiasi genere) è stata utilizzata alla base di un corso sperimentale tenuto al MIT in cui gli studenti avevano accesso a un laboratorio dotato di un parco macchine per la prototipazione rapida di quasi qualsiasi cosa, ovvero il fablab. Il fablab (*fabrication laboratory*) è diventato un formato di laboratorio che si è diffuso in tutto il mondo in seguito alla volontà di Gershenfeld di sperimentare l'applicazione della *personal fabrication* al di fuori dell'università. I fablab sono a oggi un network di circa 200 laboratori in tutto il mondo che in maniera indipendente e auto-organizzata offrono l'accesso e la formazione sulla realizzazione di artefatti o dispositivi, a partire dalla creazione di un circuito stampato alla realizzazione di uno stampo per una produzione in serie limitata ([fablabs.io](http://fablabs.io)). Il formato del fablab promuove soprattutto l'avvicinamento delle persone non esperte all'apprendimento di tecniche di produzione per veicolare, per esempio, principi di consumo alternativo di prodotti creati a livello locale piuttosto che da catene frammentate di aziende dislocate in tutto il mondo. Da un'idea di laboratorio universitario, il fablab è diventato un movimento in cui sono promossi questo tipo di principi soprattutto in funzione della definizione dal basso del futuro sviluppo tecnologico. Sebbene i principi promossi dal movimento fablab siano l'innovazione, la collaborazione, la produzione locale ad oggi gli esempi in cui essi si atualizzano sono pochi (Troxler 2014). I fablab in alcuni casi possono essere luoghi in cui sono fornite competenze avanzate su temi quali la stampa 3D<sup>10</sup>. I fablab sono spazi di comunicazione di progetti prototipali, laddove per prototipale si intende qualcosa che funziona da strumento per comunicare

---

10 Un esempio è il FabLab Maastricht che sviluppa ricerca in collaborazione con aziende e istituzioni su materiali innovativi per la stampa 3D [fablabmaastricht.nl](http://fablabmaastricht.nl)

le funzionalità e le caratteristiche di un prodotto o i benefici attesi di un progetto complesso. Infine, sono spazi di formazione in cui si offrono corsi sull'elettronica, la programmazione e le tecnologie di digital fabrication e nuovi modelli di formazione basati sul fare ([fablabatschool.org](http://fablabatschool.org)).

La relazione tra le attività promosse in seno al movimento dei fablab con la disciplina del design è a oggi poco definita. Si potrebbe affermare che i designer si confrontano maggiormente con le opportunità tecnologiche ed economiche offerte dal paradigma della manifattura digitale piuttosto che con i valori promossi dal modello dei fablab.

Le opportunità tecnologiche sono principalmente le seguenti:

— **L'attualizzazione dei modelli di produzione non industriale**, ovvero la *digital fabrication* permette di rinforzare dei modelli ibridi che combinano il design con l'artigianato, come per esempio quello dell'autoproduzione che consiste nell'insieme di attività organizzate che hanno lo scopo di materializzare nuovi prodotti-servizi attraverso un processo costituito da auto-orientamento/scelta strategica, auto-progettazione, auto-costruzione, auto-comunicazione, auto-distribuzione (Maffei 2013, pp. 7-13);

— **L'ampliamento della sperimentazione nel design computazionale**, ovvero la *digital fabrication* supporta la definizione e implementazione di nuovi processi di design in cui l'informazione, i dati e gli algoritmi sono degli elementi di un codice e una sintassi che determina nuovi modi di comporre e di produrre (Bohnacker 2012). Le tecnologie di design computazionale combinate alle tecnologie di digital fabrication permettono di implementare strutture e oggetti complessi che imitano le forme organiche della natura, che si basano su strutture matematiche o su dei pattern spaziali. Infine, esse permettono ai designer di sviluppare sistemi parametrici che supportano l'attività di co-progettazione o customizzazione da parte dell'utente finale;

— **la diffusione di una nuova estetica**, ovvero essa facilita la diffusione di un tipo di estetica che può essere definita del prototipo-prodotto. Come negli anni 60 i designer si confron-

tavano con i nuovi materiali e i processi di produzione offerti dall'industria, anche nel contesto della produzione digitalizzata si evidenzia la scelta dei designer di sperimentare e proporre soluzioni che rendono la tecnologia di produzione opaca, ovvero visibile e comprensibile da parte del consumatore. Gli oggetti possono essere prodotti interamente con macchine di prototipazione rapida come, per esempio, le 3D printed headphone dello studio Teague, cuffie audio open source realizzate interamente tramite tecnologia di manifattura additiva al fine di mostrare la possibilità di usare queste tecnologie anche per realizzare prodotti finiti e non solo prototipi (Core jr 2012). Un secondo esempio è il progetto *The machines series* ([machine-series.com](http://machine-series.com)), sviluppato da due designer italiani, Barbara Busatta e Dario Buzzini, che realizzano una linea di accessori da ufficio prodotti tramite la combinazione tra produzione digitalizzata basata su stampa 3D e tecniche di assemblaggio artigianali;

La più grande sfida riguarda la risoluzione e precisione del processo di stampa. Oggi e per ancora alcuni anni, sarà difficile raggiungere la qualità e raffinatezza della stampa ad iniezione. Riconoscendo questo limite abbiamo deciso di progettare sulla base di esso. Abbiamo creato una sezione trasversale con una linea sinuosa che ci ha permesso di fare in modo che scomparisse e, inoltre, abbiamo fatto in modo che le cuciture del percorso utensile scomparissero creando una superficie elegante e sofisticata seppur irregolare (Khemsuro 2014).<sup>11</sup>

La dimensione dell'estetica influenzata dalle tecnologie di produzione digitalizzata e di prototipazione rapida è ad uno stato embrionale. Nel caso dell'utilizzo della stampa 3D ci troviamo di fronte ad un'estetica che diventerà obsoleta, ma con la sperimentazione da parte di designer si definirà un linguaggio formale più maturo (Warnier et al 2014) che riporterà la tecnologia ad un livello trasparente, ovvero nascosto, per favorire il livello simbolico, emozionale e funzionale.

---

11 Khemsuro 2014, mia traduzione.



Fig. 10 3D printed  
headphone di studio Teague



Fig. 11 The machine series  
di Barbara Busatta e Dario  
Buzzini

#### 1.2.4 Il fai da te tecnologico Comprendere le motivazioni dei futuri utilizzatori

The maker is always plural (Dieter & Lovink 2014)

L'espressione Do it yourself (DIY) definisce una serie di attività creative in cui le persone, usano, modificano materiali per produrre diverse tipologie di artefatti per scopi diversi. Le tecniche per svolgere queste attività sono a volte codificate e condivise in modo che le persone possano riprodurle, interpretarle o estenderle (Buechley 2009). Negli ultimi due decenni, l'integrazione tra l'utilizzo di computer per la comunicazione, gli strumenti di pubblicazione on-line web 2.0 e le tecnologie web di collaborazione ha favorito la rinascita del fenomeno del DIY, in particolare, di quello legato all'utilizzo dell'elettronica e della sensoristica. Esistono migliaia di comunità del DIY la cui interazione va dal puro scambio di informazioni su come riprodurre degli artefatti fino alla vendita dei prodotti realizzati in modo fai da te: un esempio di successo è Instructables che fornisce risorse per la costruzione di oggetti e dispositivi tecnologici tramite guide passo-passo pubblicate dalle persone che realizzano progetti di ogni tipo (Kuznetsov & Paulos 2010). Nell'ambito di questo studio mi concentro sul DIY elettronico o tecnologico, ovvero quelle pratiche rivolte allo sviluppo a livello amatoriale di artefatti dotati di componenti elettroniche e tecnologiche. Questo tipo di DIY occupa un ruolo fondamentale in quello che ho definito "Ecosistema open source" poiché, da un lato, propone delle pratiche di creazione di artefatti che fungono da fondamento allo sviluppo di progetti open source e, dall'altro, fornisce una prospettiva culturale e politica per interpretare il DIY e l'open source stesso. Le pratiche-fondamento riguardano l'attività stessa di sviluppo di un progetto amatoriale in modo fai da te supportato dall'utilizzo di internet e delle tecnologie di collaborazione: tramite l'accesso a risorse gratuite sulla rete, le persone possono realizzare dispositivi tecnologici di piccola e media complessità per soddisfare un bisogno di tipo personale; possono entrare in contatto con altre persone che condividono lo stesso interesse e la stessa necessità con i quali scambiano informazioni e interagiscono creando

una comunità di interesse. La prospettiva culturale e politica riguarda, invece, le motivazioni che possono essere associate alla volontà di rispondere in maniera attiva ai modelli di consumo delineati dalla produzione globalizzata, di sviluppare competenze per promuovere la cultura del fare e il senso di appropriazione della tecnologia.

Mentre Sennett enfatizza il fare come concretizzazione del raggiungimento della qualità (Sennet 2008), la cultura del fare promosso nelle comunità DIY è interpretata da un punto di vista politico come modello al quale aspirare in futuro poiché:

Creare è il modo più potente per risolvere problemi, esprimere idee e dare forma al nostro mondo. Cosa e come creiamo definisce quello che siamo e comunica chi vogliamo essere. Per molte persone, creare è fondamentale per la sopravvivenza. Per altri, è una vocazione: un modo di pensare, inventare e innovare. E per alcuni, è semplicemente un piacere poter essere capaci di dare forma a un materiale e dire l'ho fatto io. Il poter del creare è ciò che soddisfa ognuno di questi essenziali bisogni e desideri umani (Charny 2012).<sup>12</sup>

Le motivazioni alla base della realizzazione di un progetto fai da te sono un elemento fondamentale per comprendere alcuni fenomeni contemporanei legati al ritorno della cultura del DIY e del fare pratico.

Nel suo studio su human computer interaction e il DIY, Obrist (Obrist 2008) individua sette livelli motivazionali alla base della modifica di un prodotto in modo fai da te:

- *personal motivated modification*, ovvero le modifiche che dipendendo dagli umori e dalle emozioni o da situazioni dipendenti da circostanze sociali (p.e. se si è in compagnia o si è da soli);
- *functional motivated modification*, ovvero le motivazioni legate ad una necessità funzionale connessa ad un compito specifico, che dipendono dalle competenze o dalla volontà di sviluppare una soluzione ad un problema dato;
- *aesthetic/design modification*, ovvero le modifiche dell'apparenza visiva e fisica di un sistema che dipendono dalla creatività personale dell'utente finale;

---

<sup>12</sup> Charny 2012, mia traduzione.

- *system/product motivated modification*, ovvero le modifiche sul piano della complessità tecnologica di un sistema;
- *knowledge/skill motivated modification*, ovvero le modifiche che si basano sulla consapevolezza di avere competenze per applicare tali modifiche;
- *social community motivated modification*, le modifiche derivate dall'influenza di altre persone o dai membri di comunità di interesse.

Con una prospettiva di tipo storico, Atkinson (Atkinson 2006), invece, propone dei livelli motivazionali individuando diverse aree distintive del DIY. Il *Reactive DIY* propone attività che consistono nella realizzazione di artefatti fatti a mano a livello amatoriale mediata dall'utilizzo di kit, modelli e schemi e dall'assemblaggio di componenti in cui la motivazione spazia tra l'occupazione del tempo libero al piacere personale (che potrebbe includere l'aspetto di remunerazione finanziaria derivata dalla vendita dell'oggetto realizzato). La seconda area è *Essential DIY*, ovvero le attività di mantenimento e miglioramento dell'ambiente domestico spinte da una necessità di tipo economico (risparmio) o dalla mancanza di lavoro professionale (disoccupazione). Queste attività dipendono dall'utilizzo di manuali con istruzioni. Infine nel *Lifestyle DIY* le attività di miglioramento o costruzione della casa tramite lo sviluppo di un lavoro in modo fai da te è una scelta piuttosto che un bisogno e include l'intervento di professionisti come consulenti esterni alla progettazione.

Atkinson sottolinea come il DIY è da considerare un agente di democratizzazione poiché permette alle persone di avere l'indipendenza dal supporto di professionisti, diffonde l'adozione di principi di design modernista e permette di sviluppare ambienti e oggetti basati sull'identità delle persone. Le attività DIY sono, secondo questa prospettiva, un mezzo per liberare il consumatore da editti di design prestabiliti, per favorire la manifestazione del gusto personale, per misurare l'estetica popolare del design che è una parte intrinseca della cultura materiale del nostro vivere quotidiano.

In *An in-depth profile of Makers at the Forefront of Hardware Innovation* (Makezine 2014), un progetto di ricerca supportato dal

Intel e Make, la rivista fondata nel 2005 da Dale Dougherty, emerge, invece, il ritratto della comunità dei maker, ovvero la comunità che estende e rafforza le attività e i valori della cultura DIY, inglobandola e avvicinandola all'ambito dell'innovazione tecnologica. Secondo questa ricerca, il maggior numero di persone che si dedica a progetti di fai da te tecnologico si definiscono principalmente ingegneri. Quest'aspetto cambia totalmente l'interpretazione del DIY come un ambito in cui operano solo persone non esperte che acquisiscono competenze tramite la riproduzione di artefatti esistenti e condivisi da altre persone. Nella comunità maker ci sono, infatti, persone che hanno competenza di sviluppo sia software sia hardware le cui motivazioni alla base della realizzazione di un progetto sono legate alla possibilità di acquisire competenze, di poter interagire con prodotti che si possano riparare e che siano più economici rispetto a quelli che sono disponibili sul mercato. La principale soddisfazione di queste persone risiede nella possibilità di accedere in maniera autonoma a delle risorse per sviluppare dei dispositivi tecnologici e al conseguente senso di appropriazione e potere che hanno sulla tecnologia.

La cultura promossa dal DIY e dal movimento maker è un elemento di interesse per definire l'“Ecosistema open source” poiché introducono nuovi requisiti per la progettazione di prodotti e servizi. Essi promuovono l'abbassamento delle barriere tecnologiche da parte di utenti non esperti favorendo la partecipazione allo sviluppo e l'utilizzo di artefatti open source. Nasce l'idea di prodotti *maker-friendly* che sono artefatti le cui informazioni dettagliate sono accessibili poiché:

I maker vogliono accedere ai dettagli e al prodotto stesso. Vogliono guardare all'interno e vedere le parti in movimento. Vogliono fare quello che sarebbe imprevisto come riparare o migliorare, e anche prendere i componenti una volta non è più utilizzato (Dougherty 2014).<sup>13</sup>

Le comunità coinvolte nel mondo maker e DIY Favoriscono la definizione di pratiche condivise come, per esempio, le regole di

---

13 Dougherty 2014, mia traduzione.

base per la creazione di un artefatto come la facilità di apertura dello chassis, l'integrazione della lista delle parti necessarie per riprodurre l'oggetto, l'utilizzo di strumenti comuni di produzione, l'uso di componenti che si possono sostituire con altri disponibili sul mercato (Makezine 2006) Queste pratiche permettono di creare degli standard di documentazione dei progetti che sono condivisi sia da esperti di sviluppo software e hardware e designer sia dagli utilizzatori finali di un prodotto.

Fig. 12 La carta dei diritti del Maker (The Maker's bill of rights) di Make Magazine.

makezine.com

# THE MAKER'S BILL OF RIGHTS

- Meaningful and specific parts lists shall be included.
- Cases shall be easy to open. ■ Batteries shall be replaceable. ■ Special tools are allowed only for darn good reasons. ■ Profiting by selling expensive special tools is wrong, and not making special tools available is even worse. ■ Torx is OK; tamperproof is rarely OK.
- Components, not entire subassemblies, shall be replaceable. ■ Consumables, like fuses and filters, shall be easy to access. ■ Circuit boards shall be commented.
- Power from USB is good; power from proprietary power adapters is bad. ■ Standard connectors shall have pinouts defined. ■ If it snaps shut, it shall snap open. ■ Screws better than glues. ■ Docs and drivers shall have permalinks and shall reside for all perpetuity at archive.org. ■ Ease of repair shall be a design ideal, not an afterthought. ■ Metric or standard, not both.
- Schematics shall be included.

Drafted by Mister Jalopy, with assistance from Phillip Torrone and Simon Hill.

**Make:**  
technology on your time

Platform21's

# Repair Manifesto

- 1. Make your products live longer!**  
Repairing means taking the opportunity to give your product a second life. Don't ditch it, stitch it! Don't end it, mend it! Repairing is not anti-consumption. It is anti- needlessly throwing things away.
- 2. Things should be designed so that they can be repaired.**  
Product designers: Make your products repairable. Share clear, understandable information about DIY repairs.  
Consumers: Buy things you know can be repaired, or else find out why they don't exist. Be critical and inquisitive.
- 3. Repair is not replacement.**  
Replacement is throwing away the broken bit. This is NOT the kind of repair that we're talking about.
- 4. What doesn't kill it makes it stronger.**  
Every time we repair something, we add to its potential, its history, its soul and its inherent beauty.
- 5. Repairing is a creative challenge.**  
Making repairs is good for the imagination. Using new techniques, tools and materials ushers in possibility rather than dead ends.
- 6. Repair survives fashion.**  
Repair is not about styling or trends. There are no due-dates for repairable items.
- 7. To repair is to discover.**  
As you fix objects, you'll learn amazing things about how they actually work. Or don't work.
- 8. Repair – even in good times!**  
If you think this manifesto has to do with the recession, forget it. This isn't about money, it's about a mentality.
- 9. Repaired things are unique.**  
Even fakes become originals when you repair them.
- 10. Repairing is about independence.**  
Don't be a slave to technology – be its master. If it's broken, fix it and make it better. And if you're a master, empower others.
- 11. You can repair anything, even a plastic bag.**  
But we'd recommend getting a bag that will last longer, and then repairing it if necessary.

**Stop Recycling. Start Repairing.**

[www.platform21.nl](http://www.platform21.nl)

Fig. 13 Il manifesto della riparazione (Repair Manifesto) del progetto Platform21

Stimolano la crescita di un mercato rivolto a nuove tipologie di prodotti tecnologici. In particolare, grazie a questi movimenti la crescita della domanda di nuovi dispositivi permette di produrre strumenti e tecnologie a basso costo e istituire dei business profittevoli basati su hardware e design open source. Infine, stimolano la diffusione di competenze tecnologiche tra utilizzatori di artefatti interattivi che permettono di creare un settore di mercato per il quale ideare, progettare e produrre nuovi prodotti che sono programmabili e altamente customizzabili.

### 1.2.5 Le licenze open source per il design

L'ultimo elemento dell'“Ecosistema open source” è rappresentato dalle licenze libere adottate per la distribuzione di software, hardware e design. Le licenze libere sono l'elemento che permette di differenziare ciò che è rilasciato in open source da ciò che non lo è. È infatti essenziale applicare una licenza per definire che tipo di utilizzo è concesso alle persone che scaricano la documentazione e il file sorgente dalla rete. La questione dell'adozione delle licenze è da considerare ancora aperta per quello che riguarda gli aspetti di protezione del design. Nell'ambito dello sviluppo di open source hardware sono stati adottati diversi tipi di licenze, quali la TAPR Open Hardware License (Tucson Amateur Packet Radio) (TAPR 2014) e CERN Open Hardware Licence (OHWR 2015). La piattaforma software hardware Arduino utilizza, per esempio, le licenze Creative Commons che sono anche quelle più diffuse tra i progetti di open design. Le licenze Creative Commons nascono in risposta alla digitalizzazione di contenuti e la loro distribuzione sulla rete offrendo delle soluzioni legali che definiscono il permesso al riutilizzo di una risorsa pur rispettando il diritto d'autore.

Da un punto di vista legale, le licenze Creative Commons non sono da considerare alternative al sistema dei diritti di design (Jasserand 2011) che sono stati regolamentati dalla Comunità Europea tramite delle direttive definite alla fine degli anni novanta. Le licenze Creative Commons sono di norma utilizzate per specificare i diritti di riproduzione e sono, per questo, incompatibili con l'utilizzo dei diritti di design, ovvero i design registrati e i design non registrati. Le licenze valide per rilasciare un progetto di

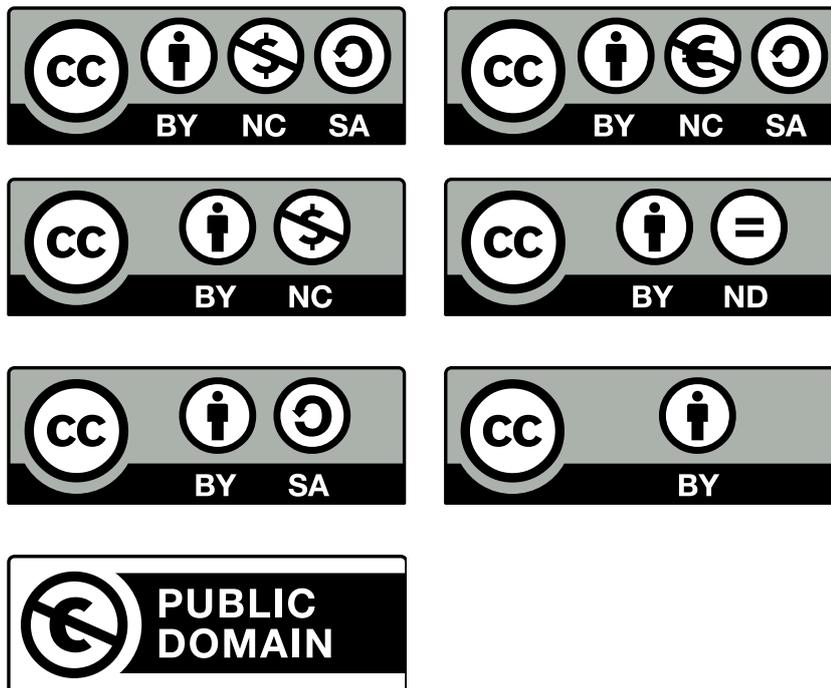


Fig. 14 Le licenze Creative Commons.

Open Design è CC0 che è utilizzata per condividere un design nel pubblico dominio dove ogni utilizzo anche a livello commerciale è concesso e senza richiesta di un permesso, e la licenza CC-BY (Attribuzione) e CC-BY-SA (Attribuzione condividi allo stesso modo) che servono per proteggere i diritti d'autore lasciando liberi gli utilizzi per la creazione di derivati a scopo commerciale. Tutte le licenze che limitano la creazione di derivati e che impongono l'uso non commerciale non sono compatibili con la definizione di Open Source (Perens 1999) e per questo non sono applicabili a progetti che puntano ad adottare un modello di sviluppo open source (Ladyada 2011).

Recentemente sono state proposte dei nuovi protocolli delle licenze Creative Commons che cercano di integrare i diritti di design e il diritto di autore: Licenze CC+. Queste licenze permettono di completare una licenza Creative Commons attraverso l'aggiunta di un testo che dichiara e specifica cosa è possibile o non è possibile fare con un artefatto di design la cui documentazione e file sorgente sono stati resi disponibili sulla rete (Wiki Creative Commons 2014).

### 1.3 Open source nel design: nuovi paradigmi o attualizzazione di manifesti pioneristici del design?

Tra la fine degli anni cinquanta e l'inizio degli anni sessanta, l'artista ungherese Victor Vasarely realizzò *Planetary Folklore Participations*, una serie di opere d'arte che le persone potevano acquistare come kit e assemblarle in modo fai da te. Le opere erano composte da blocchi colorati in plastica che potevano essere combinati per creare un numero infinito di composizioni e immagini. Ogni opera era venduta in un packaging, simile a quello dei giochi da tavolo, che conteneva i componenti del kit e le istruzioni all'uso, ovvero un manuale che spiegava le regole dell'alfabeto visivo progettato da Vasarely. La denominazione *Planetary folklore* era stata coniata da Vasarely per descrivere il suo lavoro e la sua visione di estetica universale per le masse che si basava su una metodologia didattica brevettata dallo stesso Vasarely nel marzo del 1959: lo spettatore, tramite i messaggi codificati dall'artista, comprende le basi e gli aspetti concettuali connessi alla prospettiva dell'artista sull'arte e l'architettura (Vasarely 1973). Influenzato dalle scoperte della psicologia gestaltica, della cibernetica e dell'astrofisica, Vasarely sviluppò una ricerca incentrata sull'introduzione di nuovi modi e concetti per la democratizzazione dell'arte: in particolare realizzò due serie di opere, *l'Alfabeto plastico* e *Folklore planetario*, con l'obiettivo di sperimentare la possibilità di ottenere questa democratizzazione attraverso la realizzazione di opere basate sulla percezione dello spettatore piuttosto che sulla conoscenza di nozioni teoriche. Vasarely cercò di attaccare le forme tipiche del mercato dell'arte attraverso l'avvicinamento dell'oggetto d'arte all'ambito della produzione di massa, accessibile rispetto a ogni forma di stato sociale e ricchezza.

La serie *Alfabeto plastico* consisteva in una griglia che stabiliva una relazione modulare tra le forme e i colori. I quadri erano basati su quindici forme primitive derivate dal cerchio, il quadrato e il rettangolo: le variazioni di queste forme erano dipinte usando colori con venti tonalità differenti. Ogni unità della griglia, quella che Vasarely definì unità plastica, aveva un retro e un fronte.

L'obiettivo della ricerca di Vasarely era la creazione di un linguaggio di programmazione visuale *ante litteram* che per-



Fig. 15 Victor Vasarely,  
Planetary Folklore  
Participations No.1, 1969.

mettesse di generare infinite variazioni di forme e colori al fine di creare delle opere uniche. Riconoscendo che l'evoluzione del suo linguaggio plastico lo aveva portato alla realizzazione di opere basate sulla programmazione di colori, tonalità e forme, Vasarely immaginò una macchina cibernetica che volle realizzare in collaborazione con IBM California nel contesto del programma curato da Maurice Tuchman, curatore per l'arte moderna al Los Angeles County Museum of Art (LACMA). Si trattava di uno schermo lumino-cinetico azionato da un sistema elettronico e controllabile tramite dei dispositivi di regolazione della luminosità (*dimmer*).

Sogno la creazione di una macchina in collaborazione con gli specialisti dell'elettronica. Un lavoro impressionante e oneroso. I primi contatti con quell'ambiente sono incoraggianti, ma la collaborazione è difficile da ottenere e i tempi sono lunghi. Di che cosa si tratta? di un grande schermo lumino-cinetico capace di rendere migliaia di combinazioni di colore. In pratica si tratta di una scatola metallica di 312,5 x 312,5 cm con una profondità dai 10 ai 20 cm in base alla necessità. Questa scatola è suddivisa in una griglia di 625 compartimenti. Ogni compartimento di 12,5 x 12,5 cm. Ogni compartimento contiene un cerchio di circa 10

cm. Questa griglia, come i cerchi che devono essere costruiti di lamiera metalliche solide ma molto sottili dell'ordine di un decimo di millimetro. Questa struttura conterrà il dispositivo elettronico multicolore, azionato elettronicamente e completato da un sistema di dimmer. Lascio sei colori: rosso, blu, verde, viola, giallo e grigio. Ogni colore è suddiviso in 12 tonalità che permettono di generare 72 variazioni di colore. Fa in modo che ogni compartimento e i loro cerchi possano rendere in modo alternato 72 tonalità. Esse sono l'isolamento perfetto di quei 625 compartimenti (e dei 625 cerchi connessi a essi) e uno schermo opaco che assicura una buona visibilità delle tonalità (Smith 1972).

Grazie a questo macchinario, secondo Vasarely, è possibile ricreare tutte le sue opere e realizzare delle varianti nuove e infinite. La macchina produce un repertorio riutilizzabile di immagini che possono essere utilizzate da parte degli artisti per realizzare filmati, scenografie per il cinema e la televisione. L'applicazione di un principio cibernetico nell'arte ha per Vasarely i risultati più interessanti nel dominio della psicologia sperimentale e nella possibilità di portare l'arte nel circuito della produzione di massa (Vasarely 1970).<sup>14</sup>

Le opere e le teorie di Vasarely si inseriscono in un periodo in cui il discorso artistico focalizza la sua attenzione su temi quali la partecipazione delle persone, anticipando da un lato i temi propri dell'arte multimediale e interattiva, dall'altro il passaggio da una cultura dell'artigianato alla cultura della produzione industriale introducendo l'idea di partecipazione delle persone al processo di creazione di varianti di oggetti progettati e pensati per la produzione seriale.

Riconoscendo nell'uso delle tecnologie, in particolare quelle industriali e informatiche, un veicolo di traslazione e traduzione dell'idea e della tecnica d'arte piuttosto che una minaccia all'unicità e al valore dell'opera, in una serie di note dal 1954 al 1963 Vasarely ci parla della diffusione dei multipli, ovvero la ricreazione a base di costanti di innumerevoli versioni di un'opera d'arte unica. Descrivendo la differenza tra il concetto di riproduzione e ricreazione, Vasarely descrive una prima definizione di multiplo d'arte sottolineando come esso, in quanto ricreazione, sia un'opera autentica la cui realizzazione è guidata dall'artista e realizzata con

---

14 Vasarely 1970, mia traduzione.

l'aiuto di tecnici artigiani. In questa prima definizione di multiplo d'arte riconosco l'inizio del discorso sulla riproducibilità partecipata di un artefatto che caratterizza l'incontro tra le discipline del progetto e le pratiche diffuse a partire dall'emergere dell'attuale paradigma open source nel design.

Alla fine degli anni sessanta, dieci anni dopo le prime sperimentazioni di Vasarely su oggetti d'arte replicabili, a basso costo e fai da te, Bruno Munari, artista, designer e teorico italiano, propone la sua definizione di multiplo tramite un manifesto:

I multipli sono oggetti a due o più dimensioni progettati per essere prodotti in un numero limitato o illimitato di esemplari, allo scopo di comunicare per via visiva un'informazione di carattere estetico ad un pubblico vasto e indifferenziato (Munari 1971, p. 89).

Munari teorizza il Manifesto dei multipli nel 1968, in seguito ad un lavoro sull'oggetto moltiplicato che aveva avuto inizio a partire dagli anni 50. Già nel 1959, infatti, Munari crea le *Strutture continue* per le edizioni MAT (Multiplication d'art Transformable) collaborando al progetto dell'artista svizzero Daniel Spoerri. L'Edizione MAT era un progetto la cui prima mostra ebbe luogo a Parigi nel 1959 nella Galleria Edouard Loeb in rue de Rennes. Nella mostra erano esposte opere realizzate da Agam, Marcel Duchamp, Dieter Roth, J.R. Soto, Tinguely, Vasarely e altri e consistevano in multipli, ovvero oggetti originali moltiplicati e cinetici (Ballo 1976), riproduzioni seriali di un pezzo unico originale. La caratteristica del cinetismo nei multipli dell'Edizione MAT costituiva un aspetto fondamentale poiché permetteva a chi acquistava un'opera di un'edizione limitata di avere un'opera unica grazie alla variazione data dal movimento.

Munari espande l'operazione avviata da Spoerri nel suo manifesto arrivando a indicare come i multipli siano diversi dalle riproduzioni d'arte, ovvero non sono una riproduzione meccanica di un'opera artigianale realizzata a mano, ma opere moltiplicate realizzate per trasmettere informazioni di carattere estetico, per

Fig. 16 Object to be destroyed, Man Ray, 1959



trasferire nozioni di base di estetica, di percezione cromatica, di fenomeni e problemi che non potrebbero essere esplorati con altre tecniche di progettazione basate, per esempio, sul disegno. Anche nella definizione dei multipli di Munari, la partecipazione del pubblico è un aspetto fondamentale del funzionamento dell'opera poiché solo tramite la manipolazione del multiplo le persone possono fare esperienza e memorizzare il fenomeno co-

municato dal multiplo al fine di capire come funziona il mondo che lo circonda.

L'aspetto di democratizzazione dell'oggetto d'arte, ovvero il suo ruolo di rappresentazione che permette l'accesso diretto alla conoscenza astratta da parte del pubblico è anch'essa parte dell'idea di multiplo così come definito da Munari. I multipli per Munari si realizzano in serie per permettere al largo pubblico di arricchire la propria cultura visiva. Se per Vasarely l'opera democratizza l'accesso alla prospettiva dell'artista sul mondo, gli oggetti moltiplicati di Munari sono riproduzioni dei fenomeni fisici che spiegano il mondo.

La serie dei multipli per Munari può essere limitata o illimitata in base al tipo di informazione da trasmettere e alle tecnologie a disposizione e non si basa sull'uso di materiali preziosi al fine di ottenere un rapporto tra il valore dell'opera e il prezzo di mercato che sia accessibile a molti. L'accessibilità è un termine chiave che porta in maniera naturale l'accostamento delle creazioni dei multipli al mondo del design e della produzione industriale. L'idea alla base dei multipli come oggetti d'arte accessibili al largo pubblico, infatti, si realizza grazie all'emblematica esperienza italiana del connubio tra arte, design e industria. Danese, una delle storiche fabbriche del design italiane fondata da Bruno Danese e la moglie Jacqueline Vodoz, si propose come un laboratorio pionieristico per la messa in produzione di multipli d'arte realizzati, in particolare, da quello che fu definita l'ultima avanguardia, ovvero il Movimento d'Arte Programmata e Cinetica di cui Munari fu uno dei massimi esponenti (Vergine 1983). Bruno Munari propose con Danese diversi multipli d'arte come, per esempio, il multiplo della macchina aerea che realizzò nel 1930 e che rappresenta uno dei primi multipli della storia dell'arte, riproposto da Danese nel 1972 in una tiratura di 10 esemplari.

L'oggetto, costruito in legno e metallo, era alto circa un metro e ottanta centimetri, largo circa sessanta per trenta. Le sfere erano rosse, meno una piccola nera, tutte le bacchette erano bianche. Appeso con una piccola corda al soffitto di un ambiente, si muoveva lentamente, spinto da qualche corrente d'aria. Era come una costellazione, come un gruppo di atomi, o, come si potrebbe dire

oggi, una stazione spaziale. Nelle gallerie d'arte nessuno lo voleva esporre perché non era né pittura né scultura. Dopo essere stato appeso per qualche anno nel mio studio di via Ravizza a Milano, andò distrutto in un trasloco. Dall'osservazione del comportamento di questa prima e unica macchina aerea, nacquero in seguito le macchine inutili (Munari 1971, p. 7).

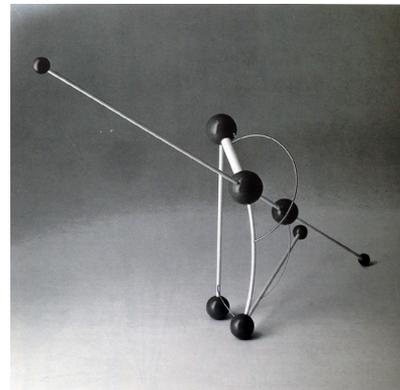
Famoso è il multiplo del 1968 Flexy, un'opera che si basa su una struttura di filamenti metallici flessibili uniti da giunti la cui forma può essere manipolata e modificata, riprodotta da Danese.

Vasarely parla dei multipli come una forma di continuità del pensiero dell'artista e del valore dell'opera, Munari si addentra nella definizione di veri e propri principi progettuali per la corretta realizzazione di un multiplo:

I multipli sono progettati con il metodo di design di ricerca. Il progettista di un multiplo non fa, come l'artista, un bozzetto meraviglioso e poi lo fa riprodurre in qualche modo tecnico: egli fa un esperimento su un fenomeno ottico, fisico, geometrico, topologico, meccanico...ne mette a punto gli elementi di evidenza

Fig. 17 Macchina aerea, Bruno Munari, 1930

Fig. 18 Macchina aerea, Bruno Munari, edizione Danese 1971



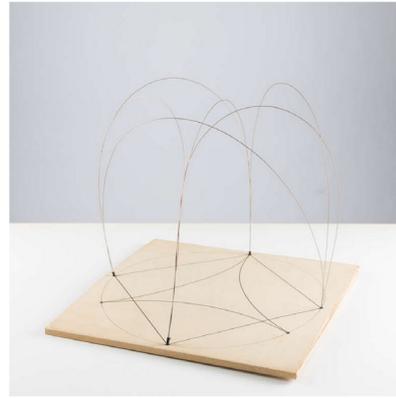


Fig. 19 Bruno Munari,  
Ora X, 1963

Fig. 20 Bruno Munari,  
Flexy, Danese, 1968

comunicativa studia il materiale migliore per realizzare l'oggetto con il massimo di comunicazione visiva e il minimo costo. Trova una tecnica meccanica la più adatta allo scopo e alla fine ne nasce un prototipo che non è il pezzo unico dell'artista, ma è il modello dal quale nascerà la serie (Munari 1971, p. 90).

La guida di Munari su come si realizzano i multipli pone l'accento sulla fase di prototipazione come vero e proprio metodo di design di ricerca finalizzata alla realizzazione di oggetti moltiplicati, che sono variazioni dell'oggetto d'arte ingegnerizzati per produzione in serie limitata o illimitata. Partendo dall'analisi di un fenomeno e dalle iterazioni attraverso la sperimentazione con materiali e tecnologie, questo metodo anticipa le pratiche e gli approcci della progettazione di sistemi interattivi, in particolare, rispetto a come si stanno configurando in base all'impatto delle tecnologie dell'informazione e della comunicazione, dell'accessibilità a piattaforme di prototipazione software e hardware e di produzione. I multipli di Munari in questo senso sono il secondo esempio di riferimento per inquadrare la relazione tra le pratiche proprie del design e le pratiche e gli approcci del paradigma open source.

Il terzo elemento che connette l'esperienza dei multipli d'arte alla progettazione open source è l'Arte programmata. Arte programmata è la definizione data al corpo di opere di un gruppo di artisti italiani attivi tra la fine degli anni cinquanta e i primi anni sessanta. La definizione fu introdotta da Bruno Munari e Umberto Eco nel dicembre 1961 nell'Almanacco Letterario Bompiani e poi nella mostra promossa presso lo showroom Olivetti a Milano nel 1962, dove furono esposti i lavori di Munari, Enzo Mari e degli

artisti del Gruppo T (Giovanni Anceschi, Davide Boriani, Gianni Colombo, Gabriele De Vecchi, Grazia Varisco) e del gruppo N. Gli artisti di questo movimento realizzarono oggetti attraverso l'applicazione di processi analoghi a quelli della ricerca tecnologica e di design, ovvero creando prototipi che erano poi riprodotti come serie di artefatti varianti così come Munari aveva teorizzato nel suo manifesto dei multipli.

Secondo Munari le opere del movimento di arte cinetica e programmata atualizzano i concetti legati all'opera aperta, all'opera cinetica e all'opera programmata poiché esse sono aperte in modo diretto e non virtuale poiché l'occhio dello spettatore non è più intermediario tra l'opera e i concetti che rappresenta, ma è elemento attivo a supporto dell'apprendimento e della scoperta dell'opera. Il cinetismo, inoltre, è il modo per rendere la mutevolezza e non lo stadio intermedio per creare una composizione fissa mentre la programmazione determina la trasformazione dell'opera nel tempo secondo situazioni che sono imprevedibili e non previste da parte dell'autore (Vinca Masini 1965).

L'arte cinetica e programmata segna il passaggio dal pensiero meccanico a quello elettronico e, infatti, le opere degli artisti del Gruppo T rappresentano un punto di riferimento chiave per interpretare un tipo di arte che gettò le basi per lo sviluppo dell'arte interattiva e di molti concetti che sono diventati propri della disciplina dell'interaction design. Lavorando alla creazione di ambienti e artefatti interattivi tramite meccanismi cinetici ed effetti ottici e combinando design, arte e tecnologia, il Gruppo T propose nuovi codici di lettura dell'arte in cui gli spettatori diventarono utenti e co-autori dell'opera partecipando all'atto creativo con l'azione diretta del loro corpo. È l'utente che può usare l'opera a modo suo, che ne diventa co-autore; è l'utente che viene messo programmaticamente al centro dell'opera.

Nel lavoro dei collettivi degli anni sessanta di arte cinetica e programmata e, in particolare, nell'esperienza del Gruppo T si estende e si modifica l'idea della partecipazione delle persone che si sposta verso il concetto di interazione come partecipazione dello spettatore al completamento dell'opera tramite il corpo. L'interattività è la direzione che i membri del Gruppo T scelgono anche

in relazione ad una posizione e un messaggio politico in cui il fare, l'agire partecipando è il contributo per migliorare e modificare la società. L'interattività è anche la scelta di un rifiuto verso i mezzi della pittura informale, filone di inizio della formazione artistica, che non permettevano di stimolare la partecipazione dello spettatore, intesa come intervento attivo dello spettatore che diventa fruitore e non mero contemplatore dell'opera.

L'artista sente l'urgenza di arretrare dalla sua posizione elitaria per diventare progettista di 'semilavorati' destinati ad essere usati e completati da chi li manipola. Il gesto creatore dell'artista, consolidato da una lunga tradizione, lascia il posto al gesto di un pubblico attivamente partecipante. Negli ambienti del Gruppo T partecipazione è prendere parte (*partem capere*), diventare parte dello spazio stesso dell'opera, usando il corpo in movimento come strumento di esplorazione attiva e costruttiva, istanza conoscitiva privilegiata (Devecchi 2013).

Il lavoro del Gruppo T attualizza la trasformazione dell'artista in un progettista di semilavorati che sono completati dal pubblico attraverso l'interazione del corpo. Questo cambiamento di prospettiva sull'idea di partecipazione come interazione del corpo dell'utente per il completamento dell'opera si distingue dall'idea di partecipazione come intesa sia nella creazione dell'opera di Vasarely tramite un codice dato, l'alfabeto plastico, sia nella realizzazione dei multipli così come teorizzati da Munari. Il filo conduttore rimane però la trasformazione del ruolo dell'artista: Argan collega il concetto di arte programmata alla contrapposizione tra progetto e non progetto. Nel secondo caso, si tratta dell'opera d'arte così come concepita della Pop Art, ovvero la trasformazione dell'oggetto comune in opere; nel primo caso, quello dei programmati, non c'è culto dell'oggetto ma cultura del progetto inteso come programma degli stati dell'opera.

Secondo l'ipotesi teorica di Argan (Argan 1964), il progetto permette all'opera di essere ricostruita anche in assenza dell'autore, il progetto prevede la definizione di tecniche e materiali necessari al raggiungimento dello scopo, il progetto proietta il fare in una dimensione costruttiva e non più rappresentativa, lontana dal procedere casuale dell'artista ispirato e molto più vicina a quello di

un tecnico. Anche per questo motivo l'artista smette di chiamarsi artista per diventare operatore.

Alla luce del pensiero di Argan, si può affermare che l'Arte programmata abbia attivato un discorso nell'arte che getta le basi teoriche per interpretare l'interazione tra design e paradigma open source in cui la riproduzione di un artefatto avviene in assenza dell'autore e secondo dinamiche collaborative e in cui il designer è il facilitatore di tale processo piuttosto che l'autore assoluto di un'opera.

Nella produzione del Gruppo T troviamo l'attualizzazione di tale pensiero e discorso.

Le opere che il Gruppo T produce si chiamano *Miriorama* che vuol dire infinite visioni (dal greco *orao*, vedere, e *myrio*, che indicava una quantità pressoché infinita). Questo nome sintetizza gli aspetti chiave della produzione del gruppo T in cui la ricerca di variazione nell'unicità dell'opera si collega in modo unico alla produzione di opere moltiplicate.

Il Gruppo T, infatti, espone insieme con Munari nel 1960 a Milano, presso lo showroom di Bruno Danese, presentando l'edizione in 10 copie numerate e firmate degli oggetti miriorama (Proverbio 2011).

Nel 2010, Alberto Alessi, fondatore dell'azienda di design italiana Alessi, ripropone il progetto di Danese presentando la riedizione dei multipli del Gruppo T in 100 copie ciascuno.

Cinquant'anni dopo, Alberto Alessi ha deciso di portare a compimento il progetto ideale di un'arte per tutti. Pur mantenendo tutte le sue caratteristiche di gradimento estetico, l'opera d'arte circola come oggetto di design tra gli oggetti di design (Anceschi 2010).

Fig. 21 Superficie magnetica, Davide Boriani, 1962



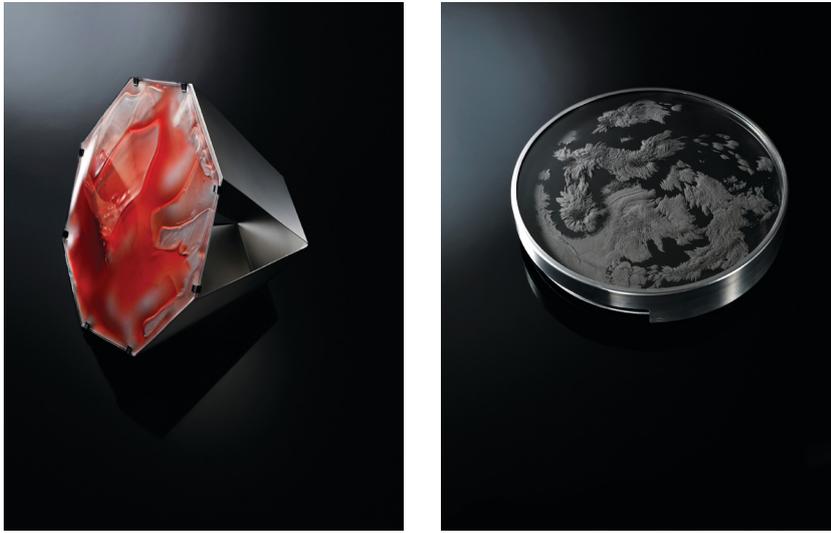


Fig. 22 Abstract video di Giovanni Anceschi, un multiplis in acciaio inossidabile e PVC di forma ottagonale contenente un liquido rosso. L'interazione si basa sulla rotazione manuale dell'opera che fa colare il liquido rosso, creando immagini continuamente diverse.

Fig. 23 Giradischi ottico-magnetico di Davide Boriani, un multiplis in alluminio, vetro e ferro in cui dei meccanismi ruotanti influenzano la superficie di polvere metallica.

Oggi la produzione dei multiplis del Gruppo T di Alessi avviene on demand e con l'applicazione di tecniche propriamente artigianali. Alcune delle componenti dei multiplis, in particolare, i meccanismi del Giradischi ottico-magnetico sono riprodotte con macchinari di manifattura digitalizzata. Finalmente le tecnologie di produzione sono pronte per l'attualizzazione di manifesti pionieristici del design.

Fig. 24 Rotoplastik di Gianni Colombo, un multiplo di legno in cui il gesto delle mani fa ruotare diversi profili sovrapposti di varia forma.



Fig. 25 Miramondo di Gabriele Devecchi, una serie di visori in ottone e alluminio che producono diversi effetti ottici.



Fig. 26 Sferisterio semidoppio di Grazia Varisco, un multiplo in resina e termoplastica in cui forme semisferiche si raddoppiano in una superficie speculare coperte da un reticolo quadro colorato.



## 2. I possibili livelli progettuali nell'open source

Nelle discussioni sull'open design si tende a farsi prendere dalla paura per tutto ciò che rischieremmo di perdere. Invece, bisognerebbe focalizzarsi sempre di più sulla potenzialità e libertà di azione che derivano proprio da questo: il cambiare punto di vista. (Romano 2014)

Il rilascio di un progetto di design, interattivo e non, in open source può essere analizzato alla luce di due punti di vista opposti: la pubblicazione in internet come volontaria perdita di controllo sia sull'artefatto di design e sul suo processo complesso di sviluppo tramite la condivisione dei file sorgente dell'artefatto; la pubblicazione in internet come perdita "controllata", ovvero il rilascio di un progetto tramite l'applicazione di buone pratiche a supporto della sua riproduzione da parte di altri. Affrontiamo in questo capitolo la descrizione e l'analisi delle varie pratiche connesse in maniera diretta o indiretta all'ambito della progettazione e dello sviluppo open source di artefatti fisici per definire i punti di cambiamento e di intervento dell'attività progettuale secondo il punto di vista che considera il rilascio di un progetto in open source come una "perdita controllata" piuttosto che come una perdita di controllo da parte dei designer..

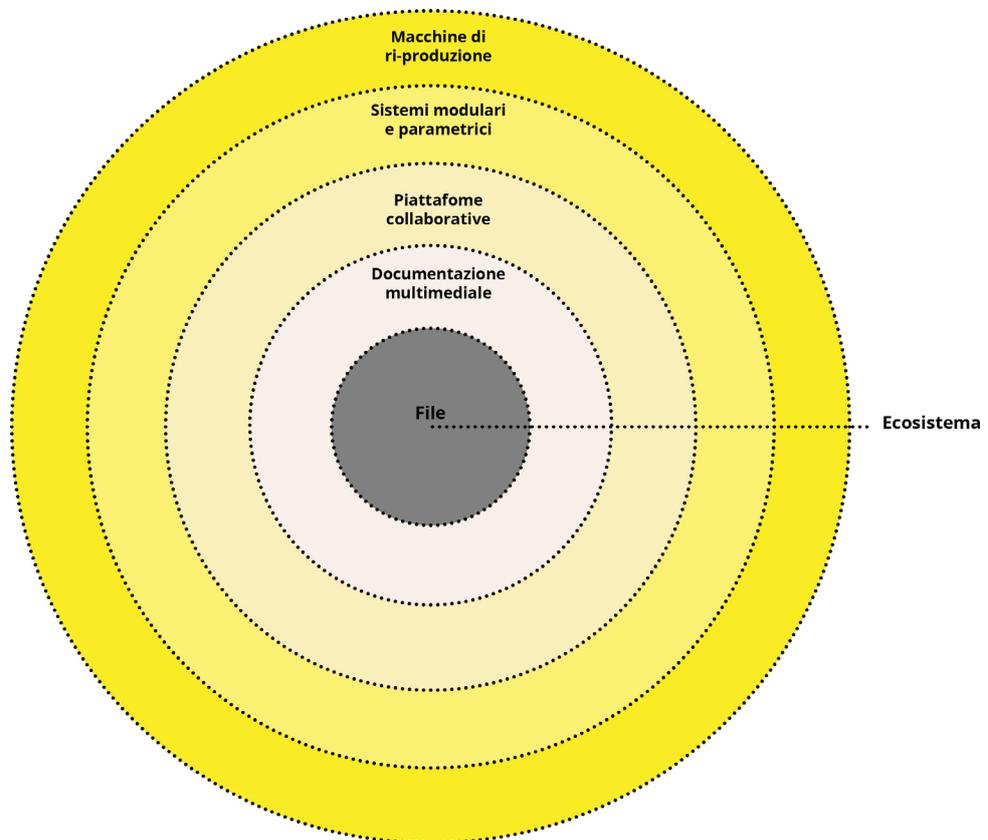
Analizzando le pratiche relative alla progettazione e sviluppo di Open Hardware, Open Design e le pratiche legate al fai da te elettronico è possibile individuare diversi modi di affrontare la progettazione di un artefatto interattivo che sarà rilasciato in open source, che enfatizzano in particolare, l'interazione tra l'autore, l'artefatto e la documentazione per riprodurlo e chi intende riprodurre l'artefatto.

## 2.1 Definizione dei livelli progettuali nel design di artefatti interattivi open source

L'interazione tra autore, artefatto, documentazione e “riproduttore” può avvenire su cinque livelli diversi:

- il livello dei file sorgente originali del progetto;
- il livello della documentazione multimediale;
- il livello dei sistemi e delle piattaforme di collaborazione;
- il livello dei sistemi modulari e parametrici;
- il livello delle interfacce, dei dispositivi e delle macchine di riproduzione.

Fig. 1 I livelli progettuali nel contesto open source



## 2.2 Il file sorgente originale come blueprint "aumentato"

In seguito all'analisi delle definizioni attuali di open source software, open source hardware e open design<sup>1</sup> è possibile considerare come livello "zero" dell'open source il rilascio di file originali sorgenti di un progetto interattivo, sia componenti software-hardware sia le parti, tramite un indirizzo web pubblico che ne permette lo scaricamento gratuito. Questo tipo di condivisione dei file è comunemente combinato alla pubblicazione di una documentazione testuale e/o multimediale (foto e video delle parti o del processo di assemblaggio) sui siti web personali dei realizzatori del progetto. Le tipologie di file sorgente originali sono di norma le seguenti (OSHWA 2015):

- file di disegni bidimensionali CAD (Computer Aided Design), ovvero i file che sono utilizzati per riprodurre le componenti e le parti fisiche di un oggetto o un dispositivo; un file di un disegno bidimensionale è per esempio il tracciato vettoriale che disegna la forma bidimensionale di una ghiera di un meccanismo all'interno di un dispositivo;

- file di disegni tridimensionali CAD, ovvero i file che sono utilizzati per riprodurre le componenti e le parti fisiche di un oggetto o dispositivo disegnati in tre dimensioni; un file di un disegno tridimensionale è realizzato con programmi di modellazione tridimensionale e riproduce digitalmente le caratteristiche fisiche di un oggetto;

- file di disegni bidimensionali CAD di circuiti stampati; questi sono accompagnati dagli schemi elettrici del circuito che sono la rappresentazione simbolica dei componenti elettronici e delle connessioni del circuito;

- i file di disegni bidimensionali che possono essere applicati sulle parti e sulle componenti hardware di un dispositivo come per esempio gli elementi grafici quali loghi, simboli;

- i file del software, ovvero il codice informatico che serve a programmare i comportamenti del dispositivo interattivo.

---

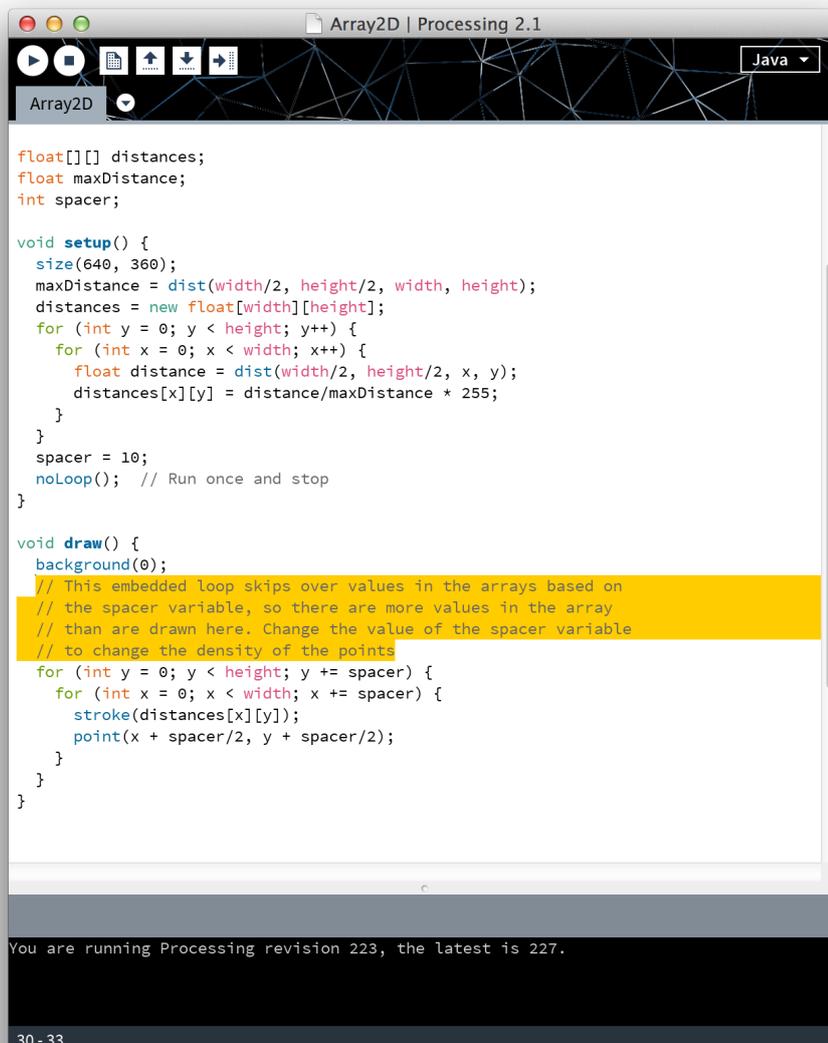
<sup>1</sup> Vedi Capitolo 1

I file sorgente originali sono rilasciati in formati adatti all'elaborazione tramite software di disegno bidimensionale, modellazione tridimensionale e di progettazione di circuiti, sia essi open source o proprietari. Nelle buone pratiche indicate da Open Hardware Association per la condivisione di open source hardware sono menzionati anche file in formati "ausiliari", come per esempio il formato PDF per i disegni bidimensionali e STL (*STereoLithography*) per quelli tridimensionali. Questi formati non sono nativi dei programmi CAD, ma possono essere esportati e aperti in altri programmi CAD della stessa tipologia di quella usata dai realizzatori del progetto. La scelta del formato del file è fondamentale quando si opera nella progettazione open source poiché dal formato dipende la possibilità che altre persone possano manipolare e modificare un progetto. Selezionare la tecnologia adottata al processo di realizzazione di un prodotto open source è un'attività importante, ma vedremo, in seguito, come gli aspetti tecnologici per se non siano gli elementi fondamentali nell'ambito dello sviluppo di un progetto di prodotto fisico open source.

Seppur nel contesto delle pratiche open source hardware, software e design, il rilascio dei file sorgente originali vada sempre di pari passo con lo sviluppo di una documentazione testuale e multimediale, la preparazione di un file sorgente originale è ricco di aspetti progettuali da tenere in conto quando l'obiettivo è supportare la riproducibilità e la riconfigurazione di un prodotto interattivo da parte di altri.

I file sorgente originali di un prodotto realizzati tramite programmi CAD (hardware, parti fisiche) sono l'esatta traduzione in codice informatico di un oggetto o una componente fisica del progetto (p.e. le espressioni matematiche che descrivono le curve di un disegno vettoriale) (Gershenfeld 2005). Il codice che crea la riproduzione digitale dell'oggetto contiene in sé la descrizione dell'oggetto e le informazioni per creare una copia digitale o reale tramite una macchina di produzione digitalizzata. Se da un lato la compatibilità dei formati e l'assenza di DRM (*Digital Rights Management*, le tecnologie per il controllo dell'uso di un file digitale) permettono a chiunque di scaricare e modificare i file sorgente di un progetto tramite un programma di progettazione CAD, dall'altro l'accessibilità dei programmi CAD stessi e la semplificazione

delle loro interfacce d'utilizzo aumentano la possibilità per molte persone, anche non esperte, di manipolare questa tipologia di file. Nel processo di sviluppo di software, hardware e design open source l'integrazione di informazioni contestuali all'interno di un file originale di un disegno bidimensionale o tridimensionale è un elemento chiave per rendere maggiormente accessibile e riproducibile un artefatto da parte di altri. Un esempio di una pratica diffusa è, per esempio, l'inserimento di commenti all'interno del codice informatico. Questa pratica è diventata un metodo e poi



```
float[][] distances;
float maxDistance;
int spacer;

void setup() {
  size(640, 360);
  maxDistance = dist(width/2, height/2, width, height);
  distances = new float[width][height];
  for (int y = 0; y < height; y++) {
    for (int x = 0; x < width; x++) {
      float distance = dist(width/2, height/2, x, y);
      distances[x][y] = distance/maxDistance * 255;
    }
  }
  spacer = 10;
  noLoop(); // Run once and stop
}

void draw() {
  background(0);
  // This embedded loop skips over values in the arrays based on
  // the spacer variable, so there are more values in the array
  // than are drawn here. Change the value of the spacer variable
  // to change the density of the points
  for (int y = 0; y < height; y += spacer) {
    for (int x = 0; x < width; x += spacer) {
      stroke(distances[x][y]);
      point(x + spacer/2, y + spacer/2);
    }
  }
}

You are running Processing revision 223, the latest is 227.

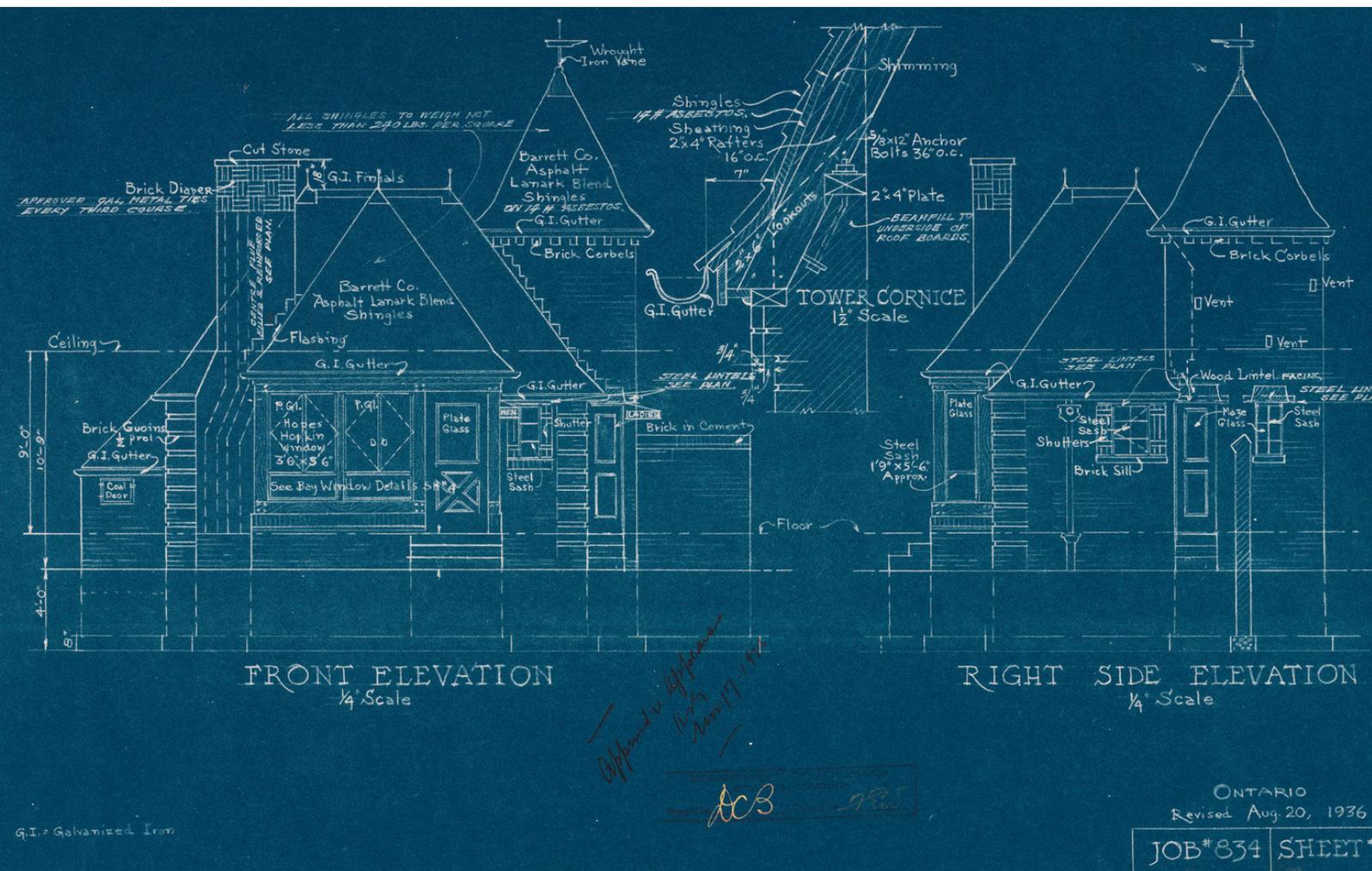
30 - 33
```

Fig. 2 Esempio di commenti in uno "sketch" di esempio (Array2D) dell'ambiente di programmazione Processing

uno vero e proprio standard nell'ambito dello sviluppo di applicazioni informatiche (Improving software 2011). Data la natura collaborativa dell'implementazione di applicazioni informatiche open source e la natura stessa del codice, che è condivisibile tramite lo scambio di semplici file di testo, "il commentare" esemplifica la volontà di rendere il codice più comprensibile sia per il programmatore stesso sia per chi si accingerà a riutilizzarlo o modificarlo (McConnell 2004).

Nello sviluppo di artefatti fisici interattivi e, in generale, degli artefatti fisici, le buone pratiche di rilascio di informazioni contestuali all'interno del file originale sono strettamente correlate alla pratica stessa della progettazione tecnica e alla storia dell'evoluzione del disegno tecnico. In design e architettura i disegni tecnici e/o piani portano il nome inglese di *blueprint*. I *blueprint* all'inizio del diciannovesimo secolo erano disegni tecnici realizzati su carta fotografica con sfondo blu che permetteva la riproduzione esatta dei disegni tecnici in più copie (Merriam Webster Encyclopaedia 2015). Con l'introduzione di programmi informatici CAD e l'arrivo delle tecnologie digitali di stampa, la tecnologia del *blueprint* è stata superata per dare spazio all'utilizzo di modelli tridimensionali e disegni vettoriali bidimensionali per la rappresentazione di un progetto di design. Oggi, infatti, il termine *blueprint* va a indicare in generale qualsiasi disegno tecnico dettagliato e, con l'arrivo delle tecnologie digitali, la sua funzione di intermediario tra designer e ingegnere (o costruttore) si modifica parallelamente all'evoluzione dei metodi di lavoro e delle tecniche di scambio di informazioni sul processo di realizzazione di un artefatto fisico.

Faccio riferimento ai *blueprint* del diciannovesimo secolo perché in quell'epoca essi rappresentavano, a mio parere, l'invenzione di una "tecnologia di riproduzione" di artefatti, prodotti, edifici, macchine piuttosto che una tecnologia di rappresentazione. I *blueprint* potevano essere riprodotti in molteplici e infinite copie per supportare un processo di produzione di un artefatto nel mondo fisico che a sua volta poteva essere prodotto molteplici volte. Come "tecnologia di riproduzione" permettevano la condivisione dei "codici sorgenti originali" di un progetto, le sue dimensioni, le caratteristiche e le forme, e delle informazioni contestuali a supporto del processo di ingegnerizzazione e creazione di un



prodotto e della sua realizzazione da parte di altri, sia essi membri stretti del gruppo di lavoro sia persone che collaboravano a distanza. In particolare, erano realizzati con la consapevolezza che qualcun altro potesse copiarli e usarli anche in assenza dell'autore sia l'architetto, il designer o l'ingegnere.

Oggi, uno degli elementi principali dell'Ecosistema open source è proprio l'utilizzo dei *blueprint* come "tecnologie di riproduzione", in particolare, in relazione alle emergenti tecnologie di produzione digitalizzata. Il file sorgente originale di un componente hardware o delle parti di un prodotto è allo stesso tempo la rappresentazione dell'artefatto fisico e l'esecutivo di produzione

Fig. 3 Blueprint, Canada, 1936 City of Toronto Archives

che può essere realizzato con una macchina a controllo numerico come per esempio una macchina a taglio laser, una fresatrice e una stampante tridimensionale. Inoltre, questo file sorgente può essere arricchito da un livello di informazioni contestuali che documentano, spiegano o mettono in evidenza aspetti del progetto così come facevano le annotazioni tecniche su un *blueprint*, ma in maniera più simile al commento integrato in un codice informatico.

Nel livello zero dell'open source il file originale funziona, quindi, come un *blueprint* aumentato, un artefatto con il quale le persone possono interagire al momento di riproduzione e riconfigurazione del prodotto interattivo. Fornisco di seguito alcuni esempi per spiegare questo concetto.

I disegni bidimensionali e i modelli tridimensionali realizzati tramite l'utilizzo di applicazioni informatiche per la grafica vettoriale e CAD sono rappresentazioni grafiche di prodotti fisici e per questo facilmente comprensibili da chiunque. Proprio per la natura visiva di questi file, l'integrazione di informazioni contestuali non si basa su una logica sequenziale, ovvero riga per riga come nel caso del codice informatico, ma sulla giustapposizione di livelli che visualizzano annotazioni testuali e elementi grafici, come etichette, simboli, marchi. Questa modalità di visualizzazione deriva dalle modalità di interazione e visualizzazione delle applicazioni informatiche stesse come Illustrator, Inkscape, AutoCAD e Eagle (per i circuiti stampati).

Nello sviluppo di componenti hardware destinate al mercato amatoriale (quello che è in molti casi sono anche componenti rilasciate in open source), la funzione di supporto alla comprensione dei disegni è delegata alle etichette testuali che servono a nominare i singoli connettori e componenti di un circuito stampato. Queste etichette, integrate nel file con un livello ausiliario, possono fornire informazioni riguardo alla quantità e alla tipologia di componenti (p.e. numero di connettori) e informazioni contestuali riguardo all'utilizzo da parte dell'utente finale (p.e. posizione corretta per la saldatura di un componente elettronico sul circuito).

Sparkfun è un'azienda statunitense che produce e distribuisce microcontrollori e circuiti stampati per il mercato dell'elettronica.

ca e della prototipazione hardware e software fai da te ([sparkfun.com](http://sparkfun.com)). La distribuzione avviene tramite il negozio on-line che è anche una piattaforma per la distribuzione di tutorial (guide passo-passo), articoli di supporto all'utilizzo dei prodotti. In un tutorial dedicato alla progettazione di una PCB (*Printed Circuit Board*) ottimizzata per la produzione (DFM *design for manufacture*), il fondatore e CEO di Sparkfun, Nathan Seidle, descrive una buona pratica per la progettazione orientata all'utente, ovvero quell'utilizzatore esperto che intende produrre o modificare l'hardware open source rilasciato dall'azienda.

Di sicuro ricorderai che cosa significa LED oggi, ma cosa succederà fra tre mesi quando controllerai di nuovo la cartella delle parti contenute nel tuo file? Io personalmente non inserisco nessuna indicazione sulle mie schede hardware. Quando monto una scheda devo sapere dove devo saldare le resistenze di 10k, i condensatori 0.1uF e i colori LED. Ma queste informazioni sono tutte parte della scheda di assemblaggio. Una volta che il dispositivo è costruito, non ho bisogno di sapere che uno specifico condensatore è di '22pF' [...]. Se devo controllare una scheda ho sempre a disposizione gli schemi e il layout. Dall'altro lato, sarebbe interessante poter sapere la funzione dei singoli connettori in modo che io non debba indovinare [...].<sup>2</sup>

Assemblare e programmare un circuito di un prototipo o di un dispositivo interattivo si basa di norma sull'accesso agli schemi elettrici e ai *datasheet* pubblicati dalle aziende produttrici dell'hardware, le quali rendono disponibile questa documentazione per far comprendere all'utente il funzionamento delle singole componenti e funzionalità del prodotto. Nonostante questa documentazione, l'integrazione di etichette e simboli grafici stampati direttamente sull'hardware, e quindi del livello contestuale all'interno del file originale, è una pratica essenziale per comprendere facilmente il funzionamento di un controllore o di un sensore:

Per chiarire l'importanza dell'uso delle etichette nel design di un circuito stampato, riporto l'esempio del tutorial di Siedle riguardo all'acceleratore a tre assi ADXL330 (Siedle 2014). Il sensore misura l'accelerazione su tre dimensioni, x, y e z, ed è utilizzato

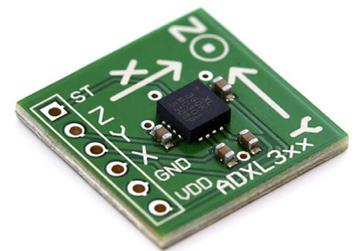


Fig 4 Accelerometro a tre assi di Sparkfun ADXL330

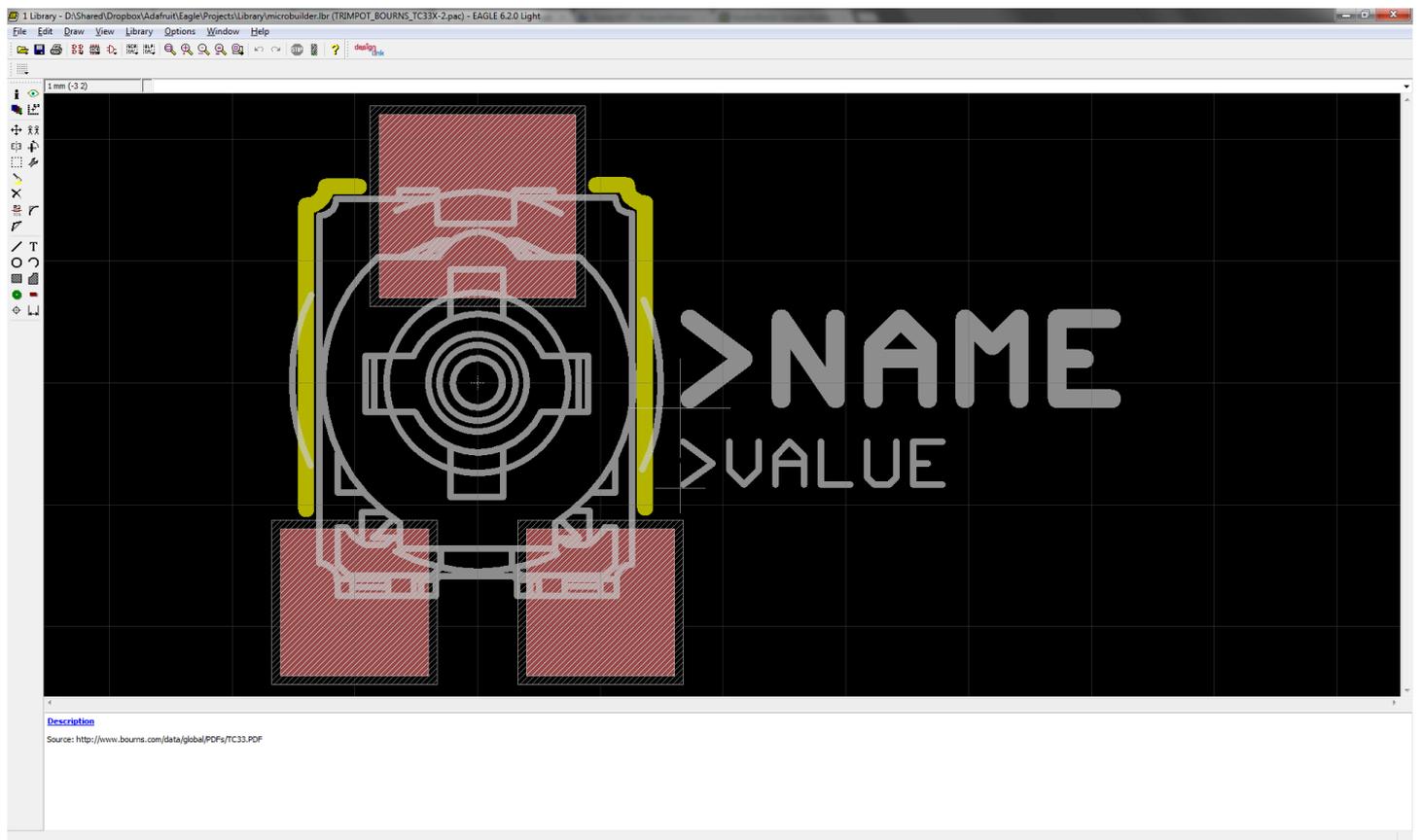
<sup>2</sup> Sparkfun 2014, mia traduzione.

per realizzare applicazioni per dispositivi mobili o giochi e creare controlli remoti che siano capaci di misurare la vibrazione, il movimento e l'accelerazione di gravità. Le etichette poste sul circuito e le frecce direzionali sono state applicate per poter permettere agli utilizzatori di riconoscere gli assi e, quindi, i valori connessi e misurati dal sensore.

Le buone pratiche di progettazione di hardware open source sono rintracciabili anche nei tutorial del Learning system di Adafruit ([learn.adafruit.com](http://learn.adafruit.com)), una delle aziende che, insieme a Sparkfun e Arduino, hanno contribuito allo sviluppo del settore dell'hardware open source producendo e distribuendo hardware e kit di elettronica fai da te open source.

Kevin Townsend (Townsend 2014) spiega il beneficio di disegnare un livello informativo aggiuntivo per descrivere una scheda

Fig. 5 Diagramma di un componente meccanico con documentazione delle aree destinate ai connettori e ad altri componenti elettronici.



hardware di grande complessità sottolineando come etichette e colori sono necessari per facilitare il riconoscimento dei connettori e la realizzazione di eventuali scatole e involucri della scheda.

Allo stesso modo dei file sorgente originali di un circuito stampato, i disegni bidimensionali di una parte o di un prodotto fisico sono rappresentazioni digitali, che dopo essere convertite in un linguaggio macchina tramite un software CAM (Computer Aided Manufacturing), possono essere realizzate con macchine a controllo numerico che tagliano e lavorano i materiali.

La conversione di un file di un disegno bidimensionale per la lavorazione tramite una macchina a taglio laser richiede, per esempio, la definizione di un primo livello informativo rappresentato dal colore delle linee vettoriali ([Vectorealism 2013](#)). I colori delle linee determinano che tipo di lavorazione la macchina deve svolgere, ovvero se bisogna ricavare un profilo tramite il taglio del materiale o se bisogna incidere il disegno sul materiale (incisione). Comunemente i programmi informatici che elaborano i file digitali per trasformarli in informazioni interpretabili da una macchina a taglio laser riescono a interpretare i colori delle linee così come definiti da chi ha realizzato il file e a comunicare alla macchina la presenza di diversi livelli associati alle diverse tecniche di lavorazione. Se da un lato, questo tipo di tecnologia potrà diventare obsoleta con l'avvento di nuovi tipi di software per la manifattura digitalizzata, nel contesto delle pratiche di progettazione di un artefatto open source è importante comprendere che il disegno esecutivo del prodotto, il suo file sorgente originale, è anche il suo *blueprint*, ovvero quel documento che fornisce informazioni a supporto della sua riproducibilità da parte di altri comunicando “in linguaggio macchina” i dettagli per la sua produzione.

I *blueprint* dei prodotti possono essere “aumentati” con un livello di informazioni contestuali quali le dimensioni, le unità di misura, le istruzioni di assemblaggio fino ad arrivare alle informazioni riguardo alle modalità con le quali interagire con esso.

Un esempio di come si integra il livello di base dei colori delle linee per distinguere le tecniche di lavorazione e il livello di informazioni contestuali è il file originale della Rolling Radio ([FabLab SUPSI 2013](#)), il kit di una radio FM fai da te composta da parti realizzate in compensato lavorato con una macchina a taglio

laser e con una stampante 3D. I file delle parti sono il *blueprint* del kit, con le linee vettoriali dei colori indicanti le diverse lavorazioni. A queste si aggiungono la legenda dei colori e le indicazioni sul tipo di materiale e sul suo spessore. Un altro progetto maggiormente esemplificativo in questo senso è rappresentato dalla collezione di *wearable* (accessori indossabili) open source del progetto Openwear ([openwear.org](http://openwear.org)). Openwear.cc è un sito internet e una comunità on-line di designer e piccoli produttori europei che condividono collezioni di moda pensate per la filiera corta e la manifattura digitale. Sviluppato nell'ambito del programma EDU-fashion ([edufashion.org](http://edufashion.org)), una piattaforma online costituitasi nel 2009 e finanziata dal programma Life Long Learning dell'Unione Europea, Openwear promuove la condivisione e la sostenibilità nella produzione di capi di moda nonché un modello di business in cui le collezioni open source sono protette dal marchio Openwear.

Introducendo i concetti legati all'open source design e nuovi modelli di business da questi generati, Openwear ha generato risorse per la formazione sui temi dell'open design e sui possibili nuovi modi di progettare capi moda secondo il modello del design open source. I risultati di un workshop pratico sviluppato sulla base del progetto Openwear ci permettono di ricavare un secondo esempio di *blueprint* aumentato: la collezione *wearable* del progetto Openwear. La collezione è stata sviluppata per insegnare i processi di manifattura digitalizzata al fine di realizzare accessori interattivi indossabili che potessero essere prodotti digitalmente e migliorati dalla comunità degli utenti. Lo sviluppo della collezione è affidata ai designer che hanno partecipato al progetto che hanno sviluppato dei file originali tenendo in considerazione la funzione formativa e la customizzazione da parte di utenti non esperti. I file originali della collezione includono una cintura, una pantofola, un guanto, un cappello, una borsa. I *wearable* sono stati progettati per permettere ai partecipanti del workshop di modificare sia le dimensioni sia le forme degli accessori e di incorporare componenti elettroniche per l'implementazione di comportamenti interattivi.

Il file originale della collezione è scaricabile Thingiverse.com, la piattaforma collaborativa di condivisione di file di disegni

tridimensionali e bidimensionali è preparato per essere lavorato da una macchina a taglio laser.

Il file è caratterizzato da un livello di informazioni contestuali quali il logo del progetto Operwear, l'indirizzo dal quale è possibile scaricare la collezione, le etichette con i nomi dei singoli *wearable*. Altri design open source del progetto Operwear sono esemplificativi rispetto all'uso del livello d'informazioni contestuale: il design open source "slipper", file originale per la creazione di pantofole ricavate da un unico pezzo di feltro, ha un'etichetta che indica l'area di customizzazione, ovvero la parte dove è possibile disegnare nuove decorazioni della tomaia.

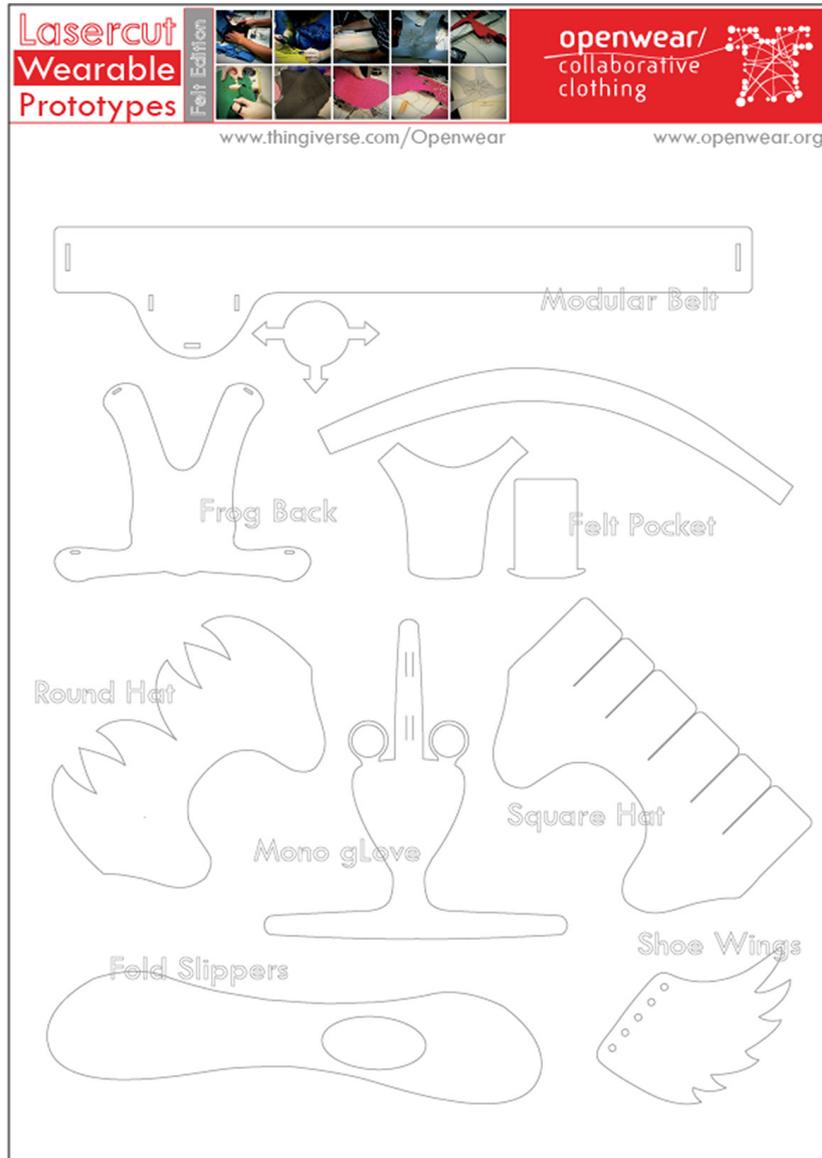
Il modello "slipper" è esemplificativo di come nell'open source design esistano dei limiti alla customizzazione che rappresenta uno degli aspetti chiave abilitati dalla possibilità di accedere ai file originali di design. Questi limiti riguardano la tipologia di prodotto che nel caso delle "slipper" è la calzatura. La calzatura prodotta in piccola serie tramite macchine a controllo numerico o a livello industriale deve essere progettata per rispondere alla necessità di calzata e di misura del piede. Con l'introduzione delle tecnologie di produzione digitalizzata e le tecnologie di *scanning* tridimensionale, si assiste alla diffusione di nuovi modelli di produzione e business nell'industria della calzatura che cercano di implementare in maniera parziale e totale il paradigma della customizzazione di massa. Il paradigma della customizzazione del prodotto calzatura si riferisce a tre diversi livelli (Duijlo, Boer 2007):

- la customizzazione dello stile, ovvero sulla base di forme predefinite e standard è possibile modificare elementi di stile della calzatura quali il colore, il materiale e gli accessori;
- la customizzazione della calzata standard (*best-matched fit*), ovvero la misurazione del piede della persona e la comparazione della forma con una libreria di forme esistenti;
- la calzata customizzata, ovvero la misurazione e l'analisi della camminata e la creazione di una forma ad hoc per la realizzazione di calzature adatte al piede di una singola persona.

Nel caso delle "slipper", la possibilità di azione da parte delle persone è limitata alla creazione di decorazioni sulla tomaia delle

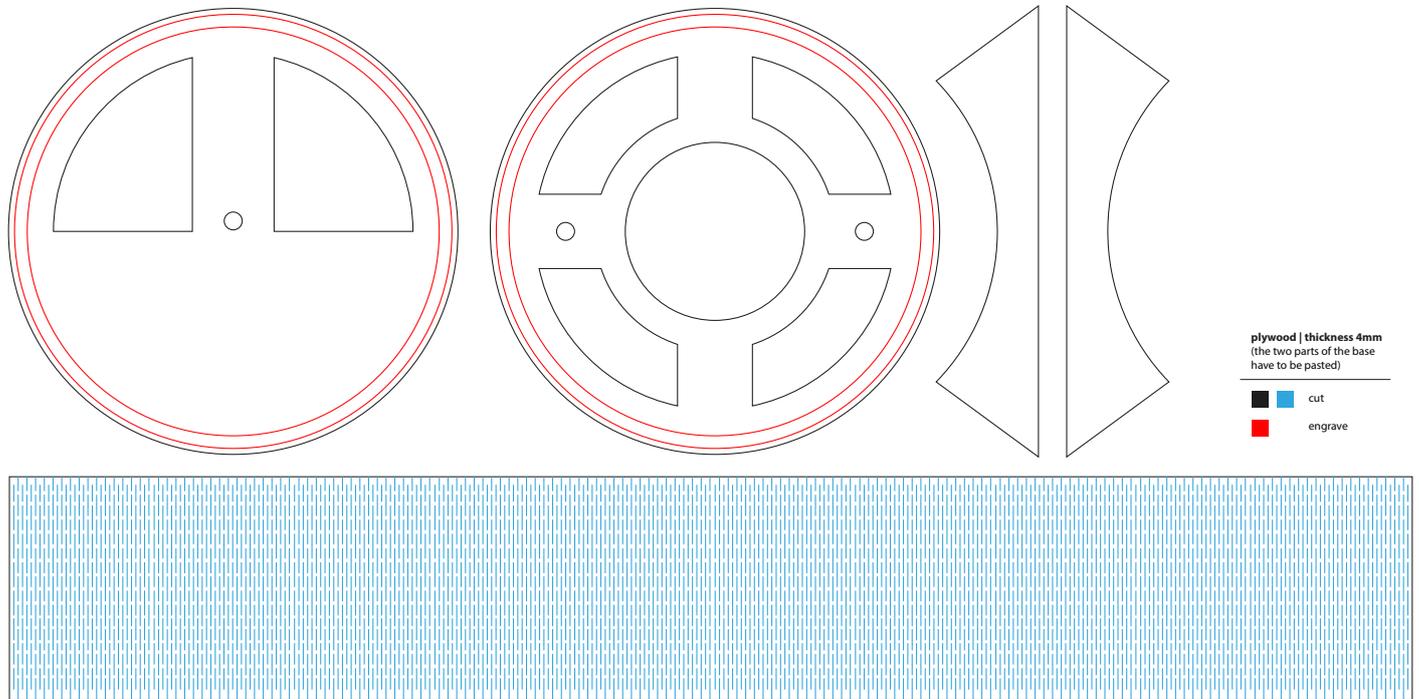
pantofoline (customizzazione dello stile) poiché è questa l'area di customizzazione definita al livello del file tramite un'etichetta testuale. Allo stesso tempo, è possibile modificare la forma della calzatura inserendo i valori delle dimensioni del piede tramite un programma di grafica vettoriale (calzata customizzata). Per questo tipo di modifica non è fornita nessuna informazione contestuale a supporto della modifica della misura della calzatura che

Fig. 6 Collezione di open source wearable del progetto Openwear.cc. Blueprint e file sorgente 2D degli accessori.



# Rolling Radio

by Giovanni Profeta



in qualche modo faciliti le persone non esperte nel processo di customizzazione fai da te del prodotto. Da qui si evince che l'intervento progettuale sul file originale risulta fondamentale in relazione alla tipologia di prodotto anche nel contesto del design open source dimostrando come l'idea di open source come perdita di controllo sul design finale sia un punto di vista debole se applicato in maniera generalista su qualsiasi tipo di progetto e prodotto.

Analizzando le pratiche di sviluppo e documentazione di file sorgente di un prodotto o un dispositivo open source come la Rolling Radio e la collezione Openwear si evince che un'area di intervento progettuale nel contesto open source è la creazione di *blueprint* "aumentati", ovvero artefatti digitali che fungono da rappresentazioni dettagliate di un design e da file originali che permettono la riproducibilità collaborativa e condivisa di un dispositivo. La progettazione di *blueprint* aumentati nel contesto delle pratiche open

Fig. 7 Rolling Radio. Blueprint e file sorgente 2D con le istruzioni sulle tecniche di lavorazione.

hardware e open design mette in gioco nuovi tipi di competenze: da un lato, le tecniche e le tecnologie restano le stesse (p.e. utilizzo di software CAD per il disegno tecnico e software per il disegno grafico), dall'altro il "linguaggio del blueprint" si allontana da quello degli standard tecnici (misure, proiezioni) per semplificarsi e integrare informazioni che possono descrivere non solo il dispositivo, ma il processo stesso di produzione e gli elementi di comunicazione visiva.

### 2.3 Le istruzioni e la documentazione a supporto della riproducibilità

Nel Capitolo 1 ho indicato come l'uso di internet abbia influenzato la nascita e la diffusione del movimento dell'elettronica fai da te e del conseguente open hardware e open design. In particolare, bisogna specificare che, se internet ha abilitato lo sviluppo di molte tecnologie dell'informazione e della comunicazione, il web nella sua versione 2.0 ha permesso di implementare e aggiornare i processi collaborativi che ora sono alla base delle pratiche adottate dal movimento open source.

Il web 2.0 è un paradigma la cui definizione si diffonde con la Web 2.0 Conference che fu istituita quando durante una sessione di brainstorming in una conferenza organizzata da O'Reilly e MediaLive International, Dale Dougherty evidenziò l'inizio di una nuova era del web in cui emergevano nuove e interessanti applicazioni in seguito al collasso dell'era del dot.com (O'Reilly 2005).

Il paradigma del Web 2.0 nasce da un'infrastruttura concettuale sulla quale ritengo si basi lo sviluppo dell'attuale Ecosistema open source. Questa infrastruttura concettuale riguarda principalmente le nuove funzionalità e le modalità di interazione con il web a seguito dell'introduzione di tecnologie e modelli di architetture dell'informazione da parte di servizi quali Yahoo, Google, Wikipedia, Delicious.

Queste funzionalità sono, ad esempio, la possibilità di aggregare i contenuti sulla base di parole chiave fornite dagli utenti (p.e. *folksonomy* vs categorie) o di generare dei risultati di ricerca sulla base dei link (p.e. Page Rank, l'algoritmo alla base del motore di ricerca di Google).

A partire da queste innovazioni, il web, secondo il nuovo paradigma 2.0, si trasforma in una piattaforma – *web as platform* – in cui l'interscambio di contenuti tra diversi servizi e la possibilità per gli utenti di creare e pubblicare contenuti sono gli elementi chiave per la definizione di nuovi modelli di business, da un lato, e nuovi tipi di interfacce e modalità di interazione, dall'altro.

Innanzitutto, i contributi degli utenti generano un effetto “network” (*network effect*) che permette di raggiungere più velo-

cemente il mercato e dominarlo. Il valore aggiunto della partecipazione degli utenti non si ottiene dal loro contributo volontario, ma dallo stabilire dei sistemi che aggregano i dati da essi forniti e che generano del valore derivato dell'uso quotidiano di un'applicazione (Brocklin 2006). Infine, analizzando il successo dei prodotti software open source, O'Reilly riconosce che questo successo che sembra derivato da una pratica di volontariato dipende, invece, da una motivazione egoistica dell'utente che crea valore collettivo come sottoprodotto del soddisfare le proprie esigenze, ovvero customizzare un software e adattarlo ai suoi bisogni. Parlando di Linux, Apache e Perl, O'Reilly dice:

Ognuno di questi progetti ha un cuore preciso, un meccanismo di estensione ben definito, e un approccio che permette ad ogni componente ben fatto di essere aggiunto da chiunque, incrementando quei livelli esterni di quello che Larry Wall, l'inventore di Perl, definisce "la cipolla". In altre parole, queste tecnologie dimostrano gli effetti network semplicemente attraverso il modo in cui sono state progettate (O'Reilly 2005).

Secondo O'Reilly il *network effect* alla base delle tecnologie basate sulla partecipazione e collaborazione degli utenti è qualcosa che è generato dal modo in cui queste tecnologie sono state progettate e per questo adotta come espressione per descrivere le logiche sottostanti queste tecnologie "l'architettura della partecipazione".

Utilizzo questo concetto come punto di partenza per spiegare le modalità sottostanti la condivisione di progetti open source tramite piattaforme collaborative i cui contenuti sono generati dagli utenti e per sottolineare come l'intervento progettuale sul piano dell'interazione tra i sistemi di condivisione di risorse per la riproduzione di un artefatto open source in internet e gli utenti determinino i processi e le modalità alla base della progettazione di un progetto open source.

Nella suddivisione analitica tra i diversi modi di affrontare il rilascio di un progetto open source, così come descritti in questo capitolo, il secondo livello delle pratiche open source è rappresentato dalla pubblicazione della documentazione e delle istruzioni per riprodurre un progetto tramite piattaforme on-line di nor-

ma supportate da comunità di persone interessate all'elettronica fai da te e, in generale, alla comunità del movimento maker (P2P Foundation 2014).

Le pratiche connesse all'elettronica fai da te abilitate dalle tecnologie web sono una delle infrastrutture chiave dell'“Ecosistema open source” e le piattaforme collaborative on-line per la condivisione di risorse in forma di tutorial sono i canali di accesso principali alle risorse necessarie per supportare la riproducibilità di un progetto open source.

Il tutorial on-line, ovvero le guide passo-passo alla creazione di un progetto, rappresenta, allo stesso modo dei *blueprint*, la tecnologia di riproduzione incorporata dalle piattaforme collaborative del fai da te: sviluppato come formato per l'apprendimento tramite i computer (*computer based learning*), un tutorial on-line si presenta sempre come la descrizione di un prodotto (p.e. un software) o la descrizione di un processo (p.e. come utilizzare una funzione del software) attraverso la suddivisione in sezioni tematiche fruibili in maniera lineare o ipertestuale (Torrey et al, 2007). Un tutorial online si basa sull'uso di diversi media:

- un video o un documento audio;
- un documento di testo;
- un'applicazione interattiva che presenta contenuti multimediali in maniera lineare o ipertestuale.

Nell'ambito della condivisione di risorse a supporto della prototipazione fai da te, il tutorial è riferito alle istruzioni passo-passo per la realizzazione di un progetto, ne documenta le fasi chiave attraverso l'uso combinato di video, audio, immagini e testo.

### 2.3.1 I tutorial generati dagli utenti

Analizzo di seguito tre tipologie di formati di tutorial come proposte dalle principali piattaforme on-line dedicate al fai da te elettronico e al mondo dei maker. Queste piattaforme sono state selezionate per: 1. l'elevato numero di utenti, 2. la tipologia di funzionalità, 3. le soluzioni di interfaccia.

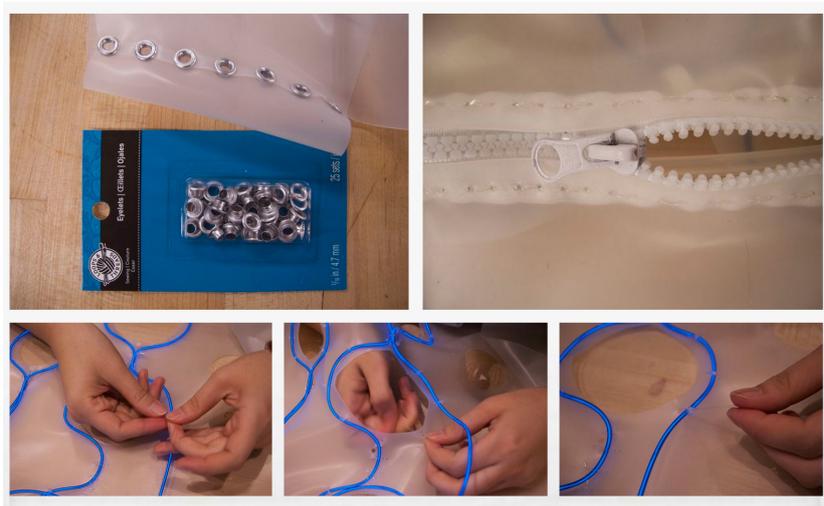
### A. Instructable

Instructable è una delle piattaforme di fai da te più popolari nel web dedicata a “persone appassionate che condividono cosa fanno e come lo fanno, che imparano dagli altri e collaborano con gli altri” (Instructable 2014). Offre un catalogo di circa 65.000 progetti organizzati in diverse categorie, dal cibo all’elettronica. Gli utenti di Instructable.com descrivono i loro progetti – definiti *instructable* - utilizzando testi, foto e video. Il sistema offre funzionalità per l’interazione e la collaborazione tra gli utenti che possono inserire commenti e discutere tramite i forum tematici. La creazione di un *instructable* si basa sulla descrizione passo-passo delle fasi necessarie per realizzare un progetto elettronico e interattivo<sup>3</sup>. Il sistema di Instructable propone una sezione di caricamento dei contenuti basata sull’inserimento di:

- immagini e fotografie;
- commenti testuali che descrivono ogni fase;
- annotazioni testuali (commenti ed etichette) sulle immagini.

Instructable è una piattaforma a supporto della comunità di persone che realizzano progetti in modo fai da te e i tutorial non sono sottoposti a una revisione e moderazione da parte della redazione centrale dei proprietari della piattaforma.

Fig. 8 Inflatable Vest.  
Esempio di instructable per realizzare un dispositivo indossabile



<sup>3</sup> La piattaforma organizza i tutorial in diverse categorie: technology, living, workshop, food, play, outside

## B. Thingiverse

Thingiverse ([thingiverse.com](http://thingiverse.com)) è la piattaforma di condivisione di file originali sorgente creata nel 2008 e connessa a MakerBot, un'azienda statunitense produttrice di stampanti tridimensionali fai da te ([makerbot.com](http://makerbot.com)). La piattaforma è uno degli elementi fondanti del modello di business dell'azienda basato sulla creazione di una comunità di utilizzatori delle stampanti Makerbot e di altre stampanti fai da te che condividono i progetti realizzati con stampanti tridimensionali open source e a basso costo (Greenberg 2012). In seguito al successo di Thingiverse per il numero di utenti e alla riconfigurazione aziendale di Makerbot industries, il 7 novembre 2012 è stata rilasciata una nuova versione della piattaforma con la riprogettazione dell'interfaccia utente e con l'integrazione di nuove funzionalità basate su un'architettura di tipo *social network* e sistemi di progettazione parametrica a supporto della customizzazione dei file tridimensionali condivisi dagli utenti.

I file accessibili e scaricabili da Thingiverse, rilasciati con licenze aperte, sono file di disegni tridimensionali e bidimensionali di oggetti, di parti fisiche di oggetti e schede hardware.

A differenza della piattaforma Instructable, i cui i contenuti condivisi sono le istruzioni per la creazione di progetti, la piattaforma Thingiverse propone un sistema in cui gli utenti possono:

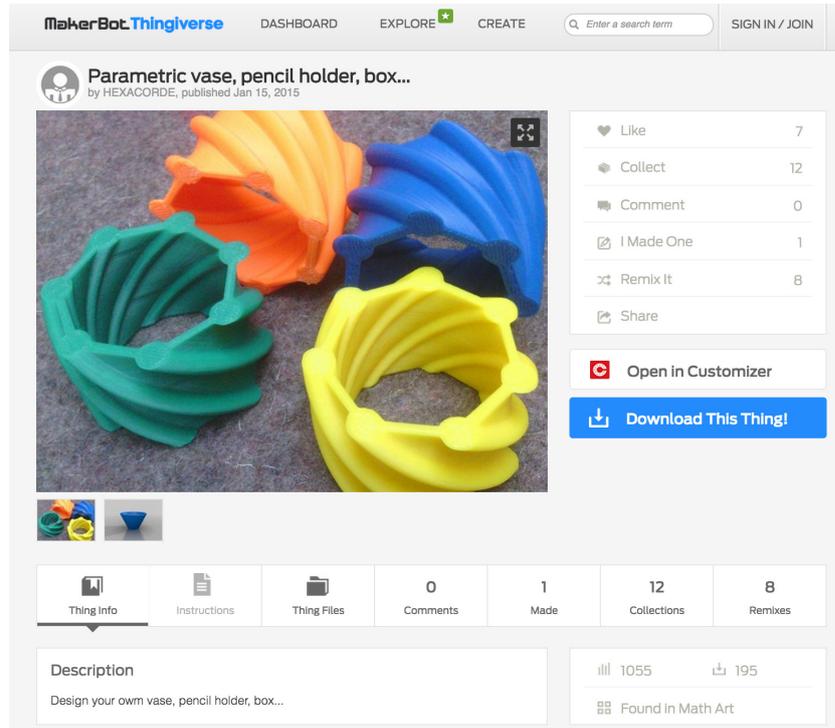
- inserire una descrizione generale del progetto;
- condividere i file sorgente originali delle singole parti che compongono un oggetto in diversi formati elettronici;
- fornire delle istruzioni tramite un campo di testo. Le istruzioni possono essere sia indicazioni tecniche sul processo di stampa dei file originali o istruzioni di assemblaggio.

A queste funzionalità si aggiunge una serie di funzionalità avanzate quali:

- gli utenti possono creare librerie di oggetti preferiti;
- condividere le pagine dei progetti sui propri social media;
- a partire dalla pagina di un progetto, pubblicare la documentazione e i file della versione di oggetto da loro riprodotta o modificata;

Come Instructable.com, anche Thingiverse è una piattaforma i cui contenuti sono generati in modo libero dagli utenti e non soggetti alla moderazione da parte di una redazione.

Fig. 9 Esempio di documentazione di un modello di un vaso parametrico pubblicato su Thingiverse.com



### 2.3.2 I tutorial redazionali

#### c. Make magazine

Make è la piattaforma on-line connessa al progetto della rivista Make fondata nel 2005 da Dale Dougherty, uno dei fondatori della casa editrice O'Reilly Media ([makezine.com](http://makezine.com)). Il progetto della rivista e la piattaforma on-line sono ispirati alle pubblicazioni dell'inizio del ventesimo secolo come Popular Mechanics ([popularmechanics.com](http://popularmechanics.com)) che proponevano articoli per supportare le persone nei loro hobby. Come spiega Dale Dougherty, queste pubblicazioni avevano l'obiettivo di aiutare le persone a trovare una comunità di riferimento con la quale confrontarsi e comunicare i propri interessi (Dougherty 2012).

I tutorial pubblicati da Make sono articoli redatti o revisio-

nati dalla redazione del progetto Make. Gli utenti possono contribuire inviando un proprio articolo pubblicando contenuti tramite un *wizard*, ovvero un tipo di interfaccia utente che presenta una serie di finestre di dialogo (*dialog box*) che guidano l'utente nello svolgimento di un compito attraverso una serie di fasi predefinite.

Il *wizard* di Make supporta l'utente nell'inserimento di:

- un titolo e una descrizione generica del progetto;
- la descrizione passo-passo del progetto tramite l'aggiunta di commenti a immagini e fotografie;
- la lista delle parti necessarie per la realizzazione del progetto, ovvero le componenti fisiche e elettroniche;
- la lista degli strumenti necessari per assemblare il progetto (p.e. cacciaviti, trapani)

L'interfaccia di tipo *wizard* presenta diversi campi d'inserimento di contenuto che permettono di creare i tutorial che sono strutturati sulla base dell'architettura dell'informazione e del layout della piattaforma.

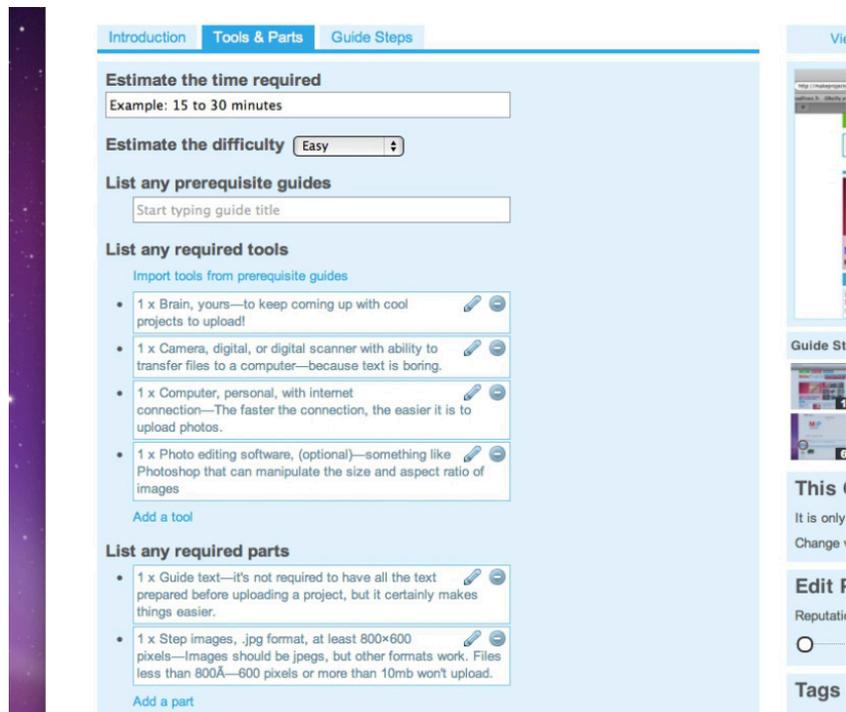


Fig. 10 Interfaccia wizard della piattaforma Make

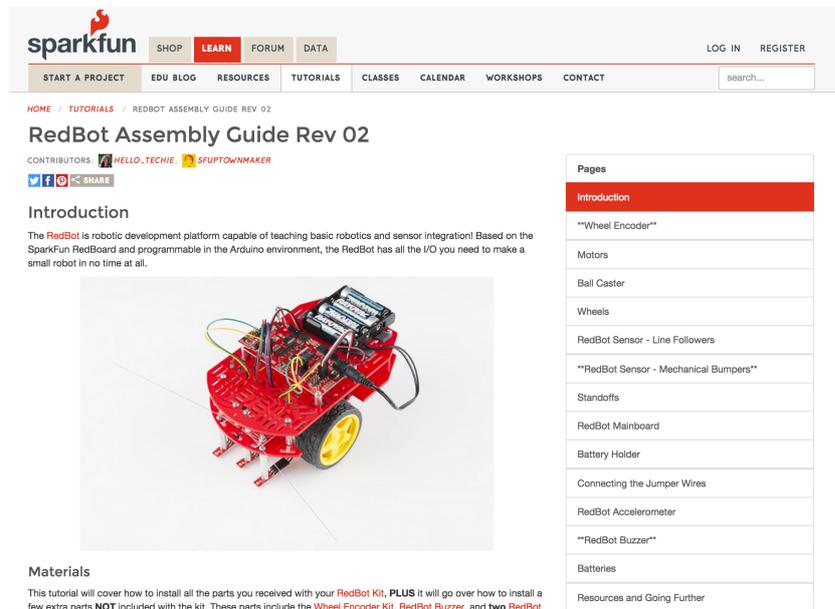
### 2.3.3 I tutorial integrati in “Learning system”

Due esempi di piattaforme per il supporto alla riproduzione di progetti open source sono le sezioni dei siti di due aziende di hardware e elettronica fai da te statunitensi: Sparkfun e Adafruit. I tutorial offerti nella sezione dedicata all'apprendimento on-line di Sparkfun e Adafruit non sono generati dagli utenti, ma sono curati da redazioni interne alle aziende. Si tratta di articoli in cui sono descritti gli elementi principali di un progetto organizzati in categorie piuttosto che in una sequenza di istruzioni passo-passo dettagliate che documentano le fasi di assemblaggio del progetto.

In entrambi i sistemi è possibile visualizzare i tutorial in maniera lineare (le sezioni del tutorial in una singola pagina) o in maniera sequenziale e ipertestuale (le sezioni suddivise in pagine diverse) in modo da supportare la fruizione del tutorial tramite dispositivi diversi, sia computer sia dispositivi mobili come i tablet e gli smartphone.

Su entrambe le piattaforme la lista delle parti e dei componenti necessari allo sviluppo del progetto è collegata al sistema di e-commerce delle aziende al fine di semplificare il ritrovamento e l'acquisto dei materiali necessari per realizzare il progetto fai da te. Questa funzionalità è stata integrata al fine di mettere in relazione

Fig. 11 Assemblaggio del RedBot Esempio di tutorial pubblicato sulla piattaforma dell'azienda Sparkfun.



The screenshot shows the Sparkfun website interface for the 'RedBot Assembly Guide Rev 02' tutorial. The page layout includes a top navigation bar with links for SHOP, LEARN, FORUM, and DATA. Below this is a secondary navigation bar with links for START A PROJECT, EDU BLOG, RESOURCES, TUTORIALS, CLASSES, CALENDAR, WORKSHOPS, and CONTACT. The main content area features a breadcrumb trail (HOME / TUTORIALS / REDBOT ASSEMBLY GUIDE REV 02) and the title 'RedBot Assembly Guide Rev 02'. It lists contributors (HELLO\_TECHIE and SFIPTOWNMAKER) and includes social media sharing icons. The 'Introduction' section describes the RedBot as a robotic development platform. A central image shows the assembled RedBot robot. Below the image is a 'Materials' section with a note that the tutorial covers installation of parts from the RedBot Kit, plus additional parts like the Wheel Encoder Kit, RedBot Buzzer, and two RedBot batteries. On the right side, a 'Pages' table of contents lists various sections of the tutorial, with 'Introduction' highlighted in red.

Pages
<b>Introduction</b>
**Wheel Encoder**
Motors
Ball Caster
Wheels
RedBot Sensor - Line Followers
**RedBot Sensor - Mechanical Bumpers**
Standoffs
RedBot Mainboard
Battery Holder
Connecting the Jumper Wires
RedBot Accelerometer
**RedBot Buzzer**
Batteries
Resources and Going Further

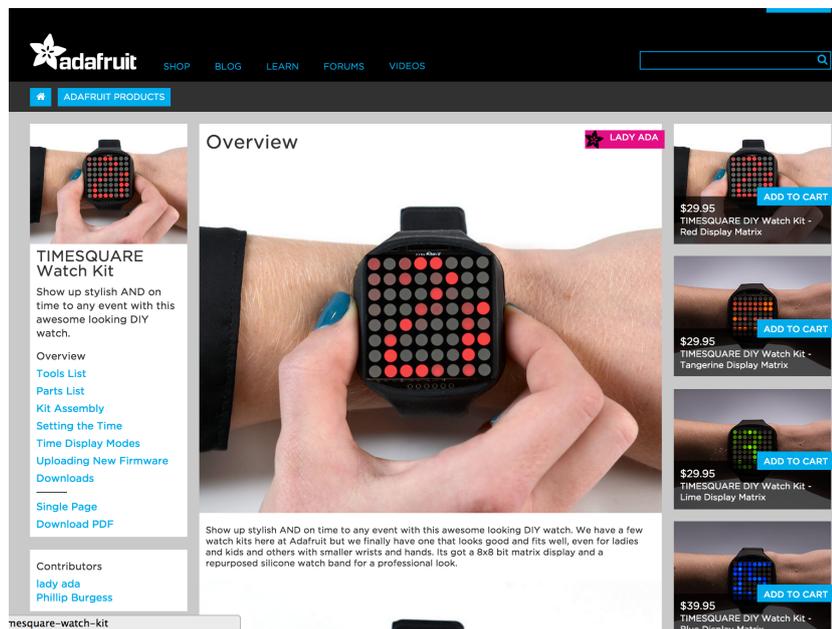


Fig. 12 Esempio di tutorial pubblicato sulla piattaforma dell'azienda Adafruit.

i contenuti gratuiti accessibili sulle due piattaforme con la vendita dei prodotti che rappresenta un modello di business emergente tra le aziende che rilasciano prodotti in open source.

Dall'analisi delle interfacce utente e delle modalità di interazione implementate in queste piattaforme web per la condivisione di risorse sulla realizzazione di progetti fai da te emerge una pluralità di pratiche relative al rilascio di progetti open source che, rispetto agli obiettivi di questo lavoro, mette in luce problematiche e opportunità per l'intervento dei designer.

Il primo aspetto problematico riguarda la necessità di combinare sistemi diversi per il rilascio della documentazione a supporto della riproducibilità di un dispositivo interattivo in cui i livelli progettuali sono molteplici (hardware, software, design del prodotto e interazione). Il "Learning system" dell'azienda Adafruit è la piattaforma che risolve questo tipo di problema poiché la tipologia di prodotti condivisi sono di elettronica, ovvero artefatti che per essere riprodotti richiedono un trasferimento di competenze di programmazione, elettronica e modellazione 3D per la realizzazione delle parti fisiche di un dispositivo. Allo stesso tempo, trattandosi di tutorial redazionali, non è possibile evincere il tipo di scelte sottostanti alla realizzazione dei tutorial né

il tipo di impatto che la struttura del tutorial ha sul processo di riproducibilità del dispositivo da parte degli utenti. Al contrario, le modalità di interazione e le funzionalità offerte da sistemi basati sul contributo degli utenti (p.e. thingiverse.com) permettono di generare tutorial collaborativi in cui le informazioni sulle varie versioni di un oggetto completano le informazioni mancanti della prima versione di tutorial.

Nello sviluppo di un progetto open source, la tensione tra la modalità redazionale e quella collaborativa è sempre costante: se, da un lato, le possibilità offerte dai modelli open source basati su licenze aperte e piattaforme di condivisione collaborative permettono di espandere la documentazione da parte di altri, migliorandola e completandola, dall'altro la riproducibilità di un progetto dipende dal corretto rilascio dei suoi file originali e della loro documentazione. La preparazione della documentazione rappresenta una fase chiave della progettazione nel contesto open source che deve essere affrontata dai designer con conoscenza e competenza delle diversificate pratiche relative alla condivisione di risorse nel contesto del fai da te e della cultura maker. Questa conoscenza deve adattarsi all'evoluzione delle pratiche poiché, secondo Shove (Shove et al 2012), le pratiche sono dinamiche ed evolvono costantemente attraverso i cambiamenti iterativi che i professionisti fanno. A dipendenza dall'evoluzione delle pratiche da parte di comunità larghe di utenti, il secondo aspetto di cui i designer devono sviluppare consapevolezza e competenza riguarda l'evoluzione dei paradigmi tecnologici e di interazione che sono generati e influenzano a loro volta le comunità di pratiche, come per esempio l'introduzione del paradigma del web 2.0 che ha portato allo sviluppo di piattaforme abilitanti gli utenti delle comunità del fai da te.

A partire dall'analisi delle piattaforme e della letteratura esistente, si evince che la progettazione di soluzioni a supporto della condivisione di tutorial sia ancora un terreno aperto e che richiede maggiori studi sulle comunità delle pratiche fai da te.

In uno studio sui metodi attraverso i quali la documentazione di un progetto è creata e utilizzata dagli utenti di Instructable, Tseng and Reisnik (Tseng, Reisnick 2014) suggeriscono diversi approcci per migliorare la struttura e il design delle piattaforme

on-line di documentazione di progetti fai da te che si basano su tecniche, per esempio, per integrare la progettazione con la documentazione. I designer sono costretti a ricreare il progetto in modo da documentare il processo con un enorme investimento di tempo. Gli strumenti che permettono di catturare il flusso di lavoro riducono le problematiche di creare una documentazione pubblicabile al termine della realizzazione del progetto. Altri approcci sono basati sull'introduzione di una documentazione basata sul processo:

molto spesso gli autori dei tutorial sviluppano la documentazione come la creazione di una ricetta, ovvero mostrando pochi fasi fondamentali per la realizzazione del progetto. Gli utilizzatori della piattaforma Instructable.com desiderano, invece, usare la piattaforma per trovare progetti da ricreare. In molti casi, necessitano di customizzare e personalizzare le fasi di sviluppo del progetto per integrarle in nuovi progetti. Spesso sostituiscono le parti, i materiali e gli strumenti e per questo motivo hanno interesse a sapere quali sono gli altri materiali e strumenti gli autori hanno scelto prima di utilizzare quelli documentati nel tutorial.

Introdurre nuovi strumenti per migliorare e “remixare” i contenuti e creare versioni in modo collaborativo sono pertanto l'attività chiave per facilitare il completamento della documentazione, ma soprattutto lo sviluppo della comunità.

Riguardo all'attività di creazione di tutorial più efficaci, uno studio evidenzia al contrario come l'accostamento e la “traduzione” della descrizione del progetto nel formato di un tutorial con quelle della creazione di ricette culinarie sia un approccio soddisfacente. Analizzando i tutorial di due progetti open source, il DIY Cellphone di David Mellis (Mellis 2014) e la lampada di cartone Fattelo (fattelo.com), e traducendoli nel formato della ricetta culinaria, gli autori della ricerca (Dalton et al 2014) giungono alle seguenti riflessioni: l'approccio minimalista del formato della ricetta culinaria non include la spiegazione delle tecniche di realizzazione di un dispositivo fai da te come per esempio saldare un circuito. I libri di cucina, invece, includono spesso sezioni in cui sono descritte le tecniche di base di ogni ricettario. Nelle pubblicazioni o sulle piattaforme riguardo al fai da te elettronico è,

quindi, estremamente utile includere la descrizione delle tecniche di base per manipolare strumenti e materiali. Al contrario, nei libri di ricette mancano sezioni come il *troubleshooting*<sup>4</sup> che sono fondamentali per supportare l'utente nella risoluzione di malfunzionamenti comuni del prodotto.

Fig. 13 DIY Cellphone di David Mellis

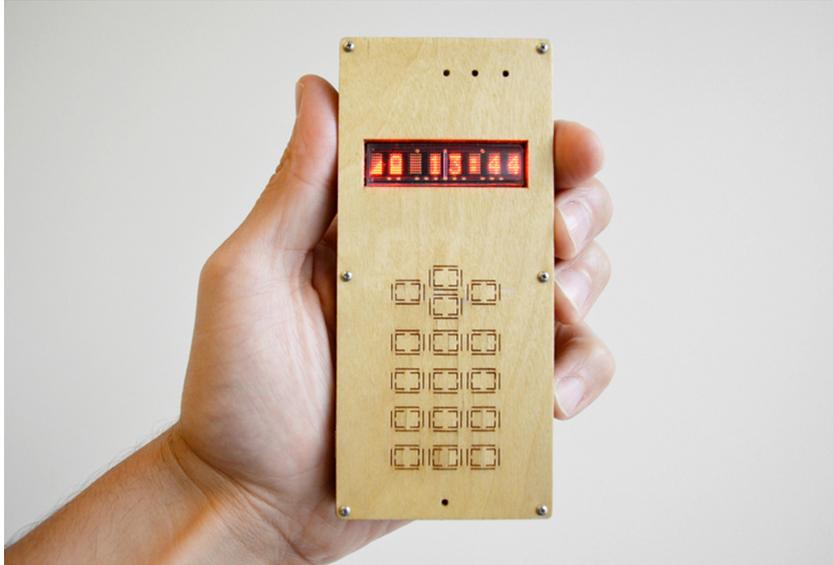


Fig. 14 Lampada Fattelo



---

4 Il troubleshooting corrisponde alle informazioni utili alla risoluzione di problemi comuni del funzionamento del prodotto

La condivisione di risorse per la realizzazione di artefatti fisici, interattivi e non interattivi, è argomento di studio in particolare nella comunità di ricerca connessa ai fablab, laboratori di fabbricazione digitale nati dal progetto del Centre for Bits and Atoms e ideato da Neil Gershefeld (Gershefeld, 2005). Rispetto alla comunità del fai da te elettronico, i fablab si presentano come una rete di laboratori connessi ad un network internazionale che condividono l'applicazione di pratiche condivise in spazi collaborativi dedicati alla digital fabrication (Fab Central 2012). Nuovi requisiti legati alla condivisione di risorse emergono in questo contesto perché l'obiettivo dei fablab è la condivisione di saperi riguardo alla possibilità di costruire "quasi ogni cosa". Se questo tipo di obiettivo è difficile da perseguire poiché i fablab si organizzano in base a modelli diversi, rimane aperta la necessità di creare metodi e tecnologie a supporto dell'aggregazione di risorse prodotte all'interno di questi spazi. I membri del network dei fablab utilizzano piattaforme come Flickr, Thingiverse, Instructables per la pubblicazione della documentazione dei progetti, ma allo stesso tempo sono consapevoli del problema che tali piattaforme non sono rilasciate come software open source e non assicurano un uso corretto delle licenze o dei termini riguardo alla proprietà intellettuale dei contenuti (Määttä, Troxler 2011). In particolare, un ulteriore problema relativo alla condivisione di risorse all'interno del network dei fablab deriva dalla propensione dei diversi laboratorio di creare dei propri strumenti di pubblicazione online piuttosto che utilizzare una piattaforma centralizzata. Ogni laboratorio adotta le proprie buone pratiche e i propri strumenti tecnologici per gestire questo tipo di risorse (Määttä 2011). Per rimediare a questo tipo di problema, nel 2011 è stato proposto un linguaggio di descrizione dei progetti realizzati presso i fablab definito FabML. Sviluppato in una serie di workshop sulla condivisione della documentazione di risorse utili ai fablab a partire dal 2010 il FabML si propone come una struttura XML adattata ai bisogni dei fablab che include vocabolari controllati per i materiali e le macchine utilizzati per completare un progetto con l'obiettivo di supportare funzionalità di ricerca su scala globale e formalizzare tutti gli aspetti legati alla manifattura digitalizzata man mano che si manifestano.

Le questioni riguardanti la documentazione di un progetto nell'ambito dei fablab risulta essere un punto di riferimento per comprendere che tipo di fattori devono essere considerati e integrati nella progettazione di un artefatto interattivo open source. Nella comunità di sviluppo di hardware open source si affida invece alle buone pratiche documentate dall'Open Hardware Association, l'organizzazione non-profit statunitense che ha l'obiettivo di supportare la comunità dell'open hardware nel mondo ([oshwa.org](http://oshwa.org)).

Fig. 15 FabLab SUPSI  
Lugano



#### 2.3.4 La documentazione dell'interazione

Le buone pratiche riguardo alla documentazione di progetti hardware open source sono utili per comprendere come un processo di progettazione in ambito open source debba includere lo sviluppo della documentazione che sia adatta al tipo di progetto o alla comunità di pratiche al quale il progetto si indirizza. Le opportunità di valorizzare le attività di documentazione riguardano principalmente la definizione di nuove modalità a supporto della comprensione dei processi di design e della pratica del design, in particolare, del design dell'interazione. Secondo Schön (Schön 1983), i processi di design possono essere considerati come una

conversazione riflessiva con i materiali, in cui il designer lavora con media e materiali diversi e sperimenta con vari aspetti del design. Attraverso un dialogo costante tra il designer e i materiali, questi apprende i problemi inaspettati e potenziali, in termini di un sistema di implicazioni tra attività successive. La descrizione di un progetto attraverso la realizzazione di un tutorial è un'attività che permette un tipo di conversazione riflessiva che influenza la pratica di progettazione favorendo la messa in luce di possibili problemi e opportunità. La creazione di un tutorial o la documentazione di un progetto può essere associata, infatti, alla creazione di un *Design Rationale*, un termine usato in molte aree del design e dell'ingegneria per indicare la documentazione delle risorse e delle decisioni che portano alla definizione di un prodotto e che migliora la qualità del prodotto e del processo di progettazione.

Secondo Lee (Lee 1997), il *Design Rationale* è uno strumento importante perché non solo include le decisioni sottostanti il design di un prodotto, ma anche la spiegazione, i compromessi e le argomentazioni che hanno portato ad una specifica decisione.

La creazione di un *Design Rationale* dettagliato così come la realizzazione a posteriori di un tutorial a supporto della creazione di un dispositivo interattivo condiviso su una piattaforma web è un'attività che richiede un enorme investimento di tempo. Come suggerito da Tseng e Reisink, questo tipo di problema può essere superato tramite la creazione di strumenti che supportano l'integrazione delle attività di documentazione durante le attività di progettazione. Questi strumenti dovrebbero poter includere potenzialmente anche i punti di vista delle persone che partecipano alla creazione di un progetto. Schoffeleene e Huybrechts (Schoffeleene, Huybrechts 2013) dimostrano come la documentazione tramite l'integrazione del "punto di vista" di coloro che partecipano allo sviluppo di un progetto collaborativo possa favorire le dinamiche di sviluppo di un progetto da parte di altre persone (*end user development*). Analizzando progetti di sviluppo collaborativo di artefatti che integrano software e hardware, i due ricercatori hanno sviluppato un metodo basato sull'utilizzo dell'approccio della *thick documentation* che si basa sull'utilizzo di uno strumento di mappatura che offre una vista dinamica della documentazione e delle prospettive soggettive di chi collabora al progetto e le informa-

zioni sul livello di *generativity* di un progetto. Il termine *generativity* (generatività) serve a descrivere progetti in cui i partecipanti continuano a partecipare al suo sviluppo senza il supporto e lo stimolo da parte dei realizzatori. Il termine *generativity* sostituisce l'uso dell'espressione *end-user development* poiché è riferito alla capacità che un progetto ha di generare la partecipazione da parte di chiunque, siano essi designer, sviluppatori, maker o utilizzatori finali di un prodotto. La "generatività" è una delle caratteristiche che un progetto deve avere per essere considerato riproducibile nel contesto delle pratiche open source ed è un aspetto che può essere progettato attraverso l'integrazione di pratiche di documentazione durante il processo di progettazione e l'inclusione di livelli di documentazione profondi (p.e. il punto di vista dei realizzatori di un progetto) che permettono di stimolare la riproducibilità di un artefatto interattivo da parte di altre persone.

La definizione dei requisiti per creare piattaforme efficaci a supporto della condivisione di risorse così come la creazione di standard per la descrizione di progetti di artefatti realizzati presso laboratori di produzione collaborativa e digitale resta un tema aperto all'interno della comunità dell'open source hardware e design. Allo stesso modo anche l'intersezione tra le pratiche proprie dell'interaction design e le pratiche open source restano ancora da investigare. La materia dell'interaction design consiste in materiali intangibili quali per esempio, la comunicazione, i feedback e i comportamenti di un artefatto interattivo che integra tecnologia hardware e software in relazione alle persone che lo utilizzano. Rendere accessibili un dispositivo interattivo in open source al fine di permettere la riproduzione di questo da parte di altri vuol dire sviluppare una documentazione che comunichi questa materia intangibile. L'attività di documentazione di un progetto è in sé l'aspetto chiave del lavoro dell'interaction designer il quale definisce la qualità dell'esperienza tra gli utenti e i dispositivi tecnologici (Chiappini, Rigamonti 2010), adottando metodi e tecniche quali lo *sketching*, il *paper prototyping* o lo sviluppo di programmi informatici e hardware che mostrano i comportamenti di un artefatto fisico in forma di prototipi funzionanti (Lindell 2012). Inoltre, gli interaction designer sono anche i professionisti

che lavorano in gruppi multidisciplinari per lo sviluppo di prodotti tecnologici e che contribuiscono al loro sviluppo attraverso l'applicazione di competenze in *design thinking*, *design literacy* (Buxton 2013).

La qualità della documentazione nello sviluppo di un progetto di dispositivo interattivo in cui sono coinvolte molte persone con diversa formazione ed esperienza, è fondamentale per catturare le fasi della progettazione, i compromessi fatti e lo scambio di informazioni tra tutti i membri del gruppo di lavoro. Questo tipo di attività è strettamente connessa al lavoro dell'interaction designer e i suoi metodi e tecniche sono da tempo codificate.

In *Communicating Design* (Brown 2011), Dan Brown fornisce una descrizione dettagliata delle tipologie di documenti (*design deliverable*) che sono realizzati durante lo sviluppo di progetto di applicazione interattiva digitale e spiegano come essi siano degli strumenti per mantenere una visione consistente del progetto da parte di tutti i membri del gruppo di lavoro (*consistency of vision*), comunicare le responsabilità ovvero le decisioni prese e l'impatto di tali decisioni sul progetto (*accountability*), infine tracciare l'evoluzione del progetto comunicando i punti di decisione (*traceability*). Brown elenca dieci diverse tipologie di documenti di cui quattro sono quelle legate al design:

- la mappa del sito (*site map*), ovvero i diagrammi che documentano tutte le informazioni contenute in un'applicazione, come è costruito e come sarà navigato dagli utenti in un'unica vista;

- i *wireframe*, ovvero le illustrazioni semplici della struttura di ogni schermata di un'applicazione web il cui scopo è comunicare l'idea iniziale di design, la struttura piuttosto che il layout grafico;

- i diagrammi di flusso (*flowchart*) documentano e visualizzano come un compito o una serie di compiti sono svolti dall'utente. Questi mostrano come le informazioni sono viste dagli utenti finali;

- le schermate grafiche (*design screen*) mostrano come l'applicazione apparirà agli utenti finali e veicolano gli aspetti legati al brand e all'obiettivo di comunicazione del progetto.

In un progetto di sviluppo di un'applicazione web le tipologie di *deliverable* hanno obiettivi diversi, ma sono spesso connessi tra di loro al fine di rappresentare e comunicare i molteplici

aspetti del progetto interattivo sia all'interno del gruppo di lavoro sia verso chi commissiona il progetto. In particolare, la visualizzazione degli aspetti legati ai feedback e all'interazione con l'utente (la materia intangibile dell'interaction design) sono delegati ai diagrammi *wireframe* e ai diagrammi di flusso che combinati insieme documentano i comportamenti di un sistema integrando la rappresentazione del sistema di retroazione.

*Wireframe* e diagrammi di flusso sono utilizzati insieme soprattutto per la progettazione di sistemi le cui interfacce sono di tipo GUI (*Graphical User Interface*). In interaction design, le GUI sono le interfacce basate su icone grafiche e indicatori visivi (p.e. le icone delle cartelle) e utilizzate per supportare principalmente l'interazione con artefatti digitali accessibili tramite schermi bidimensionali come siti web, applicazioni mobile e software, interazione che di norma avviene tramite dispositivi di input quali la tastiera e il mouse. La denominazione di GUI è legata allo sviluppo di interfacce per i computer quando la Xerox Palo Alto Research Unit introdusse il sistema grafico per sostituire l'interfacciamento tramite richieste eseguite con linee di comando testuali. Nel contesto della progettazione di artefatti interattivi, le tipologie di interfaccia si sono evolute e diversificate includendo interfacce di tipo TUI (*Tangible user interface*) e PUI (*Perceptual User Interface*). Le interfacce TUI emergono al termine degli anni novanta dalle ricerche del Tangible Media Group guidato da Hiroshi Ishii e sviluppate a partire dall'idea di ridurre la distanza tra il mondo dell'interazione con dispositivi tecnologici, basata su rappresentazioni digitali sugli schermi, con quello dell'interazione quotidiana con l'ambiente in cui viviamo. Da questo assunto, Ishii definisce le TUI come le interfacce che permettono di supportare l'interazione dando una forma fisica all'informazione digitale. Esse funzionano da rappresentazione di tale informazione anche allo stesso tempo da sistema di controllo e manipolazione delle stesse. Si tratta di oggetti fisici aumentati che offrono all'utente un doppio feedback, quello fisico ovvero un feedback passivo di tipo aptico che informa l'utente che la manipolazione dell'informazione sta avendo luogo o è completata, e digitale, ovvero output visivi o auditivi che informano l'utente dell'avvenuta interpretazione computazionale della loro azione (Ullmer, Ishii 2000).

Un progetto open source per l'ambito della progettazione delle TUI è Reactivision ([reactivision.sourceforge.net](http://reactivision.sourceforge.net)) che, sviluppato come progetto di ricerca su sistemi basati su interfacce tangibili, ha reso accessibile le informazioni a supporto dell'implementazione di dispositivi basati su interfacciamento tramite oggetti fisici: si tratta di un *computer vision framework* per diversi sistemi operativi che permette il tracciamento di *fiducial marker*, ovvero simboli grafici che possono essere stampati su carta e agganciati a oggetti fisici per associare a tali oggetti un codice identificativo riconoscibile dal software. Sulla base di questo *framework* è stato per esempio realizzato il Reactable ([reactable.com](http://reactable.com)), un tavolo interattivo per fare musica in maniera collaborativa tramite la manipolazione di moduli che, abbinati ad un'informazione digitale, generano suoni che a loro volta interagiscono con altri moduli e altri suoni.



Fig. 16 Reactable, tavolo interattivo basato sulla tecnologia open source Reactivision

Le PUI (*Perceptual Use Interface*) fanno riferimento alle interfacce in cui tra utente e dispositivo non c'è nessuna interazione basata sul tocco, per esempio, le interfacce basate sui comandi vocali e i movimenti del corpo. Esse sono interattive, multimodali e progettate per supportare un paradigma di interazione naturale uomo-uomo in cui le persone interagiscono con la tecnologia in

modo simile al come fanno tra di loro nel mondo fisico (Turk 2000). La natura percettiva di queste interfacce è di tipo bidirezionale, ovvero utilizzano allo stesso tempo le abilità percettive delle macchine rispetto all'ambiente esterno e le abilità delle persone di comunicare. La riflessione su questo tipo di interfacce nasce con l'inizio della sperimentazione nell'uso di dispositivi quali gli *head mounted display* e delle interfacce basate sul controllo vocale.

Mentre le tecniche di documentazione di sistemi interattivi basati su GUI sono condivise dalla comunità dei professionisti, nell'ambito della progettazione di TUI esistono pochi sistemi codificati per il rilascio delle specifiche di design e per le PUI praticamente nessuno. Per le TUI, un punto di riferimento è rappresentato dalla codifica delle modalità di interazione tramite interfacce basate sul tocco, in particolare, le modalità di interazione con applicazioni per dispositivi mobili. Con l'avvento delle tecnologie *touch* si assiste alla definizione di linguaggi di rappresentazione della cosiddetta interazione di tipo gestuale e con la diffusione dei sistemi operativi iOS per *smartphone* dell'azienda Apple e lo sviluppo dell'alternativa open source Android inizia la definizione di librerie di simboli grafici per rappresentare le modalità d'interazione. La documentazione di progetti basati su interfacce TUI si basa quindi sull'integrazione di simboli grafici che descrivono i gesti che richiamano un certo tipo di informazione o azione. Descrivo di seguito i gesti di base secondo le linee guida di Apple.

Al fine di supportare il processo di riproducibilità di un dispositivo open source caratterizzato da TUI o PUI, sarà necessario sviluppare dei diagrammi e delle illustrazioni che si basano su librerie di icone standard oppure elementi grafici originali che spieghino la tipologia di input e di output del dispositivo. La documentazione di un'interazione basata su interfacce di tipo TUI e PUI è di norma supportata dall'uso di documenti video, animazioni interattive e software di simulazione, sia web sia desktop. Un esempio di documentazione delle modalità di interazione è il sistema dei tutorial on-line della piattaforma Tinkerkit, kit hardware compatibile con Arduino basato sulla semplificazione dell'interfaccia della scheda hardware tramite l'uso di connettori e moduli per gestire la programmazione di sensori e attuatori ([tinkerkit.com](http://tinkerkit.com)).

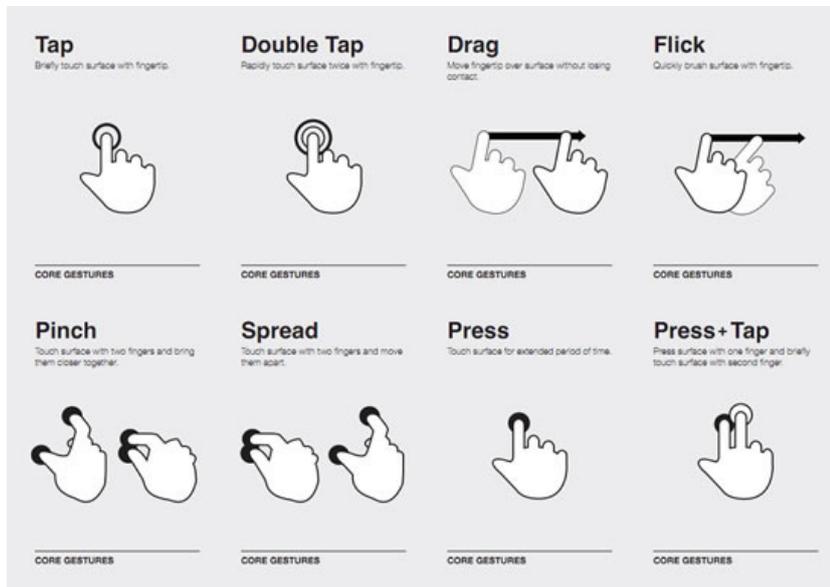


Fig. 17 Icone che descrivono i gesti di base per l'interazione con interfacce touch secondo le linee guida Apple: il tocco singolo (tap) per premere o selezionare un controllo o un oggetto; il trascinamento (drag) per effettuare lo scrolling da un lato all'altro o trascinare oggetti sullo schermo; il tocco veloce (flick) per scorrere rapidamente lo schermo; lo scorrimento, con un dito, per tornare alla schermata precedente, per rilevare una schermata nascosta o per attivare il bottone di cancellamento nella vista a tabelle; il tocco doppio (double tap) per effettuare l'ingrandimento o centrare un'immagine; la presa (pinch) verso l'esterno per ingrandire la vista, verso l'interno per rimpicciolirla; il tocco lungo (touch e hold), per ingrandire la vista dei testi al fine di controllare il puntatore; l'oscillazione del dispositivo (shake) per annullare l'operazione o compierla di nuovo.

Il tutorial è stato realizzato per spiegare le basi della programmazione hardware digitale tramite il kit basato su Arduino, ovvero come si programma il comportamento di un sensore e un attuatore.

Per spiegare, per esempio, la correlazione tra l'interazione con il potenziometro lineare e la regolazione del livello di luminosità di un LED, il tutorial offre all'utente la possibilità di interagire con un'icona interattiva del componente "potenziometro lineare" e visualizzare in tempo reale la variazione di luminosità del componente "LED" tramite la variazione dei valori massimi e minimi misurati dal componente "Potenziometro".

Questo tipo di soluzione di documentazione basata su un'animazione interattiva mostra gli aspetti problematici che riguardano la documentazione o rappresentazione di un sistema interattivo dotato di interfaccia fisica, ovvero la necessità di visualizzare animazioni complesse per simulare meccanismi di interazione e retroazione. Per questo motivo, l'integrazione di annotazioni interattive rappresenta una strategia per il miglioramento dell'efficacia dei sistemi di condivisione di tutorial, soprattutto in relazione all'aspetto dell'interazione, ovvero la documentazione di come avviene la relazione tra la persona e i comportamenti del sistema.

Le animazioni interattive rese possibili dall'evoluzione delle tecnologie dei linguaggi di programmazione web permettono di realizzare degli artefatti interattivi che integrano allo stesso tempo i file sorgente di un artefatto interattivo e la documentazione delle sue modalità di interazione. Forniamo di seguito alcuni esempi tratti da alcuni casi studio.

OpenProcessing ([openprocessing.org](http://openprocessing.org)) è una piattaforma di condivisione di progetti, da animazioni e grafica generativa a sistemi di simulazione delle forze fisiche, sviluppati tramite l'ambiente di programmazione Processing. L'implementazione di tecnologie web come Javascript, HTML5, permettono agli utenti di OpenProcessing di accedere ad un file sorgente delle applicazioni rilasciate con la licenza CC (Attribuzione-condividi allo stesso modo 3.0 -CC BY-SA 3.0 US) and GNU GPL e interagire con esse direttamente dal browser. In OpenProcessing si annulla la separazione tra il progetto (il codice) e la sua documentazione intesa come risorsa multimediale che supporta la comprensione delle funzionalità del progetto.

Sottolineo come questo tipo di soluzione è resa possibile solo dalla natura digitale degli artefatti pubblicati in OpenProcessing che sono applicazioni software realizzate con l'ambiente di programmazione Processing. Nell'ambito della progettazione di circuiti e hardware open source, due esempi che mostrano come la documentazione possa essere inclusa nel processo di progettazione sono Fritzing e il più recente 123D Circuit ([circuit.io](http://circuit.io)). Fritzing è un'iniziativa open source sviluppata nel 2007 presso Interaction Design Lab della University of Applied Sciences di Potsdam. Il software e la piattaforma web hanno l'obiettivo di facilitare l'apprendimento dell'elettronica e lo sviluppo di circuiti hardware da parte di artisti e designer o persone che non hanno una formazione ingegneristica ([fritzing.org](http://fritzing.org)). Fritzing consiste in uno strumento software e una comunità on-line che condivide la documentazione di progetti di prototipazione elettronica, utile all'insegnamento dell'elettronica e della produzione professionale di circuiti stampati. La caratteristica principale del software Fritzing è la facilità d'uso derivata dalla rappresentazione delle parti elettroniche tramite illustrazioni delle componenti elettroniche e hardware e dalla simulazione dell'ambiente di prototi-

pazione. Il software permette di progettare circuiti utilizzando un'interfaccia semplificata che supporta anche delle funzionalità per esportare gli schemi dei circuiti stampati in diversi formati di immagine in modo da renderli adatti alla pubblicazione tramite tutorial web. L'utente può disegnare il circuito manipolando i componenti e il microcontrollore e aggiungere annotazioni e suggerimenti tramite etichette testuali. Queste funzionalità supportano la documentazione del processo di design e la creazione di tutorial web condivisibili su piattaforme web quali Instructable.

Similarmente alle strategie applicate in OpenProcessing, la piattaforma 123D Circuit ([123d.circuits.io](http://123d.circuits.io)) dell'azienda Autodesk estende le funzionalità offerte da Fritzing: 123D Circuit è infatti un software web che permette di progettare circuiti stampati e, in generale, progetti hardware e prototipi di progetti interattivi tramite la rappresentazione schematica interattiva di componenti hardware come i microcontrollori, i led, le resistenze e i cavi. Il software permette di realizzare il circuito tramite la simulazione delle componenti, di progettare lo schema elettrico, di generare la lista dei componenti utilizzate e aprire il documento di progetto alla collaborazione di altri utenti del sistema. Inoltre, il sistema permette di scaricare i file sorgente del progetto.

L'aspetto più interessante dal punto di vista dell'impatto che questi sistemi hanno sui processi e metodi di progettazione collaborativa e aperta è la possibilità, ancora una volta, di passare da un modello di trasmissione di conoscenza basato sulla simulazione a un modello basato sulla prototipazione. Le simulazioni realizzate tramite i computer sono delle tecniche per restituire e studiare tramite un modello digitale della realtà un comportamento di un fenomeno fisico o di un artefatto, per esempio la relazione alla forza di gravità di un certo oggetto. Le tecniche di simulazione hanno permesso di riprodurre e studiare in laboratorio fenomeni o sistemi troppo complessi che sarebbero impossibili o troppo costosi da ricreare nella realtà (Wikipedia 2014). Una simulazione può basarsi semplicemente su un modello concettuale e matematico che riproduce le caratteristiche del sistema studiato o essere implementata in ambienti immersivi tridimensionali foto-realistici: un esempio sono i simulatori di volo o i videogiochi che sono

Fig. 18 Esempio di tutorial del progetto Tinkercad

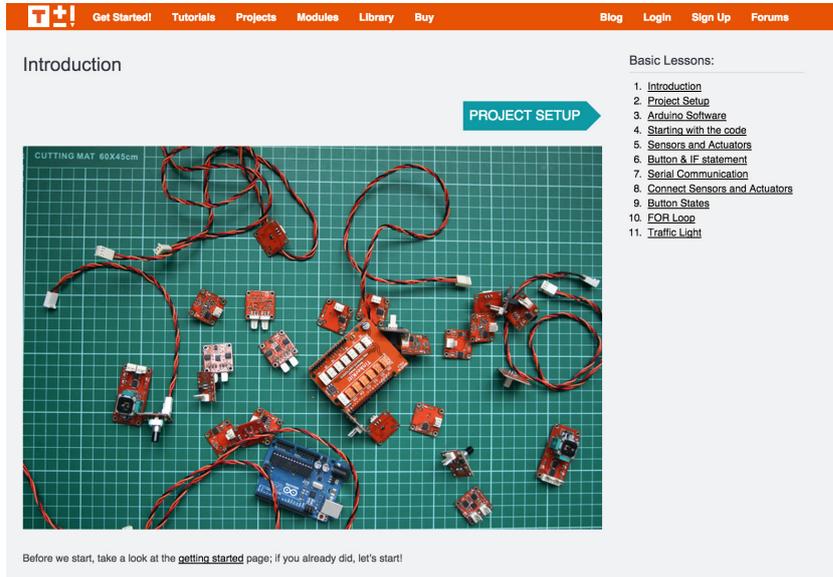
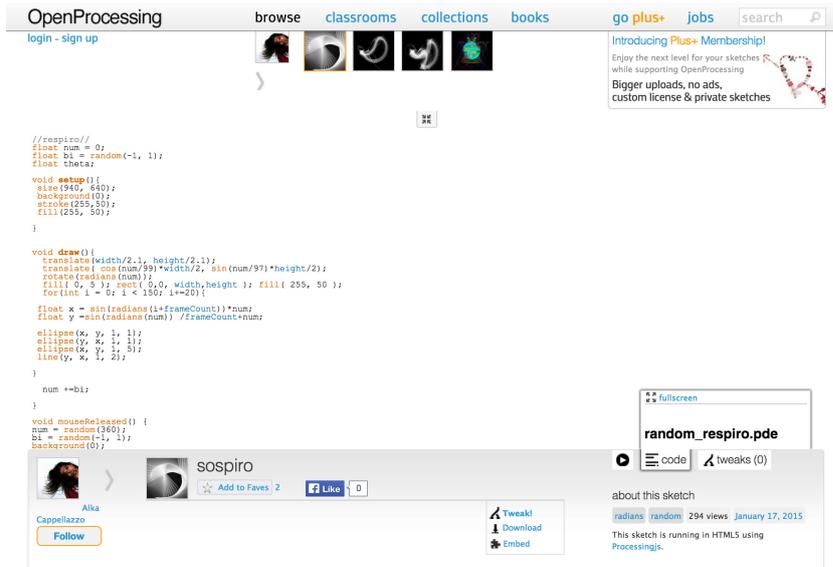


Fig. 19 Uno sketch di OpenProcessing



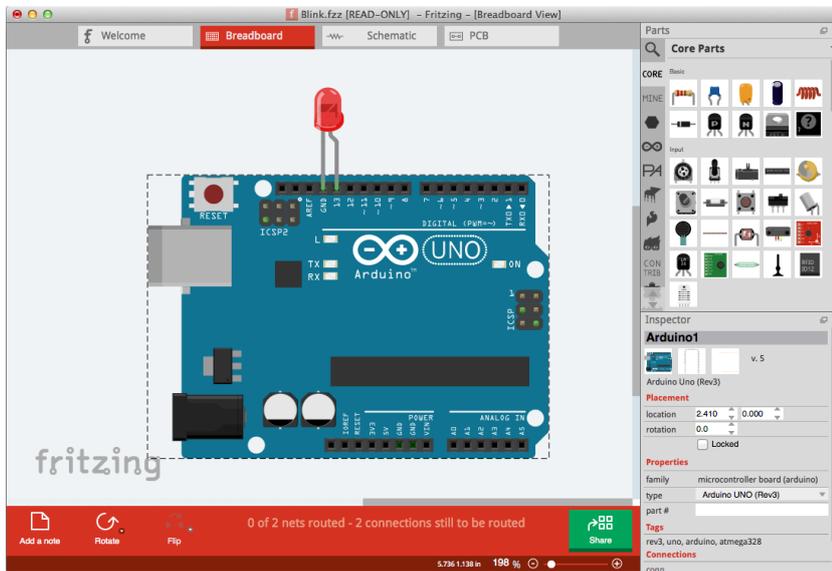


Fig. 20 L'interfaccia di Fritzing

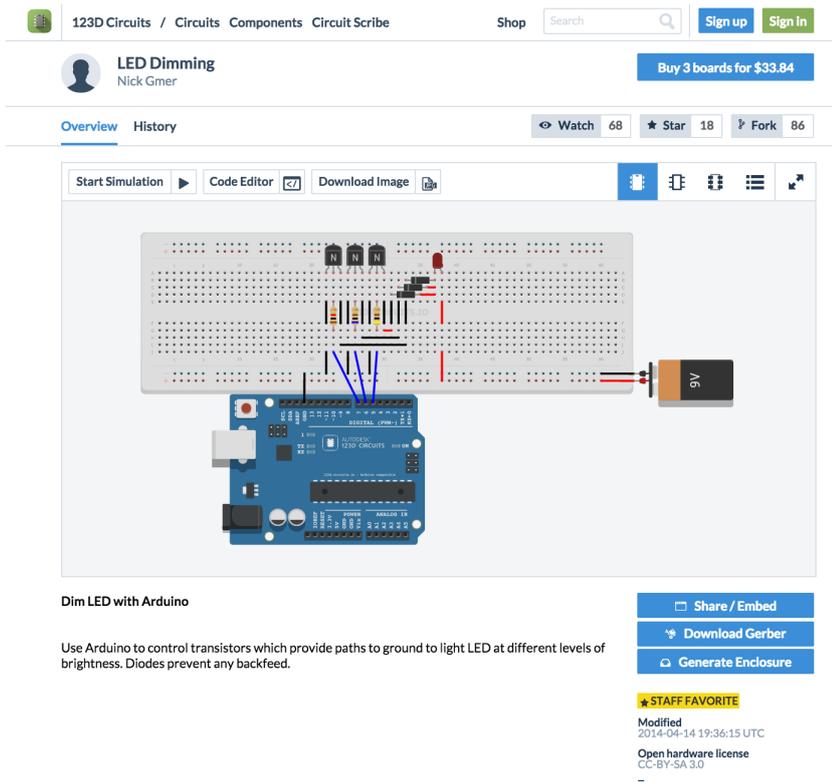


Fig. 21 L'interfaccia di 123D Circuit di Autodesk

utilizzati a scopo formativo per riprodurre simulatore di volo per esempio. Nel sistema 123D Circuit, ci troviamo di fronte a due importanti aspetti di design: da un lato abbiamo l'uso di una metafora per rappresentare per analogia l'artefatto interattivo fisico che si intende realizzare, dall'altro la possibilità di programmare tramite codice le interazioni e il funzionamento degli artefatti è supportato da una simulazione di come sarebbero nel mondo reale quei comportamenti. Gli utenti possono simulare, infatti, la realizzazione di un circuito per attivare un potenziometro che controlla l'accensione di un LED, accedendo sia alla simulazione sia al codice informatico che l'ha generata. Se si intende programmare l'accensione di cinque luci LED rosse in sequenza, bisognerà selezionare i componenti LED dalla libreria, porli sulla *breadboard*<sup>5</sup> digitale, creare connessioni con le resistenze e i connettori di Arduino e scrivere un piccolo programma che regola l'accensione delle luci LED. Una volta completato il circuito, l'utente può verificare il funzionamento tramite il simulatore e verificare il funzionamento del circuito e, in generale, del suo progetto interattivo. L'animazione delle luci LED che si accendono riproduce il funzionamento del progetto fisico finale. Scaricando il file, è possibile trasferirlo nell'ambiente di programmazione originale e ricostruire nel mondo reale il progetto immaginato.

Lo sviluppo delle tecnologie web riduce le distanze tra gli strumenti di progetto e di prototipazione rendendo possibile la traduzione di un sistema basato sulla simulazione in un codice adatto a un sistema di prototipazione.

Questo tipo di processo deriva dalla necessità di supportare le pratiche intorno alla prototipazione elettronica che influenzano la definizione di nuove interfacce e modalità di interazione con software di progettazione.

Come cambia la progettazione in base a queste pratiche emerge dal modo in cui questi artefatti interattivi digitali sono configurati. Questa traduzione non riguarda solo artefatti digitali, come piattaforme e strumenti di progettazione digitali, ma anche

---

5 La Breadboard è un componente utilizzato nell'ambito della prototipazione elettronica che serve a creare circuiti senza la necessità di saldare i componenti elettronici e i cavi.

l'ambito della progettazione di manuali e artefatti di comunicazione a supporto di attività di elettronica fai da te.

Arduino starter kit è un kit a supporto dell'apprendimento della programmazione hardware software di sistemi interattivi tramite la piattaforma Arduino. Il kit è composto dai componenti elettronici (microcontrollore Arduino, resistenze, sensori, led, moduli di legno lavorati al taglio laser) e un manuale che propone tutorial pratici che spiegano la realizzazione di progetti di piccola e media complessità.

Questo prodotto è un esempio di traduzione delle soluzioni per la condivisione di risorse per la prototipazione elettronica fai da te nel contesto della progettazione open source e dell'utilizzo delle piattaforme web collaborative su un supporto cartaceo. Il libro offre un ricettario di progetti e propone gli "ingredienti", le tecniche e gli strumenti per realizzarli.

Il libro e il kit sono stati progettati per soddisfare le esigenze di persone che apprendono la prototipazione elettronica attraverso la realizzazione di progetti ed è stato, quindi, pensato per essere uno strumento di supporto utilizzato durante lo svolgimento delle attività pratiche per la realizzazione dei progetti. Il libro include l'introduzione alle componenti base del kit e, in generale, alle conoscenze di base per utilizzare il microcontrollore Arduino e il suo ambiente di programmazione.

I singoli progetti sono descritti in forma di tutorial che presentano la seguente struttura:

- la descrizione del progetto con le informazioni riguardo al tempo per realizzare il progetto e i principali sensori o attuatori che si utilizzano;
- le istruzioni su come creare il circuito fornite tramite un'illustrazione e lo schema elettrico,
- le informazioni sul codice e il codice stesso da utilizzare per programmare il progetto.

L'aspetto chiave della struttura in cui sono descritti i tutorial si trova nella sezione relativa allo sviluppo del codice: la sezione si presenta su una doppia pagina dove nella pagina a sinistra sono fornite le informazioni sul funzionamento del codice e sulla pagina a destra è visualizzato il codice per implementare il progetto tramite l'ambiente di programmazione Arduino. Le singole parti

del codice e i paragrafi correlati con le informazioni sono perfettamente allineati.

Questo tipo di soluzione grafica riflette il funzionamento della modalità con la quale le persone utilizzano il codice informatico nei contesti del fai da te e rappresenta la traslazione e traduzione di un tutorial web.

#### 2.4 I modelli e le modalità di collaborazione

Il terzo livello di open source è relativo alla gestione della realizzazione di un progetto open source basato sulla collaborazione con un gruppo di persone tramite la rete, la comunità on-line, al fine di lavorare sia sugli aspetti progettuali sia sullo sviluppo del prodotto i cui file sorgente sono stati pubblicati sul web tramite licenze libere.

Le motivazioni alla base del rilascio open source sono strettamente legate ai benefici derivati dalla possibilità di ricevere contributi in maniera volontaria da parte da un numero indeterminato di persone. Per questo motivo gestire la collaborazione con una comunità on-line vuol dire utilizzare o definire dei modelli di interazione che spesso si sovrappongono con un modello di business (creazione, sviluppo e distribuzione) di un progetto open source.

Come è influenzato il design e lo sviluppo di un artefatto interattivo dalla collaborazione e interazione con una comunità di utenti on-line? Per poter avere un quadro completo di questo aspetto bisogna far riferimento all'ambito dello sviluppo di software open source che propone dei modelli molto definiti e di consolidato successo. Pensiamo, infatti, al modello di sviluppo del *kernel*, il modulo software di un sistema operativo, del progetto open source software più diffuso e conosciuto al mondo: Linux. Linux ha raggiunto un successo su scala globale diventando la piattaforma utilizzata da molti servizi internet quali Google, Twitter, Facebook e dal sistema operativo Android che, a sua volta, è utilizzato in molte delle piattaforme di sviluppo per telefonia cellulare.

A differenza degli altri sistemi operativi come Windows e iOS di Apple, lo sviluppo di Linux si basa sulla collaborazione tra una comunità globale di sviluppatori software, il cui numero già solo nel 2005 era di circa 8000 provenienti da circa 800 compagnie che hanno scelto di utilizzare Linux come piattaforma di base per i loro prodotti software (Linux Foundation 2015). Il progetto è iniziato nel 1991 quando Linus Torvald, studente di informatica ad Helsinki inviò un messaggio su un gruppo Minix in cui annunciava che stava sviluppando un sistema operativo a livello amatoriale che immaginava non sarebbe mai stato stabile come GNU, il sistema operativo

open source creato da Richard Stallman, attivista del software libero e iniziatore del progetto GNU. Dopo alcuni anni, Linux è diventato un caso studio che dimostra come la creazione di un modello di sviluppo basato sulla collaborazione su larga scala crei un prodotto stabile e preferibile ad altri prodotti software sviluppati con modelli classici di ricerca e sviluppo interni alle aziende e proprietari. Il primo punto di forza della metodologia di sviluppo di Linux è rappresentato in particolare dalla velocità di rilascio del software che ha portato nell'ambito dello sviluppo software a prediligere i modelli di implementazione basati su molteplici iterazioni: il Kernel Linux è aggiornato in media ogni due o tre mesi a differenza di quelli proprietari. Il secondo punto di forza è la formalizzazione del processo di sviluppo che si distingue rispetto ai modelli e alle buone pratiche degli albori del software libero.

Il processo di sviluppo del Kernel Linux si basa sui seguenti passaggi (Linux Foundation 2015):

1. gli sviluppatori creano dei cambiamenti in blocchi del codice di Linux che sono definite *patches*;
2. ogni patch descrive la linea di codice che dovrebbe essere cambiata, aggiunta o rimossa dal codice sorgente;
3. ogni patch può includere una nuova funzionalità, il supporto per un nuovo dispositivo, risolvere un problema, migliorare la performance o proposte di re-design completo;
4. gli sviluppatori inviano la loro patch alla mailing list ufficiale degli sviluppatori Linux che possono commentare i cambiamenti proposti;
5. quando una patch è considerata pronta sulla base dei commenti e dell'elevato interesse da parte della comunità, uno sviluppatore esperto inizia a valutarla per integrarla nel codice Linux. Uno sviluppatore esperto è di solito la persona responsabile per il mantenimento delle sezioni dei *repository* ufficiali del progetto Linux, ovvero le cartelle dove sono conservati i file del software in rete;
6. se la valutazione del codice proposto ha esito positivo, lo sviluppatore esperto la invia a Linus Torvald in persona e ai membri della fondazione Linus Torvald. Linus Torvald è l'unico che ha l'autorità finale di accettare quali contributi faranno parte dell'ultima versione del *Kernel*. A oggi, sono proposte circa 10.000 patch per ogni nuova versione di cui solo sei all'ora sono integrate nella

36

Project 02  
 Spaceship Interface

## THE CODE

Some notes before you start

Every Arduino program has two main functions. Functions are parts of a computer program that run specific commands. Functions have unique names, and are "called" when needed. The necessary functions in an Arduino program are called `setup()` and `loop()`. These functions need to be declared, which means that you need to tell the Arduino what these functions will do. `setup()` and `loop()` are declared as you see on the right.

In this program, you're going to create a variable before you get into the main part of the program. Variables are names you give to places in the Arduino's memory so you can keep track of what is happening. These values can change depending on your program's instructions.

Variable names should be descriptive of whatever value they are storing. For example, a variable named `switchState` tells you what it stores: the state of a switch. On the other hand, a variable named "x" doesn't tell you much about what it stores.

Let's start coding

To create a variable, you need to declare what *type* it is. The *data type* `int` will hold a whole number (also called an *integer*); that's any number without a decimal point. When you declare a variable, you usually give it an initial value as well. The declaration of the variable as every statement must end with a semicolon (;).

Configure pin functionality

The `setup()` runs once, when the Arduino is first powered on. This is where you configure the digital pins to be either inputs or outputs using a function named `pinMode()`. The pins connected to LEDs will be OUTPUTS and the switch pin will be an INPUT.

Create the loop function

The `loop()` runs continuously after the `setup()` has completed. The `loop()` is where you'll check for voltage on the inputs, and turn outputs on and off. To check the voltage level on a digital input, you use the function `digitalRead()` that checks the chosen pin for voltage. To know what pin to check, `digitalRead()` expects an *argument*.

Arguments are information that you pass to functions, telling them how they should do their job. For example, `digitalRead()` needs one argument: what pin to check. In your program, `digitalRead()` is going to check the state of

Fig. 22 Arduino Starter Kit. Soluzione di impaginato che riprende la struttura dei tutorial web e delle pratiche di sviluppo software basate sui commenti.

NOT FOR DISTRIBUTION  
 For Intro to Physical Computing  
 at ITP/NYU 2012

versione finale del Kernel. Questo tipo di modello e i tempi di sviluppo rendono il progetto Linux unico e difficilmente ripetibile.

Il modello di sviluppo di Linux si basa sull'utilizzo di sistemi informatizzati che giocano un ruolo fondamentale nella gestione della collaborazione nello sviluppo del software open source. La necessità di dover scrivere codice in maniera collaborativa ha portato alla definizione di strumenti che sono stati pensati per verificare le diverse versioni di codice, commentarle e confrontarle. Per il progetto Linux gli strumenti di comunicazione e sviluppo collaborativo sono principalmente il software Git e la mailing list ufficiale ([lkml.org](http://lkml.org)). Git è un software gratuito e open source che rientra nella categoria dei *Distributed Version Control System*: questo tipo di sistemi permettono di conservare e comparare le versioni di software scritte da diversi sviluppatori informatici, i quali lavorano direttamente su una copia in locale del *repository* del progetto. Il sistema è distribuito perché ognuno possiede una copia del progetto e può integrare i cambiamenti all'interno di un *repository* principale. Git permette a chi gestisce il progetto di verificare tutti i contributi, di poterli confrontare mettendo in evidenza le differenze e di poterle fondere insieme.

In modo simile al progetto Linux, anche altri progetti di software open source basano il loro sviluppo e il loro successo su modelli di collaborazione strutturati come, per esempio, Apache, il web server, e il browser Mozilla. Apache e Mozilla hanno un sistema basato sulla valutazione delle modifiche da parte di sviluppatori esperti, ma a differenza di Linux, gli stessi sviluppatori esperti sono anche quelli che possono decidere di integrare i cambiamenti nei moduli del software avendo l'accesso alle cartelle di un *repository* principale. L'aspetto decisionale è delegato a chi ha contribuito scrivendo la maggior quantità di codice e di migliore qualità. Nel caso di Mozilla, la decisione finale spetta alla fondazione Mozilla.org. In una comparazione dei due progetti effettuata tramite interviste ai realizzatori di Apache e Mozilla e all'analisi delle mail delle mailing list, Mockus et al (Mockus et al 2002) spiegano come il gruppo di Mozilla abbia come ruolo principale quello di supportare la comunità gestendo il piano di pubblicazione delle versioni del software o effettuando il manteni-

mento del sito, della documentazione e degli strumenti come, per esempio, Bugzilla, l'estensione software di Mozilla per verificare il codice delle applicazioni tramite il browser. Gestire la comunità on-line e le interazioni tra i suoi membri è la principale attività svolta da chi sviluppa un progetto open source: gli ideatori di un progetto non sono più i realizzatori, ma i gestori delle interazioni tra gruppi di persone in rete. Questo tipo di stato non è specifico solo dell'ambito dello sviluppo software open source, ma rispecchia l'evoluzione dell'interazione on-line a seguito della diffusione delle applicazioni di *social networking*. Creare una comunità di interesse, stimolarla, fare in modo che i suoi membri partecipino e contribuiscano è un aspetto fondamentale di qualsiasi azienda che basa il suo modello di business sulla partecipazione degli utenti tramite la rete. Nasce in ogni azienda la figura del *community manager* poiché la creazione di una comunità di persone che seguono e partecipano attivamente ad un progetto è un obiettivo complicato da raggiungere.

La connettività on-line significa che abbiamo sempre più l'abilità di comunicare con persone vicine e lontane senza limiti di luogo e tempo. Costruire e mantenere una relazione significativa con queste persone, sia essi impiegati, clienti o fan sarà una competenza chiave di ogni organizzazione che funziona (McDonald 2014).<sup>6</sup>

#### 2.4.1 I modelli di collaborazione per lo sviluppo hardware open source

Lo sviluppo di un progetto di software open source non è, come lo definisce Eric Steven Raymond, autore dello storico saggio *The Cathedral and the Bazaar*, “accuratamente lavorato da maghi o piccoli gruppi che lavorano in splendido isolamento, senza versioni beta che sono rilasciate prima del tempo...Piuttosto è un bazar rumoroso con diversi piani e approcci in cui il codice è scritto con maggiore cura e creatività perché gli sviluppatori lavorano su cose per le quali hanno una reale passione” (Raymond 1999, p. 3). Simile a un bazar, il lavoro nello sviluppo di software open

---

6 McDonald 2014, mia traduzione.

source non è assegnato, ma sono le persone a scegliere che tipo di compito portare avanti, non si basa su un'architettura definita o un design specifico, non ha un piano di sviluppo, un programma o una lista di documenti di progetto da consegnare (Raymond 1999). In generale, lo sviluppo si basa sul lavoro di volontari e su una persona o un gruppo di persone che si prende cura della pubblicazione delle versioni ufficiali e della loro distribuzione. Lo sviluppo di hardware open source si basa su modelli diversi da quelli del software. Nell'ambito dell'hardware va a cadere il concetto del controllo centralizzato e del contributo volontario e distribuito verso un singolo progetto come succede per Linux. Basandosi sulle stesse logiche di partecipazione e appropriazione della tecnologia, i modelli open source per l'hardware propongono il rilascio di hardware in modo che tutti possano adattarlo e modificarlo liberamente in base alle proprie necessità. I principali progetti di sviluppo hardware open source che consideriamo in questo studio sono i microcontrollori a supporto della prototipazione hardware-software e elettronica come Arduino, Buglabs, Littlebits. Queste piattaforme sono casi studio d'interesse poiché sono indirizzati a persone non esperte. L'ampia diffusione di questi prodotti tecnologici, ha permesso al settore dello sviluppo di hardware open source di poter maturare dal punto di vista del business e della comunità di utenti, proponendo un nuovo paradigma di sviluppo tecnologico e innovazione alla stessa stregua del software libero. A differenza di quest'ultimo, i modelli di sviluppo presentano molte differenze. In particolare, nell'open hardware impera la logica della creazione dei derivati piuttosto che dei contributi volontari in forma di *patches*: i derivati sono versioni realizzate a partire da un progetto hardware che possono variare per estetica, forma, tecnica di assemblaggio e componenti.

La piattaforma Arduino ha svariate migliaia di derivati sviluppate da altre aziende concorrenti e lo stesso sviluppo della linea di schede e microcontrollori dell'azienda Arduino si basa sulla creazione di derivati (Torrone 2009). Le variazioni tra i diversi derivati di Arduino riguardano spesso l'integrazione di funzionalità alternative come per esempio la possibilità di poter cucire i contatti degli input nella variante Arduino Lilypad ([lilypadarduino.org](http://lilypadarduino.org)), la scheda hardware che può essere cucita sui tessuti e utiliz-

zata per lo sviluppo di progetti interattivi indossabili. Come sottolinea David Mellis, co-fondatore del progetto Arduino, l'analisi dei derivati di Arduino e l'esperienza nello sviluppo della piattaforma dimostra che le modifiche alla scheda non costituiscono una forma di miglioramento o di conoscenza di ritorno al progetto hardware iniziale (Mellis, Buechley 2012). Rendere un componente hardware open source influenza la diffusione di nuove piattaforme a supporto dell'intera comunità. La differenza con lo sviluppo di software open source risiede anche nel carente uso da parte degli sviluppatori dei progetti open hardware di sistemi quali Git ([git-scm.com](http://git-scm.com)) o Apache Subversion ([subversion.apache.org](http://subversion.apache.org)), per la gestione delle versioni e della collaborazione nella comunità di sviluppatori volontari. Mellis individua diversi ostacoli all'applicazione di modelli di sviluppo del software open source al mondo dell'hardware come, per esempio, la disponibilità di componenti utilizzati per realizzare una scheda in un certo paese, l'elevato investimento nella produzione di una scheda che limitano la possibilità di svolgere iterazioni frequenti sul prodotto e di integrare le modifiche suggerite dalla comunità. Nello sviluppo hardware l'aspetto finanziario fa sì che chi si accinge a progettare e integrare nuove soluzioni preferisca farlo attraverso un proprio marchio e azienda piuttosto che "costruire sopra un marchio già esistente".<sup>7</sup>

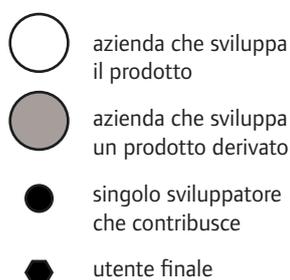
Lo studio di David Mellis documenta il primo stadio di sviluppo di progetti di hardware open source. Ancora oggi i contributi al miglioramento della piattaforma da parte delle comunità on-line sono piuttosto minimi e non influenzano lo sviluppo di questi progetti dimostrando come i modelli di collaborazione nell'ambito dell'hardware seguano delle logiche diverse da quelle del software e sono strettamente dipendenti dal modello interno di business delle aziende che rilasciano il prodotto.

#### 2.4.2 I modelli di collaborazione nello sviluppo del prodotto

Se nello sviluppo di hardware open source i modelli di collaborazione sono dipendenti dai modelli di business e la collaborazione si basa sulla proliferazione di soluzioni nuove e indipendenti (mo-

---

7 ibidem



dello decentralizzato) destinate a un'unica comunità, nell'open design è difficile poter applicare il modello centralizzato del software e allo stesso tempo decentralizzato dell'hardware come, allo stesso tempo, è difficile poter indentificare dei modelli di successo (sia finanziario sia di utenti). L'aspetto collaborativo nello sviluppo di progetti di open design può essere studiato alla luce dell'analisi di

Open source software

Open source hardware

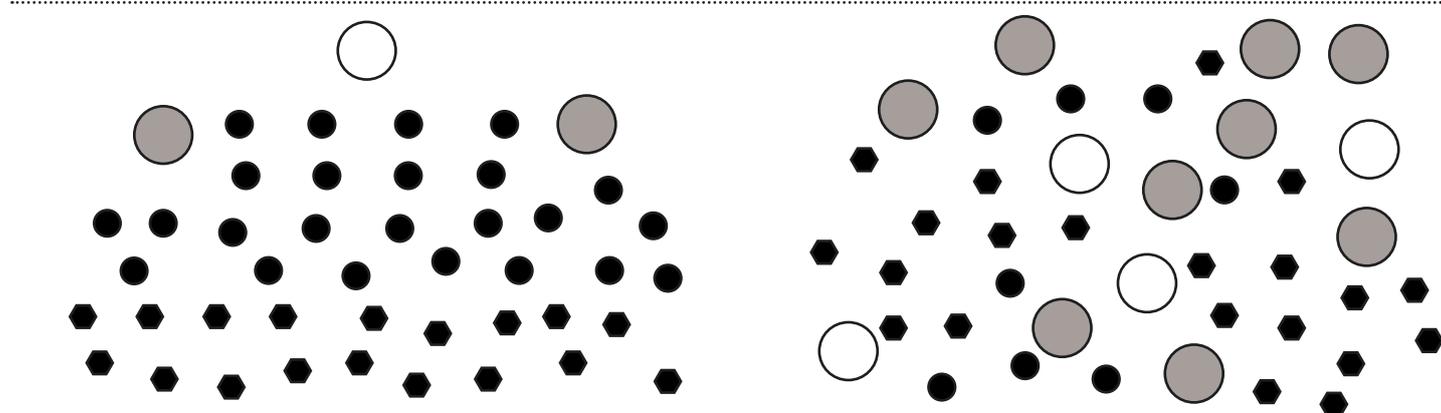


Fig. 23 Lo schema mostra i due modelli di sviluppo del prodotto nell'ambito dell'Open source software e Open source hardware. L'Open source software si basa su un modello di interazione gerarchico poiché l'azienda che sviluppa il progetto valida i contributi delle altre aziende che sviluppano prodotti derivati e i contributi degli utenti sviluppatori. Nell'Open source hardware, il modello è di tipo rizomatico poiché ogni azienda sviluppa un proprio prodotto; nascono molte aziende che sviluppano prodotti derivati; il contributo dei singoli sviluppatori è ridotto, ma aumenta il mercato degli utenti finali e degli sviluppatori stessi.

paradigmi a esso correlati come la *user driven innovation*, la *open innovation* e il *social product development*. I tre paradigmi sono strettamente connessi gli uni agli altri: mentre il primo enfatizza le opportunità d'innovazione che sono generate dalla partecipazione di singoli utilizzatori nel processo di modifica di un artefatto fisico, il secondo si concentra maggiormente su come queste opportunità nascano dall'istituzione spontanea di comunità che collaborano alla risoluzione di un problema senza l'obiettivo di creare profitto, il terzo inquadra un modello di sviluppo ben preciso di artefatti che si basa sul coinvolgimento delle persone nel processo di ideazione e definizione delle caratteristiche di un prodotto (Baldwin, von Hippel 2010).

Un caso studio sul processo collaborativo di sviluppo di un prodotto influenzato da pratiche e approcci propri dell'ambito open source è lo sviluppo della linea di sintetizzatori analogici Monotron (Kobayashi 2013).

Il Monotron è un sintetizzatore analogico sviluppato dall'a-

zienda giapponese Korg che, riconoscendo il movimento del *circuit-bending* e l'enorme comunità di persone interessata all'*hacking* di vecchi sintetizzatori analogici, decide di rilasciare la documentazione del prodotto in internet al fine di sperimentare l'integrazione del contributo degli utenti nel processo di sviluppo del prodotto.

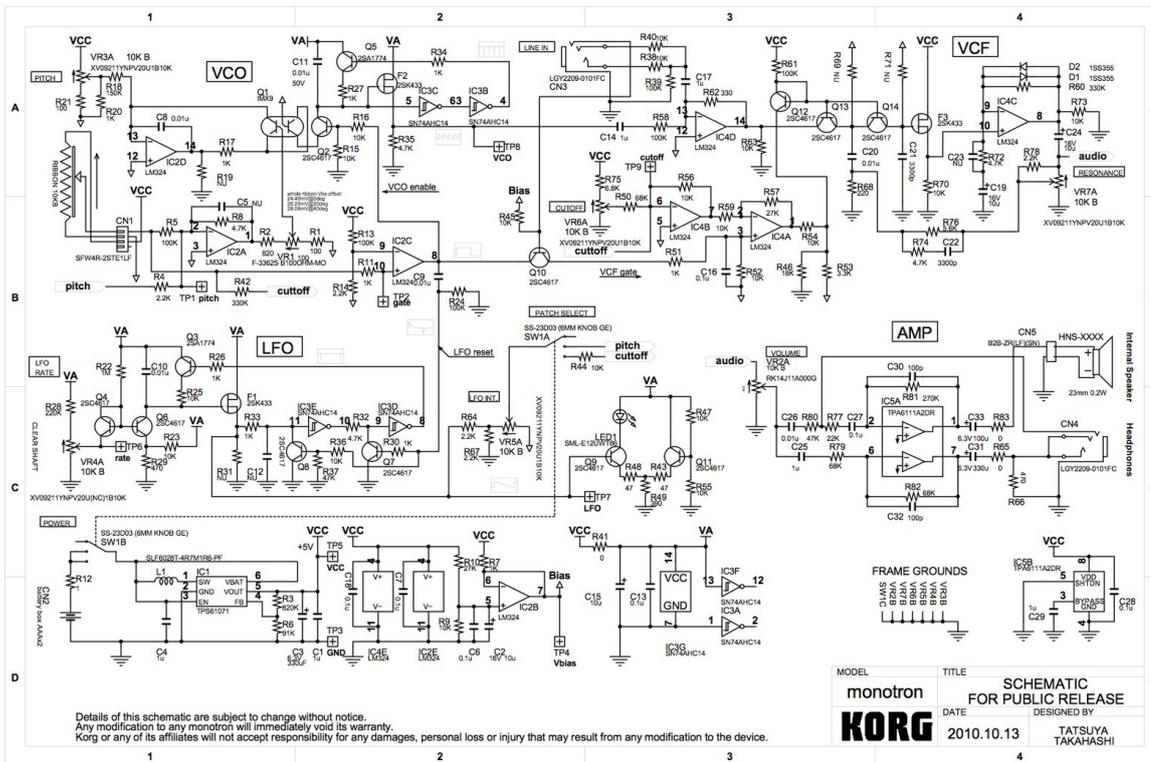
Al fine di anticipare la possibilità per gli appassionati di sintetizzatori analogici di modificare e personalizzare il loro strumento, nel processo di realizzazione dell'hardware, gli sviluppatori di Korg hanno deciso di inserire dei suggerimenti sul retro del circuito stampato per permettere agli utilizzatori di espandere le funzionalità del prodotto in accordo alle ricerche di marketing che evidenziavano come per gli utenti dei sintetizzatori analogici il Monotron è considerato un prodotto amatoriale con poche funzionalità adatte al mondo dei professionisti. In seguito al rilascio su mercato nel 2010, gli utilizzatori del prodotto hanno scoperto le indicazioni lasciate dagli sviluppatori e hanno iniziato a "hackerare" il sintetizzatore integrando delle nuove funzionalità. Durante la partecipazione alla Maker Faire di Tokyo, i rappresentanti dell'azienda Korg hanno individuato tra i progetti degli espositori alcune versioni modificate del Monotron e hanno deciso di menzionarle sul sito ufficiale del Monotron attraverso la realizzazione di una campagna di comunicazione chiamata "We Love Monotron" (Korg 2015). In seguito al successo dell'operazione di comunicazione e all'aumentare delle richieste da parte degli utenti, nel novembre del 2010 Korg ha deciso di rendere pubblico sulla rete il diagramma del circuito stampato del Monotron.

A partire dal 2011, Korg ha iniziato a produrre delle versioni del Monotron ispirate alle soluzioni degli utilizzatori, il Monotron Delay e Monotron DUO. Dopo il 2011 ha continuato a rilasciare prodotti con le indicazioni sul circuito stampato e gli schemi elettrici rendendo il processo di sviluppo aperto. In seguito a questo esperimento, l'azienda Korg è riuscita a trarre informazioni sugli interessi degli utilizzatori rispetto ad un prodotto di nicchia, per il quale i metodi di indagine proprie del marketing avrebbero portato a decidere di bloccare la produzione o lo sviluppo. Dal caso Monotron, Kobaiashi trae le seguenti lezioni:

- incorporare messaggi per gli utilizzatori finali (in particolare, la categoria dei maker e hacker) all'interno di un prodotto permette di trasformarlo in una piattaforma aperta;
- anche se la documentazione del Monotron è aperta, gli utilizzatori continuano a comprarlo al fine di modificarlo e espanderlo invece di creare dei cloni (grazie ai costi ridotti, ma anche alla fidelizzazione e al rispetto degli utilizzatori per il marchio);
- la chiave del successo del Monotron è che non è stato rilasciato in open source un componente tecnologico, ma un prodotto attraente e "hackerabile". Per questo, la possibilità di rilasciare questo tipo di prodotti è una soluzione interessante per favorire lo sviluppo di progetti di *open innovation*.

Kobayashi delinea il seguente schema di interazione tra produttori, utenti e maker. I produttori realizzano prodotti modificabili rilasciandoli attraverso licenze aperte come, per esempio, le licenze

Fig. 24 Schema elettrico del circuito stampato del Monotron



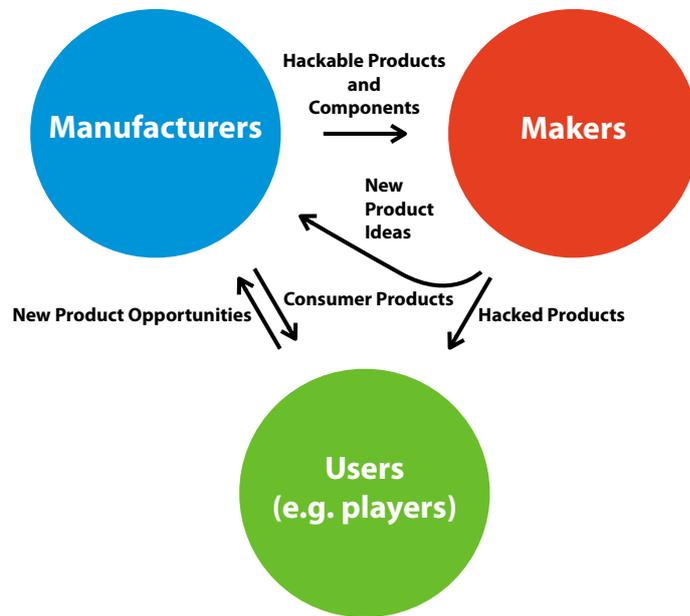


Fig. 25 Schema dell'ecosistema di prodotto di Kobayashi

Creative Commons. I maker realizzano in modo spontaneo derivati del prodotto che si interfacciano con altre tipologie di utilizzatori non esperti, che Kobayashi definisce *player*.

Lo scambio tra questi tre gruppi, porterebbe i produttori più aperti e reattivi allo sviluppo di nuove soluzioni adatte alle esigenze del mercato e questo interscambio favorisce l'innovazione.

Allo stato attuale, questi processi non sono facilmente applicabili alla manifattura di prodotti di consumo di massa, ma piuttosto ai prodotti "periferici" come gli strumenti musicali o ai dispositivi indossabili.

Open design non si riferisce solo agli artefatti fisici rilasciati in modo che altri possano modificarlo migliorarlo e riprodurlo, ma anche all'applicazione di processi mirati alla risoluzione di un problema in modo collaborativo. Questo tipo di processi sono implementati tramite la progettazione di piattaforme digitali dedicate appositamente alla collaborazione tra membri di comunità on-line. Il progetto OpenIDEO è, in tal senso, un caso studio che mette in evidenza come un processo collaborativo sia generato attraverso la creazione di una piattaforma a supporto della colla-

borazione e della creazione di un network, ovvero attraverso un modello di interazione finalizzato alla collaborazione su un progetto complesso.

Il progetto nasce nel 2008 quando IDEO, l'azienda di consulenza e design fondata da Tim Brow, forte dell'esperienza nella conduzione di progetti di design multidisciplinari, decide lavorare su come facilitare la creazione di progetti a sfondo sociale tramite la collaborazione tra membri di comunità on-line. In particolare, cercano di comprendere e risolvere come la collaborazione online possa risolvere gli aspetti di ricerca e sviluppo su problematiche complesse che a priori richiederebbero un investimento in risorse umane troppo eccessivo per un'azienda con il rischio fallimentare di non raggiungere un risultato applicabile nella realtà. La domanda di ricerca iniziale di IDEO è:

Cosa succederebbe se invece di 500 designer di IDEO che risolvono problemi globalmente avessimo un network di 50.000 persone? Che tipo di impatto positivo avremmo sul mondo? (OpenIDEO 2013)

Il primo problema che IDEO punta a risolvere attraverso il lancio del progetto OpenIDEO nel 2010 è stimolare la consapevolezza nei bambini del cibo fresco. A partire da quella sfida, il progetto ha proposto in seguito altre sfide di cui molte sono state tradotte in progetti reali.

Come definito dai realizzatori, OpenIDEO è “uno strumento online per guidare le persone attraverso un processo creativo” e il cuore del progetto è un prodotto software proprietario, OIengine, rilasciato da IDEO per supportare organizzazioni e aziende nella risoluzione di problemi complessi ([OIEngine.com](http://OIEngine.com)).

Il processo collaborativo supportato dalla piattaforma OpenIDEO si basa sulla creazione di sfide.

OpenIDEO formula delle sfide sotto forma di domande. Le sfide sono formulate dai realizzatori di OpenIDEO o da organizzazioni che sponsorizzano il progetto; in seguito alla pubblicazione della sfida, inizia una fase di ricerca e d'ispirazione durante la quale tutti gli utenti registrati partecipano

contribuendo con interviste, dati o semplici storie pubblicate in forma di articoli, immagini e video. Alla fine della fase di ricerca, i membri della comunità di utenti registrati possono proporre singolarmente una soluzione che può essere ideata anche partire da soluzioni proposte precedentemente da altri utenti. L'idea passa poi a una fase di affinamento in cui gli ideatori devono fornire una documentazione avanzata con i dettagli dell'idea, il prototipo o il test della soluzione in un contesto sperimentale. Al termine di questo processo, inizia una fase di valutazione basata sui commenti e i riscontri da parte di tutti i membri della comunità.

OpenIDEO risolve la collaborazione attraverso la realizzazione di uno strumento la cui architettura è basata sui principi delle applicazioni di social networking. Queste applicazioni possono essere definite come strumenti web i cui requisiti fondamentali sono i gruppi di funzionalità che permettono agli utenti di connettersi gli uni con gli altri e condividere contenuti on-line. Strumenti di networking sono i forum, le intranet e i portali che sono caratterizzati da queste funzionalità, ma a differenza di questi, le applicazioni e i servizi di social networking emergenti presentano specifiche peculiarità: gli utenti possono creare un profilo pubblico o semipubblico; creare una lista di connessioni con specifici gruppi di persone all'interno del sistema; trasferire queste connessioni con altri utenti del sistema (Boyd, Ellison 2014). Un servizio di social networking è tale poiché è caratterizzato da questa architettura centrata sugli individui (*individual centered*) mentre i servizi a supporto delle comunità on-line sono propriamente centrati sui gruppi (*group centered*) (Wikipedia 2014).

L'architettura di OpenIDEO è un ibrido tra il modello *individual centered* e *group centered* poiché i membri si connettono a una sfida, ma singolarmente possono contribuire mostrando un profilo pubblico in cui sono collezionati i contenuti, le idee, le ricerche.

Al fine di enfatizzare il ruolo del singolo individuo nel processo comunitario e collaborativo della risoluzione della sfida, OpenIDEO introduce alcune strategie di *gamification*. Per *gamification* si intende l'uso di elementi del game design in contesti che

non sono propri del gioco (Deterding et al 2011). La *gamification* di OpenIDEO si basa sull'assegnazione di un punteggio tramite la misurazione del valore del *design quotient* di ogni membro. Il *design quotient* è la misurazione delle competenze del singolo utente sulla base della partecipazione alle fasi ricerca, ideazione e di valutazione mirate alla risoluzione della sfida.

Terzo elemento dell'architettura della collaborazione di OpenIDEO è relativo all'introduzione del bottone "*Build on this*" che permette ai partecipanti nella fase di ideazione di contribuire attraverso un'idea derivata da un'altra idea: questa funzionalità favorisce la trasparenza del processo creativo collaborativo mostrando l'efficacia di un meccanismo tipico delle piattaforme di condivisione di file originali di design che abbiamo visto nei paragrafi precedenti (p.e. Thingiverse.com).

L'architettura della partecipazione proposta dalle applicazioni di social networking ha un ruolo fondamentale nella diffusione di pratiche di co-creazione nel design (Piller et al 2012). OpenIDEO è un esempio che dimostra la sua efficacia sul piano dei processi di co-creazione, ma tale efficacia è dimostrata anche in contesti di sviluppo di un artefatto fisico. Si parla, infatti, di *social product development* come un paradigma nella progettazione e creazione di artefatti tramite l'interazione di comunità on-line. Il caso del Monotron è un esempio di *social product development* per il quale non è stato progettato a priori un modello e uno strumento per l'interazione e la collaborazione tra gli utilizzatori del sintetizzatore. Lo strumento informatizzato per collezionare i derivati fai da te del Monotron è, infatti, un sito web con contenuti redazionali e promozionali - il sito I love Monotron - pubblicato in rete in seguito alla scoperta casuale dei derivati presentati ad una Maker Faire. Nonostante "la spontaneità" sottostante il caso Monotron, i modelli di sviluppo di un prodotto basato sulla co-creazione da parte degli utenti sono diffusi e di consolidato successo (von Hippel 1986) e assistiamo oggi a una fase in cui questi modelli sono affiancati da strumenti online progettati per facilitare la collaborazione degli utenti e ottimizzare i benefici della loro partecipazione. In tal senso, Quirky è l'esempio migliore di implementazione del paradigma

dello sviluppo collaborativo applicato all'ambito della produzione di artefatti e dispositivi tecnologici di largo consumo.

Si tratta di uno strumento on-line, una piattaforma, che offre la possibilità a chiunque di proporre un'idea per un prodotto e far sì che dopo un processo di revisione e voto da parte della comunità di utenti registrati questa idea possa essere prototipata e realizzata e poi essere messa in vendita attraverso uno negozio on-line (McKeough 2011). Il modello di Quirky si basa su tre elementi principali:

- da un lato propone l'abbassamento delle barriere riferite alle competenze di progettazione per l'utente finale che può anche presentare un'idea descrivendola con un teso e senza fornire un documento dettagliato di progetto; per questo motivo, la piattaforma non presenta *toolkit* di progettazione caratterizzati da interfacce e strumenti semplificati a supporto della progettazione in 2D o 3D di un oggetto;

- capovolge la logica del contributo volontario propria dei progetti di sviluppo open source poiché la partecipazione è remunerata, ovvero la motivazione dell'utente a fornire un'idea di prodotto è basata sull'opportunità di ricevere delle percentuali sulla vendita del prodotto; allo stesso modo la partecipazione di chi fornisce un riscontro sulle fasi del progetto (design, miglioramento, nome del prodotto e definizione del progetto) è anch'essa un'attività retribuita;

- l'implementazione del progetto, dalla prototipazione alla messa in produzione è gestita da Quirky e il suo gruppo interno di design e sviluppo.

Seppur Quirky incorpora nel suo modello e architettura tutti i livelli di un processo di design partecipato, la mancanza dell'utilizzo di licenze libere fa sì che non sia considerabile come un vero e proprio progetto open source. Il suo modello di business rientra piuttosto in un paradigma più vicino a quello della *peer-to-peer economy* in cui i principi di benessere condiviso sono capovolti in favore di un ritorno economico condiviso. Il suo modello basato sul contributo a pagamento è in contrasto con i risultati di studi sul paradigma della *user innovation* i quali dimostrano come la disseminazione gratuita di design innovativi

tramite un modello open source generi un impatto positivo sul benessere sociale e non solo di tipo economico (Baldwin, von Hippel 2010).

Che la collaborazione tra persone con formazione e competenze diverse sia veicolo d'innovazione nella produzione di artefatti innovativi tecnologici non è una scoperta messa in evidenza dai modelli di design e produzione open source. Nella ricerca scientifica, questo tipo di modello ha reso possibile la creazione e diffusione di molte delle tecnologie che utilizziamo quotidianamente. Il modello DARPA, per esempio, è un riferimento storico se si parla di modelli di collaborazione per produrre innovazione. DARPA, Defense Advanced Research Projects Agency del Pentagono, ha rilasciato in cinquanta anni le tecnologie innovative per uso militare che poi sono entrate a far parte dei dispositivi della nostra vita quotidiana (Cardinal 2011).

Il modello DARPA si basa su tre elementi principali: gli obiettivi sono sempre molto ambiziosi, il gruppo di lavoro è temporaneo e composto da membri di diverse organizzazioni, aziende private e laboratori universitari, c'è sempre indipendenza nella scelta dei progetti da sviluppare. Il modello DARPA è stato utilizzato presso altre organizzazioni come, per esempio, l'Advanced Technology and Projects (ATAP), il gruppo di ricerca e sviluppo della Motorola Mobility che è stato acquisito nel 2012 dall'azienda Google. Applicando il modello DARPA, l'ATAP ha dato inizio al Project ARA, il progetto per il design e lo sviluppo di un dispositivo di comunicazione mobile *smartphone* basato su un'architettura hardware, software e di design modulare. Similmente al concetto proposto da *PhoneBlock*, progetto nato nel 2013 per proporre un concetto di *smartphone* modulare in cui si riducessero i rifiuti elettronici, ProjectARA propone il rilascio sul mercato di un prodotto-piattaforma, ovvero un dispositivo di comunicazione mobile che ha le qualità di modularità e di espandibilità simili ad una piattaforma di prototipazione hardware. Il dispositivo è pensato per fare in modo che la comunità degli sviluppatori possa realizzare nuovi moduli del telefono con un ampio livello di libertà, dall'integrazione di nuove funzionalità a diverse soluzioni di design del prodotto. Nello scenario di

massima espandibilità, l'ARA phone potrebbe non avere più la funzione di un telefono, ma essere utilizzato, per esempio, come una macchina fotografica.

La struttura del dispositivo si basa tecnicamente su un *Endoskeleton*, ovvero una struttura rigida basata su una griglia misurata dalla grandezza dei moduli funzionali e che contiene i moduli e li fa comunicare tra loro; i moduli, che conferiscono al telefono le diverse funzionalità come il display, il microfono, la batteria e gli accessori come la videocamera, il microfono.

I moduli sono il cuore centrale del progetto poiché attraverso di essi è possibile customizzare il dispositivo: uno dei vari scenari d'uso è per esempio dare in utilizzo ai bambini il modulo del telefono con il lettore mp3 e al raggiungimento dell'età giusta per avere un telefono, agganciare al blocco mp3 tutti gli altri blocchi funzionali (Project Ara 2014).

L'idea alla base di Project ARA è assicurare non solo la customizzazione del dispositivo dal punto di vista delle funzionalità attraverso lo sviluppo di diverse tipologie di moduli, ma anche dal punto di vista del design: l'involucro esterno dei moduli può essere prodotto tramite tecnologie di stampa tridimensionale.

La sfida del progetto risiede principalmente nel rendere possibile la creazione di un dispositivo tecnologico completamente modulare, ma l'aspetto di interesse nel contesto di questo studio è il processo attraverso il quale si punta a raggiungere questo tipo di obiettivo.

Lo sviluppo del dispositivo è gestito in maniera aperta e collaborativa tramite la combinazione del modello DARPA, ovvero la collaborazione tra un gruppo di lavoro temporaneo e circa venti partner accademici e industriali, e il modello del *social product development*: chiunque può contribuire all'ideazione, al design e alla prototipazione dei moduli tramite l'invio di contributi condivisi tramite la piattaforma di ricerca utente Dscout ([dscout.com](http://dscout.com)). DScout è uno strumento per condurre la ricerca utente attraverso l'uso di un'applicazione mobile: per Project ARA è stato chiesto ai partecipanti, sviluppatori e ricercatori o persone interessate di svolgere dei compiti (definite missioni) e inoltrare tramite foto e testo i risultati delle loro attività (*snippet*). Per circa un anno la comunità di sviluppatori e interessati hanno condiviso le loro idee e

prototipi al fine di definire un prodotto stabile che rispecchiasse le reali esigenze e possibilità tecnologiche. Come dichiarato dal leader del progetto Paul Eremenko, Project ARA ha la durata di due anni e l'obiettivo ambizioso di creare un prodotto che permetta di realizzare un passaggio di paradigma, dal consumo di massa alla co-creazione di massa (Fab10 2014).

Fig. 26 Interfaccia di OpenIDEO

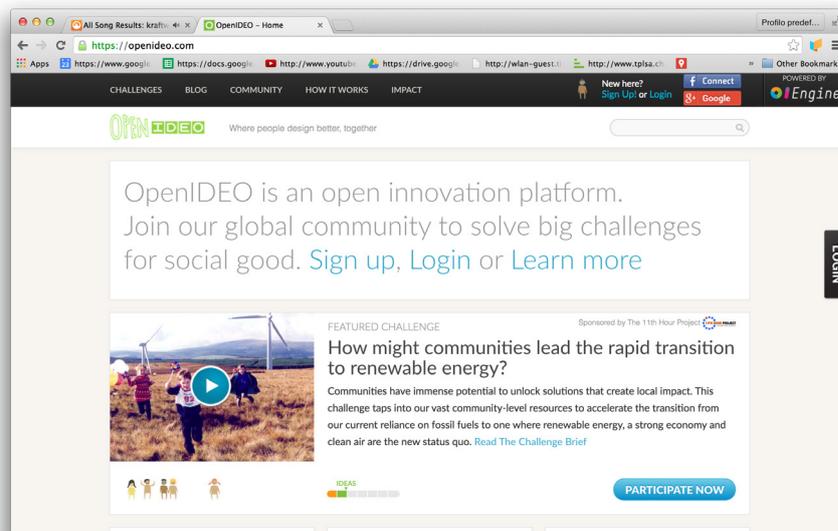


Fig. 27 Interfaccia di Quirky

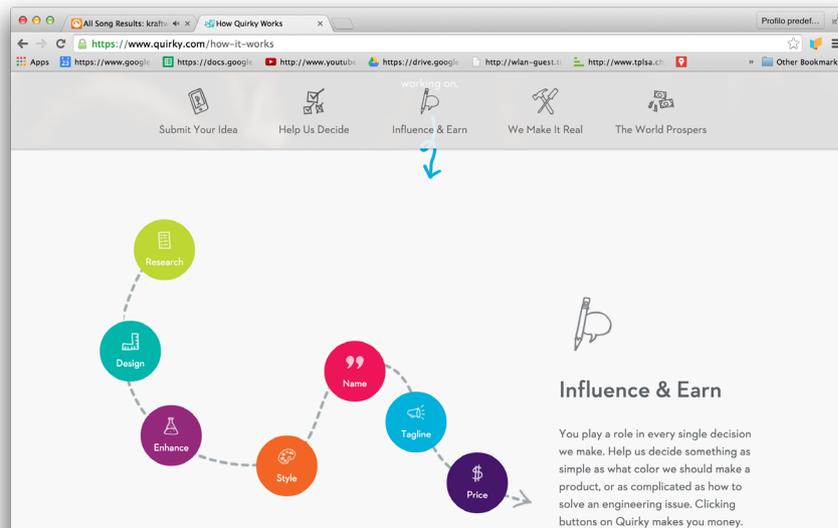




Fig. 28 Ara phone, il telefono modulare sviluppato da Project ARA

## 2.5 I sistemi modulari e parametrici

La grande e principale innovazione in Unix era la “pipe”. Una “pipe” è un meccanismo uniforme per connettere l'output di un programma con l'input di un altro programma. Esso era un elemento chiave per la modularizzazione della programmazione che è la suddivisione di un compito complesso in una serie di compiti semplici che possono essere svolti essenzialmente da moduli autonomi che comunicano attraverso interfacce standard. La modularizzazione ha portato all'idea di software non come uno strumento in sé ma come un “toolbox” (Weber 2005)

I modelli di co-creazione e open source basati sulla collaborazione di comunità di persone tramite la rete crescono e si diffondono per una serie di fattori esogeni molto potenti quali, per esempio, la diffusione di pratiche di progettazione modulare; la decrescita dei costi di progettazione e collaborazione dovuto alle tecnologie di Computer Aided Design; la riduzione dei costi di comunicazione dovuta alla crescita della connettività internet che influenza la riduzione dei costi nella diffusione di artefatti attraverso network peer-to-peer (Baldwin, von Hippel 2010). Tra questi fattori esogeni analizziamo in questo paragrafo in particolare la diffusione di pratiche di progettazione basate sulla modularità e il design parametrico. La definizione di soluzioni di design modulare e parame-

trico è il livello di azione, forse principale, del design dell'interazione nel contesto dello sviluppo di tipo open source.

### 2.5.1. La modularità

Iniziamo dall'assunto che la modularità nell'ambito della progettazione software e hardware non è un concetto nuovo: nello sviluppo software c'è stata l'introduzione del paradigma della programmazione orientata agli oggetti (OOP, Object Oriented Programming) che si basa sulla creazione di moduli software in grado di interagire gli uni con gli altri attraverso lo scambio di messaggi, ovvero oggetti che sono caratterizzati dal principio di ereditarietà che permette di aggiungere nuove classi a partire da una esistente (Wikipedia 2014). Nello sviluppo di progetti di software libero, per esempio, la modularità è anche un fattore che aumenta l'accessibilità del software perché permette a chi contribuisce di lavorare su moduli specifici del software e, allo stesso tempo, di rendere riusabile il codice per altri progetti.

Nel contesto dello sviluppo hardware, la modularità è altrettanto un fattore chiave come nello sviluppo software: la IBM negli anni sessanta propone sul mercato il computer mainframe System/360 che era basato su un'architettura modulare che influenzò il modo di progettare i desktop computer, introducendo anche diversi standard industriali per la produzione di computer (Computer History Museum 2014). Grazie a queste innovazioni, introdotte già negli anni sessanta, si è plasmato il concetto stesso di *personal computer* che deriva dalla combinazione di hardware composti da diversi moduli che comunicano tra di loro come la CPU (*Central processing unit*) e gli *hard drive*. Nell'ambito della progettazione di Open Source Hardware, la modularità è un principio chiave soprattutto rispetto ai requisiti di semplificazione dovuti all'allargamento del mercato verso gli utilizzatori non esperti. Le piattaforme di prototipazione hardware e software open source come Arduino e LittleBits sono basate sul concetto di modularità: in Arduino è stato introdotto lo sviluppo delle cosiddette *shield*, ovvero schede compatibili con Arduino che possono essere impilate su Arduino e a loro volta essere impilate le une con le altre. L'impilabilità, che nell'ambito della progettazione del prodotto è

un attributo introdotto per risolvere, per esempio, la questione della trasportabilità, nella piattaforma di open source hardware è la soluzione che permette di espandere le funzionalità dell'hardware stesso che, nel caso di Arduino, consiste in una scheda che funziona da cervello del sistema e da *shield* che, invece, offrono funzionalità avanzate (Mellis et al 2007). L'impilabilità, inoltre, è la soluzione tecnica che risolve la comunicazione tra Arduino e le *shield* che sono progettate come moduli e anche la soluzione di interfacciamento tra hardware compatibili tra i quali avviene un'interazione e una comunicazione.

In LittleBits, la modularità è il cuore del progetto poiché il kit si basa su moduli magnetici che possono essere combinati tra di loro per creare un circuito elettrico e un sistema interattivo funzionante. Ogni modulo supporta un sensore o un gruppo di componenti elettronici che restituiscono un tipo di funzionalità: per esempio, il modulo tastiera di Little Bits presenta una serie di bottoni che fungono da interfaccia tangibile che permette agli utilizzatori di riprodurre delle melodie. La combinazione in sequenza tramite i connettori magnetici di ogni modulo permette di prototipare e realizzare dei progetti interattivi in modo molto semplice, senza richiedere agli utenti di avere competenze di saldatura.

La modularità delle piattaforme di prototipazione open source deriva da pratiche storiche di progettazione e sviluppo hardware e allo stesso tempo è parte dagli emergenti principi di design di piattaforme indirizzate al mondo dell'insegnamento dell'elettronica per non esperti e per ragazzi. In un articolo pubblicato su Make Magazine Massimo Banzi, fondatore del progetto Arduino, racconta come il suo avvicinarsi al mondo dell'elettronica sia iniziato da bambino con l'utilizzo del *Lectron* e come questo kit abbia influenzato molto l'esperienza del progetto Arduino e, in generale, l'attenzione all'esperienza utente (Banzi 2013). Inventato da Georg Franz Greger e prodotto dall'azienda tedesca Egger, il *Lectron* fu riprogettato, prodotto e distribuito alla fine degli anni sessanta dall'azienda Braun con il contributo del noto designer tedesco Dieter Rams ([lectron.info](http://lectron.info)). Il kit consiste in una serie di moduli che si connettono magneticamente e permettono di realizzare progetti di elettronica in modo semplice: ogni modulo ha una forma cubica e contiene un componente elettronico (p.e. una resistenza, un amplificatore). Il

Fig. 29 IBM System/360



Fig. 30 I connettori delle shield di Arduino e i moduli di Littlebits

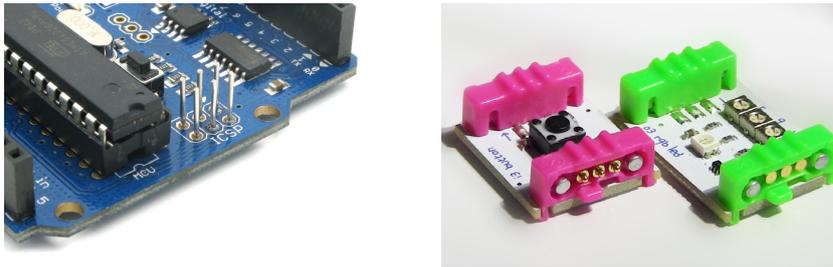
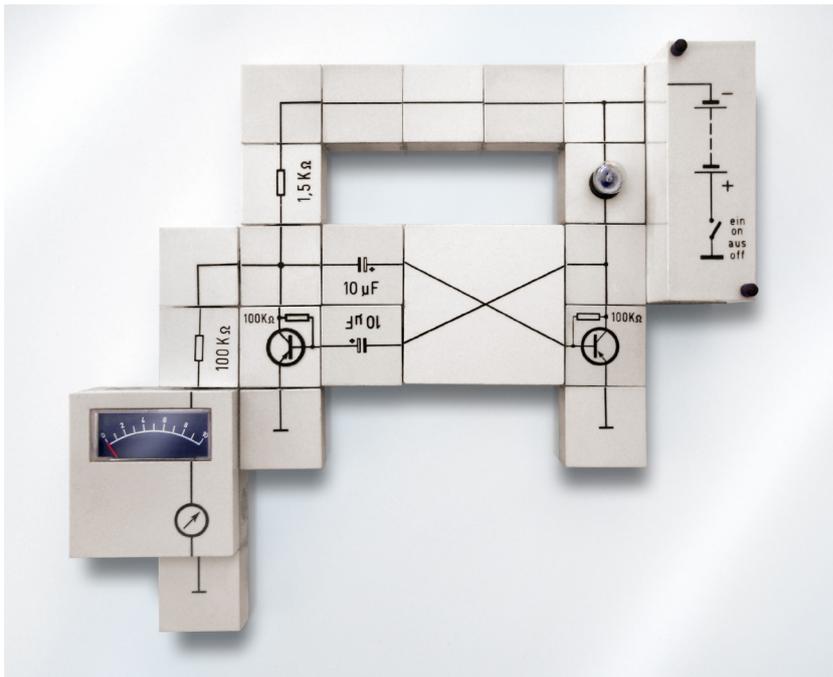


Fig. 31 I moduli del kit per prototipare oggetti interattivi Littlebits

Fig. 32 Il Raytheon Lectron kit



lato superiore di ogni modulo presenta il simbolo grafico utilizzato nelle notazioni degli schemi elettrici. Gli altri lati del modulo sono trasparenti e permettono di visualizzare il componente elettronico all'interno. Il *Lectron* è un esempio di semplificazione della tecnologia tramite interfacce tangibili modulari in cui è implementato un approccio alla progettazione di artefatti a scopo didattico adottato in molteplici progetti come, per esempio, il LEGO Mindstorm ([mindstorms.lego.com](http://mindstorms.lego.com)).

Nel contesto delle discipline del progetto la modularità è un concetto di base. Non mi riferisco all'idea di design modulare come progettazione di artefatti basati su componenti assemblabili e riconfigurabili, ma alla modularità stessa che insita nella realizzazione di un artefatto pensato per la produzione industriale e di massa.

La necessità di produrre artefatti in serie ha portato il design a ragionare su moduli e parti che sono i costituenti del prodotto finale tanto da poter affermare che tutto il design per la produzione di massa è in fondo modulare. La produzione di massa è, infatti, stata resa possibile dalla standardizzazione delle parti che potevano essere assemblate nella catena di montaggio. La catena di montaggio è un processo di produzione in cui parti interscambiabili sono aggiunte a un prodotto in maniera sequenziale per creare un prodotto finale. Il Modello T di Ford è il primo esempio di prodotto modulare poiché pensato per una piattaforma di produzione comune che si basava sulla condivisione dei moduli, dei componenti, dei processi di produzione e dei servizi annessi per ridurre il costo di ogni prodotto: il Modello T era sia un modulo di questa piattaforma sia un prodotto venduto alla rete di produttori che potevano customizzarlo per rispondere alle esigenze dei clienti. Questa unicità e innovazione introdotta da Ford è relativa proprio alla definizione di processi per la customizzazione e non come si crederebbe generalmente alla semplice produzione seriale di un prodotto uguale per tutti. Il prodotto di Ford è da considerare come la prima piattaforma per la creazione di derivati sulla base del modulo Modello T (Alizon, Shooter, Simpson 2009). Da un certo punto di vista, il Modello T è paradossalmente un riferimento importante per tutti i nuovi paradigmi di progettazione e produzione di artefatti

che offrono l'implementazione e il miglioramento di paradigmi esistenti.

Nella storia del design e dell'architettura troviamo diversi riferimenti al concetto di modularità che sono poi assimilati nelle pratiche della progettazione open source.

L'architetto Le Corbusier è uno dei pionieri in questo senso. Non lo cito per il suo *Le Modulor*, ma per il suo progetto di struttura architettonica modulare Domi-no. Si tratta di un sistema di costruzione in cemento armato che Le Corbusier concepì tra il 1914 e il 1915 che poi è diventato parte fondante della sua architettura poiché basata sull'idea di *open plan*: un sistema di case prefabbricate con uno scheletro indipendente composto da sei colonne portanti e due lastre come pavimenti connesse dalle scale. Scale, pavimenti e colonne sono fissate insieme mentre tutto il resto, come gli interni, non sono predefiniti da una griglia. Il progetto era per Le Corbusier l'applicazione di uno dei suoi principi di architettura moderna, i cosiddetti *open plan*, appunto piani aperti. Vedendo l'architettura come un oggetto industriale assemblabile Le Corbusier parlava di una casa come una macchina per l'abitare:

“Se proviamo a cancellare tutte le nozioni rigide di casa dai nostri cuori e dalle nostre menti e osserviamo il problema da un punto di vista critico e oggettivo, arriveremmo all'idea di “casa-strumento”, la casa realizzata tramite la produzione di massa per tutti, incomparabilmente più sana che vecchia (anche moralmente) e permeata dalla bellezza degli strumenti di lavoro delle nostre vite quotidiane” (Le Corbusier 1986, p. 227).<sup>8</sup>

Il riferimento alle macchine e agli strumenti della produzione industriale non era però il principio fondante della casa Domi-no: l'intenzione era di fornire un sistema di abitazioni customizzabile e a basso costo per gli abitanti dopo la devastazione nel 1914 delle Fiandre.

Il progetto Dom-ino rappresenta il primo caso della storia dell'architettura pensato come un sistema aperto, una piattaforma, in cui sono i residenti a completare e co-creare e in cui l'architetto ha il ruolo di facilitatore (Mcguirk 2014). Come sistema perfetti-

---

<sup>8</sup> Le Corbusier 1986, mia traduzione.

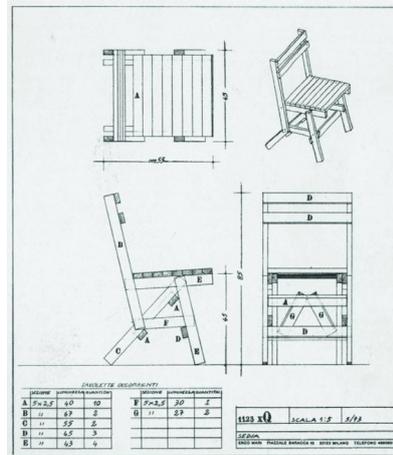
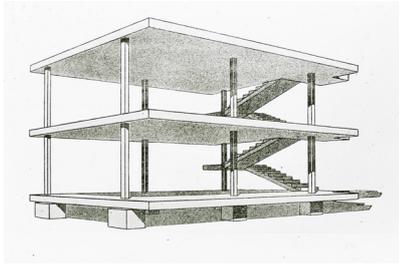


Fig. 33 Il progetto Dom-ino di Le Corbusier

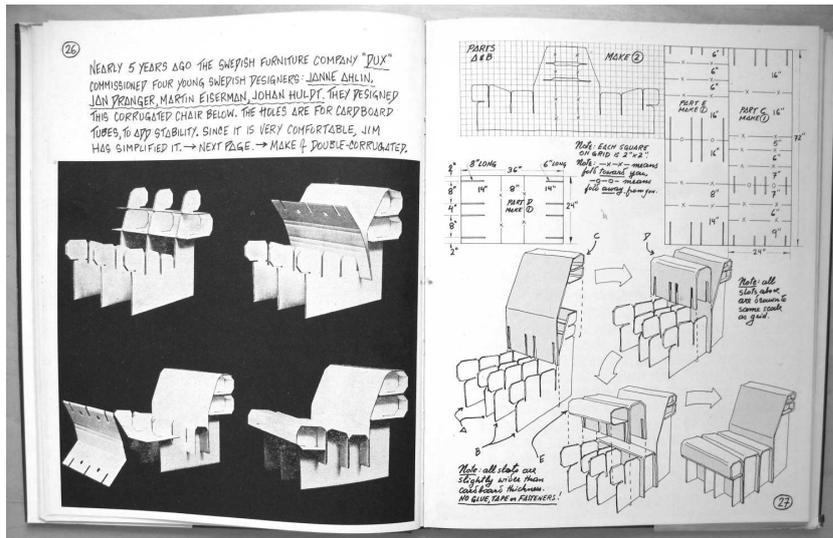
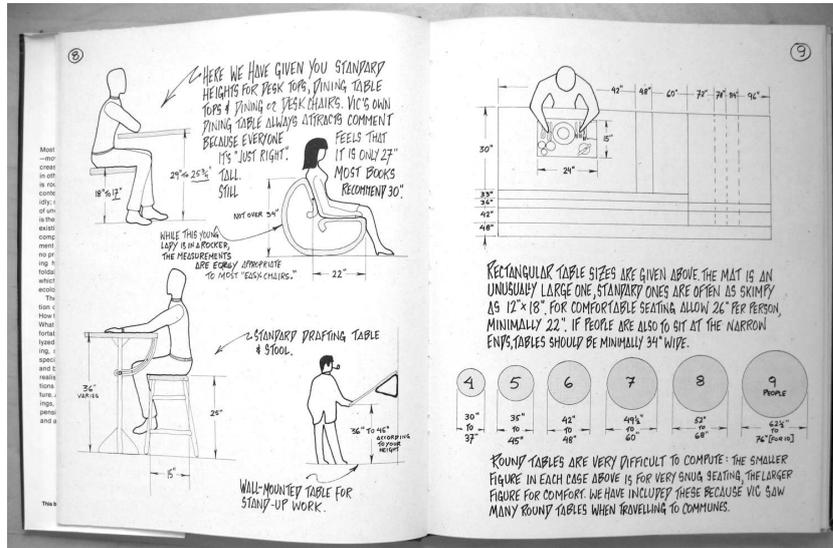
Fig. 34 La sedia di Proposta per un'autoprogettazione di Enzo Mari

bile, il piano aperto e il sistema Domi-no anticipavano gli approcci partecipativi e open source nel design.

“Ritengo che i veri modernisti volessero un design open source già cento anni fa” dice Laarman, “ma allora era impossibile farlo. Rietveld pubblico i manuali su come realizzare le sue sedie, ma nessuno realmente poté usare quelle informazioni perché non esisteva una rete di artigiani competenti. I suoi prodotti di design sembrano semplici, ma sono difficili da costruire. Oggi, possono diffonderne la conoscenza in modo che potenzialmente gli artigiani possano tornare al centro del design – non in modo idealistico, in modo romantico e naïve, ma in un modo economicamente adatto. Tutto quello di cui abbiamo bisogno sono le reti e la tecnologia meno cara e accessibile di produzione digitalizzata” (van Abel et al 2011, p. 121).

Dopo Le Corbusier, un secondo pioniere è Enzo Mari, designer italiano, che negli anni sessanta realizza “Proposta per un'autoprogettazione”, un progetto per la realizzazione di mobili con semplici assemblaggi di tavole grezze e chiodi da parte degli utilizzatori finali. Seppur “Proposta per un'autoprogettazione” è riconosciuto per il suo legame con il fai da te e il design, possiamo considerarlo un esempio interessante di trasposizione del design modulare della produzione industriale a livello del fai da te con l'obiettivo di consegnare all'utilizzatore finale un oggetto funzionale e bello e soprattutto durevole e stabile come qualsiasi altro oggetto realiz-

Fig. 35 Il catalogo di Nomadic Furniture (copertina e interni)



zato a livello industriale. Per esempio la sedia inserita nel catalogo “Proposta per un'autoprogettazione” è, infatti, costruita sulla base di moduli di legno che permettono di assemblarla facilmente, ma anche di modificarla per crearne un derivato<sup>9</sup>. Nell'operazione di Mari c'è l'intento di trasferire la disciplina del design nel mondo del fai da te e non viceversa: all'idea del fai da te come sviluppata in quegli anni negli Stati Uniti, Mari contrappone un esercizio didattico in cui attraverso la creazione dei mobili le persone possano acquisire uno spirito critico, imparando a progettare piuttosto che a ricreare pedissequamente (Mari 2002).

Un terzo riferimento è rappresentato dal lavoro di James Hennessey che, insieme con Viktor Papanek, designer e attivista del design sostenibile, nel 1973, un anno prima della pubblicazione del manuale Autoprogettazione di Enzo Mari, scrive il libro *Nomadic Furniture (How to build and where to buy lightweight furniture that folds, inflates, knocks down, stacks, or is disposable and can be recycled. – with many to easy to follow illustrations)*. Il libro è un catalogo di soluzioni di arredo adatti allo stile di vita nomadico. Indirizzato a persone che non hanno conoscenza della progettazione e del fai da te, il libro offre le indicazioni e le specifiche per costruire mobili per diversi tipi di esigenze: uno sgabello fatto di compensato realizzabile da un foglio 4x8 pollici, una scrivania che è assemblabile senza colla, ganci o viti, cubicoli per il lavoro, il relax i giochi per bambini, scaffali tenuti da sistemi di giunti recuperabili in qualsiasi negozio (Hennessey, Papanek 1973). Tutte le soluzioni state progettate da designer e proposte in questo libro come dei progetti open source le cui specifiche sono rese accessibili per favorire la riproducibilità da parte di persone non esperte. Nel 1973 questo catalogo anticipava il movimento legato all'open source design attraverso una pura operazione di “open-sourcing” di progetti di design esistenti basata sul rilascio delle istruzioni e delle specifiche costruttive nonché sul rilascio della “Bill of Materials” (BOM) e delle indicazioni su come trovare i materiali. Cito questo catalogo di design del 1973 perché esemplifica un aspetto chiave della modularità in un design

---

<sup>9</sup> “Chiunque ad esclusione delle industrie e dei commercianti, potrà utilizzare questi disegni per realizzarli da sé. L'autore spera che questa operazione possa rimanere in divenire; e chiede a quanti costruiranno questi mobili, ed in particolare, le loro varianti, di inviare le foto presso il suo studio, in Piazzale Baracca 10, 20123 Milano” (Mari 2002).

open source, ovvero l'accessibilità e riproducibilità di quelle parti che servono ad assemblare i moduli e componenti di un artefatto e dispositivo. L'accessibilità delle parti di un oggetto di design come, per esempio, i giunti di uno scaffale e l'accesso ad una loro versione standard sono altrettanto importanti come le informazioni sulle misure per poterlo realizzare.

Nel contesto della progettazione di artefatti open source la modularità può essere intesa come la qualità di un sistema costituito da parti tutte uguali, la cui componibilità permettono di riconfigurare l'oggetto o le varianti-derivati di un oggetto, ma come un sistema in cui:

- le parti funzionano da elementi strutturali e funzionali interscambiabili;
- i componenti che permettono di tenere insieme le parti sono accessibili e facilmente riproducibili.

La standardizzazione delle componenti e delle parti permettono che un artefatto open source si differenzi dal risultato di un'attività di design fai da te, che paradossalmente potrebbe proporre degli artefatti poco riproducibili poiché le componenti per l'assemblaggio sono parti non standard o non accessibili sul mercato.

Descrivo di seguito dei casi studio di progetti di open source design che mettono in evidenza l'importanza della standardizzazione delle parti come chiave della modularità nel contesto open source.

Il primo caso è Open E-components (Open E-Components 2014), un sistema di moduli dotati di componenti per la realizzazione di prodotti di elettronica di consumo come gli asciugacapelli, le lampade, i bollitori dell'acqua. Componendo i moduli in maniera diversa è possibile realizzare diversi tipi di prodotto. Il designer Weilun Tseng lavora sulla riduzione del numero di parti necessarie per realizzare un singolo prodotto: l'obiettivo del progetto è ridurre i rifiuti elettronici e proporre un sistema di vendita e riparazione basati sulla sostituzione di un singolo modulo di comuni prodotti elettronici. L'aspetto interessante di questo progetto è la riduzione delle componenti necessarie per creare un prodotto e allo stesso tempo la proposta di standardizzazione finalizzata ad un'ottimizzazione del processo di produzione, vendita e riparazione anche a livello industriale.

Questo concetto di modularità come introduzione di componenti standard la ritroviamo in un secondo caso studio: il progetto OpenStructures.

OpenStructures ([openstructures.net](http://openstructures.net)) nasce come una ricerca sulle opportunità e i limiti della realizzazione di un sistema modulare open source che permetta a una comunità di utenti, dai designer alle imprese e alle persone comuni, di sviluppare dei componenti e delle parti modulari standard e dei prodotti basati su questi. Lo standard è dato da una griglia di rettangoli di dimensione 4 cm x 4 cm i cui bordi sono le linee di taglio, le diagonali i punti di assemblaggio e i cerchi interni i diametri dei punti di connessione tra le parti. Se si sceglie di applicare, per esempio, la griglia per definire la dimensione di una parte, una delle misure deve corrispondere a 0,125cm / 0,25cm / 0,5cm / 1cm / 2cm e ai multipli di 2 cm in modo da renderla compatibile con le altre parti. La modularità in OpenStructures è vista come un mezzo per ottenere dei sistemi organizzati come un sistema organico in cui il modulo della griglia 4x4 è la cellula, le parti sono i tessuti, i componenti sono gli organi, le strutture sono gli apparati e le superstrutture (p.e. le case) sono i corpi. Questa similitudine con il corpo umano sta a indicare come la modularità definisca dei sistemi generativi e delle variazioni tramite la collaborazione da parte delle persone che propongono e condividono le loro parti nella libreria online di OpenStructures. A oggi la libreria di OpenStructures propone circa 190 parti e strutture e oggetti di design originali progettati sulla base della libreria di parti standard. Nonostante lo sviluppo di artefatti sulla base della griglia avvenga e sia possibile, la griglia di OpenStructures non può essere considerato come un sistema standard di progettazione e costruzione: durante la conferenza *Open Design Shared Creativity* del 2014, il designer e realizzatore di OpenStructures, Thomas Lommée, spiega come il sistema della griglia funzioni perché è basato su standard esistenti che ha personalmente studiato per arrivare ad un modulo che fosse compatibile con il resto delle parti disponibili sul mercato; allo stesso tempo è un sistema fragile perché non ha ancora una comunità estesa di persone che intendono adottare questo standard. La partecipazione da parte di una comunità estesa ritiene sia l'unico modo per far sì che da un progetto di ricerca, realizzato presso il centro House for Contemporary Arts Z33, possa essere trasferibile nella produzione da parte di aziende (Lommée 2014).

Fig. 35 Gli elettrodomestici di Open E-components di Weilun Tseng



Fig. 37 Il bollitore di Jesse Howard realizzato sulla base della griglia OpenStructures



I progetti Open E-components e OpenStructures permettono di comprendere cosa voglia dire modularità nel contesto dell'open source design e sebbene i prodotti sui quali si concentrano siano oggetti e parti, i principi che mettono in luce possono essere considerati validi anche nel dominio della progettazione di dispositivi interattivi.

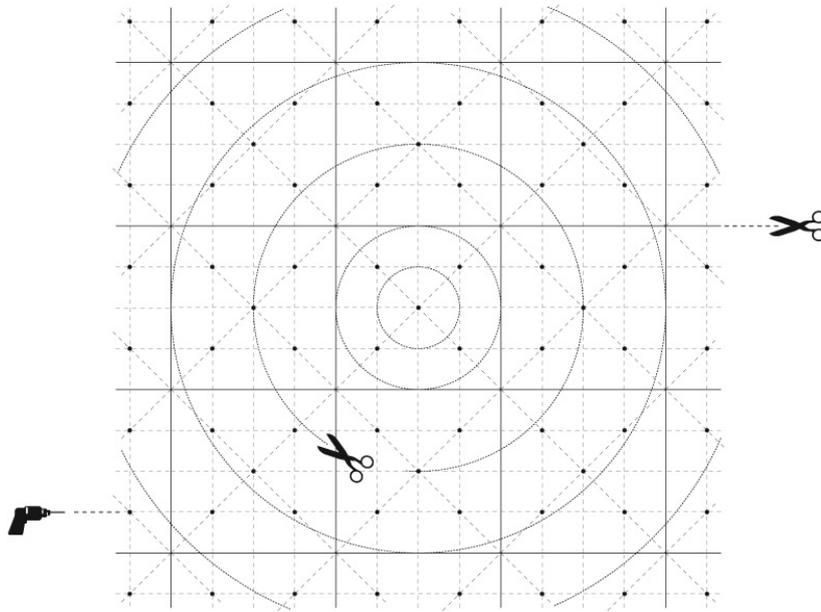


Fig. 36 La griglia di OpenStructures

Nel contesto della progettazione di dispositivi interattivi tecnologici, la modularità, intesa come sistema di connessione tra parti, è riferita agli aspetti di interfacciamento di un dispositivo interattivo con le sue parti o con altri dispositivi, alla sua caratteristica di poter comunicare con essi. Tom Igoe, professore di physical computing presso ITP, descrive la sua posizione rispetto alle pratiche di progettazione di dispositivi open source, definendo i principali aspetti tecnologici da considerare quando si parla di dispositivi aperti:

Una delle proprietà fondamentali dei dispositivi dei giorni nostri è la capacità che essi hanno di comunicare gli uni con gli altri. Se vogliamo, quindi, discutere sul perché si realizzano artefatti open, la prima motivazione risiede nella condizione che questi possano comunicare tra di loro (Igoe 2007) .

La componente tecnologica che permette ai dispositivi di comunicare tra loro è rappresentata dal loro essere dotati Control API (Application Programming Interfaces) aperte, ovvero accessibili da parte degli sviluppatori software.

I lettore di libri digitali di Amazon, il Kindle, e il lettore mp3 della Apple iPod, secondo Igoe, sono dei dispositivi in qualche modo aperti. Il sistema operativo del Kindle, infatti, si basa sul kernel Linux e quindi accessibile e modificabile in maniera gratuita (Turner 2008). Questa apertura dal punto di vista del software permette di poter connettere interfacce tangibili e creare nuovi sistemi di interfacciamento del Kindle. Nel caso dell'Apple iPod, invece, le Control API sono accessibili per gli sviluppatori per facilitare lo sviluppo di accessori compatibili con il lettore mp3 come, per esempio, gli auricolari. Questa "apertura" deriva dalla volontà di Apple di favorire le altre aziende nello sviluppo dei controlli remoti compatibili con l'iPod a soli fini di ampliare il mercato del loro prodotto.

Un secondo elemento tecnologico fondamentale per rendere un dispositivo "aperto" è l'utilizzo di protocolli di comunicazione standard o open source. Un protocollo di comunicazione standard è, per esempio, il Bluetooth. Dispositivi come il controller per la consolle Wii, il Wiimote, comunica facilmente con altri dispositivi perché funziona tramite protocollo Bluetooth. Il Wiimote è un dispositivo utilizzato molto spesso nella realizzazione di installazioni interattive o performance in cui c'è la necessità di rilevare il movimento tramite la lettura dei dati rilevati dai sensori accelerometri al suo interno.

Infine, il terzo elemento tecnologico è rappresentato dall'espandibilità del firmware, il programma di controllo alla base di quasi qualsiasi dispositivo di elettronica di consumo o di dispositivi quali il telefono cellulare e i computer stessi. La possibilità di modificare il firmware delle macchine fotografiche del marchio Canon è un ottimo esempio che spiega, similmente al caso Monotron, come un'azienda possa beneficiare dell'hackerabilità (modifica non autorizzata) del proprio prodotto per connettersi alla comunità di utenti esperti e appassionati della fotografia e sondare nuovi modi di utilizzo di un prodotto per analizzare il mercato e fare ricerca sugli utenti, così come creare dei meccanismi di fidelizzazione.

Open Control API, protocolli standard o open di comunicazione e firmware open source sono i tre elementi tecnologici che possono essere considerati i moduli alla base della progettazione e implementazione di un dispositivo interattivo open source. La

modularità negli artefatti interattivi risiede nella capacità che le parti di un dispositivo parlino tra di loro e che parlino con altri sistemi: i tre elementi tecnologici rappresentano “i giunti” di un sistema interattivo open source così come lo sono i giunti fisici di tavoli, sedie o scaffali open source.

I sistemi di interfacciamento tra dispositivi interattivi fisici permettono non solo che questi comunichino tra di loro, ma che possano essere connessi alla rete e, quindi, implementare risolvere le problematiche relative all'attualizzazione di paradigmi quali *l'internet of things*.

Non approfondisco il paradigma dell'*internet of things* in questo studio, ma lo citiamo per suggerire come gli approcci allo sviluppo e alla progettazione open source siano strettamente connessi all'evoluzione tecnologica degli artefatti interattivi: “l'openizzazione” in questo senso è l'infrastruttura che facilita la possibilità di implementare dei paradigmi di interazione futuri tra gli oggetti, le persone e la rete.

La tabella di seguito mostra le soluzioni di interfacciamento e, quindi, di modularità adottate nel contesto della progettazione sia hardware sia di prodotti interattivi finiti:

Hardware o prodotto	Modello	Dispositivo o parte	Interfacciamento
Arduino	Open source	Shield	Connettori
Little Bits	Parzialmente Open Source	Moduli Little Bits	Magneti
Reactable	Open source	Marker per il suono	Fiducial
Good night lamp	Proprietario	Lampada	Servizio web, Applicazione mobile
Olinda Radio	Proprietario	Moduli Olinda	Connettori a molle di rame e protuberanze che trasportano audio, energia e il codice
Chumby	Parzialmente Open Source	Chumby	Sito web di Chumby, <a href="http://www.chumby.com">www.chumby.com</a>

Tabella 1. Soluzioni di interfacciamento di dispositivi interattivi

Le soluzioni di interfacciamento sono un aspetto di debolezza nel contesto del paradigma open source poiché sono a volte brevettate (p.e. i magneti di Little bits) oppure dipendono da componenti commerciali non open source oppure da servizi di aziende terze. Il modello open source nel contesto degli artefatti fisici risulta essere efficace solo come modello ibrido tra elementi open source ed elementi proprietari.<sup>10</sup>

Se consideriamo l'aspetto legato alle Open API per realizzare sistemi di prodotti e dispositivi connessi tra di loro e la rete, emerge la medesima debolezza: molte aziende forniscono Open Control API del loro dispositivo per facilitarne la messa in connessione alla rete da parte di servizi terzi. Tra i servizi terzi e l'azienda possono esserci ulteriori servizi che interfacciano i diversi sistemi facilitandone la comunicazione: per esempio, Temboo ([temboo.com](http://temboo.com)) è un servizio di nuova generazione che permette di facilitare l'accesso alle API di servizi web come Gmail o Twitter e permette di processare i dati di questi servizi per realizzare applicazioni basate su Arduino. Nei progetti di artefatti interattivi connessi alla rete il sistema di interfacciamento è rappresentato da Temboo che semplifica la programmazione tramite soluzioni di interfaccia grafica e restituisce al realizzatore il codice sorgente per attivare il suo sistema. Piattaforme e servizi digitali sono i connettori degli artefatti interattivi nel contesto dell'internet delle cose. Questi moduli di interfacciamento non sono sviluppati in maniera collaborativa e open source.

La combinazione di hardware, design e software open source e API open potrebbe determinare la diffusione di ecosistemi open source che supportano la creazione di modelli di business sostenibili per la diffusione di tecnologie wearable in cui le comunità di sviluppatori, i cittadini, gli imprenditori e i governi possono collaborare per attivare delle economie di tipo *peer to peer* che non si basano sulla "commodification" delle identità biologiche (Romano, Cangiano 2015, p. 175 ).

Nei sistemi di internet delle cose la modularità è garantita dal siste-

---

<sup>10</sup> Vedi il caso di Littlebits e di MakerBot Replicator 2 che sono prodotti open source che hanno componenti non rilasciati in open source come il sistema di interfacciamento fisici e digitali.

ma di interfacciamento generalmente basato su un artefatto digitale come servizi per l'aggregazione di API. La questione della riproducibilità in questo caso è strettamente legata alla semplicità d'uso e al grado di "apertura" dei servizi digitali che fungono da moduli "giunti" e dalle licenze e i termini d'uso connessi al loro utilizzo.

### 2.5.2 La parametricità

Nel modello di produzione industriale di massa, due metodi permettono ai designer di gestire la variazione di un prodotto: il design basato sui moduli e il design scalabile (Simpson 2004). Nel design modulare i moduli sono aggiunti, sostituiti o rimossi per soddisfare specifici bisogni, come nel modello di produzione fordista; nel design scalabile, invece, sono usate le variabili per modificare i parametri di una o più dimensioni di un prodotto per soddisfare i bisogni di mercato. La definizione di design scalabile come pensata da Simpson è riferita all'introduzione delle soluzioni di Computer Aided Design nei processi di produzione industriale finalizzate alla customizzazione di massa, dove per customizzazione si intende un paradigma di produzione di beni e servizi in cui i bisogni delle persone della progettazione incontrano le necessità dell'ottimizzazione di una produzione industriale (Jiao, Tseng 2004).

La relazione tra il dominio della progettazione e sviluppo open source e il paradigma della customizzazione è duplice: storicamente possiamo associare la diffusione e il successo dell'open source software alla necessità di dover customizzare una soluzione software e adattarla alle specifiche necessità dell'utilizzatore finale, ovvero il modello open source emerge come soluzione al crescente bisogno di avere prodotti software customizzati. L'hardware open source, invece, è uno dei principali fattori che hanno influenzato la diffusione e l'accessibilità delle macchine di produzione digitalizzata come le stampanti 3D, le quali, allo stesso modo del software, permettono di supportare l'implementazione di processi di produzione customizzati rispetto a una specifica esigenza (tipologia di prodotto, sostenibilità o innovazione, dimensione della produzione e dell'azienda) o di processi di produzione di prodotti customizzabili. L'hardware

open source è da considerare per questo motivo come una delle infrastrutture tecnologiche chiave per i modelli di produzione distribuiti e digitalizzati.

La duplicità della relazione è quindi la seguente: l'open source è un sottoprodotto del paradigma della customizzazione di massa, la customizzazione di massa è un sottoprodotto dell'open source. Nella relazione inversa va però messo in evidenza come la customizzazione nel contesto open source assuma delle connotazioni nuove e diverse rispetto alla possibilità che il bisogno degli utenti incontri l'ottimizzazione del processo industriale. Come Project Ara esemplifica, il modello di creazione di nuove tipologie di artefatti tecnologici e le nuove modalità di progettazione abilitate dall'open source sono proprie di un paradigma di *mass market co-creation* piuttosto che di *mass market customization*.

Nel dominio open source il bisogno di avere un prodotto adattato al proprio bisogno è soddisfatto tramite la creazione e co-progettazione della variante customizzata del prodotto stesso, in maniera individuale o collaborativa. Alla possibilità che l'utente o la comunità di utenti possano creare una variazione del prodotto devono corrispondere tecnologie e processi adatti alla messa in produzione di tali varianti potenzialmente infinite quante sono le modifiche legate ai singoli bisogni delle persone. In questo paragrafo vogliamo mettere in evidenza come le modalità di co-creazione nel contesto open source sia legato alla progettazione di strumenti semplificati di progettazione che riducono la complessità di un file originale sorgente di un prodotto fisico mantenendo accessibile e aperto il concetto di design e il principio costruttivo che ne è alla base. Questi strumenti semplificati di progettazione li definisco "ambienti di co-progettazione parametrica", ovvero artefatti interattivi che operano da interfacce tra l'utente co-designer e il prodotto e la sua messa in produzione. L'idea di co-progettazione parametrica si riferisce alla caratteristica che questo tipo di artefatti hanno nel creare un modello bidimensionale e tridimensionale a partire dalla manipolazione delle relazioni tra gli elementi di un modello digitale tramite parametri e regole.

Gli ambienti di co-progettazione parametrica permettono all'utente di operare su diversi livelli:

— **sugli attributi dei moduli che compongono un prodotto.**

Questo livello accomuna gli ambienti di co-progettazione parametrica ai servizi digitali utilizzati per implementare sistemi di customizzazione di massa di un prodotto. Un attributo di un modulo di un prodotto è per esempio il colore della tomaia di una scarpa. NikeID (Nike 2014) è il progetto dell'azienda Nike che offre alle persone la possibilità di selezionare e combinare i colori del modello di scarpa preferito attraverso un'applicazione web. L'applicazione è dotata di pannelli per la selezione delle parti dei colori e delle dimensioni della scarpa. Come NikeID, esistono molti casi di aziende che hanno sviluppato strumenti per la customizzazione del prodotto che sono parte di campagne integrate di comunicazione. In queste campagne di comunicazione l'applicazione per la selezione e la modifica di un attributo del prodotto è considerato, un *touchpoint*, ovvero un punto di contatto tra un cliente e il brand dell'azienda, piuttosto che una interfaccia tra l'utente co-designer e il *design space* del prodotto. Questo tipo di applicazioni sono ambienti in cui la modifica del prodotto si basa su una simulazione foto-realistica e non su un'interazione diretta con il file originale sorgente che sarà inviato alle macchine di produzione digitalizzata. Al contrario, in un ambiente di co-progettazione parametrico l'applicazione, sia essa web sia essa desktop, genera un file originale sorgente che è adatto alla messa in produzione tramite tecnologie di produzione digitalizzata.

— **sulla modifica dei parametri di base come per esempio la dimensione del prodotto.** Questo tipo di funzionalità è proprio delle applicazioni web utilizzate come *touchpoint* di un prodotto e di un marchio. Utilizziamo ancora il caso di NikeID: l'applicazione offre la possibilità di selezionare la dimensione della scarpa in accordo con il genere e il tipo di calzata.

Il *fitting*, la calzata di un capo è, infatti, stata adeguata tramite degli standard di misure. L'alternativa al sistema delle calzate standard dell'industria è il prodotto fatto su misura a livello sartoriale, quindi, artigianale. In seguito alla crescita della distribuzione di capi moda tramite la rete, il problema del *fitting*

è diventato prioritario rispetto alla possibilità di implementare soluzioni di customizzazione on-line degli attributi estetici del prodotto. Con il [fits.me](#), un'interfaccia fisica controllata da un'applicazione web parametrica, è possibile verificare in tempo reale la calzatura di un capo moda tramite un sistema ibrido che combina la simulazione alla modifica reale del capo ([fits.me](#)). L'applicazione fornisce, infatti, un'interfaccia utente che permette di selezionare le taglie dei capi tramite l'indicazione delle proprie misure e di verificare in tempo reale la calzatura del capo. Questa funzionalità è resa possibile da un manichino robotico connesso all'interfaccia utente le cui dimensioni cambiano al variare dei parametri forniti dall'utente.

Cito questo caso per spiegare come la parametricità di un prodotto non sia strettamente connessa alla possibilità per l'utente finale di customizzarlo e soprattutto come non sia più utile intendere la customizzazione come unicità dell'aspetto estetico di un prodotto, ma piuttosto come la risoluzione a un tipo di esperienza di acquisto in rete.

— **sulla progettazione di forme computazionali che permettono di generare infinite versioni di un prodotto**

Nervous system ([n-e-r-v-o-u-s.com](#)) è uno studio di design computazionale che progetta e distribuisce accessori applicando degli approcci alla creazione delle forme basate su algoritmi e tecnologie di manifattura digitalizzata. Il metodo dello studio si basa sulla creazione di applicazioni web che permettono alle persone di progettare un accessorio nuovo e unico sulla base della regola algoritmica che genera un modello bidimensionale e tridimensionale che può essere prodotto tramite tecnologie di stampa additiva. Le applicazioni web di Nervous system sono ambienti di co-progettazione parametrica in cui il design consiste nel codice informatico che definisce la regola di costruzione del modello tridimensionale il quale rappresenta il parametro da manipolare per definire una variazione dell'oggetto. Le diverse variabili per generare le varianti dell'oggetto sono rappresentate da icone che facilitano la modifica di un file sorgente che viene inviato ai servizi di stampa 3D. I *Tool* di Nervous system (Nervous system 2014) offrono infatti un'interfaccia grafica non standard che semplifica il processo di progettazione dell'artefat-

to generativo. La generatività di questo tipo di ambiente e del relativo prodotto deriva dai metodi e dagli approcci stessi del design computazionale che, combinati con gli approcci e i metodi dell'open source, potrebbero essere considerati le strade progettuali più efficaci per assicurare le riproducibilità di un artefatto interattivo. Il designer nel contesto dell'open design è considerato come una figura che definisce processi piuttosto che prodotti e che manipola informazioni piuttosto che materia. Come un *metadesigner* (De Mul 2011), il designer ha la possibilità definire è una nuova possibilità e modalità di interazione con i prodotti della nostra vita quotidiana. Penso, ad esempio, a Spruzzo, un prototipo sviluppato al FabLab di Torino /Officine Arduino che usa le stesse tecniche del progetto Nervous system: Spruzzo è un germogliatore open source le cui dimensioni possono essere customizzate per l'installazione in diversi ambienti. Uno dei parametri della customizzazione è il tipo di seme che si intende ottenere e il sapore del seme. L'applicazione web del progetto genera il file bidimensionale che serve a costruire il germogliatore tramite l'utilizzo di una macchina a taglio laser e a generare il codice necessario per avviare la scheda hardware Arduino che controlla la quantità di acqua da dare ai semi in base anche alla temperatura dell'ambiente circostante (Loglio 2014).

La progettazione di sistemi modulari e parametrici, e in particolare delle loro interfacce, rappresenta, quindi, l'area di azione principale del design dell'interazione indirizzato a supportare la riproducibilità di artefatti open source. L'intervento del designer riguarda la definizione di parametri di contenuto piuttosto che di forma e dimensione.

Per questo motivo, emergono negli ultimi anni nuovi servizi digitali che permettono di supportare la progettazione parametrica di artefatti di design e la gestione della produzione di tali artefatti tramite tecnologie di digital fabrication o la creazione degli ambienti di co-progettazione o customizzazione da parte degli utenti finali ([mattermachine.com](http://mattermachine.com)).

A questi servizi si aggiungono le librerie software open source che supportano i designer nella programmazione di design parametrici che possono essere prodotti tramite tecnologie

Fig. 38 Interfaccia del servizio digitale NikeiD

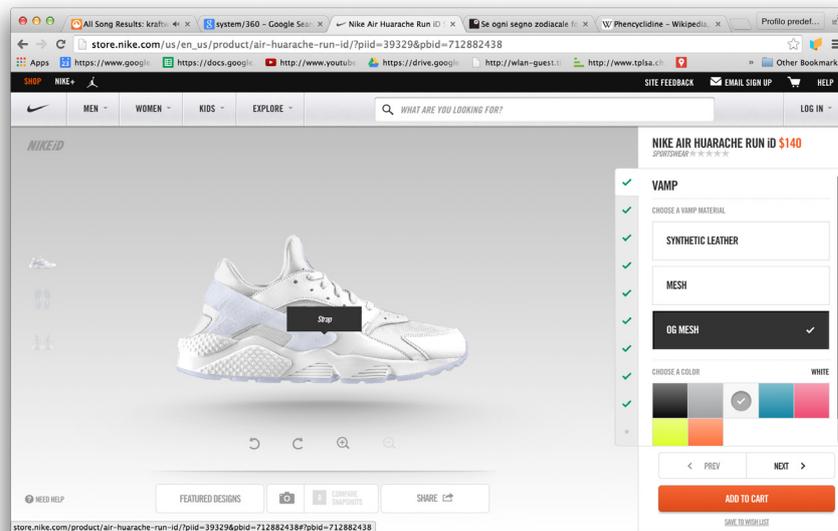
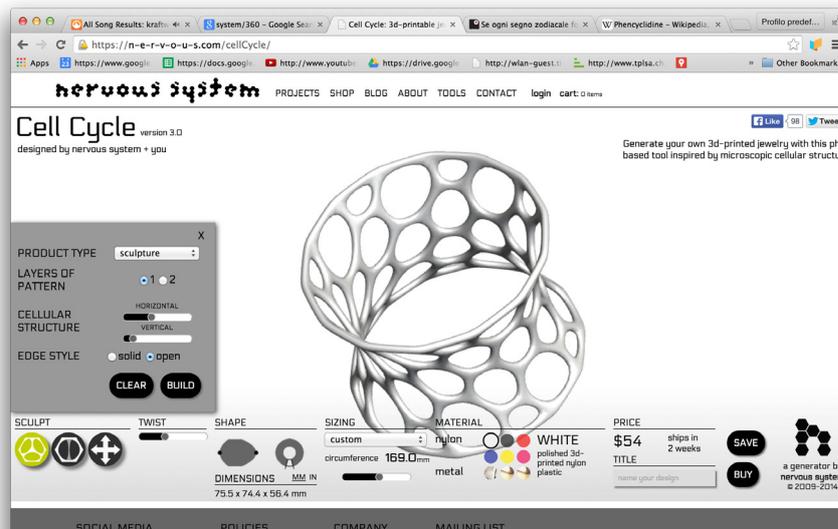


Fig. 39 Cell, l'ambiente di progettazione di accessori generativi e stampabili in 3D dello studio Nervous System



di stampa additiva ([toxiclibs.org](http://toxiclibs.org)). Fornire un'interfaccia che permette di svelare e manipolare il concetto di design e principi costruttivi di un artefatto rappresenta, infatti, una delle chiavi per rendere aperto un design poiché si rende aperto il processo e la regola attraverso il quale esso è stato generato.

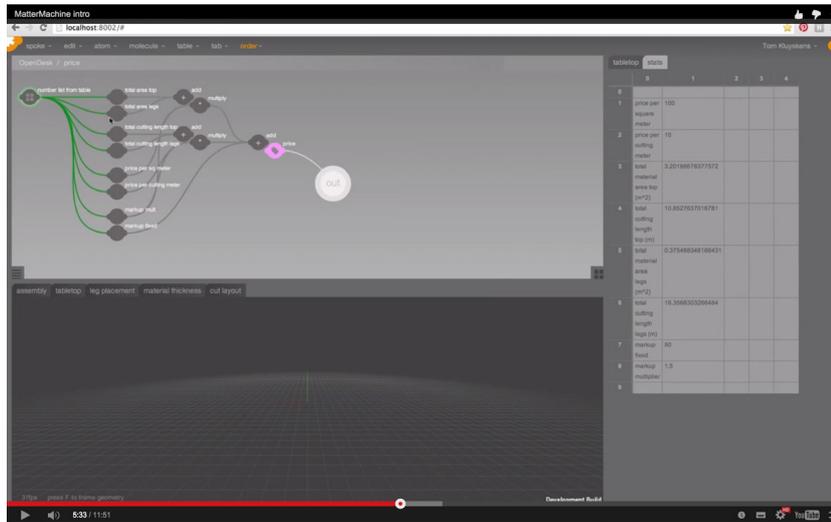


Fig. 40 Interfaccia del software Matter Machine

## 2.6 Le macchine e le interfacce di riproduzione

Il progetto Arduino, descritto nel Capitolo 1, è un caso emblematico di come le tecnologie abilitano i designer a sviluppare i propri strumenti per affrontare la progettazione di sistemi interattivi tramite la prototipazione hardware e elettronica.

L'abilità dei designer nel costruire strumenti di progettazione e produzione non è un novità nella storia delle discipline del progetto. Nell'ambito della progettazione e sviluppo open source, questa abilità è rafforzata e apre le porte di una nuova progettualità indirizzata alla realizzazione di interfacce, dispositivi e macchine per la produzione e riproduzione di artefatti fisici. Iniziamo dalle macchine di riproduzione.

RepRap è uno dei progetti che maggiormente ha contribuito allo sviluppo di stampanti tridimensionali open source, fai da te e auto riproducibili. "L'autoriproducibilità" delle stampanti RepRap consiste nella sue parti stampabili tridimensionalmente secondo un obiettivo progettuale di creare delle macchine che possono generare altre macchine. Questo principio alla base del progetto RepRap dimostra come la modularità nel paradigma open source rappresenta uno dei fondamenti principali della sua attualizzazione.

Il principio alla base dell'iniziativa RepRap, come spiega Adrian Bowyer, è stata l'investigazione delle strategie evolutive

naturali di riproduzione in cui le specie che hanno una relazione d'interdipendenza con altre specie hanno maggiore successo di sopravvivenza (Jones et al 2011). Secondo questo principio, le macchine cinetiche che si autoriproducono in maniera totalmente indipendente non possono essere implementate con successo e non supportano un modello di evoluzione sostenibile della relazione tra uomo e le macchine che sia sostenibile.

L'iniziativa RepRap propone quindi un modello di progettazione delle macchine di produzione dei nostri beni che si basa sulla condivisione totale delle istruzioni per crearle, una comunità estesa di utilizzatori e la possibilità di auto-generare le parti necessarie piuttosto che una lista di componenti commerciali prodotte a livello industriale ([reprap.org](http://reprap.org)).

A partire da RepRap e dallo scadere dei brevetti della stampa tridimensionale inizia l'era della creazione di derivati di macchine di produzione a basso costo, fai da te e open source in cui i designer accedono alle conoscenze per estendere il proprio dominio di azione alla realizzazione di strumenti di produzione. In 'Collective works' Mischa e Traxler, due designer austriaci, riflettono sui processi industriali e sul concetto di collaborazione realizzando un artefatto interattivo che permette di creare cestini di vimini. Il sistema reagisce alla presenza delle persone e attiva un meccanismo che avvolge una fascia di vimini. La fascia di vimini crea un cestino e un pennarello lo decora a intermittenza sulla base della presenza delle persone la cui presenza è monitorata dai sensori incorporati nella macchina. Il progetto punta a realizzare oggetti unici e realizzati tramite una macchina che supporta un processo di produzione collaborativo e digitalizzato (Mischa, Traxler 2014). Da questo progetto si evince che grazie all'accesso alle tecnologie e alla loro semplificazione i designer si appropriano di competenze che aprono la progettazione al processo e alle tecnologie di produzione che da sempre hanno costituito il limite della progettazione stessa. In particolare, quando il processo e le tecnologie di produzione si apre al design stimola principalmente la definizione delle modalità con le quali le persone interagiscono con la produzione di beni. Nel progetto di Mischa e Traxler, infatti, è la definizione della modalità di interazione che abilita la creazione di un nuovo processo di manifattura, ovvero l'inserimento di sensori che

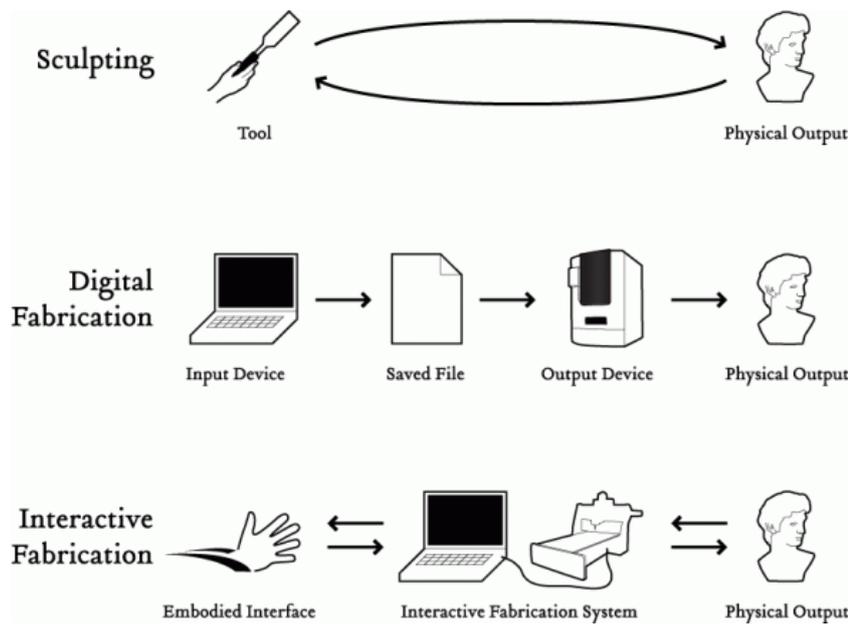


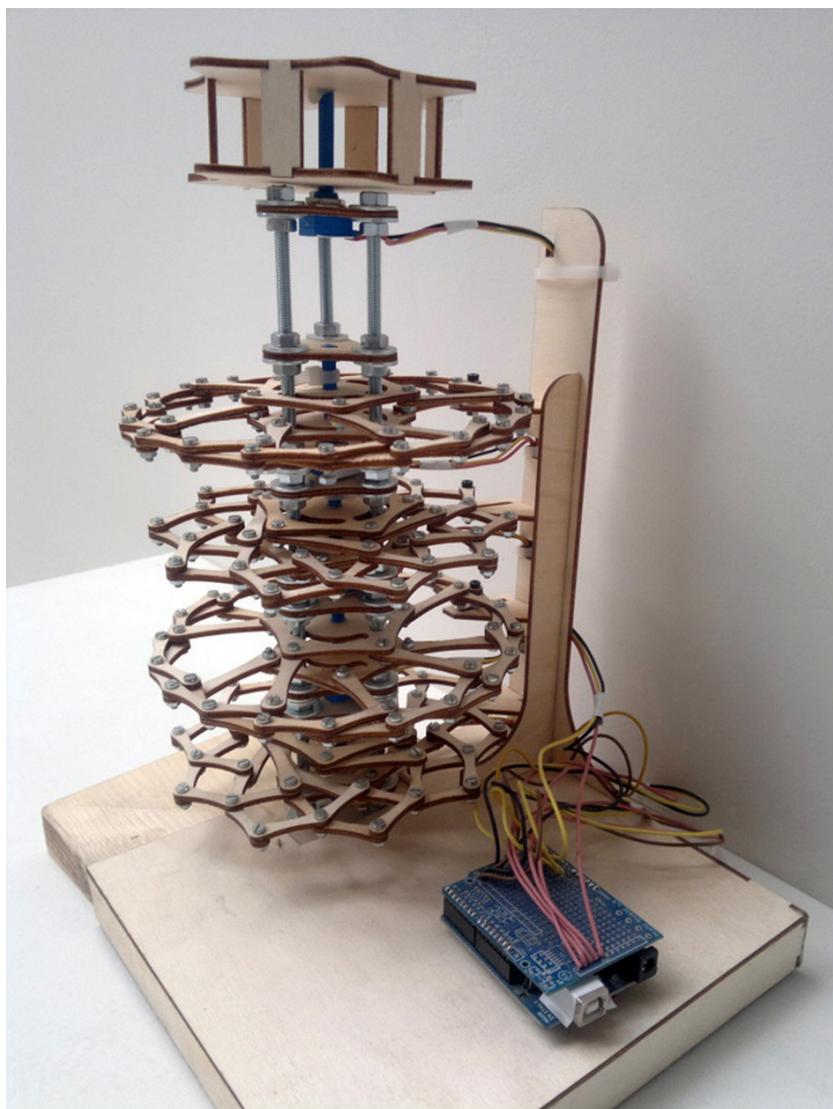
Fig. 41 Diagramma della Interactive Fabrication

rilevano la presenza delle persone. Le macchine diventano dei dispositivi di interfacciamento tra le persone e la produzione. Le interfacce sono l'elemento che sposta il focus sulla digitalizzazione della produzione sui sistemi che abilitano le persone a realizzare artefatti di vita quotidiana.

Dalla digital fabrication si passa alla "interactive fabrication" ovvero il dominio della progettazione di interfacce e dispositivi per la digital fabrication (Willis et al 2011).

Alcuni esempi sono progetti di interactive fabrication hanno una natura programmatica, ovvero puntano a dimostrare le potenzialità sottostanti la creazione di interfacce sperimentali per la produzione di oggetti. L'Artisan Electronique, per esempio, è un'installazione di studio Unfold che consiste in un'interfaccia gestuale che permette alle persone di modellare un vaso di ceramica parametrico e realizzare delle versioni customizzate di questo tramite l'utilizzo di una stampante tridimensionale ([unfold.be](http://unfold.be)). Il Tornio digitale realizzato da Enrico Bassi e Lorenzo Romagnoli è, invece, un'interfaccia fisica che permette di inviare dati a un software di modellazione e a una stampante tridimensionale a basso costo. Manipolando gli anelli del tornio si riesce a definire la forma del modello tridimensionale al fine di produrre oggetti unici e

Fig. 43 Tornio digitale di Enrico Bassi e Lorenzo Romagnoli (FabLab Italia)



customizzati (Romagnoli 2014).

Analizzando questi progetti ritorna il tema della semplificazione della tecnologia che a livello programmatico porta i designer ad offrire nuove modalità per gli utenti-produttori di oggetti, ma che permette ai designer stessi di immaginare dei nuovi processi produttivi e sperimentare nuove estetiche superando i limiti imposti dalla produzione industriale di massa.



Fig. 42 Collective works di Mischer Traxler



Fig. 44 L'artisan eletronique di studio Unfold



### 3. Due casi studio di artefatti interattivi open source: Open Mirror e Primo

In questo capitolo presento due casi studio di artefatti interattivi open source: Open Mirror di Studio Habits e Primo di SolidLabs. I due prodotti sono stati selezionati poiché sono prodotti interattivi complessi dotati di componenti hardware, di software e interfaccia tangibile e sono stati concepiti dal punto di vista progettuale in relazione al contesto dello sviluppo di open source hardware, open design e delle tecnologie di digital fabrication. Inoltre, si tratta di progetti che hanno superato una fase di sperimentazione e prototipazione e, attraverso differenti modelli di business, hanno raggiunto il mercato degli utilizzatori finali e sono entrambi dei prodotti certificati (certificazione CE).

Entrambi i progetti sono stati rilasciati con una licenza Creative Commons (CC-BY-NC-SA). Nonostante l'uso di una licenza non conforme alla definizione di Open Source di Perens (Perens 1999), i due progetti sono considerati in questo studio come casi studio utili a descrivere le opportunità e i processi sottostanti la progettazione di artefatti interattivi nel contesto delle pratiche open source.

L'analisi dei casi studio si basa sulla valutazione delle risorse e della documentazione pubblicata sui siti ufficiali dei due progetti e da interviste che ho condotto personalmente ai designer e realizzatori dei due prodotti in appendice a questa tesi.

#### 3.1 Open Mirror di Studio Habits

Innocenzo Rifino e Diego Rossi sono i fondatori dello studio di design Habits, basato a Milano. Dal 2011 sviluppano un progetto di autoproduzione chiamato Digital Habits, una linea di

oggetti interattivi che esplorano la possibilità di combinare le interfacce naturali basate sui gesti e l'esperienza di ascolto della musica. Il primo prodotto della serie Digital Habits è l'Open Mirror, uno specchio ovale che funziona da altoparlante per iPhone e che integra sensori per il controllo delle tracce musicali tramite i gesti.

### Modalità di interazione e user experience design

Lo specchio ha una forma ovale ed è costituito da due parti connesse da un carello metallico. Scorrendo la parte inferiore da sinistra verso destra, l'utente ha accesso ad un connettore per iPhone. Posizionando l'iPhone sul connettore, l'utente può attivare i controlli dell'applicazione per gestire le *playlist* musicali all'interno del dispositivo mobile. L'interfaccia è di tipo naturale: attraverso i gesti, e senza toccare lo specchio, è possibile attivare o disattivare la riproduzione di una traccia musicale, aumentare o diminuire il volume.

### Tecnologia

Lo specchio è dotato di diversi componenti hardware che permettono di gestire il controllo delle funzionalità di riproduzione delle tracce musicali tramite i gesti. I componenti hardware principali sono i tre sensori infrarossi che sono utilizzati per interpretare i gesti, il circuito stampato il cui design è basato sulla scheda Arduino (AT Mega), due altoparlanti e le luci LED.

### Design del prodotto

Lo specchio è costituito da due parti: la parte superiore che contiene il circuito stampato e gli altoparlanti, la parte inferiore che contiene i sensori e le luci LED.

Le parti fisiche del prodotto sono realizzate tramite la sovrapposizione di strati di legno tagliati con una macchina a taglio laser. Al fine di migliorare le prestazioni del suono, sulla parte inferiore è posto uno strato di neoprene.

Due componenti fisici sono prodotti a livello industriale: lo specchio che è agganciato alle parti di legno tramite dei magneti, il carello di metallo che serve a supportare lo scorrimento per l'accesso al connettore dell'alimentazione dell'iPhone.

Una parte stampata in 3D con materiale plastico ABS è applicata sulla parte frontale dello specchio per comunicare il marchio Digital Habits.

### Contesto di ricerca e sviluppo

Il livelli e temi progettuali affrontati nel progetto Open Mirror riguardano la possibilità di combinare un prodotto classico, lo specchio, con la dimensione dell'ascolto della musica tramite un dispositivo di comunicazione mobile, lo smart phone iPhone della Apple. L'aspetto dell'interazione è anche un tema principale poiché il prodotto offre la possibilità di controllare un'applicazione per l'ascolto della musica tramite l'interazione naturale dei gesti e non con il tocco diretto dello schermo del dispositivo. La produzione tramite tecnologie di prototipazione rapida permette di realizzare un prodotto che può essere percepito come un prodotto rifinito a livello industriale, ma prodotto a livello artigianale e con l'utilizzo di macchine di manifattura digitalizzata.

### Modello di business

Il progetto è autoprodotta e assemblata presso lo Studio Habits. Si tratta di un progetto di autoproduzione che combina parti realizzate presso lo studio e componenti e parti acquistate presso aziende terze. Lo specchio è venduto tramite un sito internet ed è prodotto su richiesta. La produzione è limitata: il numero massimo di specchi prodotti al giorno è circa dieci. Il progetto non è supportato da capitali esterni, ma è finanziato dai realizzatori tramite la loro attività di consulenza di design legata a Studio Habits.

#### 3.1.1 Modello di rilascio open source

I file, la documentazione e le istruzioni di assemblaggio sono scaricabili dal sito del progetto ([digitalhabits.it](http://digitalhabits.it)). La licenza utilizzata è di tipo Creative Commons, attribuzione non commerciale condivisi allo stesso modo. Non sono utilizzate piattaforme on-line per monitorare lo sviluppo collaborativo da parte di altri come, per esempio, GitHub o Instructable. Il modello open source è stato utilizzato per comunicare le competenze dei realizzatori verso l'esterno e favorire la circolazione del progetto a livello mediatico.

Non sono ancora stati sviluppati dei prodotti derivati da parte di altri e non è stato sviluppato un modello per favorire questo processo partecipativo. Lo sviluppo del progetto è curato internamente allo Studio Habits che rilascia le versioni aggiornate del prodotto (p.e. dalla connessione via cavo USB all'utilizzo del protocollo di comunicazione Bluetooth per connettere i sensori di distanza ai controlli dell'iPhone).

### 3.1.2 Processo di progettazione e sviluppo

#### Dalla consulenza all'autoproduzione per creare innovazione

Il progetto è nato come risposta all'impossibilità per lo Studio Habits di ricevere una committenza da parte di un'azienda di design per la realizzazione di un prodotto tecnologico complesso. La prima innovazione è legata alla possibilità per i due designer, Diego Rossi e Innocenzo Rifino, di combinare le competenze di progettazione di prodotti di design alle competenze di progettazione di prodotti interattivi per realizzare dei prodotti in modo indipendente sfruttando le tecnologie di manifattura digitale. Sebbene i benefici dei modelli di business basati sull'autoproduzione non siano nuovi, l'Open Mirror non nasce dalla volontà dei designer di avviare un'attività imprenditoriale basata sulla produzione limitata, ma dalla reazione all'impossibilità di accedere a competenze e capitali di aziende che operano nel settore del design. La mancanza di competenze, determina una limitazione dell'esplorazione di nuove opportunità di mercato basate sulla distribuzione di prodotti innovativi che integrano tecnologie e design.

#### Acquisizione di competenze tecnologiche tramite la prototipazione rapida

La possibilità di accedere a risorse esistenti in internet e rilasciate con licenze open source, in particolare le risorse riguardo all'implementazione di soluzioni interattive per il controllo del suono tramite sensori infrarossi, ha permesso ai designer di sviluppare e fare iterazioni molteplici in breve tempo e di acquisire velocemente competenze per la gestione degli aspetti legati all'elettronica e alla programmazione. L'accesso gratuito alle risorse e la facilità di

utilizzo delle piattaforme di prototipazione (p.e. Arduino) e delle macchine di manifattura digitalizzata hanno permesso di sviluppare competenze internamente allo studio di design. Il modello classico avrebbe previsto la consulenza a pagamento finalizzata all'ingegnerizzazione delle soluzioni tecnologiche e interattive che avrebbe comportato l'aumento dei costi del progetto in maniera esponenziale in base alle molteplici iterazioni.

#### Un processo ciclico: design – hardware – design

L'esperienza e le competenze acquisite durante lo sviluppo del progetto Open Mirror hanno permesso ai designer di sviluppare un derivato della scheda Arduino, una scheda hardware che permette di gestire i sensori e gli attuatori per il controllo del suono o il protocollo di comunicazione *bluetooth*. Queste funzionalità, che di norma non sono supportate da una scheda Arduino di base, sono integrabili tramite l'utilizzo di *shield*, schede hardware impilabili su Arduino, o attraverso circuiti e componenti di maggiore complessità. A partire dalle iterazioni sull'hardware di Open Mirror, sono stati definiti i requisiti di una scheda che può supportare la realizzazione di prodotti in cui si combinano le interfacce gestuali, le luci, i suoni e la comunicazione tra dispositivi mobili.

Questo tipo di soluzione ha permesso ai designer di avviare la progettazione di una linea di prodotti la cui esperienza utente si basa sulla combinazione di questi elementi.

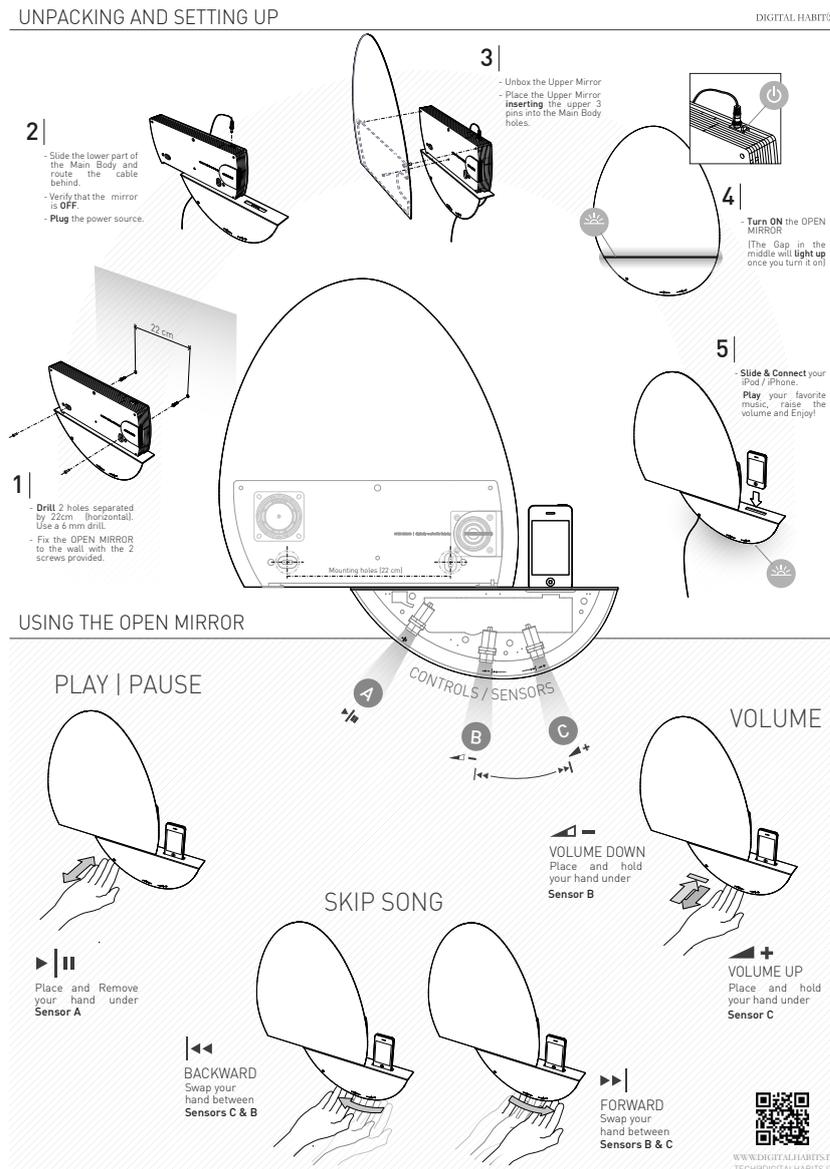
Un esempio è PACO, l'altoparlante che riproduce la musica di uno smart-phone via bluetooth e che permette alle persone di poter gestire il volume e il controllo delle tracce tramite i gesti. Lo speaker è realizzato in cemento e la struttura interna è caratterizzata da parti stampate in 3D. Le parti realizzate con macchine di prototipazione rapida sono nascoste e non visibili dall'utente finale.

La scheda hardware all'interno di PACO è dotata di una porta USB che tramite la connessione a un computer permette di riprogrammare le modalità di interazioni trasferendo sulla scheda un nuovo firmware. L'idea di "hackerabilità programmata" è stata successivamente aggiornata nel progetto Cromatica, un prodotto ibrido che combina le funzioni di un altoparlante a quelle di una lampada le cui luci sono controllabili tramite un'applicazione mo-

bile. Cromatica è distribuita come kit per programmatori o maker che possono acquistare i singoli componenti e la scheda per creare derivati del prodotto riprogrammando le sue funzionalità e le modalità di interazione.

L'evoluzione del progetto Open Mirror e la creazione, a partire da questa sperimentazione, di una linea di prodotti ibridi e riprogrammabili da parte degli utenti finali introduce un processo

Fig. 1 Open Mirror. Manuale di assemblaggio con le istruzioni per l'uso.



di progettazione innovativo in cui la definizione delle specifiche di un prodotto interattivo portano alla definizione di requisiti tecnologici. A partire dai requisiti tecnologici è stata realizzata una scheda hardware che è espandibile e riprogrammabile e dalla quale sono state generate nuove tipologie di prodotti. Questi prodotti sono riprogrammabili e questo apre il processo di creazione di derivati ad altre persone, dagli sviluppatori, ai maker, agli utilizzatori finali che necessitano di customizzazioni dei prodotti.

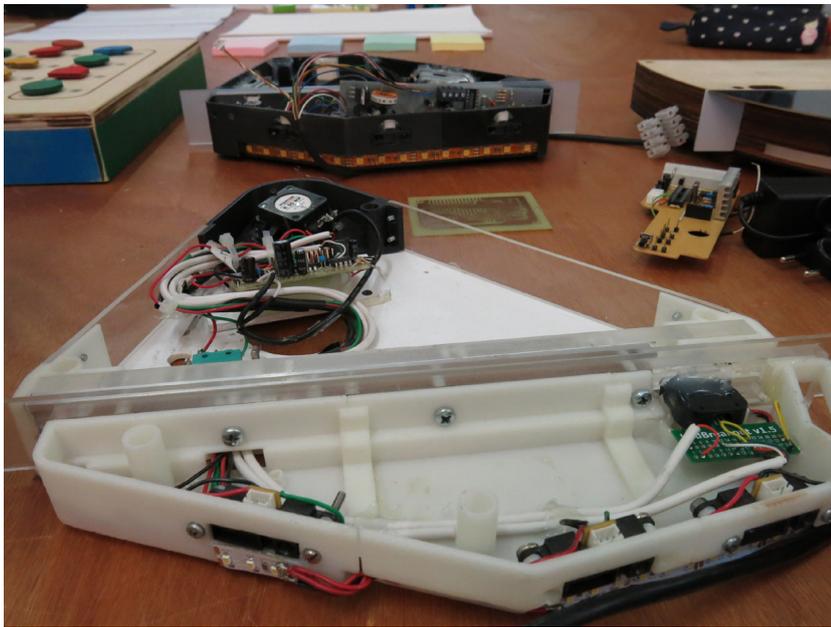


Fig. 2 Open Mirror. Uno dei prototipi realizzati con parti stampante in 3D.



Fig. 3 Open Mirror. Il prodotto.

Fig. 4 OSOUND.  
Altoparlante bluetooth  
della linea Digital Habits



Fig. 5 P.A.C.O.  
Altoparlante bluetooth  
della linea Digital Habits





Fig. 6 Cromatica. Altoparlante e lampada della linea Digital Habits. Attraverso un'applicazione per smart-phone è possibile cambiare il colore della luce.

### 3.2 Primo di Solid Labs

Primo è un'interfaccia tangibile progettata per spiegare la logica della programmazione a bambini dai 3 ai 7 anni, ovvero nell'età precedente all'alfabetismo. Il progetto è il risultato di un corso di interaction design ([maind.supsi.ch](http://maind.supsi.ch))<sup>1</sup> tenuto alla SUPSI - Scuola universitaria professionale della Svizzera italiana nel 2012. Autore del progetto è Matteo Loglio. In seguito al finanziamento ricevuto tramite una campagna di *crowdfunding* supportata dalla piattaforma Kickstarter (Kickstarter 2012), il progetto è stato sviluppato ulteriormente al fine di diventare un prodotto realizzato a livello industriale.

#### Modalità di interazione e user experience design

Il prodotto consiste in un'interfaccia tangibile (Board) attraverso la quale i bambini possono incastrare dei blocchi di legno che funzionano da istruzioni per comandare a distanza un robot a forma di cubo (Cubetto). L'obiettivo del gioco è fare in modo che Cubetto torni a casa. Per completare il gioco, i bambini devono programmare il piccolo robot incastrando nella Board una serie di blocchi con le istruzioni: avanti, sinistra, destra e funzione. Le prime tre istruzioni sono abbastanza intuitive, mentre l'ultima - la funzione - va a richiamare una "sub-routine", ovvero quello che in programmazione è un insieme di istruzioni compresse in un unico comando.

#### Tecnologia

La tecnologia alla base del robot Cubetto si basa su un progetto di scheda hardware e open source: la scheda *Oh\_Oh board* realizzata da David Cuartielles, uno dei fondatori di Arduino. Il robot è essenzialmente un Arduino a forma di cubo di legno con un Xbee, un modulo hardware per gestire la comunicazione radio (David Cuartielles 2010). Il sistema di riconoscimento dei blocchi con le istruzioni per controllare il movimento di Cubetto si basa su un progetto sviluppato presso il CIID (Copenhagen Institute of Interaction Design) chiamato "Barcode Piano" (CIID 2011). I blocchi sono riconosciuti dall'interfaccia tangibile attraverso delle resistenze che funzionano da *voltage divider* connessi alle porte di

input analogiche di Arduino. Arduino legge i valori delle resistenze e li trasforma in comandi di posizione del Cubetto.

### Design del prodotto

Primo è progettato per essere distribuito in due versioni. La versione industriale che prevede l'utilizzo di materiali plastici per la realizzazione dell'interfaccia tangibile e la versione "prototipo" che può essere costruita in modo fai da te seguendo le istruzioni e la documentazione pubblicata nel manuale on-line ([docs.primo.io](http://docs.primo.io)). Il manuale on-line include la lista di tutti i componenti e la guida "passo-passo" per realizzarlo tramite macchine a controllo numerico. La versione "prototipo" è realizzata in legno, il materiale che è stato scelto per rendere il prodotto più simile ai giochi classici, non tecnologici.

### Contesto di ricerca e sviluppo

Il progetto si inserisce nella ricerca sull'insegnamento della programmazione ai bambini che ha riferimenti nel lavoro di Seymour Papert, ricercatore e scienziato del MIT inventore del linguaggio di programmazione Logo ([papert.org](http://papert.org)). La ricerca alla base dello sviluppo di Primo è la progettazione di soluzioni che non siano basate su interfacce grafiche o sull'utilizzo di schermi o dispositivi *mobile*. Infine, il progetto propone un artefatto interattivo di facile riproduzione tramite macchine a controllo numerico e elettronica e hardware open source.

### Modello di business

Il progetto è sviluppato dallo studio Solid Labs fondato da Filippo Yacob, Matteo Loglio, Valeria Leonardi. In seguito alla campagna Kickstarter conclusa il 22 dicembre 2013, il progetto è stato supportato da fondi per l'avviamento di aziende start-up. Il gruppo di Primo.io coordina e supervisiona le iterazioni per finalizzare il prodotto per la produzione industriale. I designer sono stati supportati da consulenti esterni per lo sviluppo della parte elettronica e per l'ingegnerizzazione delle componenti fisiche. Il modello di business include l'organizzazione di laboratori pratici a pagamento indirizzati ai bambini.

### 3.2.1 Modello di rilascio open source

Il prodotto è stato rilasciato con una licenza Creative Commons: Attribuzione – Condividi allo stesso modo – non commerciale. La licenza è applicata sia alla versione industriale del prodotto sia alla versione “prototipo”. Per la versione industriale è stata sviluppata una scheda hardware compatibile con Arduino che prevede una porta USB per la riprogrammazione del firmware. Il prodotto, seppur indirizzato a utenti non esperti (bambini, genitori, insegnanti) è aperto alle modifiche delle funzionalità e delle modalità di interazione.

Il manuale e la documentazione per riprodurre un derivato della versione “prototipo” sono pubblicate tramite la piattaforma di sviluppo software collaborativo GitHub ([docs.primo.io](https://docs.primo.io)).

Attraverso l'utilizzo di una funzionalità di GitHub che permette la pubblicazione di pagine html dinamiche, i designer di Primo hanno realizzato delle pagine web con il tutorial passo-passo a supporto della costruzione del prototipo.

Per facilitare il processo di collaborazione è stata implementata una funzionalità che permette alle persone di pubblicare la documentazione dei loro derivati.

In una sezione del manuale include, inoltre, i designer di Primo forniscono una lista di categorie dei principali contributi al fine di stimolare la creazione di una comunità di persone che possano partecipare attivamente allo sviluppo di Primo su temi specifici (es. la traduzione del manuale in diverse lingue).

### 3.2.2 Processo di progettazione e sviluppo

#### Un prototipo low-tech che diventa prodotto

Primo nasce come progetto di Matteo Loglio, studente del corso Designing Advanced Artifacts del Master of Advanced Studies in Interaction Design della SUPSI. L'obiettivo del corso era di progettare artefatti open source che potessero essere customizzati e riproducibili tramite tecnologie di produzione digitalizzata. Il prototipo realizzato per il corso si basa su scelte tecnologiche e soluzioni di interfaccia che sono rimaste invariate rispetto alla ver-

sione pubblicata sulla piattaforma Kickstarter, ovvero la versione che è stata finanziata per essere prodotta a livello industriale. La differenza principale è stata la modifica della metafora del robot che inizialmente aveva una forma di macchina di legno che contiene all'interno la scheda elettronica prodotta in modo artigianale. La possibilità di utilizzare tecnologie open source, tecnologie di prototipazione rapida e accedere a risorse condivise in rete ha permesso di realizzare un prototipo sul quale è stata sviluppata la

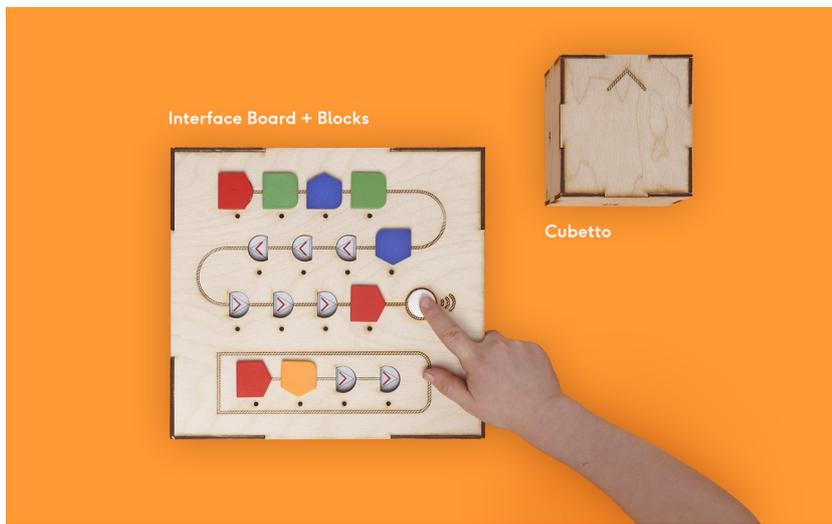


Fig. 7 Primo, il prodotto



Fig. 8 Primo, il primo prototipo

Fig. 9 La scheda hardware per controllare Cubetto

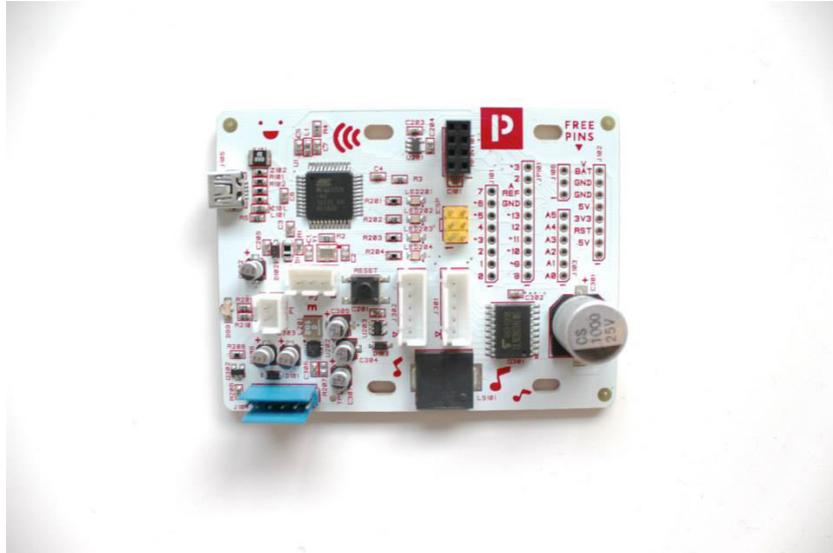
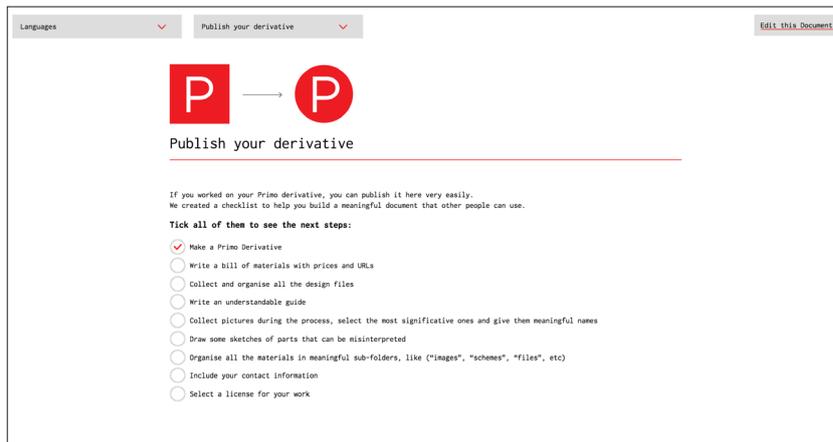


Fig. 10 Primo docs. La lista di possibili contributi per collaborare allo sviluppo del Primo



strategia di sviluppo collaborativo, ovvero il rilascio di una versione prototipo assemblabile in modo fai da te. Le scelte tecnologiche e di design “low-tech” hanno permesso di creare un prototipo e di distribuirlo come fosse un prodotto.

Un processo generativo: design – hardware –  
ecosistema di servizi e prodotti

Similmente al progetto Open Mirror, a partire dalla progettazione di Primo è stata realizzata una scheda hardware open source basata su Arduino (Primo.io 2014). Grazie a questa funzionalità il

firmware può essere riprogrammato per aggiungere o modificare le funzionalità del prodotto.

Il rilascio della documentazione ha permesso ai designer di supportare lo sviluppo di derivati da parte di altre persone: un derivato, per esempio, consiste nella programmazione di Cubetto tramite l'ambiente di programmazione visuale Scratch utilizzato per insegnare la programmazione ai bambini. Questa traduzione ha permesso di espandere il prodotto introducendo la possibilità di poter controllare Cubetto tramite un'interfaccia basata su tastiera e mouse e un'applicazione programmata con l'ambiente di programmazione grafico Scratch (Dirk 2014).

La programmabilità dell'oggetto fisico permette di estendere l'esperienza d'uso del prodotto che può essere connesso a nuove tipologie di interfaccia. Sulla base di questo principio, i designer hanno definito un ecosistema di servizi e applicazioni che estendono Primo attraverso l'integrazione di interfacce *touch* per dispositivo mobili e *tablet*.



## 4. “Progettare prodotti-piattaforma” Proposta di un framework per il design di artefatti interattivi open source

### 4.1 Cosa sono i prodotti-piattaforma?

Dopo aver definito l’“Ecosistema open source” e individuato i livelli progettuali, ovvero i possibili livelli di intervento del design nella progettazione e sviluppo di un prodotto interattivo rilasciato in open source, presento in questo capitolo la proposta di un *design framework* a supporto della progettazione di prodotti interattivi open source. In particolare, il *framework* punta a definire i concetti, le attività e i processi per la progettazione di prodotti che gli utenti possono modificare o sviluppare ulteriormente grazie a un ecosistema di servizi digitali, documentazione condivisa e licenze libere. Questi prodotti sono definiti all’interno del *framework* “prodotti-piattaforma” laddove questa definizione incarna in maniera più esatta le caratteristiche di una tipologia di artefatti interattivi in cui le pratiche open source sono informate dagli approcci, metodi e pratiche proprie della disciplina del design. Cosa sono i prodotti-piattaforma?

In informatica, una piattaforma è un gruppo di tecnologie che sono usate come base sulle quali altre applicazioni, processi e tecnologie sono sviluppate (Techopedia 2014). I prodotti-piattaforma si distinguono da altre tipologie di artefatti interattivi fisici perché:

- sono prodotti programmabili, ovvero sono dotati di interfacce fisiche e digitali che permettono agli utenti di interagire con le interfacce fisiche e digitali che permettono di manipolare e intervenire sulle sue componenti software e hardware al fine di modificarle per svariati scopi;
- sono prodotti riproducibili, ovvero possono essere realizzati

da parte di altri grazie al sistema delle licenze open source che garantisce che chiunque possa ricreare il prodotto a partire dalla documentazione condivisa e resa accessibile su internet;

— i prodotti-piattaforma sono, infine, caratterizzati da generatività poiché a partire da essi è possibile creare molteplici varianti di prodotti derivati.

Similmente alle tecnologie di prototipazione hardware e software open source come Arduino, i prodotti-piattaforma abilitano le persone a realizzare dei progetti customizzati intorno alle loro esigenze, richiedono la definizione di modelli di business basati sulla collaborazione e su modelli di produzione di tipo distribuito. A differenza delle piattaforme tecnologiche, sono prodotti le cui modalità di interazione e qualità estetico-formali sono progettate per veicolare una specifica esperienza d'uso modellata intorno all'obiettivo di soddisfare delle esigenze specifiche (p.e. ascoltare la musica, imparare la programmazione giocando), ma potenzialmente espandibile per adattarsi ad altri contesti di utilizzo.

Infine, i prodotti-piattaforma sono prodotti complessi in cui ad ogni elemento è agganciato un insieme d'informazioni che sono veicolate all'utente finale tramite supporti, formati, interfacce digitali e fisiche in modo che questi possa facilmente interagire con esse.

Al fine di progettare un prodotto-piattaforma è necessario definire dei requisiti specifici e dei processi che non sono comple-

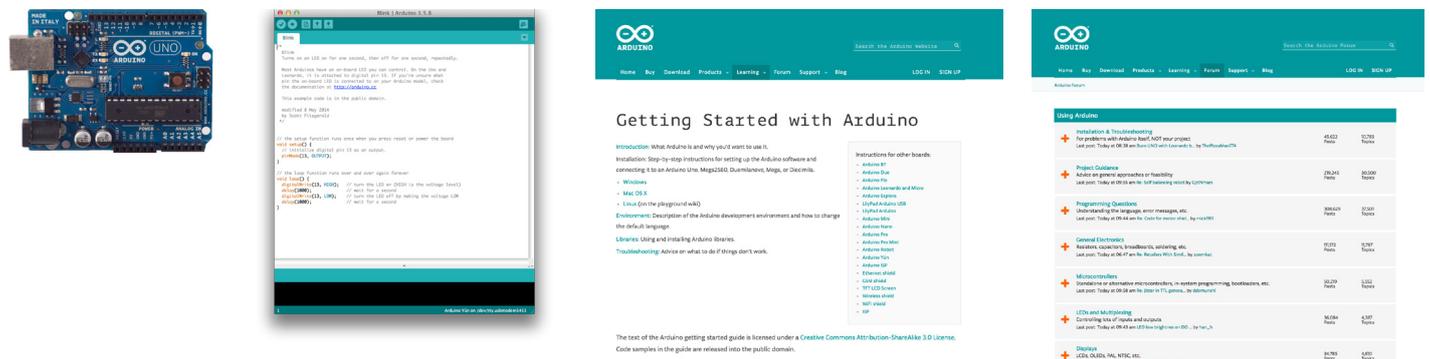
Fig. 1 Elementi generali di una piattaforma di prototipazione software-hardware

Hardware

Software

Documentazione e file sorgente

Comunità on-line



Comportamenti interattivi

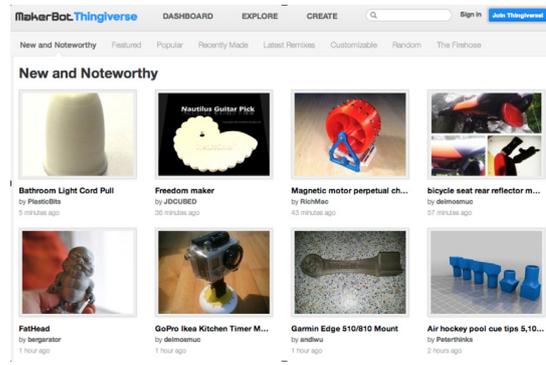
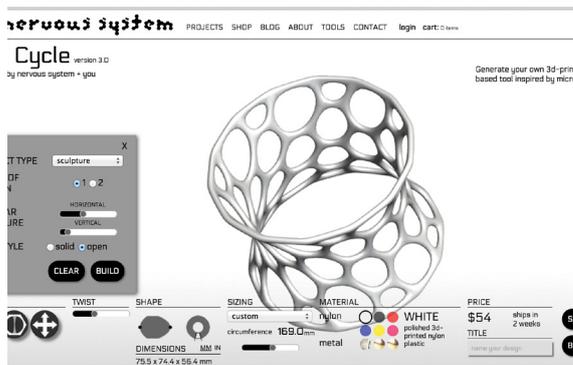
Parti fisiche

Hardware e software



Ambienti di co-progettazione

Piattaforme collaborative



tamente derivati da quelli applicati nel contesto dello sviluppo di piattaforme di prototipazione open source o dei prodotti di design non tecnologici. Il *framework* "Progettare prodotti-piattaforma" è una proposta per avviare la riflessione sulla necessità di definire nuovi metodi per la progettazione di artefatti interattivi in relazione alle emergenti pratiche di sviluppo open source.

Fig. 2 Elementi generali di un prodotto-piattaforma

#### 4.2 Descrizione del framework: dall'User Centered Design e Meta-design al design dei prodotti-piattaforma

Il *framework* di progettazione dei prodotti-piattaforma consiste in un insieme di strumenti concettuali a supporto della progettazione di artefatti interattivi che sono compatibili con le pratiche e i modelli della progettazione e dello sviluppo open source. Gli strumenti concettuali servono a definire i requisiti e le attività di progettazione per definire le funzionalità e le caratteristiche dei prodotti-piattaforma e insieme vanno a definire uno specifico processo di progettazione.

L'obiettivo del *framework* è delineare gli aspetti salienti di una progettazione mirata al rilascio di prodotti-piattaforma con un approccio che combina i modelli open source con i metodi e i processi dello User Centered Design e del Meta-design.

Lo User Centered Design (UCD) ha l'obiettivo di definire le interazioni con i prodotti e i servizi in modo che siano più appropriati per le persone e il contesto d'uso. La metodologia dello UCD è il modo attraverso il quale i designer applicano un approccio ecologico nella progettazione delle relazioni tra i prodotti e i servizi e i loro utenti. Il modo in cui gli utenti raggiungono l'obiettivo o soddisfano un loro bisogno dipende da come i designer combinano le proprietà di un prodotto in modo che possano supportare attività e bisogni delle persone. Inoltre, la metodologia UCD punta a valutare come definire un processo di progettazione e può essere applicata in base alle opportunità e ai limiti di un progetto. In definitiva, qualunque siano tali opportunità e i limiti del processo basato su UCD, le attività di progettazione puntano a generare prodotti e servizi che soddisfano diversi tipi di bisogni, da quelli funzionali a quelli relazionali e simbolici (Norman, Draper 1986).

Meta-design è un approccio alla progettazione definito nell'ambito degli studi di Human Computer Interaction e, si propone di offrire una prospettiva complementare alle metodologie di progettazione dello User Centered Design e del Participatory Design, che punta a coinvolgere gli utenti in maniera più diretta nel pro-

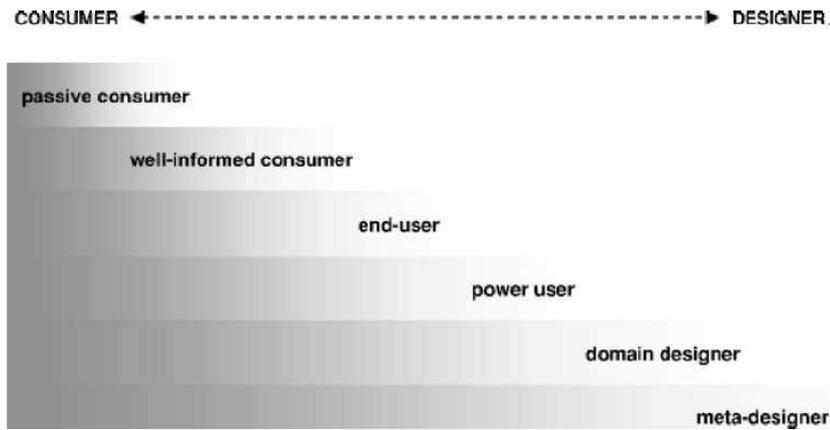


Fig. 3 Lo spettro del consumatore/designer (Giaccardi, Fisher 2005)

cesso di progettazione come co-designer delle proprie soluzioni (Fischer, Scharff 2000) (Scharff 2003).

A differenza di questi approcci, il Meta-design punta a definire un processo di progettazione di sistemi tecnologici (*system design*) che siano aperti, che possano essere modificati dagli utenti e che evolvano in base al loro uso supportando interazioni sempre più complesse. Il Meta-design estende la nozione di *system design* tradizionale, proprio della disciplina dell'informatica e dell'ingegneria del software, includendo un processo co-adattivo tra gli utenti e il sistema in cui gli utenti diventano dei co-sviluppatori. Il Meta-design cerca di soddisfare alcune necessità degli ambienti socio-tecnologici: questi devono essere flessibili ed evolversi perché non possono essere progettati completamente prima del loro utilizzo da parte degli utenti.<sup>1</sup>

Secondo l'approccio del Meta-design gli utenti sono consumatori in certe situazioni e designer in altre, quindi, consumatore o designer non è un attributo di una persona, ma il ruolo che una persona assume in un dato contesto.

Uno degli ambiti applicativi degli approcci basati sul Meta-design secondo Giaccardi e Fisher è lo sviluppo open source. I punti di contatto tra Meta-design e open source riguardano la possibilità per gli utenti di poter modificare il software senza avere l'impressione che quest'attività sia difficile in modo che siano aperti a partecipare. Queste modifiche devono essere tecnica-

1 Ibidem

Fig. 4 Tabella comparativa tra system design tradizionale e Meta-design (Giaccardi & Fisher 2005)

*Table 19.1. Traditional design versus meta-design*

Traditional design	Meta-design
Guidelines and rules	Exceptions and negotiations
Representation	Construction
Content	Context
Object	Process
Perspective	Immersion
Certainty	Contingency
Planning	Emergence
Top-down	Bottom-up
Complete system	Seeding
Autonomous creation	Co-creation
Autonomous mind	Distributed mind
Specific solutions	Solutions spaces
Design-as-instrumental	Design-as-adaptive
Accountability, know-what (rational decisioning)	Affective model, know-how (embodied interactionism)

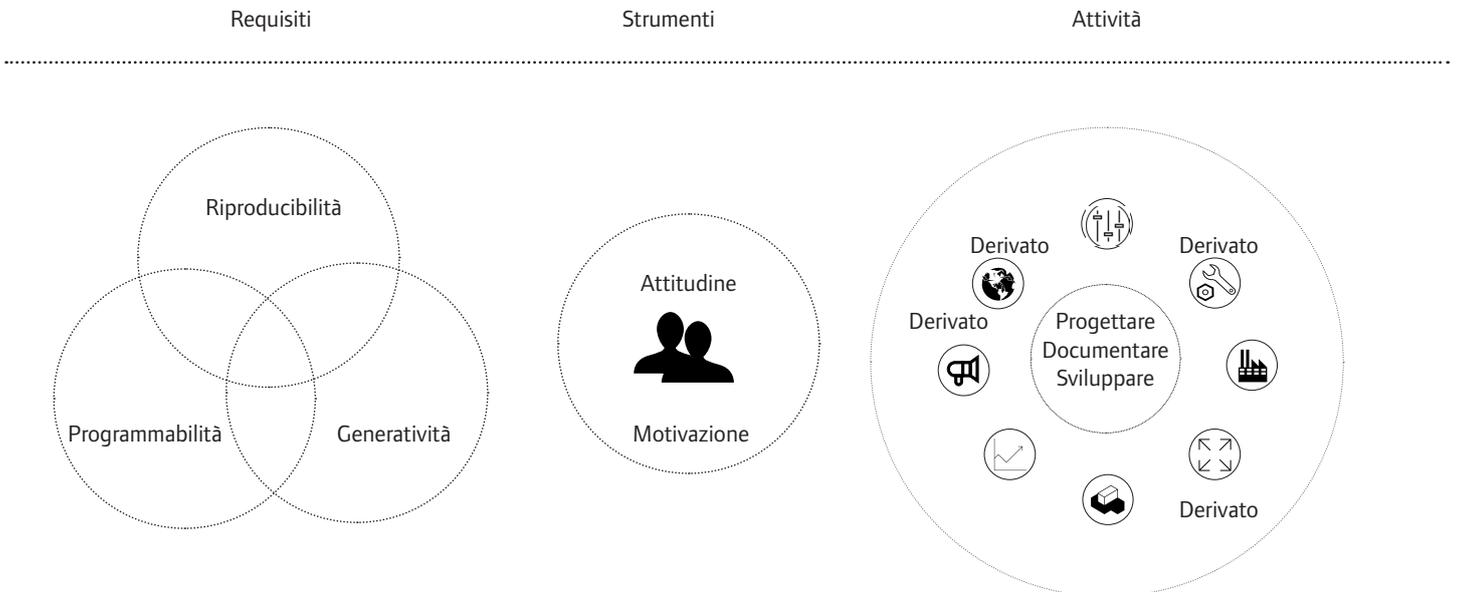
mente implementabili e gli utenti devono percepire il beneficio della loro partecipazione sia essa professionale, sociale o personale. Le attività che gli ambienti open source devono supportare sono quelle in cui gli utenti si sentono direttamente coinvolti e che non sono considerate marginali. Infine, le barriere alla condivisione devono essere molto basse in modo da favorire i partecipanti a condividere le modifiche in modo semplice e a tenere traccia di queste.

Combinando alcuni aspetti chiave proposti da UCD e dall'approccio del Meta-design, il *framework* di progettazione di prodotti-piattaforma si basa su un insieme di requisiti, strumenti e attività. I **requisiti** sono rappresentati dalle caratteristiche o condizioni necessarie a partire dalle quali si affronta la realizzazione di un artefatto interattivo open source. I requisiti sono tre: la programmabilità, la riproducibilità, la generatività. Questi tre requisiti devono guidare la definizione dell'esperienza d'uso dell'artefatto interattivo piuttosto che rispondere a condizioni di tipo tecnologico

(p.e. la programmabilità fornita dalla presenza di codice informatico per definire i comportamenti dell'artefatto). Per questo motivo i requisiti del *framework* sono da considerare come "requisiti di alto livello".

**Gli strumenti** metodologici permettono di mettere in relazione le scelte progettuali derivate dai requisiti di alto livello con i possibili utenti finali del prodotto interattivo. Questi strumenti metodologici consistono nell'insieme delle informazioni che permettono di prefigurare i bisogni dell'utente in base a caratteristiche tipo descritte come "attitudini" e "motivazioni". Infine, il terzo elemento del framework sono **le attività**. Le attività definiscono gli aspetti salienti del processo di progettazione dell'artefatto interattivo open source e il conseguente processo di sviluppo "open" del prodotto, ovvero il processo basato sull'interscambio tra il designer o gruppi di designer che guidano lo sviluppo del prodotto e le persone o i gruppi che realizzano derivati del prodotto.

Fig. 5 Macro-elementi del framework "Progettare prodotti-piattaforma"



#### 4.2.1 I requisiti. La programmabilità, la riproducibilità, la generatività

I requisiti sono le condizioni che devono essere soddisfatte per progettare e sviluppare un prodotto-piattaforma. Il processo di progettazione non deve necessariamente focalizzarsi sui tre requisiti in egual misura, ma può essere incentrato su un singolo requisito che serve da elemento principale per la definizione delle caratteristiche dell'artefatto interattivo.

##### a. La programmabilità

Dall'analisi di casi studio Primo e Open Mirror emerge che i prodotti interattivi open source si basano su schede hardware dotate di una porta USB attraverso la quale è possibile connettersi per aggiornare il firmware e, in questo modo, modificare i comportamenti del prodotto definiti attraverso la riscrittura o la modifica del codice informatico. La programmabilità dei prodotti-piattaforma è, quindi, legata principalmente all'implementazione di specifiche interfacce fisiche e dei corrispondenti protocolli di comunicazione. Inoltre, la programmabilità deriva dall'utilizzo di componenti software e hardware standard, ovvero di facile accessibilità sul mercato e con una documentazione disponibile in rete. A un livello più astratto, la programmabilità di un prodotto-piattaforma deriva dalle scelte strategiche di integrare soluzioni tecnologiche e approcci consolidati all'interno di una comunità estesa di utilizzatori. Dal punto di vista tecnologico, per esempio, il prodotto Primo è disponibile in una versione prototipo che si basa sull'utilizzo di comuni resistenze elettriche per il riconoscimento dei blocchetti con le istruzioni di movimento del Cubetto. Allo stesso tempo, il codice utilizzato per implementare i comportamenti del prodotto è stato scritto tramite il software Arduino. Infine, la programmabilità deriva dalla possibile riconfigurazione delle interfacce fisiche e digitali del prodotto che sono pensate come moduli modificabili e intelligenti, ovvero costituiti da elementi fisici e informazione digitale. Nel progetto Primo i blocchetti possono essere modificati sia nella forma così come sono modificabili le istruzioni di movimento del cubetto che avvengono tramite la modifica del software. Nel progetto Open Mirror, i

sensori di rilevamento del movimento delle mani possono essere utilizzati come interfacce di input per controllare qualsiasi applicazione digitale di uno smart-phone collegato allo specchio.

#### b. La riproducibilità

Le licenze libere permettono che un artefatto interattivo sia riprodotto da altri senza che debba essere richiesto il permesso agli autori del progetto. La possibilità di creare una riproduzione di un artefatto, che rappresenta il diritto fondamentale di chi detiene la proprietà intellettuale di un prodotto di design, nel contesto open source si traduce in una delle qualità del prodotto-piattaforma che è concepito al fine di favorire altri individui nella creazione di prodotti derivati. La replica è sostituita dal concetto di prodotto derivato ed è garantita dalla pubblicazione dettagliata della documentazione del prodotto e di tutte le sue componenti, dal software all'hardware fino alle parti fisiche (il corpo dell'oggetto). I prodotti-piattaforma possono essere riprodotti poiché caratterizzati da un alto livello di riproducibilità poiché sono rilasciati con una documentazione che include anche le informazioni sulle tecniche di assemblaggio o di produzione tramite macchine di produzione digitalizzata. I prodotti-piattaforma sono, quindi, rilasciati con una documentazione condivisa su applicazioni digitali realizzate ad hoc per il progetto o tramite servizi web di terzi (p.e. GitHub, Instructable). Un esempio è il sito ufficiale di Open Mirror che include la documentazione per riprodurre il prodotto tramite un manuale, i *blueprint* e i video che documentano le fasi di assemblaggio. La riproducibilità dei prodotti-piattaforma non è garantita solo dall'accessibilità della documentazione, ma dalla progettazione di soluzioni d'interfaccia per la distribuzione della documentazione che aumentano l'accessibilità della stessa documentazione offrendo, allo stesso tempo, un'esperienza d'uso. Nel caso di Primo, le soluzioni d'interfaccia per il caricamento delle traduzioni del manuale o della documentazione dei derivati da parte di altri favorisce il processo di partecipazione da parte di altri allo sviluppo di prodotti derivati.

La riproducibilità dipende, infine, dalla modularità dei prodotti-piattaforma intesa sia come insieme di parti distinte e sostituibili con altre parti standard sia come possibilità di ricon-

figurazione dei moduli di base che compongono il prodotto (p.e. Littlebits). Infine, la riproducibilità riguarda la possibilità di produrre un prodotto-piattaforma utilizzando tecniche e tecnologie di produzione di diverso tipo, dalle tecnologie di *personal fabrication* e di prototipazione rapida a tecnologie industriali avanzate e digitalizzate.

### c. La generatività

La generatività è il requisito che riguarda la possibilità di modifica e riutilizzo delle parti e dei componenti di un prodotto-piattaforma per creare prodotti derivati con funzionalità diverse o prodotti derivati che integrano nuove componenti che espandono le funzionalità originali del progetto originale. La generatività dei prodotti-piattaforma riguarda il rilascio della documentazione del progetto tramite sistemi che facilitano il monitoraggio della partecipazione da parte di altre persone che realizzano prodotti derivati. L'utilizzo di sistemi che semplificano l'interazione con i membri della comunità on-line risponde anche al requisito di generatività: i sistemi di *Versioning System Control*, ovvero applicazioni che supportano lo sviluppo collaborativo di codice informatico, permettono di tracciare le diverse versioni di prodotto realizzate da altri, di monitorare i diversi tipi di contributi, di interagire con chi contribuisce per beneficiare della realizzazione e dello sviluppo di altra documentazione di prodotti derivati. La generatività dipende anche dall'utilizzo di ambienti di co-progettazione parametrica (vedi paragrafo 2.4) che permettono agli utenti di progettare le parti dei prodotti derivati a partire da elementi di design di base configurabili e modificabili. La progettazione di un artefatto tramite un approccio e le tecniche di design parametrico aumenta la capacità di un progetto di essere generativo poiché le variazioni dell'artefatto costituisce una qualità del progetto e, in molti casi, il suo obiettivo finale.

Programmabilità	Riproducibilità	Generatività
<ul style="list-style-type: none"> <li>- Implementazione di interfacce fisiche e protocolli di comunicazione standard</li> <li>- Utilizzo di componenti hardware e software con documentazione accessibile in internet</li> <li>- Utilizzo di soluzioni tecnologiche condivise da una larga comunità di sviluppo</li> <li>- Progettazione di parti modulari riconfigurabili e intelligenti (informazione ed elemento fisico)</li> </ul>	<ul style="list-style-type: none"> <li>- Adozione di licenze libere</li> <li>- Documentazione delle tecniche di assemblaggio</li> <li>- Documentazione delle tecnologie di produzione</li> <li>- Condivisione delle documentazione su siti ad hoc o siti terzi</li> <li>- Implementazione di soluzioni di interfaccia di piattaforme digitali per facilitare la collaborazione e partecipazione</li> <li>- Progettazione di parti modulari sostituibili</li> <li>- Implementazione di soluzioni di design modulare e riconfigurabile</li> <li>- Produzione tramite tecnologie di personal e digital fabrication</li> </ul>	<ul style="list-style-type: none"> <li>- Creazione di prodotti derivati uguali al progetto originale</li> <li>- Creazione di prodotti derivati che modificano le funzionalità e l'esperienza del progetto originale</li> <li>- Utilizzo o implementazione di applicazioni per il monitoraggio dei contributi o dei derivati realizzati da parte di altri, per l'interazione con i membri della comunità on-line</li> </ul>

Tabella 1. Gli aspetti salienti dei requisiti del framework "Progettare prodotti-piattaforma"

#### 4.2.2 Gli strumenti: le attitudini e le motivazioni dell'utente

Al fine di integrare i bisogni degli utenti come elemento di progettazione nel *framework* ho sviluppato due strumenti che si basano uno degli strumenti metodologici più diffusi dello User Centered Design, ovvero la definizione di persone tipo e di scenari (Brown 2011). I due strumenti sono le attitudini e le motivazioni degli utenti.

**Le attitudini** definiscono le caratteristiche archetipiche degli utenti tipo e visualizzano le competenze e relazioni che questi hanno con la tecnologia. La scelta di individuare delle attitudini deriva dalla volontà di classificare gli utenti non in modo rigido, ma di dichiarare come le persone possono assumere diversi di ruoli in base al contesto. Lo strumento fa riferimento allo spettro consumatore/designer presentato all'inizio di questo capitolo (Fig. 3). La definizione delle attitudini serve, inoltre, a chiarire l'emergere di figure nuove come, per esempio, i maker che rispecchiano, a mio parere, un tipo di attitudine piuttosto che una categoria

professionale. Come lo spettro dell'utente secondo l'approccio del Meta-design, le attitudini puntano a descrivere i possibili ruoli che un utente può assumere in relazione alla tipologia di artefatto e al contesto d'uso. A differenza delle categorie generiche definite dallo spettro del Meta-design, le attitudini sono state definite a partire dall'analisi del dominio degli artefatti interattivi open source.

### Le attitudini

#### Il professionista tecnico

Il professionista tecnico ha conoscenze tecniche e tecnologiche specialistiche, ovvero è esperto di un dominio specifico come, per esempio, la programmazione informatica (*domain expert*). Svolge attività indirizzate alla risoluzione di problemi specifici, quindi, i suoi bisogni sono legati a compiti definiti, per esempio, in un processo di lavoro (p.e. modificare una funzionalità di un programma informatico). Al fine di sviluppare maggiori conoscenze e competenze, il professionista tecnico segue corsi per l'aggiornamento sui temi relativi al dominio di cui è esperto.

#### Il maker

Il maker ha conoscenze tecnologiche specialistiche sullo sviluppo hardware e software. Realizza prodotti e soluzioni "prototipando", ovvero creando artefatti funzionanti che mostrano le funzionalità e le modalità di interazione di un prodotto interattivo. Per ampliare le proprie conoscenze, in relazione alla risoluzione di un problema, utilizza servizi collaborativi per trovare e condividere guide passo passo e la documentazione utile a riprodurre un prodotto. Collabora sia con comunità di persone in luoghi fisici (p.e. fablab, makerspace, hackerspace) e comunità on-line.

#### L'amatore DIY

L'amatore DIY ha conoscenze tecnologiche di base e punta a realizzare prodotti e soluzioni di media complessità sulla base di un bisogno personale. I prodotti e le soluzioni non sono mai ingegnerizzati, ma sono prototipi e prodotti assemblati unici. L'amatore DIY utilizza internet e i manuali per riprodurre un progetto come, per esempio, le guide passo passo per realizzare una radio FM in modo fai da te.

### Il consumatore

Il consumatore ha conoscenze tecnologiche generaliste. Acquista le soluzioni di cui ha bisogno tramite internet o presso i negozi. Al fine di trovare una soluzione a un problema specifico si affida a esperti e al passaparola.

### L'imprenditore

L'imprenditore può avere conoscenze tecnologiche sia generaliste sia specifiche. Si occupa di fare ricerca sui prodotti e le soluzioni di cui ha bisogno tramite la comparazione e l'analisi dello stato dell'arte (*benchmarking*). Pensa in maniera strategica e non si focalizza su un singolo aspetto di un progetto, ma approfondisce gli aspetti legati alla distribuzione e al marketing.

### Le motivazioni

Le motivazioni descrivono i diversi obiettivi di un utente che interagisce con un prodotto-piattaforma. Le motivazioni descrivono principalmente gli obiettivi relati alla possibilità di poter accedere e modificare gli elementi hardware, software e il design di un prodotto. Le motivazioni sono state individuate a partire dall'analisi delle definizioni di open source hardware e open design e alle pratiche di condivisione dei file originali di prodotti open source così come presentati nel capitolo 2.

### Customizzare

La motivazione "customizzare" è riferita alla customizzazione dello stile, ovvero sulla base di forme e funzionalità predefinite e standard è possibile modificare gli elementi di stile di un prodotto o dispositivo quali il colore, il materiale e gli accessori. Dati un numero definito di parametri l'utente è abilitato a modificare lo stile di un prodotto. Un esempio di customizzazione è il progetto NikeID, un'interfaccia web che permette di selezionare da una libreria esistente, le parti della scarpa Nike e i diversi colori e accessori disponibili.

### Riparare

La motivazione "riparare" riguarda l'attività di aggiustare le

componenti fisiche di un prodotto o dispositivo, le sue parti hardware o correggere nei file di progettazione in vista della produzione del prodotto. Un esempio di riparazione è anche riscrivere il codice del software del dispositivo per eliminare

Fig 6. Immagini rappresentative dello strumento "Attitudini"



un errore o riprodurre una parte fisica perché si è usurata in seguito all'utilizzo.

### Migliorare

Migliorare riguarda la modifica con l'obiettivo di perfezionare o ottimizzare gli aspetti funzionali di un prodotto o dispositivo. Al fine di migliorare un prodotto, si possono, per esempio, implementare funzionalità o elementi di design nuovi o non realizzati dagli ideatori di un progetto. Nel contesto dello sviluppo software, migliorare vuol dire ottimizzare la performance di un software come la velocità di reazione ad un determinato input da parte dell'utente. Migliorare un prodotto, non va a modificare il tipo di funzione e il tipo di uso che si fa del prodotto.

### "Building upon"

"Building upon" (costruire a partire da) un prodotto o dispositivo espandendo le sue funzionalità, progettare e implementare nuove parti, sia componenti hardware e sia estensioni software, realizzare nuovi prodotti e dispositivi a partire dal prodotto iniziale o da alcune sue parti (hardware, software, design). Il "building upon" implica eventualmente il cambiamento degli ambiti di applicazione del dispositivo o la sua funzionalità principale: un esempio è la trasformazione di un quadricoptero in un dispositivo di trasporto di beni oppure utilizzare i suoi motori e le parti per realizzare un giocattolo robot per bambini.

### Espandere

Espandere un prodotto o un dispositivo vuol dire ideare, progettare e implementare nuove soluzioni: nuovi prodotti o dispositivi compatibili, nuovi sistemi di interfaccia, nuove applicazioni software o hardware compatibili con altre piattaforme o servizi e con il prodotto iniziale. A differenza del "building upon" espandere non implica il cambio degli ambiti di applicazione del prodotto, ma solo la messa in comunicazione del prodotto iniziale con altri sistemi o altri prodotti o altre tecnologie esistenti. Un esempio di espansione di un prodotto è la realizzazione di un'applicazione desktop basata su Scratch ([scratch.mit.edu](http://scratch.mit.edu)) che permette di controllare il Cubetto del progetto Primo ([primo.io](http://primo.io)) che prevede, invece, l'utilizzo di un'interfaccia fisica.

### Produrre

La motivazione dell'utente ad accedere e interagire con le informazioni e con il prodotto-piattaforma può riguardare esclusivamente la sua riproduzione singola o in serie sia industrialmente sia in modo fai da te con il supporto di tecnologie di produzione digitalizzata.

### Distribuire

L'utente finale di un progetto di prodotto-piattaforma può puntare alla sua distribuzione.

La motivazione "distribuire" inquadra, quindi, le scelte sottostanti futuri modelli di distribuzione di artefatti interattivi a livello locale e globale.

### Promuovere

La documentazione di un prodotto-piattaforma rilasciata in open source permette di supportare la circolazione libera delle informazioni sul prodotto a fini promozionali. Promuovere è la motivazione di utenti che intendono comunicare il prodotto o un dispositivo tramite canali di comunicazione tradizionali, stampa, internet, social networking.

## 4.2.3 Le attività: il processo di progettazione e il processo di sviluppo

Le attività sono legate al processo di progettazione e al processo di sviluppo del prodotto-piattaforma.

Le attività del processo di progettazione riguardano prima di tutto l'ideazione del prodotto interattivo. L'ideazione è un'attività che avviene sia prima dell'analisi dei requisiti sia dell'analisi dei bisogni degli utenti che si basa sull'utilizzo degli strumenti "attitudine" e "motivazione". Successivamente, si punta alla progettazione e prototipazione delle parti del prodotto-piattaforma. La progettazione si riferisce alla definizione e prototipazione delle caratteristiche formali e funzionali del prodotto-piattaforma e, in particolare, ha come oggetto il design delle parti fisiche del prodotto, il design dell'interazione, il design del sistema e dell'hardware, il design degli eventuali ambienti di co-progettazione pa-

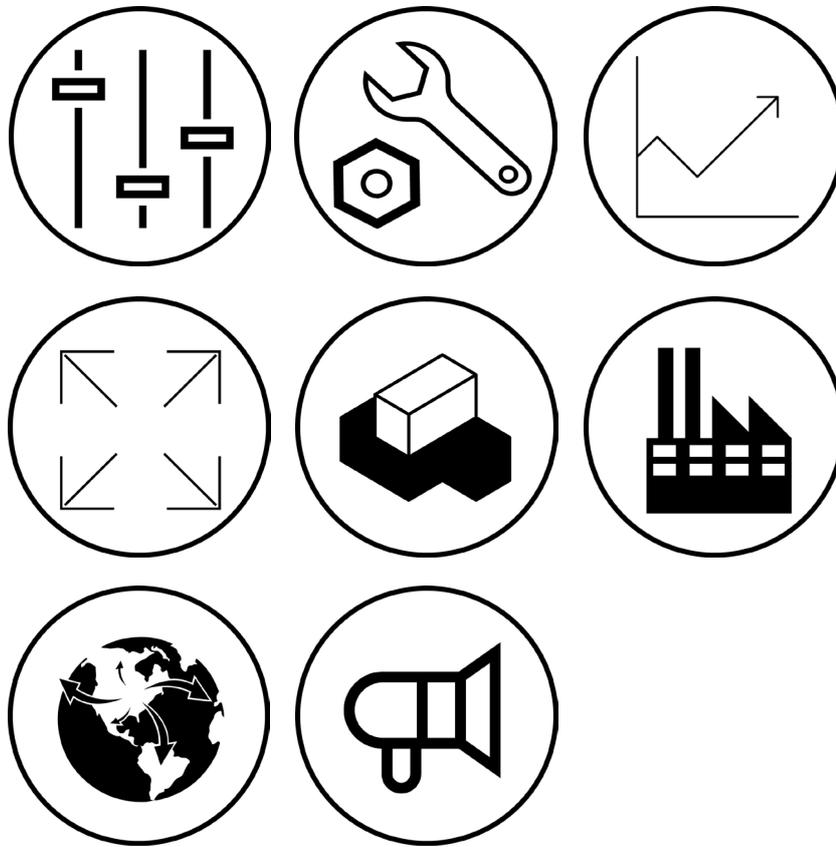
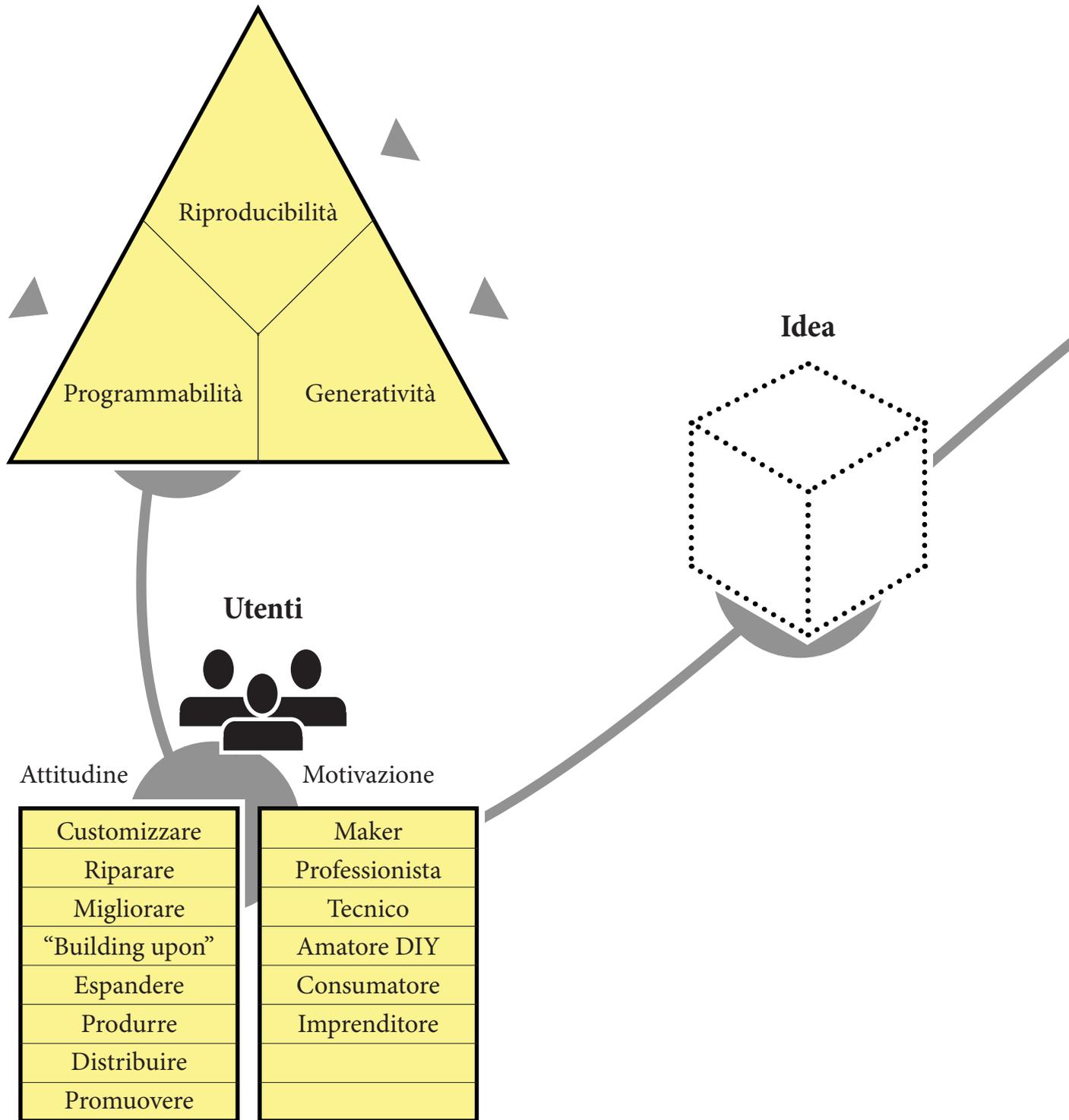


Fig. 7 Le icone rappresentative dello strumento "Motivazioni"

rametrica. Infine, si avvia l'attività di documentazione dei risultati delle attività di progettazione tramite la creazione di documenti multimediali quali testi, foto, diagrammi, video, commenti del software.

La progettazione, la prototipazione e la documentazione sono attività che dovrebbero essere iterative e non consequenziali. Alla progettazione e prototipazione di ogni nuovo elemento del prodotto-piattaforma dovrebbe seguire la sua documentazione che va migliorata al momento della pubblicazione, vendita e distribuzione del prodotto-piattaforma. Altre attività chiave del processo di progettazione sono la progettazione di applicazioni interattive web ad hoc per il rilascio del prodotto o la selezione e l'integrazione di applicazioni e servizi che organizzano e rendono accessibile la documentazione del prodotto-piattaforma. La prima attività riguarda la definizione dell'architettura dell'informazione

## Progettazione





e dell'interfaccia di un'applicazione digitale che rende accessibile la documentazione e che include funzionalità per gestire la successiva fase di sviluppo collaborativo del prodotto. L'integrazione di applicazioni interattive web e servizi terzi permettono, invece, di gestire la collaborazione e la partecipazione di altre persone al processo di sviluppo secondo degli standard condivisi da altri progetti open source. Le applicazioni possono essere le Wiki o i servizi di sviluppo collaborativo quali, per esempio, GitHub. Queste due attività dipendono dalle strategie alla base della diffusione del prodotto e da fattori quali il tempo e gli investimenti disponibili per realizzare il prodotto-piattaforma.

Le attività del processo di sviluppo consistono nel promuovere il prodotto-piattaforma al fine di creare una comunità di interesse costituita da sviluppatori e utilizzatori finali, stimolare la partecipazione tramite l'interazione diretta con i membri della comunità online e fare ricerca, ovvero studiare il dominio di riferimento del prodotto al fine di comprendere le possibili strategie di sviluppo futuro.

Un'altra attività riguarda la possibilità di migliorare il prodotto, la documentazione e le applicazioni correlate al prodotto tramite successive iterazioni in base alle necessità interne al progetto o in base ai commenti e ai contributi forniti da altre persone. Infine, è importante monitorare i contributi dei membri della comunità o la realizzazione di prodotti derivati da parte di altre persone al fine di influenzare le iterazioni destinate al miglioramento del prodotto.

Le attività del processo di sviluppo sono basate su processi iterativi e dipendono, secondo la struttura del framework, dalla combinazione sia di fattori endogeni, ovvero la capacità di innovazione interna al progetto e alle competenze del gruppo di lavoro, sia da fattori esogeni ovvero l'interazione con i membri della comunità che contribuiscono al progetto o realizzano dei derivati del prodotto.

Il *framework* punta a rappresentare e guidare le fasi iniziali del rilascio di un progetto di artefatto interattivo open source e non definisce le attività legate all'evoluzione del processo di sviluppo da parte dei membri della comunità che possono potenzialmente non interagire con gli autori del progetto originario.

### 4.3 "Progettare prodotti-piattaforma" Applicazione del design framework in un workshop

Al fine di validare il framework di progettazione di prodotti-piattaforma ho organizzato un workshop pratico che ha coinvolto designer e studenti nel processo di progettazione di prodotti interattivi derivati da prodotti open source esistenti. Il workshop ha permesso di valutare il framework di progettazione e gli strumenti progettuali sui quali si basa.

Gli obiettivi del workshop sono molteplici. Il primo è formare i partecipanti sui concetti base della progettazione e sviluppo di prodotti open source. Il prerequisito alla base delle attività di progettazione è la comprensione dei diversi elementi dell'Ecosistema open source (come definiti nel capitolo 1), le definizioni attuali di open source hardware e design e l'utilizzo delle licenze libere, in particolare, le licenze Creative Commons (trasferimento di conoscenza). Il secondo obiettivo è presentare i casi studio di prodotti interattivi open source come esempi dello stato dell'arte nell'ambito delle pratiche di progettazione open source, in particolare, legate all'innovazione. L'obiettivo ultimo è fornire ai partecipanti degli esempi pratici in cui sono implementate soluzioni a favore della programmabilità, riproducibilità e generatività di un prodotto interattivo. I casi studio sono presentati per introdurre i partecipanti ai principi di progettazione che emergono nel contesto della progettazione di prodotti open source (analisi dei casi studio). Infine, il workshop ha l'obiettivo di generare idee di prodotti interattivi e visualizzarle tramite tecniche di prototipazione "a bassa fedeltà", ovvero la realizzazione di prototipi tramite carta o disegni digitali senza l'implementazione di codici informatici. Le attività di progettazione e prototipazione sono finalizzate a definire delle idee di possibili prodotti e ad applicare in maniera pratica le conoscenze acquisite riguardo alle pratiche di progettazione open source (generazione e prototipazione di idee di prodotti interattivi).

### 4.3.1 La metodologia del workshop

Il workshop “Progettare prodotti-piattaforma” punta a coinvolgere designer e studenti nel processo di generazione di idee di prodotti interattivi open source ovvero prodotti che gli utenti possono modificare o sviluppare ulteriormente grazie a un ecosistema di servizi digitali, documentazione condivisa e licenze libere. Obiettivo del workshop è concepire e progettare nuovi prodotti interattivi a partire da progetti esistenti rilasciati in open source al fine di comprendere i limiti e le opportunità dell’attuale ecosistema open source, sia le tecnologie, le interfacce, le comunità; analizzare i livelli di configurazione di un prodotto interattivo il cui software, hardware e design sono rilasciati in open source; comprendere le differenze tra le diverse licenze libere (p.e. Licenze Creative Commons).

Uno degli scopi principali è definire le informazioni e le competenze che devono essere trasferite agli utenti finali per supportarli nel processo di customizzazione e uso di un prodotto open source.

Il workshop “Progettare prodotti-piattaforma” si sviluppa in quattro fasi:

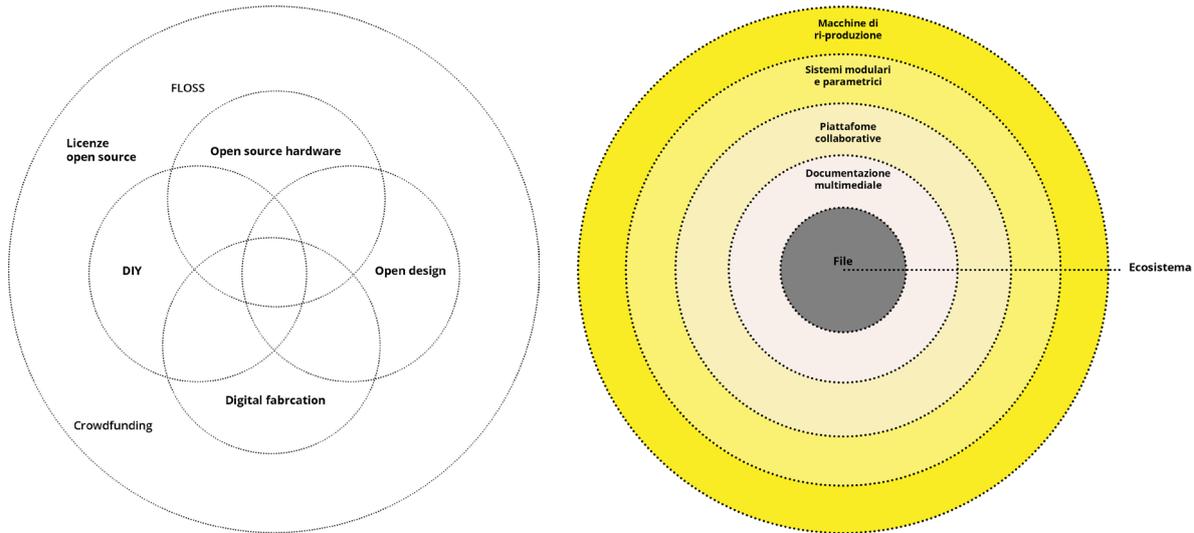
1. Introduzione degli elementi di base dell’Ecosistema open source e dei livelli progettuali nelle pratiche open source;
2. Analisi di prodotti interattivi open source;
3. Definizione dei bisogni tramite la selezione dell’attitudine e della motivazione dell’utente;
4. Definizione e prototipazione a bassa fedeltà (*low fidelity*) di un prodotto derivato.

#### 1. Introduzione agli elementi di base dell’Ecosistema open source e dei Livelli Progettuali

La prima attività del workshop consiste nella presentazione degli elementi di base dell’Ecosistema open source e dei relativi livelli progettuali così come sono stati definiti e documentati nei capitoli 1 e 2 della ricerca. Per presentare i contenuti della ricerca sono utilizzati due diagrammi: il diagramma dell’Ecosistema open source che si basa su un diagramma che visualizza gli elementi dell’eco-

sistema, ovvero l'open source hardware, l'open design, il DIY e le tecnologie di produzione digitalizzata. Il secondo diagramma mostra i livelli progettuali, ovvero il file, la documentazione multimediale, le piattaforme collaborative, i sistemi modulari e parametrici e le macchine di riproduzione.

Fig. 9 A sinistra: Digramma dell'Ecosistema Open Source. A destra: Diagramma dei livelli progettuali



## 2. Analisi di prodotti interattivi open source

La seconda attività del workshop si basa sull'analisi di prodotti interattivi open source esistenti. I prodotti sono i due casi studio descritti nel capitolo 3, Open Source Mirror di Digital Habits e Primo di Solid Labs. La scelta di questi prodotti è avvenuta sulla base dei seguenti criteri: i due progetti sono dispositivi dotati di interfaccia tangibile e naturale e i cui file sorgente sia hardware, software sia il design sono stati resi pubblici su internet tramite l'applicazione di una specifica strategia di diffusione della documentazione al fine di favorire la riproduzione da parte di altri. Entrambi i prodotti sono stati specificatamente progettati per essere riprodotti e riconfigurati a livello amatoriale, ovvero sono stati concepiti per soddisfare bisogni e intenzioni chiave della progettazione open source: l'elevata customizzabilità del design fisico (del prodotto), la riprogrammazione dei comportamenti interattivi e la riproducibilità tramite macchine di prototipazione rapida come, per esempio, macchine a taglio laser e stampanti 3D.

Obiettivo dell'analisi dei prodotti interattivi open source è fornire ai partecipanti le informazioni fondamentali sui componenti, sui meccanismi di funzionamento nonché sulle modalità di interazione del prodotto. I supporti utilizzati per spiegare le varie parti sono le immagini e i disegni tecnici pubblicati sui siti ufficiali degli autori dei prodotti open source e il prototipo dei prodotti.

### 3. Definizione dei bisogni tramite la selezione dell'attitudine e della motivazione dell'utente

Al fine di guidare la fase di ideazione del prodotto sono forniti due strumenti per la definizione dei bisogni dell'utente. Lo strumento "Attitudine" supporta l'individuazione di un profilo che descrive un possibile utente del prodotto derivato, dal professionista tecnico all'imprenditore. Lo strumento "Motivazione" descrive il tipo di motivazione alla base dell'interazione, riproduzione e riprogrammazione del prodotto-piattaforma.

### 4. Definizione e prototipazione "in bassa fedeltà" di un prodotto derivato

Al fine di documentare gli aspetti salienti dell'idea di prodotto derivato, ho definito una scheda di descrizione del progetto che fornisce le linee guida per la descrizione dell'idea di progetto. Dopo aver descritto l'idea, i partecipanti affrontano la fase finale

Fig. 10 La scheda Attitudine

ATTITUDINE	
<p><b>Maker</b></p> 	<p><b>Caratteristiche</b></p> <ul style="list-style-type: none"> <li>Ha conoscenze tecnologiche specialistiche</li> <li>Svilupa soluzioni "prototipando"</li> <li>Utilizza tutorial e servizi collaborativi per trovare soluzioni</li> <li>Collabora con comunità fisiche e digitali</li> <li>...</li> </ul> <p><a href="https://fic.kr/p/gwie7w">https://fic.kr/p/gwie7w</a></p>

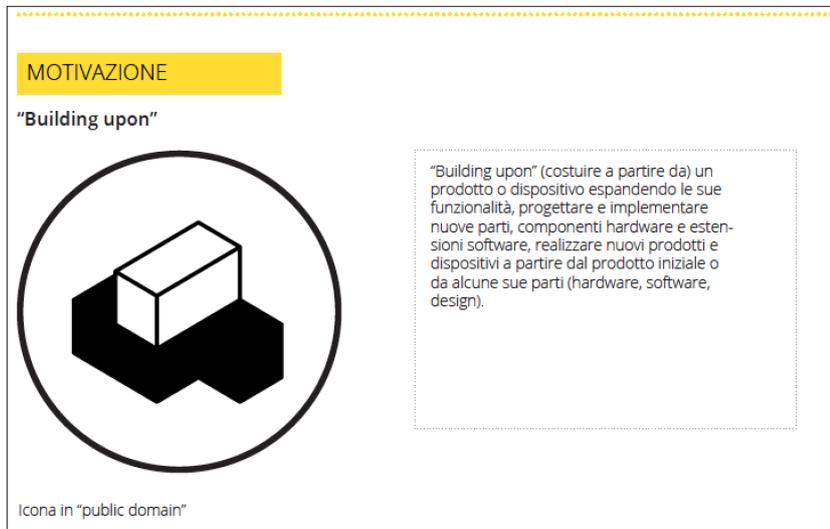


Fig. 11 La scheda Motivazione

del workshop che prevede lo sviluppo di prototipi in bassa fedeltà (*low fidelity*) del progetto di prodotto derivato. La prototipazione in bassa fedeltà può essere realizzata tramite tecniche diverse come il *paper prototyping* in 2D, il *paper prototyping* in 3D e il *Wizard of Oz*, ovvero la simulazione del comportamento interattivo (Buxton 2007). La fase di prototipazione ha l'obiettivo di dare forma all'idea di progetto e a testare le principali modalità di interazione.

#### 4.3.2 I risultati del workshop

Il workshop "Progettare prodotti-piattaforma" ha avuto luogo per la prima volta presso la sede dell'associazione WeMake, un laboratorio di produzione digitalizzata e makerspace attivo nell'organizzazione di eventi, attività e progetti di design e sviluppo open source. La sede di WeMake è in via Stefanardo da Vimercate n. 27 e i fondatori dello spazio sono Zoe Romano e Costantino Bongiorno<sup>2</sup>.

Al workshop hanno partecipato 12 persone con i seguenti profili:

1. Studenti di corsi bachelor di graphic design;
2. Studenti di corsi bachelor di interior design;

<sup>2</sup> Il workshop è stato promosso sul sito del makerspace WeMake con il titolo "Making things open" per esigenze di comunicazione (WeMake 2014)

3. Studenti di corsi master di service design;
4. Laureati in architettura;
5. Dottorandi in design;
6. Responsabili di laboratori fablab con formazione in design e architettura.

I partecipanti sono stati suddivisi in quattro gruppi di lavoro definiti **Gruppo A**, **Gruppo B**, **Gruppo C**, **Gruppo D**. Il workshop è durato 7 ore con due sessioni di lavoro. Nella prima sessione di lavoro, ho presentato gli obiettivi e il tema del workshop con una descrizione dettagliata dei casi studio, Open Mirror e Primo.

Fig. 12 Scheda di descrizione dell'idea di progetto

TEMPLATE 2 - PROGETTO		N. <input type="text"/>	Gruppo: <input type="text"/>
Nome del progetto		Sketch che spiega il funzionamento, le parti e le componenti principali	
Descrizione sintetica della caratteristica chiave del progetto			
Funzionalità e obiettivi			
Interfaccia e modalità d'interazione			
Tecnologie			

Making things open, 18.10.2014, WeMake

Successivamente, ho introdotto gli elementi del framework e gli strumenti di progettazione come la scheda di descrizione dell'attitudine e della motivazione dell'utente.

Nella seconda sessione di lavoro, i partecipanti hanno presentato il risultato delle loro attività volte a definire un utente tipo con una specifica attitudine e motivazione.

Nella terza sessione di lavoro, sono state sviluppate le idee di progetto che sono state documentate e poi presentate tramite l'utilizzo della scheda di descrizione del progetto (fig. 12). Nella fase finale del workshop, sono stati sviluppati i prototipi in bassa fedeltà dei prodotti utilizzando cartonici e disegni realizzati al computer.

Al termine del workshop sono state prodotte otto idee di progetto di prodotto derivato e quattro prototipi a bassa fedeltà. Quattro derivati sono stati sviluppati sulla base del prodotto Open Mirror, quattro derivati sulla base di Primo.

### Derivati di Open Mirror

#### 1. Track your mirror, Gruppo D

Il dispositivo "Track your mirror" aggrega e analizza le *playlist* musicali degli utilizzatori del prodotto. Il dispositivo è connesso a una piattaforma di analisi delle *playlist* musicali e monitora il tipo di musica e i comportamenti degli utenti in base anche a dove è posizionato lo specchio all'interno degli spazi domestici. Il dispositivo è stato pensato per un utente con l'attitudine "ricercatore", un'attitudine aggiunta dai partecipanti per definire un tipo di utente che punta a interagire con un prodotto solo per scopi di ricerca scientifica, accademica o professionale. La motivazione associata al prodotto è "Espandere" poiché esso aggiunge al progetto Open Mirror una piattaforma digitale per l'aggregazione delle tracce musicali ascoltate dagli utenti.

#### 2. Music composition, Gruppo D

Il dispositivo "Music composition" è composto da moduli in vetro e tessuto che incorporano i componenti elettronici dello specchio. Open Mirror, ovvero gli altoparlanti, i sensori di riconoscimento

Fig. 13 Sketch di Track your mirror

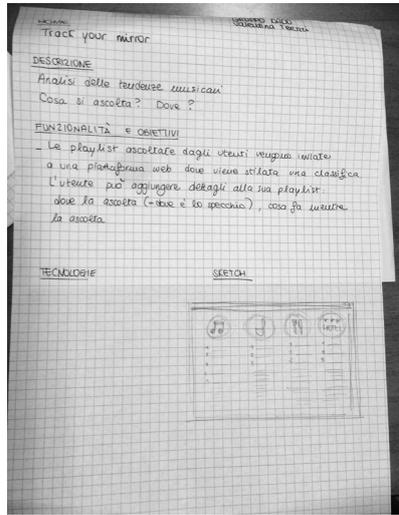
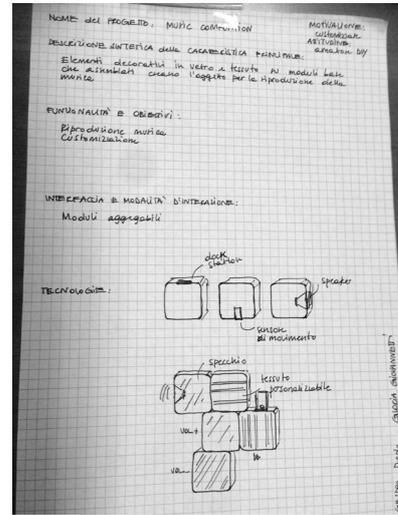


Fig. 14 Sketch di Music composition



dei gesti. I moduli sono aggregabili e possono essere posizionati con configurazioni diverse per permettere all'utente addattare il prodotto in base all'ambiente. Il dispositivo è stato pensato per un utente con attitudine "Amatore DIY" che interagisce con le informazioni del dispositivo al fine di customizzare, in questo caso, l'ambiente in cui sarà installato.

### 3. Side-Cook, Gruppo B

Il dispositivo "Side cook" consiste in uno schermo e un'interfaccia gestuale e a controllo vocale. Il dispositivo funziona da assistente per la cucina e da dispositivo di pubblicazione automatica delle foto dei piatti cucinati dagli utenti su un blog on-line. Tramite il controllo gestuale, gli utenti possono scorrere i contenuti della guida e con il controllo vocale interagire con l'assistente elettronico che suggerisce di volta in volta le ricette. L'utente del prodotto è il maker che è interessato a customizzare la forma dello specchio per installarlo in una cucina, a espandere le funzionalità introducendo il comando vocale e a costruire qualcosa di nuovo cambiando l'ambito di applicazione del progetto Open mirror.

### 4. Feel, Gruppo A

"Feel" è la versione espansa dello specchio Open Mirror. Consiste in sensori modulari che possono essere collocati in casa per controllare in maniera remota le funzionalità dello specchio (p.e.

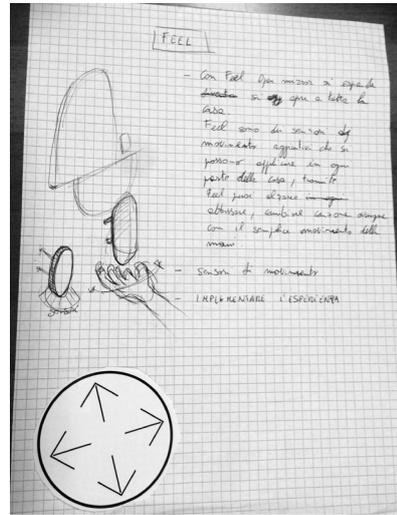
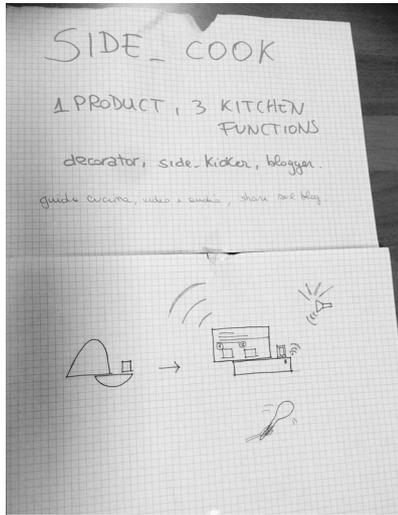


Fig. 15 Sketch di Side-Cook

Fig. 16 Sketch di Feel

il controllo del volume, il cambio della traccia). Il prodotto è pensato per un utente consumatore e punta a espandere Open Mirror attraverso la messa in comunicazione del prodotto con un ulteriore dispositivo di controllo.

### Derivati di Primo

#### 5. MyCubetto, Gruppo A

“MyCubetto” è un’applicazione web per creare derivati di Cubetto. L’applicazione permette di modificare le dimensioni e l’aspetto di Cubetto e di trovare il laboratorio di produzione più vicino per produrlo. L’applicazione è destinata a utenti maker che progettano e producono in modo fai da te i derivati di Primo customizzando il prodotto in base a diverse necessità.

#### 6. Musichetto, Gruppo C

“Musichetto” è un dispositivo che informa l’utente sul grado di precisione nell’eseguire e riprodurre una composizione musicale. Il Musichetto è un derivato di Cubetto ruota e mostra il lato positivo o negativo in base all’input sonoro rilevato da una tastiera o pianoforte. L’utente tipo non rientra nei profili del framework. Il progetto si basa sul costruire un nuovo dispositivo a partire da Primo per creare un’applicazione nell’ambito dell’insegnamento della musica.

Fig. 17 Sketch di MyCubetto

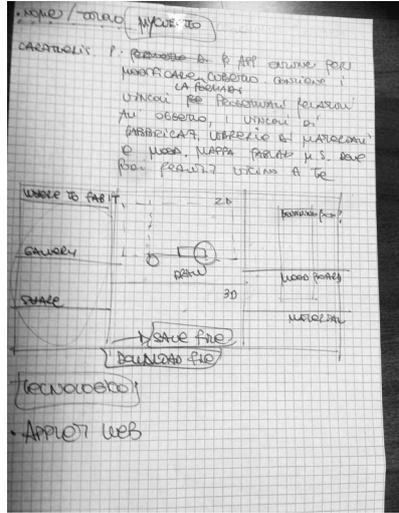


Fig. 18 Sketch di Musichetto



### 7. Prima di Primo, Gruppo D

“Prima di Primo” è un dispositivo che serve a stimolare le facoltà motorie dei neonati. Consiste in tappeto che rileva la presenza del bambino ed è connesso a Cubetto che si muove in base alla posizione assunta dal neonato. L’utente non è stato definito poiché l’applicazione è destinata ai bambini. Il progetto si basa sulla motivazione “building upon”, ovvero sulla modifica del dispositivo di input di Primo che è un tappeto e non una *board* con i moduli di movimento.

### 8. Cubetto Community Edition, Gruppo A

“Cubetto Community Edition” è una versione di Primo che permette di programmare più Cubetti contemporaneamente. L’interfaccia tangibile di Primo (*board*) include maggiori *instruction blocks* e i Cubetti possono interagire tra di loro. L’utente tipo ha

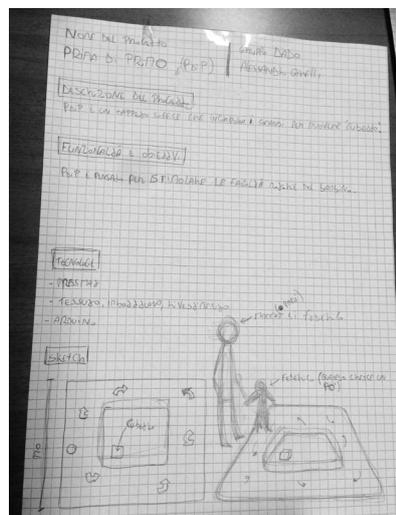


Fig. 19 Sketch di Prima di Primo

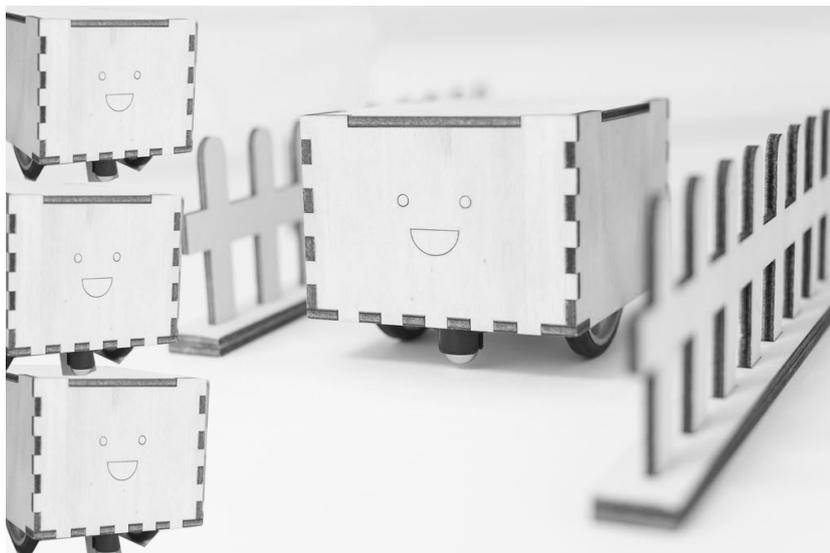


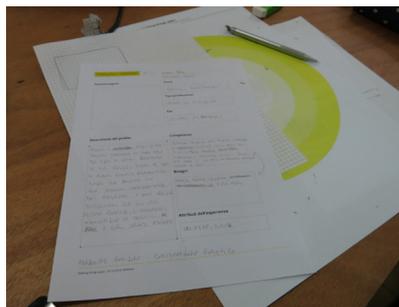
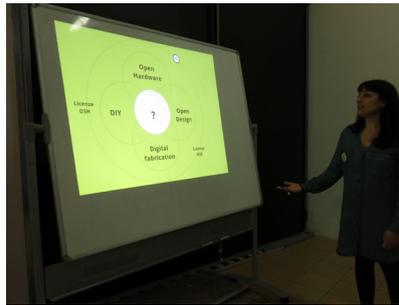
Fig. 20 Sketch di Cubetto Community Edition

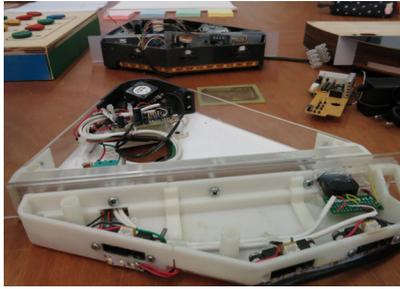
un'attitudine "maker" e la motivazione si basa sull'espandere il progetto esistente Primo con nuove parti fisiche e comportamenti interattivi.

#### 4.3.3 Alcune riflessioni sul workshop

Il workshop "Progettare prodotti-piattaforma" ha permesso di valutare il framework di progettazione e la sua applicazione nel con-

Fig. 21 Foto del workshop Progettare prodotti-piattaforma presso il makerspace WeMake.





testo della didattica. Innanzitutto, emergono alcune debolezze che sono relative, per esempio, alla durata del workshop: la complessità dell'argomento e degli strumenti richiede un formato didattico diverso che dovrebbe essere più simile a un workshop intensivo di più giorni o ad un corso a tempo parziale distribuito su più settimane.

In secondo luogo, la combinazione tra le attività d'ideazione e di prototipazione richiede maggiore impegno da parte dei partecipanti e, per questo motivo, le attività di sviluppo dei derivati diventa prioritario rispetto al lavoro sulla creazione e condivisione della documentazione di progetto. In un solo caso, il MyCubetto, l'attenzione è stata posta sull'ideazione di un'interfaccia che facilitasse la customizzazione e distribuzione di un derivato del prodotto Primo. Gli altri progetti sono stati ideati con l'obiettivo di espandere il prodotto originale e sfruttare il design e le soluzioni tecnologie esistenti.

Emerge la difficoltà di combinare le attività di progettazione con le attività di documentazione: questo dipende principalmente dalla volontà di valutare le opportunità di una progettazione basata su prodotti interattivi esistenti piuttosto che la necessità di documentare i prodotti derivati. In questo senso, il workshop

ha permesso di valutare la generatività e la programmabilità dei prodotti interattivi originali, Primo e Open Mirror. L'aspetto della riproducibilità è rimasto in secondo piano poiché nel workshop la prototipazione si basava su tecniche "a bassa fedeltà" (utilizzo di cartoncini e *mock-up* digitali) e non ha richiesto l'utilizzo di macchine di produzione digitalizzata.

## 5. Conclusioni: i punti critici nella progettazione di artefatti interattivi open source

La ricerca sulle pratiche di progettazione e sviluppo open source in relazione a nuovi modi e processi di progettare artefatti interattivi mi ha portato ad affrontare una serie di problematiche di carattere principalmente concettuale e metodologico.

Dall'analisi della letteratura esistente, dei casi studio e delle pratiche correnti si può affermare che la relazione tra open source e design sia ancora in una fase di transizione: la scelta di focalizzare l'attenzione sugli artefatti interattivi ha permesso, infatti, di poter inglobare nella ricerca molti temi e pratiche che seppur connesse tra di loro si basano su modelli e principi diversi come l'open source hardware e l'ambito della produzione digitalizzata. Open source hardware, open design e design dell'interazione vanno a delimitare un territorio multidisciplinare basato sulla sovrapposizione di discipline tecniche, discipline del progetto e aspetti e metodi dell'ambito del business e dell'economia. Considerata questa multidisciplinarietà, la ricerca punta a offrire una prospettiva sul mondo dello sviluppo open source dal punto di vista degli approcci del design e in particolare della progettazione dell'esperienza utente. Inoltre, la ricerca è un tentativo di definizione di metodi e strumenti di progettazione che aiutano il designer a comprendere e agire nella complessità sottostante le emergenti pratiche legate all'open source.

Ho cercato di affrontare le problematiche in maniera olistica e programmatica attraverso la definizione di quattro strumenti: l'Ecosistema open source, i livelli progettuali nel contesto open source, il framework "Progettare prodotti-piattaforma" e il workshop pratico "Progettare prodotti-piattaforma". Seppur i primi due strumenti siano funzionali alla definizione del framework di

progettazione, considero anch'essi dei risultati della ricerca perché possono essere utilizzati come strumenti a servizio dell'attività di progettazione di un artefatto interattivo. Il processo di definizione di questi strumenti è solo all'inizio.

Nel percorso di ricerca ho cercato di metter in luce alcuni punti critici dell'applicazione del modello open source al dominio della progettazione. La modularità, intesa come creazione di sistemi di interfacciamento tra dispositivi che comunicano tra di loro, è uno di questi: da un lato è l'aspetto che collega la progettazione open source ad una tipica attività del design per la produzione in serie (p.e. parti assemblabili per la catena di montaggio) e a un tipo di design per tutti (p.e. i mobili "nomadici" di Papanek); dall'altro è un punto di debolezza quando si trasferisce l'attività di sviluppo nel mondo tecnologico che evolve e in cui il cuore di un prodotto è il servizio web che mette in comunicazione dispositivi diversi (p.e. il servizio Temboo). La modularità in questo senso non può essere implementata se non a discapito del livello di "openness" del progetto che si oscura e diventa proprietario proprio nelle sue parti chiave, ovvero i giunti e le interfacce digitali.

La progettazione di prodotti-piattaforma si basa sulla selezione o progettazione di parti e soluzioni standard che facilitano il processo di riproduzione di un prodotto. Questo requisito rappresenta un limite alla possibile esplorazione su nuovi modi di configurazione di un dispositivo. Per affrontare questo limite, il designer può concentrare la sua attività sulla **definizione di sistemi-prodotto complessi** piuttosto che innovare dal punto di vista del design del prodotto. Questo limite, inoltre, sposta il baricentro delle attività del designer sulla progettazione di artefatti digitali che fungono da librerie di parti (p.e. i connettori tra componenti modulari). La creazione di librerie di parti customizzabili permette di moltiplicare la varietà di soluzioni derivate e di evitare, soprattutto, un effetto di omogeneizzazione dal punto di vista estetico e dell'innovazione.

Le strategie per facilitare la partecipazione e collaborazione da parte di una comunità di persone in rete richiede **nuove competenze di comunicazione**. Le principali iniziative e aziende che fondano il loro successo sul modello open source beneficiano del contributo di una grande comunità di utenti che partecipano

attraverso la produzione di risorse e documenti a supporto dello sviluppo di progetti tecnologici. La scalabilità di un prodotto interattivo open source dipende, quindi, dallo sviluppo di competenze di comunicazione all'interno di un gruppo di lavoro e questo porta all'emergere di figure professionali che lavorano principalmente alla promozione della partecipazione finalizzata allo sviluppo del prodotto da parte di altri. Il rischio dell'applicazione di questo tipo di competenze specialistiche è l'utilizzo di strategie classiche di marketing e comunicazione che portano la disseminazione di un progetto open source lontano da una necessaria relazione empatica e trasparente con i membri della comunità che collabora allo sviluppo del progetto. La relazione empatica si basa su uno scambio di contenuti riguardo al progetto e su una interazione finalizzata alla co-progettazione e allo sviluppo collaborativo.

Un altro punto critico riguarda il rilascio dei file sorgente del progetto e la pubblicazione della documentazione in rete. Un problema aperto è, infatti, la scelta tra la creazione di applicazioni digitali web ad hoc per **la disseminazione della documentazione** del progetto e l'utilizzo di piattaforme di aziende terze come supportano lo sviluppo collaborativo (p.e. GitHub). Favorire l'accessibilità di un progetto a una comunità estesa di utenti vuol dire condividere gli stessi strumenti e buone pratiche di documentazione. Le iniziative di definizione di standard di documentazione sono recenti e sono state avviate nell'ambito della progettazione e sviluppo di open source hardware (Open Source Hardware Documentation Jam 2014). Nel contesto dell'open design il discorso delle buone pratiche di documentazione è ancora aperto e vede il proliferare di pratiche e strumenti diversificati (p.e. Wikihouse).

Il processo di semplificazione dei dispositivi per facilitare l'utilizzo e la partecipazione da parte di una comunità estesa di persone (da esperti a non esperti) crea dei **livelli di opacità nei progetti open source** poiché la volontà di coinvolgere persone non esperte e la necessità di raggiungere un successo di tipo economico sposta il baricentro dell'"openness" del progetto sugli aspetti di accessibilità e usabilità piuttosto che sulla trasparenza dei componenti e le soluzioni tecnologiche e di design. Il designer che opera nel contesto open source deve confrontarsi con questo problema e sviluppare soluzioni che cercano di bilanciare gli

obiettivi di semplificazione con la natura open source tecnologica e di design del prodotto. Dovrà per questo rivolgersi a pubblici differenziati e rilasciare la documentazione e gli strumenti di co-progettazione sia per utenti esperti e sia non esperti.

Il successo di un prodotto interattivo open source dipende dalla definizione di un modello di business che sia profittevole e allo stesso tempo basato su logiche di apertura e condivisione con i concorrenti dello stesso settore di mercato. Il punto critico nella progettazione open è, quindi **la definizione di modelli innovativi di business** piuttosto che l'innovazione di prodotto e di processo.

Affrontare la progettazione di artefatti open source dal punto di vista della definizione di un ecosistema di elementi così come definiti nel *framework* è un'attività complessa che influenza la realizzazione di un prodotto-piattaforma completo. Il focus dell'attività di progetto si sposta **dalla configurazione di prodotti open source alla configurazione di servizi open source**. L'attività del designer nel momento di progettazione di un artefatto interattivo dovrà tenere in conto gli aspetti legati alla definizione di un servizio che facilita l'interazione con la comunità on-line e rende accessibile in maniera usabile la documentazione. Le interfacce e le modalità di interazione di questi servizi sono considerate accessorie nella definizione di un progetto open source, ma dovranno sempre di più essere incluse nel processo di progettazione per attualizzare il paradigma di un design "aperto". A partire da questo assunto, la progettazione di prodotti interattivi open source si estende alla progettazione di servizi i cui *blueprint* documentano l'interazione tra designer, aziende, persone, applicazioni digitali, artefatti fisici e macchine di produzione.

## 6. Apparati

Intervista a Diego Rossi e Innocenzo Rifino  
Studio Habits

Innocenzo Rifino e Diego Rossi sono i fondatori dello studio di design Habits, basato a Milano. Dal 2011 sviluppano un progetto di autoproduzione di una linea di oggetti interattivi che esplorano la possibilità di combinare le interfacce naturali basate sul gesto e l'esperienza di ascolto della musica. Il primo progetto della linea Digital habits è l'Open Mirror, uno specchio che funziona come dockstation e uno speaker per controllare e ascoltare le playlist musicali di uno smartphone tramite un'interfaccia gestuale. Lo specchio è stato rilasciato in open source ed è riproducibile tramite macchine di prototipazione rapida come macchine a taglio laser e stampanti 3D ([www.digitalhabits.it](http://www.digitalhabits.it), [www.habits.it](http://www.habits.it))

**Serena Cangiano: Come nasce il progetto Open Mirror?**

**Innocenzo Rifino:** Abbiamo iniziato a progettare e sviluppare l'Open Mirror per una reazione alla mancata capacità di un'azienda che era nostra committente di avviare la produzione di prodotto interattivo che contenesse della tecnologia e dell'elettronica al suo interno. Un'azienda, infatti, ci aveva chiesto di sviluppare uno specchio interattivo e pensava di poter affrontare il progetto dal punto di vista economico come fosse un normale prodotto, uno specchio.

Uno specchio classico si può progettare tramite uno schizzo o un disegno quotato, mentre un prodotto interattivo ha bisogno di essere prototipato e subire diverse iterazioni di sviluppo pri-

ma di arrivare alla sua forma finale. Lo specchio, quindi, nasce dall'impossibilità per noi di avere committenza per lo sviluppo di prodotti interattivi da parte di aziende che si occupano di design del prodotto, le quali non sono pronte a questo sforzo creativo, professionale e economico, ad affrontare la complessità della produzione di un dispositivo tecnologico

A quel punto ci siamo trovati di fronte alla questione se abbandonare o continuare ad andare avanti da soli.

**Diego Rossi:** il progetto inizia con una sperimentazione. Non avevamo un'idea chiara di quello che avremmo creato. Si trattava di esplorare l'ambito dell'autoproduzione per raggiungere una qualità di prodotto simile o quasi simile a quella industriale.

Gli oggetti autoprodotti non possono raggiungere la qualità di un prodotto acquistato. C'è sempre un gap tra il prodotto che puoi costruire in casa e quello che puoi comprare. Anche se realizzare prodotti in modo fai da te è diventato molto facile, c'è pur sempre un aspetto importante legato all'economia di scala.

Open Mirror puoi costruirlo, ma costa molto di più che acquistarlo perché il costo dei materiali di base delle macchine di digital fabrication è ancora molto alto.

Si farebbe presto a dire che la versione industriale verrebbe a costare meno, ma poi la consulenza dell'azienda che prepara lo stampo ad iniezione delle parti costa molto...

Abbiamo iniziato con una sperimentazione, ma la possibilità di autoprodurli in piccola serie è arrivata successivamente, dopo aver testato un processo di "industrializzazione" che fa sì che l'Open Mirror venga percepito come un buon prodotto e acquistato per questo e non perché sia open source.

**IR:** Abbiamo rilasciato il progetto Open Mirror in open source prima di tutto per fornire un ritorno di conoscenza che avevamo preso in giro dalla rete, specialmente per la parte di gestione della sensoristica e dell'elettronica. Poi, riguardo all'innovazione di prodotto, credo che la vera opportunità e innovazione dell'open source sia nel progetto e non nel prodotto. Tutto parte dalla possibilità di utilizzare la scheda Arduino senza il quale non avremmo mai potuto capire come realizzare un prodotto di questo tipo. La

vera opportunità è di avere accesso al software open source e creare un derivato della scheda Arduino per creare un nuovo tipo di prodotto. Se non ci fosse stato Arduino e la comunità di Arduino non avremmo mai potuto capire come realizzare un tipo di prodotto come l'Open Mirror perché avremmo dovuto acquistare le competenze per farlo o impiegare molto tempo per acquisire queste competenze.

Grazie all'open source è facile trovare, imparare e applicare conoscenze per realizzare un prodotto interattivo.

**SC: Quali sono gli aspetti progettuali dietro alla realizzazione dell'Open Mirror?**

I nostri oggetti dal punto di vista del comportamento sono come un oggetto della Sony. Come standard qualitativo, sia della parte acustica sia della parte dei sensori, sono molto simili a quello di un prodotto industriale.

E' la parte hardware (del corpo e della meccanica dell'oggetto) che è qualitativamente meno stabile perché le parti sono state realizzate con tecnologie normalmente usate per prototipare, le tecnologie di digital fabrication. Queste tecnologie ci hanno permesso di realizzare l'oggetto, ma applicando degli espedienti costruttivi ed estetici. Senza questi artifici (p.e. coprire l'oggetto con uno specchio normale) si rende evidente che il livello meccanico e fisico non è uguale a quello di un prodotto realizzato industrialmente.

**DR:** Il gap è principalmente il costo e la qualità. Il livello di finitura del laser è un po' basso rispetto ai prodotti che si acquistano sul mercato. Abbiamo utilizzato delle tecnologie di produzione di massa, come lo specchio e il carrello in metallo, e a questi abbiamo combinato una parte realizzata con una macchina a taglio per fare in modo che la percezione dell'acquirente finale fosse quella di un oggetto finito.

Avremmo potuto disegnare un oggetto bellissimo, ma il limite che ci era imposto dalle macchine a disposizione. Charamente, le tecnologie di digital fabrication si stanno evolvendo verso una maggiore raffinatezza del risultato, ma l'Open Mirror

nasce nella sua forma attuale proprio per verificare il limite delle tecnologie di produzione che avevamo a disposizione.

**IR:** Avviare la progettazione e produzione di oggetti come Open Mirror riguarda un ritorno in conoscenza. Non solo dal punto di vista tecnico, ma anche dal punto di vista della progettazione, abbiamo potuto fare ricerca e esperienza del funzionamento delle interazioni gestuali. C'è quindi una ricaduta di estrinseca sul piano delle conoscenze e competenze acquisite.

A questa si aggiunge una ricaduta in comunicazione, ovvero le grandi aziende ci contattano perché con lo specchio abbiamo potuto comunicare il livello di padronanza di aspetti tecnici e progettuali che non avremmo mai potuto condensare in un progetto commissionato da un cliente. La committenza esterna limita spesso le possibilità progettuali che sono supportate dall'uso della tecnologia.

Infine, c'è una ricaduta intrinseca perché l'Open Mirror, non so se dipende da come sono concepiti o costruiti, hanno degli acquirenti. Questo non credo che dipenda dal fatto che sono realizzati con tecniche di digital fabrication o perché siano open source, ma perché il prodotto visualizza degli aspetti inesplorati nel design del prodotto e, quindi, gli acquirenti sono interessati ad avere accesso a qualcosa di totalmente nuovo. Sono piacevolmente colpiti ed è per questo che vogliono comprarlo.

### **SC: Sapete chi acquista e se sono stati creati dei derivati?**

**IR:** Open Mirror è stato chiesto da un produttore di specchi. Riguardo alla creazione di derivati, è la Cromatica, lo speaker-lampada interattiva, che ci è stata chiesta per realizzare delle versioni hackerate, per esempio, associare il colore della lampada alla musica d'ambiente.

L'Open Mirror ci è stato chiesto da uno studente di informatica per fare degli esperimenti con l'integrazione di una webcam per lo sviluppo di un dispositivo medicale. Lo specchio gli serviva per coprire quegli aspetti di design che lui non avrebbe potuto progettare.

**DR:** Noi assembliamo lo specchio a mano e questo fa sì che il pro-

dotto sia altamente riproducibile da parte degli altri. Non abbiamo mai progettualmente affrontato la questione di come facilitare la riproduzione da parte degli altri perché è implicita nel processo di produzione stessa del prodotto.

**In ogni caso, Open Mirror è difficile da riprodurre a casa.**

Se gli oggetti sono riprodotti in unico esemplare, c'è bisogno di molto tempo per produrli. La nostra è una produzione in piccola serie e riusciamo ad assemblarne circa una decina al giorno perché la parte elettronica e alcuni pezzi, come per esempio il carrello metallico, sono già pronti. Abbiamo raggiunto un equilibrio, tra numero di pezzi e tecnologia produttiva. Se il numero di pezzi dovesse aumentare, sarebbe necessario utilizzare una diversa tecnologia di produzione.

**DR:** I nostri oggetti sono nati dal *mesh-up* di tecnologie esistenti rilasciate in open source gratuitamente. Siamo in debito con altri sviluppatori perché la soluzione alternativa all'open source sarebbe stato pagare quella conoscenza.

In termini di sviluppi ulteriori da parte di altri, non abbiamo invece un gran riscontro.

Siamo noi internamente allo studio che sviluppiamo maggiormente il prodotto:

la nuova versione dello specchio è migliore, funziona con il bluetooth e la parte acustica è anche stata migliorata. C'è sempre un processo di aggiornamento.

Abbiamo raffinato la tecnica della sovrapposizione degli strati di legno. Abbiamo semplificato l'assemblaggio. Abbiamo reso più solida la parte di cinematismo.

## Intervista a Matteo Loglio e Filippo Jacob Solid Labs

Primo è un'interfaccia tangibile progettata per spiegare la logica della programmazione a bambini dai 3 ai 7 anni, ovvero nell'età precedente all'alfabetismo. Il progetto è sviluppato dallo studio Solid Labs fondato da Filippo Yacob, Matteo Loglio, Valeria Leonardi. In seguito alla campagna Kickstarter conclusa il 22 dicembre 2013, il progetto è stato supportato da fondi per l'avviamento di aziende start-up.

### **Serena Cangiano: Da dove nasce l'idea di un giocattolo come Primo?**

**Matteo Loglio:** Primo nasce come progetto master alla SUPSI in un corso. L'obiettivo del corso era di progettare artefatti open source che potessero essere customizzati e riproducibili tramite tecnologie di produzione digitalizzata. Il progetto si inserisce nella ricerca sull'insegnamento della programmazione ai bambini che ha riferimenti nel lavoro di Seymour Papert. L'obiettivo era di progettare soluzioni che non sono basate su interfacce grafiche o sull'utilizzo di schermi o dispositivi mobili.

### **SC. Quali sono gli elementi di innovazione che avete voluto introdurre rispetto all'introduzione del rilascio open source?**

**Filippo Jacob:** Primo un'esperienza più inclusiva. Per fare riferimento al mondo dei giocattoli, le industrie tradizionali di giocat-

toli creano dei giocattoli *black-box*: un solo giocattolo, per un solo uso, che poi è buttato via quando i bambini crescono. Sviluppando un progetto in open source, dai la possibilità alle persone di alterare il prodotto, farlo in modo proprio.

Per esempio un nostro collega a Londra è un insegnante per le scuole speciali, insegna a ragazzi con autismo o altre disabilità da 8 ai 12. Al fine di migliorare il prodotto per adattarlo a questo tipo di ragazzi ha deciso di aggiungere dei suoni al Cubetto. In un pomeriggio abbiamo implementato questa funzionalità che ha reso possibile di rendere questa esperienza con il prodotto più adatta alle sue necessità perché abbiamo applicato un approccio open source.

**SC: Cosa vuol dire?**

ML: vuol dire che il nostro amico avrebbe potuto eventualmente modificare il Cubetto da solo accedendo alla documentazione in rete senza chiederci il permesso. Tutti possono cambiare il prodotto e non devono usarlo nel modo che noi abbiamo immaginato o progettato. Questa è un aspetto interessante.

**SC: Come gestite il contributo degli utenti o di altri sviluppatori?**

ML: Non riceviamo tanti feedback dalle persone. Due persone hanno creato dei derivati di Primo.

Una persona dalla Colombia per esempio ha costruito in modo fai da te Primo e ci ha mandato delle informazioni per migliorare i tutorial che abbiamo creato a supporto dell'assemblaggio. Altre persone, per esempio, ci aiutano a tradurre la documentazione in altre lingue.

**SC: Tutto questo avviene in maniera spontanea?**

ML: in realtà bisogna facilitare questo processo di partecipazione da parte delle persone.

Non sappiamo esattamente cosa le persone vogliono fare. Forse vogliono solo riprodurlo o migliorarlo. Noi forniamo delle istru-

zioni per partecipare allo sviluppo dei derivati, oppure cerchiamo di spiegare come organizzare i file in modo da creare una cartella strutturata di risorse disponibile in GitHub.

**SC: Credete che ci sia un grande ritorno nell'applicare questo modello open?**

In realtà sì. Le persone possono realizzare Primo oppure possono decidere di comprarlo, ma nel frattempo dopo averlo fatto parlano del nostro prodotto con altre persone. In questo modo anche altre persone verranno in contatto con il prodotto e con le modalità aperte con il quale è stato progettato.

## 7. Bibliografia

- AA.VV., *The third industrial revolution*, Special Report in *The Economist*, 21 aprile 2012, pp. 3-17.
- Adafruit Learning system, [learn.adafruit.com](http://learn.adafruit.com) (ultima visita 11 novembre 2014).
- Alizon, S. B. Shooter, T. W. Simpson, *Henry Ford and the Model T: lessons for product platforming and mass customization*, in *Design Studies*, vol. 30, n. 5, 2009, pp. 588-605
- Anceschi G., *Miriorama: un ritorno all'arte moltiplicata*, in *ViewOn. Oggetti e progetti. Alessi. Storia a e futuro di una fabbrica di design italiano*, 2010, p. 26.
- Apache Subversion, [www.subversion.apache.org](http://www.subversion.apache.org) (ultima visita 20 novembre 2014).
- Arduino Lilypad, [lilypadarduino.org](http://lilypadarduino.org) (ultima visita 20 novembre 2014).
- Arduino playground, [playground.arduino.cc](http://playground.arduino.cc) (ultima visita 27 novembre 2014).
- Arduino store, [store.arduino.cc](http://store.arduino.cc) (ultima visita 20 dicembre 2014).
- Argan C.G., *Testimonianze sulla XXXII Biennale*, in *Il Ponte*, n. 8, 1964.
- Art meets radical openness, festival dedicated to art, hacktivism and open culture*, [radical-openness.org](http://radical-openness.org) (ultima visita 26 novembre 2014).
- Arte programmata, arte cinetica, opere moltiplicate, opera aperta*, Officina d'Arte Grafica A. Lucini, Milano 1962.
- Atkinson P., *Do it yourself, democracy and design*, in *Journal of Design History*, n. 19, 2006, pp. 1-10.
- Atkison P., *Orchestral manœuvres in design*, in Van Abel B. et al, a cura di, *Open Design Now – Why design cannot remain*

- exclusive*, BIS Publishers, Amsterdam, 2012. pp. 24-31.
- Avital M., *The generative bedrock of open design*, in Van Abel B. et al, a cura di, *Open Design Now – Why design cannot remain exclusive*, BIS Publishers, Amsterdam, 2011, pp. 48-58.
- Baafi E., *Standardization Open Source Hardware Can Benefit From*, in Gibbs A., *Building Open Source Hardware – Manufacturing for the DIY Maker Movement*, Addison-Wesley Professional, Indianapolis, 2014.
- Baldwin C., von Hippel E., *Modeling a Paradigm Shift: From Producer Innovation to User and Open Collaborative Innovation*, in *Organization science*, vol. 22, n. 6, 2010, pp. 1399-1417.
- Ballo G., *La mano e la macchina. Dalla serialità artigianale ai multipli*, Sperling & Kupfer, Milano 1976, p. 195.
- Banzi M., *Opensourcing imagination*, in Botta M., Wiedmer M., a cura di, *Disruptive interaction*, Lugano, Edizioni Casagrande 2014, pp. 91-94
- Banzi M., *People over Megahertz*, in *MAKE*, Maker Media, O'Reilly Media, Sebastopol, CA, vol. 36, 2014, pp. 12-14.
- Bauwens M., *The Emergence of Open Design and Open Manufacturing*, in *We-Magazine*, vol. 2, [www.we-magazine.net/we-volume-02/the-emergence-of-open-design-and-open-manufacturing](http://www.we-magazine.net/we-volume-02/the-emergence-of-open-design-and-open-manufacturing) (ultima visita 26 novembre 2014).
- Bohnacker H., *Generative Design: Visualize, Program, and Create with Processing*, Princeton Architectural Press, New York 2012.
- Botta M. Cangiano S., *O come open design*, in *Ottagono*, n. 266, 2014, pp. 80-89.
- Boyd D.M., Ellison N.B, *Social network sites: Definition, history, and scholarship*, in *Journal of Computer-Mediated Communication*, n. 13, 2007, [jcmc.indiana.edu/vol13/issue1/boyd.ellison.html](http://jcmc.indiana.edu/vol13/issue1/boyd.ellison.html) (ultima visita 20 novembre 2014).
- Brocklin D., *The Cornucopia of the Commons: How to get volunteer labor*, 12 ottobre 2006, [www.bricklin.com/cornucopia.htm](http://www.bricklin.com/cornucopia.htm) (ultima visita 11 novembre 2014).
- Brown A., *Communicating design: developing web site documentation for design and planning*, New Riders, Berkeley, California, 2011.
- Buechley L., Rosner D., Daniela K., Paulos E., Amanda W., *DIY for*

*CHI: methods, communities, and values of reuse and customization*, in *Atti della conferenza ACM CHI 2009 Conference on Human Factors in Computing Systems 2009*, Boston, Massachusetts, USA Workshop, 5 aprile, 2009, pp. 4823-4826.

Buxton B., *On Engineering and Design: An Open Letter*, in *Business Week*, 10 settembre 2013, [www.businessweek.com/innovate/content/apr2009/id20090429\\_083139.htm](http://www.businessweek.com/innovate/content/apr2009/id20090429_083139.htm) (ultima visita 20 novembre 2014).

Buxton B., *Sketching User Experiences: Getting the Design Right and the Right Design*, Morgan Kaufmann, San Francisco, 2007, pp. 239-244.

Cangiano S. Fornari D., *Products as platforms: a framework for designing open source interactive artifacts*, in *Atti della conferenza DIS Companion '14 Proceedings of the 2014 companion publication on Designing interactive systems*, ACM New York, NY, USA, 2014, pp. 219-222.

Cardinal D., *Changing the world: DARPA's top inventions*, in *Extreme Tech*, 16 novembre 2011, [www.extremetech.com/extreme/105117-inventing-our-world-darpas-top-inventions/5](http://www.extremetech.com/extreme/105117-inventing-our-world-darpas-top-inventions/5) (ultima visita 20 novembre 2014).

Cavalcanti G., *Is it a Hackerspace, Makerspace, TechShop, or FabLab?*, 22 maggio 2013, [www.makezine.com/2013/05/22/the-difference-between-hackerspaces-makerspaces-techshops-and-fablabs/](http://www.makezine.com/2013/05/22/the-difference-between-hackerspaces-makerspaces-techshops-and-fablabs/) (ultima visita 26 novembre 2014).

Charny D., *About Making, The Power of Making*, mostra del V&A South Kensington, 6 settembre 2011 – 2 gennaio 2012, in in Victoria & Albert, [www.vam.ac.uk/content/articles/p/powerofmaking/](http://www.vam.ac.uk/content/articles/p/powerofmaking/) (ultima visita 11 novembre 2014).

CHDK Wikia, *Canon Hack Development kit*, in *CHDK Wikia*, [www.chdk.wikia.com](http://www.chdk.wikia.com) (ultima visita 20 novembre 2014).

Chiappini C., Rigamonti P., *Interaction Design*, in *Enciclopedia italiana XXI Secolo. Gli spazi e le arti*, Treccani, Roma 2010.

CIID, *Barcode piano*, in *CIID*, [www.ciid.dk/education/portfolio/idp11/courses/physical-computing/projects/barcode-piano](http://www.ciid.dk/education/portfolio/idp11/courses/physical-computing/projects/barcode-piano) (ultima visita 20 novembre 2014).

Circuit.io, [circuit.io](http://circuit.io) (ultima visita 20 novembre 2014).

Computer History Museum, *IBM System/360*, in *Computer Histo-*

- ry Museum, [www.computerhistory.org/revolution/mainframe-computers/7/161](http://www.computerhistory.org/revolution/mainframe-computers/7/161) (ultima visita 20 novembre 2014).
- Cooper A., *The Inmates Are Running the Asylum: Why High Tech Products Drive Us Crazy and How to Restore the Sanity*, Sams - Pearson Education, Indianapolis, 2004.
- Core jr, *Teague presents Prototype as product: 13.30 3D printed headphones*, in *Core77*, 2 ottobre 2012, [www.core77.com/blog/digital-fabrication/teague-presents-prototype-as-product-1330-3d-printed-headphones-23565.asp](http://www.core77.com/blog/digital-fabrication/teague-presents-prototype-as-product-1330-3d-printed-headphones-23565.asp) (ultima visita 26 novembre 2014).
- Creative Commons, *CC+: Creative Commons and Commerce*, in *Creative Commons*, [www.wiki.creativecommons.org/CCPlus](http://www.wiki.creativecommons.org/CCPlus) (ultima visita 26 novembre 2014).
- Cuartielles D., *Oh\_Oh robot*, in *David Cuartielles*, [www.david.cuartielles.com/w/Maquila2/Ohoh](http://www.david.cuartielles.com/w/Maquila2/Ohoh) (ultima visita 20 novembre 2014).
- Cuartielles D., *Arduino As Enabler of Open Design: Design using Open Source Hardware*, presentazione in *Open Design Symposium*, 23 maggio 2012, Linz, Austria, [www.open-design.at/](http://www.open-design.at/) (ultima visita 26 novembre 2014)
- Dalton M. A., Desjardins A., Wakkary R., *From DIY Tutorials to DIY Recipes*, in *Atti del convegno CHI 2014*, aprile 26–maggio 1, 2014, Toronto, Ontario, Canada, ACM New York, New York, 2014, pp. 1405-1410.
- De Mul J., *Redesigning design*, in Van Abel B. et al, a cura di, *Open Design Now – Why design cannot remain exclusive*, BIS Publishers, Amsterdam, 2011 pp. 34-45.
- Deterding S. et al, *Gamification: towards a definition*, in *Atti della conferenza CHI 2011*, 7-12 maggio, 2011, Vancouver, BC, Canada, ACM, 2011, pp. 12-15.
- Devecchi A., *Il gesto dell'inter-attore. Spazio corpo movimento. Gli ambienti del Gruppo T al Museo del Novecento di Milano*, in *Senza Cornice*, rivista on-line di arte contemporanea e critica, n. 8 Ottobre, Dicembre 2013, [www.senzacornice.org/articoli/pdf/1378114017.pdf](http://www.senzacornice.org/articoli/pdf/1378114017.pdf) (ultima visita 27 novembre 2014).
- Dieter M., Lovink G., *Theses on making in the digital age*, in Hertz G. *Critical Making*, pp. 15-20.

- Digital Habits, *Open Mirror*, in *Digital Habits*, [www.digitalhabits.it/open\\_mirror.php](http://www.digitalhabits.it/open_mirror.php) (ultima visita 20 novembre 2014).
- Dirk S., *Primo: finished!*, in *From Scratch Ed*, <http://fromscratched.nl/index.php/primo-klaar/?lang=en> (ultima visita 20 dicembre 2014).
- Domus web, *Salone 2012: Domus and Audi present The Future in the Making*, in *Domus Web*, 4 aprile 2012, [www.domusweb.it/en/upcoming-event/2012/04/04/salone-2012-domus-and-audi-present-the-future-in-the-making.html](http://www.domusweb.it/en/upcoming-event/2012/04/04/salone-2012-domus-and-audi-present-the-future-in-the-making.html) (ultima visita 20 novembre 2014).
- Dougherty D., *Maker friendly*, in *Make*, vol. 3, Maker Media, Sebastopol, CA, 2014, p. 7.
- Dougherty D., *The Maker Movement*, in *Innovations: Technology, Governance, Globalization, Estate 2012*, vol. 7, n. 3, *The MIT Press Journals*, pp. 11-14.
- Droog, *Tool designers: interview with Studio Ludens*, 15 aprile 2011, in *Droog*, [www.droog.com/news/category/design-for-download-2/](http://www.droog.com/news/category/design-for-download-2/) (ultima visita 26 novembre 2014).
- Dscout, [dscout.com](http://dscout.com) (ultima visita 20 novembre 2014).
- Duilio S., Boer C. R., *Mass Customization and Footwear - myth, Salvation or Reality?: A Comprehensive Analysis of the Adoption of the Mass Customization Paradigm in Footwear, from the Perspective of the Euroshoe (extended User Oriented Shoe Enterprise) Research Project*, Springer-Verlag, London 2007.
- Edufashion, [edufashion.org](http://edufashion.org) (ultima visita 11 novembre 2014).
- Fab Central, *The Fab Charter*, in *Fab Central*, [www.fab.cba.mit.edu/about/charter/](http://www.fab.cba.mit.edu/about/charter/) (ultima visita 20 novembre 2014).
- Fab City Symposium*, Fab10, Conferenza internazionale dei FabLab, Barcellona, 7 luglio 2014.
- FabLab at school, [fablabatschool.org](http://fablabatschool.org) (ultima visita 11 novembre 2014).
- FabLab Maastricht, [www.fablabmaastricht.nl/research-ons-fablab](http://www.fablabmaastricht.nl/research-ons-fablab) (ultima visita 11 novembre 2014).
- Fablab SUPSI, *Rolling Radio*, in *FabLab SUPSI*, [www.fablab.supsi.ch/portfolio/rolling-radio/](http://www.fablab.supsi.ch/portfolio/rolling-radio/) (ultima visita 20 dicembre 2014).
- Fablabs.io, [fablabs.io](http://fablabs.io) (ultima visita 26 novembre 2014).
- Fischer G., *Meta-design: Beyond user-centered and participatory design*, in Stephanidis C., a cura di, *Atti della conferenza HCI*

- International 2003*, Crete, Grecia, 2003, pp. 78-82.
- Fischer G., Scharff E., *Meta-design: Design for designers*, in *Atti della conferenza International Conference on Designing Interactive Systems (DIS'00)*, Brooklyn, New York, USA, 2000, ACM Press, 2000, pp. 396-405.
- Fischer H. G., Giaccardi E., 2004, *Meta-design: a manifesto for end-user development*, in Lieberman, F. Paterno, V. Wulf, a cura di, *End User Development*, Springer, Dordrecht 2005, pp. 427-457.
- Fits.me, [fits.me](http://fits.me) (ultima visita 20 novembre 2014).
- Frazer B., Lueg R., Borden R., Howard J., *Dronenet: The Quad Chronicles*, [www.eecs.ucf.edu/seniordesign/fa2013sp2014/g07/files/Conference%20Paper.pdf](http://www.eecs.ucf.edu/seniordesign/fa2013sp2014/g07/files/Conference%20Paper.pdf) (ultima visita 11 novembre 2014).
- Fritzing, *Open-source software for documenting prototypes, learning interactive electronics and PCB production*, in Fritzing, [www.fritzing.org/media/uploads/publications/FritzingInfoBrochure-lowRes.pdf](http://www.fritzing.org/media/uploads/publications/FritzingInfoBrochure-lowRes.pdf) (ultima visita 20 novembre 2014).
- Fritzing, [fritzing.com](http://fritzing.com) (ultima visita 20 novembre 2014).
- Gershenfeld N., *Fab: The Coming Revolution on Your Desktop—from Personal Computers to Personal Fabrication*, Basic Books, New York 2007.
- Git, [git-scm.com](http://git-scm.com) (ultima visita 20 novembre 2014).
- Greenberg A., *Inside Thingiverse, The Radically Open Website Powering The 3D Printing Movement*, 21 novembre 2012, [www.forbes.com/sites/andygreenberg/2012/11/21/inside-thingiverse-the-radically-open-website-powering-the-3d-printing-movement/](http://www.forbes.com/sites/andygreenberg/2012/11/21/inside-thingiverse-the-radically-open-website-powering-the-3d-printing-movement/) (ultima visita 11 novembre 2014).
- Grima J., *Adhocracy*, in *Atti del simposio M+ Matters – Asian Design : histories, collecting, curating*, 2-3 Dicembre 2012, p. 10 [www.mplusmatters.hk/asiandesign/paper\\_topic3.php?l=en](http://www.mplusmatters.hk/asiandesign/paper_topic3.php?l=en) (ultima visita 26 novembre 2014).
- Howe J., *Crowdsourcing: Why the Power of the Crowd Is Driving the Future of Business*, Crown Business, New York 2009.
- IDEO, *Three years of impact*, [www.openideo.com/content/impact](http://www.openideo.com/content/impact) (ultima visita 20 novembre 2014).
- Igoe T., *Sketching in hardware*, in *Sketching in Hardware 2012*, presentazione, Portland, 22 luglio 2012, <http://sketching-in->

- [hardware.com/2012/presentations/Sketching\\_2012\\_IoT\\_Igoe.pdf](http://hardware.com/2012/presentations/Sketching_2012_IoT_Igoe.pdf) (ultima visita 20 dicembre 2014).
- Improving software, *5 Best Practices for Commenting Your Code*, in *Improving software*, 27 giugno 2011, [www.improvingsoftware.com/2011/06/27/5-best-practices-for-commenting-your-code](http://www.improvingsoftware.com/2011/06/27/5-best-practices-for-commenting-your-code) (ultima visita 20 novembre 2014).
- Instructables, *Who we are*, in *Instructables*, [www.instructables.com/about](http://www.instructables.com/about) (ultima visita 11 novembre 2014).
- Iwamoto L., *Digital fabrications Architectural and Material Techniques*, Princeton Architectural Press, New York 2009.
- Jasserand C., *Creative Commons licences and design Are the two compatible?*, in *Jipitec Journal of Intellectual Property, Information Technology and E-Commerce Law*, vol. 2, 2011.
- Jiao J., Tseng M.M., *Customizability analysis in design for mass customization*, in *Computer-Aided Design*, vol. 36, n. 8, 2004, pp. 745-757.
- Jones R. et al, *RepRap – The Replicating Rapid Prototyper*, in *Robotica*, vol. 2, n. 1, 2011, pp. 177-191.
- Kadushin R. , *Open Design manifesto*, in *Ronen Kadushin*, 2010, [www.ronen-kadushin.com/files/4613/4530/1263/Open\\_Design\\_Manifesto-Ronen\\_Kadushin\\_.pdf](http://www.ronen-kadushin.com/files/4613/4530/1263/Open_Design_Manifesto-Ronen_Kadushin_.pdf) (ultimo accesso 26 novembre 2014).
- Kennedy G., *Joris Laarman's experiments with open source design*, in van Abel B. et al, *Open Design Now*, Bis Publisher, Amsterdam, 2011, pp. 118-125.
- Khemsuro M., *Dario Buzzini and Barbara Busatta on Machine Series*, 16 gennaio 2014, [www.sightunseen.com/2014/01/dario-buzzini-and-barbara-busatta-on-machine-series/](http://www.sightunseen.com/2014/01/dario-buzzini-and-barbara-busatta-on-machine-series/) (ultimo accesso 26 novembre 2014).
- Kickstarter, *Primo. Teaching programming logic to children age 4 to 7*, [www.kickstarter.com/projects/1039674461/primo-teaching-programming-logic-to-children-age-4](http://www.kickstarter.com/projects/1039674461/primo-teaching-programming-logic-to-children-age-4) (ultima visita 20 novembre 2014).
- Kobayashi S., *Sketching A New Product Ecosystem Starting with 'Peripheral' Products*, *Sketching*, presentazione in *Sketching in Hardware 2013*, 18 luglio 2013 Xerox Palo Alto Research Center, Palo Alto, California, U.S.A.
- Korg, *We Love Monotron*, in *Korg*, [www.korg.co.jp/Product/Dan-](http://www.korg.co.jp/Product/Dan-)

- [ce/monotron/welove](#) (ultima visita 20 novembre 2014).
- Kuniavsky M., *Smart Things: Ubiquitous Computing User Experience Design*, Morgan Kaufmann, Burlington, MA, 2010.
- Kuznetsov S. , E. Paulos, *Rise of the Expert Amateur: DIY Projects, Communities, and Cultures*, in *Atti della conferenza Nordi-CHI '10, 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries*, ACM New York, NY, USA 2010, pp. 295-304.
- Ladyada, *Open Hardware Licenses*, 17 maggio 2011, [www.ladyada.net/library/openhardware/license.html](http://www.ladyada.net/library/openhardware/license.html) (ultima visita 26 novembre 2014).
- Lampada Fattelo, [fattelo.com](http://fattelo.com) (ultima visita 20 novembre 2014).
- Le Corbusier, *Towards a new architecture*, Dover publications, New York 1986.
- Learning System Adafruit, [learn.adafruit.com](http://learn.adafruit.com) (ultima visita 27 novembre 2014).
- Lectron, [lectron.info](http://lectron.info) (ultima visita 20 novembre 2014).
- Lee J., *Design Rationale Systems: Understanding the Issues*, in *IEEE Expert*, vol. 12, n. 3, 1997, pp. 78-85.
- Lego Mindstorm, [mindstorms.lego.com](http://mindstorms.lego.com) (ultima visita 20 novembre 2014).
- Levin M., *Designing Multi-Device Experiences An Ecosystem Approach to User Experiences across Devices*, O'Reilly Media, Sebastopol, CA, 2014.
- Lindell R., *The Craft of Programming Interaction*, in Lai-Chong Law E. et al, a cura di, *atti del 2nd International Workshop on the Interplay between User Experience Evaluation and Software Development*, Copenhagen 2012, pp. 26-30.
- Linux Foundation, *How to Participate in the Linux Community*, in *Linux Foundation*, [www.linuxfoundation.org/content/how-participate-linux-community](http://www.linuxfoundation.org/content/how-participate-linux-community) (ultima visita 20 novembre 2014).
- Linux Foundation, *The future is open*, in *Linux Foundation*, [www.linuxfoundation.org/about](http://www.linuxfoundation.org/about) (ultima visita 20 novembre 2014).
- Linux mailing list, [www.lkml.org/](http://www.lkml.org/) (ultima visita 20 novembre 2014).
- Little Bits, [www.littlebits.cc](http://www.littlebits.cc) (ultima visita 26 novembre 2014).

- Loglio M., *Spruzzo*, in *Matlo*, [matlo.me/?work?spruzzo](http://matlo.me/?work?spruzzo) (ultima visita 20 novembre 2014).
- Lommes T., *Open Structures, on the Potential of Open Modular Design*, presentazione in conferenza *Open Design Shared Creativity*, 1 luglio 2014.
- M. Obrist, *DIY HCI – Do-It-Yourself Human Computer Interaction*, VDM Verlag, Saarbrücken 2008, pp. 92-95.
- Määttä A., *A talk on Fablab project documentation*, workshop presso Open Knowledge Festival 2012, [2012.okfestival.org/open-design/](http://2012.okfestival.org/open-design/) (ultima visita 20 novembre 2014).
- Määttä A., Troxler P., *Developing open & distributed tools for Fablab project documentation*, in *Atti della conferenza Open Knowledge Conference, OKCon 2011*, Berlino, 30 giugno-1 luglio 2011, CEUR Workshop Proceedings, vol. 739, 2011.
- Machine Series, [machine-series.com](http://machine-series.com) (ultima visita 26 novembre 2014).
- Maffei S., *Autoproduzione oggi*, in D. Moretti, a cura di, *Le pagine dell'ADI, Associazione per il disegno industriale*, in *Op.Cit.*, 146, gennaio 2013 pp. 7-13.
- Make Magazine, *An in-depth profile of Makers at the Forefront of Hardware Innovation*, *Make: Hardware-Innovation Workshop, 2014*, in *Make*, [www.makezine.com/hardware-innovation-workshop/research/](http://www.makezine.com/hardware-innovation-workshop/research/) (ultima visita 20 novembre 2014).
- Make, [makezine.com](http://makezine.com) (ultima visita 11 novembre 2014).
- Makerbot industries, [makerbot.com](http://makerbot.com) (ultima visita 11 novembre 2014).
- Manovich L., *Database as a Symbolic Form*, 1998, [www.manovich.net/content/04-projects/020-database-as-a-symbolic-form/19\\_article\\_1998.pdf](http://www.manovich.net/content/04-projects/020-database-as-a-symbolic-form/19_article_1998.pdf) (ultima visita 26 novembre 2014).
- Manzini E., *Enabling platforms for creative communities*, in *Doors of perception*, 18 marzo 2005, [www.doors8delhi.doorsofperception.com/presentationspdf/manzini.html](http://www.doors8delhi.doorsofperception.com/presentationspdf/manzini.html) (ultima visita 26 novembre 2014).
- Mari E., *Autoprogettazione?*, Edizioni Corraini, Mantova 2002.
- Mari E., *Dibattito critico-ideologico di E. Mari con E. Facchinelli*, in E. Mari, *Autoprogettazione?*, Edizioni Corraini, Mantova 2002, pp. 40-44.

- Master of Advanced Studies in Interaction Design SUPSI, [maind.supsi.ch](http://maind.supsi.ch) (ultima visita 20 novembre 2014).
- MatterMachine, [www.mattermachine.com](http://www.mattermachine.com) (ultima visita 20 novembre 2014)
- McConnell S., *Code Complete: A Practical Handbook of Software Construction*, Microsoft Press, Washington, 2004.
- Mcdonald T., *Community Manager: Key to the Future of Business*, in *Huffington post*, 27 gennaio 2014, [www.huffingtonpost.com/tim-mcdonald/community-manager\\_b\\_4670747.html](http://www.huffingtonpost.com/tim-mcdonald/community-manager_b_4670747.html) (ultima visita 20 novembre 2014).
- Mcguirk J., *The perfect architectural symbol for an era obsessed with customisation and participation*, 20 marzo 2014, in *Deezen*, [www.deezen.com/2014/03/20/opinion-justin-mcguirk-le-corbusier-symbol-for-era-obsessed-with-customisation/](http://www.deezen.com/2014/03/20/opinion-justin-mcguirk-le-corbusier-symbol-for-era-obsessed-with-customisation/) (ultima visita 20 novembre 2014).
- Mckeough T., *Got a Good Idea? Ben Kaufman of Quirky.com Will Be the Judge of That*, in *The New York Times*, 24 agosto 2011, [www.nytimes.com/2011/08/25/garden/ben-kaufman-quirkycom-founder-on-problem-solving-qa.html?\\_r=0](http://www.nytimes.com/2011/08/25/garden/ben-kaufman-quirkycom-founder-on-problem-solving-qa.html?_r=0) (ultima visita 20 novembre 2014).
- Mellis D., Banzi M., Cuartielles D., Igoe T., *Arduino: An open electronics prototyping platform*, in *Atti di CHI 07 Conference on Human Factors in Computing*, ACM, New York, 2007, pp. 83-90.
- Mellis D., Buechley L., *Collaboration in open-source hardware: third-party variations on the arduino duemilanove*, in *Atti di ACM 2012 conference on Computer Supported Cooperative Work (CSCW '12)*, ACM, New York, 2012, pp. 1175-1178.
- Mellis. D, Buechley L., *Do-it-yourself cellphones: an investigation into the possibilities and limits of high-tech diy*, in *CHI '14 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM New York, New York, pp. 1723-1732.
- Menichinelli M., *Open P2P design toolkit- How to co-design an open collaborative process*, [issuu.com/openp2pdesign/docs/openp2pdesign.toolkit\\_dmy](http://issuu.com/openp2pdesign/docs/openp2pdesign.toolkit_dmy) (ultima visita 26 novembre 2014).
- Merriam Webster Encyclopaedia, voce "Blueprint", in *Merriam*

- Webster Encyclopaedia*, [www.merriam-webster.com/dictionary/blueprint](http://www.merriam-webster.com/dictionary/blueprint) (ultima visita 20 dicembre 2014).
- Mischer'Traxler, *Collective works*, in Mischer'Traxler, [www.mischertraxler.com/projects\\_collective\\_works.html](http://www.mischertraxler.com/projects_collective_works.html) (ultima visita 20 novembre 2014).
- Mockus A. et al, *Two case studies of open source software development: Apache and Mozilla*, in *ACM Transactions on Software Engineering and Methodology (TOSEM)*, vol. 11, n. 3, 2002, pp. 309-346.
- Mota C., *The history of Open Hardware Movement*, in Gibbs A., *Building Open Source Hardware – Manufacturing for the DIY Maker Movement*, Addison-Wesley Professional, Indianapolis, 2014.
- Mozilla Firefox, [mozilla.org](http://mozilla.org) (ultima visita 11 novembre 2014).
- Munari B., *Codice Ovvio*, Einaudi, Torino 1971 (edizione 2008).
- Nervous system Tools, *Tools*, in *Nervous system*, [www.n-e-r-v-o-u-s.com/tools](http://www.n-e-r-v-o-u-s.com/tools) (ultima visita 20 novembre 2014).
- Nervous system, [n-e-r-v-o-u-s.com](http://n-e-r-v-o-u-s.com) (ultima visita 20 novembre 2014).
- Nike, *NikeID*, in Nike, [www.nike.com/us/en\\_us/c/nikeid](http://www.nike.com/us/en_us/c/nikeid) (ultima visita 20 novembre 2014).
- Norman D.A., Draper S.W., a cura di, *User Centered Systems Design New Perspective on Human Computer Interaction*, Lawrence Erlbaum Associates Inc, Hillsdale, NJ, 1986.
- O'Reilly T., *What is Web 2.0*, 2005, <http://www.oreilly.com/pub/a/web2/archive/what-is-web-20.html> (ultima visita 11 novembre 2014).
- OIEngine, [oiengine.com](http://oiengine.com) (ultima visita 20 novembre 2014).
- Open Design Working Group, [design.okfn.org](http://design.okfn.org) (ultima visita 26 novembre 2014)
- Open Design Working Group, *Open Design Definition V 0.3*, [www.github.com/OpenDesign-WorkingGroup/Open-Design-Definition/blob/master/open.design\\_definition/open.design.definition.md](http://www.github.com/OpenDesign-WorkingGroup/Open-Design-Definition/blob/master/open.design_definition/open.design.definition.md) (ultima visita 26 novembre 2014).
- Open E-components, [cargocollective.com/open\\_ecomponents](http://cargocollective.com/open_ecomponents) (ultima visita 20 novembre 2014).
- Open Hardware Repository, *CERN Open Hardware License*, [www.ohwr.org/cernohl](http://www.ohwr.org/cernohl) (ultima visita 11 novembre 2014).

- Open Knowledge, *About*, in *Open Knowledge*, [okfn.org](http://okfn.org) (ultima visita 26 novembre 2014).
- Open Source Definition, [opensource.org/osd](http://opensource.org/osd) (ultima visita 26 novembre 2014).
- Open Source Ecology, *The mission of Open Source Ecology (OSE) is to create the open source economy*, in *Open Source Ecology*, [www.opensourceecology.org/about-overview/](http://www.opensourceecology.org/about-overview/) (ultima visita 26 novembre 2014).
- Open Source Hardware Documentation Jam, [opensourcewarehouse.org](http://opensourcewarehouse.org) (ultima visita 20 dicembre 2014).
- Open source initiative, [www.opensource.org](http://www.opensource.org) (ultima visita 11 novembre 2014).
- OpenIDEO, [openideo.com](http://openideo.com) (ultima visita 20 novembre 2014).
- OpenProcessing, [openprocessing.org](http://openprocessing.org) (ultima visita 20 novembre 2014).
- OpenStructures, [openstructures.net](http://openstructures.net) (ultima visita 20 novembre 2014).
- Openwear, [openwear.org](http://openwear.org) (ultima visita 11 novembre 2014),
- OSHWA, Open Source Hardware Association, *Best Practices for Open-Source Hardware 1.0*, in OSHWA, [www.oshwa.org/sharing-best-practices/](http://www.oshwa.org/sharing-best-practices/) (ultima visita 20 dicembre 2014).
- OSHWA, Open Source Hardware Association, [oshwa.org](http://oshwa.org) (ultima visita 20 novembre 2014).
- OSHWA, *Open Source Hardware Definition (English)*, in OSHWA, [www.oshwa.org/definition](http://www.oshwa.org/definition) (ultima visita 24 novembre 2014).
- Oxford Dictionary, voce “Commons”, in *Oxford Dictionary*, [www.oxforddictionaries.com/definition/english/commons](http://www.oxforddictionaries.com/definition/english/commons) (ultima visita 11 novembre 2014).
- P2P Foundation, voce “Maker movement”, in *P2P Foundation*, [p2pfoundation.net/Maker\\_Movement](http://p2pfoundation.net/Maker_Movement) (ultima visita 11 novembre 2014)
- Parvin A., *Wikihouse*, in *Fab City Symposium*, 7 luglio 2014, Disney Hub Barcelona, Barcelona, in *Fab10*, [www.fab10.org/en/symposium](http://www.fab10.org/en/symposium) (ultima visita 20 novembre 2014).
- Perens B., *Open source definition*, in DiBona C., Ockman S., *Open Sources: Voices from the Open Source Revolution*, O'Reilly Media, Sebastopol, CA 1999.

- Perl Foundation, *The state of the onion*, 2009, [www.perlfoundation.org/perl6/index.cgi?state\\_of\\_the\\_onion](http://www.perlfoundation.org/perl6/index.cgi?state_of_the_onion) (ultima visita 11 novembre 2014).
- Piller T. et al, *From Social Media to Social Product Development: The Impact of Social Media on Co-Creation of Innovation*, in *Die Unternehmung*, vol. 65, n. 1, 2012.
- Popular Mechanics, [popularmechanics.com](http://popularmechanics.com) (ultima visita 11 novembre 2014).
- Primo manual, [docs.primo.io](http://docs.primo.io) (ultima visita 20 dicembre 2014).
- Primo, *Primo prototype documentation*, in *Primo*, [www.docs.primo.io](http://www.docs.primo.io) (ultima visita 20 novembre 2014).
- Primo, *Say hello to Dolphins*, in *Primo*, [www.primo.io/blog/say-hello-to-dolphins](http://www.primo.io/blog/say-hello-to-dolphins) (ultima visita 20 novembre 2014).
- Project ARA, *Project ARA developer conference*, in *Project ARA*, [www.projectara.com/ara-developers-conference](http://www.projectara.com/ara-developers-conference) (ultima visita 20 novembre 2014).
- Proverbio P., *Danese 1957 – 1991. un laboratorio sperimentale per il design*, *AISD Design, Storia e Ricerche*, in *Arte Lombarda. Nuova serie*, n. 161-162, 2011, 1-2, pp. 126-134.
- Raymond S., *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*, O'Reilly Media, Sebastopol, CA, 1999.
- Reactable, [reactable.com](http://reactable.com) (ultima visita 11 novembre 2014).
- Reactivision 1.5, [reactivision.sourceforge.net](http://reactivision.sourceforge.net) (ultima visita 20 novembre 2014).
- RepRap wiki, [reprap.org](http://reprap.org) (ultima visita 20 novembre 2014).
- Rifkin J., *The zero marginal cost society*, Palgrave Macmillan Trade, New York 2014.
- Robben B., *The history of production with computers*, in J. Walter-Herrmann, C. Büching, a cura di, *FabLab of Machines, Makers and Inventors*, transcript verlag, Bielefeld 2013, p. 131.
- Romagnoli L., *Digital Lathe*, in *Lorenzo Romagnoli*, [cargocollective.com/lorenzoromagnoli/Digital-Lathe](http://cargocollective.com/lorenzoromagnoli/Digital-Lathe) (ultima visita 20 dicembre 2014).
- Romano Z., Cangiano S., *Open Sourcing Wearables: the Impact of Open Technologies and User Engagement in the Design of Body-Borne Interactive Products*, in Bihanic D., *Empowering Users through Design*, Springer-Verlag, London, 2015, pp.

153-175.

- Romano Z., *Come hackerare una sedia nell'era dell'open design*, in *Che Futuro!*, 14 settembre 2012, [www.chefuturo.it/2012/09/come-hackerare-una-sedia-nellera-dell-open-design/](http://www.chefuturo.it/2012/09/come-hackerare-una-sedia-nellera-dell-open-design/) (ultima visita 20 dicembre 2014).
- Royal Academy of Engineering, *Additive manufacturing: opportunities and constraints*, in *Royal Academy of Engineering*, London, 2013 [www.raeng.org.uk/publications/reports/additive-manufacturing](http://www.raeng.org.uk/publications/reports/additive-manufacturing) (ultima visita 26 novembre 2014).
- Schoffeleene J., Huybrechts L., *Sharing is caring. Sharing and documenting complex participatory projects to enable generative participation*, in *Interaction Design and Architecture(s) Journal - IxD&A*, n. 18, 2013, pp. 9-22.
- Schön D. A., *The reflective practitioner: how professionals think in action*, Basic Book, London, 1983.
- Scratch, [scratch.mit.edu](http://scratch.mit.edu) (ultima visita 20 dicembre 2014).
- Seidle N., *Designing better PCBs in Eagle*, 5 novembre 2008, [www.sparkfun.com/tutorials/115](http://www.sparkfun.com/tutorials/115) (ultima visita 11 novembre 2014).
- Sennet R., *Luomo artigiano*, Feltrinelli, Milano 2008.
- Seravalli A., *Making commons (attempts at composing prospects in the opening of production)*, tesi di dottorato in Interaction Design, Malmö University, School of Arts and Communication, Faculty: Culture and Society, 2014.
- Shove E., M. Pantzar, M. Watson, *The dynamics of social practice – everyday life and how it changes*, SAGE Publications, London 2012.
- Simpson T.W., *Product platform design and customization: status and promise*, in *Artificial Intelligence for Engineering Design, Analysis & Manufacturing*, vol. 18, n. 1, 2004, pp. 3-20.
- SkectChair, [sketchchair.com](http://sketchchair.com) (ultima visita 26 novembre 2014).
- Smith G.W., *Computer art and real art*, in *PAGE – Bulletin of Computer Art Society*, n. 22, 1972.
- Sparkfun Electronics, *Triple Axis Accelerometer Breakout ADXL330*, in *Sparkfun*, [www.sparkfun.com/products/retired/692](http://www.sparkfun.com/products/retired/692) (ultima visita 20 dicembre 2014).
- Sparkfun, [sparkfun.com](http://sparkfun.com) (ultima visita 11 novembre 14).
- Studio Ludens, [studioludens.com](http://studioludens.com) (ultima visita 11 novembre

- 2014).
- TAPR, *The TAPR Open Hardware License*, in TAPR, [www.tapr.org/ohl.html](http://www.tapr.org/ohl.html) (ultima visita 11 novembre 2014).
- Techopedia, voce “Platform”, in *Techopedia*, <http://www.techopedia.com/definition/3411/platform> (ultima visita 20 dicembre 2014).
- Temboo, [temboo.com](http://temboo.com) (ultima visita 20 novembre 2014).
- The Economist, *Open sesame*, in *The Economist*, 7 giugno 2008, <http://www.economist.com/node/11482589> (ultima visita 20 dicembre 2014).
- Thingiverse, [thingiverse.com](http://thingiverse.com) (ultima visita 11 novembre 2014).
- Tinkerkit, *Tutorials*, in *Tinkerkit*, [www.web.archive.org/web/20131231044649/www.tinkerkit.com](http://www.web.archive.org/web/20131231044649/www.tinkerkit.com) (ultimo accesso 26 novembre 2014).
- Tinkerkit, [tinkerkit.com](http://tinkerkit.com) (ultima visita 20 novembre 2014).
- Tonze J. et al, *Open Design: Contributions, Solutions, Processes and Projects*, in *The Design Journal*, vol. 17, n. 4, dicembre 2014, Bloomsbury Journals, London, pp. 538-559.
- Torrey C. et al, *How-To Pages: Informal Systems of Expertise Sharing*, in L. Bannon et.al, *ECSCW 2007: Proceedings of the Tenth European Conference on Computer-Supported Cooperative Work*, 24-28 settembre 2007, Limerick, Irlanda, Springer, Netherlands, 2007, pp. 391-410.
- Torrone P., *Arduino: Open source hardware 2009 – The definitive guide to open source hardware projects in 2009*, in *Makezine*, 11 dicembre 2009, [makezine.com/2009/12/11/arduino-open-source-hardware-2009/](http://makezine.com/2009/12/11/arduino-open-source-hardware-2009/) (ultima visita 20 novembre 2014).
- Torrone P., *The Bill Of Rights*, in *Make*, [www.cdn.makezine.com/make/MAKERS\\_RIGHTS.pdf](http://www.cdn.makezine.com/make/MAKERS_RIGHTS.pdf) (ultima visita 26 novembre 2014).
- Townsend K., *Creating an accurate footprints in Eagle*, in *Adafruit Learning System*, [www.learn.adafruit.com/creating-accurate-footprints-in-eagle/overview](http://www.learn.adafruit.com/creating-accurate-footprints-in-eagle/overview) (ultima visita 20 dicembre 2014).
- Toxilibs, [toxiclibs.org](http://toxiclibs.org) (ultima visita 20 novembre 2014).
- Troxler P., Maxigas, *Editorial Note: We Now have the Means of Production, but Where is my Revolution*, in *Journal of Peer Pro-*

- duction*, n. 5, Shared Machine Shops, ottobre 2014, <http://peerproduction.net/issues/issue-5-shared-machine-shops/editorial-section/editorial-note-we-now-have-the-means-of-production-but-where-is-my-revolution/> (ultima visita 20 dicembre 2014).
- Tseng T., Reisnick M., *Product Versus Process: Representing and Appropriating DIY Projects Online*, in *Atti del convegno Crafting Design*, DIS 2014, giugno 21-25, 2014, Vancouver, BC, Canada, pp. 425-428.
- Turk M., *Perceptual User Interfaces*, in *Communications of the ACM*, vol. 43, n. 3, 2000, pp. 32-34.
- Turner D., *Amazon Kindle*, in *Technology Review*, 19 febbraio 2008, [www.technologyreview.com/hack/409569/amazon-kindle/](http://www.technologyreview.com/hack/409569/amazon-kindle/) (ultima visita 20 novembre 2014).
- Ullmer B., Ishii H., *Emerging frameworks for tangible user interfaces*, in *IBM Systems Journal*, 2000, vol. 39, n. 3-4, pp. 915-931.
- Unfold, *L'artisan eletronique*, in *Unfold*, [www.unfold.be/pages/l-artisan-electronique](http://www.unfold.be/pages/l-artisan-electronique) (ultima visita 20 novembre 2014).
- Vasarely V., *Plasti-Cité: L'oeuvre Plastique Dans Votre Vie Quotidienne*, Casterman, Parigi, 1970.
- Vasarely V., *Planetary Folklore Participations*, New York Graphic Society, New York, 1973.
- Vectorealism, *Illustrator starter kit*, in *Vectorealism.com*, [www.en.vectorealism.com/support/getting-started/illustrator-starter-kit/#how-this-works](http://www.en.vectorealism.com/support/getting-started/illustrator-starter-kit/#how-this-works) (ultima visita 20 dicembre 2014)
- Vergine L., *Arte programmata e cinetica (1953-1963). L'ultima avanguardia*, Mazzotta, Milano, 1983.
- Vinca Masini L., *Arte programmata. Op Art*, in *Domus*, n. 422, 1965.
- von Hippel E., *Democratizing Innovation*, MIT Press, Cambridge, MA, 2005.
- von Hippel E., *Lead Users: A Source of Novel Product Concepts*, in *Organization Science*, vol. 32, n. 7, 1986, pp. 791-805.
- Warnier C. et al, *Printing Things Visions and Essentials for 3D Printing*, Gestalten, Berlino, 2014.
- Weber S., *The success of open source*, Harvard University Press,

Cambridge, 2005.

- WeMake, *Make things open*, in *WeMake*, [www.wemake.cc/making-things-open](http://www.wemake.cc/making-things-open) (ultima visita 20 dicembre 2014).
- Wikipedia, voce “Free and open source software”, in *Wikipedia*, [www.en.wikipedia.org/wiki/Free\\_and\\_open-source\\_software](http://www.en.wikipedia.org/wiki/Free_and_open-source_software) (ultima visita 24 novembre 2014).
- Wikipedia, voce “Metodo”, in *Wikipedia*, [it.wikipedia.org/wiki/Metodo\\_%28programmazione%29](http://it.wikipedia.org/wiki/Metodo_%28programmazione%29) (ultima visita 26 novembre 2014).
- Wikipedia, voce “Object oriented programming”, in *Wikipedia*, [en.wikipedia.org/wiki/Object-oriented\\_programming](http://en.wikipedia.org/wiki/Object-oriented_programming) (ultima visita 20 novembre 2014).
- Wikipedia, voce “Simulation”, in *Wikipedia*, [www.en.wikipedia.org/wiki/Simulation](http://www.en.wikipedia.org/wiki/Simulation) (ultima visita 20 novembre 2014).
- Wikipedia, voce “Social networking service”, in *Wikipedia*, [www.en.wikipedia.org/wiki/Social\\_networking\\_service](http://www.en.wikipedia.org/wiki/Social_networking_service) (ultima visita 20 novembre 2014).
- Willis K. D.D. et al, *Interactive Fabrication: New Interfaces for Digital Fabrication*, in atti della conferenza *TEI'11 fifth international conference on Tangible, embedded, and embodied interaction*, ACM New York, 2011, pp. 69-72



# Ringraziamenti

Il maker è sempre “plurale” così come lo è chi si occupa di ricerca. Una tesi di dottorato sul tema del design in relazione alle pratiche dell’open source ha un debito con tante comunità, da quella scientifica alla comunità del design, dei maker e delle persone comuni, degli imprenditori e degli inventori.

Voglio ringraziare Davide Fornari, collega e amico che mi ha accompagnato in questo percorso dall’inizio condividendo il suo sapere e i suoi metodi.

Giovanni Profeta, dottorando della SUPSI e amico, che mi ha aiutato a visualizzare meglio le idee e i contenuti del diagramma del design framework.

I professori e ricercatori del Consiglio del curriculum in Scienze del design della Scuola di Dottorato dell’Università Iuav di Venezia, Raimonda Riccini, Monique Arnaud, Laura Badalucco, Alberto Bassi, Medardo Chiapponi, Massimiliano Ciammaichella, Emanuela Fanny Bonini Lessing, Paola Donati, Walter Le Moli, Mario Lupano, Simonetta Morini, Gigi Pescolderung, Michele Sinico, Marco Zito.

Ringrazio il mio relatore Davide Rocchesso.

I miei colleghi della Scuola di Dottorato: Claudio Beorchia, Ruggero Blasi, Maria Teresa Dal Bò, Pietro Costa, Silvia Gasparotto, Silvio Lorusso, Giulia Ciliberto, Anastasia Kozlova, Daniele Savasta, Maddalena Mometti.

Ringrazio i colleghi ricercatori, collaboratori e assistenti del Labo-

ratorio cultura viva della Scuola universitaria professionale della Svizzera italiana: Leonardo Angelucci, Massimo Botta, Jean-Pierre Candeloro, Vanessa De Luca, Fabian Frei, Elisabetta Lazzaroni, Sara Medica, Iolanda Pensa, Marta Pucciarelli.

Per il sostegno e la borsa di studio, ringrazio la Direzione del Dipartimento ambiente costruzioni e design e la Direzione della SUPSI.

Ringrazio Innocenzo Rifino e Diego Rossi di Habit(s) e Matteo Loglio e Filippo Jacob di SolidLabs per avermi sostenuta nella preparazione del workshop “Progettare prodotti-piattaforma”. Ringrazio i partecipanti del workshop e l’associazione e maker-space WeMake di Milano per averlo ospitato.

I miei esperti preferiti di open source hardware, open design e interaction design: Massimo Banzi, Costantino Bongiorno, David Cuartielles, Fabio Franchino, Massimo Menichinelli, Giorgio Olivero, Zoe Romano.

I membri delle comunità dei FabLab, delle Maker Faire, del movimento open hardware e open design che ho incontrato nei miei viaggi di ricerca e negli interessanti spazi di discussione nel web.

Le persone coinvolte nel progetto “Arte ri-programmata: un manifesto aperto” e, in particolare, Azalea Seratoni che mi ha guidato alla scoperta del fantastico mondo del Gruppo T e Giovanni Anceschi che mi ha permesso di realizzare il mio primo vero progetto di artefatto interattivo open source.

Gli amici che hanno atteso pazientemente durante la scrittura della tesi: Martina Povoledo, Annina Sedlajec, Olivia Blum, Alfio Mazzei, Mario Pegoraro, Monica Lombardi, Titti Amendola, Enrico Sabatino, Antonio Affinita, Viola Bottalico, Antonella Kurzen, Paola Querin, Elena De Pascale, Roberto Landi, Concita Brunetti, Francesca De Chiara, Sabina Barcucci.

Ringrazio i miei genitori, Mariapia e Vincenzo, per il loro costante supporto anche a distanza e mio fratello Mimmo per le lunghe discussioni sul futuro del design e della tecnologia.