FACILE DA USARE

UNA MODERNA INTRODUZIONE ALL'INGEGNERIA DELL'USABILITÀ

ROBERTO POLILLO

Università degli Studi di Milano Bicocca
Dipartimento di Informatica, Sistemistica e Comunicazione

VERSIONE FINALE
27 MARZO 2010

SOMMARIO

PREFAZIONE **6**

1. SISTEMI INTERATTIVI E INTERFACCE D'USO **11**

Sintesi del capitolo 11
Sistemi e interfacce 11
Le dimensioni della complessità 14
La diversità degli utenti 17
La velocità del cambiamento 18
Iperfunzionalismo e altri problemi 22
Complessità d'uso e divario digitale 24
Il ruolo dell'interfaccia utente 25
La Human Computer Interaction 27
Ripasso ed esercizi 30
Approfondimenti e ricerche 31

2. EVOLUZIONE DEI PARADIGMI D'INTERAZIONE **32**

Sintesi del capitolo 32
Paradigmi e tecnologie di interazione 32
Il terminale scrivente: scrivi e leggi 33
Il terminale video: indica e compila 35
Il personal computer: non dirlo, fallo 36
Il browser web: point & clic 41
Il mobile: alzati e cammina 45
Il social computing 52
L'intelligenza ambientale 55
Ripasso ed esercizi 56
Approfondimenti e ricerche 56

3. USABILITÀ **58**

Sintesi del capitolo 58
Un modello dell'interazione 58
Affordance e feedback 63
La nozione di usabilità 66
Apprendibilità e memorabilità 69
Sussidi all'utente 71
Usabilità universale 76
Accessibilità 77
Ripasso ed esercizi 80
Approfondimenti e ricerche 80

4. CONOSCERE L'UTENTE **81**

Sintesi del capitolo 81
La diversità degli utenti 81
Modelli dell'utente 83
L'attenzione 86
La memoria 90
La visione 99
Il sistema motorio 102

L'UTENTE NEL SUO CONTESTO 108
L'ETNOGRAFIA 110
RIPASSO ED ESERCIZI 112
APPROFONDIMENTI E RICERCHE 113

5. PROGETTARE PER L'UTENTE 114

SINTESI DEL CAPITOLO 114
CHE COSA SIGNIFICA PROGETTARE 114
PROGETTARE L'INTERAZIONE 115
PROGETTAZIONE HUMAN-CENTRED 116
UN ESEMPIO 117
I CASI D'USO 119
PROGETTAZIONE UNIVERSALE 120
LIVELLI DI MATURITÀ DELLA PROGETTAZIONE 124
CHI È L'INTERACTION DESIGNER 125
RIPASSO ED ESERCIZI 126
APPROFONDIMENTI E RICERCHE 126

6. L'INGEGNERIA DELLA USABILITÀ 127

SINTESI DEL CAPITOLO 127
LE DIVERSE INGEGNERIE 127
IL MODELLO "A CASCATA" 128
IL CICLO COMPITO-ARTEFATTO 129
MODELLI ITERATIVI 130
IL MODELLO ISO 13407 132
IL RUOLO DELL'UTENTE NEL PROCESSO DI PROGETTAZIONE 136
L'ESEMPIO DEI SITI WEB 138
LE PROFESSIONI DELL'USABILITÀ 139
COSTI E BENEFICI 141
RIPASSO ED ESERCIZI 142
APPROFONDIMENTI E RICERCHE 142

7. I REQUISITI 144

SINTESI DEL CAPITOLO 144
CHE COSA SONO I REQUISITI DI PRODOTTO 144
IL PROCESSO DI DEFINIZIONE DEI REQUISITI 145
LA FASE DI ESPLORAZIONE 146
SCENARI D'USO 149
I CASI D'USO 152
IL DOCUMENTO DEI REQUISITI 160
RIPASSO ED ESERCIZI 162
APPROFONDIMENTI E RICERCHE 162

8. INGEGNERIA E CREATIVITÀ 163

SINTESI DEL CAPITOLO 163
DAI REQUISITI AL DESIGN CONCEPT 163
I PROCESSI DELL'INVENZIONE 164
MIMESI 165
IBRIDAZIONE 167
METAFORA 171

VARIAZIONE 175
COMPOSIZIONE DI DESIGN PATTERN 177
INNOVAZIONE E COMUNICAZIONE 180
RIPASSO ED ESERCIZI 181
APPROFONDIMENTI E RICERCHE 181

9. I PROTOTIPI 183

SINTESI DEL CAPITOLO 183
CHE COS'È UN PROTOTIPO 183
TIPI DI PROTOTIPI 183
SCHIZZI, STORYBOARD E DIAGRAMMI 187
PROTOTIPI INIZIALI 191
PROTOTIPI INTERMEDI 199
PROTOTIPI FINALI 199
RIPASSO ED ESERCIZI 200
APPROFONDIMENTI E RICERCHE 200

10. PRINCIPI E LINEE GUIDA 201

SINTESI DEL CAPITOLO 201
PRINCIPII, LINEE GUIDA, REGOLE DI PROGETTO, STANDARD 201
GLI STANDARD DELLA HUMAN-SYSTEM INTERACTION 202
I PRINCIPI DEL DIALOGO SECONDO LA ISO 9241-110 205
ADEGUATEZZA AL COMPITO 207
AUTO-DESCRIZIONE 210
CONFORMITÀ ALLE ASPETTATIVE 213
ADEGUATEZZA ALL'APPRENDIMENTO 220
CONTROLLABILITÀ 225
TOLLERANZA VERSO L'ERRORE 227
ADEGUATEZZA ALL'INDIVIDUALIZZAZIONE 228
SINTESI DELLE LINEE GUIDA 232
RIPASSO ED ESERCIZI 234
APPROFONDIMENTI E RICERCHE 234

11. PROGETTARE PER L'ERRORE 235

SINTESI DEL CAPITOLO 235
L'ERRORE UMANO 235
PREVENZIONE 237
DIAGNOSI 243
CORREZIONE 247
CONCLUSIONI 250
RIPASSO ED ESERCIZI 250
APPROFONDIMENTI E RICERCHE 250

12. PROGETTARE LA GRAFICA 252

SINTESI DEL CAPITOLO 252
DESIGN DELL'INTERAZIONE E COMUNICAZIONE VISIVA 252
LE LEGGI DELLA GESTALT 256
VICINANZA 262
SOMIGLIANZA 265
CHIUSURA 268

Allineamento 270
Colore 271
Percorsi visivi 274
Ripasso ed esercizi 277
Approfondimenti e ricerche 277

13. PROGETTARE IL TESTO 279

Sintesi del capitolo 279
L'usabilità del testo 279
La tipografia digitale 281
Legibility 287
Readability 292
I manuali di stile 295
Il testo nel Web 298
L'uso creativo del testo 300
Ripasso ed esercizi 303
Approfondimenti e ricerche 303

14. VALUTARE L'USABILITÀ 305

Sintesi del capitolo 305
Verifiche e convalide 305
Valutazioni euristiche 306
Test di usabilità 309
Test formativi e test sommativi 312
Test di compito e test di scenario 315
Misure 316
Come condurre un test di usabilità 317
Il rapporto di valutazione 319
Test di usabilità: costi e benefici 321
Altre tecniche di valutazione 322
Ripasso ed esercizi 322
Approfondimenti e ricerche 323

POSTFAZIONE 324

PER APPROFONDIRE 326

APPENDICE. NOTAZIONE PER GLI STATECHART 330

INDICE ANALITICO 338

Prefazione

Questo libro fornisce un'introduzione generale all'ingegneria dell'usabilità, per la progettazione di prodotti facili da usare, in particolare sistemi interattivi ad alto contenuto di software. È nato come libro di testo per corsi universitari di carattere introduttivo per le discipline dell'interazione uomo macchina o dell'interaction design, oggi presenti in quasi tutti i corsi di laurea in Informatica, ma può essere letto anche da chi non si occupi specificamente di informatica. Infatti, a parte una generica comprensione dei concetti generali dell'informatica e del software, il libro non presuppone nel lettore particolari nozioni di carattere tecnico. Può quindi essere usato senza difficoltà anche da studenti di orientamento diverso, per esempio dei corsi di laurea in Scienze della Comunicazione o Disegno Industriale, o da chiunque desideri avere un inquadramento introduttivo a questi argomenti.

La disciplina dell'interazione uomo macchina ha più di un quarto di secolo, ed è oggi vasta e diversificata. Attinge alle conoscenze di numerose altre discipline, e si declina in molte specializzazioni diverse, alcune orientate al "fare", altre al "conoscere". Questo libro è fortemente orientato al "fare", ed ha un tema molto specifico: si occupa della progettazione di sistemi interattivi che possano dirsi usabili, cioè sistemi che supportino l'essere umano nell'esecuzione delle sue attività quotidiane, rendendole più efficaci, meno faticose e più gradevoli. In ultima analisi, sistemi che migliorino la qualità della nostra vita. Il libro adotta un approccio elementare, anche se concettualmente preciso e non riduttivo, cercando di focalizzarsi sulle idee fondamentali e di arrivare al punto in un numero limitato di pagine. Per non ridurre lo spessore della disciplina, che è notevole, ogni capitolo propone poi alcuni approfondimenti, prevalentemente accessibili in rete, che sono lasciati all'iniziativa del lettore.

Il sottotitolo di questo libro è "Un'introduzione moderna all'ingegneria dell'usabilità". La parola "moderna" merita qualche spiegazione. Anche se l'ingegneria dell'usabilità ha un paio di decenni di vita, il contesto è completamente cambiato da quando Jakob Nielsen, con il suo testo *Usability Engineering*, nel 1993 portò questo termine all'attenzione di un vasto pubblico – almeno nell'ambito dell'informatica. Nel 1993 Internet era nella sua infanzia, la telefonia mobile pure. Il cambiamento introdotto da questi due strumenti, e dalle tecnologie che li hanno resi possibili, è di natura epocale, ed ha modificato profondamente, da allora, non solo il modo con cui ci rapportiamo ai sistemi interattivi, ma anche i nostri comportamenti quotidiani. Oggi – e sempre di più con il passare del tempo – l'impatto della tecnologia è pervasivo, ed ha profonde ripercussioni sulla vita di *tutte* le persone in tutti gli ambiti, dal lavoro al tempo libero. Inoltre la tecnologia genera continuamente nuova tecnologia, e l'accelerazione in questi ultimi anni si è fatta frenetica. Sul mercato vengono continuamente immessi nuovi strumenti, che si propongono di modificare, a volte radicalmente, le nostre abitudini e i nostri comportamenti.

Tutto ciò, se non cambia i principi di base dell'ingegneria dell'usabilità, attribuisce a questa disciplina un ruolo significativamente diverso dal passato, quando costituiva una nicchia seguita da un piccolo numero di addetti. A chi progetta sistemi interattivi, si richiede un atteggiamento molto più consapevole di quanto non lo fosse in un passato non lontano, quando la tecnologia si rivolgeva a un pubblico di utenti molto più ristretto, e relativamente specializzato. Il progettista deve ora essere in grado di collocare i prodotti del suo lavoro nel contesto in cui saranno utilizzati, e di valutarne gli effetti su chi li utilizzerà. Deve essere in grado di scegliere le soluzioni migliori, non solo e non tanto dal punto di vista tecnico – la tecnologia in molti casi è oggi una *commodity* – quanto dal punto di vista complessivo dell'impatto sulla qualità della vita degli utilizzatori. Lo studio della tecnologia fine a se stessa può produrre mostri o, nel caso migliore, gadget sofisticati. In un pianeta in cui la metà della popolazione – più di tre miliardi di persone – vive con meno di 2,5 dollari al giorno, questo, semplicemente, non può essere accettato.

Sono convinto che la formazione che i progettisti di sistemi interattivi ricevono tradizionalmente, nel nostro Paese, nei corsi di laurea in informatica e in ingegneria sia fortemente inadeguata per questo nuovo contesto, perché prescinde totalmente dallo studio dell'*uso* dei sistemi. Ai progettisti viene insegnato a trovare soluzioni tecniche a problemi tecnici. Non viene mai insegnato a sollevare – sia pure per un momento – lo sguardo dal codice dei programmi o dagli

schemi tecnici, per riflettere sul senso di ciò che stanno facendo, e di esaminare l'effetto delle loro scelte progettuali sull'attività degli utenti. A essi non si chiede di pensare, ma solo di trovare il modo migliore per realizzare ciò che altri hanno pensato, senza chiedersi perché. Io non credo che questo sia un modello corretto per il mestiere del progettista. La progettazione, nel senso più nobile del termine, è l'arte di cambiare il mondo. Il progettista deve conoscere approfonditamente le possibilità che la tecnologia offre, e le tecniche per utilizzarla nel modo più appropriato. Ma questa conoscenza è soltanto strumentale, e non un obiettivo in sé. Ciò che conta è lo scopo finale, una visione del futuro che si ritiene desiderabile, e che il prodotto della progettazione rende più vicino.

Questo libro, pertanto – sia pure con l'approfondimento limitato che può fornire un testo introduttivo – invoca un radicale *cambiamento di paradigma* nelle discipline della progettazione, che da progettazione orientata ai sistemi si trasformi in progettazione orientata all'uso e, possibilmente, all'uso *universale*. Ciò significa – al di là delle differenze fra le metodologie proposte dai diversi autori – una focalizzazione costante, consapevole, informata e attenta sui bisogni degli utilizzatori dei sistemi che progettiamo, sui diversi contesti del loro uso e sugli effetti che essi producono. Che il progettista alzi, finalmente, lo sguardo dal tavolo di lavoro (o dallo schermo del computer) e si guardi intorno. E che l'università, finalmente, lo aiuti in questo cambiamento di prospettiva.

° ° °

Ogni libro porta dietro di sé il mondo e l'esperienza del suo autore. La mia formazione di base è l'ingegneria del software. Anche se mi sono a lungo occupato di problemi di organizzazione e di comunicazione, è principalmente dal punto di vista di chi progetta software che, inevitabilmente, affronto i temi dell'ingegneria dell'usabilità e dell'interaction design. In particolare, il libro pone molta enfasi sull'approccio iterativo alla progettazione, con l'utilizzo di prototipi e di prove con gli utenti fin dalle primissime fasi del progetto. Non propone alcuna specifica metodologia di progettazione, ma adotta l'approccio generale alla progettazione *human-centred* proposto dallo standard ISO 13407. I concetti base sono ispirati agli standard ISO (dai quali si adotta anche la definizione di usabilità) e all'impostazione dei primi libri di Donald Norman. Il libro evita di trattare temi soggetti a una rapida obsolescenza (per esempio i diversi apparati di interazione) e riduce al minimo indispensabile il trattamento dei concetti di psicologia cognitiva e della percezione. Questo non perché io ne sottovaluti l'importanza in relazione ai temi trattati, ma semplicemente perché, provenendo da una formazione differente, ho voluto evitare le approssimazioni del dilettante. Ho prestato molta attenzione – sia pure lasciandole inevitabilmente nel background del testo – alle tendenze recenti del Web, che costituiscono attualmente l'oggetto principale del mio interesse.

Detto questo, l'organizzazione del libro in capitoli non richiede particolari spiegazioni. Dopo un'introduzione generale sul concetto di interfaccia utente e sulla disciplina della human-computer interaction (capitolo 1), si prosegue con una rapida rassegna dei principali paradigmi di interazione che si sono consolidati negli anni. Segue una introduzione alla nozione di usabilità (capitolo 3), alle motivazioni alla base di una progettazione human-centred (capitolo 4) e qualche approfondimento sull'utente, che da concetto astratto deve incarnarsi, e diventare il fulcro della progettazone. I capitoli 6,7,8 e 9 approfondiscono i metodi dell'ingegneria dell'usabilità, descrivendo inizialmente l'approccio iterativo alla progettazione dei sistemi (capitolo 6), quindi i metodi utilizzati per la definizione dei requisiti (capitolo 7), e la costruzione di prototipi (capitolo 9). Il capitolo 8 discute il rapporto fra ingegneria e creatività, mostrando esempi delle tecniche utilizzate nell'invenzione di nuovi prodotti. I quattro capitoli successivi discutono, con numerosi esempi, i principi e le linee guida da seguire nella progettazione dei sistemi usabili. In particolare, il capitolo 10 descrive dettagliatamente i sette principi del dialogo proposti dallo standard ISO 9241-Parte 110; il capitolo 11 approfondisce le linee guida per il trattamento degli errori dell'utente; il capitolo 12 si concentra in particolare sulla progettazione grafica e, infine, il capitolo 13 indica come progettare testi dotati di buona leggibilità. Il capitolo 14 tratta le principali tecniche per la valutazione dell'usabilità e, in particolare, i test di usabilità. Il libro non riporta una bibliografia estesa, che spesso serve solo a dimostrare la cultura dell'autore, e non è di aiuto a chi vuole approfondire. Per evitare questo problema, mi sono limitato a indicare alcuni libri che ritengo utili per approfondire i temi trattati. L'appendice descrive la notazione degli *state-chart*, strumenti formali molto utili per la descrizione non ambigua dei dialoghi fra utenti e sistema.

° ° °

Il libro nasce dall'esperienza di dieci anni d'insegnamento del corso di Interazione uomo-macchina per la laurea in Informatica dell'Università degli Studi di Milano Bicocca. Questo corso ha l'obiettivo di trasmettere agli studenti (in

buona parte futuri progettisti di sistemi) una sensibilità alle problematiche della costruzione di sistemi usabili - e in particolare di prodotti software, o con un significativo contenuto di software. A questo scopo, insiste sui concetti di usabilità, di progettazione human-centred e sui metodi di valutazione dell'usabilità che coinvolgono l'utente.

Un corso con quest'obiettivo deve dare spazio consistente a esperienze pratiche di progettazione, senza le quali non sarebbe di alcuna utilità. Infatti, l'esperienza mostra che gli studenti tendono a sottovalutare questi concetti, considerandoli quasi banali, nel contesto degli altri corsi a orientamento più tecnico e formale presenti nei curriculum delle lauree in Informatica, senza soffermarsi a esaminarne con la necessaria attenzione il senso profondo, che banale certamente non è. Per superare questa difficoltà, è indispensabile affiancare alla presentazione dei concetti un laboratorio di tipo pratico, in cui gli studenti possano vederne le applicazioni e le implicazioni concrete. Secondo la mia esperienza, il modo più proficuo è organizzare gli studenti in piccoli gruppi di lavoro che progettino, con l'impostazione iterativa descritta nel testo, il prototipo di un semplice sistema, sottoponendolo, a ogni ciclo d'iterazione, a test con gli utenti o a valutazioni di usabilità effettuate assieme al docente.[1]

Anche con questa impostazione, si scopre ben presto che trasmettere, nell'ambito di un corso della durata di 6 o 8 crediti formativi, una ragionevole capacità di impostare *consapevolmente* l'interfaccia di un semplice sistema interattivo, è compito didattico non banale. Naturalmente, non mancano gli studenti in grado di produrre prototipi eccellenti. Questo deriva quasi sempre dalla disponibilità di prodotti interattivi di qualità, che vengono presi a modello o che comunque costituiscono precise fonti di ispirazione. Ma non basta che il progettista software sia in grado di progettare una buona interfaccia copiandola dal suo cellulare o dal suo iPhone: così non produrrà mai innovazione. Il buon design è la risultante dell'applicazione *consapevole* di principi generali e ben noti, e dell'applicazione, ancora una volta consapevole, di un processo di progettazione pianificato per questo specifico scopo. Questo il progettista inesperto non lo sa fare, e non serve che, qualche giorno prima, il docente gli abbia spiegato, in astratto, come si fa. La difficoltà sta nel fatto che i principi del buon design sono molto generali, e per riconoscerli e applicarli nelle situazioni specifiche serve molta esperienza. Il principiante vedrà le difficoltà, ma non sarà in grado di individuarne le cause, cioè le decisioni di progetto sbagliate che, spesso, producono conseguenze negative che si manifestano molto più avanti. Il bravo designer sa che una deroga a un principio importante produrrà inevitabilmente, prima o poi, dei problemi, ed eviterà di farla. Lo studente non riconosce queste trappole, per le quali serve un occhio allenato. In pratica è necessario che il docente, seguendo da vicino ogni singolo progetto, gli mostri di volta in volta le conseguenze delle sue scelte e lo aiuti a dipanare le situazioni più complicate, spiegandogli *perché* si sono verificate e mostrandogli che, con scelte diverse, l'usabilità del prodotto migliora.

In conclusione, un corso sull'ingegneria dell'usabilità deve dare uno spazio consistente ad attività di laboratorio, senza le quali non sarebbe di alcuna utilità. Ecco perché questo libro, da solo, non è sufficiente. I concetti illustrati devono essere "fatti vivere" nella pratica della progettazione e della valutazione critica di specifici sistemi. La relativa snellezza di questo libro – al confronto con i testi più noti di Interazione Uomo Macchina, sempre molto corposi – è stata concepita proprio per lasciare spazio alla sperimentazione di laboratorio. In pratica, ciascun capitolo può essere svolto in una o due lezioni di due ore ciascuna, secondo il livello di approfondimento scelto, per un totale di circa 35-40 ore di lezione, corrispondenti a 4 o 5 crediti formativi. Questo lascia ampio spazio alle attività di progettazione (e, soprattutto, al confronto periodico con il docente) e, a seconda dell'ampiezza del corso, ad approfondimenti o complementi che potranno essere scelti a discrezione del docente.

° ° °

Questo libro è pubblicato in due edizioni: una edizione a stampa, acquistabile in libreria, e una edizione elettronica. Per la edizione a stampa, tutti i diritti sono riservati, a norma di legge, all'editore. La edizione elettronica è resa disponibile,

[1] Nel caso specifico del corso citato, i prototipi vengono realizzati prima su carta e quindi con PowerPoint (o altro prodotto simile), che permette di definirne in modo piuttosto preciso l'aspetto visivo, e di ottenere rapidamente un prototipo navigabile, collegando fra loro le varie schede con link ipertestuali. Anche se questo strumento non è stato concepito per attività di prototipazione, i risultati sono eccellenti, e sicuramente superiori rispetto all'uso di altri strumenti che, oltre a richiedere un addestramento specifico, risultano troppo "invasivi", orientando in modo eccessivo le soluzioni di design (Flash, generatori di pagine Web, ecc.).

gratuitamente, sul Web con licenza Creative Commons "Attribuzione - Non commerciale - Condividi allo stesso modo - 2.5 Italia":[2]



Questo significa, in pratica, che il testo può essere liberamente modificato, aggiungendo o eliminando delle parti per adattarlo a specifiche esigenze didattiche, a patto che il prodotto risultante sia reso disponibile gratuitamente con lo stesso tipo di licenza, e che ne sia citata la fonte e l'autore originale. La versione è elettronica è raggiungibile a partire dal sito dell'autore www.rpolillo.it o dell'editore www.apogeonline.com.

° ° °

Nella stesura di questo libro ho usato diverso materiale pubblicato in miei precedenti lavori. In particolare, il libro è un ampliamento del mio capitolo sull'*Introduzione all'Ingegneria dell'Usabilità,* contenuto nel libro *Human Computer Interaction – Fondamenti e prospettive*, a cura di Alessandro Soro, pubblicato nel 2008 dalla casa editrice Polimetrica, anche accessibile gratuitamente in rete. Questo testo, composto da una serie di rassegne monografiche a scopo didattico sui principali temi della HCI, può essere adottato come utile complemento al presente libro. Inoltre, l'appendice sugli *state-chart* è tratta dal mio precedente *Plasmare il Web – Road map per siti di qualità*, edizioni Apogeo, 2006. Da quest'ultimo testo, che adotta un approccio e una terminologia completamente coerenti con questo libro, sono state tratte anche diverse pagine del capitolo relativo alla stesura dei requisiti.

Devo molto agli studenti dei miei corsi di Interazione Uomo Macchina e di Laboratorio Internet all'Università di Milano Bicocca. Non solo perché mi hanno costretto, negli anni, ad approfondire meglio i concetti esposti a lezione ma soprattutto perché mi hanno fornito, con i loro progetti - molte centinaia, ciascuno un piccolo caso di studio – un gran numero di esempi concreti di progettazione di sistemi interattivi. Ciò mi ha permesso di comprendere le difficoltà che un progettista alle prime armi incontra nella progettazione della *user experience*, e quindi di identificare le priorità dei vari argomenti dal punto di vista didattico. Diverse illustrazioni del libro sono tratte da questi progetti. In particolare, il prototipo di Figura 158, Figura 165 e Figura 166 è stato realizzato da Stefano Berbenni, Regina Fantucci e Debora Gerini. Quello di Figura 167 da Marco Brenna, Domenico Pierro Domenico e Simone Songia. Il prototipo di Figura 164è stato realizzato da Simone Mandelli, Marco Pelucchi e Marianna Riceputi. Il prototipo di Figura 163, e le relative registrazioni video dei dei test di usabilità (Figura 284 e Figura 285) sono stati realizzati da Giuseppe Pogliani, Giuseppe Ragno e Giuseppe Maggi, e si riferiscono al prototipo. A tutti vanno i miei ringraziamenti.

I commenti e i suggerimenti di Luca Colombo, e la sua tesi di laurea magistrale sulla *New Web Typography* (da cui ho tratto diverse figure) mi sono stati molto utili per il capitolo sulla tipografia e la usabilità dei testi.

Devo molto anche all'amico Piero Schiavo Campo, che ha condiviso con me, per molti anni, il compito di esaminare e commentare i progetti di entrambi i corsi, durante le revisioni nelle esercitazioni e negli esami di fine corso. Il fatto che, in queste revisioni, i nostri commenti su uno stesso progetto fossero spesso del tutto differenti, mi ha costretto ad accettare il fatto che, anche in questa disciplina, le certezze assolute non esistono, e la qualità nasce sempre dalla discussione e dal confronto delle diverse opinioni.

[2] Per il significato dei simboli, si veda http://creativecommons.org/licenses/by-nc-sa/2.5/it/. Il testo della licenza si trova in http://creativecommons.org/licenses/by-nc-sa/2.5/it/legalcode.

Infine, sono molto grato a mia moglie Patricia Caprotti la quale, anche in questo libro, si è sobbarcata il compito di rivedere il testo per renderlo più scorrevole ed eliminare i termini gergali che il mondo dell'ingegneria del software produce a getto continuo.

Sarò molto grato a chi volesse ampliare o produrre versioni successive di questo testo.

Roberto Polillo
polillo@disco.unimib.it
www.rpolillo.it

1. Sistemi interattivi e interfacce d'uso

Sintesi del capitolo

Questo capitolo introduttivo ha lo scopo di collocare i temi di questo libro nello scenario dell'evoluzione degli strumenti dell'uomo, e in particolare di quelli interattivi, dotati di elevata intelligenza. Dopo avere definito il concetto d'interfaccia d'uso, si osserva come, con la crescita della complessità dei sistemi, il ruolo dell'interfaccia sia diventato sempre più importante, non solo come mezzo di governo, ma anche come strumento di semplificazione, per trasmettere all'utente una visione del sistema coerente con le sue specifiche necessità. Promuovere la nozione di semplicità e facilità d'uso fra chi progetta e produce sistemi complessi può allora contribuire in modo significativo a migliorare la qualità della nostra vita e a ridurre il divario digitale derivante dalle differenze di età, istruzione e censo degli utilizzatori della tecnologia. Comprendere come si possano progettare sistemi facili da usare è uno dei compiti della disciplina della Human-Computer Interaction, di cui si menzionano gli obiettivi e alcune tappe fondamentali.

Sistemi e interfacce

L'essere umano, nella sua storia, si è sempre dotato di *strumenti* che gli permettessero di svolgere compiti che con il solo impiego delle sue doti di natura sarebbero stati impossibili. Gli utensili (e, fra questi, anche le armi) gli hanno permesso di vivere con i proventi della caccia, dell'allevamento e della coltivazione della terra. Gli hanno permesso di preparare cibi e indumenti, costruire abitazioni confortevoli, difendersi dai nemici, o aggredirli, creare musica. Armi e utensili hanno costituito, negli ultimi millenni, delle *protesi* artificiali, che hanno permesso all'*homo sapiens* di superare le limitazioni fisiche del suo corpo e di aumentarne le capacità.

Fino a un passato non lontanissimo, queste protesi erano di natura relativamente semplice. Il coltello, l'aratro, la spada, le frecce, il tamburo, pur potenziando enormemente le possibilità del corpo umano, permettevano di svolgere compiti ancora strettamente legati alle sue capacità meccaniche, e che da queste non potevano prescindere. L'uso di questi strumenti richiedeva l'acquisizione di abilità manuali specifiche, che non di rado presupponevano un lungo addestramento. Negli ultimi secoli, l'evoluzione della tecnologia, e soprattutto la scoperta delle tecniche per la produzione di energia (la trazione animale, la macchina a vapore, l'elettricità, …) hanno radicalmente cambiato questo scenario. Sono stati progettati strumenti capaci di svolgere compiti sempre più complessi e, soprattutto, capaci di operare in modo *autonomo*, alimentati da fonti di energia non provenienti dal corpo umano. Non più protesi del nostro corpo, quindi, ma sistemi da governare attraverso appositi meccanismi di vario tipo (leve, pulsanti, quadri di controllo).

Ma è soltanto da pochi decenni che la situazione è, ancora una volta, profondamente mutata. L'informatica ha permesso di dotare questi sistemi non solo di autonomia, ma anche di *intelligenza*, attraverso una componente software che, negli anni, è divenuta sempre più evoluta e pervasiva. Essa permette a questi sistemi di eseguire procedure complesse e prendere decisioni in modo autonomo, sulla base delle diverse condizioni che si verificano durante il loro funzionamento.

L'utilizzo di questi sistemi non richiede più l'acquisizione di abilità manuali specifiche, ma avviene attraverso la mediazione di *interfacce d'uso* appositamente progettate, che permettono una interazione anche molto stretta con il suo utilizzatore. Il governo di questi sistemi da parte dell'uomo prende sempre più la forma di un *dialogo* fra due partner intelligenti (Figura 1). L'interazione, che nei sistemi più semplici richiedeva all'utente abilità motorie di carattere elementare (premere un pulsante, alzare una leva), nei sistemi più evoluti avviene sempre più spesso a livello cognitivo. In altre parole, il dialogo fra utente e sistema implica sempre più, da parte di entrambi gli interlocutori, l'esecuzione di ragionamenti complessi.

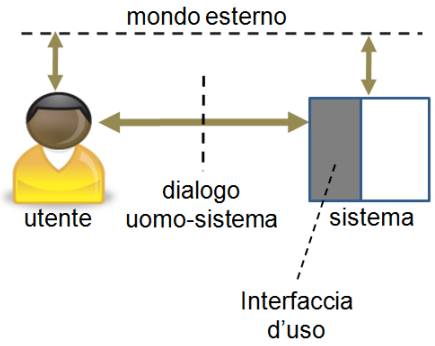


Figura 1. L'interfaccia d'uso

I termini "sistema interattivo", "interfaccia d'uso", "dialogo", "interazione" in questo libro saranno usati molto spesso: è quindi opportuno definirli con precisione. Adotteremo le definizioni dell'ISO 9241, lo standard principale relativo all'usabilità dei sistemi interattivi, di cui parleremo più diffusamente nel capitolo 10.[3]

Per *sistema interattivo* intendiamo, in modo del tutto generale, qualsiasi "combinazione di componenti hardware e software che ricevono input da un utente umano, e gli forniscono un output, allo scopo di supportare l'effettuazione di un *compito*". Questa definizione è molto ampia, e comprende tutti i sistemi che possono interagire con un utente umano, da quelli più semplici (come un frullatore o un robot da cucina) a quelli più complessi, come un telefono cellulare, il cruscotto di un aereo, un sistema di realtà virtuale (Figura 2). In pratica, la definizione esclude solamente quei sistemi che interagiscono esclusivamente con altri sistemi, senza alcun intervento umano, come i sistemi di controllo di processo "a ciclo chiuso", che intervengono sul processo controllato senza alcun intervento dell'operatore.

Per *interfaccia d'uso* (o *interfaccia utente, user interface*) intendiamo l'insieme di "tutti i componenti di un sistema interattivo (software o hardware) che forniscono all'utente informazioni e comandi per permettergli di effettuare specifici compiti attraverso il sistema."

Con il termine *compito* (in inglese, *task*), di uso molto frequente in questo contesto, si intende infine qualsiasi "insieme di attività richieste per raggiungere un risultato."[4]

[3] L'ISO 9241 è composto da un insieme di documenti molto ampio, in evoluzione da una ventina d'anni. Inizialmente, esso trattava essenzialmente gli aspetti ergonomici dei terminali video utilizzati per il lavoro di ufficio, ed aveva come titolo *Ergonomic requirements for office work with visual display terminals (VDTs)*. Da allora, gli obiettivi si sono ampliati, ed è in corso un processo di revisione dell'intero standard, rinominato di recente, più genericamente, *Ergonomics of Human-System Interaction*.

[4] Le definizioni di interactive system, task, user interface, dialogue sono tratte (in nostra traduzione) dal documento *ISO 9241 Part 110: Dialogue principles*.



Figura 2. Esempi di sistemi interattivi
(a) Cruscotto aereo; (b) realtà virtuale immersiva; (c) Apple iPhone; (d) Robot da cucina

L'ISO 9241 preferisce usare il termine *dialogo*, al posto del più generico – ma equivalente - termine *interazione* definendolo come "l'interazione fra un utente e un sistema interattivo, intesa come una sequenza di azioni compiute dall'utente (input) e di risposte del sistema (output), allo scopo di raggiungere un certo obbiettivo" (Figura 3).

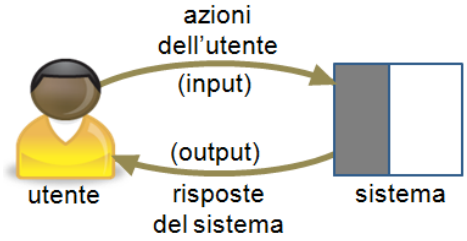


Figura 3. Il dialogo utente-sistema

Il dialogo fra un utente e un sistema interattivo può essere realizzato attraverso svariati *dispositivi d'interazione,* come suggeriscono gli esempi di Figura 2. Da un lato, il sistema può utilizzare una varietà di *dispositivi di output*, i cui messaggi sono raccolti dai sensi dell'utente (vista, udito, tatto). Dall'altro, l'utente può governare il sistema utilizzando

vari *dispositivi di input*: digitando i dati su tastiere o utilizzando dispositivi di manipolazione di vario tipo, la sua voce o, più raramente, lo sguardo o la postura del suo corpo (Figura 4).

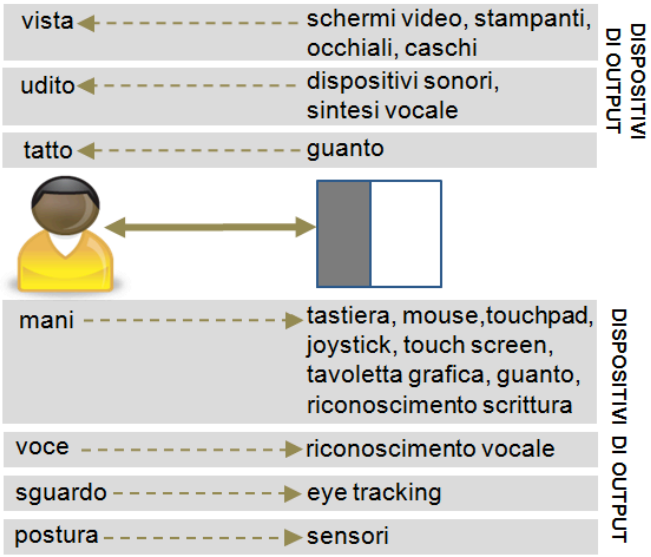


Figura 4. Alcuni dispositivi di interazione

Le dimensioni della complessità

Si è detto più volte che i sistemi interattivi oggi possono essere molto complessi. È ora opportuno precisare questo concetto. Infatti, un sistema può essere considerato complesso per aspetti diversi: perché composto da molti componenti che interagiscono fra loro in modo complicato, oppure perché è destinato a supportare numerose attività. In altre parole, perché permette al suo utilizzatore di fare molte cose diverse. Per il primo caso, possiamo usare il termine di *complessità interna* (o *strutturale*), per il secondo quello di *complessità esterna,* o *funzionale*. Per esempio, un coltello da lancio è molto semplice, sia dal punto di vista interno che da quello funzionale: è composto soltanto da una lama e da un manico, e serve a un solo scopo, quello di colpire un bersaglio lontano. Una sega elettrica da boscaiolo è più complessa dal punto di vista interno, perché costituita da numerosi componenti fra loro interagenti: un motore, un meccanismo di trasmissione del movimento, due lame mobili, un interruttore. Mantiene tuttavia una relativa semplicità funzionale: il suo scopo è pur sempre quello di tagliare, anche se, data la complessità interna, deve permettere di compiere alcune semplici funzioni collaterali, quali per esempio l'avvio e l'arresto del motore. Infine, un *iPhone*, col suo ricco corredo di funzionalità, realizzate attraverso tecnologie sofisticate, è molto complesso sia funzionalmente sia strutturalmente.

Queste due dimensioni della complessità dei sistemi non sono necessariamente fra loro correlate: esistono sistemi internamente semplici ma funzionalmente complessi (per esempio, un temperino da tasca multi-uso, con il suo corredo di lame e di arnesi estraibili), ed esistono sistemi internamente complessi ma funzionalmente semplici. Per esempio, un orologio da parete: dentro è molto complicato, ma ha l'unico scopo di indicare l'ora. D'altro canto, la complessità interna genera spesso una certa complessità funzionale. Infatti, se un oggetto è internamente complesso, potrebbero verificarsi molti possibili malfunzionamenti di diverso tipo. Questi malfunzionamenti si renderanno, in ultima analisi, visibili all'utente, che dovrà intraprendere le opportune azioni correttive.

Possiamo mettere a confronto la complessità funzionale e strutturale di un insieme di prodotti utilizzando un semplice diagramma come in Figura 5.

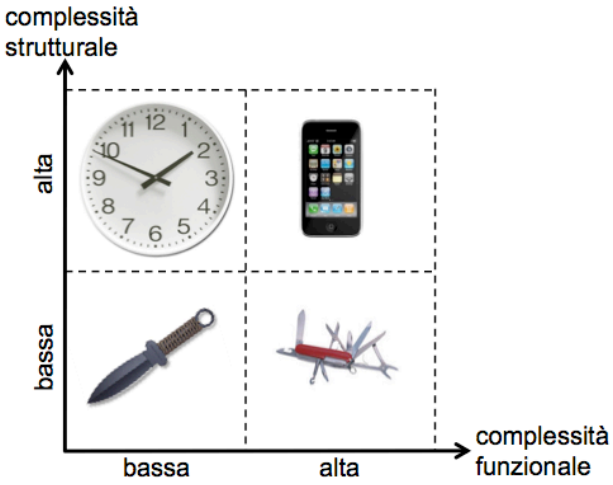
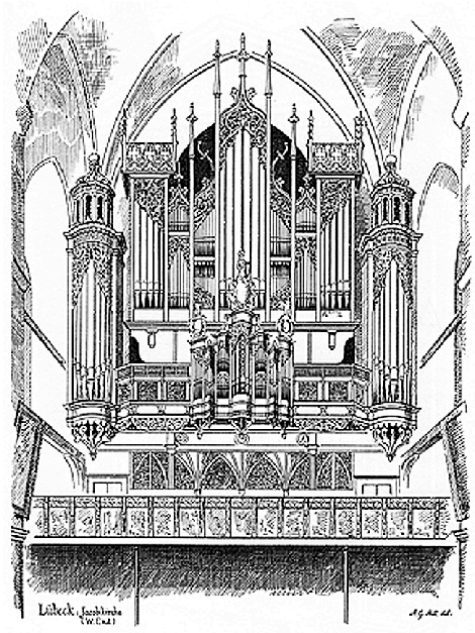


Figura 5. Complessità funzionale e strutturale a confronto

I manufatti complessi non richiedono necessariamente una tecnologia avanzata, e quindi non sono propri della nostra epoca. Basti pensare alla complessità di certi strumenti musicali (Figura 6), alle macchine di Leonardo da Vinci, alle macchine a vapore. Tuttavia questi, per quanto complessi, non possono raggiungere la complessità dei sistemi che usano le tecnologie del software. Le moderne applicazioni software possono raggiungere complessità strutturali e funzionali elevatissime. Per misurarle, sono state definite varie metriche. Tipicamente, la complessità strutturale di un programma può essere misurata contando le linee di codice sorgente che lo costituiscono. Per misurarne la complessità funzionale, invece, è stato definito il concetto di *punto funzione* (*function point*), unità di misura delle funzionalità visibili all'utente, indipendente dalla particolare tecnologia utilizzata per realizzarle.[5]



[5] I punti funzione misurano la dimensione di un sistema software quantificando le funzionalità fornite all'utente sulla base solamente del progetto logico e delle specifiche funzionali, indipendentemente dal linguaggio di programmazione usato per l'implementazione. Il conteggio dei punti funzione può avere diversi obiettivi: misurare la complessità funzionale del sistema; stimarne i costi di sviluppo e di manutenzione; fornire una misura normalizzata che permetta di confrontare progetti e organizzazioni di sviluppo diversi (cfr. il sito dell'International Function Point User Group, in http://www.ifpug.org/).

Figura 6. Un organo rinascimentale

Non è questa la sede per approfondire questi concetti, che ci porterebbero lontano. Ci limitiamo a fornire, nella tabella di Figura 7, il numero dei punti funzione e delle linee di codice sorgente di alcuni noti sistemi software. [6] Da queste analisi si vede chiaramente che il rapporto fra i due valori è molto variabile, dipendendo dal linguaggio di programmazione utilizzato, dall'intrinseca complessità degli algoritmi e dalla produttività media dell'organizzazione in cui il sistema viene sviluppato.

SISTEMA	FP ($\times 10^3$)	SLOC ($\times 10^6$)	SLOC/FP
U.S. Air Traffic Control	306	65.3	213
SAP	297	23.7	80
MS Vista	158	10.1	64
MS Office Professional	93	6.0	64
MS Word 2007	3	0.2	64
MS DOS	1.3	0.3	213
Google search engine	19	1.2	64
Amazon.com	18	0.5	27
Mozilla Firefox	1.3	0.075	53
F115 avionics package	22	2.1	91

Figura 7. Linee di codice sorgente (*source lines of code*, SLOC) e punti funzione (*function points*, FP) di alcuni sistemi software (fonte: Capers Jones, cit.)

Consideriamo ora la *complessità d'uso* di un sistema, cioè la maggiore o minore facilità con cui siamo in grado di utilizzarlo. Diremo che la sua complessità d'uso è bassa se esso è facile da usare. Preciseremo meglio questo concetto nel capitolo 3, con la definizione della nozione di usabilità: per ora, accontentiamoci del significato che tutti noi, intuitivamente, attribuiamo a queste locuzioni.

Anche se la cosa può sembrare contro-intuitiva, complessità funzionale e complessità d'uso sono concetti diversi, e largamente indipendenti. Un sistema può realizzare molte funzioni, ma essere facile da usare. D'altro canto, esistono sistemi funzionalmente semplici che creano grosse difficoltà a chi li usa. Per esempio, un sistema elementare come il coltello da lancio in Figura 5 richiede, per essere usato con precisione, grande destrezza, ottenibile solo con un lungo allenamento. Invece, l'iPhone, funzionalmente molto ricco, è considerato di solito molto facile da usare.

La Figura 8 confronta i quattro oggetti già visti in Figura 5 dal punto di vista della complessità funzionale e d'uso. In questo diagramma, il temperino multi-funzione è stato considerato difficile da usare, poiché chi scrive ne possiede uno che non si riesce ad aprire senza spuntarsi le unghie. Ovviamente, altri prodotti di questo tipo potrebbero non avere questo difetto, ed essere quindi valutati diversamente.

[6] Da Capers Jones, *A New Business Model for Function Point Metrics*, Version 7.0, maggio 2008, disponibile in rete

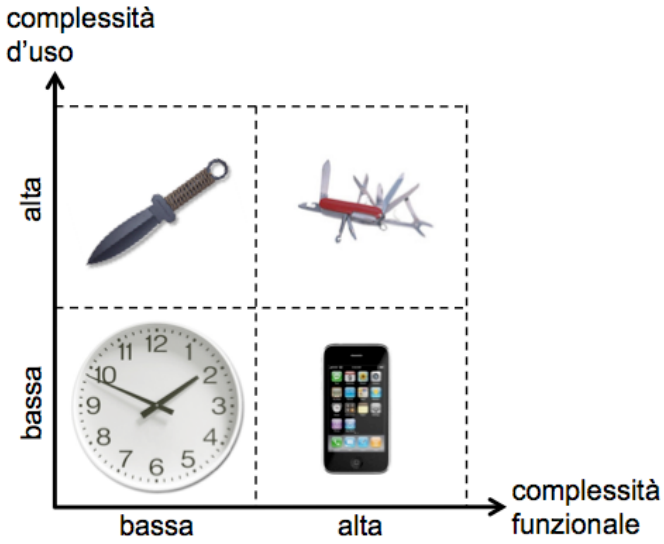


Figura 8. Complessità funzionale e d'uso a confronto

In sintesi, esistono quindi tre dimensioni, fra loro largamente indipendenti, della complessità di un sistema: la complessità strutturale, la complessità funzionale e la complessità d'uso (Figura 9).

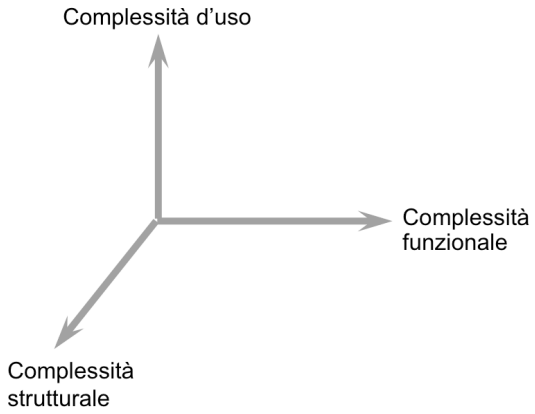


Figura 9. Le tre dimensioni della complessità di un sistema

La diversità degli utenti

Finora abbiamo esaminato i vari aspetti della complessità dei sistemi interattivi, senza mai soffermarci sulla natura dell'utente. In tutte le nostre figure, l'utente è stato sempre rappresentato nello stesso modo, con la stessa immagine astratta di Figura 1. Ma nella realtà non è così. Tutti gli utenti sono *diversi*. Nonostante le tendenze all'uniformità prodotte dalla globalizzazione, le diversità presenti fra gli abitanti del pianeta sono – fortunatamente – molto numerose. Basti pensare alla diversità linguistica: nella sola Unione Europea, anche se le lingue ufficiali sono "solo" 23, si parlano fra le 30 e 40 lingue native (il numero varia secondo la nozione di lingua che si utilizza). Fra queste, i parlanti nativi della lingua a diffusione massima, il tedesco, non superano il 20% dell'intera popolazione. Anche all'interno di ogni singolo Paese la frammentazione culturale è in aumento, a causa dell'immigrazione. In Italia, dove pure il fenomeno dell'immigrazione è relativamente recente, gli stranieri sono oggi più del 6% della popolazione italiana, e in continua crescita.

Gli utenti non sono solo diversi nelle loro caratteristiche individuali (lingua, cultura, scolarità, abitudini, preferenze, ecc.), ma anche nei rapporti con il sistema. Ogni utente chiede al sistema cose diverse, e si rapporta con esso in modo

specifico, differente da quello di altri utenti. Lo stesso word processor è utilizzato dallo studente per scrivere la tesi di laurea, dal giornalista per i propri articoli, dal romanziere, dal saggista, dall'impiegato che lo usa per compilare le fatture della propria ditta. Ciascuno parla la propria lingua e utilizza il lessico specifico della sua professione, utilizza i caratteri del proprio alfabeto, possiede cultura e abitudini derivanti dalla propria specifica formazione. Alcuni sono mancini, altri no. Alcuni hanno difficoltà nella lettura di caratteri molto piccoli, altri ci vedono bene. Ciascuno chiede al sistema di supportarlo nell'esecuzione di compiti specifici, che non sono gli stessi di altri utenti. Alcuni utilizzano il prodotto con un computer potente, altri dispongono soltanto di una macchina obsoleta, che può avere delle limitazioni. Alcuni sono giovani, e abituati fin da piccoli all'utilizzo degli strumenti dell'informatica, altri sono anziani, e possono avere nei confronti dello strumento un atteggiamento di diffidenza o di paura.

Lo stesso temperino multi-uso della Figura 8, che chi scrive ha giudicato difficile da usare, potrebbe essere valutato diversamente da utenti con un diverso livello di manualità.

In definitiva, la diversità degli utenti pone altri problemi di complessità d'uso, che non sono intrinseci allo strumento, ma derivano dall'interazione fra lo strumento e il suo utente. Interazione che deve avvenire con modalità diverse a seconda delle caratteristiche e delle necessità di ogni particolare utilizzatore.

In un mondo di apparati semplici e di esigenze uniformi, la semplicità delle interfacce d'uso può essere considerata un problema minore. Nel mondo di oggi, caratterizzato da un'enorme varietà di strumenti funzionalmente ricchi, progettati per risolvere problemi complessi di utenti fra loro molto differenti, il problema di disporre di interfacce d'uso adeguato diviene, come vedremo meglio nel seguito, assolutamente critico.

La velocità del cambiamento

L'ambiente della nostra esistenza quotidiana si fa sempre più complesso. Nel lavoro e nel tempo libero dobbiamo interagire con prodotti tecnologici sempre più sofisticati, il cui uso richiede spesso capacità e competenze non banali. Capacità e competenze che, peraltro, divengono rapidamente obsolete, perché i prodotti della tecnologia evolvono di continuo e in modo sempre più rapido.

Chi ha consolidato le proprie abitudini di vita con l'utilizzo di certi strumenti deve continuamente modificarle, per imparare a usare gli strumenti delle generazioni tecnologiche successive. Questi sono spesso radicalmente differenti, non soltanto dal punto di vista delle modalità di interazione, ma anche perché inducono comportamenti del tutto diversi nei loro utilizzatori. Basti pensare a come è cambiato l'uso del telefono in Italia nei dieci anni a cavallo della fine del secolo scorso. Fino al 1995, il telefono era sostanzialmente disponibile solo da postazioni fisse, in ufficio, a casa, nei locali pubblici o nelle cabine telefoniche in strada. Oggi, in Italia, il numero medio di abbonamenti al telefono cellulare pro-capite è addirittura superiore all'unità. In strada, la gente telefona mentre cammina, guida o viaggia nei mezzi pubblici. Gli sms sono stati introdotti nel 1993, e oggi non ne possiamo fare a meno. Secondo Wikipedia, solo dieci anni dopo, in tutto il mondo, ne venivano inviati circa 500 miliardi ogni anno. Dopo altri 5 anni, si stima che fossero 2500 miliardi. Parallelamente all'evoluzione del telefono, a partire dai primi anni '90, la posta elettronica ha modificato radicalmente le modalità della comunicazione scritta. Più recentemente, con i siti di social network e con il microblogging, altri canali di comunicazione hanno acquisito enorme diffusione, affiancandosi agli altri senza sostituirli. Oggi tutti noi comunichiamo con modalità sostanzialmente differenti da quelle di solo due decenni fa. Questa forte accelerazione del cambiamento è sotto gli occhi di tutti, ed è sofferta, a volte in modo drammatico, soprattutto dalle persone più anziane, che spesso non sono in grado di adattare i loro comportamenti al nuovo contesto tecnologico.

La Figura 10 mostra tre gruppi di strumenti tipici, del lavoro e del tempo libero, del 1920, 1965 e 2010: un riproduttore di musica, un apparecchio telefonico e un sistema per scrivere. Ciascun gruppo appartiene a un mondo totalmente diverso da quello degli altri, eppure non è difficile trovare, ancora oggi, chi è entrato in contatto con tutti e tre. Per esempio, l'autore di questo libro (che appartiene alla generazione formatasi negli anni '60), lo ha digitato su un laptop molto evoluto, ma ha imparato a dattilografare, da bambino, sulla macchina per scrivere del nonno (classe 1883), molto simile a quella del 1920 in figura.

La distanza temporale fra il primo e il secondo gruppo e fra il secondo e il terzo è la stessa: 45 anni, ma la complessità funzionale è molto diversa. Fra il primo e il secondo gruppo la tecnologia si perfeziona, ma le funzioni sono,

sostanzialmente, le stesse. Il telefono serve sempre per telefonare, la macchina per scrivere per comporre testi. Il fonografo serve per riprodurre dei suoni registrati, anche se il supporto fisico e la tecnologia di riproduzione sono completamente cambiati. Fra il secondo e il terzo gruppo, invece, è avvenuto il passaggio all'era digitale, e i cambiamenti sono incommensurabilmente più profondi. La forma degli strumenti non ce ne suggerisce più l'uso. La musica, il testo, la voce si sono digitalizzate, e vengono gestite dal software, che mette a disposizione dell'utente una varietà di funzioni del tutto nuove. L'iPhone in figura può servire per telefonare, ma è sostanzialmente un computer general purpose, come il netbook che gli sta accanto, ma più piccolo e più mobile. È geolocalizzato e in grado di percepire il proprio orientamento. È sensibile al tocco e può connettersi a reti di tecnologie diverse: la rete cellulare a larga banda e internet. Entrambi gli apparati possono essere utilizzati per riprodurre musica e comporre un testo da stampare su carta con l'uso della stampante laser. Ma le possibilità sono molte di più. Per esempio, entrambi permettono l'accesso alle informazioni e ai molteplici servizi disponibili in rete. Si tratta di cambiamenti drastici, che richiedono capacità di adattamento molto maggiori, e che inducono comportamenti nuovi.



Figura 10. Accelerazione dell'evoluzione degli strumenti quotidiani

Le cause di questa accelerazione così evidente dell'evoluzione dei prodotti tecnologici sono molteplici, e indicate schematicamente nella Figura 11. Innanzitutto, i bisogni degli utenti fanno nascere nuovi prodotti, i quali a loro volta inducono nuovi bisogni: l'esperienza d'uso ne suggerisce sempre delle modifiche migliorative, che si concretizzano in nuove versioni o in prodotti nuovi (ciclo a in Figura 11). Poi l'evoluzione della tecnologia offre continuamente nuove possibilità. Anche in questo caso c'è un ciclo di feedback: i prodotti esistenti stimolano la ricerca di innovazioni tecnologiche, e queste permettono miglioramenti ai prodotti (ciclo b). Questo fa sì che i prodotti esistenti siano continuamente rimpiazzati da prodotti che utilizzano tecnologie di nuova generazione. Rimpiazziamo la televisione analogica con quella digitale, e poi ancora quest'ultima con quella ad alta definizione. Rimpiazziamo le automobili con nuovi modelli meno inquinanti, e così via.

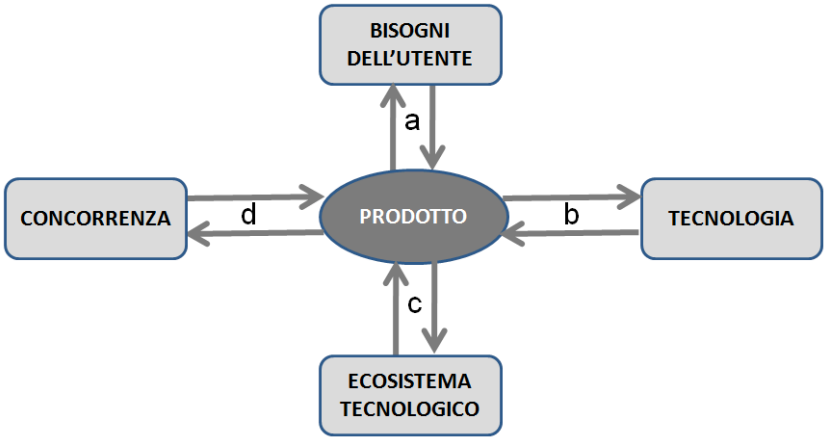


Figura 11. Le cause dell'evoluzione dei sistemi

La situazione dei personal computer è emblematica. È noto che l'evoluzione della tecnologia ha permesso di raddoppiare il numero di transistor per circuito integrato approssimativamente ogni due anni (legge di Moore, Figura 12). Poiché le prestazioni di molti componenti elettronici sono correlate alla densità dei transistor su un chip, questo ha prodotto, negli ultimi trent'anni, un enorme miglioramento delle prestazioni dei processori e delle capacità di memoria (Figura 13). Ciò ha permesso una continua e rapidissima diminuzione del rapporto costo/prestazioni. Per esempio, nel periodo 1993-2008, il costo delle memorie di massa si è quasi dimezzato ogni anno.[7] Il rovescio della medaglia è una continua e rapidissima obsolescenza dei sistemi, che impone agli utilizzatori un continuo ricambio dei prodotti. I PC devono, in pratica, essere sostituiti ogni 3-4 anni al massimo, perché obsoleti. L'evoluzione dei telefoni cellulari è anche più rapida: secondo alcune statistiche, la vita media di un cellulare, cioè l'intervallo di tempo dopo il quale l'utente lo sostituisce con uno nuovo, è di 23 mesi.

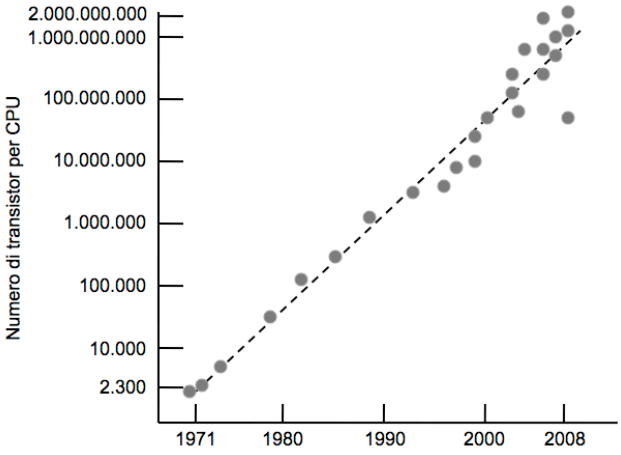


Figura 12. Legge di Moore

[7] Cfr. http://www.mattscomputertrends.com .

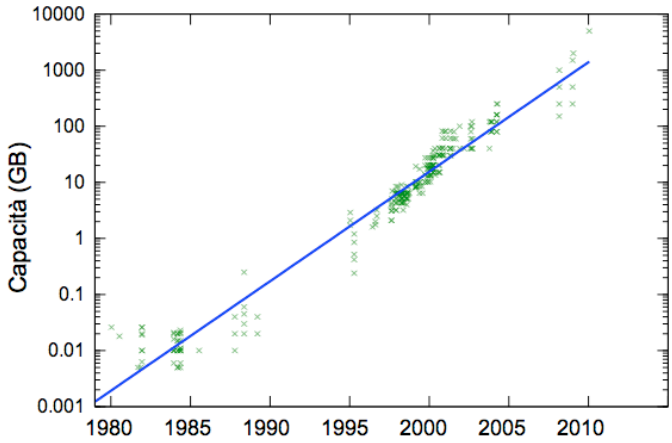


Figura 13. Crescita della capacità degli hard disk
(elaborazione di H-K Nienhuys per Wikipedia)

Un'ulteriore, imponente spinta verso un'evoluzione rapida e continua è data dalla forte competitività del mercato dei prodotti hi-tech, che obbliga i produttori a fornire versioni sempre più sofisticate di un prodotto per battere la concorrenza. Ogni prodotto stimola la concorrenza a superarne le prestazioni, e deve a sua volta superare i prodotti che nascono da questa competizione (Figura 11, ciclo d). Questi processi di competizione sono ulteriormente accelerati dall'evidente necessità dei produttori di mettere sul mercato versioni sempre più aggiornate dei loro prodotti, per alimentare un mercato di sostituzione che permetta di conservare – e di accrescere – i livelli di ricavi già raggiunti.

Infine, i prodotti dell'informatica e delle telecomunicazioni costituiscono un vero e proprio *ecosistema*, in perenne evoluzione. L'evoluzione di un prodotto produce la necessità di cambiamento nei prodotti a esso correlati, che devono adattarsi alle sue nuove caratteristiche, in un complesso sistema di condizionamenti reciproci (Figura 11, ciclo c). Ciò vale sia per le componenti hardware che per quelle software. I sistemi operativi devono poter supportare le nuove periferiche lanciate sul mercato, e le nuove periferiche devono essere compatibili con le nuove versioni dei sistemi operativi. Hardware e software devono conformarsi ai nuovi standard elaborati dai gruppi di lavoro dedicati a questo scopo, e i nuovi standard devono tener conto dei vincoli posti dai prodotti già sul mercato. Tecnologie nuove sono continuamente messe a punto e rendono possibili l'inserimento di funzionalità prima non realizzabili. Tecnologie vecchie divengono obsolete e sono sostituite dalle tecnologie di nuova generazione. E così via. Agli ecosistemi naturali abbiamo affiancato ecosistemi tecnologici, le cui "specie" sono in continua e rapida co-evoluzione. Questi ecosistemi costituiscono l'ambiente di tutte le nostre attività, e da essi non possiamo più prescindere.

Questa tendenza all'evoluzione accelerata dei prodotti software e all'iperfunzionalismo potrebbe ridursi, almeno in parte, con la trasformazione, in atto da tempo, del software da prodotto a servizio, erogato attraverso la rete (*software as a service*, *SaaS*). In questo caso, infatti, l'adattamento del software ai cambiamenti dell'ecosistema sarebbe effettuato a cura del fornitore del servizio, senza che l'utente debba necessariamente esserne consapevole. D'altra parte, i ricavi del fornitore del servizio non derivano più dalla vendita delle licenze per l'uso delle nuove versioni, più aggiornate e funzionalmente più ricche (come nel mercato tradizionale dei prodotti software), ma dal pagamento dei canoni del servizio, secondo il modello *pay per use*, o dalla pubblicità veicolata attraverso di esso. Non sarà quindi più costretto a proporre al mercato versioni sempre nuove di ogni prodotto, per mantenere costante o in crescita il flusso dei ricavi, ma sarà più interessato alla fidelizzazione degli utenti e alla crescita del loro numero.

W. Brian Arthur, economista e pioniere delle scienze della complessità, ha descritto molto bene, nel suo libro *The Nature of Technology*, i meccanismi che determinano l'evoluzione della tecnologia:

> *Non appena nuove tecnologie individuali sono prodotte, esse divengono potenziali building block per la costruzione di altre nuove tecnologie. Il risultato è una forma di evoluzione combinatoria, il cui meccanismo di base differisce da quello dell'evoluzione darwiniana standard. Le nuove tecnologie sono create da*

building block che sono essi stessi tecnologie, e diventano potenziali building block per la costruzione di ulteriori nuove tecnologie. Ciò che alimenta questa evoluzione è la conquista e lo sfruttamento di fenomeni nuovi, ma ciò richiede la disponibilità di tecnologie che ne permettano la conquista e lo sfruttamento. Da queste due affermazioni possiamo dire che la tecnologia crea se stessa a partire da se stessa. In questo modo l'insieme delle arti meccaniche disponibili a una cultura si autoalimenta, generando da pochi building block iniziali molti building block, e da elementi semplici elementi più complicati. […]

Al cuore di questo processo vi è la combinazione di parti e funzionalità adatte a formare una soluzione, concettuale o fisica. Ma questa non è l'unica forza che guida l'evoluzione della tecnologia. L'altra è la necessità, la richiesta di nuovi modi di fare le cose. E le necessità, a loro volta, nascono più dalla tecnologia che direttamente dai desideri umani; esse derivano principalmente dalle limitazioni delle tecnologie stesse e dai problemi da esse generati. Questi devono essere risolti mediante ulteriori nuove tecnologie, cosicché, nella tecnologia, la necessità deriva dalla soluzione tanto quanto la soluzione deriva dalla necessità. L'evoluzione combinatoria consiste nel costituirsi dei bisogni tanto quanto delle soluzioni agli stessi. Il processo complessivo con cui tutto questo avviene non è né uniforme né piano. In ogni momento, il collettivo della tecnologia evolve aggiungendo o eliminando tecnologie, creando nicchie di opportunità per ulteriori tecnologie, e scoprendo nuovi fenomeni. Interi corpi di tecnologie evolvono, nel senso stretto di un continuo sviluppo: essi emergono, cambiano costantemente il loro "vocabolario", e sono assorbiti dalle industrie dell'economia. E anche le singole tecnologie evolvono – sviluppandosi. Per fornire migliori prestazioni, esse cambiano continuamente le loro parti interne, sostituendole con assemblaggi più complessi. Il risultato è un continuo modificarsi a tutti i livelli. A tutti i livelli appaiono nuove combinazioni, vengono aggiunte tecnologie nuove, e le vecchie scompaiono. In questo modo la tecnologia esplora costantemente l'ignoto, crea costantemente nuove soluzioni e nuovi bisogni e, con questi, una perpetua novità. Il processo è organico: strati nuovi si formano sopra quelli vecchi, e creazioni e rimpiazzi si sovrappongono nel tempo. Nel suo significato collettivo, la tecnologia non è semplicemente un catalogo di parti individuali. È una chimica metabolica, un collettivo quasi illimitato di entità che interagiscono e costruiscono da quello che c'è, per produrre nuove entità – e nuovi bisogni.[8]

Iperfunzionalismo e altri problemi

Il complesso sistema di cicli di feedback, schematizzato in Figura 11, ha generato tradizionalmente una forte tendenza all'*iperfunzionalismo* dei prodotti tecnologici: i prodotti sul mercato tendono a fornire prestazioni in eccesso rispetto alle esigenze degli utenti. Questa tendenza è particolarmente visibile nei prodotti software (e in quei prodotti che contengono una componente importante di software), che sono i manufatti evolutivi per eccellenza: modificare il software non richiede modifiche a impianti di produzione, e le nuove versioni possono essere distribuite, attraverso la rete, a costi sostanzialmente nulli.

Donald Norman, nel suo libro *Il computer invisibile*, ha presentato un modello dell'evoluzione tipica dei prodotti ad alta tecnologia, in cui si mettono a confronto, da un lato, le prestazioni del prodotto durante la sua evoluzione e, dall'altro, le necessità degli utenti che il prodotto è in grado di soddisfare (Figura 14).

[8] W.Brian Arthur, *The Nature of Technology – What it is and how it evolves*, Free Press, 2009, pagg.204-205 (nostra traduzione dall'inglese).

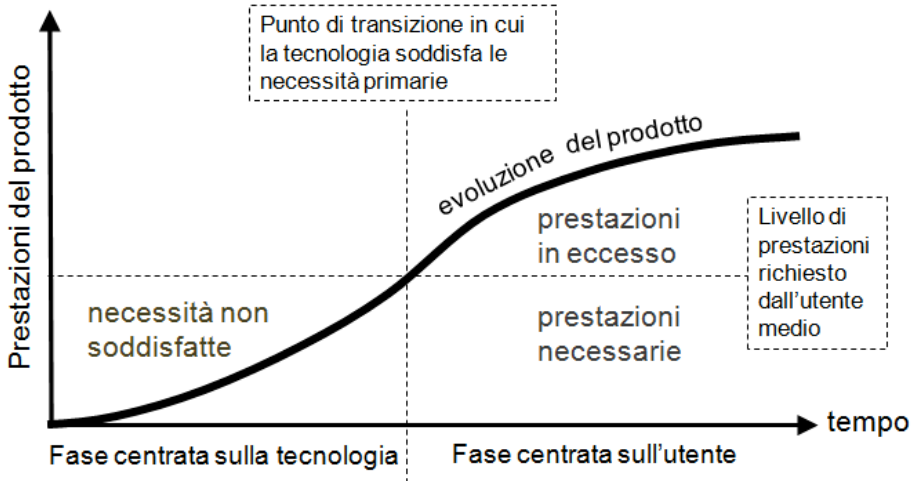


Figura 14. Evoluzione dei prodotti high-tech secondo D.Norman

Secondo questo modello, nella prima fase di vita di ogni prodotto le sue prestazioni sono inadeguate rispetto ai bisogni degli utenti. In questa fase il prodotto è ancora immaturo: si tratta, per così dire, di una tecnologia che cerca, senza ancora riuscirci appieno, di soddisfare le esigenze dei suoi utenti (fase centrata sulla tecnologia). In seguito all'evoluzione del prodotto, esso – ammesso ovviamente che sopravviva nel mercato – raggiunge il "punto di pareggio", nel quale le prestazioni eguagliano i bisogni del suo utente tipico, quello a cui si rivolge prioritariamente. In seguito a ulteriori evoluzioni, il prodotto entrerà in una fase in cui le prestazioni eccedono i bisogni di questo utente, perché cercherà via via di soddisfare le esigenze di *tutti* i possibili utenti (fase centrata sull'utente*)*. Il livello di iperfunzionalismo conseguente può essere più o meno spinto. Nel caso dei prodotti software la tendenza all'iperfunzionalismo è esasperata. Oggi esistono prodotti in cui l'utente tipico usa solo una piccola – o piccolissima – parte delle funzioni disponibili. Appartengono a questa categoria le tradizionali suite software di office (dai word processor ai fogli elettronici) e di elaborazione grafica, presenti sul mercato fin dall'inizio dell'era dei personal computing e ciononostante ancora in continua trasformazione, dopo essersi evolute attraverso numerose versioni.

Il modello di Figura 14, che Norman ha applicato ai singoli prodotti, può essere applicato anche a interi ecosistemi tecnologici. Pensiamo, per esempio, al gran numero di prodotti "ancillari" che vengono messi sul mercato a complemento di specifici prodotti di largo successo. Pensiamo, solo per fare due esempi noti, alle 100.000 (centomila!) applicazioni software disponibili agli utenti di iPhone dopo soli tre anni dal suo lancio sul mercato, e all'indotto di applicazioni software installabili sulle pagine di Facebook.

Questa ricchezza funzionale può indubbiamente essere considerata un vantaggio per l'utente, che può trovare sul mercato soluzioni adeguate per le esigenze più varie, anche quelle più sofisticate. A fronte di questi evidenti vantaggi, tuttavia, gli svantaggi sono numerosi.

Innanzitutto, la continua e accelerata evoluzione delle soluzioni nel tempo crea numerosi e significativi problemi. L'utente, selezionando un certo prodotto, sa che il suo ciclo di vita sarà breve – e sempre più breve col passare del tempo. L'evoluzione della tecnologia lo renderà presto obsoleto, e probabilmente non più compatibile con gli altri prodotti del suo ecosistema. Sarà quindi forzato ad acquistare nuove versioni del prodotto, anche al di là delle sue reali necessità, a causa della non compatibilità delle versioni vecchie con il nuovo contesto operativo. Anche senza considerare lo svantaggio di carattere economico, che può essere rilevante, ciò lo costringe a imparare di nuovo, almeno in parte, un sistema già noto. D'altra parte, l'accumulo continuo di nuove funzionalità produce versioni sempre più sofisticate e complesse, progettate per rispondere alle esigenze di tutti i possibili utenti o, più semplicemente, per superare le prestazioni della concorrenza. Questa complessità funzionale mette a dura prova chi ha esigenze "normali", che deve districarsi fra mille funzioni a lui non necessarie.

A questi svantaggi si aggiunge il rischio di instabilità del prodotto: la crescita della complessità strutturale aumenta inevitabilmente la probabilità di errori nel software. D'altro canto, la frequenza dei rilasci, che si susseguono a distanza ravvicinata, rende difficile al produttore stabilizzare il software, che può contenere malfunzionamenti anche gravi.

L'utente è costretto, per risolvere i suoi problemi, a ricorrere all'assistenza di tecnici specializzati i quali, d'altra parte, non sono sempre adeguatamente aggiornati: anche per loro non è facile mantenere una competenza adeguata su sistemi in costante evoluzione. Come conseguenza inevitabile, assistiamo alla proliferazione di comunità di utenti che, comunicando in rete attraverso forum, blog o social network, condividono le loro esperienze per aiutarsi reciprocamente a fronteggiare i problemi posti dall'uso dei prodotti della tecnologia.

Complessità d'uso e divario digitale

Nonostante una generalizzata tendenza all'eccesso, che ci fa spesso percepire la frequenza del ricambio tecnologico come un'imposizione forzata e non necessaria, noi non siamo in grado di isolarci dalla tecnologia. Le possibilità di comunicare con i nostri simili, di operare negli ambienti di lavoro, di acquistare le merci e i servizi di cui abbiamo bisogno, di informarci e di studiare, di ricrearci, di viaggiare e di curarci ci impongono in modo sempre più massiccio l'uso di sistemi tecnologici di vario tipo. La società odierna si basa sulla tecnologia, ed è fondamentale che essa sia egualmente accessibile a tutti coloro che ne possano beneficiare, pena la discriminazione fra chi è in grado di usufruirne e chi non lo è, e l'isolamento di questi ultimi dal contesto sociale ed economico.

In un contesto in cui gli strumenti di uso quotidiano sono funzionalmente abbastanza semplici e soggetti a un'evoluzione lenta, come avveniva nell'era pre-digitale, il problema della complessità d'uso può essere considerato relativamente marginale. La semplicità funzionale e, soprattutto, la stabilità nel tempo permettevano all'utente di sfruttare a lungo le conoscenze acquisite per il loro utilizzo. L'utilizzo di un telefono fisso degli anni '60, come quello al centro della Figura 10, si imparava in poco tempo, e questa conoscenza era utilizzabile molto a lungo. Oggi, come si è visto, la situazione è profondamente mutata. In questo nuovo contesto, il problema della complessità d'uso diventa potenzialmente drammatico.

Il cosiddetto *divario digitale (digital divide),* che separa chi può accedere alle tecnologie utili e ai conseguenti vantaggi da chi non può farlo, ha molte cause e molte facce. La discriminante non è solo di natura economica. Vengono "tagliati fuori" anche tutti coloro che, per motivi di età, di cultura, di formazione, di lingua, di geografia non hanno accesso ai sistemi indispensabili per la vita di oggi. Secondo uno studio dell'Unione Europea condotto nel 2005 in 14 Paesi membri, il divario digitale è primariamente legato all'età e al livello di istruzione. In percentuale, i giovani scolarizzati che utilizzano i computer o Internet sono molti di più degli anziani non scolarizzati: la percentuale fra i giovani dai 16 ai 24 anni è tre volte maggiore di quella degli anziani dai 55 ai 74 anni. Una differenza analoga si osserva quando si confrontano persone con alto e basso livello di istruzione. Inoltre, il divario digitale è maggiore nelle aree rurali scarsamente popolate.

Gli anziani o, più in generale, tutti coloro che non sono "nativi digitali", hanno notevoli difficoltà ad avvicinarsi agli strumenti dell'informatica, che i più giovani utilizzano con naturalezza anche senza uno specifico addestramento. Questo *gap* generazionale non è destinato a risolversi spontaneamente con la scomparsa delle generazioni più anziane, come si potrebbe ottimisticamente – anche se alquanto cinicamente - pensare. Come abbiamo visto nelle pagine precedenti, il tasso di cambiamento è tale che, con ogni probabilità, la situazione attuale si riprodurrà – in modo sempre più drammatico – nelle generazioni future. I nativi digitali di oggi saranno gli anziani di domani, alle prese con tecnologie sempre più lontane dalla loro esperienza e formazione.

Chi scrive ha tentato invano, per anni, di convincere la madre ottantenne a utilizzare sistematicamente un telefono cellulare per restare in contatto con la famiglia. Anche se il dispositivo era fra i più semplici presenti sul mercato, l'impresa si è rivelata pressoché impossibile, per la presenza di numerose difficoltà. Queste difficoltà sono innanzitutto di tipo culturale. La "distanza" fra il dispositivo e gli apparecchi telefonici tradizionali, ai quali l'anziano è abituato fin da bambino, non è solo relativa alla forma. Tutta l'interazione è profondamente diversa, e questa diversità si rivela a volte molto più difficile da superare di quanto si potrebbe aspettare chi ha già "metabolizzato" l'uso dello strumento.

Prima difficoltà. Sul telefono tradizionale si compone il numero e *poi si alza la cornetta*. Sul telefono cellulare questa non esiste, è esso stesso, se così si può dire, cornetta, e l'avvio della conversazione si fa premendo il tasto verde, *prima o dopo* aver composto il numero. Seconda difficoltà. I tasti di un cellulare non sono protetti, ed è facile che siano premuti inavvertitamente durante il trasporto, generando chiamate spurie all'ultimo numero in memoria. Occorre allora bloccarli, con una manovra che richiede la pressione sincronizzata di due tasti lontani, con una manovra che l'anziano può trovare difficile. Un apparecchio che a volte produce, non richiesto, delle telefonate "fantasma" verrà allora considerato inaffidabile. Terza difficoltà. Gli apparecchi più semplici sono spesso i più piccoli, e tendono a sfuggire di mano. La tastiera è piccola, ed è facile premere i tasti sbagliati. L'anziano ha spesso difficoltà di udito, ma l'altoparlante non è visibile, a differenza delle vecchie cornette, e può essere posizionato male vicino all'orecchio. D'altro canto, il tasto di regolazione del volume può essere poco evidente e operabile con difficoltà. Quarta difficoltà. La pratica di operare attraverso menu, che nei nativi digitali è acquisita fin dai primi anni di vita, è del tutto estranea all'anziano, che non sa districarsi fra le numerose opzioni previste anche nei cellulari più semplici. Un banale errore nella navigazione, o la pressione involontaria di un tasto, può portare il telefono in uno stato indesiderato, non facilmente riconoscibile dalle informazioni presenti su uno schermo troppo piccolo o troppo poco luminoso per chi non ha una buona vista. A volte, per riconoscere lo stato dell'apparecchio bisogna effettuare una navigazione specifica, facile solo per chi possiede un modello mentale del suo funzionamento. Tutto questo – e sarebbe facile continuare – contribuisce a farlo percepire come un oggetto estraneo, se non addirittura ostile.

D'altro canto, non è possibile segregare gli anziani dalla tecnologia. Per esempio, gli strumenti che permettono di comunicare e di informarsi, come il telefono cellulare, la televisione e Internet, sono oggi indispensabili per mantenersi in contatto con la comunità. Si è fatto l'esempio del cellulare, ma sarebbe facile descrivere le difficoltà nell'accesso a Internet, con i browser funzionalmente sempre più complessi. Anche la televisione, prodotto culturalmente già da molti anni metabolizzato dalla grande maggioranza della popolazione, nel processo di digitalizzazione si modifica drasticamente. L'anziano ha oggi a che fare con uno o più decoder, telecomandi con numerosi tasti e, ancora una volta, menu di navigazione con opzioni sempre più numerose. Come è noto, in Italia il passaggio al digitale terrestre ha dovuto superare notevoli difficoltà.

Il problema dell'accesso degli anziani alla tecnologia è di vasta portata, considerato la continua tendenza all'invecchiamento della popolazione dei paesi sviluppati, conseguenza delle migliorate condizioni di vita e dell'assistenza sanitaria. Nell'Unione Europea, le persone di almeno 65 anni di età, che erano il 17% della popolazione nel 2008, diventeranno il 30% nel 2060. Quelle di almeno 80 anni passeranno dal 4,4% al 12% nello stesso periodo. Il processo di invecchiamento è ancora più rilevante in Italia, dove il tasso di natalità è basso. Secondo l'ISTAT, a fine 2008 un italiano su cinque aveva almeno 65 anni, con una crescita di 5 punti percentuali rispetto a 10 anni prima. Gli ultraottantenni rappresentavano invece il 5,6% della popolazione italiana, con una crescita di 2,5 punti percentuali nello stesso periodo. Sempre in Italia, l'aspettativa di vita superava, nel 2008, gli 83 anni per le donne e i 77 anni per gli uomini, con un incremento superiore ai 10 anni, per entrambi, rispetto al 1960.

Anche per le persone di età intermedia (oggi due terzi degli italiani hanno dai 15 ai 64 anni), la complessità d'uso degli strumenti può costituire una barriera importante al loro utilizzo. Chi non ha preso familiarità con la tecnologia fin dall'età scolastica, e non deve utilizzarla nel lavoro d'ufficio, è facile che ne rimanga lontano. Significativamente, la presenza di bambini in famiglia stimola fortemente l'accesso alle tecnologie dell'informazione: secondo lo studio del 2005 più sopra citato, la percentuale di famiglie che possiedono un personal computer è del 50% più alta se ci sono bambini. Lo stesso vale per le connessioni internet e la banda larga.

Oltre a coloro che hanno difficoltà nell'accesso alle tecnologie per motivi economici, di età o di istruzione, occorre considerare coloro che soffrono di particolari disabilità che possono in qualche modo impedirne l'utilizzo: sordità, ipovisione, daltonismo, cecità, disabilità motorie, disabilità cognitive e così via. Oppure coloro che dispongono di tecnologie obsolete o, in un mondo in cui l'utilizzo della rete è sempre più pervasivo, di connessioni internet particolarmente lente.

Il ruolo dell'interfaccia utente

La riduzione del divario digitale può essere affrontata da più punti di vista. Innanzitutto, garantendo con ogni mezzo a tutti i cittadini la possibilità di accesso alla tecnologia, in particolare a Internet e alla banda larga, e promuovendo una

diffusa alfabetizzazione informatica. A questo, per esempio, sono dedicati i programmi di *e-inclusione* promossi dall'Unione Europea:

> *L'E-Inclusione ("e" per elettronica) punta ad assicurare che le persone svantaggiate non siano escluse per mancanza di alfabetizzazione digitale o accesso internet. E-inclusione significa anche trarre vantaggio dalle nuove opportunità offerte dai servizi tecnici e digitali per l'inclusione delle persone socialmente svantaggiate e delle aree meno favorite. La Società dell'Informazione ha la possibilità di distribuire la conoscenza più equamente e di offrire nuove opportunità di lavoro, superando le barriere tradizionali alla mobilità e alla distanza geografica.*[9]

Queste iniziative considerano, sostanzialmente, la complessità d'uso della tecnologia una variabile indipendente. In altre parole, se la tecnologia pone delle difficoltà, si opererà in primo luogo sui suoi utenti, istruendoli e avvicinandoli a essa in ogni modo possibile. Un approccio diverso, e complementare, è quello, per così dire, di modificare la tecnologia dall'interno, promuovendo fra chi la progetta e la produce una *cultura della semplicità,* che consideri la facilità d'uso non come una semplice caratteristica fra le altre (il peso, il prezzo, il colore, …) ma come un prerequisito indispensabile.

Secondo questa filosofia, la facilità d'uso non deve essere considerata solo dal punto di vista economico, come un mezzo per vendere di più i prodotti in un mercato di massa o per aumentare la produttività di chi li usa. Il suo scopo principale è di permettere l'accesso a strumenti che semplificano – o rendono possibili - i compiti quotidiani, facendoci risparmiare tempo e migliorando la qualità della nostra vita. Da questo punto di vista non può e non deve essere considerata un optional, ed è precisa responsabilità dei progettisti acquisire le conoscenze, i metodi e gli strumenti per la progettazione di sistemi che siano utilizzabili senza problemi da tutti i loro potenziali utenti. Progettare *per tutti* significa allora tenere conto di queste diversità e preservarle, facendo sì che ciascuno possa accedere in modo naturale agli strumenti che gli servono, senza difficoltà o forzature. La disciplina della progettazione, tradizionalmente centrata sulla risoluzione dei problemi tecnologici e della produzione industriale dei prodotti, deve trasformarsi, e considerare, come punto di partenza e arrivo, le necessità dell'utente.

In questo contesto, l'interfaccia d'uso dei sistemi riveste un ruolo fondamentale. Essa ha il compito di "filtrare" la complessità, presentando all'utente un'immagine semplificata del prodotto, e congruente con i compiti che egli deve svolgere (Figura 15). Una buona interfaccia non solo nasconde la complessità interna del sistema, ma ne riduce la complessità funzionale, mettendo a disposizione dell'utente funzioni di più alto livello, in grado di effettuare compiti complessi con un grado di automatismo maggiore. Ciò viene realizzato integrando numerose funzionalità semplici in funzionalità più potenti, con il risultato di semplificare il dialogo fra l'utente e il sistema. Per esempio, i primi word processor non avevano funzioni di controllo ortografico, che erano invece realizzate da programmi separati. L'utente doveva quindi imparare a usare questi programmi, e attivarli dopo avere composto il testo. In seguito, queste funzioni furono integrate nei word processor, che fornirono nuovi comandi dedicati allo scopo. I word processor odierni realizzano un ulteriore livello di semplificazione: il controllo ortografico può essere effettuato automaticamente dal sistema, durante la digitazione del testo, senza che l'utente lo debba richiedere esplicitamente.

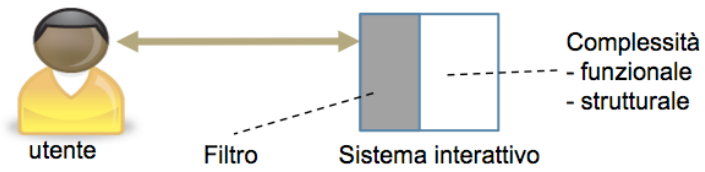


Figura 15. L'interfaccia utente come filtro semplificatore

[9] Da http://ec.europa.eu/information_society/activities/einclusion/index_en.htm (nostra traduzione). Il documento i2010, *Strategy for an innovative and inclusive European Information Society*, approvato dalla Commissione Europea 2005 (http://ec.europa.eu/12010), forniva un insieme di linee guida generali, di tipo politico e strategico, per la Società dell'Informazione nell'Unione Europea, fino all'anno 2010.

È interessante osservare che questo ruolo di semplificazione svolto dall'interfaccia, a fronte della sempre crescente complessità funzionale dei sistemi, produce come conseguenza una significativa crescita quantitativa del software che la gestisce. Già nel 1992, uno studio relativo a una settantina di sistemi software di ogni tipo e dimensione mostrava che, in media, il 48% del codice era dedicato alla gestione dell'interfaccia con l'utente. I costi di progettazione, di sviluppo e di manutenzione di tale codice incidevano, rispettivamente, per il 45%, il 50% e il 37% sui costi complessivi di ciascuna di queste tre attività. [10]

Oggi, molto più di ieri, il progettista ha di fronte a sé una sfida non eludibile: conciliare complessità (strutturale e funzionale) e semplicità d'uso – *per tutti*. La sfida è difficile, poiché richiede un modo di progettare del tutto nuovo rispetto al passato. Come vedremo nei prossimi capitoli, essa può essere vinta soltanto a patto di modificare completamente l'approccio tradizionale alla progettazione dei sistemi. Un vero e proprio *nuovo paradigma nell'ingegneria,* che richiede di cambiare le metodologie di progettazione e sviluppo, la composizione dei gruppi di progetto e la formazione stessa dei progettisti. Da una progettazione sistema-centrica è necessario passare a una progettazione centrata sull'essere umano, che consideri le esigenze dell'utente prima di ogni altra cosa. Questa trasformazione può apparire banale, ma è di vasta portata. Anche se di progettazione centrata sull'essere umano si parla da una ventina d'anni, le pratiche tradizionali nelle attività di progetto, molto lontane da questa filosofia, sono ancora ben radicate. Non si comprende che la facilità d'uso è un attributo che si deve conquistare faticosamente, durante tutta la durata di un progetto, attuando interventi e pratiche specifiche, e non un facile slogan da scrivere sulle brochure commerciali.

La Human Computer Interaction

Questo nuovo ruolo dell'interfaccia, che da strumento di controllo diviene strumento di semplificazione, assume importanza rilevante con la diffusione dei prodotti tecnologici sui mercati di massa, e in particolare dei personal computer, a partire dai primi anni '80 del secolo scorso. Nel 1981 l'IBM lancia sul mercato il suo personal computer, avviando una rivoluzione che in pochi anni modificherà radicalmente il panorama dell'informatica. I nuovi prodotti software dell'informatica personale, che sono sviluppati in quegli anni, non si rivolgono più a un mercato di specialisti, come le applicazioni degli anni precedenti, ma a una massa di utilizzatori senza competenze di informatica. Questi pretendono strumenti di facile uso, e ciò promuove le ricerche sui metodi legati alla realizzazione di "buone" interfacce utente. Proprio in quegli anni nasce allora una disciplina nuova, subito denominata *Human-Computer Interaction* (abbreviata con la sigla HCI).

Il SIGCHI, lo *Special Interest Group on Computer-Human Interaction* dell'ACM, l'associazione dei professionisti americani dell'informatica, nasce nel 1982. Le principali conferenze del settore nascono anch'esse in quegli anni: nel 1983 la *Computer-Human Interaction Conference* dell'ACM (CHI), organizzata annualmente dallo stesso SIGCHI; nel 1984 l'*Interact Conference* dell'IFIP, nel 1985 la *HCI Conference* della British Computer Society, e così via. Alcune università iniziano a offrire corsi di HCI all'interno dei curriculum di informatica.

Un anno simbolicamente importante nella storia dell'HCI è il 1992, quando il SIGCHI pubblica un'articolata proposta per un curriculum di studi universitari sulla Human-Computer Interaction, che viene così definita:

> *HCI è una disciplina che si occupa della progettazione, valutazione e realizzazione di sistemi interattivi basati su computer destinati all'uso umano e dello studio dei principali fenomeni che li circondano.*[11]

Come si vede, si tratta di una definizione molto ampia, che colloca l'HCI all'intersezione di più discipline molto diverse: l'ingegneria, le scienze dell'uomo, le scienze dei computer. Infatti, come recita ancora il documento del SIGCHI:

> *L'HCI nel suo complesso è un'area interdisciplinare. Sta emergendo come una specializzazione all'interno di parecchie discipline, con enfasi differenti: la scienza dei computer (la progettazione delle applicazioni e l'ingegnerizzazione delle interfacce umane), la psicologia (l'applicazione delle teorie dei processi cognitivi e*

[10] B.A.Myers, M.B.Rosson, *Survey on User Interface Programming*, in ACM CHI 92 Conference Proceedings, pp.195-202.
[11] ACM SIGCHI, *Curricula for Human Computer Interaction,* 1992, vedi http://www.acm.org/sigchi/cdg/.

l'analisi empirica dei comportamenti degli utenti), la sociologia e l'antropologia (le interazioni fra la tecnologia, il lavoro e l'organizzazione), e l'industrial design (i prodotti interattivi).

Questa disciplina, seppur nuova, non nasceva dal nulla. Da un lato, infatti, faceva riferimento all'ergonomia, la scienza sviluppata, soprattutto a partire dagli anni successivi alla seconda guerra mondiale, per studiare il *fit* fra le persone e il loro ambiente di lavoro. Dall'altro, s'ispirava alle idee di alcuni pionieri dell'informatica, che già dagli anni 60 avevano iniziato a studiare nuove e più strette modalità di interazione fra uomo e calcolatore.

Inizialmente, l'ergonomia (dalle parole greche ergon, lavoro e nomos, legge) studiava soprattutto le compatibilità fra le caratteristiche fisiche dell'uomo e della macchina, studiando la disposizione ottimale dell'ambiente e delle apparecchiature di lavoro in funzione dei compiti da svolgere (Figura 16), e mettendo spesso in evidenza l'esistenza di situazioni problematiche, che potevano portare a varie forme di disabilità. A questo scopo, gli ergonomi utilizzavano i risultati di varie discipline, quali l'antropometria, la biomeccanica, l'ingegneria, la fisiologia e la psicologia. Successivamente, con l'aumento della complessità dei compiti, l'ergonomia ha spostato sempre più la propria attenzione sullo studio dei processi cognitivi e di elaborazione delle informazioni sottostanti ai processi del lavoro umano e, in particolare, all'interazione fra essere umani e tecnologia (*ergonomia cognitiva*).

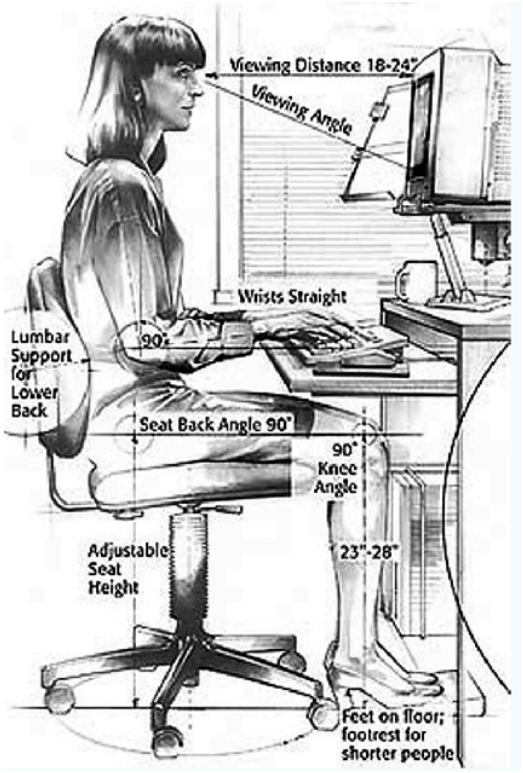


Figura 16. Postura ergonomica al posto di lavoro (da Wikipedia)

Secondo la definizione dell'AIE (Associazione Italiana di Ergonomia),

L'Ergonomia (o scienza del Fattore Umano) ha come oggetto l'attività umana in relazione alle condizioni ambientali, strumentali e organizzative in cui si svolge. Il fine è l'adattamento di tali condizioni alle esigenze dell'uomo, in rapporto alle sue caratteristiche e alle sue attività. Nata per studiare e far rispettare nella progettazione una serie di norme che tutelano la vita del lavoratore e accrescono l'efficienza e l'affidabilità dei sistemi uomo-macchina, l'ergonomia ha allargato il proprio campo di applicazione in funzione dei cambiamenti che sono sopravvenuti nella domanda di salute e di benessere. L'obiettivo attuale è quello di contribuire alla progettazione di oggetti, servizi, ambienti di vita e di lavoro, perché rispettino i limiti

dell'uomo e ne potenzino le capacità operative. L'ergonomia si alimenta delle acquisizioni scientifiche e tecnologiche che permettono di migliorare la qualità delle condizioni di vita, in tutte le attività del quotidiano.[12]

Gli uomini della HCI venivano tuttavia, in buona parte, dalla scienza dei computer. Molti di loro sperimentavano da tempo nuove modalità di interazione con la macchina, nella convinzione che i calcolatori non servissero solo a elaborare e gestire grandi quantità di dati aziendali o scientifici, ma potessero diventare delle vere e proprie *protesi cognitive,* utilizzabili dalle persone per eseguire compiti complessi. Nel 1962, in un famoso rapporto relativo a una ricerca per "aumentare l'intelletto umano" attraverso gli strumenti dell'informatica[13], Douglas Engelbart scriveva:

Aumentare l'intelletto umano significa per noi incrementare le capacità di una persona di affrontare una situazione problematica complessa, di raggiungere la comprensione necessaria a scopi particolari e di trovare soluzioni ai problemi. Per incremento delle capacità, in questo contesto, intendiamo una miscela delle cose seguenti: comprensione più rapida, comprensione migliore, possibilità di ottenere un livello di comprensione utile in una situazione precedentemente troppo complessa, soluzioni più rapide, soluzioni migliori, e la possibilità di trovare soluzioni a problemi che prima sembravano insolubili. E tra le situazioni complesse includiamo i problemi professionali dei diplomatici, dei manager, degli scienziati sociali, degli scienziati della vita, dei fisici, degli avvocati, dei progettisti – che la situazione problematica esista per venti minuti o per venti anni. Non stiamo parlando di trucchi intelligenti e isolati che sono di aiuto in situazioni particolari. Ci riferiamo a un modo di vivere in un dominio integrato dove le intuizioni, i tentativi, le cose intangibili e il "senso della situazione" dell'uomo coesistano utilmente con concetti potenti, con terminologie e notazioni efficienti, con metodi sofisticati e con ausili elettronici di grande potenza.

Le ricerche pionieristiche di Engelbart presso lo Stanford Research Institute produssero molte invenzioni fondamentali, fra cui mouse e puntatori, display editor, outline processing, finestre, ipertesti, video conferenze, help contestuale. La sua visione influenzò profondamente tutti gli sviluppi successivi del personal computing. In particolare, i ricercatori del *Palo Alto Research Center* (PARC) della Xerox, che svilupparono le sue idee e produssero già negli anni '70 i primi prototipi di workstation personali. Con il personal computer, lanciato sul mercato di massa fra la fine degli anni '70 e l'inizio degli anni 80, nascono numerosi strumenti di nuova concezione: il word processor, lo spreadsheet e gli altri strumenti di elaborazione personale che, a partire da quegli anni, generano la nuova industria dei pacchetti software per il mercato di massa.

Dai progetti pionieristici dei primi anni, molte cose sono successe. Il personal computer, da strumento "da scrivania" si è evoluto in strumento portatile, e la successiva evoluzione delle reti ha prodotto una nuova enorme crescita delle possibilità – e della complessità – degli strumenti. Abbiamo costruito strumenti che ci permettono di elaborare idee e informazioni enormemente complesse, e che ci permettono di gestirle e di comunicarle istantaneamente e massivamente a interlocutori sparpagliati negli angoli più remoti del pianeta. La diffusione della telefonia mobile (dalla fine degli anni '80) e della rete Internet (dall'inizio degli anni 90) hanno dato un'accelerazione formidabile a questi processi. Questa situazione è davanti agli occhi di tutti, e non è necessario fornire esempi. Si pensi soltanto alla complessità dei mercati finanziari di oggi, che non potrebbero esistere senza l'informatica evoluta degli ultimi anni. Reti di operatori che, dalle varie piazze del mercato globale, effettuano compravendite non di prodotti tangibili, ma di oggetti astratti costituiti da certificati azionari, debiti e crediti, impegni, scommesse.

Da allora la disciplina della Human-Computer Interaction si è sviluppata in modo considerevole, in molte direzioni, come si vede dalla Figura 17, che riporta i 50 termini più frequentemente citati negli atti delle conferenze CHI tenute nei primi 23 anni (dal 1983 al 2006).[14]

---

[12] Cfr. http://www.societadiergonomia.it.

[13] D.Engelbart, *Augmenting Human Intellect: A Conceptual Framework*, Stanford Research Institute, Ottobre 1962 (reperibile in rete).

[14] Da N.Henry, H.Goodell, N.Elmqvist, J.-D. Fekete, *20 Years of Four HCI Conferences: A Visual Exploration*, International Journal of Human Computer Interaction, 23(3), pagg.239-285, disponibile anche in rete.



Figura 17. *Frequency cloud* dei termini usati negli atti delle CHI Conference tenute dal 1983 al 2006

Oggi, gli argomenti principali considerati dalla disciplina della HCI, secondo Wikipedia, sono i seguenti:

- metodologie e processi per la progettazione delle interfacce;
- tecniche per l'implementazione delle interfacce (per esempio, algoritmi, strumenti e librerie software);
- tecniche per valutare e confrontare le interfacce;
- sviluppo di nuove interfacce e di nuove tecniche di interazione;
- sviluppo di modelli descrittivi e previsionali, e di teorie dell'interazione.

Questi obiettivi non sono sostanzialmente diversi da quelli enunciati nel curriculum del 1992 dell'ACM. È però profondamente cambiato l'oggetto di studio: dallo studio dell'interazione con i computer in senso stretto, essa si rivolge oggi a quello dell'interazione con oggetti interattivi di ogni tipo. Il nome Human-Computer Interaction è rimasto, ma, con l'evoluzione delle applicazioni, il computer in molti casi si è reso "invisibile", inserito come componente interno di oggetti intelligenti di varia foggia, specializzati nello svolgimento dei compiti più disparati. Il computer c'è sempre – anzi è sempre più pervasivo – ma, come nel titolo di un famoso libro di Donald Norman dedicato all'argomento, è sempre più "invisibile".

Ripasso ed esercizi

1. Spiega che cosa s'intende per interfaccia d'uso.
2. Costruisci due diagrammi come quelli di Figura 5 e Figura 8 utilizzando quattro prodotti diversi da quelli utilizzati nel libro.
3. Identifica e descrivi sinteticamente i cambiamenti più importanti avvenuti negli ultimi 5 anni nelle tue attività quotidiane, a seguito dell'adozione di nuovi prodotti di natura tecnologica.
4. Riassumi le cause principali dell'evoluzione accelerata dei prodotti software per personal computer, e sintetizza vantaggi e svantaggi di questa accelerazione.
5. Che cosa s'intende con accessibilità?
6. Analizza e descrivi le cause e il livello del divario digitale, se esiste, fra te e gli altri membri della tua famiglia.
7. Di che cosa si occupa la disciplina della Human Computer Interaction? Elencane le principali aree d'interesse.
8. Quali sono le principali differenze fra ergonomia e HCI?

Approfondimenti e ricerche

1. Approfondisci il concetto di digital divide, prendendo le mosse dalle voci (in italiano e in inglese) di Wikipedia.
2. Esaminando il portale tematico della Commissione Europea sulla e-inclusion (http://ec.europa.eu/information_society/activities/einclusion/index_en.htm), individua le politiche avviate su questo tema.
3. Il seguente articolo contiene una sintesi interessante della storia della HCI. S.Bagnara, S.Pozzi, *Fondamenti, Storia e Tendenze dell'HCI*, in A.Soro (ed.), *Human Computer Interaction – Fondamenti e prospettive*, pagg. 17-42, Ed.Polimetrica, 2009 (disponibile anche in rete).
4. Uno dei principali convegni scientifici nel campo della HCI è il convegno CHI, organizzato annualmente dal SIGCHI dell'ACM. Gli atti di questo convegno sono disponibili in rete, in http://www.acm.org (in questo sito, selezionare Proceedings, poi CHI). Esamina gli atti dell'ultimo convegno, per farti un'idea del tipo di temi affrontati.

2.Evoluzione dei paradigmi d'interazione

Sintesi del capitolo

Questo capitolo traccia una breve storia dell'evoluzione dei principali paradigmi per l'interazione uomo-computer che si sono consolidati nell'ultimo mezzo secolo, in stretta relazione con le tecnologie per l'interazione: la teletype, il terminale video, il personal computer, il browser web, il mobile. Con il social computing l'interfaccia utente assume, infine, il compito di mediare l'interazione fra più utenti connessi in rete. A conclusione del capitolo, si accenna all'evoluzione in atto verso la cosiddetta intelligenza ambientale.

Paradigmi e tecnologie di interazione

Con i primi computer l'utente non aveva alcuna interazione diretta. Di solito, lasciava al centro di calcolo il pacco di schede (*batch*) con i lavori (*job*) da svolgere, e passava a ritirare i tabulati con i risultati qualche ora dopo, o il giorno successivo. Il computer veniva gestito da operatori specializzati, ed era sostanzialmente inavvicinabile dall'utilizzatore finale, che era comunque un tecnico. Da quando, con i sistemi time-sharing degli anni '60, questo utente ha avuto la possibilità di interagire direttamente con la macchina attraverso un terminale interattivo, la comunicazione fra uomo e calcolatore si è evoluta consolidando un certo numero di *paradigmi d'interazione* diversi, che vogliamo qui ripercorrere brevemente. Come filo conduttore sceglieremo l'evoluzione della tecnologia: i differenti dispositivi che si sono di volta in volta maggiormente diffusi hanno suggerito modalità di interazione differenti e, a loro volta, ne sono stati influenzati, in un ciclo di retroazione fra lo strumento e il suo utilizzo che ha prodotto una evoluzione continua e radicale delle modalità di interazione uomo-macchina, tuttora in corso (Figura 18).

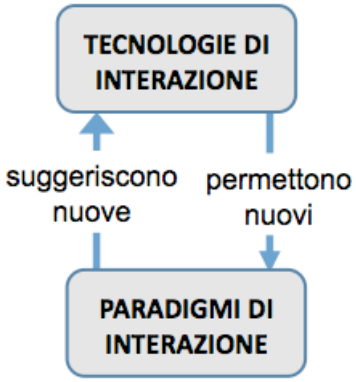


Figura 18. Evoluzione dei paradigmi d'interazione uomo-computer

Dovendo schematizzare, prenderemo come riferimento sei tecnologie fondamentali, che hanno determinato altrettanti paradigmi di interazione fra uomo e computer, la cui evoluzione nel tempo è indicata approssimativamente nella Figura 19:

- il terminale scrivente;
- il terminale video;
- il personal computer;
- il browser web;
- il mobile.

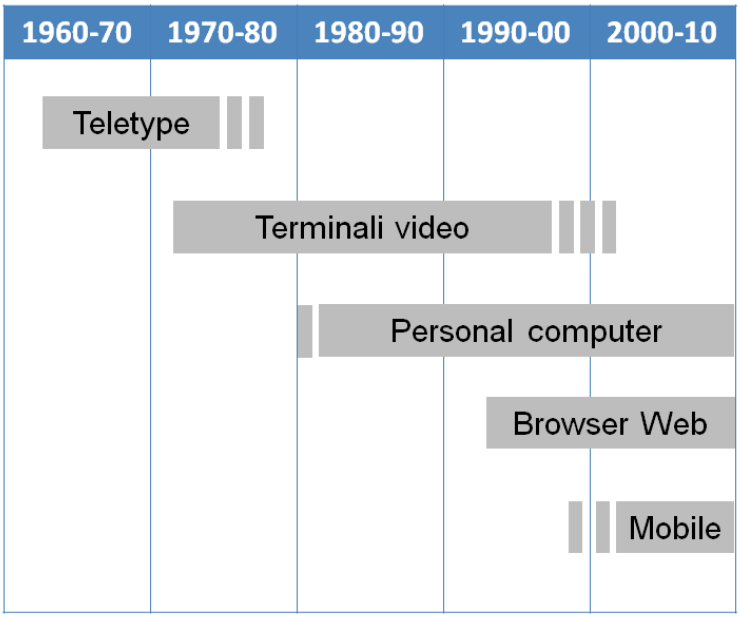


Figura 19. Evoluzione dei paradigmi/tecnologie d'interazione uomo-computer

Come si vede dallo schema di Figura 19, e come vedremo meglio in seguito, l'evoluzione dei paradigmi d'interazione non è sequenziale. In ogni momento convivono paradigmi differenti, in un gioco di sovrapposizione e d'interazioni molto complesso. Questo perché, normalmente, il ciclo di vita di una generazione tecnologica è più lungo del ciclo dell'innovazione della tecnologia. In altre parole, l'innovazione tecnologica produce prodotti di nuova generazione prima che i prodotti della generazione precedente abbiano concluso la loro esistenza nel mercato.

Tipicamente, il software ha un ciclo di vita piuttosto lungo: un sistema di software applicativo può vivere, rimanendo essenzialmente se stesso nonostante i continui interventi di manutenzione, anche per diecine d'anni. Ne consegue che raramente il software in uso presso un'organizzazione viene rinnovato contestualmente al rinnovo delle tecnologie hardware e, in particolare, di quelle relative all'interazione utente-calcolatore. Si osserva quindi in molti casi, per così dire, un ritardo dei paradigmi d'interazione adottati dal software effettivamente in uso, rispetto a quanto sarebbe possibile in relazione all'evoluzione della tecnologia. Così, ad esempio, programmi progettati per essere utilizzati con terminali scriventi sono spesso sopravvissuti a lungo dopo l'adozione di terminali video. Si pensi, ad esempio, ai *line editor,* sopravvissuti a volte per molti anni all'introduzione dei terminali video, nonostante che questa tecnologia permettesse, con gli editor full-screen, un trattamento del testo ben più agevole; si pensi, ancora, ad alcuni grossi sistemi di prenotazione voli basati su un'interazione a comandi ancora in uso, ecc. Si osserva così, spesso, una sorta di *sovrapposizione di paradigmi* diversi in uno stesso sistema di calcolo, che in qualche modo riflette l'evoluzione tecnologica avvenuta durante la vita del sistema stesso.

Il terminale scrivente: scrivi e leggi

Il *terminale scrivente* o *teletype* era essenzialmente un apparato composto da tastiera e stampante integrata, a foglio continuo (Figura 20). A confronto con i terminali sviluppati successivamente, le prestazioni erano molto modeste, sia per quanto riguarda la velocità di stampa (per esempio, 30 caratteri al secondo) che per quanto riguarda la velocità di trasmissione lungo la linea verso il calcolatore (per esempio, qualche diecina di caratteri al secondo). Con la mediazione di tale dispositivo, con il calcolatore *si dialoga per iscritto;* il paradigma d'interazione che si è inizialmente consolidato è pertanto quello della comunicazione scritta:

SCRIVI E LEGGI

Tipicamente, il calcolatore segnala all'utente il suo stato di attesa comandi; l'utente digita allora un comando, cui segue la risposta dell'elaboratore e il sollecito (*prompt*) successivo, che vengono stampati sul rullo di carta. Questa è la modalità comunicativa di molti *command language* (come per esempio quello tradizionale dei sistemi Unix, MS-DOS e Linux), o dei *query language* per l'interrogazione di basi di dati (come ad esempio il linguaggio SQL) o, ancora, di molti *adventure game* della prima generazione, di tipo testuale.

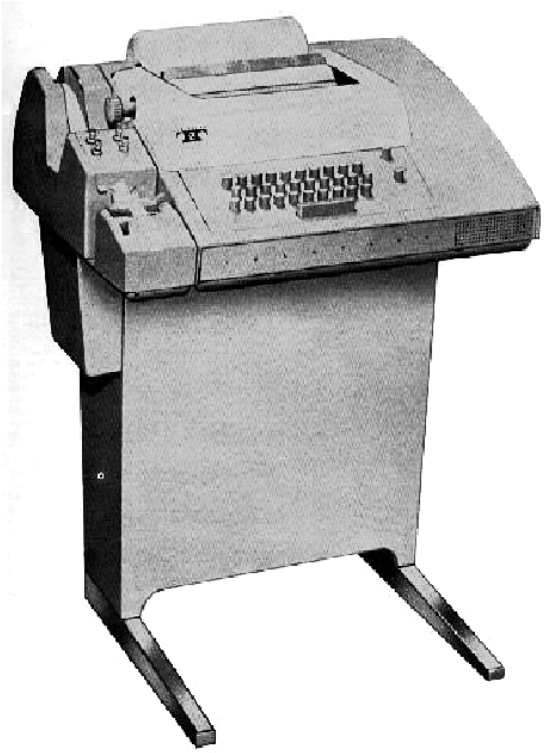


Figura 20. Terminale scrivente (teletype)

In tutti questi esempi, è l'utente che ha il controllo del dialogo: il computer ha un ruolo passivo, limitandosi a riconoscere le richieste e a fornire le risposte. Sono stati tuttavia sviluppati, anche sistemi in cui questi ruoli sono invertiti, e il dialogo è totalmente controllato dalla macchina: è l'utente ad avere un ruolo passivo, e si limita a fornire le risposte richieste. Esempi tipici sono i *sistemi esperti*, gli *advisory system,* e in generale tutti quei sistemi in cui è il calcolatore ad avere competenza del problema e potere di decisione nella conduzione della conversazione.

La Figura 21 mostra un tipico esempio di questo stile di interazione, tratto da Mycin, un sistema esperto progettato nei primi anni ’70, il cui scopo era suggerire gli antibiotici più adatti per curare specifiche infezioni batteriche. Mycin poneva al medico una serie di domande, in un ordine predeterminato in funzione delle risposte precedenti.

```
(1) Patient's name: (first-last)
**FRED SMITH
(2) Sex:
**MALE
(3) Age:
**55
(4) Have you been able to obtain positive cultures
  from a site at which Fred Smith has an infection?
**YES
(5) What is the infection?
**PRIMARY-BACTEREMIA
(6) Please give the date and approximate time when
  signs of symptoms first appeared
**
```

Figura 21. Dialogo controllato dal sistema (Mycin)

Paradigmi basati su una vera e propria *conversazione* fra utente e sistema, in questa fase dell'evoluzione del dialogo uomo-macchina, sono ancora al di fuori delle possibilità della tecnologia, non solo per le difficoltà tecniche insite nell'elaborazione del linguaggio naturale, ma soprattutto per la grande complessità e ricchezza intrinseche nella nozione stessa di conversazione.

Il terminale video: indica e compila

Con l'introduzione, dal 1971, del *terminale video* (Figura 22), il tabulato continuo della teletype viene sostituito dallo *schermo video* (tipicamente di 24 linee di 80 caratteri). La velocità di visualizzazione di una singola schermata è praticamente istantanea, e quella di trasmissione sulla linea verso il calcolatore aumenta in modo considerevole (tipicamente, tra 150 e 1200 caratteri al secondo). La tastiera si arricchisce di svariati *tasti funzione*, che attivano servizi eseguiti dal software residente sul calcolatore remoto o direttamente dal terminale. Altro aspetto innovativo è la presenza, sul video, di un *cursore* spostabile in ogni direzione mediante tasti appositi. Questo permette all'utente di indicare una posizione precisa del video. Il cursore è una sorta di "pennino" che "deposita sul video" il carattere digitato alla tastiera, o un "dito" con cui indicare l'elemento informativo d'interesse: un carattere, un campo di input, una voce di menu, ecc. Permettendo una nuova dimensione "gestuale" (l'atto di indicare sul video qualcosa con il cursore), il terminale video introduce, sia pure in embrione, un paradigma nuovo, che si svilupperà appieno con i sistemi dotati di mouse: quello della *manipolazione diretta*.



Figura 22. Terminale video IBM 3270 (1972)

La possibilità di visualizzare rapidamente sul video l'informazione che si sta elaborando, e di indicare col cursore la specifica porzione d'interesse, suggerisce lo sviluppo di modalità di comunicazione completamente diverse da quelle che si potevano realizzare con una teletype. Per esempio, negli *editor full-screen,* l'utente che crea o corregge un testo non opera più su una pagina invisibile e pertanto soltanto "immaginata", come avveniva in precedenza nei *line editor,* ma su una pagina ben visibile sullo schermo, indicando col cursore il punto desiderato (per esempio, il punto in cui inserire altro testo, o il brano da cancellare).

I sistemi software di questa generazione propongono tipicamente un paradigma di interazione basato su *menu* e *form*, da compilare spostando il cursore sui vari campi di input:

INDICA E COMPILA

A questa categoria di sistemi appartiene la maggior parte delle applicazioni gestionali sviluppate a partire dagli anni '70 per oltre un quarto di secolo: la compilazione, campo per campo, di una form visualizzata sullo schermo diviene il paradigma standard per le applicazioni di tipo transazionale (come i sistemi informativi aziendali).

Per evidenziare meglio la struttura delle form (differenziando i campi variabili rispetto al testo fisso, separando le varie zone della maschera, ecc.), i terminali video si arricchiscono negli anni di funzioni grafiche, anche se rudimentali: caratteri di testo visualizzati in *doppia intensità*, sottolineati o in *reverse,* caratteri grafici con cui comporre cornici o separare zone del video, ecc. Una molteplicità di tasti funzionali specializzati permette un'interazione scorrevole ed efficiente.

Le transazioni da svolgere sono tipicamente selezionate attraverso *menu:* su più linee del video sono proposte le varie alternative possibili, che l'utente può selezionare con semplici operazioni alla tastiera, per esempio digitandone il numero d'ordine. Spesso i menu sono organizzati in modo gerarchico, con la voce di un certo livello che richiama un menu di livello inferiore, e così via, fino ad arrivare alla form della transazione selezionata.

Dal punto di vista della qualità dell'interazione, questo paradigma semplifica molto la natura del dialogo uomo-computer che, se da un lato guadagna semplicità (le scelte possibili sono ben visibili sullo schermo), dall'altro perde ricchezza (le scelte possibili sono *solo* quelle visibili sullo schermo). Con menù e maschere l'utente è completamente guidato, ma il computer assume un ruolo alquanto più passivo che in precedenza: da soggetto di un dialogo (che almeno metaforicamente, e con molta generosità, poteva ricordare il dialogo fra due interlocutori umani - si pensi all'immagine del "cervello elettronico", ricorrente negli anni '50 e '60), diviene ora puro oggetto di manipolazione, più o meno come un telefono, una lavatrice, un televisore.

Il personal computer: non dirlo, fallo

Nel *personal computer,* al video e alla tastiera si aggiungono potenza di calcolo locale, possibilità di archiviazione su dischetti flessibili asportabili o su dischi interni e possibilità di stampa locale. Dopo i primi *home computer,* nati alla fine degli anni settanta e destinati essenzialmente al mercato degli amatori o al nuovo incerto mercato dell'home computing, dal 1981 il personal computer entra nel mondo delle aziende con il PC IBM (Figura 23a). Questo si afferma subito come standard di fatto, al quale quasi tutti i costruttori si allineano proponendo macchine compatibili.



Figura 23. Il primo PC IBM (a) e il primo Apple Macintosh (b)

Con il personal computer l'elaboratore esce dal centro edp e arriva sulla scrivania di utenti non informatici: impiegati, professionisti, studenti. Dal punto di vista dell'interazione uomo-macchina, nasce una nuova era. Nell'arco di pochi anni vengono realizzate e diffuse applicazioni software innovative, concepite per un mercato di largo consumo, che non hanno eguali fra i prodotti dell'informatica tradizionale, per facilità d'uso e attenzione complessiva all'interfaccia con l'utente.

Rispetto al terminale video, due sono le grandi novità che modificano drasticamente la qualità dell'interazione. Da un lato, la potenza di calcolo *locale* permette al computer di reagire agli stimoli dell'utente con tempi di risposta quasi immediati, a differenza dei terminali tradizionali, che dovevano attendere la risposta dell'elaboratore collegato da linee dati ancora relativamente lente. Dall'altro lato, la possibilità di archiviazione locale cambia profondamente il rapporto dell'utente con i dati gestiti dall'elaboratore. Questo, da depositario di dati conservati in archivi centrali e accessibili soltanto per il suo tramite, diviene piuttosto uno strumento di manipolazione dei dati, che l'utente gestisce in modo flessibile, decidendo di volta in volta dove depositarli, se su supporti asportabili (dagli iniziali floppy-disc fino agli attuali pen-drive) oppure sul disco interno alla macchina.

Il personal computer introduce una forte discontinuità con il passato, e avvia un processo di radicale trasformazione degli strumenti informatici. Dal punto di vista dell'interazione uomo-computer, l'esempio più significativo è probabilmente quello del *foglio elettronico* (o *spreadsheet*), la "killer application" del personal computer, cioè il prodotto software che, da solo, ha più contributito alla rapida diffusione sul mercato di questo strumento.

Visicalc, il primo foglio elettronico, fu realizzato da Dan Bricklin e Bob Frankston inizialmente per il personal computer Apple II nel 1979 (Figura 24). Esso sfruttava appieno la possibilità di un'interazione estremamente rapida con il calcolatore. Con un programma di questo tipo, utente e computer operano alternativamente su una tabella visibile sullo schermo, contenente dati legati fra loro da relazioni matematiche, anche molto complesse. Ogniqualvolta l'utente aggiorna un valore in tabella, il programma, in modo pressoché istantaneo, aggiorna tutti i valori correlati, permettendo all'utente di verificare le conseguenze delle proprie variazioni (in un bilancio, in un budget, in un piano di vendite, ...), e di modificare nuovamente, di conseguenza, i valori forniti. Il computer diviene uno strumento flessibile di simulazione *what if* ("che cosa succederebbe se..."), in un rapporto interattivo con il suo utente sconosciuto in passato. Per non rallentare queste manipolazioni, i menu e i dati (che nelle applicazioni transazionali classiche apparivano in momenti diversi sul video) vengono ora collocati assieme in un'unica schermata, che contiene sia la tabella del foglio elettronico che un menu orizzontale, ridotto a una o due righe dello schermo *(menu bar)*.

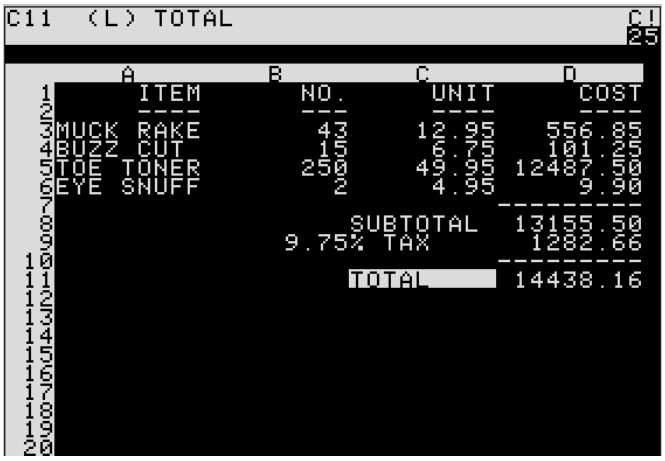


Figura 24. Visicalc per Apple II (1979)

Il sistema operativo del PC IBM, l'MS-DOS della neonata Microsoft, aveva un'interfaccia utente piuttosto rudimentale, basata su un linguaggio di comandi del tipo scrivi-e-leggi, e quindi di concezione piuttosto antiquata. La grande rivoluzione, che determinerà il paradigma del *personal computing* fino ai nostri giorni, doveva nascere nei laboratori del PARC (Palo Alto Research Center) della Xerox. Qui, un gruppo di pionieri della HCI stava da tempo sperimentando prototipi di *personal workstation* per il lavoro di ufficio. Il risultato principale di queste attività fu il sistema Star (1981), un computer personale dotato di elevata potenza di calcolo, video grafico di buona risoluzione, a doppia pagina, tastiera e *mouse* (Figura 25).

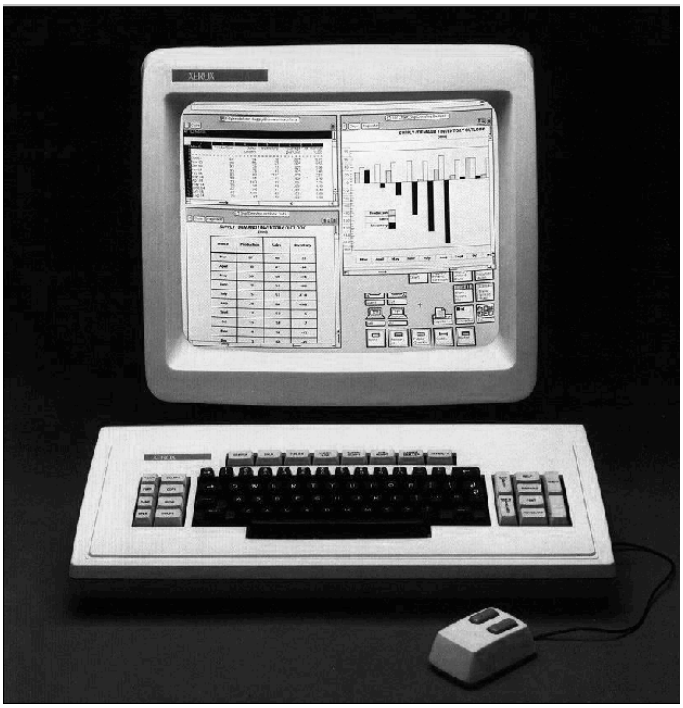


Figura 25. Personal computer Xerox Star (1981)

Il mouse, inventato da Douglas Engelbart a metà degli anni 60, apparirà per la prima volta sul mercato con lo Star, e diventerà in seguito il corredo standard di ogni personal computer. L'utente, spostando il mouse sul piano della scrivania, è in grado di muovere un *puntatore* sul video, con grande precisione e rapidità. Il mouse è dotato, secondo i

modelli, di uno, due o tre pulsanti, premendo i quali è possibile comunicare al calcolatore informazioni aggiuntive. È, in sostanza, come se l'utente indicasse un oggetto sul video e dicesse, contemporaneamente, "questo!".

Il mouse introduce possibilità d'interazione completamente nuove rispetto alla semplice tastiera, permettendo di comunicare con il calcolatore, per così dire, a gesti. Nonostante la sua semplicità, esso permette infatti una buona varietà di azioni: puntare (*pointing*), cliccare (*clicking*) una o due volte (*double-clicking*), col tasto sinistro o destro del mouse (*right-clicking*), premere (*pressing*), trascinare (*dragging*). La disponibilità di un video grafico a buona risoluzione, su cui rappresentare oggetti con accurata resa grafica, e di un mouse con cui indicare, selezionare, trascinare questi oggetti qua e là per lo schermo, suggeriscono allora un paradigma d'interazione del tutto nuovo, che possiamo sintetizzare con lo slogan:

NON DIRLO, FALLO

A questo paradigma è stato dato il nome di *manipolazione diretta*[15], perché l'utente può operare direttamente sugli oggetti rappresentati graficamente sul video, selezionandoli, spostandoli, manipolandoli in vario modo. In questo modo si elimina - o almeno si riduce notevolmente - l'intermediazione del linguaggio scritto nella comunicazione fra uomo e calcolatore. Anche se gli oggetti grafici da manipolare possono essere di qualsiasi tipo, il paradigma della manipolazione diretta è stato principalmente utilizzato per la realizzazione di sistemi basati sulla *metafora della scrivania (desktop)*. Questa, inventata dai progettisti del PARC della Xerox, apparve per la prima volta sullo Star, con tutti gli elementi base dei sistemi odierni: *finestre* (*windows*), *icone* che rappresentavano *documenti* e *cartellette* (*folder*), *menu* (Figura 26). Lo Star però era ancora troppo costoso, e fu un flop commerciale. L'idea fu invece portata con grande successo sul mercato di massa dalla Apple con il primo Macintosh (1984), una macchina a basso costo con un'interfaccia desktop molto semplificata, ma eccezionalmente gradevole (Figura 27). Successivamente il paradigma fu adottato dalla Microsoft, e si diffuse universalmente, a partire dal sistema Windows 3.0 (1990) e soprattutto con Windows 95 (Figura 28), fino a diventare il paradigma standard per i personal computer. Nonostante le differenze e i numerosi miglioramenti introdotti nei vari sistemi, gli elementi costitutivi di questo stile di comunicazione sono, a trent'anni di distanza, ancora quelli proposti nel 1981 con il sistema Star. A questo stile d'interfacce si dà spesso il nome di WIMP, acronimo per Windows, Icons, Mouse, Pointer.

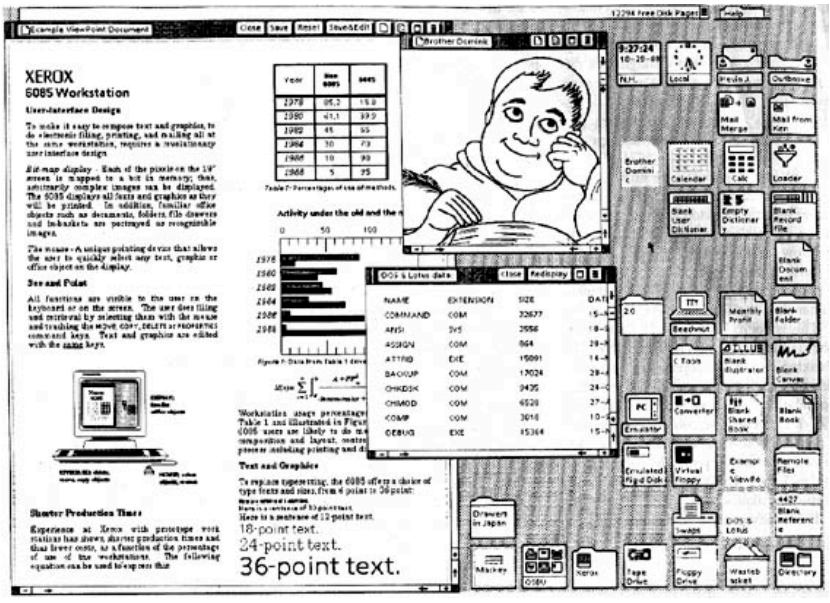


Figura 26. Il desktop dello Xerox Star (1981)

[15] Il nome è stato coniato in B.Shneiderman, *Direct manipulation: a step beyond programming languages*, in IEEE Computer 16(8) (agosto 1983), pagg.57-69.

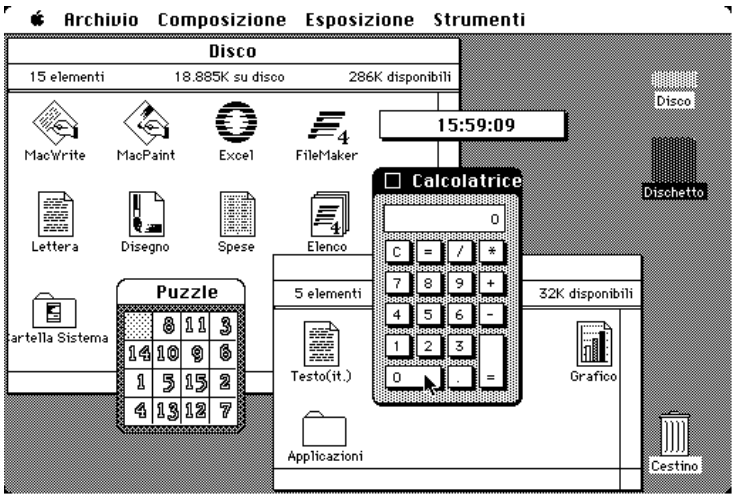


Figura 27. Il desktop del primo Apple Macintosh (1984)

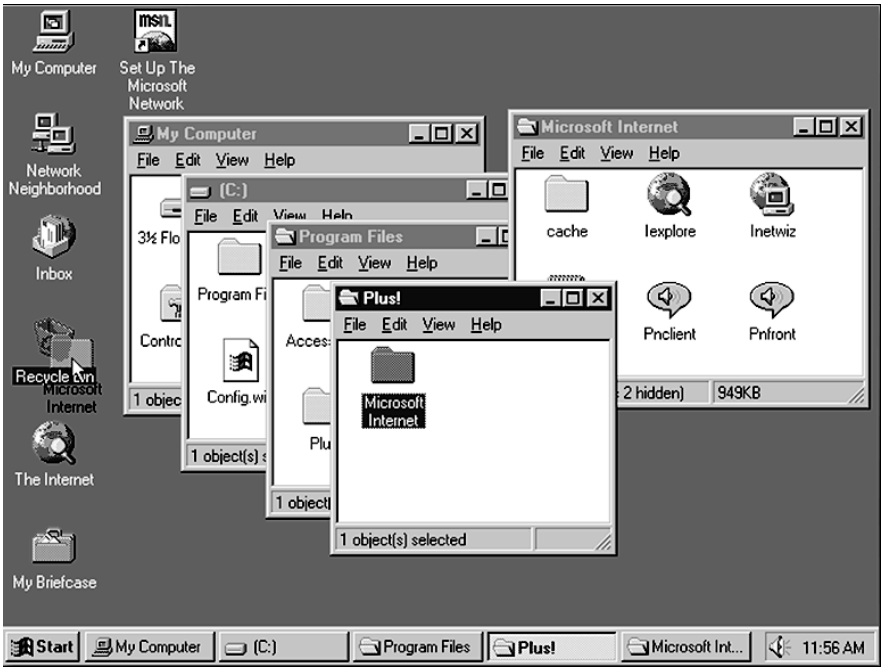


Figura 28. Il desktop di Microsoft Windows 95 (1995)

Il paradigma della manipolazione diretta sarà poi applicato, oltre che nelle interfacce di tipo desktop, nei sistemi basati sulla *metafora del pannello di controllo*. Qui, il video tende ad assomigliare al pannello di un'apparecchiatura elettronica, con pulsanti, interruttori, slider, spie luminose, display numerici, ... Esempi di questa tendenza sono numerosi, dai sistemi che realizzano quadri di comandi per impianti di controllo di processo, ai vari simulatori, come il classico Flight Simulator della Microsoft. In esso, a partire dai primi anni ’80, il video mostrava, fedelmente riprodotto nei dettagli, il quadro dei comandi di un aereo (indicatore della velocità dell'aria, altimetro, orizzonte artificiale, bussola, misuratore del carburante, pressione e temperatura dell'olio, ...).

Oggi si preferisce usare il termine “manipolazione diretta” quando l’interazione avviene *direttamente* su uno schermo tattile. In effetti, l’interazione attraverso il mouse o dispositivi quali *touchpad* o tavoletta grafica è di tipo *indiretto*:

invece di operare sullo schermo, l'utente opera sul piano della scrivania (reale) o sulla tavoletta. Con la tecnologia degli schermi *multi-touch*, la manipolazione diretta degli oggetti rappresentati sullo schermo si arricchisce in modo sostanziale, permettendo di utilizzare nel dialogo uomo-macchina una gestualità naturale, sviluppata nell'interazione con gli oggetti reali. La Figura 29 ne mostra un semplice esempio: i gesti utilizzati per ingrandire (a sinistra) e rimpicciolire (a destra) un'immagine visualizzata sullo schermo di un iPhone. Non è difficile trovare altri esempi, più complessi.



Figura 29. Esempio di manipolazione diretta con schermi multi-touch (dal manuale dell'Apple iPhone, 2007)

Il browser web: point & clic

L'uso del mouse, con l'azione di puntare e cliccare come metodo base dell'interazione, suggerisce naturalmente l'idea di presentare sul video dei "bottoni" virtuali sui quali operare premendo i bottoni (reali) del mouse. Si delinea così un nuovo stile del dialogo uomo-computer, in cui la comunicazione di base è molto semplificata, riducendosi alla semplice pressione di bottoni. I bottoni stessi si evolvono e, per così dire, si smaterializzano, trasformandosi in *testo cliccabile* o *aree sensibili* su immagini grafiche:

POINT & CLIC

Questa modalità d'interazione si diffonderà, a partire dalla fine degli anni '80, con i primi sistemi per la gestione di *ipertesti.* Essenzialmente, un ipertesto è un testo costituito di parti chiamate *nodi*, fra loro collegati da *link*, che associano nodi semanticamente correlati (Figura 30). Il lettore di un ipertesto non è più vincolato a una lettura sequenziale, ma lo può "esplorare" lungo percorsi diversi e personalizzati, in funzione dei suoi interessi.

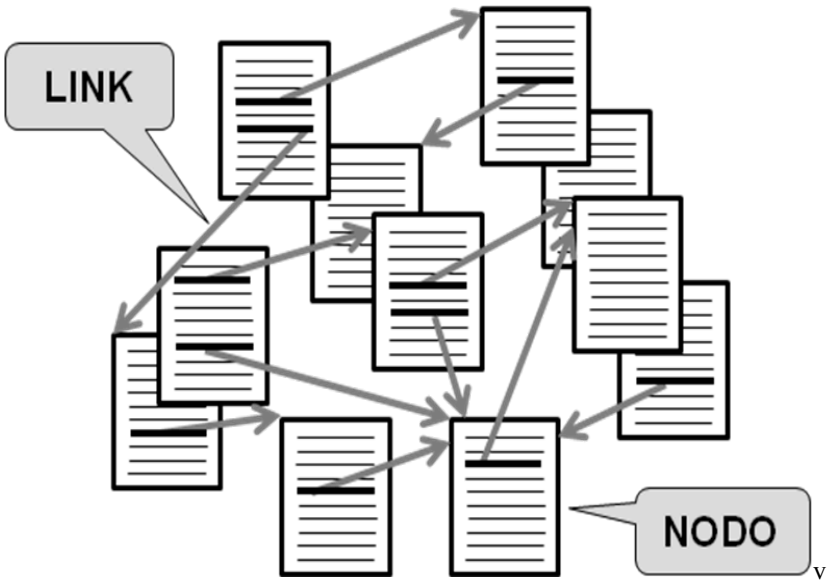


Figura 30. Ipertesto

Nel caso in cui i nodi contengano, oltre al testo, anche componenti multimediali (immagini, audio, animazioni, video) si usa il termine più generale di *ipermedia*. L'anno di diffusione su larga scala dell'idea di ipermedia può essere considerato il 1987, con due eventi importanti: la prima edizione del convegno biennale dell'ACM su questa tecnologia (*ACM Conference on Hypertext and Hypermedia*), e il lancio sul mercato di Hypercard, il software per la creazione la gestione di ipermedia realizzato da Bill Atkinson per il Macintosh della Apple. Questo programma, per quel tempo rivoluzionario, permetteva di costruire ipertesti (chiamati *stack*) composti da pagine grafiche (chiamate *card*), sulle quali si potevano definire delle aree sensibili al clic del mouse, ciascuna collegata a un link. Cliccando su una di tali aree si attivava il link, che portava all'esecuzione di un programma (*script*) ad esso associato, scritto in un linguaggio di programmazione molto semplificato (*hypertalk*). Nei casi più semplici, questo script si limitava a prelevare dallo stack una specificata card, e a mostrarla sul video (Figura 31). In altri casi, poteva eseguire calcoli complessi.

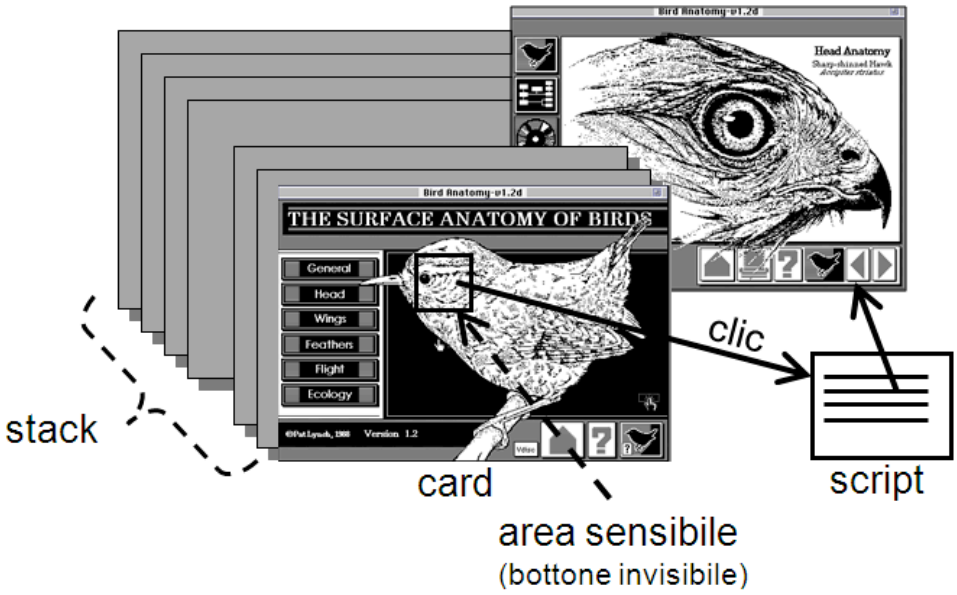


Figura 31. Hypercard (Apple, 1987)

Per i primi anni gli ipertesti furono esclusivamente off-line, realizzati per il nuovo mercato dei CD multimediali. La vera diffusione su scala planetaria del paradigma "point and click" avverrà però solo qualche anno dopo, dai primi anni '90, con l'invenzione del Web per opera di Tim Berners-Lee. Il World Wide Web altro non è, infatti, che un gigantesco ipertesto, i cui nodi (chiamati *pagine*) non sono contenuti in un unico archivio locale (come in Hypercard), ma sono distribuiti geograficamente, sui server connessi alla rete Internet. L'utente naviga all'interno del Web attraverso un *browser:* un programma per personal computer che, sostanzialmente, visualizza una pagina web e supporta l'utente nella navigazione attraverso le funzioni di point & clic. Con questa tecnologia, il Web assume ben presto una dimensione planetaria. La crescita del World Wide Web è esponenziale: dal primo sito messo in rete da Berners-Lee nel 1991, si arriverà agli oltre 200 milioni di siti a inizio 2010, distribuiti su centinaia di milioni di server (oltre 700 milioni nel 2009). La Figura 32 mostra una stima della crescita dei siti web nel mondo, a partire dal 1996. La linea più alta rappresenta il numero totale di siti, quella più bassa solamente i siti che risultano attivi.

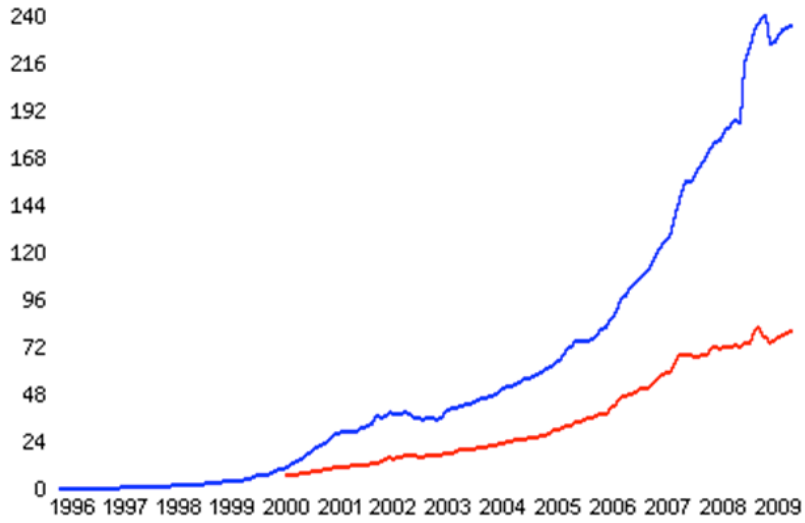


Figura 32. Siti web nel mondo (x1000)[16]

Per risolvere il problema del reperimento delle informazioni all'interno di questo spazio, nascono allora le *directory* (per esempio Yahoo) e i *motori di ricerca*, che indicizzano l'enorme quantità di informazioni presenti sul Web e ricercano, in questi indici, le parole chiave fornite dall'utente.

Attraverso il browser web, che convive con l'interfaccia classica di tipo desktop del personal computer, gli utenti imparano a *navigare* nella rete. In pochi anni, il personal computer assume una dimensione nuova: da sistema stand-alone su cui svolgere prevalentemente lavoro di word processing, calcolo e archiviazione personale, a *information appliance*, un dispositivo per la ricerca e l'accesso all'informazione in rete. I comportamenti degli utenti si modificano radicalmente: la diffusione della banda larga e la riduzione dei costi di connessione permettono a una crescente percentuale di utenti collegamenti *always-on*, che permettono un accesso potenzialmente istantaneo a tutta l'informazione disponibile sul Web.

Nonostante questa profonda trasformazione, l'interfaccia dei personal computer resta sostanzialmente ancorata al paradigma del desktop, nato più di due decadi prima. Il programma di accesso alla mail e il browser web non sono in alcun modo delle applicazioni privilegiate, nonostante che in molti casi l'uso del PC sia sostanzialmente incentrato su questi due programmi. In particolare, i browser web non si discostano sostanzialmente dal modello proposto inizialmente da Mosaic (1993): un visualizzatore di pagine web – una pagina alla volta, nella finestra aperta sul desktop – tra le quali navigare con il modello del point&click e con l'ausilio di un motore di ricerca.

[16] Dati da http://www.netcraft.com , dicembre 2009. Si veda anche http://www.gandalf.it per statistiche aggiornate sul Web.

Se l'interfaccia rimane la stessa, il Web evolve continuamente, e in modo radicale. Dai primi anni 2000, permette all'utente di trasformarsi, da navigatore e consumatore passivo dell'informazione presente in rete, a creatore di contenuti. Si diffondono sistemi che permettono agli utenti di effettuare l'*upload* di propri contenuti in rete, senza che per questo sia necessario disporre di particolari competenze tecniche e di server propri. In molti casi, il servizio è disponibile gratuitamente, o a costi molto contenuti. Nascono milioni di *blog*, in cui gli utenti pubblicano i loro pensieri, gigantesche collezioni di immagini (per esempio www.flickr.com), di video (per esempio, www.youtube.com), di musica; nascono opere collettive realizzate da migliaia di volontari, come www.wikipedia.com. Sulla copertina del Natale 2006 della rivista *Time*, tradizionalmente dedicata al personaggio dell'anno, non compare alcun ritratto, ma semplicemente uno specchio fissato sullo schermo di un personal computer, con la scritta: YOU. L'utente, da navigatore, si è fatto *protagonista* del Web. A partire dai primi anni del nuovo secolo, la rete diventa l'infrastruttura di base attraverso cui gli utenti comunicano, socializzano, interagiscono e collaborano fra loro. Il Web, che prima era considerato sostanzialmente un gigantesco ipertesto, si trasforma nel *social Web*. Gli utenti, a milioni, si tengono costantemente in contatto attraverso i siti di *social networking:* nel 2010, Facebook (fra i tanti, il più popolato) conta più di 350 milioni di utenti.

I siti web, da contenitori di informazioni accessibili sostanzialmente in sola lettura, evolvono radicalmente, e si trasformano in applicazioni software a disposizione (molto spesso gratuitamente) degli utenti. La rete è vista come un gigantesco computer, in grado di erogare informazioni e servizi applicativi a milioni di utenti (*cloud computing*).[17] In questo nuovo contesto, il paradigma point&clic si arricchisce di possibilità. L'utente "punta e clicca" non solamente per navigare fra le pagine, ma anche per eseguire applicazioni. Mentre nel vecchio Hypercard le applicazioni risiedevano localmente sul proprio PC, ora sono servizi erogati attraverso la rete da sistemi remoti, potenzialmente sparpagliati sull'intero pianeta. Spesso un'applicazione integra servizi provenienti da fornitori diversi, senza che l'utente ne sia consapevole (*mashup*). Un esempio paradigmatico è costituito da Hyperwords, un semplice *plugin* per browser. Esso permette all'utente, cliccando su una parola di una qualsiasi pagina web, di attivare, su quella parola, un servizio di rete scelto fra un menu di possibilità. La Figura 33 ne mostra un esempio. Cliccando su una parola qualsiasi (in questo caso, "United Kingdom") e selezionando Reference e poi Google Definition dai menu che appaiono, viene visualizzata la definizione di "United Kingdom" secondo Google. Cliccando invece Reference e poi Wikipedia comparirà la pagina di Wikipedia relativa alla stessa parola. Le possibilità sono numerose, quanto i servizi in rete. Per esempio, è possibile trovare la parola in un dizionario, ricercarla con il motore di ricerca preferito, trovarne le illustrazioni disponibili in un repository d'immagini, perfino tradurla in una lingua selezionata, usando il traduttore di Google. Il tutto è realizzato, semplicemente, attivando il servizio e trasmettendogli la parola selezionata come parametro.

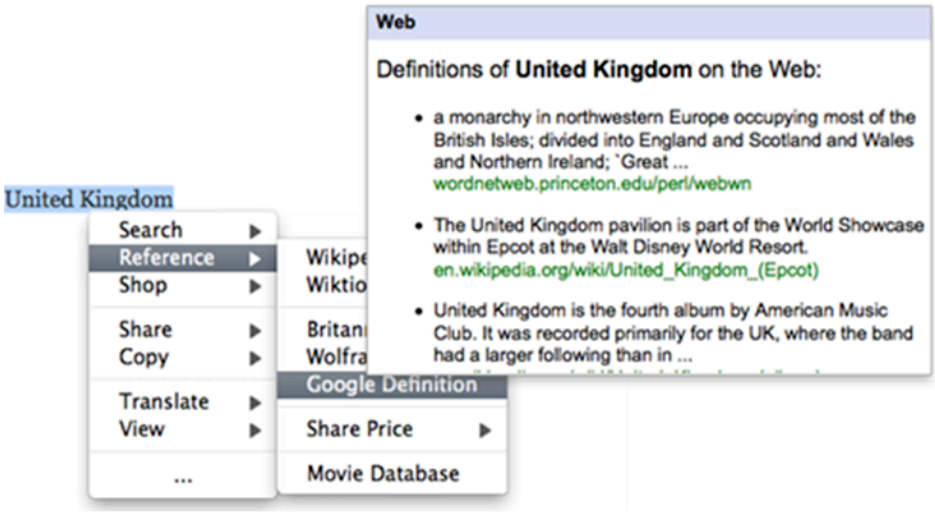


Figura 33. http://www.hyperwords.com (2009)

[17] Il termine *cloud*, inglese per "nuvola", deriva dal fatto che, nelle rappresentazioni grafiche, la rete internet viene molto spesso rappresentata a forma di nuvola.

Il mobile: alzati e cammina

Il paradigma della *mobilità* (*mobile computing*) ha una storia che inizia da lontano. Già nei primi anni '70 Alan Key, allora ricercatore al PARC della Xerox nel *dream team* che avrebbe determinato gran parte dell'evoluzione dei modelli d'uso del calcolatore, aveva immaginato uno strumento che per quei tempi sembrava fantascientifico, al quale aveva dato il nome di Dynabook, per *dynamic book*, libro dinamico. In un articolo visionario del 1977, aveva scritto:

> *Immaginate di avere il vostro personale manipolatore di conoscenza in confezione portatile, della dimensione e della forma di un normale blocco da appunti. Supponete che esso abbia abbastanza potenza da superare i vostri sensi della vista e dell'udito, abbastanza capacità da immagazzinare, per un successivo reperimento, migliaia di pagine-equivalenti di materiale di riferimento, poemi, lettere, ricette, registrazioni, disegni, animazioni, spartiti musicali, forme d'onda, simulazioni dinamiche, e qualunque cosa voi possiate desiderare di ricordare e modificare. Pensiamo a un apparecchio piccolo e portatile per quanto è possibile, in grado sia di ricevere sia di fornire informazioni in quantità paragonabili a quelle del sistema sensorio umano. L'output visivo dovrebbe essere, almeno, di qualità migliore di quella ottenibile dai giornali. L'output uditivo dovrebbe avere un simile standard di alta fedeltà. Non ci dovrebbe essere alcuna pausa percepibile fra causa ed effetto. Una delle metafore che abbiamo usato per progettare tale sistema è quella di uno strumento musicale, come un flauto, che è posseduto dal suo utilizzatore e risponde in modo istantaneo e consistente ai desideri del suo proprietario. Immaginate quanto sarebbe assurdo il ritardo di un secondo fra l'atto di soffiare una nota e il suo ascolto!* [18]

Uno strumento di questo tipo era ancora ben lontano dalle possibilità della tecnologia, hardware e software: lo si poteva solo immaginare o, al più, visualizzare con modellino di cartone (Figura 34). Tuttavia il lavoro di Alan Kay ebbe una grandissima influenza sulle evoluzioni successive dei modelli d'interazione.



Figura 34. Modello di cartone del Dynabook, circa 1971-72[19]

Subito dopo la diffusione dei personal computer all'inizio degli anni '80, accanto alle macchine destinate a una postazione fissa (chiamati *desktop computer,* o computer da scrivania), iniziarono ad apparire sul mercato computer personali *portatili*. I primi modelli non assomigliavano né al Dynabook né ai portatili di oggi, erano ingombranti e pesanti. L'Osborne 1, commercializzato nel 1981 quasi contemporaneamente al primo PC IBM, pesava circa 11 chili ed

[18] A. Kay, A. Goldberg, *Personal Dynamic Media*, in *Computer*, vol. 10, no. 3, pp. 31-41, Mar. 1977, anche disponibile in rete in http://www.newmediareader.com/book_samples/nmr-26-kay.pdf.
[19] ibid.

era privo di batteria. La situazione migliorò dal 1982-83, quando apparvero i primi *laptop*, cioè dei PC che potevano essere tenuti "in grembo" (*lap*) e il cui schermo poteva essere ripiegato sulla tastiera, "a conchiglia". Mancando l'appoggio sulla scrivania, il mouse dovette essere sostituito da altri dispositivi di manipolazione, fra cui i *touchpad* disposti sulla tastiera. I laptop meno ingombranti, idealmente delle dimensioni di un libro da appunti, furono in seguito chiamati *notebook* (Figura 35). I notebook specialmente concepiti per l'utilizzo in rete (mail, browsing, servizi applicativi in rete), piccoli e leggeri (per esempio, meno di due chili) e con buona autonomia e basso costo, furono allora chiamati *netbook* (circa 2007).



Figura 35. Laptop (Lenovo ThinkPad, circa 2008)

Con i notebook, e soprattutto con i netbook dotati di accesso wireless alla rete, l'utente non ha più la necessità di disporre di una postazione fissa: può portare con sé una stazione d'accesso a Internet utilizzabile da qualsiasi località con copertura di rete (*nomadic computing*). Questo modello d'interazione è vincente, e sostituirà in breve tempo quello da postazione fissa. La data che simbolicamente inizia l'"era dei notebook" è il settembre 2009: nel terzo trimestre del 2009, infatti, le vendite di notebook in tutto il mondo superano, per la prima volta, quelle dei PC da scrivania. Da questo momento, il notebook si afferma come lo strumento per tutti, e non più soltanto per il mercato business, al quale si era inizialmente indirizzato.[20] Non si tratta ancora, tuttavia, di un uso in mobilità: notebook e netbook, per quanto piccoli e leggeri, devono comunque essere usati "da fermo e da seduto". Le modalità d'interazione, per quanto più snelle, sono ancora sostanzialmente quelle messe a punto per l'accesso da postazioni fisse, a parte l'uso del mouse, sempre più spesso sostituito dai touchpad, anche realizzati con tecnologie *multi-touch*.[21] L'uso di strumenti di comunicazione e di calcolo "in mobilità e in piedi" (*mobile computing*) richiede modalità d'interazione completamente diverse.

Queste modalità iniziano a diffondersi dagli anni '90, con l'introduzione dei *palmtop* (da *palm*=palmo della mano) o, in italiano, *palmari,* piccoli computer chiamati così perché capaci di stare in una mano. Con rifermento alle funzioni inizialmente prevalenti, questi computer sono stati spesso denominati *PDA*, per *personal digital assistant* (*assistenti digitali personali*). I primi PDA non si connettevano alla rete, ma servivano a gestire la rubrica di indirizzi, l'agenda degli appuntamenti (in sincronizzazione con quelle sul desktop), a prender appunti, e così via. Il primo PDA di grande successo, il Palm Pilot (Figura 36), sul mercato dal 1996, era dotato di un piccolo schermo tattile resistivo e di uno *stilo*, con cui l'utente poteva scrivere testi utilizzando una tastiera virtuale sul video, oppure utilizzando una forma semplificata di scrittura a mano libera, denominata Graffiti. Una parte di questa alfabeto è riportato in Figura 37 (il pallino su ogni carattere indica da dove si deve iniziare per tracciarlo).

[20] Cfr. http://www.isuppli.com/NewsDetail.aspx?ID=19823.

[21] Il mouse è stato uno dei device più di successo nella storia dell'informatica. La sola Logitech ha prodotto, a tutto il 2008, più di un miliardo di mouse, di diversi modelli.

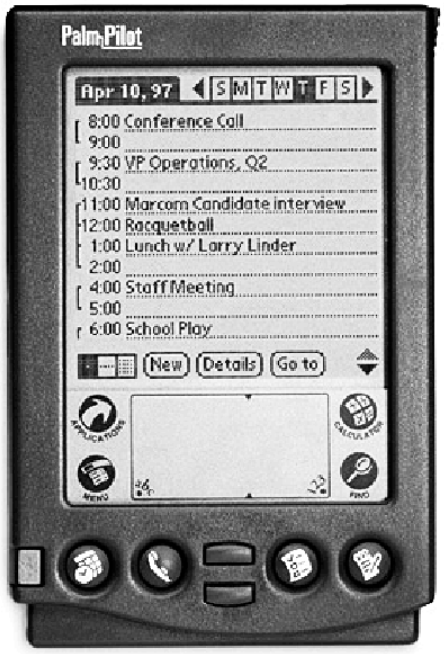


Figura 36. Palm Pilot 1000 (1996)

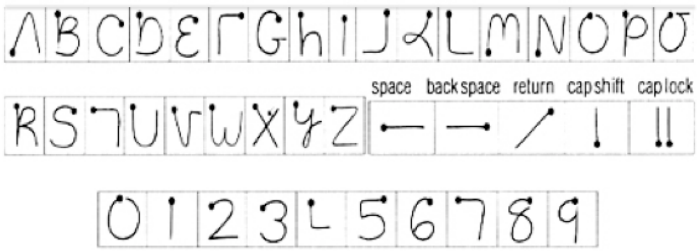


Figura 37. Alfabeto Graffiti (Palm Pilot)[22]

Negli stessi anni, si assiste allo sviluppo esplosivo della telefonia mobile. Inizialmente i telefoni cellulari sono semplici, fornendo le funzionalità di base per effettuare telefonate e inviare sms. Ma ben presto il telefono cellulare inizia a integrare funzioni sempre più complesse, e si trasforma rapidamente in un dispositivo del tutto nuovo. Se ne esaminiamo l'evoluzione dal punto di vista delle modalità d'interazione con l'utente, possiamo identificare cinque generazioni di apparati[23], che si susseguono (e si sovrappongono) sul mercato nello spazio di pochi anni (vedi Figura 38).

[23] Cfr. Brian Fling, *Mobile Design and Development*, Ed. O'Reilly, 2009.

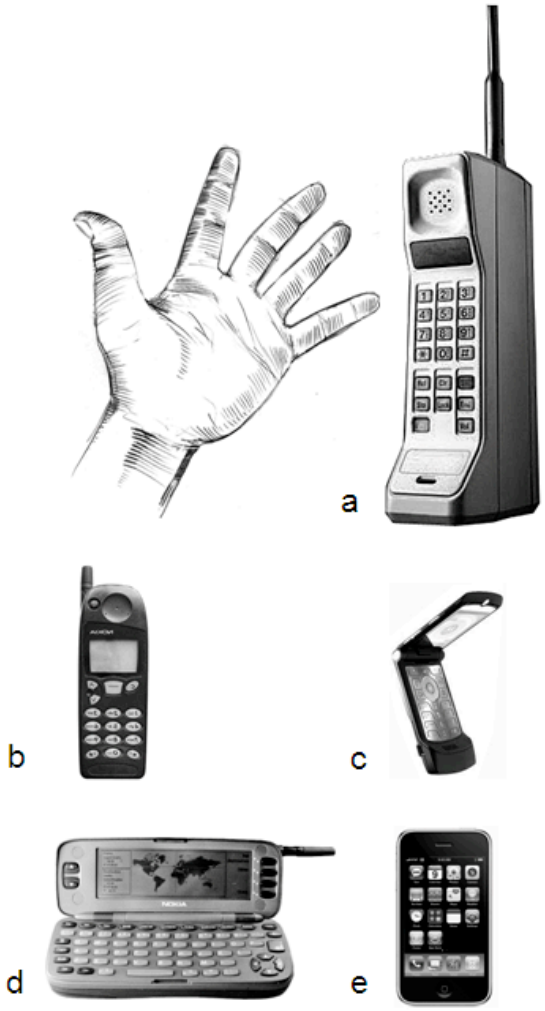


Figura 38. Modelli tipici delle 5 generazioni di telefoni cellulari.
a: Motorola Dyna TAC 8000 (1G, 1983); b: Nokia 5110 (2G, 1998);
c: Motorola V3 RAZR (2.5G, 2005); d: Nokia 9000 Communicator (3G, 1996);
e: Apple iPhone 3G (2008). Le dimensioni sono approssimativamente in scala.

- Prima generazione (1G, circa 1978 1988).

E' la "preistoria" della telefonia mobile. Gli apparati sono ingombranti e costosi, anche per il peso di batterie con la potenza sufficiente per collegarsi a una rete poco sviluppata e quindi geograficamente molto sparsa. L'uso è limitato a utenti con necessità di telefonare in mobilità per lavoro (per esempio, rappresentanti, venditori, ecc.). Un prodotto tipico di questa generazione è il DynaTAC della Motorola, introdotto nel 1983 (Figura 38a).

- Seconda generazione (2G, circa 1988-1998)

Con la seconda generazione inizia la diffusione di massa della telefonia mobile. La disponibilità della tecnologia, le necessità degli utenti e i notevoli investimenti effettuati per la costruzione di una rete cellulare capillare avviano un circolo virtuoso che determina una crescita esplosiva del mercato, soprattutto in Giappone e in Europa. La densità delle nuove reti cellulari permette di usare batterie molto più piccole, e i telefoni diventano tascabili, assumendo la caratteristica forma detta *candybar*.[24] I costi dei device si riducono sostanzialmente, anche se le tariffe (a consumo)

[24] Il termine inglese *Candybar* denota i dolci industriali a forma di barretta – spesso biscotti ricoperti di cioccolato.

sono alte. Le tecnologie per la trasmissione sono varie, ma prevale lo standard GSM, promosso soprattutto in Europa. Nascono gli operatori di telefonia mobile e inizia una radicale trasformazione dell'industria delle telecomunicazioni.

In questa fase ci si rende conto che i telefoni possono servire non solo per telefonare, anche se i prodotti sul mercato offrono ancora funzioni aggiuntive molto semplici: sms (dal 1993), orologio, sveglia, rubrica, calcolatrice, giochi, suonerie. Gli sms, inizialmente concepiti per inviare messaggi di servizio, hanno un enorme e inaspettato successo, anche a causa dei bassi (o nulli) costi applicati dagli operatori. Gli schermi sono piccoli e monocromatici, le tastiere hanno 12 tasti. Un esempio è il Nokia 5110, prodotto tra il 1998 e il 2001, uno dei telefoni più popolari dell'epoca. (Figura 38b).

- Generazione 2.5 (2.5G, circa 1998-2008)

Questa generazione può essere considerata di transizione. La tecnologia di comunicazione evolve, per permettere al cellulare di trasmettere e ricevere dati a media velocità (es.: GPRS). Il cellulare si arricchisce di una varietà di funzioni e servizi, e integra una fotocamera, in conseguenza dell'evoluzione della fotografia digitale. E' possibile inviare e ricevere immagini (mms), installare suonerie, giochi e applicazioni software di vario tipo. È possibile collegarsi a internet per accedere alla posta elettronica e al Web. Internet si apre all'accesso mobile, anche se, in questa fase, l'utilizzo resta molto limitato, soprattutto per l'inadeguatezza degli apparati (che hanno schermi troppo piccoli e tastiere inadatte) e del numero ancora limitato di siti predisposti a un accesso mobile. Un esempio tipico è il Motorola V3 RAZR (Figura 38c), dalla caratteristica forma a conchiglia (*clamshell*, detta anche *flip*). Esso, prodotto in oltre 100 milioni di esemplari, è uno dei cellulari più venduti di tutti i tempi. I prodotti di questa generazione sono numerosissimi, di forma diversa (prevalentemente *candybar* e *clamshell*).

- Terza generazione (3G, dal 2002)

Questa è l'era dei cosiddetti *smartphone*, che appaiono sul mercato poco dopo gli apparecchi della seconda generazione, e convivono con essi. Anche se non esiste una definizione standard di smartphone, possiamo dire che esso offre tutte le funzioni dei cellulari delle generazioni precedenti, ma possiede di solito uno schermo più grande, una tastiera alfanumerica (per esempio QWERTY) o uno stilo per scrivere su una tastiera virtuale, una connettività alla rete a banda larga resa possibile da protocolli di comunicazione di terza generazione, a volte una connettività Wi-Fi. Il telefono è dotato di sistema operativo e assomiglia sempre più a un piccolo computer portatile. Integra le funzioni caratteristiche dei vecchi PDA (agenda personale, appunti, ecc.), che in pratica scompaiono dal mercato. La tecnologia di comunicazione permette l'accesso a internet a velocità abbastanza elevata. Le funzioni sono molto varie: telefonia, sms, email, accesso al Web, fotocamera, multi-media messaging (mms) per inviare e ricevere foto o video, connettività bluetooth, giochi, player MP3, radio, GPS, con possibilità di installare una grande varietà di applicazioni anche prodotte da terze parti.

Un precursore importante di questa categoria di apparati fu il 9000 Communicator della Nokia, una sorta di ibrido fra un telefono e un piccolo laptop, sul mercato dal 1996 (Figura 38c). Ma la varietà delle forme proposte è molto alta, i costruttori sono alla ricerca di una forma che il mercato accolga con favore. Oltre alla forma a conchiglia (come il Treo e lo stesso Communicator) compaiono apparecchi a *brick* (mattoncino), come il Blackberry, a *slider* (in cui la tastiera, di notevoli dimensioni, non si ripiega sul video, come nelle forme nella conchiglia, ma "scorre" sotto di esso). La forma di molti di questi apparecchi permette di utilizzare la posta elettronica in modo abbastanza agevole. L'uso mobile della email quindi cresce considerevolmente. Tuttavia, gli smartphone conquistano solo una piccola parte del mercato della telefonia mobile. Apparecchi più semplici e meno costosi continuano ad avere enorme diffusione e, come vedremo fra poco, a diffondersi con straordinaria rapidità anche nei paesi più poveri.

- Il mobile

Nel giugno del 2007, la Apple lancia sul mercato l'iPhone, un apparato che modifica profondamente la concezione degli apparati mobili, e ridefinisce in modo significativo questa industria. La forma, a brick (Figura 38e) è quasi interamente occupata da uno schermo multi-touch, di buona risoluzione (320 x 480 pixel), che permette di controllare le funzioni con una varietà di gesti delle dita, con la pressione di bottoni (vedi Figura 147 a pag.175) e con una tastiera virtuale (vedi Figura 187 a pag.218). Nonostante le dimensioni limitate, lo strumento integra un'ampia varietà di tecnologie:

fotocamera e player multimediale, GPS, mail e browser web, wi-fi, bussola digitale, input vocale e una varietà di sensori, di movimento (accelerometro), di prossimità, di luce ambientale. Anche se la maggior parte di queste tecnologie erano disponibili da tempo, esse sono assemblate in un modo del tutto innovativo. La risposta del mercato all'iPhone è enormemente favorevole: dopo un anno e mezzo dal lancio, le vendite assommano a 17 milioni di unità. E gli altri produttori annunciano ben presto prodotti ad esso ispirati. Finalmente, l'accesso mobile a Internet è una realtà.

Tre anni dopo l'annuncio dell'iPhone, lo store online della Apple proponeva più di 100.000 applicazioni software utilizzabili da questo device. Nello stesso periodo, gli utenti avevano effettuato, dallo stesso store, tre miliardi di download di applicazioni software.

Dal punto di vista dell'interazione uomo macchina, l'iPhone può essere considerato il primo device che incarna compiutamente il paradigma che abbiamo chiamato con il termine inglese *mobile*. Il mobile non è un telefono e non è un computer: è un oggetto del tutto nuovo, uno strumento insieme di comunicazione, d'informazione e d'interazione con l'ambiente. È pensato per essere sempre connesso alla rete, e destinato a un uso strettamente personale, quasi "intimo" da parte dell'utente, che lo porta con sé in ogni circostanza, senza spegnerlo mai. Con il suo sistema di sensori, il mobile è in grado di raccogliere automaticamente informazioni su se stesso, sull'utente e sull'ambiente, e di utilizzarle per fornire a chi lo usa informazioni contestualizzate. Questi nuovi "sensi" lo arricchiscono di potenzialità funzionali vastissime, ancora tutte da esplorare: servizi geo-referenziati (*location based services*) che, basati sul GPS, offrono all'utente informazioni sull'ambiente circostante; applicazioni di *augmented reality* che utilizzano la fotocamera ad alta risoluzione come "occhio" sulla realtà circostante, che viene integrata con informazioni specifiche, reperite in tempo reale dalla rete; servizi di identificazione di oggetti circostanti (per esempio attraverso lettura di tag RFID), e di interazione con gli stessi, per esempio per effettuare pagamenti, per segnalare la propria presenza, e così via.

ALZATI E CAMMINA!

Al momento della stesura di queste pagine, per il mobile computing non si è ancora consolidato un paradigma d'interazione ben riconoscibile al di là delle differenze fra i diversi dispositivi. Questo è dovuto a diversi fattori, fra i quali la diversità degli utenti cui questi prodotti si rivolgono. Il telefono cellulare – e la sua evoluzione – è lo strumento personale per eccellenza, e come tale deve riflettere gusti, esigenze, abilità, personalità del singolo utente. Tuttavia, la situazione è in evoluzione molto rapida, e la linea di tendenza sembra ormai abbastanza chiara. Il mobile, con la sua ricchezza funzionale, non deve essere considerato un gadget per appassionati di tecnologia. Con l'inevitabile abbattimento di costi prodotto dalla crescita del mercato e dalla concorrenza, è presumibile che la sua diffusione sia molto vasta, e che si avvii il ciclo virtuoso che abbiamo già visto per i personal computer e per i cellulari tradizionali. Già oggi il telefono cellulare, e non il personal computer è la tecnologia più diffusa. Secondo stime delle Nazioni Unite, nel 2007 sono stati venduti, nel pianeta, 1 miliardo di cellulari, e "solo" 400 milioni di PC. Secondo la stessa fonte, nel 2009 il 90% della popolazione del pianeta ha accesso alla telefonia mobile, eventualmente attraverso amici o parenti.

Il cellulare Nokia 1100 (Figura 39), progettato appositamente per i paesi in via di sviluppo e messo sul mercato nel 2003, con oltre 200 milioni di esemplari è stato il telefono e il prodotto di elettronica di consumo più venduto di tutti i tempi.[25] Si tratta di un modello GSM di costo contenuto e funzionalmente poco sofisticato: sms, sveglia, calcolatrice, lista di contatti, suonerie, giochi, torcia elettrica, protezione anti polvere e impugnatura antiscivolo per ambienti umidi.

[25] Per una classifica dei modelli di cellulari più venduti si veda http://en.wikipedia.org/wiki/List_of_best-selling_mobile_phones.



Figura 39. Il cellulare più diffuso (Nokia 1100)

L'uso del cellulare è comunque ancora in fortissima crescita. Secondo stime dell'International Telecommunication Union, nel 2002 il numero di abbonamenti di telefonia mobile (includendo nel conteggio le schede prepagate) ha superato, nel mondo, quello degli abbonamenti a un telefono fisso. Sempre nel 2002, il numero degli abbonati a un operatore di telefonia mobile erano il 19% della popolazione mondiale; nel 2008 erano saliti al 61%. Al confronto, gli utenti Internet sono molto meno: secondo stime della stessa ITU, nel 2008, erano "solo" il 23% della popolazione mondiale. [26]

Non c'è dubbio che il cellulare, dalla seconda metà degli anni '90, ha completamente modificato le abitudini comunicative della popolazione dell'intero pianeta (Figura 40). I nuovi paradigmi d'interazione che avranno la massima diffusione nasceranno dagli apparati mobili, e non dai PC.

Con l'accesso mobile alla rete, al Web tradizionale si affianca il *mobile Web*. Il primo è fatto di quei siti e servizi ai quali si accede dal browser del proprio computer (desktop, laptop o netbook), stando seduti. Il secondo è composto da quei siti e servizi che si utilizzano in mobilità, in qualunque istante e luogo, spesso stando in piedi. Dal punto di vista tecnico, il Web è sostanzialmente uno solo. Ma dal punto di vista dei paradigmi d'interazione, il desktop Web e il mobile Web sono due media completamente differenti.

[26] Cfr. International Telecommunication Union, *Measuring the Information Society*, The ICT Development Index, 2009, http://www.itu.int/ITU-D/ict/publications/idi/2009/material/IDI2009_w5.pdf.



Figura 40. Mobilità

Il social computing

Finora abbiamo posto la nostra attenzione sul singolo utente, che interagisce con un sistema di vario tipo, un terminale connesso a una macchina condivisa, oppure un dispositivo personale (Figura 1). Questa visione non riflette bene la grande varietà dei sistemi interattivi, e trascura un insieme di applicazioni che, col tempo, hanno acquisito un ruolo molto importante nella nostra vita quotidiana. Infatti, i computer possono essere anche utilizzati efficacemente come strumenti d'intermediazione e facilitazione della comunicazione fra persone. Già nel 1968, J.C.R.Licklider, un altro grande pioniere della HCI, aveva previsto che "entro pochi anni, gli uomini potranno comunicare più efficacemente attraverso una macchina che di persona".[27] Da allora, enormi progressi sono stati compiuti, e specifiche aree della disciplina della HCI si sono grandemente sviluppate, come per esempio l'area del *Computer Supported Cooperative Work* (CSCW), che si occupa di come le attività cooperative effettuate da gruppi di persone possono essere supportate e coordinate dai computer, includendo lo studio degli effetti psicologici, sociali e organizzativi di questo uso dei sistemi.

All'informatica individuale nata con i primi personal computer è subentrata pertanto l'*informatica sociale*: agli strumenti per l'individuo si affiancano strumenti per i gruppi e per le comunità, sempre più sofisticati. Facendo riferimento agli esempi di Figura 4 a pag.14, possiamo identificare i *sistemi personali*, che vengono utilizzati normalmente sempre dalla stessa persona (l'iPhone), i *sistemi mono-utente,* che vengono usati da una sola persona alla volta (il robot da cucina), e i sistemi *multi-utente,* usati contemporaneamente da più persone (il cruscotto dell'aereo, utilizzato dal pilota e dal co-pilota, oppure quei sistemi in cui diverse persone possono condividere lo stesso ambiente virtuale). A questi tipi di sistemi si aggiungono, quindi, i *sistemi sociali*, che non interagiscono solo con singoli individui in un'interazione uno-a-molti (come il cruscotto di Figura 4a), ma sono soprattutto strumenti

[27] J.C.R.Licklider, *The Computer as a Communication Device*, in Science & Technology, aprile 1968, disponibile anche in rete.

d'intermediazione fra interlocutori spazialmente e, spesso, temporalmente distanti. Essi permettono loro di comunicare e di collaborare in compiti complessi: sono strumenti d'intermediazione intelligente, che – sempre più spesso – entrano nel merito della conversazione, la supportano e la facilitano. Con la diffusione di Internet, questi sistemi possono a volte soddisfare le esigenze di comunicazione e di socializzazione d'intere collettività di grandi dimensioni, in qualche caso composte da diecine o centinaia di milioni d'individui.

I sistemi che realizzano tale intermediazione assumono varie forme e si appoggiano a tecnologie diverse, che evolvono continuamente in modo molto rapido: dai siti di social networking di vario tipo (a partire da Facebook, sviluppatosi in modo impressionante a partire dalla sua nascita nel 2004), alle piattaforme di blogging, fino alle applicazioni che supportano il lavoro cooperativo in rete di gruppi più meno ampi: wiki, online office suite, e così via.

Uno studio riferito alla fine del 2008[28] stimava che due terzi degli utenti di Internet visitassero blog o siti di social networking, totalizzando quasi il 10% del tempo globale speso in rete. Queste percentuali sono in continua crescita. L'enorme quantità di tempo impiegato nell'accesso a siti che permettono alle persone di interagire all'interno di comunità di varia dimensione e tipologia sta modificando i comportamenti dell'umanità. Le modalità d'interazione e di comunicazione fra le persone, già profondamente modificato con la diffusione della telefonia mobile, della posta elettronica e degli sms, assume continuamente nuove forme, via via che nuovi strumenti si diffondono in rete. Si consolida così un nuovo paradigma d'interazione, che possiamo chiamare *social computing*. Non più interazione fra più utenti e un sistema, ma interazione fra più utenti mediata da un sistema.

Dal punto di vista dell'interazione con l'utente, queste applicazioni non utilizzano dispositivi specifici, ma normali PC o netbook connessi in rete e, sempre più spesso, anche dispositivi che consentano l'accesso in mobilità. Al di là delle inevitabili differenze, la gran parte di questi sistemi è accomunata dalla presenza di *profili personali*, più o meno dettagliati, attraverso i quali ogni utente si mostra agli altri. I profili possono essere pubblici, o riservati a un sottoinsieme di utenti considerati *amici*, o agli amici di questi, secondo livelli di *privacy* definiti dall'autore di ciascun profilo. Ciò caratterizza fortemente questi sistemi, i quali possono essere considerati, a tutti gli effetti, delle "reti di persone" (*reti sociali*, nel senso attribuito a questo termine dagli studiosi di sociologia), di fronte alle quali le caratteristiche funzionali che li differenziano passano inevitabilmente in secondo piano (Figura 41).

[28] Nielsen, "Global faces and networked places", http://blog.nielsen.com/nielsenwire/wp-content/uploads/2009/03/nielsen_globalfaces_mar09.pdf, marzo 2009

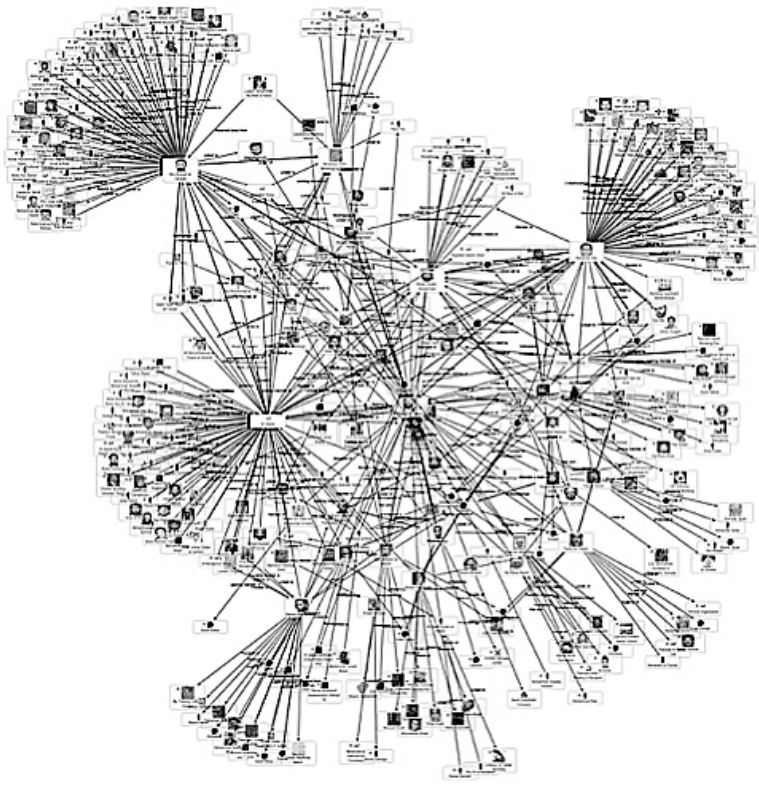


Figura 41. Mappatura di una social network in rete (da: http://www.fmsasg.com/SocialNetworkAnalysis/)

Tra queste comunità di utenti, la rete assume sempre più un ruolo attivo e intelligente: non più semplice intermediario per il trasporto o l'archiviazione delle informazioni, ma interlocutore a sua volta, capace di collaborare in compiti via via sempre più complessi. Scriveva Tim Berners-Lee, l'inventore del Web, ancora nel 2001:

> *Ho fatto un sogno riguardante il Web… ed è un sogno diviso in due parti. Nella prima parte, il Web diventa un mezzo di gran lunga più potente per favorire la collaborazione tra i popoli. Ho sempre immaginato lo spazio dell'informazione come una cosa a cui tutti abbiano accesso immediato e intuitivo, non solo per navigare ma anche per creare. [...] Inoltre, il sogno della comunicazione diretta attraverso il sapere condiviso dev'essere possibile per gruppi di qualsiasi dimensione, gruppi che potranno interagire elettronicamente con la medesima facilità che facendolo di persona. Nella seconda parte del sogno, la collaborazione si allarga ai computer. Le macchine diventano capaci di analizzare tutti i dati sul Web, il contenuto, i link e le transazioni tra persone e computer. La "Rete Semantica" che dovrebbe renderlo possibile deve ancora nascere, ma quando l'avremo i meccanismi quotidiani di commercio, burocrazia e vita saranno gestiti da macchine che parleranno a macchine, lasciando che gli uomini pensino soltanto a fornire l'ispirazione e l'intuito. Finalmente, si materializzeranno quegli "agenti" intelligenti sognati per decenni. Questo Web comprensibile alle macchine si concretizzerà introducendo una serie di progressi tecnici e di adeguamenti sociali attualmente in fase di sviluppo.*[29]

[29] Tim Berners-Lee, L'architettura del nuovo Web, Feltrinelli, 2001.

L'intelligenza ambientale

L'evoluzione della rete procede in diverse direzioni. Fra quelle più importanti, per chi si occupa delle problematiche della Human Computer Interaction, vi è la tendenza verso la cosiddetta "intelligenza ambientale" (*ambient intelligence*). Con questo termine ci si riferisce alla progettazione di ambienti sensibili alla presenza delle persone, e che possono interagire con queste in vari modi. In sostanza, uno spazio popolato di oggetti intelligenti e fra loro interconnessi, che offrono agli esseri umani funzionalità utili per comunicare, controllare l'ambiente e accedere all'informazione.

Si tratta di una visione del futuro dell'elettronica di consumo, delle telecomunicazioni e dell'informatica sviluppata dalla fine degli anni '90. Secondo questa visione, il mondo si popolerà di dispositivi che interagiscono fra loro e cooperano per supportare le persone nelle loro attività quotidiane. Questi dispositivi sono dotati d'intelligenza e possono accedere a dati e informazioni disponibili nella rete, alla quale sono sempre connessi. Via via che questi dispositivi diventano più piccoli e più integrati nell'ambiente fisico, essi scompaiono dalla nostra vista, e ciò che rimane percepibile è soltanto l'interfaccia d'uso. Come scriveva Donald Norman nel suo libro *Il computer invisibile* (1998):

> *[...] una generazione di tecnologie personali in cui la tecnologia scompare nello strumento, attivando valide funzioni ma senza essere visibile. La generazione in cui il computer scompare all'interno di strumenti specializzati a seconda dell'attività. La generazione del computer invisibile.*[30]

Il paradigma dell'intelligenza ambientale si fonda su tecnologie che sono:

- *embedded*: i dispositivi sono fra loro interconnessi e integrati nell'ambiente;
- *context aware*: i dispositivi sono in grado di percepire informazioni provenienti dall'ambiente in cui si trovano, e di interpretarle in base al contesto;
- *personalizzate*: i dispositivi possono essere configurati in relazione alle specifiche necessità degli utenti;
- *adattive*: i dispositivi sono in grado di apprendere durante il loro uso, e modificare di conseguenza il loro comportamento;
- *anticipatorie*: i dispositivi possono anticipare i desideri e le necessità dell'utente.

Gli scenari d'uso che possono essere immaginati sono molto diversi. Riportiamo, come esempio, lo scenario riportato da Wikipedia alla voce "Ambient intelligence":[31]

> *Ellen rientra a casa dopo una lunga giornata di lavoro. Alla porta d'ingresso viene riconosciuta dalla telecamera intelligente di sorveglianza, che disattiva l'allarme e apre la porta. Entrata in casa, la mappa della famiglia indica che suo marito Peter si trova a una fiera d'arte a Parigi, e che la loro figlia Charlotte è nella sua stanza, a giocare con uno schermo interattivo. Al servizio di sorveglianza remota per i bambini viene notificato che Ellen è a casa, quindi la connessione online viene disattivata. Quando entra in cucina, il quadro dei messaggi si accende per segnalare che ci sono nuovi messaggi. La lista della spesa che era stata predisposta richiede una conferma per essere inviata al supermarket per gli acquisti. C'è anche un messaggio che la avverte che il sistema di casa ha trovato nuove informazioni nel Web semantico su delle villette economiche con vista mare per le vacanze in Spagna. Ellen si connette brevemente con la stanza di Charlotte per salutarla, e la sua immagine video compare automaticamente sullo schermo piatto che Charlotte sta utilizzando. Quindi si connette con Peter alla fiera d'arte di Parigi. Egli le mostra, con la fotocamera connessa alle sue lenti a contatto, alcune sculture che vorrebbe comprare, ed Ellen approva la scelta. Nel frattempo seleziona uno dei menu visualizzati, che le indica che cosa può preparare con i cibi presenti in dispensa e nel frigorifero. Poi accende il televisore sul canale on-demand per vedere il programma con le ultime notizie. Dopo avere dato il comando "Seguimi", si sposta in camera da letto. Il programma viene allora visualizzato automaticamente sul monitor piatto in camera da letto, dove va per*

[30] Op.cit., nella edizione italiana di Apogeo, 2000, pag. 271.
[31] Tratto da http://en.wikipedia.org/wiki/Ambient_intelligence , 16 gennaio 2010 (nostra traduzione dall'inglese).

fare della ginnastica personalizzata. Più tardi, dopo il rientro di Peter, chiacchierano con un amico in soggiorno, con l'illuminazione personalizzata. Guardano il presentatore virtuale che li informa sui programmi e sulle informazioni che sono state memorizzate, nella giornata, nel loro server di casa.

Chi si occupa di tecnologia, e ne esplora le possibilità e le innovazioni, subisce spesso il fascino di questi scenari, senza soffermarsi a riflettere in modo adeguato sulle loro implicazioni. Chi si occupa di human-computer interaction, e in particolar modo il progettista dei sistemi interattivi, ha tuttavia il dovere di interrogarsi costantemente sulla desiderabilità dei prodotti che propone o progetta. Questa va considerata non esclusivamente dal punto di vista dell'utente del prodotto, ma anche dal punto di vista complessivo della società di cui questo utente fa parte. Chi scrive ritiene che la comunità della human computer interaction stia ancora affrontando in modo troppo limitato queste problematiche. Nonostante il dichiarato interesse per il miglioramento dell'ambiente e per i valori della sostenibilità, è innegabile che, oggi, la comunità internazionale della HCI sia ancora in prevalenza, assorbita da problematiche che riguardano il futuro di una piccola parte dell'umanità. Adottando un punto di vista più globale, la prospettiva è molto diversa, e ciò permette anche di riflettere meglio sui valori impliciti nelle diverse possibili visioni del futuro, e sulle conseguenti priorità, come, per esempio, tra i vari ambiti della ricerca svolta dalle Università.

Secondo dati pubblicati nel 2008 dalla Banca Mondiale, nel 2005, su circa 6,5 miliardi di abitanti del pianeta, l'80% vive con meno di 10 dollari al giorno; il 49% con meno di 2,5 dollari al giorno e quasi il 14% con meno di un dollaro al giorno. La situazione all'uscita di questo libro, un quinquennio dopo, com'è noto non è migliorata, e la "distanza" fra ricchi e poveri è in continuo aumento. In questo contesto, scenari come quello riportato qui sopra generano dubbi e riflessioni. Fra gli infiniti possibili scenari d'uso di una stessa tecnologia, a quali dobbiamo assegnare le nostre priorità? Quali dobbiamo proporre al mercato, che ne sancirà, in ultima analisi, il successo o il fallimento? La responsabilità di queste scelte è di vasta portata perché, come vedremo nei prossimi capitoli, ogni tecnologia interagisce in modo profondo con i suoi utilizzatori, e ne cambia i comportamenti. I paradigmi d'interazione con le tecnologie di oggi sono, come si comprende facilmente, strettamente legati ai modelli dei nostri comportamenti quotidiani, nel lavoro e nel tempo libero.

Ripasso ed esercizi

1. Discuti i rapporti fra tecnologie e paradigmi d'interazione.
2. Quali sono le principali differenze fra la modalità di interazione mediante una teletype e un terminale tradizionale?
3. Che cosa s'intende per paradigma di manipolazione diretta?
4. Che cosa si intende con la sigla WIMP?
5. Che cosa caratterizza il paradigma "point & clic"?
6. Descrivi le differenze fra il nomadic computing e il mobile computing.
7. Quali sono, secondo la tua esperienza, le caratteristiche che accomunano le social application?
8. Che cosa si intende per intelligenza ambientale?

Approfondimenti e ricerche

1. La filosofia del progetto del primo sistema basato sulla metafora della scrivania, realizzato presso i laboratori dello Xerox PARC alla fine degli anni '70, è riassunta nel classico articolo di Smith, Irby, Kimball, Verplank, e Harslem, *Designing the Star User Interface*, pubblicato sulla rivista *Byte* nel 1982, e reperibile in rete in numerosi siti. Leggi questo articolo e riassumi le principali analogie e differenze fra la filosofia di questo sistema e quella del sistema desktop da te usato.
2. Un interessante articolo sulla storia del mouse si trova in http://weburbanist.com/2009/04/05/evolution-of-the-mouse-classic-to-cutting-edge/
3. Una interessante storia dell'ipertesto, dalle origini del concetto al Web, si trova in http://www2.polito.it/didattica/polymath/ICT/Htmls/Argomenti/Appunti/StoriaIpertesto/StoriaIpertesto.htm.
4. Leggi il classico articolo sul Dynabook: A. Kay, A. Goldberg, *Personal Dynamic Media*, in *Computer*, vol. 10, no. 3, pp. 31-41, Mar. 1977, anche disponibile in rete in http://www.newmediareader.com/book_samples/nmr-26-kay.pdf. Confronta l'idea del Dynabook con le tecnologie disponibili oggi: che rapporto ha il personal

dynamic medium concepito da Alan Kay con le internet appliance disponibili oggi? Suggerimento: consulta l'interessante lavoro di J.W.Maxwell, *Tracing the Dynabook: As a Study of Technocultural Transformations*, Tesi di PhD, University of British Columbia, Novembre 2006, reperibile in rete in http://thinkubator.ccsp.sfu.ca/Dynabook/Maxwell-DynabookFinal.pdf, e in particolare il capitolo *The Dynabook today: how far have we come*. Questa tesi è stata pubblicata nel 2006: negli anni trascorsi da allora, la situazione è cambiata?

5. Il paradigma della manipolazione diretta è ben lontano dall'avere mostrato tutte le sue potenzialità. La rappresentazione in tre dimensioni degli oggetti manipolati e gli schermi multi-touch offrono possibilità ancora in buona parte da esplorare. Esplora la rete alla ricerca di video dimostrativi interessanti su questo tema. A puro titolo di esempio, puoi iniziare su YouTube, con video sul sistema BumpTop, su applicazioni di Microsoft Surface, sull'interfaccia multi-touch a GoogleMap.
6. L'articolo di S.Sanna, *Mobile computing*, in A.Soro (ed.), *Human Computer Interaction – Fondamenti e prospettive*, ed.Polimetrica, 2009, pagg.253-288 (disponibile anche in rete) contiene un'interessante rassegna sulle problematiche dell'interfaccia utente dei dispositivi per il mobile computing.
7. Un'autrice che, da diversi anni, ha studiato i siti di social network è Danah Boyd. Tra i numerosi saggi da lei scritti (reperibili in http://www.danah.org/papers), puoi approfondire queste tematiche, per esempio, in *Social Network Sites: Definition, History and Scholarship*, scritto con N.Ellison (2007), o in *Why Youth (Hearth) Social Network Sites: The Role of Networked Publics in Teenage Social Life* (2007), in cui l'autrice analizza la nozione di spazi pubblici mediati dalla tecnologia.

3. Usabilità

Sintesi del capitolo

Questo capitolo ha lo scopo di introdurre i concetti base che saranno approfonditi nel corso di tutto il libro. In particolare, sarà precisato il concetto intuitivo di "facilità d'uso" e saranno discussi i principali ostacoli all'usabilità degli oggetti interattivi. Per questo sarà introdotto il modello di Norman sui sette stadi dell'azione, e i concetti di affordance e di feedback. Questi concetti sono chiariti attraverso semplici esempi. Viene quindi discussa la nozione di usabilità proposta dall'ISO 9241 e introdotti i concetti di apprendibiltà e di memorabilità di un prodotto. Sono infine introdotte le nozioni di accessibilità e di usabilità universale.

Un modello dell'interazione

Viviamo quotidianamente le difficoltà nel rapporto con gli oggetti che ci circondano, che percepiamo spesso come difficili da usare. Queste difficoltà non riguardano necessariamente gli oggetti di alta tecnologia; possiamo incontrare dei problemi anche nell'uso di dispositivi relativamente semplici: un fornello, una doccia, l'interruttore della luce. Quali sono le radici di queste difficoltà? La risposta a questa domanda è ovviamente molto importante: se possiamo individuare le cause profonde delle nostre difficoltà, possiamo studiare il modo di rimuoverle. È quindi utile analizzare il modo con cui noi interagiamo con gli oggetti, per individuare dove nascono le difficoltà nel loro uso, e perché.

Il modello più semplice dell'interazione fra un sistema e il suo utilizzatore è rappresentato dal *ciclo di feedback* (*feedback loop*), rappresentato in Figura 42. L'utente, per raggiungere il proprio scopo, fornisce un input al sistema, e riceve da questo una risposta (*feedback*), che viene interpretata e confrontata con lo scopo iniziale. Il risultato di questo confronto porta alla successiva azione dell'utente, innescando così un nuovo ciclo di stimolo-risposta. Le frecce della figura rappresentano pertanto l'informazione che fluisce da un interlocutore all'altro durante l'interazione. Questa informazione, nei sistemi informatici, è molto spesso di natura testuale (nei due sensi) o grafica (dal sistema all'utilizzatore). Può tuttavia essere di natura diversa: gestuale (per esempio, quando si usa il mouse come dispositivo di input), vocale, eccetera.

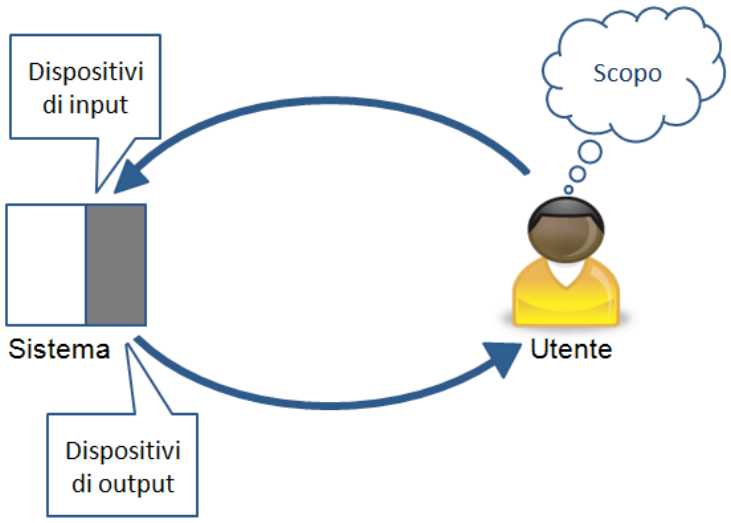


Figura 42. Interazione utente-sistema come ciclo di feedback

Il sistema rappresentato in Figura 42 può essere di natura qualsiasi: un computer, un telefono cellulare, un prodotto software, o anche, semplicemente, un oggetto non intelligente - un interruttore della luce, la leva del cambio di un'automobile, il rubinetto della doccia. Ciò che importa è che, ricevendo un certo stimolo, esso reagisca producendo qualche tipo di risposta. Così, per esempio, ruotando il rubinetto della doccia, si ottiene un flusso d'acqua di una certa intensità. Anche se questo libro si occupa della progettazione di quei sistemi (spesso con un elevato contenuto di software) in grado di fornire risposte complesse agli stimoli prodotti da utilizzatori umani, i concetti trattati sono del tutto generali, e il lettore potrà riferirli a sistemi di qualsiasi natura.

Il modello di Figura 42, nella sua semplicità, non permette di comprendere l'origine delle difficoltà che sperimentiamo nell'interazione con i sistemi. Perché alcuni sistemi ci paiono difficili da usare? Per analizzare meglio quest'aspetto è molto utile un modello più articolato – e tuttavia ancora molto semplificato - proposto da Donald Norman nel 1986[32] e rappresentato in Figura 43.

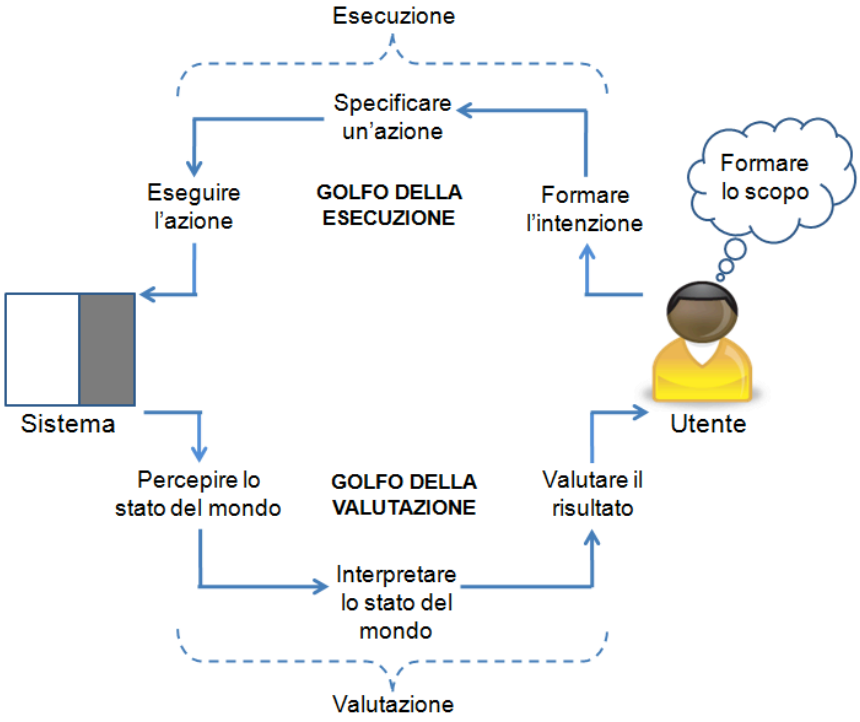


Figura 43. Il modello di Norman

Questo modello scompone il nostro operare sugli oggetti in sette passi (o stadi) principali:

1. Formare lo scopo: decidiamo quale scopo vogliamo raggiungere

Esecuzione (la fase in cui pianifichiamo ed effettuiamo le azioni sul sistema):

2. Formare l'intenzione: decidiamo che cosa intendiamo fare per raggiungere lo scopo prefissato
3. Specificare un'azione: pianifichiamo nel dettaglio le azioni specifiche da compiere

[32] Donald A.Norman è considerato uno dei padri della moderna psicologia cognitiva, e si è occupato di ergonomia e di design, con particolare riferimento al mondo della tecnologia, nel cui ambito ha scritto numerosi libri. Il modello di Norman è descritto in dettaglio nel suo libro *The Psychology of Everyday Things*, Basic Books, 1988, tradotto in italiano col titolo *La caffettiera del masochista – Psicopatologia degli oggetti quotidiani*, Gruppo Editoriale Giunti, 1990 e successive riedizioni. Si tratta di un libro storicamente molto importante, di facile e gradevole lettura, ricco di esempi tratti dalla vita quotidiana, che si consiglia di leggere a integrazione di questi capitoli.

4. Eseguire l'azione: eseguiamo effettivamente le azioni pianificate

Valutazione (la fase in cui confrontiamo quello che è successo con lo scopo che volevamo raggiungere):

5. Percepire lo stato del mondo: osserviamo come sono cambiati il sistema e il mondo circostante dopo le nostre azioni
6. Interpretare lo stato del mondo: elaboriamo ciò che abbiamo osservato, per dargli un senso
7. Valutare il risultato: decidiamo se lo scopo iniziale è stato raggiunto.

Come esempio, consideriamo un sistema costituito da una doccia, controllabile attraverso la manopola rappresentata in Figura 44, e scomponiamo secondo il modello di Norman i passi necessari per aprire il getto d'acqua.



Figura 44. La manopola del rubinetto della doccia dell'esempio

1.	Formare lo scopo:	desidero aprire il getto d'acqua per fare la doccia;
2.	Formare l'intenzione:	a questo scopo, intendo operare sul rubinetto in figura…
3.	Specificare un'azione:	… ruotandolo con la mano destra verso sinistra, fino in fondo;
4.	Eseguire l'azione:	eseguo quanto sopra;
5.	Percepire lo stato del mondo:	sento che il rubinetto non può ruotare ulteriormente verso sinistra, e vedo un consistente flusso di acqua uscire dalla doccia; sento che l'acqua è calda;
6.	Interpretare lo stato del mondo:	comprendo che il rubinetto è arrivato a fine corsa, e che il flusso dell'acqua calda è conseguenza della mia azione sul rubinetto;
7.	Valutare il risultato:	stabilisco che ho raggiunto lo scopo che mi ero prefisso.

Questo modello, nella sua semplicità, può essere applicato a qualsiasi tipo di azione. Ovviamente, azioni complesse dovranno essere decomposte in azioni sufficientemente semplici, ciascuna delle quali comporterà il passaggio attraverso i sette stadi. Per esempio, la stesura di una lettera utilizzando un word processor dovrà essere decomposta in numerosi operazioni più semplici, quali: aprire un documento vuoto, scrivere l'intestazione, scrivere il corpo della lettera, stampare il file, e così via, arrivando fino al livello di "granularità" più opportuno. Chiarisce infatti Norman, nel libro citato (pag.67):

> *I sette stadi costituiscono un modello approssimativo, non una teoria psicologica completa. In particolare, gli stadi quasi certamente non sono entità separate e distinte. La maggior parte dei comportamenti non richiede*

che si ripassino tutti gli stadi nell'ordine, e nella maggior parte delle attività un'azione singola non basta. Devono esserci numerose sequenze e l'intera attività può durare ore o anche giorni. C'è un continuo anello di retroazione, in cui i risultati di un'attività sono usati per indirizzarne altre, in cui gli scopi conducono a scopi collaterali e sussidiari, le intenzioni a sub-intenzioni. Ci sono attività in cui gli scopi vengono dimenticati, scartati o riformulati.

Ciò che interessa, in questo contesto, è il fatto che il modello permette di individuare con grande chiarezza i momenti in cui possono presentarsi dei problemi. Nel percorrere i sette stadi dell'azione, infatti, è possibile che s'incontrino delle difficoltà nel passare da uno stadio all'altro o, come dice Norman, nell'attraversare i *golfi* che li separano. In particolare, ci sono due golfi che possono essere particolarmente difficili da superare (Figura 43):

- il *golfo della esecuzione*, che separa lo stadio delle intenzioni da quello delle azioni, e
- il *golfo della valutazione*, che separa lo stadio della percezione dello stato del mondo da quello della valutazione dei risultati.

In altre parole, il golfo dell'esecuzione è quello che separa le intenzioni dalle azioni che permettono di realizzarle: per superarlo, dovrò identificare, fra le azioni che è possibile eseguire con il sistema, quelle che mi permetteranno di raggiungere lo scopo. Nel caso dell'esempio, non ci sono state difficoltà: il rubinetto è facilmente riconoscibile dal bollino rosso che lo identifica come rubinetto dell'acqua calda, e il suo comportamento è identico a tutti gli altri rubinetti – per aprire l'acqua occorre ruotarlo in senso antiorario. Il golfo dell'esecuzione, perciò, è facile da attraversare. Nel caso, invece, in cui mancasse il bollino rosso, l'utente sarebbe costretto a effettuare varie prove per identificare il rubinetto giusto, e il golfo dell'esecuzione sarebbe, nella metafora di Norman, più difficile da attraversare.

Il golfo della valutazione, invece, è legato alle difficoltà che l'utente deve superare per interpretare lo stato fisico del sistema dopo le azioni effettuate, e comprendere se ha raggiunto o meno lo scopo prefisso. A questo proposito, nel nostro esempio, si possono verificare varie situazioni. Che cosa pensare se, per esempio, il flusso d'acqua iniziale fosse freddo e restasse tale per parecchi secondi? L'utente non sarebbe in grado di valutare immediatamente se le sue azioni hanno raggiunto lo scopo desiderato, e dovrebbe attendere per un certo periodo - a priori non determinato - con il dubbio che lo scaldabagno sia spento. Il golfo della valutazione sarebbe, in questo caso, più difficile da attraversare. La situazione sarebbe ancora peggiore se il rubinetto dell'acqua fredda e dell'acqua calda non fossero fra loro distinguibili (attraverso il bollino colorato visibile in Figura 44). In questo caso, come tutti noi abbiamo sperimentato almeno una volta, dovremmo procedere per tentativi per identificare il rubinetto giusto, e il semplice compito di aprire il flusso d'acqua calda potrebbe richiedere anche diversi minuti.

Come secondo esempio, consideriamo un normale fornello a gas da cucina (Figura 45). Nella versione di sinistra (a), l'associazione fra manopole e piastre è evidente: la disposizione fisica delle manopole segue da vicino quella delle piastre, ed è naturale presupporre (com'è nella realtà) che, per esempio, la manopola più a sinistra controlli la piastra situato nell'angolo inferiore sinistro. In questo caso, il golfo dell'esecuzione è facile da superare: è immediato identificare la manopola che governa l'erogazione di gas di una particolare piastra. Anche se l'apparato è molto simile, la situazione del fornello di Figura 45b è molto diversa, dal punto di vista della facilità d'uso. Qui l'associazione manopola/piastra non è ovvia, perché la disposizione fisica delle manopole non fornisce alcun indizio. Per superare il golfo dell'esecuzione l'utente dovrà fare delle ipotesi, che potrebbero risultare errate. Certamente, la probabilità di operare sulla manopola sbagliata sarà ora molto maggiore.

In entrambi i fornelli, invece, il golfo della valutazione è molto facile da superare: l'utente può verificare immediatamente il risultato della sua azione, osservando l'accensione della fiamma sulla piastra scelta. Se invece il fornello fosse dotato di piastre elettriche, il golfo della valutazione sarebbe indubbiamente più ampio. A meno che l'accensione non sia segnalata da una spia luminosa, l'utente avrebbe qualche problema nel riconoscere l'avvenuta accensione della piastra desiderata. Dovrebbe, tipicamente, toccarla con un dito per verificarne il calore, eventualmente aspettando qualche secondo in attesa che il riscaldamento risulti sensibile.

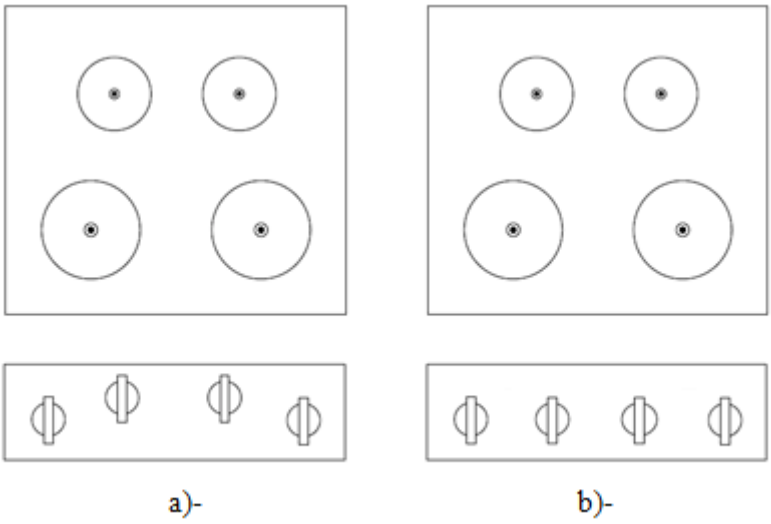


Figura 45. Due esempi di fornelli: sono simili, ma la facilità d'uso è molto differente

La situazione peggiore si avrebbe nel caso in cui, in un fornello elettrico senza spie luminose, le singole manopole non fossero facilmente associabili ai vari fuochi, come nell'esempio di Figura 46, in cui alle manopole dei fuochi sono accostate le manopole del forno. In questo caso, entrambi i golfi (esecuzione e valutazione) sono piuttosto ampi.



Figura 46. Un fornello difficile da usare

L'esempio dei fornelli ci mostra come, spesso, la facilità d'uso sia determinata, o seriamente compromessa, da elementi di modesta entità. Fra la soluzione a) e la soluzione b) di Figura 45 le differenze sono minime, e probabilmente del tutto irrilevanti sia dal punto di vista tecnico sia da quello dei costi di produzione. Si tratta, in definitiva, di un modesto disallineamento delle due manopole centrali: un dettaglio nell'economia generale del progetto. Ma è questo dettaglio che, come in molti altri casi, fa la differenza. Come rileva continuamente chiunque conduca prove d'uso con gli utenti, questi spesso s'inceppano su piccoli particolari: basta spostare un campo di input di qualche millimetro sullo schermo, cambiare il colore di un pulsante, riformulare un messaggio di errore, eliminare un acronimo o spostare una virgola in un testo, e l'ampiezza di uno dei due golfi viene modificata in modo sostanziale. Nella progettazione dei sistemi, come nell'arte, la qualità del risultato finale dipende non soltanto dalla concezione complessivo del prodotto, ma dall'interazione d'innumerevoli piccoli dettagli.

Affordance e feedback

Con il termine, intraducibile in italiano, di *affordance*, si denota la proprietà di un oggetto di influenzare, attraverso la sua apparenza visiva, il modo in cui viene usato. Si tratta di un concetto molto importante, introdotto nel 1966 dallo psicologo statunitense James J.Gibson, studioso della percezione, e poi ripreso da Donald Norman nell'ambito dell'interazione uomo macchina.

Un oggetto che possiede una buona affordance "invita" chi lo guarda a utilizzarlo nel modo corretto, cioè nel modo per cui è stato concepito. Per esempio, l'aspetto di una maniglia ben progettata dovrebbe far intuire immediatamente come la porta vada aperta: se tirandola, spingendola, o facendola scorrere. Per esempio, fra i due maniglioni antipanico di Figura 47, quello di sinistra ha un'ottima affordance, perché invita chiaramente a "spingere", mentre l'affordance di quello di destra è più ambigua: devo spingere o tirare? Così, una porta che si apre automaticamente al passaggio ha una scarsa affordance, poiché non fornisce indizi sul suo funzionamento.

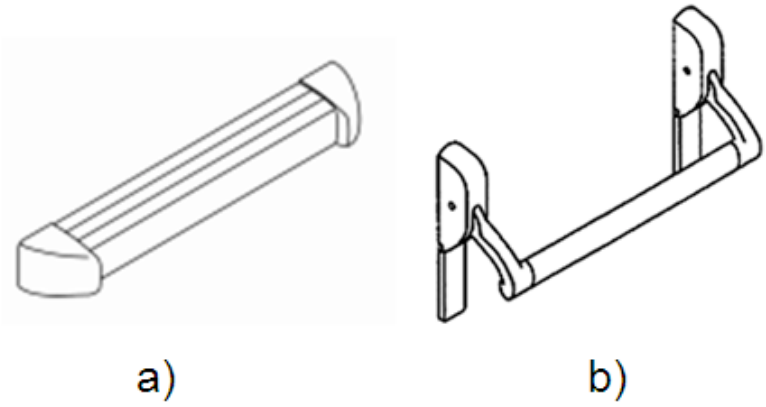


Figura 47. Due maniglioni antipanico con affordance diversa

Anche l'affordance della maniglia di Figura 48a lascia molto a desiderare: la sua forma non invita a ruotarla – come si dovrebbe fare per aprire o chiudere la finestra – ma a usarla per appendervi degli oggetti, come mostrato in Figura 48b.

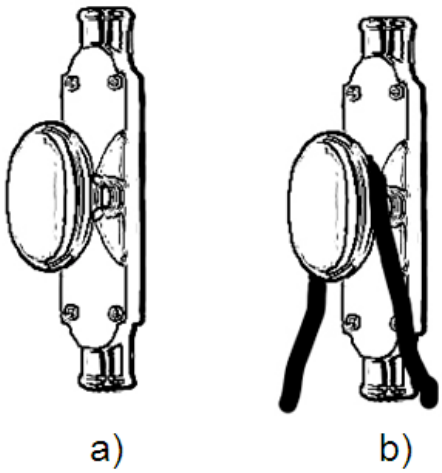


Figura 48. La maniglia di una finestra con cattiva affordance

Gli oggetti di Figura 49 sono invece dotati di una buona affordance. La forma del Mighty Mouse wireless della Apple fornisce chiari indizi su come usarlo. I due tasti laterali invitano a stringerlo fra le dita, e la posizione della pallina per muovere il puntatore sullo schermo, da ruotarsi con il polpastrello del dito indice, suggerisce chiaramente il corretto orientamento dello strumento. La particolare impugnatura della forbice, con i due anelli di forma diversa, invita chiaramente a utilizzarla nel modo corretto.



Figura 49. Oggetti con buona affordance. A sinistra: Mighty Mouse della Apple

Gli oggetti ad alta tecnologia sono spesso privi di affordance. Confrontiamo i due telefoni di Figura 50: l'apparecchio tradizionale possiede chiaramente un'affordance molto migliore dello smartphone di destra (il modello N97 della Nokia). La cornetta ha un manico centrale che permette di tenerla saldamente in mano per avvicinarne la parte superiore all'orecchio. Così facendo, la parte inferiore, che contiene il microfono, si porta naturalmente in prossimità della bocca dell'utilizzatore. Né può esistere il dubbio su quale sia la parte superiore: il cavo che unisce la cornetta al corpo dell'apparecchio rende difficoltoso utilizzarla nel verso sbagliato. La forma della forcella, modellata perfettamente su quella della cornetta, invita chiaramente ad appoggiarvela sopra. Il disco combinatore può ruotare soltanto nel verso giusto, e i suoi fori si adattano bene alle dimensioni delle dita. La resistenza del disco consente di formare i numeri con uno sforzo uniforme e non eccessivo, fino al fermo di metallo che accoglie il dito a fine corsa. Non c'è dubbio che, nella sua forma tradizionale, il telefono costituiva un eccellente prodotto di design.



Figura 50. Quale telefono è dotato di migliore affordance?

Una buona affordance riduce quindi il golfo dell'esecuzione. Per ridurre l'ampiezza del golfo della valutazione, invece, gli oggetti dovranno fornire un feedback facilmente interpretabile, cioè un segnale che indichi chiaramente all'utente quali modifiche le sue azioni abbiano prodotto sullo stato del sistema. Per esempio, un bottone virtuale visualizzato su uno schermo dovrà mostrare chiaramente quando viene premuto, come nell'esempio di Figura 51. In questo caso, sia

l'affordance sia il feedback sono ottimi: il bottone invita chiaramente a premerlo, e mostra in modo evidente il risultato di quest'azione.



Figura 51. Un pulsante virtuale con buona affordance e buon feedback

Il feedback deve essere ben comprensibile e specifico: l'utente deve essere in grado di interpretarlo senza fatica. Meglio ancora, dovrebbe essere formulato nel modo che l'utente si aspetta. Importante è la sua tempestività: solo così l'utente lo può porre facilmente in relazione con l'azione cui si riferisce. Se la distanza temporale fra azione e feedback è significativa (Figura 52a), essi possono essere interpretati come eventi tra loro indipendenti: a volte bastano pochi secondi di ritardo per disaccoppiare, nella percezione dell'utente, i due eventi.

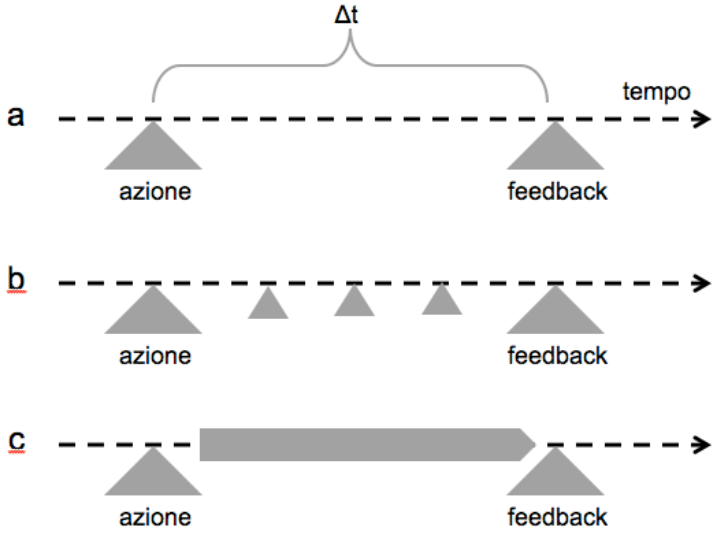


Figura 52. Azione e feedback: possibilità

In questi casi, è opportuno inserire dei feedback intermedi, che segnalino chiaramente all'utente il progredire dello stato del sistema verso lo stato finale desiderato. Questi feedback intermedi possono essere discreti (Figura 52b), oppure continui (Figura 52c), realizzati per esempio con delle animazioni. Non è però sufficiente segnalare che "qualcosa è in corso", mostrando semplicemente una figura animata sul video, come per esempio la barra ruotante di Figura 53a: l'utente non si accontenta di sapere che il processo è in corso, ma vuole sapere quanto dovrà ancora aspettare. Gli si dovrà quindi mostrare, ove possibile, una stima *quantitativa* del tempo mancante, come in Figura 53b o, meglio ancora, in Figura 53c.

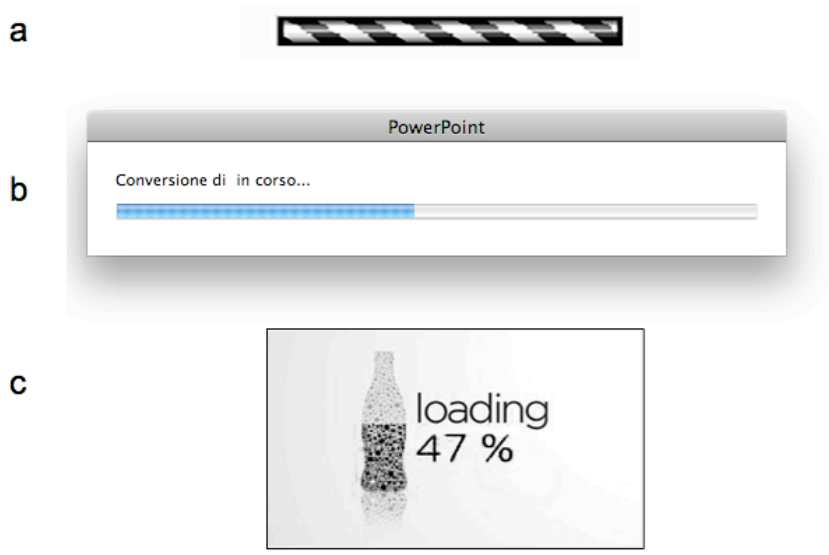


Figura 53. Feedback continuo realizzato con animazioni
(a: Mac OS 8; b: PowerPoint Mac:2008; c: http://ww.cocacola.it)

In conclusione, il compito del progettista è pertanto quello di progettare oggetti con buona affordance, per ridurre l'ampiezza del golfo della esecuzione, e con buon feedback, per ridurre l'ampiezza del golfo della valutazione.

La nozione di usabilità

La nozione di facilità d'uso, che abbiamo finora utilizzato informalmente senza mai definirla, ci sembra semplice e intuitiva ma, come abbiamo visto commentando il modello di Norman, è in realtà piuttosto articolata, perché si riferisce ai processi coinvolti nella nostra interazione con il mondo. È quindi ora di tentare di definirla nel modo più preciso possibile. A questo scopo, alla dizione corrente di "facilità d'uso" si preferisce usare il termine più specifico di *usabilità* (in inglese, *usability*), proprio per segnalare che intendiamo riferirci a un concetto definito in modo preciso.

Il modello di Norman spiega quando e perché nascono i problemi di usabilità, ma non offre una definizione del termine. Ciò che ci interessa è una definizione operativa, che permetta di *quantificare* l'usabilità, dandone, per quanto è possibile, una misura oggettiva. Le definizioni riportate in letteratura sono numerose, ma non sempre utili a questo scopo. Una definizione che fa al caso nostro è quella proposta nel già citato standard ISO 9241, non solo perché di fonte autorevole ma perché, nella sua semplicità, è ricca d'implicazioni di carattere pratico, e ci permette, come vedremo, di definire delle misure:

> *L'usabilità di un prodotto è il grado con cui esso può essere usato da specificati utenti per raggiungere specificati obiettivi con efficacia, efficienza e soddisfazione in uno specificato contesto d'uso.*[33]

Si tratta, per così dire, di una definizione multidimensionale, che scompone l'usabilità su tre assi, relativi a tre variabili sostanzialmente indipendenti: efficacia, efficienza e soddisfazione degli utenti, e il cui valore può in qualche modo essere "misurato" (Figura 54).

[33] Il testo in inglese recita: "the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use" (ISO 9241-11:1998).

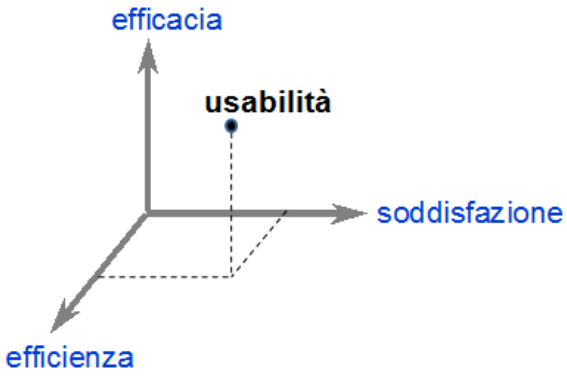


Figura 54. Le tre dimensioni dell'usabilità secondo la ISO 9241

- L'*efficacia* viene definita come *la accuratezza e completezza con cui gli utenti raggiungono specificati obbiettivi.* Essa considera pertanto il "livello di precisione" con cui l'utente riesce a raggiungere i suoi scopi, misurato in qualche modo numericamente.
- L'*efficienza* è definita come "la quantità di risorse spese in relazione all'accuratezza e alla completezza con cui gli utenti raggiungono gli obiettivi". Tali risorse potranno essere di natura differente secondo le situazioni, e potranno anch'esse essere quantificate. Per esempio: il tempo impiegato per ottenere un determinato risultato, il numero di tasti da premere per realizzare una certa funzione, il numero di operazioni di un certo tipo da effettuare, ecc.
- La *soddisfazione*, infine, è definita – in modo in effetti un po' contorto - come "la libertà dal disagio e l'attitudine positiva verso l'uso del prodotto".[34]

Applicando questa definizione, potremo "misurare" l'usabilità associandole tre grandezze numeriche, che ne quantifichino l'efficacia, l'efficienza e la soddisfazione dell'utente. Queste grandezze dovranno essere definite caso per caso, in funzione della natura dello specifico sistema. Per quanto riguarda la soddisfazione, la quantificazione sarà normalmente effettuata chiedendo agli utenti, attraverso opportuni questionari, di attribuire dei "voti" a specifiche caratteristiche del sistema (o al sistema nella sua totalità). Tutti i valori saranno ovviamente di tipo statistico, e verranno calcolati, per esempio, come media di un insieme significativo di misure.

Questa definizione di usabilità non è in alcun modo legata a caratteristiche specifiche dei prodotti: si tratta di una definizione del tutto generale, applicabile a qualsiasi manufatto, anche il più semplice. Per esempio, consideriamo ancora la manopola della doccia di Figura 44. Per misurarne l'usabilità, potremmo definire le seguenti metriche:

- efficacia: la capacità di regolazione precisa del flusso d'acqua, misurata sulla base dei litri aggiuntivi erogati al secondo per ogni giro completo della manopola, a partire dalla posizione di chiusura totale del flusso;
- efficienza: per esempio, una funzione del numero n di giri di manopola necessari per raggiungere il flusso massimo. In alternativa (o in aggiunta), potremmo considerare lo sforzo necessario per ruotare la manopola, utilizzando come misura il momento torcente. Nel primo caso, la manopola sarà considerata efficiente se permette di ottenere il flusso massimo in pochi giri; nel secondo caso, se lo sforzo richiesto per ruotarla è limitato;
- soddisfazione: gradimento soggettivo medio espresso da un campione di utenti, per esempio con un voto da 0 a 10.

Come si comprende anche da questo semplice esempio, l'usabilità non è una proprietà assoluta degli oggetti, ma è sempre *relativa* al compito da svolgere, all'utente che lo svolge e al contesto d'uso. Nel nostro esempio, se il compito d'interesse non fosse quello di chiudere/aprire completamente l'acqua, ma quello di regolare l'acqua al 20% della portata del rubinetto, le nostre misure sul campione indicherebbero un'efficacia molto inferiore, perché non avremmo alcun modo di conoscere, durante l'uso, la portata corrente. Per questo compito, la manopola si rivelerebbe quindi uno strumento assai poco usabile. Ancora, se l'utente non avesse alcuna familiarità con i rubinetti, e non sapesse quindi che,

[34] Nel testo in inglese: "freedom from discomfort, and positive attitudes to the use of the product".

di solito, il flusso dell'acqua viene chiuso con una rotazione in senso orario, ci sarebbe una percentuale significativa di rotazioni nel senso sbagliato, e quindi l'efficienza modesta. Infine, anche l'ambiente d'uso può influenzare drasticamente l'usabilità della manopola. Se, come caso estremo, il rubinetto fosse situato in una stanza diversa da quella in cui si trova la manopola che lo controlla, l'utente non potrebbe ricevere un feedback immediato dalle sue azioni sulla manopola. Sarebbe probabilmente costretto ad effettuare numerosi tentativi, peggiorando efficacia, efficienza e, molto probabilmente, soddisfazione. Lo stesso standard ISO 9241 sottolinea molto bene il fatto che l'usabilità è una nozione relativa:

> *Il termine usabilità è usato spesso per riferirsi alla capacità di un prodotto di essere usato con facilità. […] Comunque, gli attributi richiesti da un prodotto per essere usabile dipendono dalla natura dell'utente, del compito e dell'ambiente. Un prodotto non possiede alcuna usabilità intrinseca, ma solo la capacità di essere usato in un particolare contesto. L'usabilità non può essere valutata studiando un prodotto in isolamento.*

Secondo la definizione utilizzata, l'usabilità è una nozione intrinsecamente tri-dimensionale: efficacia, efficienza e soddisfazione derivante dall'uso. L'importanza relativa di queste tre grandezze andrà quindi valutata caso per caso, in funzione degli obbiettivi del sistema. Come considerare, per esempio, un prodotto che offre all'utente una maggiore gratificazione nell'uso ma gli richiede più risorse, rispetto a un sistema di uso più efficiente ma meno "divertente"? Consideriamo, per esempio, il meccanismo per controllare la sveglia sull'iPhone. L'ora della sveglia viene impostata ruotando i due dischi orari mostrati in Figura 55, facendo scorrere il dito sullo schermo tattile. L'operazione è piuttosto gratificante (si resta affascinati dalla perfetta simulazione del movimento dei dischi, che ruotano proprio come due dischi reali, accelerando e poi, più lentamente, fermandosi sul valore prescelto), ma richiede tempo. Un meccanismo tradizionale, in cui l'ora è impostata semplicemente digitandone il valore per mezzo di una tastiera, richiede un tempo inferiore di circa il 50%, ma è sicuramente molto meno divertente.[35] In questo caso, i progettisti della Apple hanno preferito la soluzione più insolita e divertente, tenendo conto dei particolari obiettivi di mercato dell'iPhone, e del fatto che l'impostazione della sveglia è operazione relativamente poco frequente. Per questo motivo, l'utente non viene troppo penalizzato dalla sostanziale inefficienza dell'operazione.

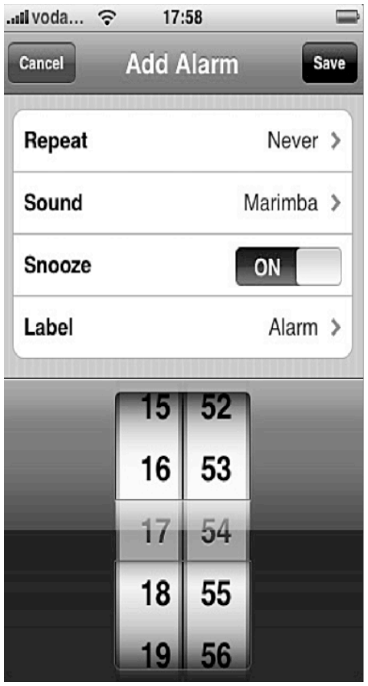


Figura 55. Impostazione della sveglia sull'iPhone Apple (2009)

[35] Il confronto con un cellulare Nokia è stato fatto da M. van Welie, in http://www.welie.com/thoughts .

Apprendibilità e memorabilità

La definizione di usabilità va ulteriormente approfondita. Infatti, occorre prendere in considerazione l'evoluzione che può subire l'utente nel tempo, nella sua relazione con il sistema. All'inizio, egli non lo conosce affatto (utente *novizio*), poi inizia ad usarlo (utente *principiante*), fino a diventare *competente* e, in qualche caso, *esperto* del prodotto (Figura 56).

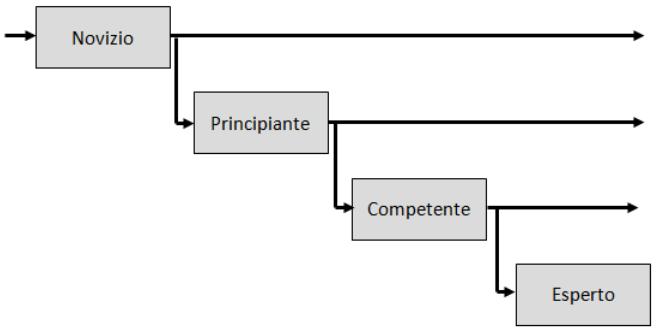


Figura 56. Evoluzione dell'utente nel rapporto con il sistema

In questo processo di apprendimento, l'utente può incontrare difficoltà più o meno grandi, a seconda delle caratteristiche del sistema. Prodotti anche molto simili per quanto riguarda le funzioni offerte possono infatti avere *profili di apprendimento* molto diversi. La Figura 57 mostra due prodotti simili (che qui non interessa individuare) con profili diversi. Il prodotto A ha, per così dire, una bassa soglia di apprendimento: come si vede dal grafico, all'utente è sufficiente un tempo piuttosto breve per ottenere una buona usabilità con il prodotto. In altre parole, l'utente principiante è in grado di imparare in poco tempo a svolgere i compiti che gli interessano con buona efficacia, efficienza e soddisfazione. Il prodotto B ha un profilo diverso: richiede un addestramento molto più lungo ma, in seguito, ripaga ampiamente l'utente del suo investimento iniziale, permettendogli di raggiungere, a regime, un'usabilità molto più elevata. Un sistema che sia facile da imparare si dice dotato di elevata *apprendibilità* (*learnability*).

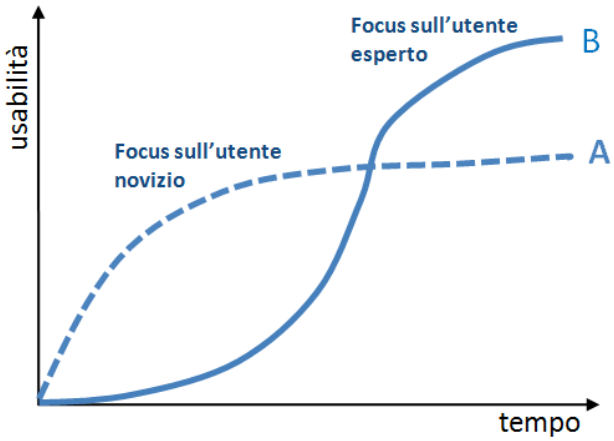


Figura 57. Profili di apprendimento

Nella progettazione di un sistema, il progettista ha di fronte a sé diverse scelte possibili:

- considerare come principali destinatari del prodotto gli *utenti occasionali,* cioè coloro che non hanno la necessità di utilizzarlo frequentemente, e quindi non sono disposti a investire una cospicua quantità del loro tempo in attività di apprendimento, oppure

- progettare in primo luogo per gli *utenti continuativi*, cioè per coloro che lo utilizzeranno in modo frequente e continuativo, e pertanto saranno disposti a investire anche una significativa quantità di tempo per imparare ad utilizzarlo con la massima efficacia ed efficienza.

I risultati della progettazione, nei due casi, saranno prodotti molto differenti, destinati a due fasce di mercato sostanzialmente diverse. Una terza possibilità è quella di indirizzare il prodotto a entrambi i tipi di utente, progettandolo in modo che possa fornire entrambi i profili di apprendimento esemplificati nella Figura 57. In altre parole, il prodotto offrirà funzioni di rapido apprendimento (profilo A) e funzioni di più lento apprendimento, ma che permettano di ottenere gli stessi risultati con maggiore efficienza o efficacia (profilo B). L'utente potrà così apprendere molto rapidamente a eseguire i compiti di base, e imparare, in un tempo più lungo, a eseguire funzioni complesse in modo sempre più efficiente ed efficace. Si pensi, per fare un esempio, ai molteplici tasti funzione di Photoshop: sono difficili da ricordare, ma permettono, a chi li conosca e abbia acquisito la necessaria manualità, un'altissima efficienza operativa. Oppure, ancora in Photoshop, alla possibilità di definire delle macro personalizzate che svolgano in modo automatico le sequenze di operazioni corrispondenti ai compiti più ricorrenti. Il loro uso richiede certamente un addestramento significativo, ma i risultati possono comportare grossi guadagni di efficienza.

Nel caso degli utenti occasionali, è utile che le modalità d'uso del prodotto siano facili da ricordare o, come si dice, che il prodotto sia dotato di un'elevata *memorizzabilità* (*memorability*). In caso contrario, a ogni nuovo utilizzo l'utente dovrà, per così dire, ricominciare da capo, e riapprendere modalità d'uso dimenticate. Questa caratteristica è particolarmente importante per i prodotti destinati agli anziani, nei quali le capacità di memorizzazione sono spesso indebolite, e per i prodotti che, pur destinati a un uso poco frequente, siano critici.

Un esempio tipico è costituito dai sistemi domestici anti-intrusione. Le segnalazioni di allarme sono eventi rari – potrebbero non verificarsi mai – ma, quando si verificano, l'utente deve essere in grado di intervenire immediatamente e senza fare errori: per esempio, per richiedere al sistema la causa dell'allarme, o per disattivare la sirena, e così via. In questo caso la memorabilità è essenziale: non è pensabile che l'utente sia costretto a ricorrere, in tali circostanze, al manuale d'istruzioni. Potrebbe non essere a portata di mano oppure, più semplicemente, potrebbe non essercene il tempo. L'utente dovrà necessariamente ricordare, per giunta in condizioni di stress, comandi appresi anche molto tempo prima e probabilmente mai utilizzati fuori dall'addestramento.

Apprendibilità e memorabilità sono così importanti che diversi autori li considerano componenti primari da incorporare nella stessa definizione di usabilità. Per esempio, Jakob Nielsen[36] definisce l'usabilità come la somma dei cinque attributi seguenti:

- *Apprendibilità*
  Il sistema dovrebbe essere facile da imparare, in modo che l'utente possa rapidamente iniziare a ottenere qualche risultato dal sistema;
- *Efficienza*
  Il sistema dovrebbe essere efficiente da usare, in modo che, quando l'utente ha imparato a usarlo, sia possibile un alto livello di produttività;
- *Memorabilità*
  Il sistema dovrebbe essere facile da ricordare, in modo che l'utente occasionale sia in grado di ritornare al sistema dopo un periodo di non utilizzo, senza dover imparare tutto di nuovo;
- *Errori*
  Il sistema dovrebbe rendere difficile sbagliare, in modo che gli utenti facciano pochi errori durante l'uso e in modo che, se ne fanno, possano facilmente recuperare. Inoltre, non devono avvenire errori catastrofici.
- *Soddisfazione*
  Il sistema dovrebbe essere piacevole da usare, in modo che gli utenti siano soggettivamente soddisfatti quando lo usano.

[36] Jakob Nielsen, *Usability Engineering*, Academic Press, 1993, pag.26.

Una definizione di usabilità così articolata, pur mettendo in evidenza aspetti molto importanti, non è strettamente necessaria. Infatti, la definizione data nell'ISO 9241, che si focalizza sugli *effetti* dell'usabilità in termini di efficacia, efficienza e soddisfazione, in relazione a uno specifico contesto di utilizzo e a uno specifico utente, è in grado di tenere in considerazione anche gli effetti di apprendibilità o memorabilità insoddisfacenti. Se per esempio un certo utente non fosse in grado di ricordare determinati comandi, in una data situazione d'uso, questo comporterebbe necessariamente una riduzione dell'usabilità secondo l'ISO 9241: efficacia, efficienza e soddisfazione nell'uso del prodotto, infatti, ne risentirebbero.

Sussidi all'utente

Un sistema interattivo è normalmente corredato da una serie di *sussidi*, che permettono ai suoi utenti di utilizzarlo agevolmente. Alcuni possono essere integrati nel prodotto stesso, come i sistemi di *help online*, altri possono essere forniti a parte, come i *manuali utente*, altri ancora sono costituiti da servizi, forniti dal produttore, come gli *help desk* erogati attraverso call center, o da altri utenti, che volontariamente offrono il loro aiuto partecipando a comunità in rete variamente organizzate (mediante *newsgroup, forum, chat*). Tutti questi sussidi contribuiscono a determinare l'usabilità complessiva del sistema; è opportuno quindi discuterne brevemente.

La Figura 58 riassume i sussidi più comuni, in uno schema "a strati". Essi sono di natura più o meno tangibile: gli strati interni sono costituiti da oggetti fisici (anche se in formato elettronico o costituiti da componenti software), mentre quello esterno è composto da servizi erogati attraverso sistemi di comunicazione, tipicamente il telefono o la rete Internet.

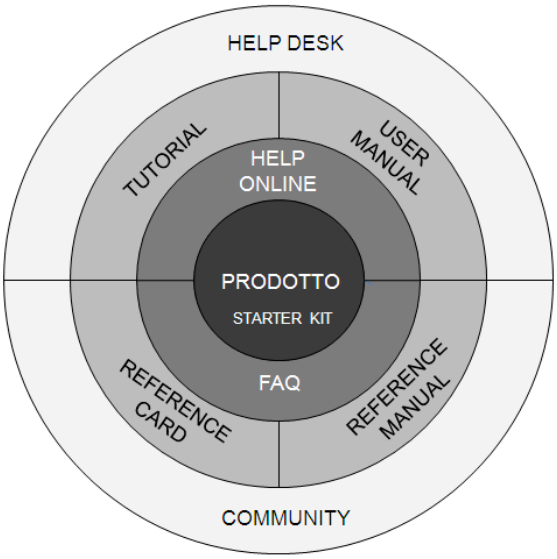


Figura 58. Il prodotto e i suoi sussidi

Questi sussidi hanno lo scopo di assistere l'utente in vari modi. Alcuni lo accompagnano nell'uso iniziale del sistema, quando non lo conosce ancora, come gli *starter kit* e i *tutorial*. I primi sono di solito brevi istruzioni che lo guidano nella prima installazione o configurazione del sistema, e sono la prima cosa da esaminare dopo avere aperto la confezione del prodotto. I secondi sono guide all'uso iniziale del sistema, di solito piuttosto brevi, che hanno lo scopo di fargli prendere familiarità con le funzioni di base. Possono essere realizzati con manuali cartacei, con testi online o, sempre più spesso, con brevi video accessibili in rete.

Altri sussidi hanno lo scopo di assistere l'utente durante l'uso successivo. I *manuali utente* (*user manual*) sono testi che descrivono il sistema in modo completo, per tutti quegli aspetti che possono interessarlo: le sue caratteristiche e come

usarlo. Sono di solito concepiti per essere letti sequenzialmente, dall'inizio alla fine. I *manuali di riferimento* (*reference manual*) sono invece pensati come strumenti di consultazione, per trovare informazioni specifiche durante l'uso. Sono quindi organizzati in modo tale da permettere un accesso rapido, non sequenziale, ai contenuti, per esempio con le voci disposte in ordine alfabetico. Le *schede di riferimento* (*reference card*) hanno la stessa funzione, ma sono molto più sintetiche: spesso una sola pagina in cui tutte le informazioni necessarie sono riassunte in schemi sinottici, concepiti per essere facilmente comprensibili da chi il sistema lo conosce già. Un agile memorandum da tenere accanto a sé durante l'uso.

L'utente, tuttavia, preferisce di gran lunga, ai manuali scritti, la disponibilità di qualcuno che gli suggerisca rapidamente che cosa fare nella situazione specifica: un amico già esperto che vada subito al punto, e gli indichi la soluzione per il problema che sta affrontando in quel momento. Infatti, spesso non è necessario, per un uso soddisfacente di un sistema, che l'utente ne abbia elaborato un modello concettuale completo, capace di spiegarne le modalità operative in ogni contesto, e l'intrinseca coerenza. Oggi, come abbiamo già osservato, alcuni sistemi sono molto complessi, e ogni utente usa solo una piccola parte delle funzioni disponibili. Pertanto, si accontenterà di conoscere un insieme (anche piuttosto limitato) di regole specifiche, del tipo "se devo ottenere x, allora dovrò fare y": non sarà interessato a inquadrarle in una visione generale coerente e organica, che descriva anche le caratteristiche che non gli servono.

Questo desiderio di arrivare subito "al punto", senza dover imparare troppe cose, è un aspetto molto importante del comportamento degli utenti, che spiega la grande diffusione di strumenti come le *FAQ* (*Frequently Asked Questions*), e i *servizi online*. Le prime sono elenchi di domande tipiche, che gli utenti si pongono nell'uso del sistema: "Come faccio a…?", "Dove trovo …?", "Perché succede che…?", e così via. Questi elenchi di domande non devono essere necessariamente organizzate in un quadro organico: spesso è utilizzata una semplice funzione di ricerca per parole chiave per trovare richiesta specifica, e la sua risposta. Nei secondi, in assenza di un esperto al fianco, l'utente interroga la rete, rivolgendosi all'help desk del fornitore del sistema, oppure alla comunità degli altri utenti. Gli strumenti utilizzati sono i forum o i newsgroup, o le chat. Nei primi, l'interazione avviene in tempo differito: l'utente esamina i testi delle conversazioni già avvenute, per verificare se qualcun altro, avendo esposto un problema simile, ha già ottenuto dalla comunità una risposta utile. In caso contrario, descrive il proprio problema, e lo inoltra in rete, sperando che, prima o poi, qualcuno risponda. Le chat, invece, supportano conversazioni in tempo reale: in questo caso l'interlocutore è in linea nello stesso momento, e la conversazione avviene con una serie di scambi domanda-risposta. Negli help-desk, le chat sostituiscono a volte i tradizionali call center telefonici.

Da molti anni è in atto una decisa *smaterializzazione* dei sussidi, che sono trasferiti dal supporto cartaceo (manuali a stampa) a quello elettronico (manuali su CD). La documentazione in formato elettronico viene poi, a partire da anni più recenti, sempre più spesso erogata esclusivamente attraverso la rete. In un mondo in cui le connessioni alla rete erano lente e i computer connessi solo quando necessario, essa era predisposta per il download da parte dell'utente, e sostanzialmente statica. In un mondo di sistemi sempre connessi (*always on*), l'utente non scarica più i manuali, ma accede alle informazioni in rete, navigando all'interno di documenti ipertestuali che risiedono permanentemente sui server dei produttori (Figura 59). Ciò permette al produttore un rapido e continuo aggiornamento della documentazione, non soltanto per allinearla alle nuove versioni del prodotto, ma anche per migliorarne la struttura dei contenuti, a seguito dei feedback da parte degli utenti. Per esempio, dopo l'accesso a un articolo della documentazione online di Microsoft Office 2007, all'utente viene chiesto: "Le informazioni contenute in questo articolo ti sono state utili?". Le risposte (Sì, No, Non so) serviranno per migliorare i successivi aggiornamenti della documentazione.

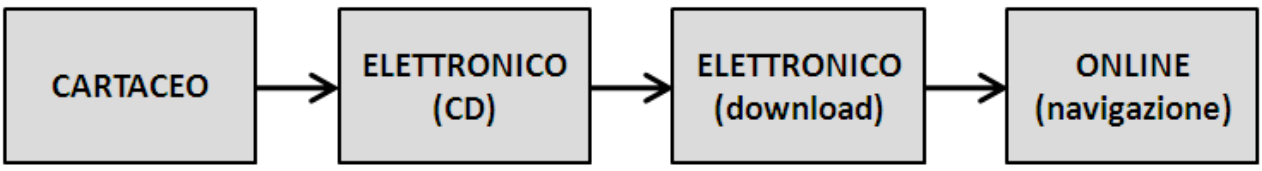


Figura 59. Evoluzione dei supporti dei sussidi all'uso

Contemporaneamente alla tendenza verso la smaterializzazione dei sussidi, e come conseguenza di questa, è in atto da tempo una tendenza all'*integrazione* degli stessi con il prodotto. L'insieme dei sussidi non è più visto come un insieme di componenti separati dal prodotto, ma come un vero e proprio *sistema di aiuto* costituito da elementi correlati e strettamente integrati con il prodotto cui si riferiscono. Prodotto e sistema di aiuto sono così visti come due componenti non separabili di uno stesso sistema. Entrambi hanno lo scopo di supportare l'utente nelle varie situazioni possibili, operando congiuntamente. La Figura 60 mostra un esempio d'integrazione stretta fra sistema e sistema di aiuto. Nel Mac, selezionando Aiuto nella barra dei menu, e ricercando la voce Seleziona tutto, comparire l'elenco delle sezioni del manuale che trattano argomenti connessi a tale voce. Inoltre, il menu Composizione viene aperto automaticamente, e appare una grande freccia che indica all'utente la voce Seleziona tutto in tale menu. È come se il sistema si sdoppiasse, e dicesse all'utente: "la voce che mi hai chiesto si trova qui".



Figura 60. Integrazione sistema - sistema di help (Apple Finder, 10.6, 2009)

I sistemi di aiuto odierni non si limitano a fornire strumenti di consultazione, sia pure integrati come in Figura 60, secondo il paradigma point&clic della navigazione ipertestuale. A volte realizzano veri e propri dialoghi con l'utente, secondo il modello (seppure ancora in una forma embrionale) dell'*assistente virtuale* ipotizzato, un quarto di secolo fa, nel video del Knowledge Navigator, con il quale la Apple immaginava i computer del futuro (vedi la Figura 117, a pag.152). Una tecnica diffusa utilizza i cosiddetti *wizard* (letteralmente: maghi), componenti software che dialogano con l'utente (attraverso semplici dialog box), guidandolo attraverso i passi necessari per effettuare un certo compito. Essi sono prevalentemente utilizzati per operazioni complesse o poco frequenti, come per esempio la configurazione iniziale di un sistema.

La smaterializzazione e l'integrazione progressiva dei sussidi all'uso si realizza completamente nelle applicazioni erogate online attraverso siti web, per le quali lo schema di Figura 58 si trasforma come in Figura 61.



Figura 61. Smaterializzazione e integrazione dei sussidi

Le applicazioni web più diffuse, essendo destinate a un pubblico vasto e indistinto, non necessariamente esperto nell'uso dei computer, contengono spesso soluzioni innovative per abbassare al minimo la "soglia di ingresso" al sistema. È frequente la dichiarazione che all'utente "bastano pochi clic" per iniziare a lavorare con profitto. Tipicamente, l'utente viene invitato a provare gratuitamente il sistema, registrandosi mediante la compilazione di una semplice form: l'esplorazione iniziale delle funzioni principali non richiede la lettura preventiva di alcuna documentazione. I vantaggi derivanti dall'uso del sistema gli vengono spesso spiegati con un breve video dimostrativo (Figura 62).



Figura 62. La home page di http://www.flickr.com invita l'utente a esplorare il sistema (2010)

In sintesi, l'usabilità di un prodotto va valutata considerando il sistema complessivo dei suoi sussidi, che spesso non sono distinguibili dal prodotto stesso. Un sistema interattivo allo stato dell'arte dovrebbe fornire *al suo interno* tutti gli strumenti per accompagnare l'utente dall'uso iniziale a un uso evoluto, aiutandolo via via a superare le difficoltà che incontrerà. Infatti, i manuali d'uso tradizionali sono letti di rado. Gli utenti preferiscono "rischiare" e provare comunque a utilizzare il sistema, anche se non lo conoscono. Ricorrono ai manuali di malavoglia e in casi estremi, a fronte di specifici problemi o impedimenti, e solo in assenza di alternative. Ce ne sono troppi: siamo circondati da manuali di ogni tipo, su carta o in formato elettronico. Se contassimo i manuali d'uso presenti in una normale abitazione di un paese sviluppato, supereremmo molto probabilmente il centinaio.[37]

Quand'anche i manuali fossero completi, di facile lettura, ben scritti o ben tradotti dalla lingua originale (e raramente lo sono), non avremmo il tempo di studiare una così imponente massa d'informazioni.

[37] Molto probabilmente troveremmo un manuale per ciascuno dei seguenti prodotti: fornello, frigorifero, forno, forno a microonde, lavastoviglie, robot per cucinare, frullatore, macchina per caffè, cappa anti-fumo, lavatrice, asciugatrice, scaldabagno, ferro da stiro, televisore, radio, player DVD, amplificatore, due decoder, condizionatore, telefono fisso, sveglia, orologi personali, oltre naturalmente al manuale del cellulare di ciascun membro della famiglia, e ai manuali relativi a tutti i personal computer e alle console per videogiochi disponibili (e relativi prodotti software e periferiche). E poi un manuale per ciascuno dei piccoli elettrodomestici della cucina e del bagno, per l'eventuale sistema di allarme. E inoltre per automobile, autoradio, sistema di guida satellitare, macchina fotografica, videocamera, e ogni altro apparecchio utilizzato nel tempo libero.

Un sistema usabile dovrebbe mettere in grado i suoi utenti di utilizzarlo senza alcun tipo di sussidio esterno al sistema stesso. In questo senso va interpretata la frase che Donald Norman, provocatoriamente, scrisse quasi un quarto di secolo fa nel suo libro *La caffettiera del masochista*:

> *Ho una regola semplice per individuare il cattivo design. Tutte le volte che trovo indicazioni su come usare qualcosa, si tratta di un oggetto progettato male.*

Usabilità universale

Come si è più volte osservato, l'usabilità è un concetto relativo. Non ha senso affermare che un prodotto è usabile in assoluto: è necessario specificare per quali utenti, per quali obiettivi e in quali contesti d'uso, come mette bene in evidenza la definizione dell'ISO 9241. Alcuni prodotti sono destinati a una ristretta categoria di utenti, per un utilizzo in contesti molto particolari. Altri sono destinati a un pubblico molto più ampio, per essere utilizzati in situazioni molto varie. A seconda dei suoi destinatari e contesti d'uso, prodotti destinati a fornire all'utente funzioni simili possono differenziarsi in modo considerevole.

Per esempio, la Figura 63 mostra quattro diversi tipi di orologi. Servono tutti a indicare l'ora, ma a utenti e in condizioni di utilizzo completamente diverse. L'orologio da parete, nel quadrante in alto a destra, è destinato a utenti generici, e può essere utilizzato in contesti molto vari: in casa, in ufficio, in un locale pubblico, e così via. Al contrario, l'orologio da polso subacqueo del quadrante in basso a sinistra è destinato a un utilizzo molto particolare. Può essere usato anche fuori dall'acqua, ma è concepito per essere utilizzato soprattutto da un subacqueo in immersione, ed è in questo contesto che dovremmo valutarne l'usabilità. Gli altri due esempi sono ancora diversi. La meridiana (quadrante in basso a destra) è destinata a chiunque sappia leggere i numeri romani, e può essere utilizzata solo quando c'è il sole. Infine, l'orologio *braille* da polso (quadrante in alto a sinistra) è destinato al pubblico – molto specifico – degli utenti non vedenti che conoscono l'alfabeto *braille*. Questi utenti lo possono indossare in qualunque situazione.

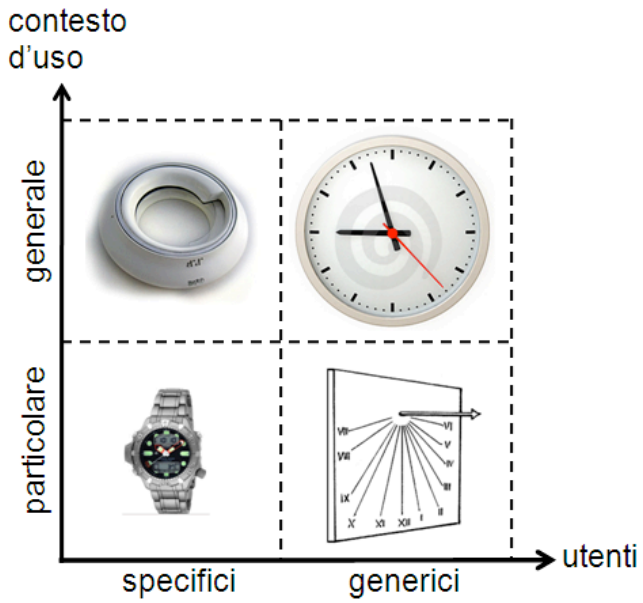


Figura 63. Classificazione dei prodotti in rapporto alla specificità della loro destinazione (utenti e contesti d'uso)

Per ciascuno di questi quattro diversi orologi, l'usabilità non può essere valutata in astratto: si dovrà tenere conto del particolare tipo di utenti ai quali è destinato, e degli specifici contesti d'uso per cui è stato concepito. L'usabilità

dell'orologio braille, per chi non sappia leggere questo alfabeto, sarà molto bassa, così come quella di una meridiana collocata in una stanza in cui i raggi del sole siano filtrati da pesanti tendaggi.

Per i prodotti e i servizi destinati a un'utenza generica, e che risultano usabili per tutti, in contesti generici, è stato coniato il termine di *usabilità universale* (*universal usability*, Figura 64).[38]

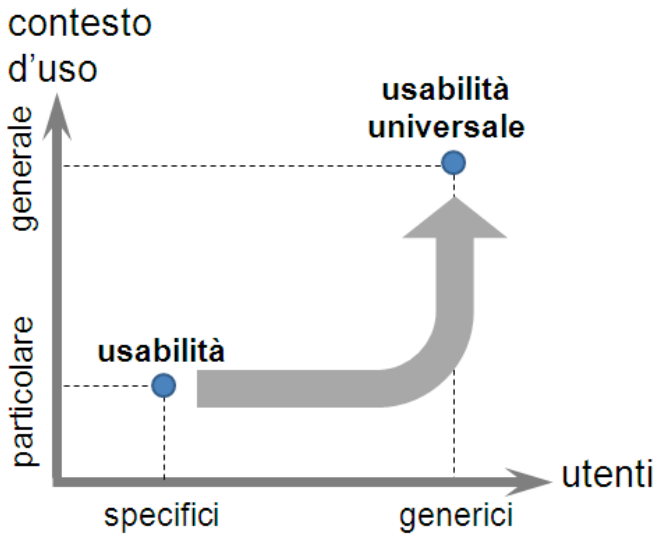


Figura 64. Dall'usabilità all'usabilità universale

La nozione di usabilità universale è chiaramente molto importante: un prodotto o servizio universalmente usabile può essere utilizzato facilmente da tutti, senza discriminazioni. Moltissimi prodotti possono essere progettati, senza troppe difficoltà, in modo da essere universalmente usabili: un coltello, un bicchiere, una penna. Per i sistemi interattivi più complessi, come per esempio i prodotti software, le cose sono chiaramente molto più complicate, come vedremo nel capitolo 5.

Accessibilità

Strettamente correlato al concetto di usabilità universale è quello di *accessibilità* (*accessibility*). Questo termine è nato in ambito architettonico, dove è utilizzato da molti anni per indicare la possibilità di accedere agli edifici da parte di persone con disabilità motorie (tipicamente, utilizzatori di sedie a rotelle), senza che esistano delle barriere architettoniche che ne ostacolino la mobilità. Il termine è stato successivamente adottato anche nell'ambito dell'informatica. In questo caso, le barriere che impediscono l'accesso ai sistemi da parte di utenti con disabilità non sono, ovviamente, architettoniche, ma di altro tipo. Per esempio, un non vedente non è in grado di interagire con un sistema informatico o un sito web che gli comunichi le informazioni necessarie all'uso soltanto attraverso il canale visivo; un utente affetto da daltonismo potrebbe avere difficoltà a discriminare informazioni veicolate soltanto attraverso il colore, e così via.

Le persone affette da qualche tipo di disabilità costituiscono una percentuale significativa della popolazione. Almeno il 10% della popolazione mondiale è disabile, cioè, secondo la definizione dell'Organizzazione Mondiale della Sanità, è incapace di svolgere le normali attività della vita quotidiana a seguito di qualche menomazione.[39] Secondo stime

[38] Il termine è stato proposto da Ben Shneiderman, nell'articolo *Universal Usability*, Communications of the ACM, vol.43, n.5 (maggio 2000), pagg.85-91.

[39] Per menomazione si intende qui il danno biologico che una persona riporta a seguito di una malattia (congenita o meno) o di un incidente. Si noti che il concetto di disabilità è cambiato considerevolmente nel corso degli anni. La *Convenzione sui diritti dei*

dell'Istat sulla base di dati del 2004-2005, le persone con disabilità sono, In Italia, circa 2 milioni e 800 mila, corrispondenti al 5% circa della popolazione del Paese.[40]

In Italia, l'accessibilità dei sistemi informatici è regolata dalla legge n.4 del 9 gennaio 2004, *Disposizioni per favorire l'accesso dei soggetti disabili agli strumenti informatici*. Questa legge si propone di abbattere le barriere che limitano l'accesso dei disabili alla società dell'informazione e li escludono dal mondo del lavoro, dalla partecipazione democratica e da una migliore qualità della vita, in applicazione del principio di eguaglianza sancito dalla nostra Costituzione. Essa definisce l'accessibilità come

> *la capacità dei sistemi informatici, nelle forme e nei limiti consentiti dalle conoscenze tecnologiche, di erogare servizi e fornire informazioni fruibili, senza discriminazioni, anche da parte di coloro che a causa di disabilità necessitano di tecnologie assistive o configurazioni particolari.*

Per *tecnologie assistive* la legge intende

> *gli strumenti e le soluzioni tecniche, hardware e software, che permettono alla persona disabile, superando o riducendo le condizioni di svantaggio, di accedere alle informazioni e ai servizi erogati dai sistemi informatici.*

Tecnologie assistive sono quindi, per esempio, i lettori di schermo (che leggono "ad alta voce" i testi visualizzati sullo schermo del computer, per permetterne l'accesso a utenti non vedenti), le tastiere Braille, e così via. Al di fuori dell'informatica, si possono considerare tecnologie assistive, per esempio, le stampelle, le sedie a rotelle, le protesi, ecc.

Negli ultimi anni, i legislatori dei diversi Paesi hanno prestato particolare attenzione alle problematiche dell'accessibilità dei siti web, considerando la pervasività della rete nella vita quotidiana. In Italia, la legge 4/2004 già citata prescrive che i contratti stipulati dalla pubblica amministrazione per la realizzazione di siti web siano nulli, qualora non rispettino opportuni requisiti di accessibilità. In sostanza, i siti web degli Enti della Pubblica Amministrazione italiana *devono* (o dovrebbero) essere, per legge, tutti accessibili.[41]

Anche se, nell'uso comune, il termine accessibilità è associato soprattutto ai soggetti disabili, esso viene usato spesso con una valenza più ampia, per indicare la possibilità di accesso ai sistemi non solo da parte di portatori di handicap in senso stretto, ma anche da chi soffre di disabilità temporanee o dispone di attrezzature obsolete o comunque con prestazioni carenti, per esempio connessioni internet molto lente. Anche queste costituiscono, infatti, delle "barriere" che separano l'utente dagli strumenti informatici, compromettendone o impedendone l'utilizzo (Figura 65).

---

disabili promulgata dall'ONU nel 2007 definisce le persone disabili come "coloro che presentano una duratura e sostanziale alterazione fisica, psichica, intellettiva o sensoriale la cui interazione con varie barriere può costituire un impedimento alla loro piena ed effettiva partecipazione nella società, sulla base dell'uguaglianza con gli altri." Oggi quindi il termine identifica le difficoltà di funzionamento della persona sia a livello personale che nella partecipazione sociale.

[40] Per questi dati, e altri sulla situazione italiana, si veda il sito http://www.disabilitaincifre.it, promosso dal Ministero della Solidarietà Sociale e realizzato dall'Istat.

[41] Esistono vari livelli di accessibilità. Lo spazio a disposizione non ci permette di entrare nei dettagli. Il lettore interessato può consultare il sito http://www.pubbliaccesso.gov.it/, realizzato a cura del CNIPA (Centro Nazionale per l'Inbformatica nella Pubblica Amministrazione). Esso raccoglie la normativa italiana in tema di accessibilità informatica, i documenti di approfondimento, manuali e testi di riferimento, studi e recensioni, prove di prodotti hardware e software ed esempi di siti accessibili.

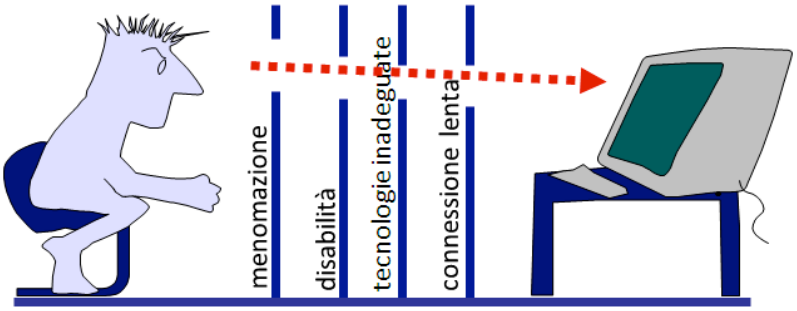


Figura 65. Le barriere all'accesso ai sistemi informatici

A volte, si usa anche il termine *accessibilità universale* (*universal accessibility*) per enfatizzare ulteriormente un'accessibilità estesa a *tutti i possibili utenti*, indipendentemente dalle eventuali ulteriori barriere costituite dalla loro classe sociale, lingua, etnia, cultura, collocazione geografica o altro.

Non bisogna confondere usabilità e accessibilità, sono due concetti diversi, come si comprende facilmente rileggendone le definizioni. L'accessibilità garantisce la possibilità d'accesso al sistema, mentre l'usabilità ne garantisce un uso efficiente, efficace e soddisfacente. Quindi un sistema può essere accessibile, ma non usabile. Per esempio, un non vedente potrebbe riuscire a conoscere i contenuti di una pagina web mediante l'uso di un lettore di schermo, anche se questa non fosse stata strutturata in modo ottimale a questo scopo. In altre parole, vi può accedere, ma in modo poco efficiente, poco efficace e poco soddisfacente.

Inoltre, come abbiamo più volte notato, l'usabilità è un concetto relativo: si riferisce a *specifici* utenti, compiti e contesti d'uso. Il termine accessibilità viene invece, in prevalenza, utilizzato con un significato assoluto: un sistema accessibile è un sistema accessibile *a tutti* (o quasi). Ne segue che un sistema può essere usabile ma non accessibile. Infatti, potrebbe essere usabile (cioè efficace, efficiente, soddisfacente) per utenti dotati di normali abilità e dotazione tecnologica, ma inaccessibile ad altri utenti che non si trovano in queste favorevoli condizioni (Figura 66).[42]

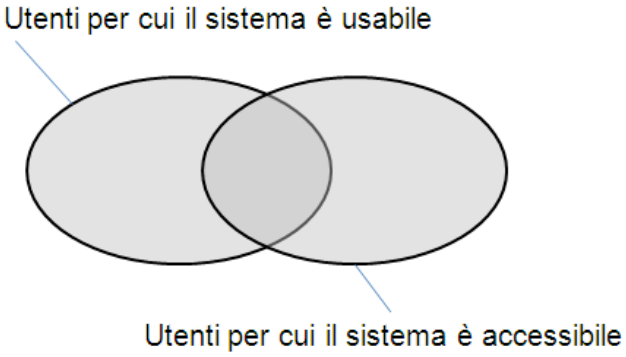


Figura 66. Usabilità e acccessibilità

[42] Invece, un sistema universalmente usabile è a maggior ragione universalmente accessibile (se non posso accedere, non posso considerarlo facile da usare). Un sistema universalmente accessibile non è detto che sia universalmente usabile.

Ripasso ed esercizi

1. Spiega che cosa intende Donald Norman per "golfo dell'esecuzione" e per "golfo della valutazione".
2. Spiega il concetto di affordance, e fornisci tre esempi di oggetti dotati di affordance e tre esempi di oggetti senza affordance, spiegandone il perché.
3. Analizza il processo di utilizzo dell'ascensore di casa tua utilizzando il modello di Norman. Come valuti l'ampiezza del "golfo dell'esecuzione" e del "golfo della valutazione" e perché?
4. Analizza le affordance di tale sistema. Possono essere migliorate? Come?
5. Che cosa significa usabilità secondo l'ISO 9241?
6. Utilizzando la definizione dell'ISO 9241, analizza l'usabilità dell'ascensore di casa tua. Può essere considerato usabile? Perché? Quali metriche utilizzeresti per confrontarne l'usabilità con quella di altri ascensori?
7. Utilizzando la definizione di usabilità dell'ISO 9241, analizza l'usabilità di un elettrodomestico di casa tua (es.: il frigorifero, il forno a microonde, il fornello). Può essere considerato usabile? Perché? Quali metriche useresti per valutarne quantitativamente l'usabilità?
8. Individua nel tuo cellulare una funzione che consideri poco usabile, e spiegane i motivi, facendo riferimento alla nozione di usabilità dell'ISO 9241.
9. Che cosa significa "apprendibilità"? Per quali categorie di prodotti è una proprietà importante?
10. Che cosa significa "memorabilità"? Per quali categorie di prodotti è una proprietà importante?
11. Spiega il senso della seguente affermazione di Donald Norman: "Tutte le volte che trovo indicazioni su come usare qualcosa, si tratta di un oggetto progettato male".
12. Che cosa significa usabilità universale?
13. Definisci la nozione di accessibilità e confrontala con quella di usabilità.

Approfondimenti e ricerche

1. Leggi il classico libro di Donald Norman, *La caffettiera del masochista* (edizione Giunti, 1990 e successive edizioni). Si tratta di un libro breve e divertente, che ha avuto una enorme influenza sugli studi sulla usabilità.
2. Cerca in rete diverse definizioni di usabilità, e confrontale con quella discussa nel presente capitolo. Puoi iniziare, per esempio, da http://www.upassoc.org/usability_resources/about_usability/definitions.html.
3. Approfondisci il concetto di affordance, per esempio iniziando dalla nota di Donald Norman in http://www.jnd.org/dn.mss/affordances_and.html.
4. Analizza i sussidi all'utente disponibili nel sistema operativo che utilizzi normalmente, e identificane le diverse tipologie sulla base di quanto discusso nel presente capitolo. Confrontali con i sussidi disponibili in un'applicazione web che utilizzi spesso (per esempio, Facebook).
5. Leggi l'articolo di Shneiderman, *Universal Usability*, citato più sopra. È disponibile in rete all'indirizzo http://www.cs.umd.edu/~ben/p84-shneiderman-May2000CACMf.pdf .
6. Cerca in rete il testo della legge 4/2004 sull'accessibilità, e riassumine il contenuto.

4. Conoscere l'utente

Sintesi del capitolo

In questo capitolo si osserva che gli utenti possono essere considerati da molteplici punti di vista. Possiamo studiare i processi cognitivi che ne governano pensieri e azioni, le caratteristiche personali che li caratterizzano nella loro individualità, i comportamenti, e, infine, il loro ruolo nei confronti dei sistemi che utilizzano. Lo studio degli utenti da questi diversi punti di vista richiede i metodi e le conoscenze di discipline molto diverse, che possono fornire utili contributi alla disciplina della human-computer interaction. Per il progettista di sistemi interattivi, particolarmente utili sono le conoscenze della psicologia sperimentale e i metodi dell'etnografia. Per quanto riguarda la prima, il capitolo introduce brevemente alcune nozioni di base, utili al progettista, relative all'attenzione, alla memoria, alla visione e al sistema motorio umano. Accenna quindi ai metodi e alle funzioni nell'etnografia nel progetto di sistemi interattivi.

La diversità degli utenti

Nel capitolo precedente si è visto come l'usabilità non sia una proprietà intrinseca dei sistemi interattivi, ma una proprietà relativa allo specifico utente, compito da svolgere e contesto di utilizzo. La grande diversità degli esseri umani fa sì che, anche considerando compiti e contesti d'uso simili, un oggetto potrebbe risultare usabile per un certo utente e del tutto inusabile per un altro. Ecco perché la conoscenza dell'utente è di importanza fondamentale per chi progetti sistemi. In questo capitolo vogliamo approfondire questo tema.

La parola *utente* deriva dal latino *utens*, participio del verbo *uti*, che significa usare. Quindi utente è, semplicemente, "colui che usa", e in particolare, nel nostro caso, colui che utilizza un prodotto o un servizio interattivo. Questo termine spoglia, per così dire, l'utente della sua individualità. Definendo qualcuno "utente", scegliamo di ignorare tutto ciò che lo caratterizza come persona, e di qualificarlo semplicemente in relazione al prodotto o al servizio di cui si serve. Poiché non ne sottolineiamo le caratteristiche individuali, siamo portati a considerarlo quasi una entità astratta. Questo è chiaramente pericoloso, perché sposta la nostra attenzione sul sistema, di cui l'utente diviene quasi un'appendice inessenziale. Come vedremo meglio nei prossimi capitoli, l'approccio dell'ingegneria dell'usabilità, oggetto di questo libro, è porre l'utente, e non il sistema, al centro dell'attenzione del progettista (Figura 67).

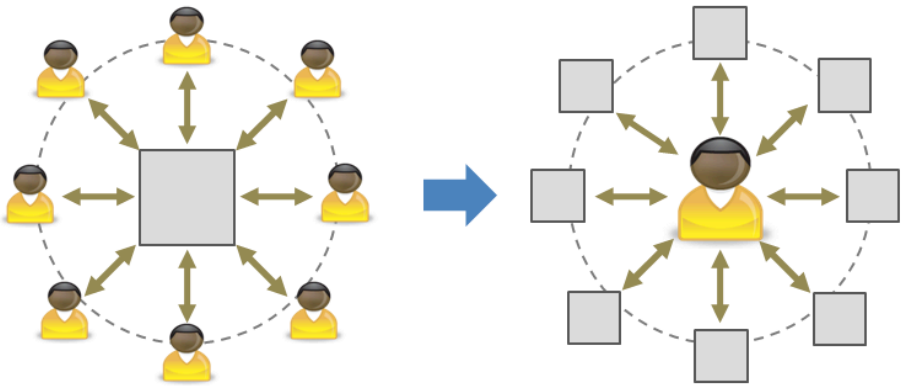


Figura 67. Da una visione centrata sul sistema a una visione centrata sull'utente

Lo studio dell'utente può essere compiuto a differenti livelli. Al livello *cognitivo,* consideriamo l'utente come essere umano dotato di specifiche abilità, derivanti dalla struttura del suo corpo e della sua mente. Una creatura che percepisce, interpreta, decide e agisce in un certo modo innanzitutto perché è dotato di un apparato sensoriale, cognitivo e motorio con specifiche caratteristiche, diverse da quelle di altre specie animali. Per fornirgli strumenti che possa utilizzare con profitto, è necessario conoscere queste caratteristiche: non possiamo chiedergli di svolgere compiti che non è fisicamente in grado di eseguire, e non possiamo pretendere elaborazioni che il suo cervello non può effettuare. A

questo scopo, utilizziamo le conoscenze dell'ergonomia e dell'ergonomia cognitiva, che a loro volta si basano sulle conoscenze scientifiche derivanti dallo studio della psicologia e della fisiologia umana.

Le diversità fra esseri umani, anche considerando solo il livello cognitivo, sono rilevanti. Quando poi, da quest'ambito strettamente funzionale, passiamo a considerare gli utenti come *persone*, ciascuno con una specifica formazione, cultura e storia personale, le diversità si fanno ancora più marcate, e caratterizzano ogni specifico utente nella sua individualità. La persona Roberto non sarà solo descritta da parametri che ne determinano le prestazioni fisiche e cognitive. Roberto avrà un'età, un luogo di nascita, una lingua, una formazione, una professione, una storia personale. Avrà motivazioni, preferenze, interessi, amicizie, relazioni e sogni. Tutto ciò ne fa un individuo unico, diverso da ogni altra persona, e ne influenzerà i comportamenti, in modo molto specifico.

Quando, oltre alle caratteristiche individuali, vogliamo studiare il *comportamento* delle persone, dobbiamo considerarne anche i rapporti che queste hanno con gli altri, e in particolar modo con i membri delle comunità e organizzazioni cui appartengono. La conoscenza dei comportamenti degli utenti, nelle loro relazioni a uno specifico prodotto o servizio, ha un grande valore per il progettista. Solo conoscendone in dettaglio i comportamenti egli sarà in grado di comprenderne i problemi e le necessità, e, quindi, di individuare le soluzioni più adeguate.

Un livello di analisi dell'utente ancora diverso è quello che ne analizza il suo *ruolo* in rapporto al sistema. In una rappresentazione teatrale, o in film, il ruolo è la parte sostenuta da un attore: diciamo che un certo attore interpreta il ruolo di protagonista, o di caratterista, o di comparsa o, ancora, del marito, o del seduttore, e così via. Il ruolo è la funzione che questa persona svolge in rapporto alla rappresentazione.[43] Analogamente, gli utenti di un sistema interattivo possono rivestire ruoli diversi, secondo la funzione che esercitano in rapporto ad esso. Per esempio, un utente di un sito web con il ruolo di *amministratore* svolge una funzione diversa da quella dei *redattori* dello stesso sito. Infatti, essi hanno compiti, responsabilità e diritti di accesso differenti. L'amministratore può cambiare la struttura complessiva del sito, il redattore può soltanto creare o modificare gli articoli nelle sezioni a lui riservate. Ancora diverso è il ruolo dell'*utente finale*, cioè colui che accede al sito dalla rete, per conoscerne i contenuti.

Persona e ruolo sono concetti diversi. Roberto e Marco sono persone differenti, ma entrambi potrebbero rivestire uno stesso ruolo, per esempio quello di redattore. Al contrario, due ruoli diversi potrebbero essere rivestiti da una stessa persona (Figura 68). La definizione del ruolo permette di separare le caratteristiche individuali della persona dagli aspetti legati allo scopo dell'interazione con il sistema.

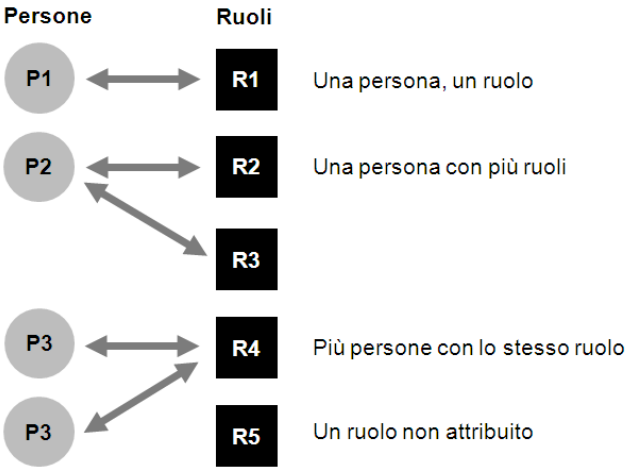


Figura 68. Persone e ruoli

[43] Infatti, il termine ruolo deriva dal francese *role*, che a sua volta deriva dal latino *rotulus*, foglio di pergamena arrotoloto dal quale gli attori teatrali, nell'antichità, leggevano le battute.

In conclusione, lo studio dell'utente può essere condotto su piani diversi, in funzione degli aspetti che cui siamo interessati, come suggerito nella Figura 69. Qui, lo schema a piramide intende segnalare che le diversità fra gli utenti emergono ad ogni livello: nelle prestazioni cognitive, nelle caratteristiche personali, nei comportamenti, nei ruoli che gli utenti possono rivestire nell'interazione con uno specifico sistema.

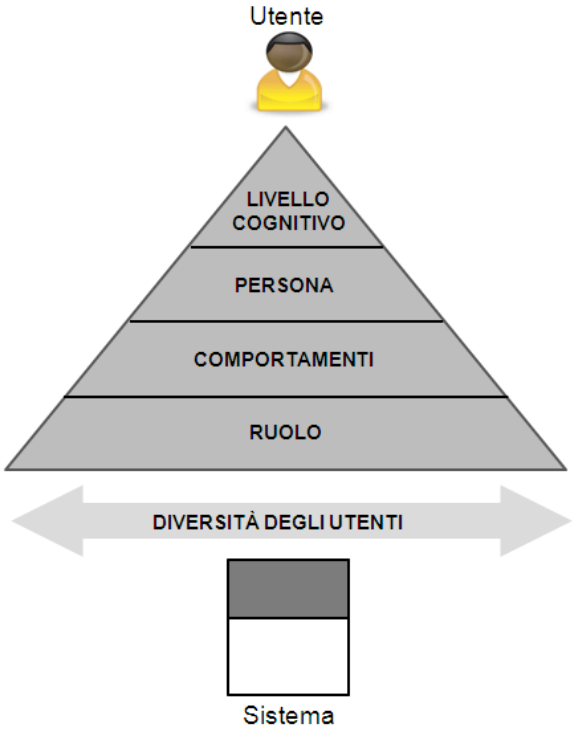


Figura 69. Livelli di descrizione dell'utente

Ogni livello deve essere studiato con metodi diversi. In particolare, nella progettazione dei sistemi interattivi, sono particolarmente utili, come vedremo nel seguito di questo capitolo, i metodi della *psicologia sperimentale*, che cerca di determinare, con opportuni esperimenti, le prestazioni e i comportamenti dell'essere umano in specifiche circostanze, e quelli dell'*etnografia*, che indaga i comportamenti delle persone, attraverso metodi basati sull'osservazione diretta, nel contesto della loro vita quotidiana.

Modelli dell'utente

Le caratteristiche del sistema cognitivo dell'uomo sono studiate dalla psicologia, che fornisce molte informazioni utili per chi si occupa di interazione uomo-macchina. Nell'ambito di quest'ultima disciplina, sono stati proposti diversi modelli che rappresentano l'essere umano come un sistema per elaborare le informazioni (*human information processor*). Essi non pretendono di descrivere l'apparato cognitivo umano come è nella realta, ma si limitano a metterne in evidenza le caratteristiche più rilevanti per chi desideri studiare – o progettare – l'interazione uomo-macchina. In particolare, questi modelli hanno permesso di definire dei parametri quantitativi della "macchina umana" che permettono di eseguire calcoli precisi sulle prestazioni teoricamente possibili per l'esecuzione di particolari compiti, per esempio la digitazione di un testo su una tastiera.

Una pietra miliare di questi studi è stata la pubblicazione, nel 1983, del libro *The Psychology of Human-Computer Interaction,* in cui gli autori Card, Moran e Newell rappresentavano l'utente con un modello ispirato alla struttura di un computer (*model human processor*, *MHP,* Figura 70). Da questo modello, e utilizzando i risultati degli studi della

psicologia sperimentale, riassumevano una serie di parametri e di leggi di funzionamento della "macchina umana", che permettevano analisi quantitative dei compiti:

> *Il Model Human Processor può essere diviso in tre sottosistemi interagenti: (1) il sistema percettivo, (2) il sistema motorio, e (3) il sistema cognitivo, ciascuno con le sue memorie e processori. Il sistema percettivo consiste di sensori e associati buffer di memoria. I buffer più importanti sono quello per le immagini visive (Visual Image Store) e quello per le immagini auditive (Auditory Image Store), che servono a contenere gli output del sistema sensoriale mentre vengono codificati in modo simbolico. Il sistema cognitivo riceve l'informazione codificata da questi buffer nella memoria di lavoro (working memory) e usa l'informazione immagazzinata in precedenza nella memoria a lungo termine (long-term memory) per prendere le decisioni su come rispondere. Il sistema motorio porta a termine la risposta. In modo approssimato, l'elaborazione dell'informazione da parte dell'essere umano sarà descritta come se ci fossero dei processori separati per ciascun sottosistema: un processore percettivo, uno cognitivo e uno motorio. Per alcuni compiti (premere un tasto in risposta a una luce) l'essere umano si comporta come un processore seriale. Per altri (digitare su una tastiera, leggere, fare una traduzione simultanea) sono possibili operazioni parallele e integrate dei tre sottosistemi, come se fossero tre processori in pipeline: l'informazione fluisce con continuità dall'input all'output con un ritardo temporale caratteristico, che mostra che i tre processori stanno lavorando simultaneamente. Le memorie e i processori sono descritti da pochi parametri.* [44]

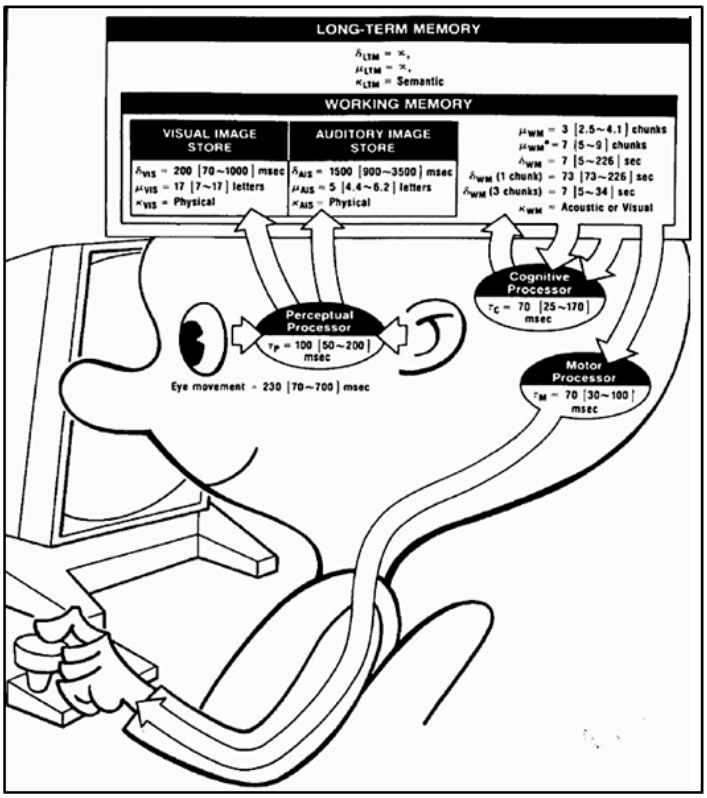


Figura 70. Model Human Processor (Card, Moran, Newell, 1983)

Il modello è complesso e, nelle quasi tre decadi trascorse da allora, le conoscenze sulle caratteristiche dei processi umani sono cresciute in modo considerevole. Non è quindi il caso di addentrarci, in questa sede, in dettagli che possono essere studiati su testi specificamente dedicati a questi argomenti. Ciò che qui interessa sottolineare è la grande portata innovativa del metodo proposto da Card, Moran e Newell. In sostanza, la conoscenza dei valori associati ai parametri

[44] S.K.Card, T.P.Moran, A.Newell, *The Psychology of Human-Computer Interaction*, Lawrence Erlbaum Associates, 1983, pag. 24-25 (nostra traduzione).

che caratterizzano i processi sensoriali, cognitivi e motorii dell'essere umano permette al progettista di predire le prestazioni dell'utente tipico nell'effettuazione dei compiti richiesti dal sistema in fase di progettazione. Questo può essere fatto sulla carta, con calcoli più o meno complessi, senza la necessità di condurre esperimenti di utilizzo del sistema da parte dell'utente (*analisi dei compiti*, o *task analysis*). Il vantaggio di questo approccio è evidente. Il progettista potrà valutare comparativamente soluzioni di progetto diverse, prima della loro realizzazione, per scegliere quella più conveniente dal punto di vista prastazionale.

In particolare, il metodo *GOMS* (*Goals, Operators, Methods, Selection rules*), proposto nello stesso libro e in seguito sviluppato in numerose varianti, decompone l'interazione con il sistema nelle sue azioni elementari (che possono essere fisiche, come la pressione di un tasto, percettive, o cognitive). I *Goals* sono gli obiettivi che l'utente intende raggiungere, gli *Operators* le azioni da compiere per raggiungerli. I *Methods* sono le sequenze di operatori disponibili per raggiungere ogni singolo goal. Quando ci sono più metodi possibili, le *Selection rules* descrivono i criteri di selezione di un metodo rispetto agli altri. Ad ogni azione elementare vengono associati dei tempi di esecuzione, derivanti dal modello, e questi vengono poi sommati tenendo conto dei metodi e delle regole di selezione.

Questa tecnica permette di individuare dei limiti prestazionali, per esempio il tempo minimo necessario per copiare in una form sul video i dati presenti su un documento cartaceo. Sono informazioni importanti, ma hanno dei limiti. L'uomo non è una macchina, che – se costruita con gli stessi componenti – si comporta sempre allo stesso modo. Nell'essere umano le variazioni individuali sono rilevanti, e i parametri utilizzati nel calcolo possono avere notevoli variazioni. Per esempio, l'età o il possesso di particolari abilità o disabilità possono influenzare in modo rilevante le nostre prestazioni. Inoltre, il nostro comportamento è influenzato da numerosi fattori di cui il modello non tiene conto: la fatica, l'ambiente circostante, i vincoli organizzativi, le motivazioni che abbiamo nell'esecuzione del compito, e così via. Poi, durante l'interazione possiamo fare degli errori, ed essere costretti a ripetere più volte alcune sequenze di azioni. Oppure, ancora, possiamo non conoscere bene il sistema, e quindi usarlo in modo diverso da quello teoricamente ottimale.

Dato lo scopo introduttivo di questo libro, non approfondiremo più oltre i metodi di analisi dei compiti, come il GOMS. Descriveremo tuttavia, nel seguito di questo capitolo, alcune caratteristiche del funzionamento dello "human processor", che sono di grande importanza per la progettazione e per lo studio dei sistemi interattivi. Questi aspetti faranno riferimento allo schema di Figura 71, tratto dal MHP, con qualche semplificazione e aggiunta. [45] Esaminiamo dunque singolarmente alcuni di questi componenti, mettendone in evidenza le caratteristiche di nostro interesse. Data la natura introduttiva di questo libro, e l'ampiezza delle problematiche coinvolte, ci limiteremo a pochi cenni essenziali, rimandando il lettore che desideri maggiori informazioni ai testi di psicologia generale.

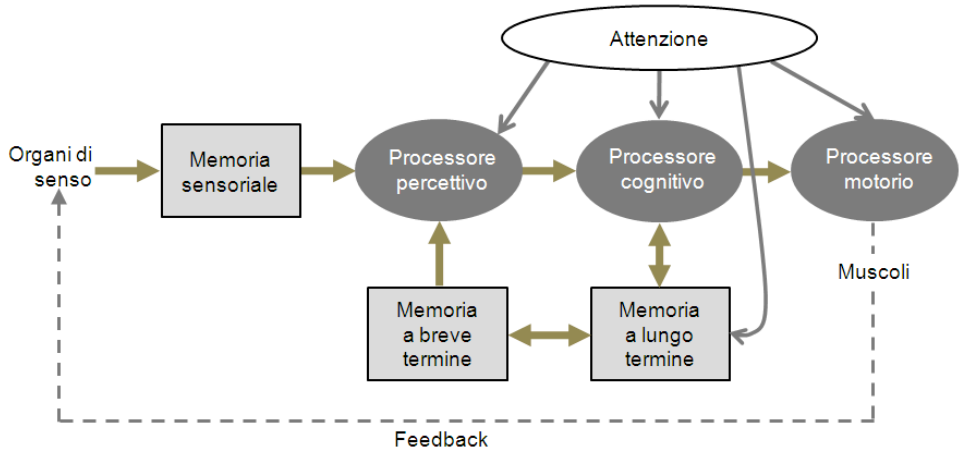


Figura 71. Model Human Processor adattato

[45] Il modello è adattato dalle slide del corso *User Interface Design and Implementation* di Rob Miller, MIT, in http://courses.csail.mit.edu/6.831/archive/2008.

L'attenzione

Tutti noi crediamo di sapere che cosa è l'attenzione, ma definirla scientificamente è molto difficile, e le opinioni dei ricercatori sono discordi. Infatti, questo termine non si riferisce a un fenomeno unitario, ma piuttosto a una serie di processi psicologici fra loro molto differenti. Per gli scopi di questo libro, la possiamo intendere, semplicemente, come quell'insieme di processi cognitivi che ci permettono di selezionare, fra tutte le informazioni che arrivano ai nostri sensi, quelle che in qualche modo ci interessano. Selezionare significa ignorare determinati stimoli, a favore di altri: per questo, viene spesso usata la metafora del filtro. Se non possedessimo un meccanismo di filtraggio, l'informazione da elaborare in ogni istante della nostra vita sarebbe eccessiva, perché il nostro apparato cognitivo ha capacità limitate. Basti pensare alla quantità d'informazione visiva che si presenta in ogni momento ai nostri occhi quando ci guardiamo intorno, per comprendere che, se non ci fossero dei meccanismi di selezione, saremmo costretti a elaborare un'enorme massa di dati che non ci servono. [46] Per esempio, quando parliamo con una persona, di solito concentriamo la nostra attenzione sulle sue parole, la sua bocca e i suoi occhi, e ignoriamo gli altri stimoli che provengono dall'ambiente circostante. Se siamo in una strada rumorosa, fra tutti gli stimoli che arrivano alle nostre orecchie selezioniamo solo le parole del nostro amico, ignorando i clacson delle automobili e il rumore del traffico, anche se fossero molto forti. Li percepiamo, ma restano in sottofondo: descriviamo questa situazione dicendo, appunto, che non vi prestiamo attenzione. Se però, mentre stiamo chiacchierando, qualcuno pronuncia il nostro nome, noi ce ne accorgiamo. Non occorre che urli: anche se lo fa a bassa voce, la nostra attenzione – che ignorava gli stimoli più forti dei rumori stradali – viene attratta da questo nuovo stimolo, e ci consente di elaborarlo. È come se scattasse un allarme, che ci distoglie per un momento da ciò cui stavamo prestando attenzione, e ci costringe a rivolgerla altrove. In altre situazioni, la nostra attenzione riesce a focalizzarsi (sia pure in modo limitato) su più eventi in contemporanea, per esempio quando guidiamo l'automobile mentre parliamo al cellulare.

Si parla di *attenzione selettiva* (*selective attention*) quando ci focalizziamo su un singolo evento escludendo gli altri, e di *attenzione divisa* (*split attention*) quando seguiamo contemporaneamente più eventi. Non si tratta di due fenomeni indipendenti, ma di due aspetti dello stesso fenomeno. Le teorie sull'attenzione che sono state elaborate, e i numerosi studi sperimentali che cercano di confermarle, sono al di fuori degli scopi di questo libro. Esaminiamo, invece, alcuni aspetti importanti per i temi di nostro interesse.

Innanzitutto, l'attenzione può essere influenzata da fattori esterni (*esogeni*) e interni (*endogeni*). Nel primo caso, sono le caratteristiche degli stimoli che arrivano ai nostri sensi che la influenzano, e numerosi studi cercano di determinare queste caratteristiche. Come quando qualcuno ci chiama per nome. Oppure come quando la nostra attenzione è attratta dal cerchio nero di Figura 72a, che spicca per la sua diversità fra tutti gli altri cerchi bianchi.

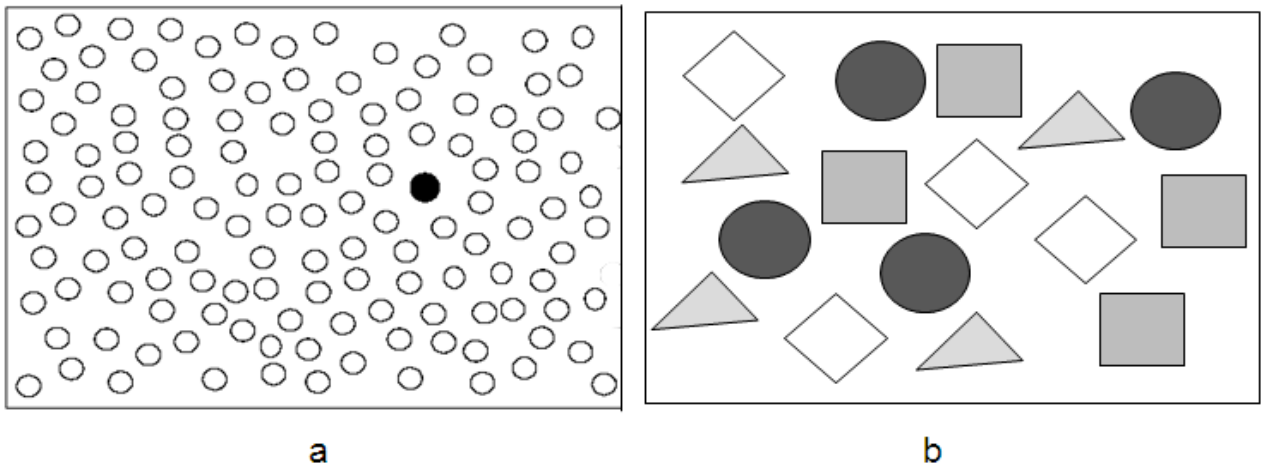


Figura 72. Test per attenzione selettiva

[46] Per esempio, possiamo quantificare questa informazione in termini di numero di byte necessari per rappresentarla, alla risoluzione percepita dal nostro apparato visivo.

Nel secondo caso, la nostra attenzione è guidata da noi stessi: dai nostri obiettivi, dalle nostre emozioni, dai nostri pensieri. L'importanza dei fattori endogeni è resa evidente da molti semplici esperimenti, come il seguente. Si mostri a qualcuno la Figura 72b, e gli si chieda di contare i triangoli. Poi si nasconda la figura e gli si chieda il numero dei quadrati bianchi. Il nostro interlocutore non sarà in grado di rispondere: il compito che si era assegnato (contare i triangoli) gli aveva fatto ignorare tutte le altre informazioni. La sua attenzione era rivolta esclusivamente ai triangoli.

La nostra attenzione è tanto più focalizzata quanto più siamo impegnati in un compito difficile. Se si mostra a un soggetto un breve video di una squadra di pallacanestro in azione, chiedendogli di contare quante volte i giocatori si passano la palla, egli sarà così concentrato su questo compito, piuttosto impegnativo se i passaggi sono rapidi, da non vedere altri eventi che si svolgono nel campo da gioco. Per esempio, non si accorgerà che un attore vestito da gorilla passeggia tranquillamente fra i giocatori.[47]

Consideriamo ora l'attenzione divisa. Essa ha possibilità limitate: non siamo in grado di prestare attenzione a troppe cose contemporaneamente. Durante la visione del video di cui sopra, quasi certamente non riusciremo a contare, contemporaneamente, i passaggi di palla, i giocatori con i baffi e quelli con i capelli biondi. Gli esperimenti mostrano che quando cerchiamo di prestare attenzione a più compiti contemporaneamente, le prestazioni sono in genere peggiori di quelle ottenute quando siamo impegnati negli stessi compiti, ma uno per volta. I sistemi che richiedono all'utente attenzione divisa possono risultare quindi poco usabili. Interfacce con queste caratteristiche sono per esempio quelle che servono a controllare apparati complessi, come il cruscotto di un aereo (Figura 73). Il progettista dovrà tenerne conto, e semplificare l'interfaccia in modo opportuno.

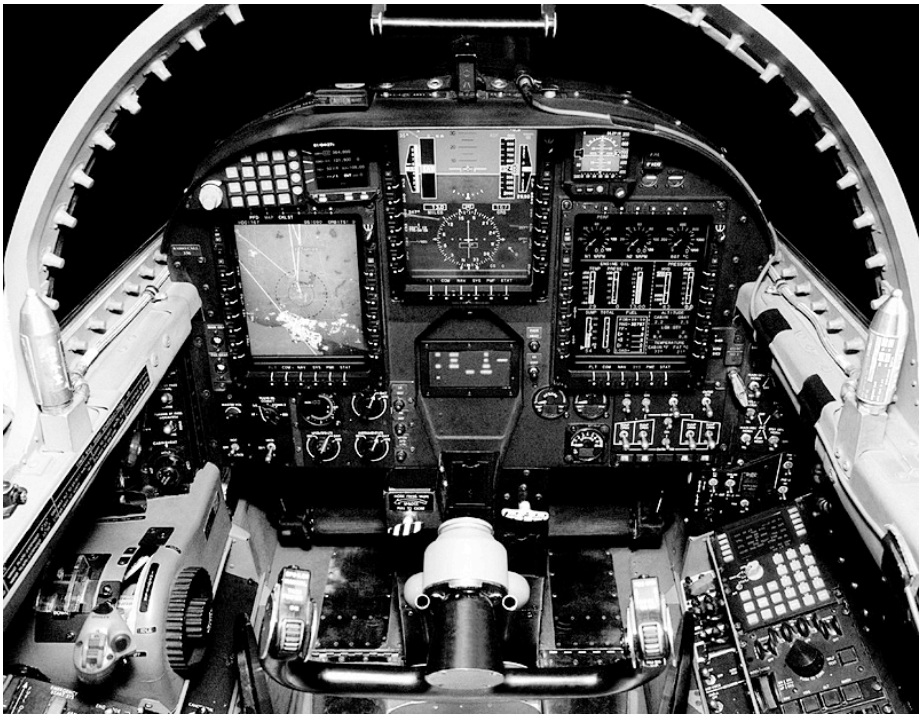


Figura 73. Il cruscotto dell'U2, un aereo da ricognizione monoposto della Lockheed prodotto dagli anni '50

I sistemi operativi oggi permettono di eseguire più programmi contemporaneamente (*multitasking*): per un computer, la capacità di "attenzione divisa" è limitata solo dalla potenza del processore. Per gli esseri umani, queste capacità sono, come abbiamo visto, piuttosto limitate. Sembra, tuttavia, che l'abitudine a seguire al computer più processi che avvengono in contemporanea costituisca una sorta di addestramento, che migliora le nostre capacità di attenzione divisa. In effetti, i giovani abituati a usare il computer fin dai primi anni di vita (*nativi digitali*) eseguono spesso molti

[47] In rete esistono molti video di questo tipo, in diverse varianti. Per trovarne degli esempi, si cerchi su http://www.youtube.com, usando la frase chiave "awareness test".

compiti in parallelo: rispondono alle mail, chattano con gli amici, inviano e ricevono messaggi su una social network, ascoltano musica, navigano in internet. Questo porta a una grande frammentazione delle attività, che persone di generazioni precedenti non riescono a gestire con la stessa facilità.[48]

In ogni caso, è evidente che il Web pone oggi agli utenti formidabili problemi di esercizio dei propri meccanismi attentivi. Come osserva Howard Rheingold:

> *L'attenzione è l'alfabetizzazione fondamentale. Ogni secondo che passo online, io prendo decisioni su dove dirigere l'attenzione. Devo dedicare la mia mente a questo commento o a questo titolo? – una domanda alla quale devo rispondere ogni volta che un link interessante attira il mio sguardo. Semplicemente diventando consapevole che la vita online richiede questo tipo di decisioni, questo è stato il mio primo passo nell'apprendere a regolare un filtro fondamentale a ciò che permetto di entrarmi in testa – un filtro che è sotto il mio controllo soltanto se mi esercito a controllarlo. Il secondo livello di decisione è se desidero aprire un tab nel mio browser perchè ho deciso che questo elemento meriterà un po' del mio tempo domani. La terza decisione: creo un bookmark a questo sito perché mi interessa e potrei desiderare di utilizzarlo in un imprecisato futuro? Domare la propria attenzione online inizia con ciò che i meditatori chiamano "mindfulness" – la semplice, controllata consapevolezza di come la nostra attenzione vaga qua e là.*[49]

In sintesi, le richieste che un sistema pone ai meccanismi attentivi dell'utente possono influenzarne in modo molto rilevante l'usabilità. Durante la progettazione di un sistema interattivo si dovranno, quindi, considerare approfonditamente i seguenti aspetti:

- come mantenere costantemente l'attenzione dell'utente sugli elementi desiderati dell'interfaccia;
- come evitare interferenze indesiderate, che sottraggano l'attenzione dell'utente da tali elementi;
- come ridurre al minimo le richieste di attenzione divisa poste dall'interfaccia.

Il primo obiettivo si può conseguire realizzando degli opportuni *indizi attentivi* (*attentional cue*) che attraggano e guidino l'attenzione dell'utente dove si desidera. L'esempio forse più semplice è il puntatore del mouse, che non serve solo a selezionare l'oggetto desiderato, ma anche a portare l'attenzione dell'utente sulla zona dello schermo più rilevante per il compito che sta eseguendo.

Quando il campo visivo è troppo ampio o troppo complesso, si dovranno studiare degli indizi attentivi particolari. Per esempio, la Figura 74 mostra una sofisticata soluzione basata sulla metafora del proiettore teatrale (*spotlight*), per guidare l'attenzione degli spettatori in una sala per presentazioni dotata di 6 grandi schermi. L'area d'interesse dello schermo viene illuminata da un ampio spot circolare, che segue il cursore controllato dal presentatore. Contemporaneamente, le aree rimanenti in tutti gli schermi vengono leggermente oscurate. Quando il cursore si muove, l'area intorno allo spot è meno scura del resto, per permettere a chi guarda di seguirlo con facilità.[50]

---

[48] Diversi sperimentatori sostengono che l'uso continuo del computer stia modificando alcune caratteristiche funzionali della nostra mente, riducendo l'*attention span* e aumentando le capacità di *multitasking*. Questa opinione è tuttavia ancora molto controversa. Si veda, per esempio, il libro di D.Tapscott, *Grown up digital – How the net generation is changing your world*, McGraw-Hill, 2009, che contiene un'analisi molto interessante delle caratteristiche specifiche della cosiddetta *net generation*.

[49] H.Rheingold, *Attention is the Fundamental Literacy*, in http://www.edge.org/q2010/q10_2.html.

[50] A.Khan, J.Mateika, G.Fitzmaurice, G.Kurtenbach, *Spotlight: directing users' attention on large displays*, in *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, 2005, pagg.791-798.



Figura 74. Guida all'attenzione in una sala multi-schermo

Non è necessario che gli indizi attentivi siano di grandi dimensioni. La Figura 75 mostra la homepage del sito di uno studio grafico, di un colore rosso vivo, con scritte bianche. Un minuscolo bottone nero con una freccia permette di entrare nel sito. Non importa che sia molto piccolo: la diversità di colore è così marcata, che l'attenzione dell'utente è immediatamente attratta su questo punto, come nel caso del pallino nero di Figura 72.

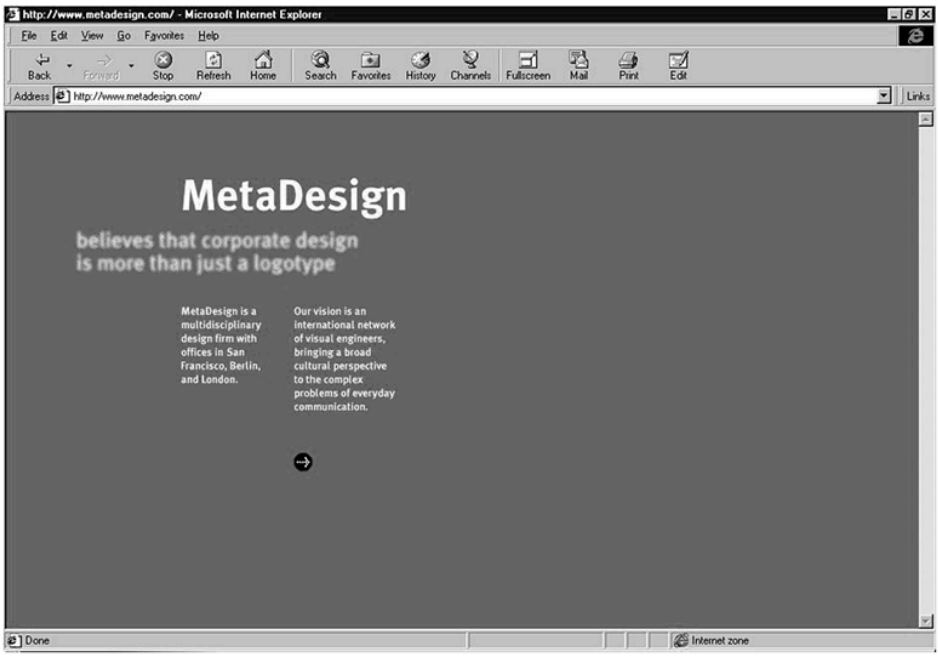


Figura 75. http://www.metadesign.com (circa 1996-2000)

La Figura 76 mostra la tecnica utilizzata nell'iPhone per segnalare la presenza di telefonate, sms e mail non ancora letti: un piccolo cerchio (di colore rosso vivo, e quindi ben visibile) appare sulle icone che permettono di gestire, rispettivamente, telefonate, sms e mail. Il numero nel cerchio segnala quanti sono gli elementi nuovi.



Figura 76. Indizi attentivi nell'iPhone (Apple, 2007)

La memoria

La memoria umana può essere rappresentata con il semplice modello di Figura 77, spesso chiamato *modello modale*.[51]

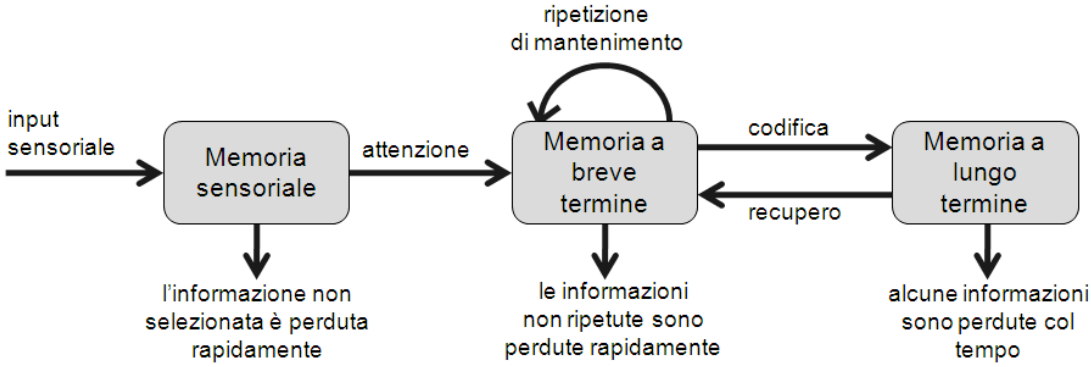


Figura 77. Modello modale della memoria umana

Questo modello, chiaramente ispirato all'information processing, ipotizza l'esistenza di una serie di fasi, attraverso cui l'informazione transita, e una serie di "magazzini" destinati a contenerla. In particolare, i dati sensoriali non ancora elaborati sono inizialmente acquisiti in *memorie sensoriali* (sorta di buffer o registri temporanei associati agli organi di senso, dove i dati grezzi permangono per un tempo molto breve – frazioni di secondo). Vengono quindi selezionati attraverso meccanismi in cui gioca un ruolo importante l'attenzione, e caricati in una memoria *a breve termine (short-term memory, o STM)*, che è ritenuta il componente principale dove avvengono le elaborazioni mentali. La permanenza nella STM è molto breve: essi sono utilizzati per supportare i processi cognitivi in atto, e subito eliminati oppure trasferiti, con un atto consapevole, nella *memoria a lungo termine* (*long-term memory*, LTM) per una ritenzione permanente. In pratica, secondo questo modello, la memoria a breve termine viene utilizzata per contenere i dati in corso di elaborazione; per questo, viene anche chiamata *memoria di lavoro* (*working memory*). La memoria a lungo termine, invece, è il deposito permanente – o comunque di lunghissima durata – di tutto quanto conosciamo.

Questo modello corrisponde bene alla nostra conoscenza intuitiva del funzionamento della memoria umana. Quando dobbiamo telefonare a un amico, cerchiamo prima il numero di telefono nell'agenda. L'informazione visiva transita nella memoria sensoriale del sistema visivo, e poi, trasformata in qualche modo in numeri, viene depositata nella memoria a breve termine. Da qui la preleviamo quando componiamo il numero. Subito dopo, essa viene di solito dimenticata, perchè non ci serve più.

[51] Questo modello è stato proposto da Atkinson e Shiffrin nel 1968, nel loro articolo *Human memory: a proposed system and its control processes*, in W.K.Spence, T.J.Spence (ed.), *The psychology of learning and motivation*, Vol.2: *Advances in learning theory*, Academic Press. Il modello è utile essenzialmente a fini didattici; oggi si ritiene che la memoria a breve termine e la memoria a lungo termine non siano sistemi separati, ma funzioni diverse di uno stesso sistema. La Figura 77 è tratta dal libro di testo *Psicologia*, di P.Gray (Seconda edizione italiana, Zanichelli, 2004), pag.291, al quale ci si può riferire per ulteriori dettagli.

Se dopo avere memorizzato il numero veniamo interrotti da qualcuno che ci chiede un'informazione, usiamo la memoria a breve termine per questo nuovo compito, e scordiamo il numero (*interferenza*). In ogni caso, non possiamo ricordare il numero a lungo: lo dobbiamo usare quasi subito, altrimenti lo scordiamo. Se lo vogliamo ricordare più a lungo, dobbiamo trasferirlo nella memoria a lungo termine. Per questo, dobbiamo ripeterlo mentalmente molte volte (*reiterazione*, o *rehearsal*), o utilizzare altre tecniche che facilitino la memorizzazione.

Gli psicologi hanno effettuato numerosi esperimenti per verificare la correttezza del modello seriale di Figura 77, e per individuare le prestazioni dei diversi sistemi di memoria. Descriveremo solo quelle che sono particolarmente utili a chi si occupa di interazione uomo-macchina.

La memoria a breve termine

Per quanto riguarda la memoria a breve termine, una pietra miliare è il classico articolo *The magical number seven, plus or minus two*, pubblicato da G.A.Miller nel 1956.[52] In questo lavoro, Miller quantifica la *capacità* della STM in 7±2 elementi di informazione, detti *chunk* (il "magico" numero 7 del titolo). Un chunk (letteralmente, "pezzo"), è un raggruppamento di elementi che trattiamo in modo unitario. In sostanza, il massimo numero di raggruppamenti che la memoria a breve termine dell'uomo può contenere è compreso fra 5 e 9, a seconda dei casi.

Definire con precisione che cosa sia un chunk non è facile, ma lo si può capire facendo degli esperimenti. Riferendoci alla Figura 78, consideriamo la sequenza 1, composta da 6 lettere. Essa può essere facilmente ricordata nella memoria a breve termine: possiamo considerare ogni lettera un chunk. La sequenza 2, composta da 9 lettere/chunk, è al limite: la ricordiamo, ma probabilmente con qualche difficoltà. Invece, è molto probabile che non riusciremo a ricordare la sequenza 3, composta da 15 lettere/chunk, perché troppo lunga. Se però consideriamo la sequenza 4, anch'essa composta da 15 lettere, non avremo difficoltà a ricordarla. Le lettere, infatti, non sono fra loro scorrelate, ma costituiscono una coppia di elementi che percepiamo come unitari: il nome WILLIAM e il cognome MCMILLAN. Nello stesso modo, la sequenza 5 non ci crea troppe difficoltà: in questo caso, ogni parola costituisce un singolo chunk, anche se è composta da più lettere. Poiché le parole sono sette, la sequenza non eccede la capacità della STM. Invece, la sequenza 6 (11 chunk/parole scorrelate) è troppo lunga.

La memoria a breve termine può rivelarsi considerevolmente capace, se effettuiamo un opportuno *chunking* (se, cioè, creiamo degli opportuni raggruppamenti dotati di significato). La sequenza 7 in Figura 78, composta di ben 75 lettere e 15 parole, non è difficile da ricordare. Possiamo spezzarla infatti in pochi chunk. Quanti? Dipende da come la scomponiamo. Per esempio, in tre parti: LA PICCOLA VOLPE ROSSA - SALTÒ SUL GROSSO CANE RANDAGIO - E LO FECE RUZZOLARE SUL MARCIAPIEDE.

1. B X M L T D
2. W B V A P R D S N
3. M F B G R T L H J F Z O Z L S
4. W I L L I A M M C M I L L A N
5. GATTO, CANE, DISCO, LATTE, CASA, AUTO, TOPO
6. GATTO, OROLOGIO, DISCO, LATTE, CASA, AUTO, TOPO, ACQUA, MIELE, LIBRO, CANE
7. LA PICCOLA VOLPE ROSSA SALTÒ SUL GROSSO CANE RANDAGIO E LO FECE RUZZOLARE SUL MARCIAPIEDE

Figura 78. Test per il "magico numero sette"

Si può notare che anche per ricordare i numeri di telefono usiamo la tecnica del chunking. Fino alle classiche sette cifre, non abbiamo troppi problemi. Se però dobbiamo ricordare anche i prefissi, le cifre diventano una dozzina, e allora scomponiamo il numero in chunk di più cifre, per esempio così: +39-335-XXXX-YYY.

[52] G.A.Miller, *The magical number seven, plus or minus two: Some limits on our capacity for processing information*, in *Psychological Review*, 63, pagg.81-97 (disponibile anche in rete in http://psychclassics.yorku.ca/Miller/).

Per quanto riguarda la *permanenza* dell'informazione nella STM, altri esperimenti dimostrano che è molto breve (meno di 20 secondi), a meno che l'informazione non venga "ripetuta mentalmente" (*reiterazione*, o *rehearsal*), con un processo che richiede attenzione. Così facendo, siamo in grado di ritenere un'informazione anche molto a lungo. La durata può ridursi ulteriormente, se facciamo del lavoro cognitivo impegnativo che, in qualche modo, "sovrascrive" i contenuti precedenti (*interferenza*).

Un esperimento classico per dimostrare la limitata durata della memoria a breve termine fu condotto da Peterson e Peterson, nel 1959.[53] A un insieme di soggetti veniva presentato un gruppo di tre consonanti (per esempio FDZ), immediatamente seguito da un numero di tre cifre (per esempio 100). Il soggetto doveva iniziare subito a contare all'indietro, per tre, dal numero indicato (100, 97, 94, …). Quando veniva interrotto dallo sperimentatore, doveva rievocare le tre consonanti. Il conteggio all'indietro serviva a impedire che "ripetesse mentalmente" le lettere da ricordare. La Figura 79 mostra la percentuale di rievocazioni corrette, in funzione del tempo trascorso dalla presentazione. Si vede che la capacità di rievocazione degrada molto in fretta: dopo una diecina di secondi, è circa del 30%, dopo una ventina di secondi, del 10%. In altre parole, dopo venti secondi, nel 90% dei casi il contenuto della memoria a breve termine è già perso.

Nella stessa figura, la linea tratteggiata mostra che cosa succedeva quando ai soggetti venivano presentate tre parole, invece di tre consonanti. Le prestazioni erano identiche, a dimostrazione di quanto detto a proposito del chunking: una parola viene trattata come un singolo chunk, proprio come una lettera.

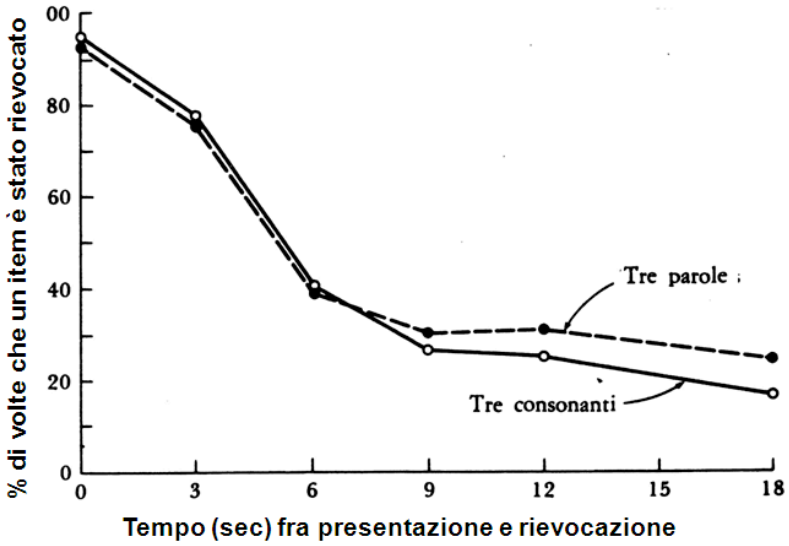


Figura 79. L'esperimento di Peterson e Peterson (1959)

Non sempre i progettisti tengono nella dovuta considerazione le limitazioni della memoria a breve termine: non sono pochi i sistemi che richiedono agli utenti prestazioni che non sono in grado di svolgere. La Figura 80, tratta da PowerPoint 2003, mostra un esempio tipico. Durante la preparazione di una presentazione, se si chiede al sistema di help come si cambia il layout di una slide, si ottene sul video una finestra con le istruzioni richieste. Fin qui, tutto bene. Ma, quando iniziamo a eseguire queste istruzioni, la finestra con la slide si porta in primo piano, coprendo quella con le istruzioni. Per eseguirle, dobbiamo ricordarle, poichè non le possiamo vedere. Questo risulta difficile quando le istruzioni superano i limiti espressi dal "magico numero sette". Come se non bastasse, mentre eseguiamo le istruzioni dobbiamo compiere altre attività cognitive: muovere il mouse, aprire i menu indicati, selezionare le voci corrette,

[53] L.Peterson, M.Peterson, *Short-term retention of individual verbal items*, in *Journal of Experimental Psychology*, 58, 1959, pagg.193-198.

verificare i risultati di queste operazioni, e così via. Tutto ciò ci impedisce di ripetere mentalmente le istruzioni da compiere, per ricordarle. In sostanza, le dimentichiamo prima di averle portate a termine.

Per superare queste difficoltà, dovremmo cliccare alternativamente sulle due finestre, quella con la slide che stiamo preparando e quella con le istruzioni, per rileggerle quando ci servono. Oppure, semplicemente, dovremmo prendere nota delle istruzioni su un foglio di carta. Ma allora a che serve il computer? La soluzione progettuale corretta è ovvia (e infatti è stata adottata nelle successive versioni di PowerPoint): fare in modo che la finestra con le istruzioni rimanga sempre in primo piano, anche mentre si opera sull'altra. Ovviamente, l'utente dovrà poterla spostare dove vuole, per evitare che intralci le operazioni.

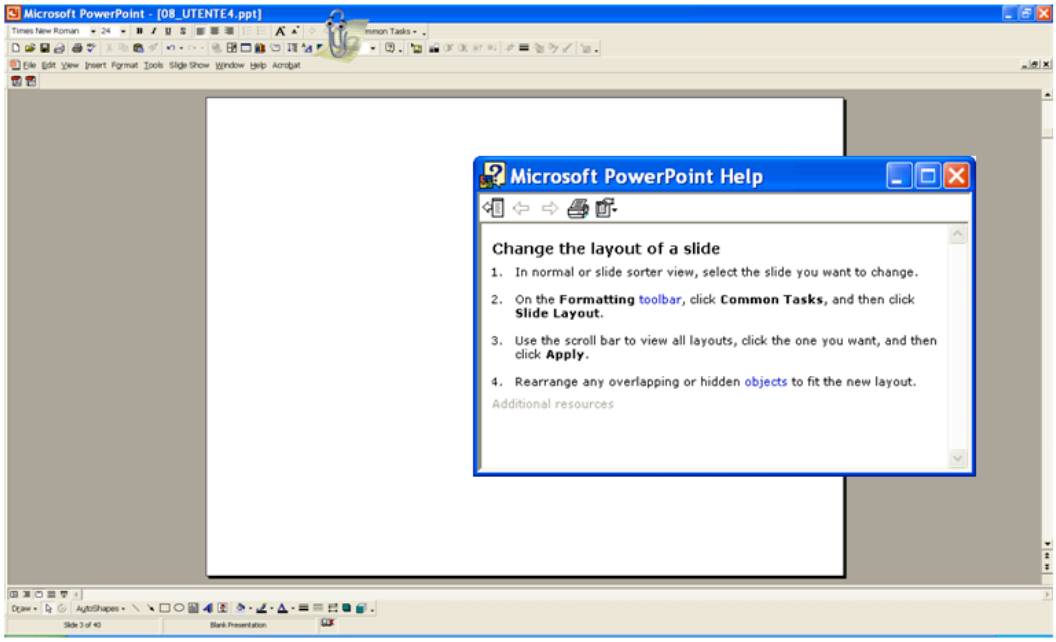


Figura 80. Help in Microsoft PowerPoint 2003

Nei prossimi capitoli vedremo altri esempi di sovraccarico (*overloading*) della memoria a breve termine dell'utente. Sono tutte situazioni tipiche, che si incontrano spesso nei sistemi interattivi, e che dovrebbero essere sempre evitate. La Figura 176, a pag. 208, mostra un esempio simile a quello appena descritto, relativo a un messaggio di help. La Figura 214, a pag.245, mostra un lungo elenco di segnalazioni di errori commessi dall'utente, che questi deve memorizzare prima di poterli correggere. La Figura 210, a pag.242 segnala un altro caso molto frequente: le istruzioni vocali di un call center telefonico, con un numero eccessivo di opzioni. L'utente le deve memorizzare tutte, prima di poter decidere quella che meglio si adatta ai suoi scopi. Anche in questo caso, viene coinvolta la memoria a breve termine.

In sintesi, il progettista di sistemi interattivi dovrà sempre evitare di sovraccaricare la memoria a breve termine dell'utente:

- facilitandogli il chunking, cioè richiedendogli di memorizzare solo elementi che abbiano un significato o che siano familiari, e in numero limitato (7±2);[54]
- minimizzando comunque le richieste di memorizzazione in presenza di altre attività cognitive, per evitare interferenze.

[54] La "regola del 7±2" va applicata solo ai casi in cui si richiede all'utente di *memorizzare* delle informazioni. Una affermazione che a volte fanno i progettisti inesperti è: "allora, tutti i menu devono contenere al massimo 7±2 voci". Questo è, evidentemente, un nonsenso. Quando l'utente seleziona la voce di un menu, questo è tutto visibile, e l'utente non deve memorizzare nulla. Quindi, anche se le voci fossero molto numerose, ciò non implicherebbe alcun sovraccarico della memoria a breve termine.

La memora a lungo termine

La memoria a lungo termine è un sistema molto complesso. Essa ci consente di ricordare le informazioni per molto tempo: potenzialmente, per tutta la vita. Alcuni autori la considerano costituita da due grandi sottosistemi, che svolgono compiti diversi: la *memoria dichiarativa*, e quella *procedurale*. La prima riguarda tutte le conoscenze esprimibili a parole che si hanno sul mondo; la seconda conserva le conoscenze su *come fare* le cose: andare in bicicletta, annodare il nodo della cravatta, giocare a tennis, digitare alla tastiera di un computer. Queste conoscenze non sono verbalizzabili: ci facciamo agevolmente il nodo della cravatta, ma non siamo in grado di descrivere con precisione quali movimenti compiamo. È come se la memoria procedurale ricordasse i movimenti del nostro corpo, ma non le loro descrizioni.

La memoria dichiarativa, a sua volta, può essere decomposta in due sottosistemi: la *memoria episodica* e quella *semantica*. La prima, come suggerisce il nome, registra i fatti che avvengono nel tempo: la caduta del muro di Berlino, i risultati delle ultime elezioni. Se gli episodi riguardano la vita della persona che li ricorda (per esempio dove ha passato le ultime vacanze), si parla di *memoria autobiografica*. Nella memoria semantica, invece, conserviamo le conoscenze generali sul mondo: il prezzo di un oggetto, il nome del Presidente della Repubblica, tutto ciò che abbiamo imparato a scuola.

Un sistema interattivo sollecita tutte le funzioni della memoria dei suoi utenti. Innanzitutto, la memoria semantica, per quanto riguarda ciò che l'utente conosce, dal punto di vista teorico, sull'uso del sistema: il significato delle voci dei menu o dei tasti di scelta rapida, l'esistenza o meno di determinate funzioni. Quindi la memoria procedurale, per le modalità di interazione: per esempio, l'uso della tastiera e del mouse, la selezione della voce di un menu. Infine, la memoria episodica, per quanto riguarda la storia dell'interazione con il sistema. Per esempio, quando utilizziamo un word processor, ricordiamo di avere salvato il file pochi minuti fa, oppure che ieri ne abbiamo cambiato i valori di default.

Gli psicologi studiano i processi di *codifica* (*encoding*) e di *recupero* (*retrieval*) dell'informazione nella/dalla memoria a lungo termine (Figura 77). Per codifica intendiamo i processi che attuiamo per memorizzare qualcosa: quando studiamo un libro, quando "impariamo a memoria" una poesia, oppure quando memorizziamo la password di accesso al nostro computer. Il recupero si riferisce, invece, ai processi che attiviamo per rievocare qualcosa che abbiamo memorizzato. Vediamo brevemente gli aspetti che ci interessano.

Codifica

Le informazioni che ci sono state presentate ripetutamente sono più facile da ricordare. Pertanto, la tecnica più elementare per memorizzare un'informazione nella memoria a lungo termine consiste nel "ripeterla mentalmente" una o più volte, prestando attenzione al suo significato, come facciamo quando ripassiamo una lezione (*reiterazione*, o *rehearsal*). Non bisogna confondere la semplice ripetizione con la reiterazione. La ripetizione si riferisce al fatto che un elemento viene incontrato più di una volta. Reiterazione, invece, significa che esso viene "pensato ripetutamente", in un processo volontario che richiede attenzione. Per esempio, quando sul giornale incontriamo più volte il nome di un ministro, è ripetizione. Quando lo ripetiamo mentalmente per ricordarcelo, è reiterazione.

Sappiamo che il recupero di un'informazione dalla memoria è facilitato dalla presenza di *associazioni* fra l'informazione cercata e altre già note. Per esempio, per ricordare il nome di Bill Gates potremmo associarlo all'immagine del cancello del nostro giardino, perché Gates in inglese significa "cancelli". Possiamo anche inventare degli *ausilii mnemonici* (*memory aids*) più sofisticati, che utilizzano associazioni di vario tipo. Per esempio, per ricordare la suddivisione delle Alpi, si utilizza spesso la frase:

MA CON GRAN PENA LE RECA GIÙ

composta dalle sillabe iniziali dei nomi da ricordare:

MArittime, COzie, GRAie, PENnine, LEpontine, REtiche, CArniche, GIUlie.

Il progettista di sistemi interattivi può suggerire in molti modi queste associazioni all'utente, affinchè ricordi meglio i comandi del sistema. Per esempio, in Microsoft Office 2003, i tasti di scelta rapida sono definiti utilizzando le iniziali dei comandi cui si riferiscono: Print ↔ CTRL P; Save ↔ CTRL S; Open ↔ CTRL O (Figura 81).[55]

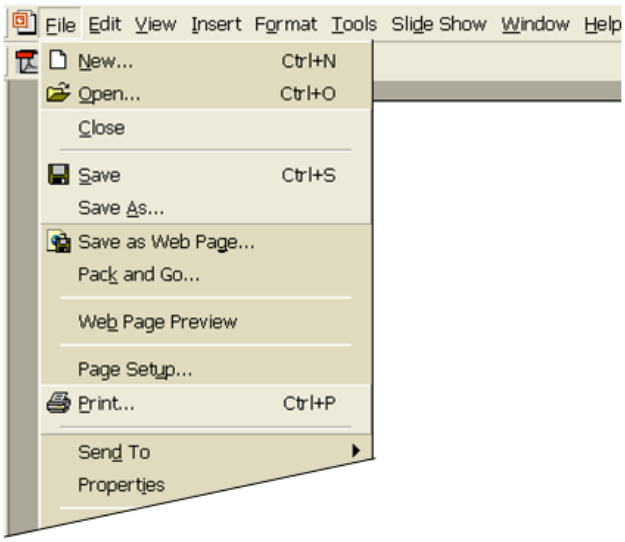


Figura 81. Ausilii mnemonici in Microsoft Office 2003

I metodi per memorizzare le informazioni si chiamano *mnemotecniche*. Gli esempi che abbiamo visto sono molto semplici, ma ne sono stati elaborati di assai più sofisticati, che però richiedono uno specifico addestramento. Nell'antichità, gli oratori utilizzavano mnemotecniche raffinate per ricordare i discorsi da pronunciare senza l'ausilio di un testo scritto, (la carta è invenzione relativamente recente). Esse sfruttavano il fatto che ricordiamo meglio le informazioni associate a immagini visive o a storie. Per esempio, *Ad Herennium*, un testo anonimo dell'85 a.C. circa, consiglia di associare alle informazioni da ricordare immagini di natura eccezionale (*imagines agentes*):

> *Quando, nella vita di ogni giorno, vediamo cose meschine, usuali, banali, generalmente non riusciamo a ricordarle, perché la mente non ne riceve nessuno stimolo nuovo o inconsueto. Ma se vediamo o udiamo qualcosa di eccezionalmente basso, vergognoso, inconsueto, grande, incredibile o ridicolo, siamo soliti ricordarcene a lungo. […] Dobbiamo, dunque, fissare immagini di qualità tale che aderiscano il più a lungo possibile nella memoria. E lo faremo fissando somiglianze quanto più possibile straordinarie; se fissiamo immagini che siano non molte o vaghe, ma efficaci (imagines agentes); se assegnamo ad esse eccezionale bellezza o bruttezza singolare; se adorniamo alcune di esse ad esempio con corone o manti di porpora per rendere più evidente la somiglianza, o se le sfiguriamo in qualche modo, ad esempio introducendone una macchiata di sangue o imbrattata di fango o sporca di tinta rossa, così che il suo aspetto sia più impressionante; oppure attribuendo alle immagini qualcosa di ridicolo, poiché anche questo ci permette di ricordarle più facilmente.*

Un metodo molto diffuso nell'antichità per memorizzare un discorso consisteva nell'associarne i diversi passaggi alle stanze di un palazzo, e agli oggetti che contenevano (meglio se rappresentati da imagines agentes, nel senso descritto sopra). L'oratore, quindi, poteva percorrere con la mente le stanze in un ordine prefissato, ritrovando gli oggetti contenuti e recuperando così, via via, i passaggi del discorso ad essi associati. Come spiega Cicerone nel *De Oratore,* il suo trattato sull'arte oratoria:

[55] Ma come si giustifica l'associazione Paste ↔ CTRL V?

Le persone desiderose di addestrare la memoria devono scegliere alcuni luoghi e formarsi immagini mentali delle cose che desiderano ricordare, e collocare quelle immagini in quei luoghi, in modo che l'ordine dei luoghi garantisca l'ordine delle cose, le immagini delle cose denotino le cose stesse, e noi possiamo utilizzare i luoghi e le immagini rispettivamente come la tavoletta cerata e le lettere scritte su di essa.

Questa "tecnica dei luoghi" fu usata fino al Rinascimento in numerose varianti. La Figura 82 riporta un'illustrazione di un testo cinquecentesco di mnemotecnica. In questo caso, si utilizza un'abbazia con gli edifici annessi, raffigurati nell'immagine a sinistra. A destra sono rappresentati gli oggetti da collocare nei diversi luoghi dell'abbazia (il cortile, la biblioteca, la cappella), che fungeranno da ancore mnemoniche per i diversi passaggi del discorso da ricordare. Ogni quinto posto è contrassegnato da una mano, e ogni decimo da una croce, con un'ovvia associazione alle dita delle due mani, con le quali ci si aiutava nell'enumerazione.[56]

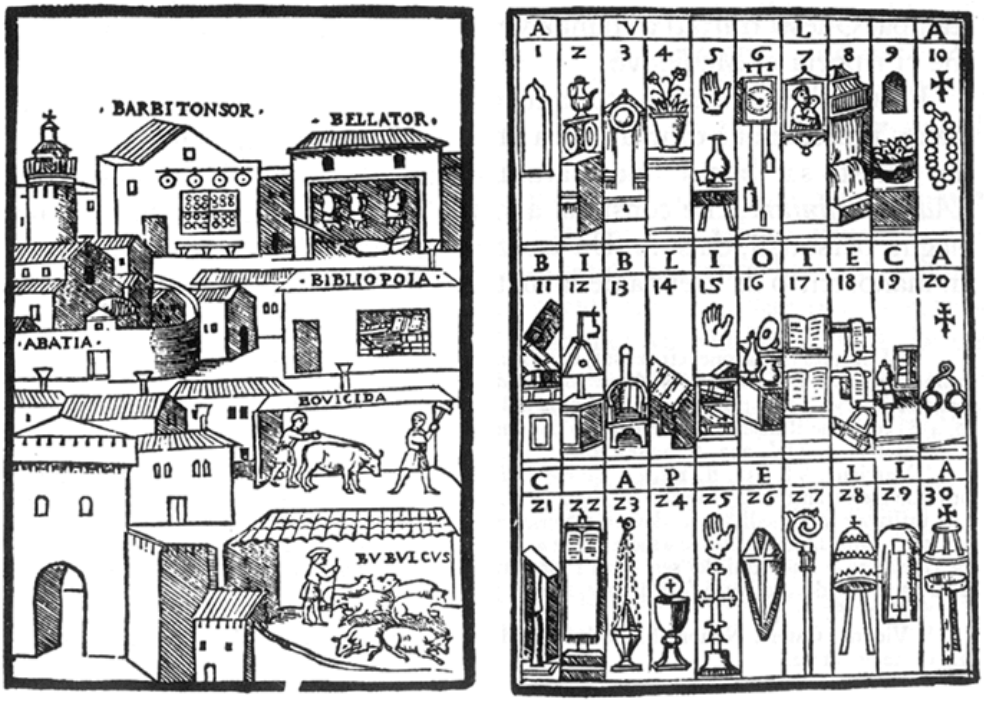


Figura 82. Johannes Romberch, *Congestorium artificiose memorie* (Venezia, 1533)

Oggi i discorsi sono fatti con l'ausilio del computer, e non abbiamo più bisogno di imparare queste tecniche complesse. Tuttavia, possiamo ancora seguire i consigli degli antichi, e utilizzare le *imagines agentes*. Per esempio, il nome del WWF è strettamente associato all'immagine del panda, animale raro e quindi insolito, assurto a simbolo del movimento ambientalista (Figura 83).

[56] Un libro affascinante sulle mnemotecniche antiche è *The Art of Memory*, di F.A.Yates, 1966. Le citazioni da *Ad Herennium* e da Cicerone, e la Figura 82, sono state tratte dalla traduzione italiana, pubblicata da Einaudi nel 1972, col titolo *L'arte della memoria* (pag.11, 3 e 99 rispettivamente).



Figura 83. L'immagine del panda, simbolo del WWF

Recupero

Per quanto riguarda il recupero delle informazioni dalla memoria a lungo termine, un aspetto importante è la distinzione fra riconoscimento e rievocazione. *Rievocare* (*recall*) significa estrarre dalla memoria un'informazione precedentemente memorizzata: "qual è la capitale del Nicaragua?", "Quale film hai visto ieri sera?", "Come si chiama il Presidente degli Stati Uniti?". *Riconoscere* (*recognition*), invece, significa individuare, fra diverse alternative, quella (o quelle) che fanno al caso nostro: "La capitale del Nicaragua è San Josè, Tegucicalpa o Managua?", "Il Presidente degli Stati Uniti è Clinton, Bush o Obama?"

Gli esperimenti dimostrano che è più facile riconoscere che rievocare. Supponiamo di presentare a un soggetto una lista di parole fra loro scorrelate. Se, in seguito, gli chiediamo di elencare quelle che si ricorda, egli ne rievocherà un certo numero, che diminuirà con l'aumentare del tempo intercorso fra presentazione e rievocazione. Dopo pochi minuti, ne elencherà parecchie, dopo una settimana, pochissime. Se invece, dopo la presentazione della lista, gliene presentiamo una seconda, chiedendogli di riconoscere le parole presenti anche nella prima, la percentuale di parole riconosciute sarà più alta di quelle rievocate. Anche la capacità di riconoscimento degrada col tempo, ma in modo più lento della capacità di rievocazione. La Figura 84 mostra i risultati di un tipico esperimento di confronto fra riconoscimento e rievocazione. In questo caso, dopo due giorni, i soggetti riconoscevano ancora il 75% delle parole, ma non erano in grado di rievocarne praticamente nessuna.

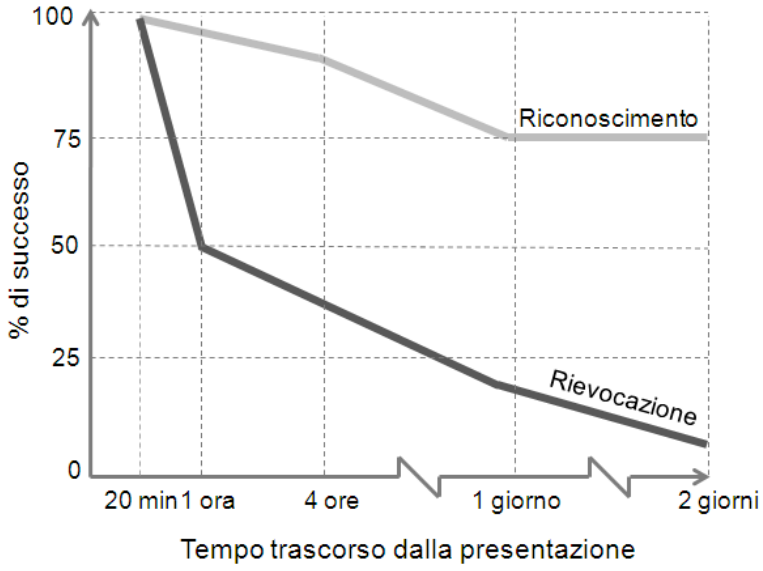


Figura 84. Riconoscimento e rievocazione (Luh, 1922)

È molto importante che il progettista di sistemi interattivi tenga sempre presente la differenza fra rievocazione e riconoscimento, e il fatto che le prestazioni dell'utente, nel secondo caso, sono migliori. Quando si comunicava con i computer esclusivamente con il paradigma "scrivi e leggi" (capitolo 2, pag.33), l'utente era costretto a fare continuamente esercizio di rievocazione, per ricordare i nomi dei comandi del sistema. Con l'introduzione dei terminali video e poi dei personal computer e l'adozione dei paradigmi "indica e compila" (pag.35) e manipolazione diretta (pag.36), all'utente viene solo chiesto di riconoscere il comando desiderato all'interno di un gruppo di alternative possibili, e non di rievocarne il nome. Le alternative sono presentate in vari modi: come voci di un *menu*, come icone in una *barra di strumenti*, come *bottoni* in una pulsantiera. Nell'adventure game di Figura 85, i comandi possibili (Examine, Open, Close, Speak, Operate, Go, Hit, Consume) sono elencati sul video, e l'utente non deve compiere alcuno sforzo di rievocazione.

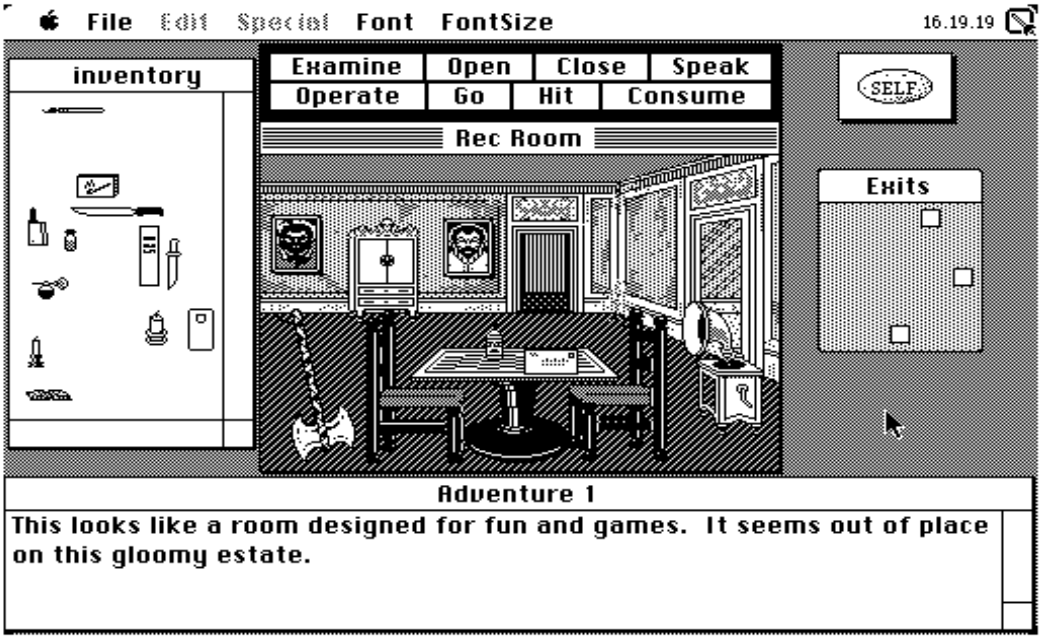


Figura 85. Pulsantiera dei comandi in Uninvited (computer game per Macintosh, della Mindscape, 1986)

In Microsoft Office 2003 (Figura 81), esiste un modo alternativo di selezionare rapidamente le voci, digitando, per ogni menu, una sola lettera che la identifica univocamente. Per esempio, per selezionare Open all'interno del menu File, si digitano Alt e F (per aprire il menu File), e quindi O. Oppure, per creare un nuovo file, si digita Alt seguito da F e da N. L'utente non deve rievocare le lettere da digitare, ma le deve semplicemente riconoscere. Infatti, sono evidenziate con una sottolineatura nelle voci dei menu (nel nostro caso, File e Open, vedi figura). Le difficoltà sorgono, evidentemente, quando le lettere più adatte a rappresentare una voce sono già utilizzate per altre voci.

I sistemi più evoluti utilizzano spesso, come in questo caso, sia il riconoscimento che la rievocazione: un comando può essere eseguito selezionandolo da un menu di alternative (riconoscimento), oppure digitando una (breve) sequenza di *tasti di scelta rapida*, che l'utente deve conoscere (rievocazione). Questa seconda modalità viene utilizzata dagli utenti più esperti, che desiderano un'interazione più veloce. Generalizzando, possiamo dire, infatti, che l'interazione basata sul riconoscimento è più facile ma più lenta, mentre quella basata sulla rievocazione è più rapida, ma richiede che l'utente sia opportunamente addestrato.

La visione

Anche per quanto riguarda la visione umana, ci limiteremo a pochi cenni, relativi agli argomenti di maggiore rilevanza per l'interazione uomo-macchina.

Quando osserviamo una scena, il nostro sguardo la esplora seguendo percorsi complessi e irregolari, soffermandosi su quegli elementi cui prestiamo attenzione. Durante l'esplorazione, l'asse visivo del nostro occhio si sposta seguendo percorsi che possono essere tracciati dagli apparati di *eye-tracking*. Questi dispositivi mostrano che il movimento dei nostri occhi è molto irregolare: lo sguardo si fissa per un certo tempo su un determinato punto, per acquisire l'informazione visiva (*fissazione*), e quindi si sposta su un altro punto, con un movimento rapidissimo (chiamato *saccade*) durante il quale l'occhio è cieco. In media sono eseguite tre-quattro fissazioni al secondo. Approfondiremo questo tema nel capitolo 12 (pag.274), quando tratteremo della grafica, e ulteriormente nel capitolo 13 (pag.287), a proposito del testo.

L'informazione visiva viene "proiettata" attraverso il *cristallino*, capovolta, sulla *retina*, una membrana che riveste la parete interna del globo oculare (Figura 86). Essa è rilevata da cellule sensibili ai raggi luminosi (*fotorecettori*), poste in grande quantità sulla *retina*, che la trasformano in segnali elettrici inviati al cervello. Queste cellule sono di due tipi: i *coni* (*cones*), particolarmente sensibili al colore, che si addensano nell'area centrale della retina (*fovea*), sull'asse visivo, e i *bastoncelli* (*rods*), molto sensibili alla luce anche di bassa intensità (visione notturna), ma non al colore, che si addensano nelle aree periferiche della retina, lontane dall'asse visivo. La sensibilità dei bastoncelli alle variazioni di luce li rende molto sensibili al movimento.

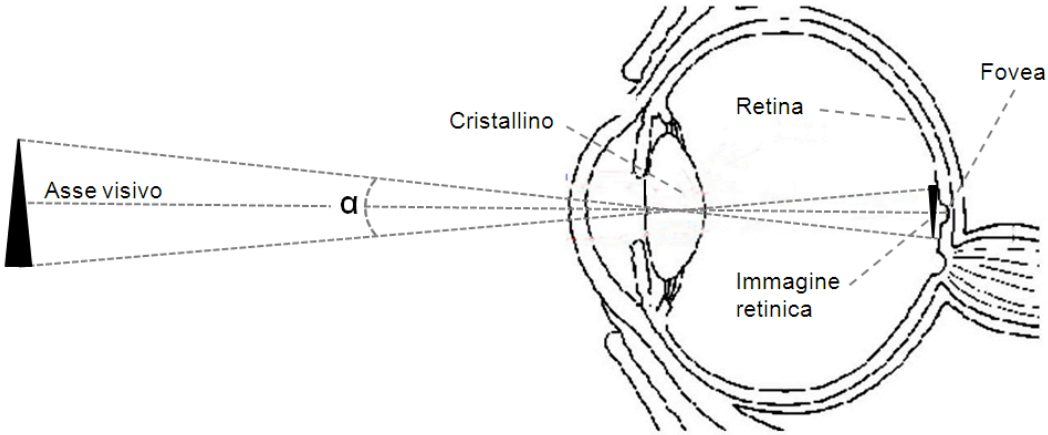


Figura 86. L'occhio umano

La Figura 87 mostra l'andamento della densità dei coni e dei bastoncelli in funzione dell'angolo visuale dalla fovea. Questa disposizione dei fotorecettori fa sì che noi distinguiamo meglio i dettagli e i colori degli oggetti quando li fissiamo direttamente, al centro dell'asse visivo (*visione foveale*), e siamo molto sensibili ai movimenti che percepiamo, come si dice "con la coda dell'occhio" (*visione periferica*). La visione periferica, però, non distingue dettagli né colori, perché è prodotta solo dai bastoncelli. Il grafico mostra, anche, che il campo di visione a risoluzione elevata, in cui la densità dei fotorecettori è alta, è abbastanza ristretto.[57]

[57] Circa 50° complessivamente, più o meno quello di un obiettivo fotografico normale. Per fissare le idee, si consideri che l'angolo sotto cui vediamo il nostro dito indice, quando stendiamo completamente il braccio, è di circa 1 grado.

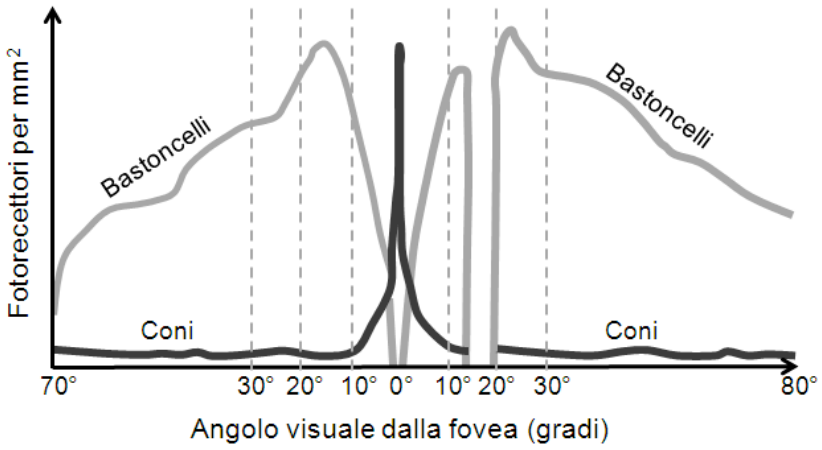


Figura 87. Distribuzione dei fotorecettori sulla retina

Quando fissiamo il puntatore del mouse sullo schermo di un computer, la nostra visione foveale ci consente di vedere in dettaglio solo il suo immediato intorno (qualche grado). I nostri occhi lo seguono con continuità, e prestiamo poca – o nessuna – attenzione a quanto gli è lontano. In questo modo, potremmo non accorgerci di cambiamenti che occorrono in altre aree del monitor. Quando questi avvengono, e l'utente li deve considerare, la sua attenzione dovrà essere "catturata" con mezzi opportuni. Per esempio, con messaggi lampeggianti, che possano essere percepiti dai bastoncelli alla periferia della retina. Nel Mac, quando un'applicazione ha qualcosa da segnalare, la sua icona si mette a "saltellare" nel *dock.*[58] Il movimento viene percepito attraverso la visione periferica dell'utente, che sposta la sua attenzione sull'icona ed esamina il messaggio.

Si dice *acuità visiva* (*visual acuity*) la capacità dell'occhio di distinguere due punti vicini. Essa si misura considerando l'angolo minimo α sotto cui devono essere visti perché l'occhio li percepisca separatamente. Se tale angolo vale 1'[59], le loro immagini si trovano sulla retina a una distanza di 5 millesimi di millimetro e stimolano due fotorecettori non contigui, condizione indispensabile perché siano visti come distinti da un occhio normale. Più precisamente, l'acuità visiva si misura dando il valore reciproco dell'angolo α, cioè $1/\alpha$. Per esempio, se tale angolo è di 1', l'acuità visiva è pari a 1/1, cioè 10/10. Se l'angolo è di 2', l'acuità visiva è pari a ½, cioè 5/10.[60] L'acuità visiva è massima in corrispondenza della fovea centrale, e diminuisce verso la periferia. Questo è il motivo per cui, per distinguere bene i particolari di una figura, la dobbiamo fissare direttamente.

L'acuità visiva è molto variabile da soggetto a soggetto, e tende a diminuire con l'avanzare dell'età. Secondo la OMS (Organizzazione Mondiale della Sanità), la *cecità* (*blindness*) corrisponde a un'acuità visiva inferiore ai 3/60 (anche con una eventuale correzione, per esempio occhiali). L'acuità compresa fra 3/60 e 6/18, sempre con eventuale correzione, è definita *ipovisione* (*low vision*). Secondo un rapporto della stessa OMS, nel 2006 gli ipovedenti erano ben 269 milioni in tutto il mondo, i ciechi 45 milioni. In totale, 314 milioni di persone: quasi il 5% della popolazione del pianeta.[61] La maggior parte sono anziani; le donne sono più a rischio degli uomini, a tutte le età. In sostanza, la percentuale delle persone che hanno gravi problemi di visione (*visually impaired*) è molto significativa.

Le differenze fra le persone non riguardano solo l'acuità visiva, ma anche la capacità di distinguere correttamente i colori. Nell'occhio umano, come abbiamo già detto, la percezione dei colori è affidata ai coni della retina. Queste

[58] Nel Mac, il *dock* è quella struttura (di solito posta nella parte inferiore dello schermo) nella quale vengono allineate le icone delle applicazioni o i folder utilizzati di frequente.
[59] 1° (grado) = 60' (minuti primi) = 3600'' (secondi).
[60] Il valore 10/10 non è l'acuità visiva massima, come erroneamente si pensa. Esso rappresenta il valore *minimo* per una visione da considerarsi soggettivamente normale e accettabile. L'acuità visiva può infatti essere ben superiore a 10/10.
[61] Si veda il documento del 2007, *Global Initiative for the Elimination of Avoidable Blindness, Action Plan 2006-2011*, della World Healt Organization, reperibile in http://www.who.int/blindness/Vision2020_report.pdf .

cellule sono di tre tipi, ciascuno sensibile alla luce di un certo intervallo di lunghezze d'onda. I tre tipi di coni sono chiamati R (*Red*), G (*Green*) e B (*Blue*), secondo la sensazione cromatica che si sperimenta quando un particolare tipo è più attivo degli altri. La Figura 88 mostra il grafico della sensibilità dei tre tipi di coni, in funzione delle lunghezze d'onda dello spettro visibile. Si vede, per esempio, che i coni R sono quelli più sensibili alla luce di lunghezza d'onda elevata, con una sensibilità massima attorno ai 570 nm (nanometri), corrispondente al giallo.[62]

In tal modo, quando un fascio di luce colorata colpisce la retina, le sue componenti cromatiche vengono percepite dai coni ad esse più sensibili, che invieranno in risposta segnali elettrici di intensità diversa al cervello. La Figura 88 mostra come un fascio di luce azzurra (di lunghezza d'onda attorno ai 500 nm) e uno arancione (di 600 nm) producono risposte di intensità diversa da ciascun tipo di coni. Per esempio, la luce azzurra produce nei coni G una risposta superiore a quella dei coni R, e molto superiore a quella dei coni B. Quella arancione, invece, eccita i coni R più dei coni G, e i coni B quasi per nulla. Questi segnali elettrici di intensità diversa, elaborati dal cervello, ci fanno vedere due colori differenti.

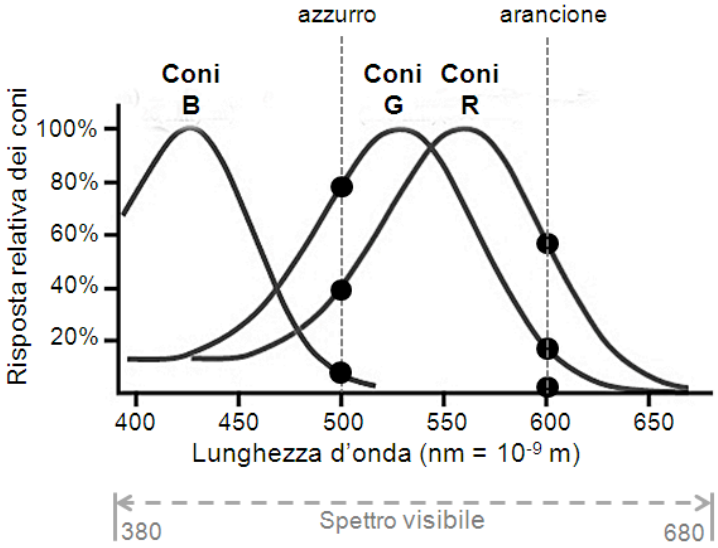


Figura 88. Sensibilità dei coni R, G, B dell'occhio umano

Quando i coni di un certo tipo sono difettosi, o mancanti, la visione del colore viene alterata, perché la componente cromatica ad essi associata non viene percepita. Se, per esempio, non funzionano i coni di tipo R, l'occhio non vede la componente rossa, e i colori che la contengono risulteranno alterati. Quest'alterazione nella percezione dei colori si chiama *cecità cromatica* (*color blindness*), o anche *daltonismo*.[63] I tipi di daltonismo sono diversi, a seconda dei tipi di coni difettosi. Quello più frequente è causato dal difettoso funzionamento dei coni G, e determina una riduzione della sensibilità al verde. La conseguenza è una difficoltà a distinguere le differenze cromatiche nella regione rosso-arancio-giallo-verde dello spettro. Tale anomalia è presente in un gran numero di persone: secondo alcune stime, negli Stati Uniti circa il 7% degli uomini e lo 0,4% delle donne.[64]

La diffusione della cecità cromatica richiede grande attenzione nella progettazione delle interfacce visive dei sistemi interattivi. È necessario evitare che le informazioni rilevanti siano veicolate esclusivamente attraverso il colore. Infatti, gli utenti che non possono distinguere i colori utilizzati per comunicare queste informazioni non sarebbero in grado di comprenderle. In particolare, non bisogna mai assumere che gli utenti sappiano distinguere il rosso dal verde, perché

[62] In effetti, nonostante vengano impropriamente associati al rosso, i coni R non hanno la massima sensibilità su questo colore, che si trova nella parte destra nello spettro.
[63] Il termine deriva dal nome del chimico inglese John Dalton, che descrisse per primo (nel 1794) questo fenomeno, di cui egli stesso era affetto. Più precisamente, il termine si riferisce alla varietà di cecità cromatica più diffusa, chiamata deuteranopia.
[64] Per link a statistiche e a dettagli sul daltonismo, si veda la voce color blindness di Wikipedia.

questa è la forma di cecità cromatica più diffusa (invece, la grande maggioranza delle persone riesce a distinguere correttamente il giallo e il blu). Alcuni esempi a questo proposito sono discussi nel capitolo 12, dedicato alla grafica (Figura 250 e Figura 254).

Quando poi consideriamo anche gli utenti ipovedenti o ciechi, i problemi si fanno molto più complessi, data anche la varietà di patologie esistenti, e le tecniche utilizzabili. In questo libro di carattere introduttivo non è possibile entrare nei dettagli, per i quali rimandiamo ai testi specializzati.

I meccanismi della visione umana hanno notevoli implicazioni sulla progettazione dei sistemi interattivi. Alla progettazione grafica sarà interamente dedicato il capitolo 12 e, per quanto riguarda gli aspetti relativi alla leggibilità dei testi, parte del capitolo 13.

Il sistema motorio

Per quanto riguarda il processore motorio, ricorderemo qui soltanto due temi importanti: l'apprendimento motorio, e la legge di Fitts.

Apprendimento motorio

Con questo termine s'intende l'apprendimento di particolari sequenze di movimento che coinvolgono l'apparato muscolare, come giocare a golf, suonare il piano, pronunciare parole in una lingua straniera. L'utilizzo dei sistemi interattivi, spesso, presuppone qualche forma di apprendimento motorio, che richiede un particolare coordinamento fra occhio e mano (*eye-hand coordination*). Per esempio, per l'uso di device di manipolazione diretta, quali mouse, touchpad, trackball, tavoletta grafica, joystick. In qualche caso l'apprendimento può richiedere molto tempo, come nei programmi di grafica, che richiedono molta manualità e un perfetto coordinamento occhio-mano, e i computer game d'azione, in cui l'eccellenza nel gioco si raggiunge solo dopo un lungo esercizio. Ovviamente, non tutti i sistemi sono impegnativi come un computer game. Anche operazioni apparentemente molto semplici, per esempio il doppio-click del mouse, richiedono apprendimento. Tuttavia, anche i sistemi più semplici richiedono qualche tipo di abilità, che l'utente non è sempre in grado di sviluppare agevolmente. Per esempio, come abbiamo già osservato, gli anziani possono avere difficoltà a svolgere compiti manuali anche elementari, come l'interazione con un telefono cellulare o con un mouse. I sistemi interattivi destinati a un'utenza generale dovrebbero pertanto possedere un'elevata learnability (pag.69) anche per quanto riguarda questi aspetti. Quando il sistema richiede comunque l'apprendimento di abilità motorie particolari, dovrebbe essere sempre permesso di adattarne i parametri alle esigenze individuali degli utilizzatori. Per esempio, nel caso del mouse, modificando la velocità del doppio clic, o addirittura eliminando certe funzioni, come lo zoom effettuato con la rotellina superiore (Figura 89).

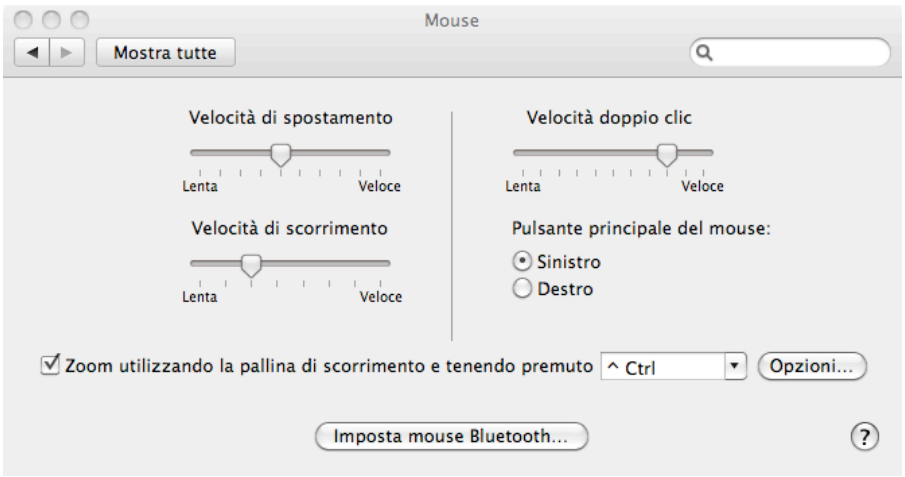


Figura 89. Personalizzazione del mouse (MacOS Finder 10.6, 2009)

Come ogni tipo di apprendimento, anche quello motorio procede per gradi. Inizialmente, eseguiamo i movimenti in modo approssimato, incerto, commettendo frequenti errori; queste prime prove ci forniscono tuttavia dei feedback che ci permettono di correggerci. Con la ripetizione, il movimento si fa più fluido e preciso, e facciamo meno errori. Via via che procediamo nell'addestramento, le nostre prestazioni migliorano. Un elemento essenziale in questo processo è la presenza di feedback. Esso ci permette di confrontare le intenzioni con i risultati, e di correggere il movimento di conseguenza. Senza feedback le nostre prestazioni non migliorano, perché operiamo, per così dire, alla cieca: secondo il modello di Norman (pag.58), il golfo della valutazione è troppo ampio, e non c'è apprendimento.

Il feedback può essere *qualitativo*, come per esempio quando impariamo il movimento che ci permette di ingrandire o rimpicciolire una fotografia dello schermo multi-touch dell'iPhone (Figura 29 a pag.41). In questo caso, in corrispondenza del movimento delle nostre dita, l'immagine si modifica in modo continuo. Quando riteniamo che l'immagine abbia raggiunto – *più o meno* – la dimensione desiderata, ci fermiamo. Oppure, il feedback può essere *quantitativo*, come quando impostiamo l'ora della sveglia, ancora sull'iPhone (Figura 55 a pag.68). In questo secondo esempio, selezioniamo l'ora della sveglia strisciando il dito sulla rotella delle ore, che ruota in modo corrispondente al movimento. A differenza di prima, però, la rotella ci mostra in ogni istante ora e minuti selezionati. Siamo così in grado di valutare con precisione *quanto* manca al raggiungimento dell'obiettivo, e di graduare il nostro movimento, in maniera più fine. Quando il feedback è quantitativo, come in questo caso, l'apprendimento è solitamente più preciso, e facciamo meno errori.

Nei casi specifici, possiamo quantificare i miglioramenti prodotti dall'apprendimento, per esempio contando gli errori commessi in funzione del numero di prove. Quante prove servono per imparare a mandare una pallina in buca con una mazza da golf? Quanti tasti occorre premere per imparare a scrivere al computer raggiungendo certe prestazioni? Queste valutazioni si possono effettuare con degli opportuni esperimenti. In generale, si trova che il tempo necessario per effettuare un compito motorio diminuisce con la pratica, secondo una legge di potenza: la cosiddetta *legge di potenza della pratica* (*Power Law of Practice*).

Questa legge ci dice che, se T_1 è il tempo impiegato per eseguire la prima prova, T_n è il tempo impiegato per effettuare l'n-esima prova

$$T_n = T_1 n^{-\alpha}$$

dove α è una costante che dipende dal tipo di compito, il cui valore si trova tipicamente nell'intervallo 0.2 - 0.6.

La curva rappresentata da questa equazione è quella di Figura 90. In pratica, essa ci dice che il tempo necessario per effettuare un compito (per esempio, annodarsi la cravatta), all'inizio migliora sensibilmente ad ogni nuova prova. Via via che il numero delle prove aumenta, i miglioramenti si fanno sempre più modesti. Quando sappiamo fare il nodo velocemente, ulteriori miglioramenti richiedono moltissimo esercizio. Questa legge esprime in modo preciso un fatto ben noto, che tutti abbiamo sperimentato. È relativamente facile imparare a tirare una pallina con una mazza da golf, e i progressi durante le prime ore di lezione con un istruttore possono essere molto incoraggianti, ma poi i miglioramenti diventano più lenti. E per diventare campione del mondo è necessario esercitarsi molto, molto a lungo.

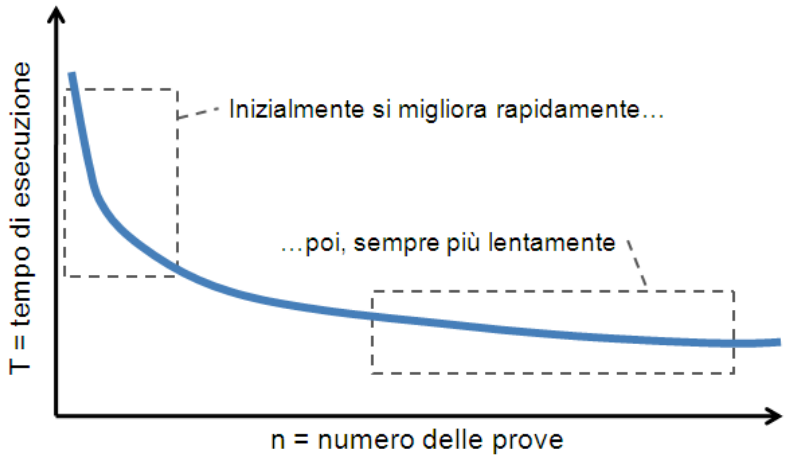


Figura 90. Legge di potenza della pratica

La curva di Figura 90, valutata per un particolare compito motorio su un campione significativo di utenti, visualizza, in pratica, la *curva di apprendiment media* (*learning curve*) di quel compito. Se la curva scende molto in fretta, avvicinandosi alle ascisse quasi subito, significa che il compito è molto facile da apprendere. Se la curva scende lentamente, significa che, per il campione di utenti con i quali è stato condotto l'esperimento, il compito è difficile.

L'interazione con i sistemi interattivi spesso si sviluppa soprattutto sul piano cognitivo, senza richiedere all'utente abilità manuali particolarmente complesse. Ci si limita di solito alla digitazione di caratteri con una tastiera (reale o virtuale) di varie dimensioni, e al puntamento su uno schermo, direttamente col dito (touch screen) o mediante device di manipolazione indiretta (mouse, touchpad, o simili). La facilità di apprendimento dell'uso di questi device, proprio per la loro grande diffusione, è un elemento fondamentale della usabilità complessiva del sistema.

La legge di Fitts

Una delle azioni più frequenti che compie chi interagisce con un sistema è quella di spostare il dito (o un sostituto del dito, come il puntatore del mouse) su un bersaglio. Per esempio, quando muoviamo un dito per premere un tasto della tastiera o un bottone su un touch screen, oppure quando muoviamo il puntatore del mouse sulla voce di un menu per selezionarla, e così via. In tutti questi casi, possiamo utilizzare un modello matematico per prevedere il tempo T necessario per raggiungere il bersaglio, in funzione della sua dimensione S e della distanza D dal punto di partenza.

Questo modello è stato proposto dallo psicologo americano Paul Fitts nel 1954, ed è quindi noto come *legge di Fitts*.[65] ci fornisce. Essa stabilisce che:

$$T = a + b \log_2 (D/S + 1)$$

Dove:

T è il tempo medio necessario per effettuare il movimento;

D è la distanza fra il punto di partenza e il centro del bersaglio (Figura 91);

[65] P.Fitts, *The information capacity of the human motor system in controlling the amplitude of movement*, pubblicato in *Journal of Experimental Psychology*, vol. 47, n.6, 1954, pagg.381-391. (Ripubblicato in *Journal of Experimental Psychology: General*, vol.121,n.3, 1992, pagg.262-269.

S è la dimensione (*size*) del bersaglio, misurata lungo l'asse del movimento. S può anche essere considerato il margine di tolleranza sulla posizione finale, poiché questa deve cadere nell'intervallo ±S/2 dal centro del bersaglio (Figura 91);

a e b sono due costanti che dipendono dallo strumento di puntamento utilizzato: dito, mouse, touchpad, trackball, e così via, e devono essere ricavate sperimentalmente.
In particolare, a rappresenta il tempo necessario per mettere in movimento/fermare il device (per la mano libera è nullo).

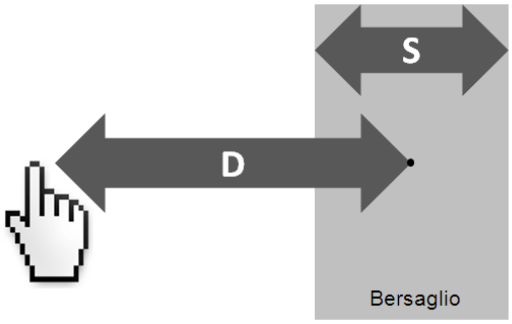


Figura 91. I parametri della legge di Fitts

In sostanza, la legge di Fitts dice che tempo T necessario per muovere la mano su un bersaglio di dimensione S a distanza D dipende dalla precisione relativa richiesta (rapporto D/S). Così, per esempio, il tempo per raggiungere un bersaglio di 2 cm da una distanza di 20 cm è uguale al tempo necessario per raggiungere un bersaglio di 1 cm da una distanza di 10 cm, perché 2/20 = 1/10. Se la distanza è grande e il bersaglio piccolo, ci vorrà un tempo maggiore di quello necessario per raggiungere un bersaglio grande da una distanza breve.

Consideriamo la Figura 92a, e supponiamo che il tempo necessario per spostare il dito sul bottone dalla distanza indicata dalla freccia sia T_0. Se desideriamo ridurre questo tempo di un certo ΔT, per esempio del 30%, potremo muovere il bottone più vicino al punto di partenza (soluzione b in figura), oppure, lasciando invariata la distanza, ingrandire opportunamente il bottone (soluzione c).[66]

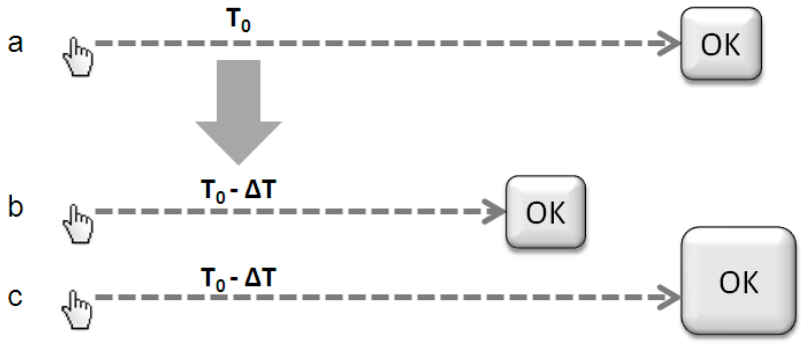


Figura 92. Conseguenze della legge di Fitts

[66] Se, per esempio, e il movimento venisse fatto a mano libera, la Figura 92 rappresenta approssimativamente le proporzioni reali delle due soluzioni, per ottenere un guadagno ΔT di circa il 10%. Per una trattazione più approfondita, con esempi di calcoli prestazionali, ci si può riferire al libro *The Psychology of Human-Computer Interaction*, già citato.

La legge di Fitts si può applicare a una varietà di situazioni, per modellare azioni di point&clic, o di drag&drop con svariati pointing device. Ciò che cambia sono le costanti della formula, che devono essere derivate sperimentalmente. In ogni caso, si applica solo ai movimenti lungo una sola dimensione, in condizioni di abilità normali (cioè, senza particolari, specifici allenamenti). Non si può, evidentemente, applicare alla presenza di ausilii software che servano a facilitare il movimento, per esempio bersagli che attirino a sé il puntatore.

La legge di Fitts non sorprende: anche se non conosciamo la relazione matematica precisa che lega D e S, è abbastanza ovvio che, ingrandendo il bersaglio o diminuendo la distanza dal punto di partenza, questo viene raggiunto più in fretta. Ci si aspetterebbe quindi che i sistemi fossero progettanti in modo da tenere in debito conto questo fatto elementare. È sorprendente, invece, quanto spesso ciò non avvenga. Un caso molto frequente è costituito dai bottoni di una pagina web. La Figura 93 mostra una porzione del menu principale di www.repubblica.it. Mentre per ogni voce di primo livello l'intera area grigia è cliccabile, per quelle di secondo livello bisogna cliccare esattamente sul testo: l'area circostante non è sensibile al clic del mouse. Ciò restringe inutilmente la dimensione del bottone, già molto ridotta per la lunghezza del menu. Rendere cliccabile l'intera area bianca attorno ad ogni voce aumenterebbe l'usabilità, senza utilizzare spazio aggiuntivo.

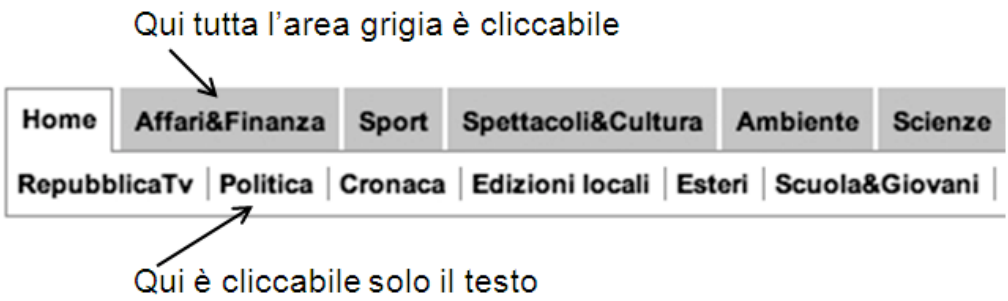


Figura 93. Il menu principale della homepage di www.repubblica.it (2010)

Questa soluzione è stata, invece, adottata nelle pulsantiere delle applicazioni di Office 2007, proprio considerando la legge di Fitts. Qui, i progettisti hanno reso sensibile al clic tutta l'area attorno ad ogni icona, includendo anche l'etichetta (Figura 94).



Figura 94. Bottoni in PowerPoint 2007. Il riquadro indica l'area sensibile al clic, sensibilmente più grande dell'icona

Un altro accorgimento suggerito dalla legge di Fitts è di utilizzare, al posto degli abituali menu a tendina, dei menu pop-up, che appaiono accanto al puntatore quando si preme il tasto destro del mouse. Questi hanno il vantaggio di ridurre la distanza fra il punto di partenza e il bersaglio. La Figura 95 mostra una variante insolita dei menu pop-up: una

pulsantiera pop-up, realizzata in Word 2007. Quando, durante la scrittura del testo, si seleziona una parola e si sposta il mouse di qualche pixel verso l'alto, accanto ad essa appare un pop-up con i bottoni più usati per cambiare gli attributi del testo. Le diverse opzioni sono molto vicine alla parola, e il percorso del puntatore verso il bersaglio si accorcia notevolmente. La pulsantiera svanisce automaticamente se si riporta il cursore un po' più in basso, o se si prosegue nella scrittura, in modo da non intralciare l'utente.

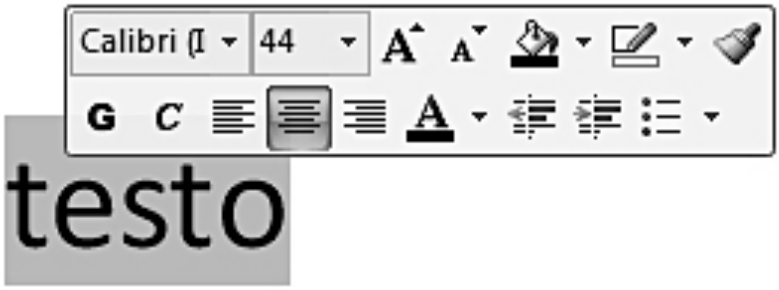


Figura 95. Pulsantiera pop-up in Microsoft Word 2007

Una variante dei menu pop-up è costituita dai *menu a torta* (*pie menu*), come quello di Figura 96, usato in Second Life. Cliccando sul tasto destro del mouse, questi menu compaiono proprio attorno al puntatore, che rimane visibile al centro. Le voci sono disposte a raggiera, equidistanti dal puntatore: in questo modo, il tempo necessario per raggiungerle è lo stesso per tutte. Le prestazioni sono ulteriormente migliorate dalla forma "a cuneo" delle voci del menu, che aumenta sensibilmente la superficie di ogni bersaglio, rispetto ai tradizionali menu a tendina.

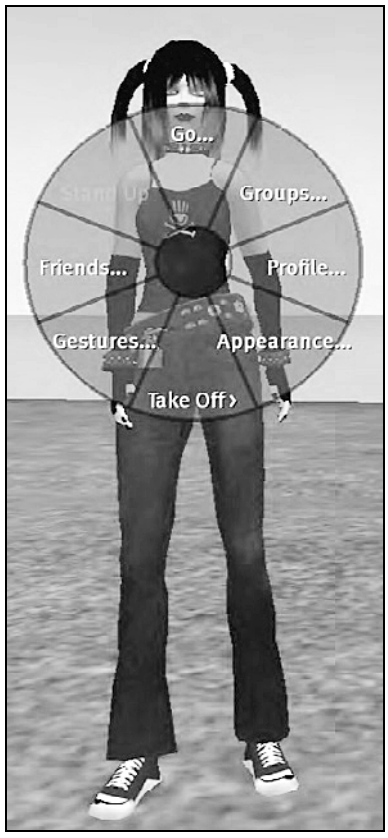


Figura 96. Pie menu in Second Life

Una conseguenza interessante della legge di Fitts è che i bersagli disposti lungo i bordi dello schermo sono particolarmente veloci da raggiungere. Infatti, il puntatore non può oltrepassare il bordo, comunque lo si muova: nella formula di Fitts è come se il bersaglio avesse dimensione S infinita, e quindi sarebbe, semplicemente, T=a. Per questo motivo, la barra dei menu delle applicazioni nel Mac, che si trova al bordo superiore del monitor, si raggiunge in modo più rapido di quella delle applicazioni di Windows, che si trova sul bordo superiore della finestra che contiene l'applicazione. Infatti, il bordo della finestra può essere oltrepassato dal puntatore, a differenza del bordo del monitor, e quindi il movimento per raggiungere il bersaglio risulta inevitabilmente più lento.

Una soluzione ancora migliore, per quanto riguarda la legge di Fitts, è quella in cui il bersaglio si trova in uno dei quattro angoli del monitor. In questo caso, infatti, i bordi dello schermo fungono da guida per il cursore, il quale, anche se diretto in modo approssimativo, "scivola" lungo il bordo fino a raggiungere l'angolo. Da questo punto di vista, il bottone START di Windows, che tradizionalmente si trova nell'angolo inferiore sinistro dello schermo, è in una posizione ideale. Questa favorevole posizione non era, però, correttamente sfruttata in Windows 95, perché il bottone era separato dal bordo della schermo da un margine di pochi pixel, non sensibili al clic del mouse, come si vede in Figura 97 (in alto). A causa di questi pochi pixel, quando il cursore toccava il bordo dello schermo, si trovava fuori bersaglio, e il pulsante non poteva essere selezionato. Questo errore, in seguito, è stato corretto. La Figura 97 (in basso) mostra il pulsante START in Windows XP, che si estende infatti fino ai bordi del monitor. Per lo stesso motivo, il grosso pulsante circolare comune a tutta la suite Office 2007 è stato posto nell'angolo in alto a sinistra dello schermo.[67]



Figura 97. Il bottone START in Windows 95 (in alto) e in Windows XP (in basso)

L'utente nel suo contesto

Nelle pagine precedenti abbiamo visto diversi esempi che mostrano come lo studio dell'usabilità dei sistemi interattivi non possa prescindere dallo studio della percezione e della cognizione umana e delle abilità che derivano dalla struttura psico-fisica delle persone. Anche se queste analisi cercano spesso di individuare le caratteristiche e le prestazioni funzionali cosiddette *normali*, cioè che caratterizzano l'essere umano "medio", le differenze individuali emergono continuamente, e possono essere sostanziali.

Queste differenze si rivelano ancora più marcate quando ampliamo i nostri studi e, dalla sfera della psicologia, ci portiamo in quella, più generale, dell'antropologia, per studiare l'uomo nella sua globalità: dal punto di vista sociale, culturale e dei suoi comportamenti quotidiani. Anche questi studi sono importanti per chi si occupa di usabilità, perché gli artefatti dell'uomo e i suoi comportamenti, individuali e sociali, sono strettamente legati.

[67] Questo esempio, e gli altri esempi di Figura 94 e Figura 95 sono tratti dal blog di Jensen Harris, dello user experience team della Microsoft, dove si trovano altri interessanti esempi di applicazione della legge di Fitts in Microsoft Office: http://blogs.msdn.com/jensenh/archive/2006/08/22/711808.aspx.

Infatti, l'influenza reciproca fra strumenti e comportamenti è molto profonda, soprattutto per gli strumenti a più ampia diffusione. Abbiamo già osservato come la diffusione "epidemica" di nuovi strumenti di comunicazione interattiva (il telefono cellulare, gli sms, la posta elettronica, i social network) abbia radicalmente trasformato nell'arco di pochi anni i comportamenti comunicativi della popolazione, almeno (per ora) dei paesi a più alto reddito. Al momento della stesura di questo libro, secondo Facebook, ciascuno dei 400 milioni di utenti del sito vi trascorre, in media, più di 55 minuti al giorno.[68] Questo a meno di sei anni dalla nascita del servizio. D'altro canto, la diffusione delle nuove modalità di comunicazione influenzano a loro volta l'evoluzione degli strumenti, che vengono continuamente modificati e migliorati via via che gli utilizzatori ne scoprono nuovi impieghi.

La rapidità e la dimensione di queste trasformazioni fa sì che le loro implicazioni siano ancora poco note. Per analizzarle, ci possiamo avvalere dei metodi e degli strumenti delle scienze dell'uomo. Queste indagini sono molto importanti, non solo perché ci permettono di "conoscere", ma anche, e soprattutto, perché ci possono orientare al "fare". Gli studi dell'interazione uomo-macchina, intesa nel senso più ampio d'*interazione società-tecnologia*, possono aiutarci a capire meglio quali scelte progettuali siano utili per il miglioramento complessivo della qualità della vita delle persone, e quali producano effetti indesiderabili.

I sistemi non esistono solo nelle vetrine di chi li vende, isolati da tutto, ma vivono nel mondo, nell'interazione continua con esso, ed è anche da questo punto di vista che devono essere considerati. Studiando i comportamenti che si sviluppano "attorno" ai sistemi, e nell'interazione con le persone, siamo in grado di capire che cosa deve essere migliorato, che cosa non va e, soprattutto, che cosa potrebbe essere fatto e ancora non c'è. Questi comportamenti non riguardano solo il singolo utilizzatore, nel suo rapporto con il sistema, ma l'intera comunità delle persone che cooperano in uno stesso contesto ambientale e sociale, siano o meno coinvolti anch'essi con il sistema.

Consideriamo l'esempio della Figura 98. Essa mostra un reparto ospedaliero di pediatria, in cui ai bambini ricoverati, a volte con degenze molto lunghe e lontani dalla propria città, è stato dato in dotazione un personal computer connesso a Internet. Ciò ha permesso di ridurre la "distanza" fra la permanenza in ospedale e la vita di tutti i giorni, permettendo ai piccoli pazienti non solo di giocare e di navigare su Internet, ma anche di rimanere in contatto con la famiglia e gli amici attraverso Skype, chat e posta elettronica, e con le attività didattiche della propria scuola. Alcuni bambini, di origine straniera, hanno potuto restare in contatto diretto con la famiglia lontana, residente nel paese di provenienza. Si tratta di un progetto tecnologicamente piuttosto semplice, che si avvale di tecnologie standard e si può realizzare rapidamente e a basso costo, ma di elevato valore sociale.[69] È evidente che l'analisi di questo sistema non può limitarsi allo studio della semplice interazione del singolo bambino con il computer a esso affidato. Il bambino e il suo computer sono parte di un sistema organizzativo, sociale e umano più ampio, composto di numerosi attori: medici, infermieri, genitori, amici, altri bambini ricoverati. Alcuni attori sono lontani, e intervengono attraverso la rete con modalità diverse (video, audio, messaggi testuali, in diretta o in differita). Altri sono vicini, in visita nel reparto, e potrebbero anch'essi utilizzare il computer per scopi ancora diversi. Per esempio, i genitori che assistono i bambini in ospedale potrebbero tenersi in contatto con il posto di lavoro, da cui si sono temporaneamente allontanati. E poi ci sono le possibili interazioni con la scuola e i suoi insegnanti, che apre interessanti prospettive di prosegimento dell'attività scolastica a distanza. La presenza dei computer determina la nascita di un contesto del tutto nuovo, nel quale si formano nuovi comportamenti e si delineano nuove esigenze.

Per comprendere questo nuovo contesto, individuarne gli eventuali problemi e le possibili soluzioni, è necessario studiarlo da vicino, ed eventualmente "immergervisi" e sperimentarlo di persona. Questo è l'obiettivo che si propongono i cosiddetti *studi sul campo* (*field study*). I metodi per questi tipi di analisi sono quelli dell'etnografia.

[68] Dati riportati nel sito di Facebook, alla sala stampa: http://www.facebook.com/press (aprile 2010).
[69] Il progetto è stato realizzato dai volontari di Informatici Senza Frontiere, con l'aiuto di altre associazioni di volontariato e con PC usati donati da varie aziende, per l'ospedale di Brescia e altri ospedali (www.informaticisenzafrontiere.org).



Figura 98. Utilizzo di Internet in un reparto di pediatria (cortesia Informatici Senza Frontiere)

L'etnografia

L'*etnografia* è un metodo ben consolidato nel campo delle ricerche sociali e antropologiche, sviluppato dalla fine dell'800, quando le potenze coloniali sentivano la necessità di approfondire la conoscenza delle loro colonie. Il ricercatore etnografico, da solo o in team, s'immerge nelle attività quotidiane di una società, comunità o organizzazione, allo scopo di raccogliere dei dati che, una volta interpretati, permettano di comprenderne la cultura, l'organizzazione, i comportamenti, le usanze. L'etnografo, come suggerisce il nome,[70] non costruisce teorie o modelli della società, si limita a raccogliere e a registrare informazioni, che saranno utilizzate da altri.[71] Immergendosi direttamente nell'ambiente oggetto di studio, è in grado di raccogliere dati che sarebbe impossibile ottenere in altri modi. Lavora con l'osservazione diretta, con interviste o questionari; a volte diventando anch'egli, per il periodo dello studio, parte attiva della comunità che descrive. L'etnografo esamina il contesto complessivo, nella convinzione che le comunità si comprendano meglio considerandone tutti gli aspetti: i comportamenti, l'ambiente, gli artefatti, i riti, la lingua, la cultura, i valori, le credenze, e così via. Registra, ma non esprime giudizi, nella convinzione che ogni cultura debba essere compresa basandosi sui propri parametri, e non su quelli dell'osservatore.

Dagli anni '90, i metodi dell'etnografia sono stati adottati anche nella human-computer interaction, e in particolare nell'ambito della progettazione dei sistemi interattivi. In quest'area, gli obiettivi delle ricerche dell'etnografo possono essere diversi. In alcuni casi avrà il compito di osservare l'utilizzo dei prodotti della tecnologia in un certo contesto, per esempio il reparto pediatrico della Figura 98, per individuarne i problemi e riportarli ai progettisti per i necessari interventi migliorativi. In altri casi avrà il compito di analizzare i comportamenti all'interno di un gruppo o di un'organizzazione, per fare emergere i bisogni ancora inespressi, che possono suggerire prodotti o servizi nuovi. Per esempio, osservando i comportamenti all'interno di un reparto ospedaliero non ancora informatizzato, per individuarne necessità e problemi. In ogni caso, il suo scopo è di osservare, comprendere e descrivere i comportamenti dell'utente (attuale o potenziale), per riportarli a chi progetterà le soluzioni più adatte. Per questo, la formazione del ricercatore

[70] Etnografia deriva dalle parole greche *ethnos*=nazione, popolo, e *graphein*=scrivere; significa quindi, letteralmente, "descrizione dei popoli".
[71] Non bisogna confondere l'etnografia con l'etnologia. Quest'ultima (da *ethnos*=nazione, popolo e *lògos*=parola, discorso) si avvale delle ricerche della etnografia per confrontare le caratteristiche dei vari popoli, e elaborare teorie e modelli.

etnografico è di solito specifica: egli proviene dalle scienze umane, e non dall'informatica o dalla progettazione dei sistemi.

Andy Crabtree, autore di un libro sui metodi dell'etnografia nella progettazione dei sistemi collaborativi,[72] in un suo articolo così descrive il lavoro dell'etnografo in questo campo:

> *L'etnografia opera come i nostri "occhi e orecchi sul campo", per informarci sulle pratiche correnti, il cui significato altrimenti passerebbe inosservato. Il ruolo dell'etnografia nella progettazione, così come è emerso nei nostri studi, consiste nella sua capacità di fornire il senso concreto degli aspetti reali di un certo ambiente: vedere le attività come azioni sociali immerse in un dominio socialmente organizzato, compiute attraverso le attività quotidiane dei suoi abitanti, e comunicare queste informazioni ai progettisti. Questo approccio si focalizza sulle specifiche attività che i progettisti desiderano comprendere, analizzare e ricostruire, e le documenta. È la capacità dell'etnografia di descrivere l'ambiente sociale così come viene percepito dagli "utenti", che la fa apprezzare dai progettisti. Di conseguenza, l'etnografia è molto valida nell'identificare le eccezioni, le contraddizioni e le contingenze delle attività lavorative, che costituiscono le reali condizioni dell'ambiente ma che di solito non figurano nelle descrizioni ufficiali o formali.*[73]

Per esempio, durante uno studio sul campo presso un'organizzazione aziendale, il ricercatore etnografico potrà raccogliere le seguenti informazioni:

- Organigrammi aziendali e ordini di servizio che definiscono le responsabilità e ruoli assegnati all'interno dell'organizzazione;
- Procedure di lavoro formalizzate per l'esecuzione delle attività, e relativa modulistica;
- Raccolta di artefatti rilevanti utilizzati, quali esempi di moduli compilati, diagrammi, tabulati, comunicazioni interne;
- Differenze osservate fra i ruoli ufficiali e quelli informali;
- Descrizione di come le procedure ufficiali vengono effettivamente eseguite, e delle cause delle eventuali deviazioni;
- Schemi dei processi di lavoro osservati, e descrizione delle difficoltà;
- Descrizione di come sono gestite le situazioni anomale o eccezionali;
- Layout e fotografie degli ambienti di lavoro, con l'indicazione degli spostamenti abituali;
- Registrazione di interviste significative con le persone dell'organizzazione;
- Video o foto di situazioni rilevanti.

Queste ricerche possono essere svolte in ambienti molto diversi, e non necessariamente all'interno di organizzazioni formali come un'azienda, un ospedale, un ente pubblico. Possono svolgersi presso comunità informali, come un quartiere, un villaggio, un gruppo sociale, allo scopo di analizzare come determinati prodotti o servizi vengano utilizzati nella vita quotidiana, o individuare necessità ancora inespresse.

Il ricercatore etnografico si trova al confine fra due mondi: quello delle scienze dell'uomo, e quello della progettazione (Figura 99). Raccoglie informazioni che dovranno permettere di incrociare i bisogni degli utenti con le possibilità della tecnologia. Può essere un compito difficile, soprattutto quando i bisogni sono inespressi, e i potenziali destinatari di una soluzione non sono in grado di descrivere con chiarezza le loro esigenze. Sentono il disagio della situazione corrente, ma non sanno metterne a fuoco le cause e indicarne le possibili soluzioni. Gli utenti raramente conoscono le potenzialità della tecnologia corrente, e tendono a ritenere che ciò che non è mai stato fatto non si possa fare. Non sanno che cosa chiedere, e, di conseguenza, i progettisti non sanno che cosa dare. Il ricercatore etnografico ha il compito di facilitare questo incrocio.

---

[72] A.Crabtree, *Designing Collaborative Systems: A Practical Guide to Ethnography*, Springer-Verlag, 2003.

[73] A.Crabtree, D.M.Nichols, J.O'Brien, M.Rouncefield, M.B.Twidale, *Ethnomethodologically Informed Ethnography and Information System Design*, in Journal of the American Society for Information Science, vol.51 (7), pagg.666-682, disponibile anche in rete (nostra traduzione).

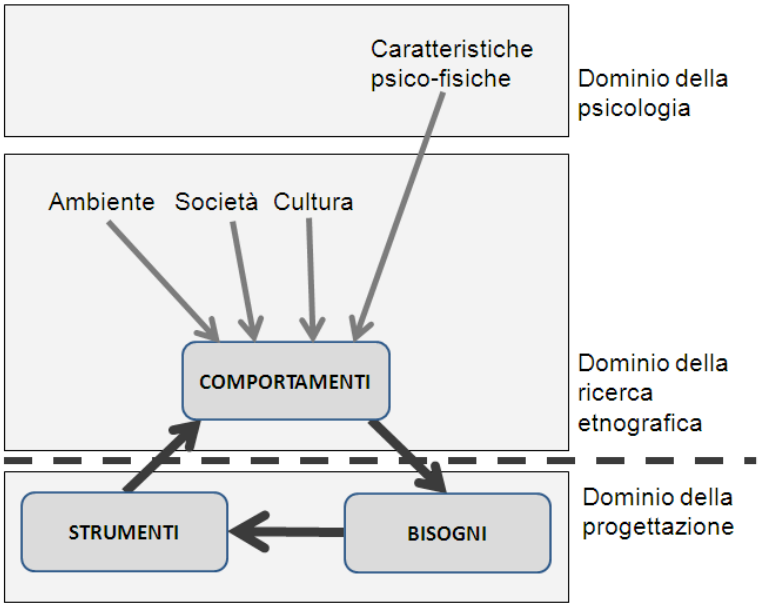


Figura 99. I diversi domini nello studio dell'utente

Per questa sua posizione "borderline", può interpretare il suo ruolo in modi diversi. Alcuni lo considerano un ruolo ancillare, di puro supporto al progettista: i suoi "occhi e orecchi" sul campo, come afferma il brano sopra citato. Altri, andando oltre i metodi dell'etnografia tradizionale, interpretano il proprio ruolo in modo più ampio: non si limitano a descrivere ciò che vedono, ma cercano di estrarne il senso, dandone una lettura critica, che possa orientare il progetto in modo sostanziale:[74]

> *Comprendere la esperienza dell'utente con le tecnologie richiede attenzione ai significati sociali e culturali: che cosa significa il prodotto per l'utente, che cosa significa nel contesto di particolari culture, e che cosa significa nei termini del suo impatto complessivo sull'ambiente sociale e globale? Attraverso il possesso e l'uso di particolari tecnologie e artefatti, noi facciamo dichiarazioni su noi stessi e sui nostri valori. Nella progettazione, produzione e commercializzazione di nuove tecnologie digitali, noi, anche, introduciamo e normalizziamo certi tipi d'interazione sociale e di valori. […] La human-computer interaction si avvale già di discipline non ingegneristiche come l'etnografia e il design, allo scopo di comprendere meglio l'esperienza e l'estetica nel design della tecnologia. Discipline basate sulle scienze umane, come l'antropologia, la letteratura, e gli studi sulle culture e sui media, possono fornire un ulteriore insieme di tecniche e metodi per comprendere come ci rapportiamo alla tecnologia e agli artefatti culturali.*[75]

Ripasso ed esercizi

1. Che cosa s'intende per human information processor?
2. Quali indicazioni fornisce lo studio dell'attenzione al progettista di sistemi interattivi?
3. Spiega le relazioni fra i sistemi di memoria umana, secondo il modello modale.
4. Quali sono le caratteristiche della memoria e a breve termine?
5. A che cosa si riferisce il "magico numero sette" e quali implicazioni ha sull'usabilità dei sistemi interattivi?
6. Cerca, nei programmi che utilizzi normalmente, una funzione che sovraccarica in modo eccessivo la memoria a breve termine dell'utente.

[74] Per una discussione su queste due diverse filosofie, si veda A.Crabtree, T.Redden,P.Tolmie, G.Button, *Ethnography Considered Harmful*, Proceedings of the AC CHI Conference 2009, pagg.879-888 (in cui si sostiene il primo punto di vista).
[75] Dalla presentazione del dibattito su *Designing Culturally Situated Technologies for the Home*, con G.Bell, M.Blythe, B.Gaver, P.Sengers, P.Wright, in Proceedings of ACM CHI 2003 Conference, pagg.1062-1063 (nostra traduzione).

7. Come possiamo facilitare la codifica delle informazioni nella memoria a lungo termine?
8. Spiega la differenza fra riconoscimento e recupero in relazione alla memoria a lungo termine.
9. Quali implicazioni hanno le caratteristiche della memoria a lungo termine sull'usabilità dei sistemi interattivi?
10. Come si definisce l'acuità visiva?
11. Che cosa s'intende per visione foveale e visione periferica?
12. Che cosa s'intende per cecità cromatica, e a che cosa è dovuta?
13. Fra i programmi che usi di frequente, quali richiedono apprendimento motorio? Quali tipi di feedback forniscono?
14. Spiega la legge di potenza della pratica.
15. Spiega la legge di Fitts, e le sue implicazioni sull'usabilità dei sistemi interattivi.
16. Considerando la legge di Fitts, quali tipi di menu permettono prestazioni migliori?
17. Che cos'è l'etnografia?
18. Quale ruolo ha il ricercatore etnografico nella human computer interaction?

Approfondimenti e ricerche

1. Per approfondire i meccanismi dell'attenzione, della memoria e, più in generale, della cognizione umana si può partire da un manuale universitario di psicologia generale. Per esempio, P.Gray, *Psicologia* (Seconda edizione italiana condotta sulla quarta edizione americana), Zanichelli, 2004.
2. In rete esistono numerosi servizi gratuiti che simulano i diversi tipi di cecità cromatica, mostrando come certi gruppi di colori sarebbero visti da chi ne è affetto. Per esempio, http://colorschemedesigner.com o http://www.vischeck.com. Esperimenta qualcuno di questi servizi, identificando le scelte cromatiche più adatte anche per chi ha problemi nella distinzione dei colori.
3. Approfondisci le possibili applicazioni della legge di Fitts nella progettazione di sistemi interattivi. Suggerimento: Bruce Tognazzini, nel suo sito web, propone una interessante serie di quiz (con relativa risposta) sull'applicazione di questa legge a tipici problemi di design (http://www.asktog.com/columns/022DesignedToGiveFitts.html).
4. Analizza l'interfaccia utente di alcuni siti web di grande diffusione dal punto di vista della legge di Fitts. A tuo parere, i progettisti ne hanno tenuto conto nel design dal layout grafico?
5. Approfondisci il ruolo e i metodi dell'etnografia per la progettazione di sistemi interattivi. Un buon punto di partenza può essere l'articolo di M.D.Myers, *Investigating Information Systems with Ethnographic Research*, in Communications of the Association for Information Systems, vol.2, article 23 (1999), disponibile anche in rete in http://www.qual.auckland.ac.nz/Myers%20CAIS%20article.pdf.

5. Progettare per l'utente

Sintesi del capitolo

Questo capitolo si occupa della progettazione dei sistemi usabili. L'approccio tradizionale della progettazione centrata sul sistema viene confrontato con quello della cosiddetta progettazione centrata sull'essere umano, che mette l'utilizzatore, con i suoi bisogni, abitudini e comportamenti, al centro del processo di progettazione. In questo modo, il progettista di sistema diventa, in primo luogo, progettista dell'interazione fra utente e sistema: il processo di progettazione prende le mosse dai casi d'uso, e non dalle funzionalità offerte dal sistema. Questo approccio corrisponde a un livello di maturità più elevato del processo di progettazione. L'attività finalizzata alla progettazione di sistemi usabili per tutti prende il nome di progettazione universale di cui sono brevemente ricordate le diverse strategie.

Che cosa significa progettare

Nella lingua italiana, e soprattutto nella pratica dell'informatica, il termine *progettare* (con i suoi derivati: *progetto*, *progettazione, progettista*) è spesso utilizzato in modo impreciso. È quindi opportuno definirlo con precisione. Nel vocabolario troviamo la seguente definizione:

> **Progettare** [dal francese *projeter*, dal latino *proiectāre*, intensivo di *proĭcere*, "gettare avanti", composto di *prō* "avanti" e *iăcere* "gettare"] : 1. Immaginare, ideare qualcosa e studiare il modo di attuarla; 2. Ideare la costruzione di un edificio, di una struttura, di una macchina, ecc., compiendo i relativi calcoli e disegni per la sua realizzazione.[76]

In sostanza, nell'attività di progettazione si parte da un esame della situazione attuale (*ciò che è*), per riconoscerne i difetti o i limiti e, sulla base delle possibilità offerte dalla tecnologia (*ciò che potrebbe essere*), si concepisce e si specifica la situazione futura (*ciò che vogliamo che sia*, Figura 100). Progettazione è quindi un'attività di natura sia intellettuale sia pratica: non basta una "visione" del futuro desiderato, ma occorre anche definire tutti i dettagli che ne permetteranno la realizzazione.

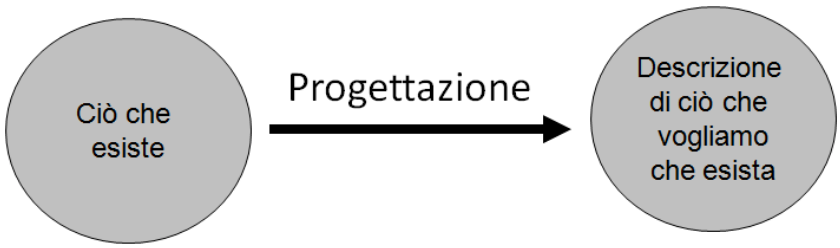


Figura 100. Significato di progettazione

Progettare è, pertanto, attività completamente diversa dal realizzare. Nello stesso vocabolario troviamo, infatti:

> **Realizzare** [dal francese *réaliser*, da *réel* "reale", da cui dipende direttamente anche l'inglese *to realize*]: 1. Rendere reale qualcosa attuandola praticamente; 2. …

Realizzare è quindi un'attività molto concreta (il termine deriva, in definitiva, dal latino *res*, che significa "cosa"): si parte da un progetto (il prodotto dell'attività di progettazione) e lo si attua concretamente. Per esempio, a partire dal progetto di un edificio si organizza il cantiere per la sua costruzione, e lo si costruisce.

[76] Vocabolario della lingua italiana di N.Zingarelli, ed. Zanichelli, 2002

Nella pratica corrente, soprattutto in informatica, il termine progettare è spesso usato in modo impreciso, per comprendere non soltanto le attività di progettazione in senso proprio, ma anche la successiva realizzazione. Così, per *progetto* non si intende solo il risultato della progettazione, come sarebbe corretto (ancora dal vocabolario: "progetto, insieme di calcoli, disegni, elaborati necessari a definire inequivocabilmente l'idea in base alla quale realizzare una qualsiasi costruzione"), ma spesso, in modo più ampio, tutte le attività connesse allo sviluppo di un sistema, dalla progettazione alla sua realizzazione concreta.

Molto usato in questo contesto è anche il termine inglese *design*. Il verbo *to design* significa, semplicemente, "progettare". Tuttavia, a confondere ulteriormente le cose, questa parola viene spesso usata, soprattutto in Italia, con sfumature diverse. Per esempio, quando usiamo il termine *industrial design* (che significa "progettazione industriale") a volte intendiamo sottolineare i valori di natura estetica o formale dei prodotti della progettazione. Quando diciamo *design italiano* vogliamo spesso sottolineare la stessa cosa.

In questo libro, il termine progettazione sarà usato in modo coerente con il suo significato etimologico, e il termine design sarà usato come sinonimo di progettazione.

Progettare l'interazione

La progettazione di sistemi *usabili* richiede un drastico cambiamento di mentalità rispetto all'approccio di progettazione tradizionale. Nella progettazione tradizionale, l'oggetto principale dell'attenzione è il *sistema* da progettare (Figura 101 a). Il processo di progettazione parte dalla definizione dei suoi *requisiti funzionali*, cioè dall'identificazione delle *funzionalità* (o delle *funzioni*)[77] che esso deve fornire al suo utente, che vengono descritte in dettaglio in un documento di *specifiche funzionali*, a partire dal quale il sistema viene progettato e quindi realizzato. In questo approccio, l'utente del sistema ha un ruolo, tutto sommato, abbastanza marginale: il progettista concentra la sua attenzione sulle funzionalità, e sugli aspetti tecnici connessi alla loro realizzazione, per arrivare a soddisfare le specifiche con un rapporto costo/qualità accettabile.

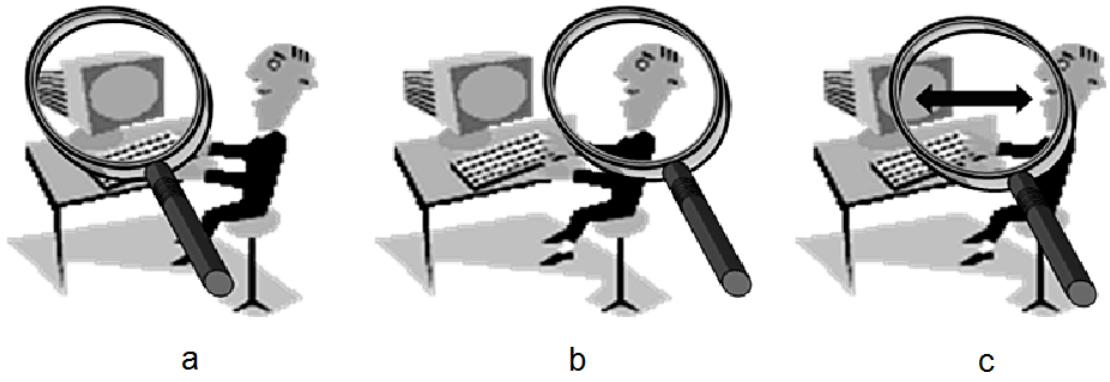


Figura 101. Dalla progettazione tradizionale alla progettazione dell'interazione

Se l'obiettivo è la progettazione di un sistema usabile, questo approccio non funziona. In questo caso, il progettista dovrà porre la sua attenzione, in primo luogo, sull'utente (Figura 101b), e dovrà studiarne le caratteristiche, le abitudini e le necessità in relazione all'uso del sistema. Dovrà preconfigurare i vari *contesti* in cui il sistema sarà utilizzato, e i suoi diversi *casi d'uso*; dovrà analizzare in dettaglio i *compiti* che l'utente svolgerà con il sistema. Tutto questo allo scopo di progettare un sistema che si adatti all'utente, per così dire, come un vestito su misura. Secondo questa impostazione, il compito del progettista non sarà più semplicemente quello di progettare le funzioni del sistema, ma quello di progettare l'interazione fra il sistema e il suo utente (o i suoi utenti), come indicato in Figura 101c. Si parla, così, di *interaction design* e, per sottolineare che il punto di partenza è l'utente, di *progettazione centrata sull'essere umano* (in inglese, *human-centred design* o, semplicemente, *HCD*).

[77] In questo libro, useremo in modo equivalente le dizioni *funzionalità del sistema* e *funzione del sistema*.

Progettazione human-centred

La progettazione centrata sull'essere umano è l'oggetto di un altro standard molto importante per gli argomenti di questo libro, l'ISO 13407: *Human-centred design processes for interactive systems.* Questo documento ha lo scopo, come specifica nell'introduzione, "di aiutare chi ha la responsabilità di gestire i processi di progettazione di hardware e software a identificare e pianificare efficaci e tempestive attività di progettazione human-centred. Esso complementa i vari metodi e approcci alla progettazione esistenti." Si tratta di un documento molto importante, e ormai ben consolidato, che sarà ripreso più volte nel seguito. La filosofia e le motivazioni della progettazione human-centred sono ben riassunte dalla seguente definizione, contenuta nei paragrafi introduttivi di questo documento:[78]

> *La progettazione centrata sull'essere umano (human-centred design) è un approccio allo sviluppo dei sistemi interattivi specificamente orientato alla creazione di sistemi usabili. È un'attività multi-disciplinare che incorpora la conoscenza e le tecniche dei fattori umani e dell'ergonomia. L'applicazione dei fattori umani e dell'ergonomia alla progettazione dei sistemi interattivi ne potenzia l'efficacia e l'efficienza, migliora le condizioni del lavoro umano e contrasta i possibili effetti avversi dell'uso sulla salute, sulla sicurezza e sulle prestazioni. Applicare l'ergonomia alla progettazione dei sistemi richiede che si tenga conto delle capacità, delle abilità, delle limitazioni e delle necessità umane. I sistemi human-centred supportano gli utenti e li motivano a imparare. I benefici possono includere una maggiore produttività, una migliore qualità del lavoro, riduzione dei costi di supporto e di addestramento e una migliore soddisfazione dell'utente.*

La progettazione human-centred non è una specifica metodologia di progettazione, ma un approccio generale, che può essere concretamente sviluppato in molti modi, in funzione della natura dei prodotti da realizzare e delle caratteristiche dell'organizzazione che ospita il progetto. A questo proposito, l'ISO 13407 specifica che, quali che siano i processi e i ruoli adottati, l'utilizzo di un approccio human-centred è caratterizzato dai seguenti quattro punti:

a. il coinvolgimento attivo degli utenti e una chiara comprensione dei requisiti degli utenti e dei compiti;
b. un'assegnazione appropriata delle funzioni fra utenti e tecnologia;
c. l'iterazione delle soluzioni di progetto;
d. una progettazione multi-disciplinare.

Avremo modo di sviluppare ampiamente, nel seguito, i punti a), b) e c), e lo loro implicazioni. Per ora desideriamo sottolineare che questo approccio è intrinsecamente multi-disciplinare (punto d). E' questo, forse, il cambiamento più rilevante rispetto a un approccio progettuale tradizionale, perchè richiede un'organizzazione diversa dei gruppi di progetto. L'ingegnere (e l'ingegnere del software in particolare) di formazione tradizionale non è attrezzato per risolvere i problemi che la centralità dell'utente pone al progettista. Per questo sono necessarie altre competenze, relative alle scienze dell'uomo e non alla tecnologia. L'analisi e la comprensione dei comportamenti e delle loro motivazioni, l'analisi e la conoscenza dei processi percettivi e cognitivi coinvolti nell'interazione con i sistemi, la comprensione delle problematiche ergonomiche, la competenza sulle diverse modalità della comunicazione umana: tutto questo deve inevitabilmente far parte dei processi che portano alla progettazione e alla realizzazione di sistemi usabili. Data l'ampiezza delle conoscenze e la diversità delle sensibilità coinvolte, i team di progettazione devono necessariamente coinvolgere professionisti di formazione molto differente, che svolgono mestieri di nuovo tipo: esperti di usabilità, ergonomi cognitivi, esperti di user experience, e così via.

Di questo tratteremo meglio nel seguito. Per ora vogliamo sottolineare il fatto che lo HCD produce *risultati completamente diversi* da quelli ottenuti con l'approccio tradizionale. Questo è un punto d'importanza fondamentale, che deve essere ben compreso. L'esperienza nella didattica dello HCD insegna che, molto spesso, i progettisti con un background tecnico (per esempio, i progettisti di software) tendono a sottovalutare l'impatto di un'impostazione human-centred sui risultati del loro lavoro. La raccomandazione di partire dall'analisi dell'utente e dei suoi bisogni viene considerata ovvia, e quindi non meritevole di particolari riflessioni e approfondimenti. Ma non è così. Se non si comprende il senso profondo contenuto in questo approccio e la sua diversità, è facile tornare alle vecchie abitudini, e progettare non interazioni, ma funzioni, a scapito dell'usabilità del prodotto finale.

[78] Tutte le citazioni sono tratte dalla versione del 1999, in nostra traduzione dall'originale in lingua inglese.

Un esempio

Un esempio emblematico è costituito dai sistemi audio-video domestici. Si tratta di sistemi realizzati collegando fra loro componenti diversi, con un approccio di tipo modulare: un amplificatore, un lettore di DVD, un monitor televisivo, un sistema di altoparlanti, un decoder, e così via. Ogni componente offre un insieme molto articolato di funzioni, controllabili sia da un pannello frontale che da un telecomando. L'utente ha quindi la possibilità di governare singolarmente ciascun componente. L'approccio, dal punto di vista ingegneristico, sembra perfetto: la modularità permette di connettere componenti di vario tipo, anche di produttori diversi, consentendo di configurare il sistema in modo molto flessibile, a seconda delle particolari esigenze. Ma, come tutti noi sappiamo per esperienza diretta, l'usabilità di questi sistemi è bassissima.

La Figura 102 mostra il sistema di chi scrive, costituito da schermo televisivo, amplificatore, decoder, player DVD, VHR, giradischi. Il sistema prevede l'uso di ben 5 telecomandi separati (il giradischi, di vecchia produzione, non ha telecomando), dotati complessivamente di poco più di 200 pulsanti (!). A questi si aggiungono una settantina di pulsanti e manopole presenti sui pannelli frontali dei vari apparati. Per "semplificare" la situazione, è stato fornito un ulteriore (sesto) telecomando "universale", in grado di simulare tutti gli altri (con altri 48 pulsanti, che porta il totale a circa 320…). Quest'ultimo però non è in grado di simulare tutte le funzioni degli altri telecomandi, ma solo un sottoinsieme abbastanza limitato, pertanto i cinque telecomandi non possono essere eliminati: a essi si dovrà ricorrere per funzioni particolari, di uso non frequente, ma comunque necessarie. Il sistema è corredato di 7 manuali di istruzioni: uno per ogni componente, più uno per il telecomando universale (Figura 103).

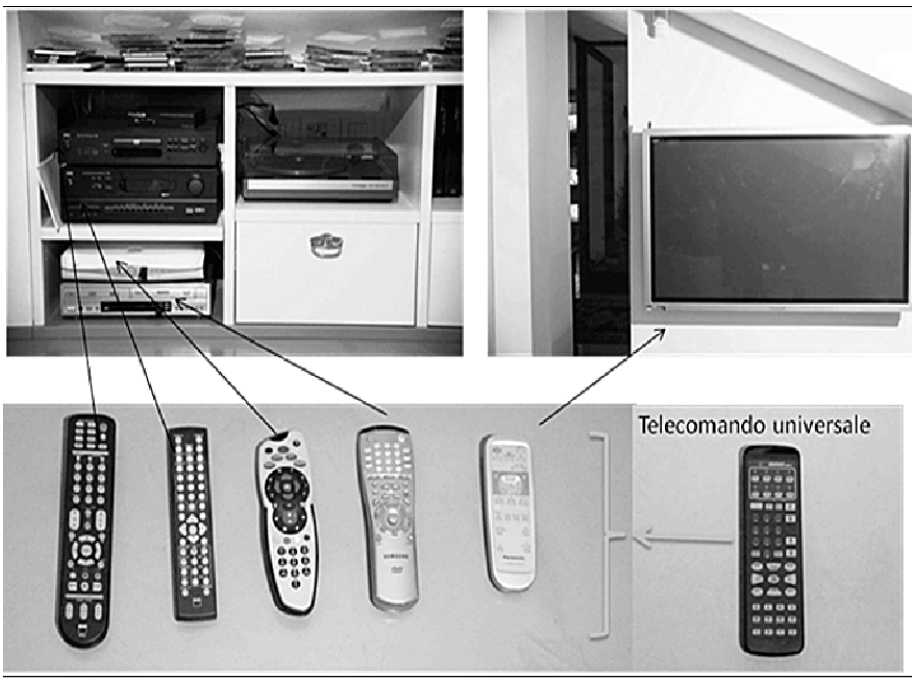


Figura 102. Il mio sistema audio-video

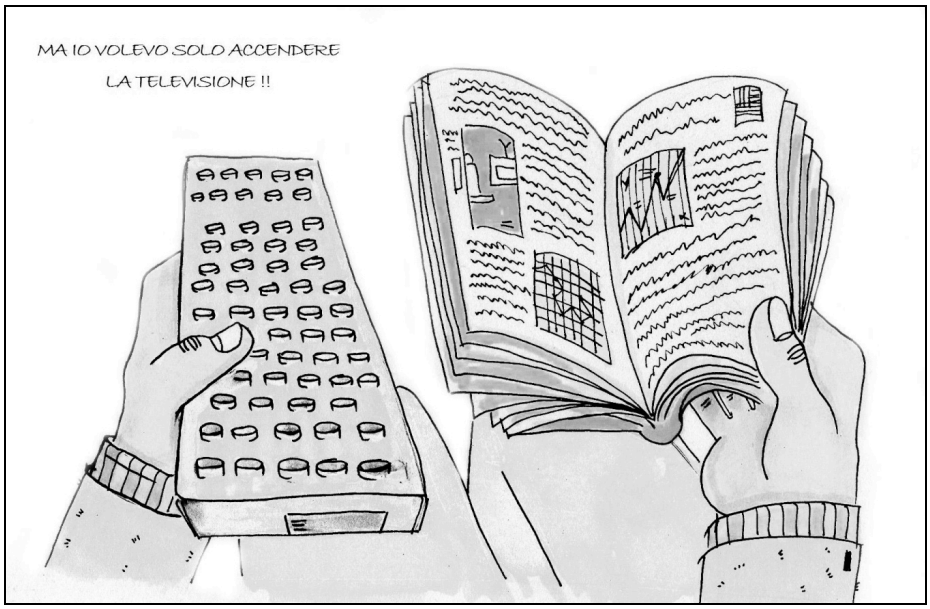


Figura 103. Ma io volevo solo accendere la televisione! (Disegno di Alberto Iobbolo)

Se ora analizziamo le necessità degli utenti di questo sistema, vediamo una situazione completamente diversa da quella che sembra avere ispirato i progettisti. Le funzioni che interessano agli utenti non vengono fornite dai singoli componenti modulari, ma dalla loro cooperazione. Le combinazioni di comandi che sono effettivamente utili nell'uso quotidiano sono in numero enormemente inferiore rispetto a quelle potenzialmente ottenibili con gli oltre 300 pulsanti: le sequenze realmente significative sono poche e ricorrenti. Per vedere il telegiornale della sera, si dovrà accendere il decoder, l'amplificatore e il monitor televisivo, connetterli in qualche modo fra loro, e selezionare il canale televisivo. Poi si dovrà regolare il volume. Questa sequenza corrisponde a un singolo caso d'uso molto frequente. Per chi scrive corrisponde, anzi, al caso d'uso di gran lunga più importante, quello che, da solo, giustificherebbe l'acquisto dell'impianto. Gli altri casi d'uso corrispondono alla visione di un DVD e all'ascolto di un CD musicale. Ecco che, in una progettazione human-centred, il sistema avrebbe potuto avere un numero molto limitato di comandi di base (qualche unità) ai quali aggiungere alcuni comandi per le regolazioni iniziali o durante gli sporadici interventi di manutenzione, che avrebbero potuto essere resi visibili soltanto ai tecnici dell'assistenza. Tutto questo senza una sostanziale riduzione di prestazioni, ma con un significativo guadagno in termini di usabilità.

Quest'analisi potrebbe essere ulteriormente approfondita, considerando per esempio la posizione fisica dei componenti in relazione a quella dell'utente durante l'utilizzo dei telecomandi. Dove si trova quando accende la TV per guardare il telegiornale? Ci sono delle barriere architettoniche che intercettano i segnali del telecomando da tale posizione? Queste analisi non sono state fatte – e non vengono normalmente mai fatte - in fase di progettazione. In effetti, l'uso iniziale del sistema, dopo la sua installazione, ha rivelato notevoli problemi di usabilità, ed ha generato diversi interventi di modifica successivi all'acquisto. Un approccio che parta dalle necessità dell'utente, nel caso specifico, avrebbe prodotto una configurazione dell'impianto molto diversa, senza alcuna necessità di cambiare i componenti standard, ma solo posizionandoli e interconnettendoli in modo diverso. Tutto ciò sembra ovvio, ma non viene mai fatto: il venditore all'inizio, e l'installatore successivamente, non si preoccupano di indagare sulle abitudini di chi compra. Eppure, sarebbe proprio quest'analisi a dare il reale valore d'uso al sistema.

Si dovrebbe partire dall'utente nella progettazione di qualsiasi strumento, semplice o complesso: una scopa, un frigorifero, il cruscotto di un jumbo jet.

I casi d'uso

La nozione di caso d'uso, richiamata nella sezione precedente senza definirla, è di grande importanza nella progettazione human-centred, e merita un approfondimento. In termini del tutto generali, un caso d'uso può essere definito come *un insieme d'interazioni fra l'utente (o più utenti) e il sistema, finalizzate a uno scopo utile per l'utente*.

Non bisogna confondere i casi d'uso con le funzionalità del sistema. In un caso d'uso, il soggetto è l'utente. Nell'esempio visto sopra, Ascoltare un CD o Guardare un DVD sono casi d'uso: è l'utente che ascolta o guarda un CD. Una funzione, invece, è una prestazione realizzata dal sistema, per esempio Caricamento del CD, Espulsione del CD, Accensione del decoder, Selezione della traccia successiva del CD, Selezione del menu del CD. In questo esempi, il soggetto è il sistema. La distinzione è sottile, ma fondamentale. Un caso d'uso viene realizzato, di solito, mediante la esecuzione di più funzioni del sistema; d'altro canto, una funzione del sistema potrà essere utilizzata da diversi casi d'uso. Un caso d'uso è un insieme d'interazioni che, considerate nel loro insieme, *producono un risultato utile dal punto di vista dell'utente*. Il punto fondamentale è proprio questo: l'utilità per l'utente. Così, per esempio, non ha molto senso considerare casi d'uso delle azioni elementari dell'utente, come Espellere il CD, o Accendere il decoder. Ciascuna di esse, eseguita da sola, non ha un grande valore per l'utente, perché non gli permette di raggiungere uno scopo significativo.

Per indicare un caso d'uso, si può utilizzare un verbo alla terza persona singolare, per sottintendere che il soggetto è l'utente. Il verbo potrà essere seguito da un complemento, di solito un complemento oggetto: Ascolta un CD, Guarda un DVD. Come ulteriore esempio consideriamo un telefono cellulare. Nella Figura 104 sono indicati, a sinistra, due casi d'uso (Fa una telefonata e Invia un SMS*)*. Essi sono realizzati mediante le funzioni elencate sulla destra. Le frecce indicano l'associazione fra casi d'uso e funzioni. Si vede che la funzione Cercare nome in rubrica viene utilizzata per trovare il numero di telefono del destinatario, in entrambi i casi.

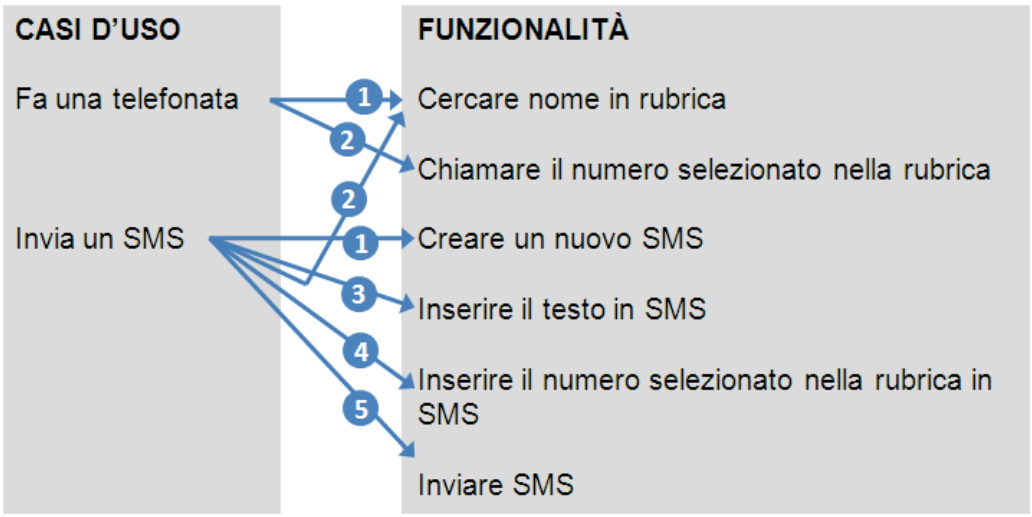


Figura 104. Casi d'uso e funzionalità

Semplificando al massimo, possiamo dire che il progettista orientato al sistema si occupa di progettare funzioni, lasciando all'utente il compito di "metterle nella sequenza giusta", per ottenere ciò che gli serve. Si accontenta di fornire i mattoni di base, lasciando all'utente l'incombenza di costruirsi ciò che gli serve. Segue un approccio *bottom-up,* come chi ha progettato il mio sistema audio-video. Il progettista orientato all'utente, invece, desidera conoscere *perchè* l'utilizzatore adopererà il sistema, e vuole permettergli di raggiungere questi obiettivi nel modo più semplice e lineare. Segue un approccio *top-down:* non parte dalle funzioni, ma dagli obiettivi – ciò che abbiamo chiamato casi d'uso - le funzioni saranno definite dopo, di conseguenza.

L'identificazione dei casi d'uso è un'attività fondamentale nella progettazione human-centred. Disporre di un elenco ben fatto dei casi d'uso del sistema costituisce un primo passo indispensabile per poterlo progettare. La formulazione di questo elenco può sembrare un compito banale, ma non è così. Per eseguirlo correttamente, occorre superare diverse difficoltà. Innanzitutto, esiste sempre il rischio di scambiare i casi d'uso con le funzionalità, e di ricadere nelle tradizionali pratiche della progettazione orientata al sistema. Inoltre, occorre individuare il giusto livello di astrazione. Come abbiamo già osservato, un caso d'uso è un insieme d'interazioni *che producono un risultato utile per l'utente*. Quindi, non ogni insieme d'interazioni può essere considerato un caso d'uso. Per esempio, in uno sportello Bancomat, Preleva contante è un caso d'uso, ma Digita la password non lo è. Infatti, quest'ultima azione non ha un valore di per sé, ma è solo parte di un processo che permette di effettuare il prelievo. È solo questo che interessa all'utente, ed è per questo che deve effettuare una serie di compiti, per così dire, ancillari, come anche Inserire la tessera, Selezionare l'operazione desiderata, Ritirare la tessera, e così via. Un'ultima difficoltà è data dal fatto che l'elenco dei casi d'uso deve essere il più possibile *completo*. Ciò permette di avere un quadro preciso, anche se ancora molto generale, del rapporto fra utente e sistema e della sua complessità funzionale. Per esempio, l'elenco completo dei casi d'uso di un tipico sportello Bancomat potrebbe essere il seguente:

- Preleva contante
- Visualizza il saldo del conto corrente
- Ricarica scheda prepagata del cellulare.

La nozione di caso d'uso, proposta da Ivar Jacobson a partire dal 1967, è ampiamente usata nell'ambito dell'ingegneria del software. La descrizione dei casi d'uso del sistema da realizzare è, come vedremo, un contenuto importante del documento dei requisiti. Pertanto, ne parleremo più estesamente nel capitolo 7, dedicato alla stesura di questo documento.

Progettazione universale

Nell'esempio del sistema audio-video, abbiamo visto come una progettazione che prenda le mosse dalle esigenze dell'utente debba tenere conto delle specifiche situazioni d'uso del sistema. Quanto maggiore è la conoscenza dei casi d'uso del sistema, tanto meglio riusciremo a progettarne l'usabilità. D'altra parte, nel capitolo precedente, abbiamo introdotto il concetto molto importante di usabilità universale (pag. 76). Vorremmo poter costruire sistemi che risultino usabili non solo per una specifica persona o gruppo di persone, ma anzi per *ogni* categoria di utenti, indipendentemente dalla loro classe sociale, etnia, lingua, cultura, collocazione geografica, dotazione tecnologica o eventuali disabilità. Un sistema universalmente usabile non discrimina e non divide – un mondo di prodotti universalmente usabili sarebbe certamente più equo.

La progettazione di sistemi universalmente usabili è stata chiamata *progettazione universale* (*universal design*) o più recentemente, in ambito europeo, *design for all* (abbreviato in *DfA*) o, ancora, *inclusive design*. Secondo la definizione di Ron Mace: [79]

> *Universal design è la progettazione di prodotti e ambienti usabili da tutte le persone, al massimo grado possibile, senza la necessità di adattamenti o progettazioni speciali.*

La filosofia del design universale non riguarda solo i sistemi interattivi, e può applicarsi a molti ambiti diversi, quali il disegno industriale, l'architettura, l'urbanistica, l'ambiente, le infrastrutture di mobilità e comunicazione. In effetti, il concetto (e il termine che lo denota) non nasce nell'informatica. Alla fine degli anni 90, un gruppo di architetti, designer industriali, ingegneri e ricercatori ambientali del Center for Universal Design della North Carolina State University,

[79] Ron Mace appartiene al Center for Universal Design della North Carolina State University, dove il concetto di universal design è stato inizialmente proposto, http://www.design.ncsu.edu/cud .

negli Stati Uniti, ha elaborato i seguenti sette principi generali del design universale, per orientare i progettisti, indipendentemente dal particolare ambito di progettazione (prodotti industriali, ambiente, comunicazioni, eccetera):[80]

1. Equità d'uso: il prodotto della progettazione è utile e vendibile a persone con abilità diverse.
2. Flessibilità d'uso: il prodotto della progettazione supporta un ampio spettro di preferenze e abilità individuali.
3. Uso semplice e intuitivo: l'uso del prodotto della progettazione è facile da comprendere, indipendentemente dall'esperienza, conoscenza, capacità linguistica o livello di concentrazione corrente dell'utente.
4. Informazione percepibile: il prodotto della progettazione comunica efficacemente l'informazione necessaria all'utente, indipendentemente dalle condizioni ambientali o dalle abilità sensoriali dell'utente.
5. Tolleranza agli errori: il prodotto della progettazione minimizza i rischi e le conseguenze avverse di azioni accidentali o non intenzionali.
6. Ridotto sforzo fisico: il prodotto della progettazione può essere usato in modo efficace, confortevole e con sforzo minimo.
7. Dimensione e spazio adatti all'uso e all'approccio: vengono forniti dimensioni e spazi appropriati per l'avvicinamento, la manipolazione e l'uso, indipendentemente dalla corporatura, postura o mobilità dell'utente.

Nel caso dei prodotti interattivi, la progettazione universale rappresenta una sfida difficile per il progettista. Una cosa è costruire rampe d'accesso a un edificio, utilizzabili anche da persone disabili, ben altra cosa è progettare un sistema software complesso usabile per tutti (pensiamo, per esempio, a un word processor, a un browser, a un sito web di e-commerce, a un mobile phone). Le difficoltà derivano sostanzialmente da tre ragioni:

1. Diversità delle tecnologie disponibili ai diversi utenti.

   Come si è già osservato, gli ecosistemi tecnologici sono in continuo, rapido cambiamento. Pensiamo per esempio ai personal computer. Nonostante la loro elevatissima standardizzazione, i sistemi utilizzati, in ogni momento, sono molto diversi fra loro. Le configurazioni e la potenza delle varie macchine sono molto differenti, e così le versioni di software installate. Ciò pone enormi problemi di compatibilità fra i sistemi e di diversità di prestazioni. Una discriminazione importante riguarda l'accesso alla rete. Gli utenti che dispongono di connessioni a banda larga possono usufruire di servizi che altri utenti, per limitazioni prestazionali, non possono, in pratica, utilizzare. Chi dispone di connessioni mobili gode di una flessibilità sconosciuta a chi può accedere alla rete solo da postazioni fisse, magari lontane dalla propria abitazione.
2. Diversità degli individui.

   Le persone sono molto diverse fra loro. Queste differenze rientrano approssimativamente in tre grandi categorie: fisiche, cognitive, socio-culturali. Le prime sono relative ai parametri che differenziano gli individui dal punto di vista del loro genere, della struttura corporea e delle loro prestazioni fisiche: altezza, peso, acuità visiva, udito, uso prevalente della mano destra o sinistra, capacità di coordinamento mani/occhi, e così via. Le seconde riguardano le capacità cognitive: memoria, attenzione, attitudine al ragionamento logico o analogico, capacità di ragionamaneti complessi, ecc. Le diversità socio-culturali sono legate agli ambienti sociali e culturali nei quali i vari individui operano e dai quali vengono formati. Nonostante la globalizzazione in atto, queste diversità fra le culture presenti sul pianeta permangono e, anzi, vengono coltivate e sviluppate, spesso in reazione ad essa.

3. Diversità nella capacità d'uso della tecnologia.

   Le persone sono diverse nella loro capacità di relazionarsi con i sistemi tecnologici. Per esempio, chi fa lavoro d'ufficio e utilizza correntemente delle applicazioni software ha un rapporto con la tecnologia molto diverso da chi, opera in aree rurali, nell'agricoltura o nell'allevamento. Esiste inoltre un forte *gap generazionale*, che differenzia la generazione di chi è entrato in contatto con le tecnologie digitali fin da bambino, e le generazioni precedenti. È nota

[80] Nostra traduzione in italiano dalla versione 2.0 (1997) dei *Principi del Design Universale*. Copyright © 1997 NC State University, The Center for Universal Design. Cfr. http://www.design.ncsu.edu/cud/index.htm. Ogni principio è corredato di opportune linee guida (29 in tutto), anch'esse molto generali.

la spontaneità con cui gli appartenenti alla cosiddetta *net-generation*[81] utilizzano le tecnologie digitali, anche senza un particolare addestramento.

Dal punto di vista generale, ci sono essenzialmente due strade per produrre un design universale. La prima (Figura 105 A) rappresenta l'approccio tradizionale. Esso consiste nel definire un utente "medio" (o "normale") del sistema, cioè quel tipo di utente al quale il prodotto prioritariamente s'indirizza. Si analizzano i suoi bisogni e i suoi contesti d'uso, e si progetta il sistema "per lui". Per gli altri utenti – chiamiamoli utenti "speciali" – si realizzeranno dei componenti ad hoc, separati dal sistema, i quali, in sostanza, lo adatteranno alle varie situazioni d'interesse. In sostanza, gli utenti speciali dovranno dotarsi di un'interfaccia aggiuntiva e accedere al sistema con la mediazione di questa interfaccia. Come abbiamo già visto, nel caso degli utenti con disabilità, queste interfacce speciali sono chiamate *tecnologie assistive* (lettori di schermo, tastiere *braille*, e così via). Già il termine indica che esiste una discriminazione: gli utenti speciali devono essere "assistiti" con dispositivi ad hoc. Questa è la strada finora maggiormente seguita per la costruzione di sistemi accessibili a utenti con disabilità.

Questo approccio pone diversi problemi. Progettando il sistema per l'utente medio, il progettista non è in grado di tenere conto fin dall'inizio delle necessità degli utenti diversi. Gli adattamenti potranno quindi rivelarsi complessi da realizzare, o comunque portare a risultati non soddisfacenti per gli utenti ai quali sono destinati. Per esempio, come adattare in modo soddisfacente un'interfaccia di tipo desktop a utenti non vedenti? Basta esaminare le funzioni che i vari sistemi operativi offrono per rendere accessibile il computer a utenti con disabilità visive, per convincersi che la cosa crea notevoli difficoltà. L'adattamento risulta complicato e farraginoso, sostanzialmente estraneo alla filosofia utilizzata nel design concept iniziale del sistema. Un altro tipo di difficoltà è dovuto alla necessità di mantenere la compatibilità fra il sistema e le diverse tecnologie assistive destinate agli utenti speciali.

A fronte di queste difficoltà, ha incominciato a farsi strada una filosofia di progettazione molto diversa. Secondo questo approccio (Figura 105 B), la progettazione non privilegia l'utente medio, ma tiene in considerazione fin da subito le necessità di *tutte* le categorie di utenti. Non serviranno adattatori o tecnologie assistive esterni al sistema: questi sarà in grado di adattare il proprio funzionamento a ogni specifica classe di utenti, una volta che le informazioni necessarie per questa personalizzazione gli siano state fornite, per esempio dall'utente stesso in una fase iniziale di configurazione (*sistema adattabile*).

[81] Con il termine *net-generation* (chiamata anche *generazione Y*) si suole indicare la generazione dei nati fra il 1977 e il 1997. Queste persone sono entrate in contatto con i personal computer e con internet fin da bambini, ed hanno sviluppato un rapporto immediato e spontaneo con queste tecnologie, che considerano quasi parte dell'ambiente naturale. Questo rapporto particolare con la tecnologia della net-generation (e della generazione successiva) è stato studiato da varie parti. Si veda, per esempio, il best-seller di Don Tapscott, *Grown Up Digital – How the Net Generation is Changing Your World*, McGraw-Hill, 2009. Vedi anche http://dontapscott.com.

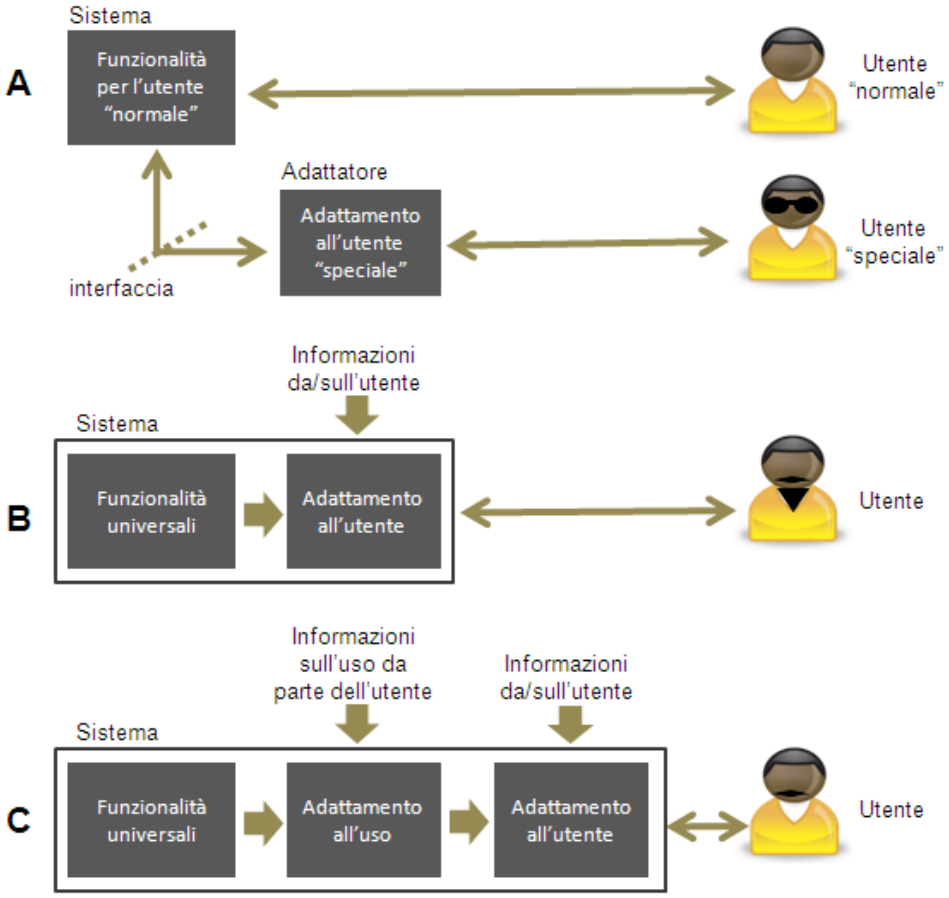


Figura 105. Possibili soluzioni per la progettazione universale: uso di adattatori (A); sistema adattabile (B); sistema adattativo (C)

È evidente che i due approcci sono molto diversi. Il primo (A) semplifica l'attività di progettazione, che deve considerare una casistica limitata agli utenti "normali". Ma, come abbiamo visto, non è esente da difficoltà, che sono semplicemente trasferite ad altri, cioè ai progettisti dell'adattatore. Il quale potrà porre notevoli difficoltà, nel caso in cui l'interfaccia fra questo e il sistema non sia stata ben concepita in anticipo. Il secondo approccio (B) richiede che tutti i problemi siano affrontati e risolti in modo sistematico, fin dall'inizio. Questa strategia è tipicamente più complessa, anche se, ovviamente, le difficoltà dovranno essere esaminate nelle specificità dei diversi sistemi.

Esiste, infine, una terza possibilità, ancora più sofisticata (Figura 105C). In questo caso il sistema, oltre ad essere personalizzabile in fase di configurazione iniziale, è in grado di monitorare con continuità i comportamenti e le modalità d'uso dell'utente, e di adattarvisi modificando il proprio comportamento. In sostanza, il sistema *impara durante l'uso*. Esempi tipici di questi sistemi, che si dicono *adattativi*, sono quelli che devono trattare input molto variabili da utente a utente, come i sistemi di riconoscimento vocale, o di riconoscimento della scrittura. Questi devono normalmente portare a termine una fase di *addestramento iniziale*, per apprendere le caratteristiche comportamentali dello specifico utente (che quindi si dovrà identificare a ogni uso). Superata questa fase iniziale, l'addestramento continua durante gli utilizzi successivi, con l'aiuto dell'utente. Questi dovrà, in qualche modo, correggere il sistema ogni volta che non si comporterà in modo soddisfacente, per arricchire la casistica dei comportamenti ad esso noti.

I tre approcci non sono alternativi, ma complementari, e possono essere in qualche misura compresenti in uno stesso sistema. In definitiva, potremmo definire la progettazione universale come l'approccio secondo il quale i sistemi sono progettati per essere sufficientemente intelligenti da adattarsi alle richieste o alle modalità di utilizzo dei loro diversi utenti o, quando ciò non sia possibile o troppo complesso, da permettere un facile interfacciamento con adattatori speciali.

Livelli di maturità della progettazione

La discussione precedente suggerisce che system-centred design e human-centred design non dovrebbero essere considerati due approcci alternativi, fra i quali scegliere secondo le situazioni. Lo human-centred design può essere considerato un approccio più maturo, che contiene al suo interno le problematiche tecniche del system-centred design, ma le inserisce in un contesto più ampio, che ci permette di comprendere in modo più approfondito le finalità del sistema. Possiamo, in effetti, classificare le attività di progettazione in differenti *livelli di maturità*:

- Primo livello di maturità**:** il prodotto funziona.

A questo livello, il progettista si occupa principalmente della risoluzione di problemi di natura tecnologica, e si accontenta che le funzioni previste nel sistema siano operative, e non ci siano errori di funzionamento. Questo è il livello più elementare, in cui si accetta di realizzare un sistema anche rudimentale, purché permetta di eseguire alcuni compiti ritenuti importanti. È il livello in cui sono superate le difficoltà tecniche basilari, e non ci si preoccupa se, per utilizzare il sistema, si devono porre in essere particolari accorgimenti o accettare delle limitazioni. A questo livello si collocano spesso i prototipi realizzati con finalità di ricerca, il cui obiettivo non è tanto quello di realizzare un prodotto rifinito in tutti gli aspetti, quanto quello di dimostrarne la fattibilità tecnica.

- Secondo livello di maturità: il prodotto fornisce le funzionalità necessarie.

A questo livello, il sistema non soltanto funziona, ma realizza *tutte* le funzionalità ritenute necessarie per gli scopi per cui è concepito. L'attenzione del progettista è posta sulla completezza e sulla qualità delle funzioni del sistema, di cui cura l'affidabilità, le prestazioni, la flessibilità, la modularità. Da parte dell'utente, che nell'ambito del progetto riveste sostanzialmente un ruolo di comparsa, ci si aspetta che esegua disciplinatamente le operazioni specificate nel manuale d'uso, possibilmente senza commettere errori. È il livello del system-centred design, in cui si colloca l'ingegneria tradizionale. Ne abbiamo visto un esempio nel sistema audio-video discusso in precedenza.

- Terzo livello di maturità**:** il prodotto è facile da usare.

Questo è il livello della progettazione human-centred. Non solo il prodotto funziona e offre tutte le funzionalità richieste, ma le organizza in modo adeguato rispetto alle tipologie e alle necessità dei suoi utenti, nei diversi contesti d'uso. La progettazione parte dall'analisi delle caratteristiche degli utenti cui il sistema è destinato, e dalla definizione precisa dei casi d'uso nei vari contesti. Nella mente del progettista l'utente, da comparsa, diviene protagonista, e si cerca di assecondarne nel modo migliore i comportamenti, i gusti, le idiosincrasie. L'utente è pesantemente coinvolto nel processo di progettazione, anche come soggetto attivo nella concezione, sperimentazione e valutazione del sistema.

- Quarto livello di maturità: il prodotto è invisibile durante l'uso.

Questo è il livello al quale ogni bravo progettista dovrebbe tendere. In questo caso il prodotto funziona, fornisce tutte le funzionalità richieste, è usabile e, inoltre, s'integra in modo così armonico e poco intrusivo con i comportamenti del suo utente che questi, durante l'uso, *non si accorge di usarlo*. In altre parole, esso permette all'utente di concentrare la propria attenzione sul compito che sta eseguendo, e non sullo strumento che lo supporta. Lo strumento diventa, per così dire, *invisibile*. Così, quando usiamo una buona penna per scrivere una lettera, siamo concentrati sul testo della lettera e non sullo strumento che utilizziamo per scriverlo. La penna è sostanzialmente invisibile, diventa una sorta di *protesi* dell'utente il quale, come avviene con ogni protesi ben progettata, tende a non essere consapevole della sua esistenza. Ne percepisce la presenza solo quando qualcosa non va, per esempio quando l'inchiostro termina o il pennino è rovinato. Allora, e soltanto allora, l'attenzione dell'utente viene sottratta al compito e rivolta allo strumento.

Un altro esempio tipico (ce ne sono molti) è il posto di guida di un'automobile. Durante la guida, l'attenzione di un guidatore esperto, se i comandi dell'auto sono ben progettati, sarà rivolta alla strada e non ai comandi stessi, che manovrerà in modo pressoché automatico. Ottenere questo risultato (che potremmo chiamare *design protesico,* da protesi) costituisce l'obiettivo più sfidante per il progettista dell'interazione. Questo livello richiede, di solito, una lunga esperienza nella progettazione di prodotti simili: un prodotto maturo è quasi sempre il risultato di una lunga evoluzione,

nella quale i difetti delle versioni precedenti vengono corretti nelle versioni successive, in un processo evolutivo simile, per certi versi, all'evoluzione naturale.

Chi è l'interaction designer

Le discussioni precedenti fanno comprendere che le competenze richieste a un *interaction designer* (il progettista dell'interazione) sono molto diverse – perché più ampie – da quelle richieste a un *system designer* (il progettista dei sistemi). Mentre quest'ultimo dovrà possedere essenzialmente competenze di natura tecnologica nel dominio cui appartiene il sistema da progettare e progettuale (i metodi e gli strumenti da utilizzare nelle attività di progettazione), il primo dovrà essere in grado di analizzare e comprendere le caratteristiche e i bisogni dell'utente per definire, a partire da queste, le modalità d'interazione più opportune.

L'Interaction Design Association (IxDA), che raccoglie più di 10.000 membri in tutto il mondo, nel suo sito web descrive in questo modo l'interaction design, e la professione dell'interaction designer[82]:

> *L'interaction design (IxD) è una disciplina professionale che chiarisce la relazione fra le persone e i prodotti interattivi che esse utilizzano. Pur avendo solide fondamenta nella teoria, pratica e metodologia del design tradizionale, essa si occupa specificamente della definizione dei dialoghi complessi che avvengono fra le persone e i dispositivi interattivi di ogni tipo – dai computer agli apparati per la comunicazione mobile agli elettrodomestici.*
>
> *Gli interaction designer si sforzano di creare prodotti e servizi utili e usabili. Seguendo i principi fondamentali della progettazione centrata sull'utente, la pratica dell'interaction design si fonda sulla comprensione degli utenti reali – dei loro obiettivi, compiti, esperienze, necessità e desideri. Affrontando la progettazione da una prospettiva centrata sull'utente, tentando di bilanciare le necessità dell'utente, gli obiettivi di business e le possibilità tecniche, gli interaction designer producono soluzioni a sfide progettuali complesse, e definiscono prodotti e servizi interattivi nuovi ed evolutivi.*

Questo richiede competenze e sensibilità specifiche, che non sono fornite nel curriculum formativo di un designer industriale o di un progettista software. Infatti, i sistemi odierni sono sempre più complessi, e l'interazione non è soltanto quella "fisica" considerata dall'ergonomia tradizionale (postura, sforzo, illuminazione, ecc.), ma è – soprattutto – di tipo cognitivo. L'ergonomia diventa, quindi, *ergonomia cognitiva*, e il compito dell'interaction designer è, anche, quello di conoscere e assecondare i meccanismi cognitivi coinvolti nell'interazione utente-sistema, in modo che ne risultino sistemi gradevoli e facili da usare. L'interaction designer utilizza quindi competenze e metodi provenienti da varie discipline, con un atteggiamento fortemente multidisciplinare (Figura 106).

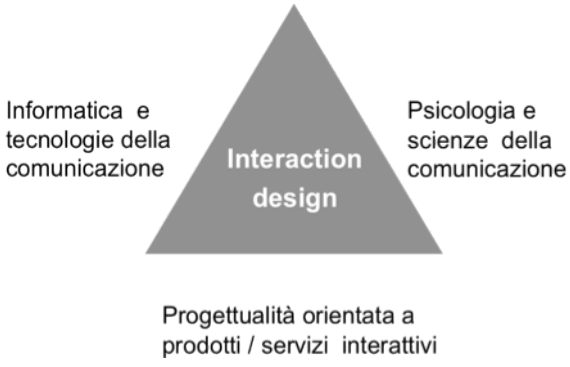


Figura 106. Multi-disciplinarità dell'interaction design

Dovrà avere una conoscenza di base dei principali meccanismi della percezione e della cognizione (visione, udito, tatto, memoria, attenzione, …), dei meccanismi della comunicazione umana (il linguaggio, la comunicazione scritta, visiva e

[82] Nostra traduzione da http://www.ixda.org.

multimediale,…) e della psicologia sociale. Dovrà conoscere le potenzialità della tecnologia (componenti di base, apparati utilizzati nella comunicazione uomo-macchina e nella comunicazione umana mediata dalla tecnologia, e loro possibilità). Dovrà essere in grado di comprendere le necessità degli utenti nel contesto delle loro attività, individuarne le diverse tipologia e analizzarne compiti. Dovrà, infine, conoscere i metodi e le tecniche della progettazione e dell'ingegneria dell'usabilità. Dovrà possedere una mentalità aperta all'innovazione e attitudini creative. Uno "strano incrocio", si potrebbe dire, fra uno psicologo, un artista e un ingegnere.

Il mestiere dell'interaction designer si declina poi in diverse specializzazioni, a seconda del tipo di prodotto oggetto della progettazione: dai prodotti tecnologici di largo consumo, fino ai siti e alle applicazioni web (*web designer*).

Ripasso ed esercizi

1. In che senso la progettazione human-centred è diversa dalla progettazione intesa in senso tradizionale (progettazione centrata sul sistema)?
2. Spiega il concetto di caso d'uso, e la differenza fra caso d'uso e funzionalità.
3. Considera un elettrodomestico di casa tua che utilizzi spesso (il fornello, il forno a microonde, la lavatrice, la lavapiatti, o altro), ed elencane i casi d'uso, facendo riferimento alla tua esperienza specifica. Esaminandone l'usabilità in rapporto alle tue esigenze, potresti affermare che tale elettrodomestico sia stato progettato con un processo human-centred? Perché?
4. Che cosa si intende per "progettazione universale"? Quali sono gli approcci progettuali possibili?
5. Individua alcuni prodotti "invisibili durante l'uso" nel senso discusso in questo capitolo.
6. Perché la progettazione human-centred richiede necessariamente un atteggiamento multi-disciplinare?

Approfondimenti e ricerche

1. La nota di Donald Norman *The perils of Home Theatre* descrive il punto di vista di questo autore sul design degli apparati di home theatre (http://jnd.org/dn.mss/the_perils_of_home_theater.html).
2. Oltre alla nota di cui sopra, il sito di Donald Norman (http://jnd.org) contiene un'ampia collezione di scritti di grande interesse sul tema dello human-centred design e della progettazione dei sistemi interattivi.
3. Indicazioni per approfondire la nozione di caso d'uso sono fornite nella sezione Approfondimenti e ricerche del capitolo 7.
4. Per una rassegna sul concetto di e-Inclusione e di Design-for-All, e sui progetti Europei sull'argomento, si veda P.L.Emiliani, Inclusione nella Società dell'Informazione, in A.Soro (ed.), Human Computer Interaction – Fondamenti e prospettive, ed Polimetrica, 2008 (in rete), pagg. 47-109. Un'altra risorsa molto utile per il design universale e i problemi di accessibilità è il sito del Trace Research & Development Center dell'Università del Wisconsin-Madison (http://trace.wisc.edu), ricco di documenti e link ad altre risorse sull'argomento.

6. L'ingegneria della usabilità

Sintesi del capitolo

Questo capitolo introduce i concetti base dell'ingegneria dell'usabilità, che saranno approfonditi nei capitoli successivi. Viene ricordato il tradizionale modello dei processi di progettazione e sviluppo "a cascata" adottato inizialmente dall'ingegneria del software, e ne viene motivato il fallimento. Quindi, dopo avere discusso il cosiddetto ciclo compito-artefatto, si introduce il modello iterativo di progettazione e sviluppo, e se ne discutono brevemente vantaggi e svantaggi. Viene quindi descritto più specificamente il modello per i processi di human-centred design proposto dallo standard ISO 13407. Dopo un esempio di applicazione di questi concetti nella progettazione e sviluppo di siti web, si discute brevemente la problematica del rapporto costi-benefici nell'adozione degli approcci dell'ingegneria dell'usabilità.

Le diverse ingegnerie

Il termine *ingegneria* può essere definito in molti modi. Per esempio, il *WordNet 2.0 Dictionary* definisce l'ingegneria, un po' sbrigativamente, come "la disciplina che si occupa dell'arte o della scienza di applicare la conoscenza scientifica a problemi pratici". Più specificamente, l'associazione degli ingegneri americani (American Engineer's Council for Professional Development), la definisce come

> *L'applicazione creativa di principi scientifici alla progettazione o allo sviluppo di strutture, macchine, apparati o processi di produzione, o di opere che li utilizzano singolarmente o in combinazione; o alla costruzione o esercizio degli stessi con piena consapevolezza del loro progetto; o alla previsione del loro comportamento in specifiche condizioni operative; tutto ciò in relazione a desiderate funzioni, economie di esercizio e obiettivi di sicurezza verso la vita o la proprietà.*

Queste definizioni molto generali possono essere declinate in molti modi, in relazione ai diversi settori di applicazione. Per esempio, *l'ingegneria del software* si occupa dei metodi e delle tecniche per la progettazione e realizzazione di sistemi software di qualità, senza sprechi. Essa, nata negli Stati Uniti una quarantina di anni fa sulla spinta dei grandi progetti software di origine militare[83], si è occupata, tradizionalmente, di sistemi software molto complessi, il cui sviluppo coinvolge diecine di persone (o più). Pertanto, non stupisce che, all'inizio, questa disciplina abbia seguito approcci molto strutturati e formali, definendo e studiando processi di progettazione e sviluppo organizzati in fasi predefinite, con grande enfasi sulle attività di pianificazione e di specifica. In seguito, questi modelli si sono profondamente evoluti, verso modelli iterativi di vario tipo, o modelli relativamente poco strutturati e "agili".

Più recentemente è entrato nell'uso il termine *ingegneria dell'usabilità* (*usability engineering*), per denotare la disciplina che studia le tecniche, i metodi e i processi che possono essere utilizzati per progettare e sviluppare sistemi usabili. Il termine, entrato nell'uso soprattutto per merito del libro di Jakob Nielsen *Usability Engineering*, del 1994, era stato introdotto nel 1986 da alcuni progettisti della Digital Equipment Corporation, in un'accezione che enfatizzava fortemente un approccio quantitativo alla definizione degli obiettivi di usabilità nella progettazione:

> *L'ingegneria dell'usabilità è un processo, basato sull'ingegneria classica, che consiste nello specificare, quantitativamente e in anticipo, quali caratteristiche e in qual misura il prodotto finale da ingegnerizzare dovrà possedere. Questo processo è seguito dall'effettiva realizzazione del prodotto, e dalla dimostrazione che esso effettivamente possiede le caratteristiche pianificate. L'ingegneria non è il processo di costruire un sistema perfetto con risorse infinite. Piuttosto, l'ingegneria è il processo di costruire economicamente un sistema funzionante che soddisfa una necessità. In assenza di specifiche misurabili di usabilità, non c'è alcun*

[83] Il termine *software engineering* è stato usato per la prima volta nella storica NATO Software Engineering Conference, tenutasi a Garmish, in Germania, nell'ottobre 1968

modo di determinare le esigenze di usabilità di un prodotto, o di misurare se il prodotto soddisfi o meno tali esigenze. Se non possiamo misurare l'usabilità, non possiamo avere un'ingegneria dell'usabilità. [84]

La parola ingegneria vuole sottolineare l'approccio pragmatico e basato su fondamenti scientifici di questa disciplina, che si propone di dare indicazioni concrete e operative a chi abbia il compito di progettare e sviluppare sistemi interattivi. Inizialmente, l'ingegneria dell'usabilità si è focalizzata sul design dell'interfaccia utente dei sistemi software. Oggi, questo termine viene usato in un'accezione più ampia, che comprende la totalità delle pratiche utilizzate nel processo di progettazione e sviluppo dei sistemi interattivi, a partire dalla raccolta e analisi iniziale dei requisiti. Al di là delle specifiche definizioni ed enfasi date dai diversi autori negli anni, i principi cardine di questa disciplina possono considerarsi ben consolidati fin dalla metà degli anni 80. In estrema sintesi, essi si possono riassumere nei seguenti tre punti chiave:

1. Focalizzazione sull'utente, all'inizio e durante tutto il processo di progettazione;
2. Prove con l'utente durante l'intero processo di progettazione, con analisi qualitative e misure quantitative;
3. Modello di progettazione e sviluppo iterativo, per prototipi successivi[85].

Senza entrare in appofondimenti non rilevanti in questo contesto, poniamo a confronto le idee essenziali dell'approccio tradizionale all'ingegneria del software (il cosiddetto modello "a cascata"), e dell'approccio più moderno, basato su processi di sviluppo iterativo, per prototipi successivi, adottato nell'ingegneria dell'usabilità.

Il modello "a cascata"

Un tempo, quando la disciplina dell'ingegneria del software era agli esordi, si pensava che per realizzare un progetto di successo fosse necessario procedere per fasi logiche ben sequenziate, ognuna delle quali ponesse le basi per la fase successiva. Si partiva dalla raccolta dei requisiti, poi si definivano le specifiche del sistema da realizzare. Quindi si progettava l'intero sistema "sulla carta" e lo si codificava nel linguaggio di programmazione prescelto. Lo si collaudava e infine lo si rilasciava. Si passava alla fase successiva solo quando la precedente era completata e i suoi "prodotti" approvati formalmente dal committente.

Si pensava che in un processo ordinato, condotto da professionisti e guidato da un capo progetto esperto, non si dovesse mai retrocedere. Per descrivere questo processo si usa la metafora della cascata: come in una cascata l'acqua scorre soltanto verso il basso e non torna mai indietro, così dalla fase iniziale di un progetto si arriva, passo passo, al rilascio del sistema senza ritornare mai sui passi precedenti (*waterfall model*, Figura 107).

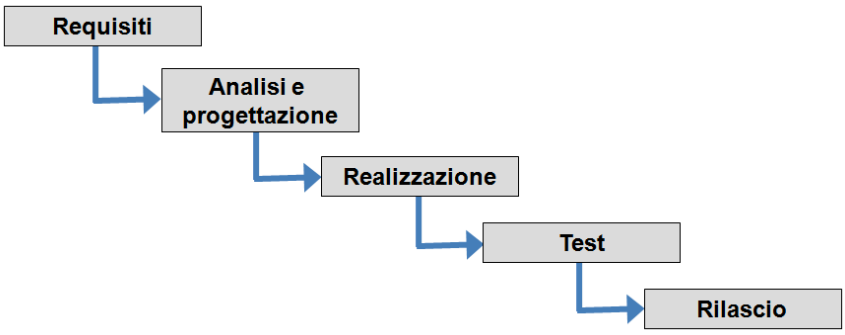


Figura 107. Processo di progettazione e sviluppo "a cascata"

[84] M.Good, T.M.Spine, J.Whiteside, P.George, *User-derived impact analysis as a Tool for Usability Engineering*, in Proceedings CHI 1986

[85] Questi principi sono stati per la prima volta proposta nel 1985 da J.Gould e C.Lewis, *Designing for Usability: Key principles and what designers think*, in Communications of the ACM, 28(3), e parzialmente riformulati in successivi articoli degli autori. Per una analisi storica e critica dei tre principi, nell'ottica odierna, si veda G.Cockton, *Revisiting Usability's Three Key Principles*, in Proceedings CHI 2008.

Questa impostazione sembra molto sensata, quasi ovvia: per costruire qualcosa (una casa, un ponte, un'automobile, un sito web) bisogna prima decidere che cosa si vuole ottenere e descriverlo dettagliatamente; poi si passerà alla sua realizzazione, quindi al collaudo finale e alla consegna al committente. Eppure ci si accorse ben presto che non funzionava sempre così: nella pratica, in nessun progetto reale, anche se ben gestito, le cose procedevano in maniera così semplice e lineare. Si rendeva spesso necessario ritornare sui passi precedenti, per rivedere e modificare decisioni già prese, anche se erano state ritenute assolutamente consolidate.

Le cause potevano essere molteplici: il committente, in fase avanzata di realizzazione, richiedeva delle varianti che modificavano le specifiche già approvate. Oppure i progettisti scoprivano difficoltà tecniche inattese, che consigliavano di cambiare rotta. Oppure, ancora, magari nella fase di rilascio del sistema, i primi utenti segnalavano delle difficoltà nell'uso che non erano state previste da nessuno e richiedevano cambiamenti consistenti. Tutti questi rifacimenti, non previsti nella pianificazione iniziale, producevano costi aggiuntivi anche considerevoli. I budget inizialmente assegnati venivano immancabilmente disattesi. Per molto tempo, queste difficoltà furono imputate a una cattiva conduzione dei progetti. Era compito di un buon capo progetto, si diceva, tenere a freno le richieste dei committenti e degli utenti e far loro comprendere l'importanza di controllare accuratamente le specifiche e di accettare che, una volta approvate, queste dovessero essere considerate "congelate" fino al rilascio del sistema.

Con la maturazione della disciplina dell'ingegneria del software, e dopo molti anni e molti fallimenti, si capì che le cose non funzionavano, perché *non possono* funzionare così. Ci si rese conto che nessun sistema complesso può essere realizzato con il modello della cascata, perché è impossibile specificarne tutti gli aspetti all'inizio e poi realizzarlo senza modificare nulla. Le ragioni di questa impossibilità sono sia di carattere pratico, sia teorico-concettuale.

Dal punto di vista pratico, è molto difficile prevedere "sulla carta" *tutti* gli aspetti di un sistema complesso, che non esiste ancora. Possiamo (e dobbiamo) tentare di farlo, ma inevitabilmente non saremo in grado di anticipare tutti i problemi che incontreremo durante la realizzazione, per risolvere i quali potremo essere costretti a cambiare rotta. Queste difficoltà non si verificano soltanto nel software, ma in progetti di ogni tipo. Pensiamo, per esempio, al progetto di ristrutturazione di un appartamento. Anche in questo caso inizieremo con una descrizione "sulla carta" delle opere murarie e degli impianti da realizzare. Se il modello a cascata funzionasse bene, giunto a questo punto, il committente potrebbe disinteressarsi del cantiere e affidarlo a un buon direttore dei lavori, che gli consegnerà alla fine l'appartamento realizzato esattamente come da specifiche. Chi ha fatto questa esperienza, tuttavia, sa bene che le cose non funzionano così. Sa che, durante i lavori, s'incontrano difficoltà non previste e *non prevedibili.*

Per risolvere queste difficoltà può essere necessario cambiare le specifiche iniziali e realizzare un appartamento diverso, per qualche aspetto, da quello progettato inizialmente. Una soletta si rivela poco resistente e occorre rinforzarla con una putrella. Questa impedisce il passaggio dei tubi dell'impianto di riscaldamento dove era previsto: di conseguenza, il calorifero dovrà essere installato in un posto diverso. Oppure, a lavori avviati, ci accorgiamo che il vicino ha l'abitudine di ascoltare musica fino a tardi e decidiamo di insonorizzare la parete con uno strato di materiale isolante. Questo modifica, anche se solo di pochi centimetri, le misure della stanza e bisogna rivedere alcune decisioni sulla posizione dei mobili. E così via: le varianti in corso d'opera potrebbero essere diecine. Non necessariamente dovute a errori di progettazione, ma a situazioni oggettive che non potevano essere previste e che impongono delle modifiche senza le quali il risultato non sarebbe accettato dal committente. Il direttore dei lavori non potrà certo rifiutarsi di realizzarle, appellandosi al progetto iniziale regolarmente approvato.

Il ciclo compito-artefatto

C'è anche un motivo più profondo, di natura teorico-concettuale, che fa sì che il modello a cascata non possa funzionare. Questo motivo è racchiuso in un principio generale, che abbiamo già incontrato varie volte (Figura 11, pag.20 e, in altra forma, Figura 99, pag.112), e che possiamo enunciare nel seguente modo:

Ogni nuovo strumento cambia i bisogni del suo utilizzatore e genera nuovi bisogni, che suggeriscono modifiche non previste allo strumento stesso.

In altre parole, per soddisfare le nostre necessità, produciamo strumenti che, a loro volta, generano nuovi bisogni. Costruiamo allora nuovi strumenti, o modifichiamo quelli già disponibili, in un ciclo evolutivo infinito, al quale è stato

dato il nome di *task-artifact cycle*.[86] Questo principio vale per ogni strumento, semplice o complesso, dal cacciavite al cruscotto di un jumbo jet, a un sistema informativo (Figura 108).



Figura 108. Il ciclo compito-artefatto

Quando definiamo i requisiti di un prodotto che non esiste ancora e che vogliamo realizzare, lo facciamo tenendo conto di determinati bisogni insoddisfatti. Per ottenere questo risultato, noi progettiamo il prodotto ipotizzando degli scenari d'uso che ci sembrano plausibili e realizzando quelle funzioni che, *nelle nostre ipotesi*, ci sembrano necessarie. Anche se siamo degli ottimi progettisti, non potremo mai essere certi di avere immaginato correttamente come i nostri utenti utilizzeranno effettivamente il sistema negli specifici contesti d'uso e come questo modificherà i loro bisogni. Per verificare la correttezza delle nostre ipotesi, dobbiamo prima realizzare il prodotto, farlo usare agli utenti e osservare come lo utilizzeranno effettivamente, nelle diverse specifiche situazioni. Ci potremo allora accorgere che gli scenari immaginati corrispondono quasi, ma non completamente, all'uso effettivo. Ma soprattutto potrà capitare che l'interazione fra utente e prodotto faccia nascere nuovi bisogni, in modi imprevisti. Tutto questo ci suggerirà di modificare il prodotto: senza queste modifiche, i nostri utenti non saranno soddisfatti. Come scrisse Douglas Engelbart, uno dei pionieri della human-computer interaction, già più volte ricordato, "non appena viene introdotto un nuovo manufatto, inizia una co-evoluzione del manufatto e di chi lo usa."

In sostanza, non è possibile valutare completamente l'adeguatezza dello strumento ai suoi utenti, prima che questi lo usino effettivamente. Ecco perché il modello a cascata tradizionale non può funzionare. Esso prevede che gli utenti siano coinvolti nel processo solo in due momenti: all'inizio, per contribuire a requisiti e specifiche e alla fine, dopo il rilascio (o tutt'al più per il collaudo). Tuttavia, nella stesura delle specifiche iniziali, anche gli utenti, come i progettisti, non possono far altro che ipotizzare le caratteristiche necessarie. Alla fine, quando la correttezza di queste assunzioni può essere verificata in concreto, è troppo tardi per intervenire: il sistema è già stato sviluppato.

Modelli iterativi

Se il modello a cascata è inadeguato, ci serve un modello diverso, che coinvolga gli utenti fin da subito, non solo nella stesura di requisiti e specifiche, ma anche, e soprattutto, per sperimentare l'uso di versioni preliminari del sistema e aiutarci, con le loro reazioni e le loro indicazioni, a correggere il tiro, in un processo di prove e aggiustamenti successivi.

L'idea è di procedere con la realizzazione di una serie di *prototipi*, via via più vicini al sistema finale. S'inizia con un prototipo preliminare, realizzabile a costi ridotti, e lo si sottopone all'utente, che *prova a usarlo*. Questa prima prova sarà normalmente limitata, perché il sistema sarà molto semplificato, con funzioni realizzate solo parzialmente, o addirittura simulate in qualche modo. Tuttavia ci permetterà di verificare alcune assunzioni di partenza ed eventualmente di aggiustare il tiro. Un po' come quando un pittore schizza un bozzetto prima di dipingere il quadro, per averne un'idea di massima ed eventualmente per farlo approvare dal committente. Si realizza quindi un nuovo prototipo, sempre incompleto, ma un po' più somigliante al sistema finale e lo si sottopone ancora alla prova degli

[86] Cfr. J.M.Carroll, W.A.Kellogg, M.B.Rosson, *The Task-Artifact Cycle*, in J.M.Carroll (ed.), *Designing Interaction – Psychology at the Human computer Interface*, Cambridge University Press, 1991.

utenti, e così via, per approssimazioni successive, fino alla conclusione del progetto. In sostanza, *le prove d'uso diventano parte integrante del processo di progettazione*. La Figura 109 mostra una schematizzazione di questo modo di procedere.

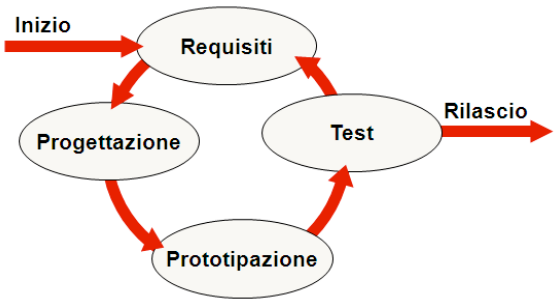


Figura 109. Processo di progettazione e sviluppo per prototipi successivi

Ovviamente, nelle varie iterazioni, le diverse attività mostrate in figura avranno pesi diversi. Per esempio, al primo giro, dopo avere specificato i requisiti, ci si concentrerà sulle attività di progettazione, mentre le attività di realizzazione del primo prototipo richiederanno sforzi limitati. Il primo prototipo sarà infatti piuttosto rudimentale: in molti casi, soltanto un *mock-up* con il quale effettuare un primo confronto con gli utenti e, naturalmente, con il committente. Questo confronto sarà condotto nella fase di test in figura. Al giro successivo, sulla base dell'esito del confronto, si apporteranno le necessarie modifiche ai requisiti e al progetto, e si realizzerà un secondo prototipo, più evoluto. In questo secondo giro, lo sforzo dedicato ai requisiti – se non sono stati evidenziati grossi problemi – sarà di solito piuttosto limitato (serviranno solo alcuni ritocchi), mentre la realizzazione del secondo prototipo sarà più impegnativa. Anche il test effettuato al secondo giro, con un prototipo più evoluto, richiederà maggiori sforzi. E così via: all'avanzare del progetto, in sostanza, lo sforzo complessivo si sposta progressivamente dalle fasi iniziali del ciclo tradizionale (requisiti e progettazione) alle fasi finali (test e rilascio).

In pratica, tutte le attività rappresentate in Figura 109 vengono portate avanti "in parallelo" per tutta la durata del progetto, ma l'impegno dedicato a ciascuna di esse cambia nel tempo. L'avanzamento del progetto non è più scandito dal passaggio da un'attività alla successiva, ma dalla realizzazione dei diversi prototipi. A ogni iterazione un'attività prevale sulle altre, ma tutte sono comunque portate avanti, anche solo per apportare le modifiche rese necessarie dai test con gli utenti.

Questa situazione è visualizzata nella Figura 110, che mostra, per un progetto ipotetico, l'andamento nel tempo dell'impegno di risorse sulle singole attività (per esempio, il numero di persone impegnate).[87]

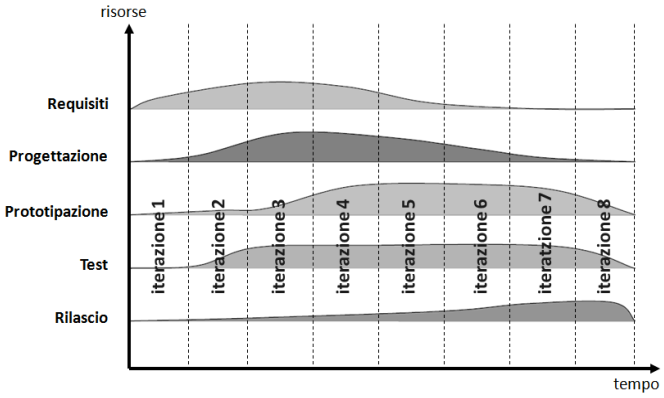


Figura 110. Allocazione delle risorse secondo il modello iterativo

[87] Cfr. I.Jacobson, G.Booch, J.Rumbaugh, *The Unified Software Development Process,* Addison-Wesley, 1999

Il processo di progettazione per prototipi successivi è il modello concettualmente corretto per la realizzazione di sistemi complessi: il prodotto si vede (anche se in modo parziale), fin dall'inizio e viene perfezionato per incrementi successivi; le scelte effettuate possono essere sperimentate anticipatamente e si possono scartare quelle sbagliate.

A fronte di questi grandi vantaggi, esiste il rischio che il processo diverga, a causa delle richieste di modifiche che nascono durante le attività di valutazione dei vari prototipi. Per evitare queste difficoltà, per ogni progetto sarà quindi necessario pianificare il processo iterativo di progettazione e sviluppo in modo che, per così dire, non sfugga di mano. Si tratterò, in sostanza, di prevedere fin dall'inizio la successione dei diversi prototipi da realizzare, in funzione degli scopi specifici che con ciascuno di essi si desidera raggiungere. Ovviamente, questa pianificazione dovrà tenere conto degli obiettivi e caratteristiche del sistema da realizzare: non è possibile stabilire un modello di processo generale, valido per tutti i sistemi, poichè ciascun progetto ha esigenze specifiche. Alla conclusione di questo capitolo vedremo un esempio di pianificazione dello sviluppo iterativo per i siti web.

Il modello ISO 13407

Il modello iterativo, presentato in Figura 109 in modo del tutto generale, può essere precisato e perfezionato in vari modi, che in questa sede non è possibile analizzare. Nell'ambito dell'ingegneria dell'usabilità assume una particolare autorevolezza, la descrizione che ne dà lo standard ISO 13407, dal titolo *Human-centred design processes for interactive systems*, che ha proprio lo scopo, come si legge nella sua introduzione, di "fornire una guida alle attività di progettazione centrata sull'essere umano lungo il ciclo di vita dei sistemi interattivi basati su computer".

In questo standard, il modello iterativo di Figura 109 è rappresentato, più specificamente, come in Figura 111.

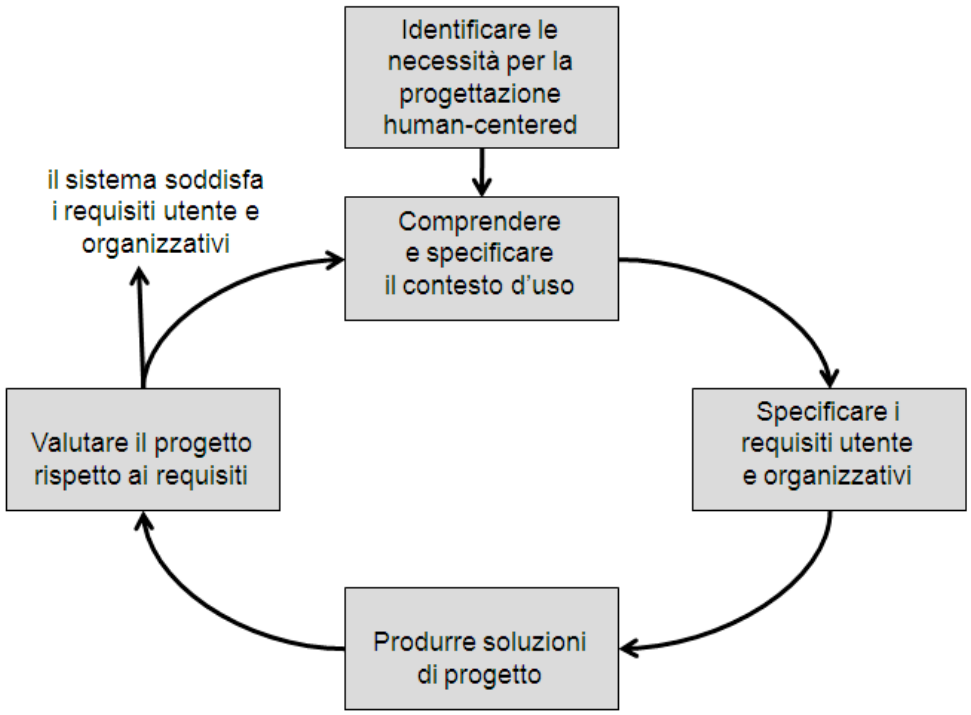


Figura 111. Il processo di progettazione human-centred secondo l'ISO 13407

L'ISO 13407 non descrive una metodologia specifica (non può essere questo lo scopo di uno standard), ma fornisce linee guida ampie dettagliate a chi desideri organizzare un processo di progettazione human-centred. Nel seguito di questa sezione riassumiamo la descrizione dello schema di Figura 111 contenuta nell'ISO 13407, facendo riferimento alla versione del 1999.[88] I vari aspetti del processo di progettazione human-centred verranno quindi ulteriormente approfonditi nei prossimi capitoli di questo libro.

Comprendere e specificare il contesto d'uso

Il contesto in cui il sistema sarà utilizzato è definito dalle caratteristiche degli utenti, dei compiti e dell'ambiente fisico e organizzativo. È importante identificare e comprendere i dettagli di questo contesto, per orientare le decisioni iniziali del progetto, e per fornire una base per la loro successiva convalida. Tutto ciò, sia che si debba progettare un sistema nuovo, sia che si debba modificare un sistema esistente per migliorarlo. In questo secondo caso, potranno essere molto utili le informazioni sulle reazioni degli utenti provenienti dai rapporti dell'help desk o da specifiche indagini esplorative.

La descrizione del contesto d'uso del sistema dovrebbe comprendere i seguenti argomenti:

- Le caratteristiche degli utenti.

  Le caratteristiche rilevanti potrebbero essere le competenze, le abilità, le esperienze, la formazione, le caratteristiche fisiche, le abitudini, le preferenze. Spesso sarà necessario classificare gli utenti in diverse categorie, per esempio in funzione dei diversi ruoli nei confronti del sistema o dei differenti livelli di esperienza.
- I compiti che gli utenti dovranno eseguire.

  Dovrebbero essere analizzati i compiti che possono influenzare l'usabilità del sistema, per esempio indicandone la frequenza e durata. Si dovranno descrivere eventuali implicazioni riguardanti la salute e alla sicurezza degli utenti.
- L'ambiente nel quale gli utenti utilizzeranno il sistema.

  Si descriveranno le caratteristiche rilevanti dell'ambiente fisico e sociale, includendo l'ambiente di lavoro, le tecnologie utilizzate, gli eventuali standard adottati, il contesto normativo (per esempio, le leggi e i regolamenti applicabili), la struttura organizzativa e le procedure di lavoro, le abitudini consolidate, ecc.

Lo standard ricorda esplicitamente che tutte queste descrizioni non potranno essere congelate in un documento immutabile, ma dovranno essere raccolte in un documento di lavoro, che sarà revisionato, corretto e ampliato più volte, in accordo con la natura iterativa del processo di progettazione e sviluppo. Dovranno essere sempre verificate e confermate dagli utenti del sistema, o da chi li rappresenta nel progetto.

Specificare i requisiti utente e organizzativi

Nella maggior parte dei processi di progettazione, esiste una consistente attività per la specifica dei requisiti del prodotto o del sistema. Nella progettazione human-centred, quest'attività dovrebbe essere ampliata, per descrivere i requisiti in relazione al contesto d'uso più sopra specificato.

Si dovrebbero considerare i seguenti aspetti:

- le prestazioni richieste al nuovo sistema in relazione agli obiettivi operativi ed economici;
- i requisiti normativi e legislativi rilevanti, compresi quelli relativi alla sicurezza e alla salute;
- la comunicazione e la cooperazione fra gli utenti e gli altri attori coinvolti;
- le attività degli utenti (inclusa la ripartizione dei compiti, il benessere e la motivazione degli utenti);
- la ripartizione dei compiti fra esseri umani e sistemi tecnologici;
- le prestazioni dei diversi compiti;
- la progettazione delle procedure di lavoro e dell'organizzazione;

[88] La presente sezione contiene una sintesi di alcune pagine dello standard. Il suo fine è puramente didattico e introduttivo, e non dovrebbe essere utilizzato in sostituzione del documento originale, per esempio per valutare la conformità allo standard delle procedure in atto presso una certa organizzazione.

- la gestione del cambiamento, incluse le attività di addestramento e il personale coinvolto;
- la fattibilità delle diverse operazioni, comprese quelle di manutenzione;
- la progettazione dei posti di lavoro e la interfaccia uomo-computer.

I requisiti e gli obiettivi dovrebbero essere stabiliti, ove necessario, operando opportuni compromessi fra eventuali requisiti fra loro conflittuali. I requisiti dovrebbero essere organizzati per livelli di priorità, e formulati in modo da permettere la loro successiva convalida mediante opportuni test. Dovrebbero essere confermati o aggiornati lungo tutta la durata del progetto.

Produrre soluzioni di progetto

In questa fase si individuano le possibili soluzioni di progetto, basandosi sullo stato dell'arte, sulle conoscenze ed esperienze dei partecipanti e sui risultati dell'analisi del contesto d'uso. Lo standard identifica le seguenti attività.

- Utilizzare le conoscenze disponibili per sviluppare proposte di progetto con un approccio multi-disciplinare.

  Esiste un vasto corpo di teorie e conoscenze scientifiche nell'ergonomia, nella psicologia, nelle scienze cognitive, nelle scienze della progettazione e in altre discipline rilevanti, che possono suggerire possibili soluzioni di progetto. Molte organizzazioni dispongono di linee guida per l'interfaccia utente, conoscenze sul prodotto e sul mercato che possono essere utili nelle fasi iniziali del progetto, soprattutto quando si progettano prodotti di tipo già noto, per esempio versioni migliorative di un sistema già in uso. Inoltre, esistono linee guida e standard per l'ergonomia e i fattori umani, elaborati dagli enti di standardizzazione.

- Rendere le soluzioni di progetto più concrete, utilizzando simulazioni, modelli e prototipi di vario tipo.
  L'uso di prototipi permette ai progettisti di comunicare più efficacemente con gli utenti, e riduce la necessità di costosi rifacimenti, che possono essere invece necessari quando il prodotto viene sottoposto a revisione soltanto più tardi nel ciclo di vita – addirittura dopo il rilascio finale agli utenti reali. Dedicheremo a questo argomento l'intero capitolo 9.
  L'attività di prototipazione può essere compiuta in tutte le fasi del progetto, dalle prime idee basate sulla descrizione del contesto d'uso (per esempio, utilizzando scenari d'uso), fino ai prototipi di pre-produzione, virtualmente completi di ogni dettaglio. Un prototipo può essere semplice quanto uno schizzo a matita, o complesso come una simulazione su computer, quasi indistinguibile dal prodotto reale.

- Presentare le soluzioni di progetto agli utenti, permettendo loro di eseguire i compiti che il sistema è destinato a supportare (eventualmente in modo simulato).

  Gli utenti possono essere coinvolti molto presto nel progetto, mediante l'uso di modelli statici realizzati sulla carta. È possibile presentare agli utenti le bozze delle schermate o una rappresentazione del prodotto, chiedendo loro di provarli in un contesto realistico. In tal modo si possono valutare rapidamente ed economicamente aspetti del progetto (per esempio, quanto sia facile navigare attraverso una gerarchia di menu). Per i prodotti hardware, analoghi benefici possono essere ottenuti con l'uso di modelli tridimensionali statici, costruiti con materiali semplici. Nelle fasi iniziali, anche i prototipi più rudimentali possono risultare preziosi, per esplorare soluzioni alternative. Anche se può essere utile presentare le soluzioni di progetto nel modo più realistico possibile, è consigliabile evitare di investire troppo tempo o denaro nella loro realizzazione, anche perché ciò potrebbe produrre una resistenza alle modifiche da parte dei progettisti.

  In un approccio human-centred, un prototipo non è semplicemente una demo per mostrare un'anteprima del prodotto agli utenti. Esso serve a raccogliere le loro reazioni, per poi utilizzarle nell'orientare le attività di progettazione successive. Quando non fosse consigliabile mostrare i prototipi agli utenti all'inizio del processo di progettazione (per esempio, per ragioni di riservatezza), le valutazioni potranno essere condotte da esperti. Queste possono essere utili e poco costose, e complementare i test con l'utente. In ogni caso, in un processo di progettazione human-centred, almeno i test finali dovrebbero essere condotti con utenti reali.

- Modificare il progetto in conseguenza delle reazioni degli utenti, e ripetere questo processo fino a che gli obiettivi della progettazione non siano raggiunti.

  La natura dei prototipi e il numero delle iterazioni variano in funzione di numerosi fattori, fra cui l'importanza attribuita alla qualità del progetto finale. Nello sviluppo di software, si può iniziare presentando sulla carta degli schizzi delle schermate, e proseguire attraverso diverse iterazioni, fino a produrre software interattivo con funzionalità sufficienti a supportare un sottoinsieme dei compiti dell'utente. Nelle fasi avanzate della progettazione, i prototipi potranno essere valutati in contesti più realistici. Per ottenere i massimi benefici, è conveniente effettuare numerose iterazioni con l'utente. In seguito, per decidere se gli obiettivi complessivi sono stati raggiunti, dovrebbero essere condotte delle valutazioni più formali in un contesto realistico, per esempio, senza suggerimenti o interruzioni da parte del valutatore. I commenti dell'utente, o le difficoltà osservate durante l'utilizzo del prototipo, suggeriranno modifiche al progetto, per migliorarne l'usabilità. In qualche caso, questi feedback potranno anche essere di aiuto per raffinare gli obiettivi complessivi del sistema.

- Gestire l'iterazione delle soluzioni di progetto.

  Per tenere sotto controllo i progressi della progettazione iterativa, si dovrebbero registrare i risultati delle attività precedenti. Questa documentazione potrà essere esclusivamente descrittiva, o includere i prodotti stessi della progettazione, come i prototipi hardware e software. La documentazione descriverà lo scopo dei vari prototipi, le modalità operative del loro utilizzo e i problemi individuati, con le conseguenti modifiche del progetto.

Valutare il progetto nei confronti dei requisiti.

La valutazione è un passo essenziale in una progettazione human-centred, e dovrebbe essere compiuta in tutte le fasi del ciclo di vita del sistema, e non soltanto in fase di progettazione. All'inizio del progetto, l'obiettivo principale sarà la raccolta d'indicazioni per orientare le attività di progettazione successive. Nelle fasi più avanzate, con la disponibilità di prototipi più completi, sarà invece possibile quantificare il livello di raggiungimento degli obiettivi dell'utente e dell'organizzazione. Dopo il rilascio del sistema, sarà possibile monitorarne l'adeguatezza ai nuovi contesti di utilizzo.

Lo standard identifica le seguenti attività:

- Produrre il piano di valutazione

  Il processo di valutazione dovrebbe essere pianificato, precisando, tra l'altro, quali parti del sistema devono essere valutati e come; quali prototipi dovranno essere realizzati e come deve essere eseguita la valutazione e con quali risorse; quali dovranno essere le interazioni con gli utenti e come dovrà essere condotta l'analisi dei risultati. Le tecniche di valutazione variano secondo i casi. La scelta è determinata dalla natura del sistema, dai vincoli economici e di tempo, e dalla fase del ciclo di sviluppo in cui si svolge la valutazione.

- Fornire feedback per la progettazione

  Per influenzare la progettazione, la valutazione dovrebbe essere condotta in ogni fase del ciclo di vita del sistema. La valutazione condotta soltanto da esperti, senza il coinvolgimento degli utenti, può essere veloce ed economica, e permettere di identificare i problemi maggiori, ma non basta a garantire il successo di un sistema interattivo. La valutazione basata sul coinvolgimento degli utenti permette di ottenere utili indicazioni in ogni fase della progettazione. Nelle fasi iniziali, gli utenti possono essere coinvolti nella valutazione di scenari d'uso, semplici mock-up cartacei o prototipi parziali. Quando le soluzioni di progetto sono più sviluppate, le valutazioni che coinvolgono l'utente si basano su versioni del sistema progressivamente più complete e concrete. Può anche essere utile una valutazione cooperativa, in cui il valutatore discute con l'utente i problemi rilevati.

- Verificare se gli obiettivi sono stati raggiunti

  La valutazione può essere usata per dimostrare che un particolare progetto soddisfa i requisiti human-centred, oppure per verificare la conformità a standard internazionali, nazionali, locali, aziendali o legali. In ogni caso, per ottenere risultati validi, la valutazione dovrebbe utilizzare metodi appropriate, con un campione rappresentativo di utenti che eseguono compiti realistici.

- Validazione sul campo

  Lo scopo della validazione sul campo è provare il funzionamento del sistema finale durante l'uso effettivo, per assicurare che esso soddisfi i requisiti degli utenti, dei compiti e dell'ambiente. A questo scopo si possono analizzare i dati raccolti dall'help desk, rapporti dal campo, feedback da utenti reali, dati prestazionali, rapporti sull'impatto sulla salute, richieste di miglioramenti e di modifiche da parte degli utenti.

- Monitoraggio di lungo termine

  Dovrebbe esistere un processo pianificato per il monitoraggio di lungo termine dell'uso del prodotto o del sistema, che consiste nel raccogliere input dagli utenti, con modalità differenti, lungo un certo periodo di tempo. Infatti, alcuni effetti dell'utilizzo di un sistema interattivo non sono riconoscibili fino a che il sistema non sia stato utilizzato per un certo periodo di tempo, e ci possono essere effetti che derivano da fattori esterni, per esempio da cambiamenti imprevisti nelle modalità lavorative o nel contesto di mercato.

- Documentazione dei risultati

  Allo scopo di gestire il processo di progettazione iterativo, i risultati delle valutazioni dovrebbero essere registrati in modo sistematico. In particolare, dovrebbe esistere un'adeguata evidenza che un numero adeguato di utenti abbia partecipato al test e che questi utenti siano rappresentativi delle categorie identificate nei requisiti, che i test effettuati siano adeguati a fornire indicazioni attendibili per i vari casi e contesti d'uso e, infine, che siano stati usati metodi appropriati per il test e la raccolta dei dati.

Il ruolo dell'utente nel processo di progettazione

Come si è già osservato, il modello dell'ISO 13407 è del tutto generale, e può essere realizzato concretamente in molti modi diversi. In effetti, sono stati definiti e applicati vari approcci che, pur rientrando tutti nell'ambito di un'impostazione human-centred, differiscono fra loro nei dettagli e, soprattutto, nel ruolo che l'utente assume durante il processo della progettazione. Infatti, a seconda delle scelte organizzative, l'utente può essere coinvolto secondo modalità diverse nelle varie attività del processo, indicate con A, B, C e D in Figura 112.

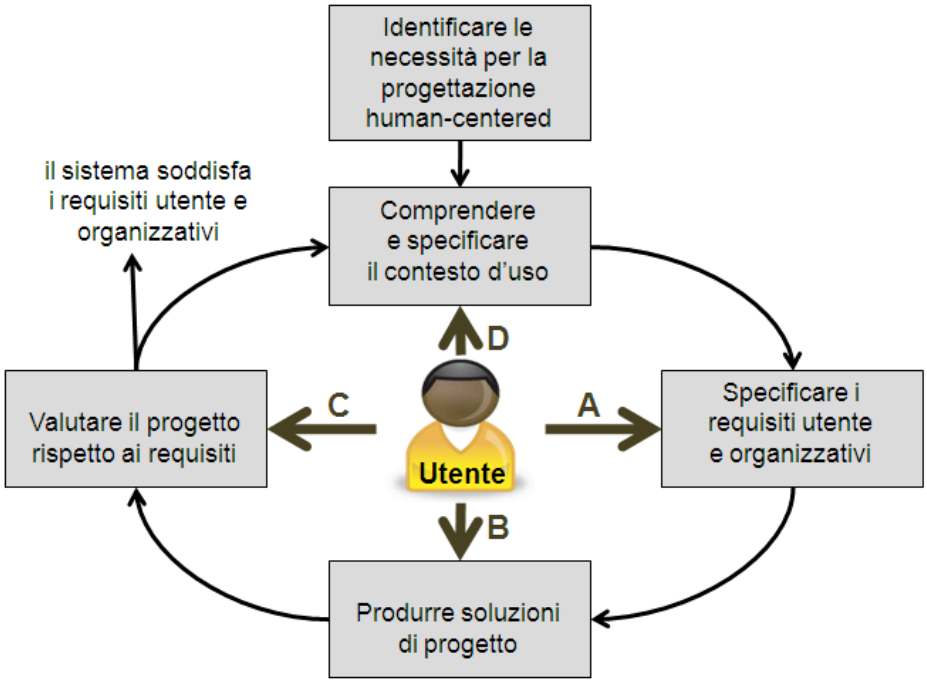


Figura 112. L'utente può essere coinvolto in una o più attività del processo di progettazione human-centred

L'approccio più conosciuto è noto come *progettazione centrata sull'utente*, o *user-centred design* (UCD), proposto da Norman e Draper un quarto di secolo fa.[89] Secondo questa impostazione, l'utente viene sostanzialmente coinvolto solo nelle fasi A e C della Figura 112. Pertanto, ha un ruolo fondamentale nell'acquisizione dei requisiti del sistema (A) e nell'effettuazione delle prove d'uso dei diversi prototipi prodotti nelle varie iterazioni del progetto (C). Non è, invece, coinvolto nelle attività di progettazione e realizzazione dei vari prototipi (B), condotte dai progettisti e dagli sviluppatori del sistema.

In altri approcci, il coinvolgimento dell'utente avviene con modalità diverse. Per esempio, nella progettazione partecipativa (*participatory design,* inizialmente chiamato *cooperative design*), l'utente viene coinvolto anche nelle attività di progettazione (B), partecipando attivamente, con modalità e strumenti opportuni, alla realizzazione dei prototipi. Al contrario, nello *usage-centred design*, proposto da Constantine e Lockwood[90], il centro dell'attenzione è sull'uso – cioè sulle attività effettuate dall'utente e sui compiti che egli compie – e non sull'utente, che viene coinvolto nel processo di progettazione in modo molto limitato.

Questo libro adotta un'impostazione pragmatica, basata sostanzialmente sulla filosofia dello user-centred design, ma interpretato in modo flessibile. Infatti, se è vero che l'utente costituisce una fonte importantissima d'informazioni per la stesura dei requisiti e un attore primario nella convalida dei prototipi via via realizzati, è tuttavia indispensabile che i suoi desideri e suggerimenti non siano interpretati come dictat assoluti. L'utente non è un progettista - non ne ha l'esperienza né la forma mentis. In molti casi, può non essere in grado di valutare le implicazioni di suggerimenti che, visti fuori dal contesto del progetto, potrebbero sembrare del tutto corretti. Può capitare infatti che alcuni suggerimenti, se attuati, producano degli effetti collaterali indesiderabili. Oppure, semplicemente, che esistano altri modi, più convenienti, di ottenere gli stessi risultati. Il progettista esperto, invece, è in grado di valutare le implicazioni delle indicazioni degli utenti, e di comprenderne le eventuali controindicazioni.

Chiunque abbia avuto un'esperienza, sia pur modesta, di progettazione in contesti reali, sa che cosa significa dover "lottare" con i suoi utenti, per correggere indicazioni che sa fondamentalmente sbagliate. Un aspetto importante da tenere in considerazione è la naturale resistenza al cambiamento negli utenti. Le persone non amano cambiare abitudini e modalità operative, e la prospettiva di dover modificare il proprio modo di lavorare genera spesso delle resistenze. Ciò fa sì che raramente prodotti realmente innovativi, che modificano radicalmente le abitudini operative consolidate, siano proposti dai loro futuri utenti. Questi prodotti nascono più dalle visioni di designer innovativi che da proposte originate dai futuri utenti.

Lo stesso Donald Norman, autore della filosofia user-centred, ha sentito la necessità, più recentemente, di correggere il tiro spostando l'enfasi dall'utente alle sue attività, in un approccio da lui chiamato *activity-centred design*:

> *Una filosofia di base dello human-centred design è ascoltare gli utenti, e prendere le loro lamentele e le loro critiche sul serio. Sì, ascoltare i clienti è sempre saggio, ma accettare le loro richieste può condurre a progetti eccessivamente complessi. Parecchie grandi società di software, orgogliose della loro filosofia human-centred, soffrono di questo problema. Il loro software diventa più complicato e meno comprensibile a ogni nuova revisione. La filosofia activity-centred tende a proteggerci da questo errore perché si focalizza sulle attività, non sull'essere umano. Ne risulta un modello di progetto coerente e bene articolato. Se un suggerimento dell'utente non rientra in questo modello, dovrebbe essere scartato. Ahimè, troppe società, orgogliose di ascoltare i propri utenti, lo accetterebbero. Qui, ciò che serve è un progettista forte e autorevole in grado di esaminare i suggerimenti e valutarli nei termini dei requisiti dell'attività. Quando è necessario, è essenziale poter ignorare le richieste. Questo per conseguire coerenza e comprensibilità. Paradossalmente, il modo migliore di soddisfare gli utenti è, qualche volta, di ignorarli. […]*

---

[89] D.A.Norman, S.W.Draper, *User Centred System Design*, in *New Perspectives on Human-Computer Interaction*, L.Erlbaum Associates Inc., 1986.

90 Constantine, L. L., and Lockwood, L. A. D., Software *for Use: A Practical Guide to the Essential Models and Methods of Usage-Centred Design*, Addison-Wesley, 1999.

A volte ciò che serve è un dittatore che dica "ignorate ciò che dicono gli utenti: io so ciò che è meglio per loro." Il caso della Apple è esemplificativo. I prodotti della Apple sono stati a lungo ammirati per la loro facilità d'uso. Tuttavia, la Apple ha sostituito il suo famoso e rispettato team di progetto con un unico, autorevole (dittatoriale) leader. L'usabilità ne ha sofferto? Al contrario: i suoi nuovi prodotti sono considerati esempi di grande design. "Ascolta i tuoi utenti" produce design incoerenti. "Ignora i tuoi utenti" può produrre storie di orrore, a meno che la persona alla guida abbia una chiara visione del prodotto, ciò che ho chiamato "modello concettuale". La persona alla guida deve seguire quella visione e non temere di ignorare i fatti. Sì, ascoltate i clienti, ma non fate sempre quello che dicono. [91]

E conclude:

Lo human-centred design garantisce buoni prodotti. Può condurre a netti miglioramenti di prodotti cattivi. Inoltre, lo human-centred design evita i fallimenti. Assicura che i prodotti funzionano, che la gente li può usare. Ma è un buon design lo scopo? Molti di noi desiderano un design grande. Il design grande, io affermo, deriva dalla rottura delle regole, dall'ignorare le pratiche generalmente accettate, dal procedere con un chiaro concetto del risultato finale, non importa quale. Questo design egocentrico e guidato dalla visione conduce a grandi successi o a grandi fallimenti. Se lo volete grande piuttosto che buono, questo è ciò che dovete fare.

L'esempio dei siti web

Gli schemi di Figura 109 o Figura 111 sono ancora troppo astratti per essere realmente utili in un progetto reale. Infatti nulla ci dicono su come procedere, in pratica, nel caso specifico. Quanti prototipi (e quindi quante iterazioni) dobbiamo realizzare? Quali obiettivi ci dobbiamo porre nella realizzazione e nella valutazione di ciascun prototipo? Quali tecniche di prototipazione dobbiamo utilizzare? Come possiamo realizzarli a costi ridotti? Come possiamo tenere sotto controllo i costi complessivi del progetto? A queste domande non è possibile rispondere in generale, e cioè in modo indipendente dal tipo e dalle caratteristiche del sistema in oggetto. È, invece, possibile mettere a punto specifiche strategie per specifiche classi di sistemi.

Per esempio, è possibile dare indicazioni molto precise su come impostare il processo di progettazione e sviluppo di un sito Web, specificando quanti prototipi è conveniente realizzare, in quali fasi, con quali tecnologie e con quali obiettivi specifici, da valutare attraverso specificate attività di test. A questo argomento è dedicato un altro libro di chi scrive.[92] Rimandando il lettore interessato a questo libro per approfondimenti, ci limitiamo qui a indicare che, in questo caso, il processo iterativo user-centred si può convenientemente sviluppare in sette macrofasi, con cinque prototipi principali, ciascuno dei quali ha tecniche realizzative e finalità specifiche (Figura 113).

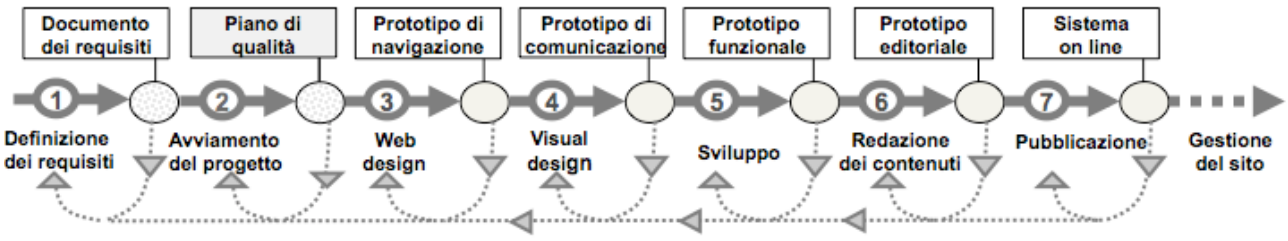


Figura 113. Processo iterativo per la progettazione e sviluppo di un sito web

[91] Nostra traduzione da D.A.Norman, *Human-Centred Design Considered Harmful*, in *Interactions*, 12,4 (luglio-agosto 2005). Disponibile anche in rete, sul sito di Norman http://www.jnd.org.

[92] Questa impostazione è analiticamente descritta nel libro: R.Polillo, *Plasmare il Web – Road map per siti di qualità* (Apogeo, 2006), nel quale viene dettagliata una completa "road-map" in sette fasi per la progettazione e sviluppo di siti di medie dimensioni.

In sintesi, dopo la fase di realizzazione del documento dei requisiti e di avviamento del progetto (il cui documento finale è denominato Piano di qualità), vengono realizzati i seguenti prototipi:

- Primo prototipo (*prototipo di navigazione*): ha lo scopo di consolidare l'architettura informativa e di navigazione del sito. Permette all'utente di vedere l'ossatura del sito (ancora privo di grafica e di contenuti informativi, ma dotato delle strutture di navigazione principali, per esempio i menu), e di navigare al suo interno.
- Secondo prototipo (*prototipo di comunicazione*): ha lo scopo di consolidare l'impostazione grafica del sito e tutti gli aspetti riguardanti la comunicazione. Mancano ancora del tutto i contenuti informativi e le funzioni interattive, ma la cornice grafica è quella finale.
- Terzo prototipo (*prototipo funzionale*): ha lo scopo di consolidare le funzioni interattive del sito. I contenuti informativi sono ancora assenti, ma il "contenitore" è sostanzialmente pronto, e l'utente può provarne le funzioni interattive, sia pure in un contesto ancora lontano da quello reale.
- Quarto prototipo (*prototipo editoriale*): ha lo scopo di consolidare i contenuti informativi e la (eventuale) base dati del sito. Il sito è praticamente pronto: i test con l'utente possono svolgersi in un ambiente completo e realistico, anche se su sistemi di sviluppo, e non di produzione.
- Quinto prototipo (*prototipo finale*): ha lo scopo di permettere di valutare le prestazioni di funzionamento del sito sui sistemi finali di produzione.

Naturalmente, le prove effettuate su ciascun prototipo potranno suggerire delle iterazioni, come indicato dalle frecce all'indietro in Figura 113. Normalmente, se il processo è condotto bene, questi ricicli saranno sostanzialmente confinati all'interno della macro-fase corrente. Così, per esempio, le prove d'uso del prototipo di comunicazione produrranno aggiustamenti nella grafica, con successive iterazioni all'interno della fase 4 in figura, ma solo di rado richiederanno modifiche alle fasi precedenti (prototipo di navigazione e requisiti). In questo modo, per così dire, il modello iterativo di Figura 109 o Figura 111 viene, per così dire, sostanzialmente linearizzato, assomigliando, se tutto va bene, al modello "a cascata", con notevoli benefici in termini di controllo dei costi e qualità del risultato finale.

Le professioni dell'usabilità

Nel contesto dei processi dell'ingegneria dell'usabilità, alcune professionalità specifiche assumono un ruolo rilevante. Esse possono venire raccolte sotto la generica etichetta di *usability professional*, come propone l'associazione internazionale dei professionisti dell'usabilità (UPA, *Usability Professional Association*), che, nel suo sito web (www.upassoc.org), definisce usability professional…

> *...in generale, chiunque lavori per la usabilità di un prodotto, o ne sia un promotore. Alcuni si specializzano nel condurre test o ricerche sugli utenti, mentre altri praticano l'usabilità come parte delle loro attività di progettazione di prodotti, servizi, applicazioni software o siti web.*

La formazione e il background professionale dei professionisti dell'usabilità è altrettanto ampia. Aggiunge infatti la UPA, che "molti hanno qualifiche in campi strettamente correlati, come la interazione uomo-macchina (HCI), l'information design o la psicologia. Altri hanno usato il loro background nella computer science, nel project management, nel giornalismo, nelle arti, nelle scienze bibliotecarie, o nel business come parte del loro viaggio verso questa professione."

Lo spettro di attività nelle quali trova impiego un professionista dell'usabilità è altrettanto ampio. La stessa UPA, nel suo sito (che fa specifico riferimento al modello di progettazione dell'ISO 13407 sopra descritto), menziona le seguenti attività, ripartite fra le fasi del progetto:

In fase di analisi:

- incontrare gli stakeholder[93] per impostare la visione;
- inserire nel piano di progetto le attività relative all'usabilità;
- organizzare un team multidisciplinare per assicurare un'esperienza completa;
- sviluppare gli obbiettivi di usabilità;

[93] Gli stakeholder sono tutti coloro che hanno interesse nel progetto, se ne parlerà a pag. 145.

- condurre studi sul campo;
- esaminare prodotti concorrenti;
- creare i profili degli utenti;
- sviluppare analisi dei compiti;
- documentare scenari d'uso;
- documentare i requisiti relativi alle prestazioni degli utenti.

In fase di progettazione:
- effettuare brainstorming sui design concept e sulle metafore;
- sviluppare le sequenze di schermate e i modelli di navigazione;
- revisionare i design concept;
- avviare il progetto con carta e matita;
- creare prototipi a bassa fedeltà;
- condurre test di usabilità su prototipi a bassa fedeltà;
- creare il design di dettaglio per i prototipi ad alta fedeltà;
- effettuare ancora i test di usabilità;
- documentare standard e linee guida;
- creare specifiche di progetto.

In fase di realizzazione:

- effettuare valutazioni durante il processo;
- lavorare a stretto contatto con il team di sviluppo per la rifinitura del design;
- condurre test di usabilità quanto prima possibile.

In fase di avviamento:
- utilizzare questionari per ottenere feedback dagli utenti;
- condurre studi sul campo per ottenere informazioni sull'utilizzo effettivo;
- verificare il raggiungimento degli obiettivi per mezzo di test di usabilità.

Considerando la varietà dei temi trattati, i professionisti dell'usabilità costituiscono una popolazione piuttosto variegata, in funzione delle specifiche competenze. Le specializzazioni più diffuse, ancora secondo la UPA, sono:

- User Experience Practitioner
- Interface Designer
- Usability Practitioner
- User-Centred Design Practitioner
- Information Architect
- Usability Manager

Altre professioni strettamente collegate, sempre secondo l'UPA, includono quelle di:

- Web Designer
- Technical Writer
- Business or Requirements Analyst
- Technical or Software Analyst
- Market Researcher
- Instructional Designer
- Industrial Designer.

Costi e benefici

Nei paragrafi precedenti abbiamo illustrato le attività necessarie per la progettazione e realizzazione di sistemi usabili. Da quanto detto è evidente che l'usabilità non nasce "per caso". Essa va accuratamente pianificata, progettata e monitorata durante tutto l'arco del progetto, utilizzando risorse e attività specifiche, secondo i metodi dell'ingegneria dell'usabilità. Tutto questo, ovviamente, ha dei costi. È naturale, quindi, chiedersi quale sia il rapporto fra i costi e i benefici ottenibili.

I costi sono essenzialmente di due tipi. Innanzitutto, ci sono i costi dell'investimento necessario per trasformare un'organizzazione di progetto tradizionale in un'organizzazione che utilizza i metodi dell'ingegneria dell'usabilità. Questo richiede attività di addestramento (per i progettisti e i responsabili di progetto) e reclutamento di nuove risorse (per esempio, consulenti esperti di usabilità) da inserire in quei team di progetto multi-disciplinari di cui abbiamo parlato nel capitolo 3. Si tratta quindi di gestire un cambiamento organizzativo e culturale, che nel caso di grandi organizzazioni di progetto potrebbe essere non banale, e richiedere una prima fase di sperimentazione attraverso progetti pilota. Come si possono quantificare i ritorni di questi investimenti?

In secondo luogo, supponendo di considerare un'organizzazione che sappia già fare dell'ingegneria dell'usabilità, ci sono i costi delle specifiche attività finalizzate all'usabilità, e che non verrebbero eseguite in un processo d'ingegneria tradizionale. Come quantificarne il rapporto costi/benefici? Com'è del tutto evidente, non si tratta di un'operazione banale. Come osserva Nielsen, il modo più corretto per quantificare il rapporto costi/benefici sarebbero quello di realizzare due versioni "equivalenti" dello stesso prodotto, in un caso senza porre in essere alcuna attività specifica finalizzata all'usabilità, e nell'altro adottando un processo di progettazione centrato sull'utente, tenendo traccia dei costi complessivi di progettazione in ciascuna situazione. In seguito, si dovrebbero far utilizzare in contesti simili entrambi i prodotti per un periodo di tempo sufficientemente lungo e da parte di un numero significativo di utenti, misurandone l'usabilità, per esempio definendo opportune metriche di efficacia, efficienza e soddisfazione, come abbiamo visto nel capitolo 1, che si possano tradurre in termini economici. Per esempio, per quanto riguarda l'efficienza, potremmo quantificare i tempi medi di esecuzione dei vari compiti in entrambi i casi, valorizzando il tempo sulla base del costo medio del personale utilizzato. Dovremmo, ovviamente, considerare nei due casi sia i tempi di apprendimento dei prodotti, sia i tempi richiesti dal loro uso a regime. Per quanto riguarda l'efficacia, potremmo poi considerare la frequenza degli errori d'uso in un caso e nell'altro, e tradurre ancora una volta questi dati in termini economici. Infine, per quanto riguarda la soddisfazione dell'utente, dovremmo compiere delle indagini attraverso questionari, ed eventualmente formulare delle ipotesi, da tradurre ancora una volta in termini economici, sul mercato potenziale di ciascun prodotto sulla base del gradimento medio.

Tutto questo non è evidentemente realizzabile in pratica e, anche se lo fosse, le variabili coinvolte nell'esperimento sono così numerose da renderne comunque le conclusioni piuttosto discutibili. È tuttavia possibile, e relativamente poco costoso, condurre per così dire degli "esperimenti concettuali", quantificando, sotto opportune ipotesi, i potenziali guadagni di una migliorata usabilità. I risultati saranno ipotetici, certamente opinabili ma, nel caso di prodotti destinati a un mercato di massa, spesso convincenti, almeno in termini di benefici "sociali".

Per esempio, proviamo a ipotizzare, per un certo prodotto (per esempio un sistema di word processing), una determinata Riduzione del Tempo di Apprendimento medio (RTA), dovuta a una migliorata usabilità, e una determinata Riduzione del Tempo medio di Esecuzione di un certo compito importante e frequente c (RTM_c), dovuta sia a un'interazione più agevole, sia a una riduzione statistica del numero degli errori effettuati dall'utente. Se ipotizziamo, anche senza condurre analisi raffinate, che RTA=4h (mezza giornata lavorativa) e che RTM_c = 10 secondi, e se ipotizziamo che il compito c venga eseguito, in media, 1 volta al giorno, su una popolazione di 100.000 utenti il risparmio nel primo anno di utilizzo del prodotto sarebbe:

(4h x 100.000) + (10 sec x 365gg x 100.000) =circa 400.000h + 100.00h = 500.000h.

Questo corrisponde a un risparmio complessivo, sulla popolazione considerata, di circa 62.500 giornate lavorative ovvero, considerando 200 giornate lavorative medie annue pro-capite, un risparmio di tempo di circa il 3 per mille. Poiché le ipotesi fatte sono del tutto ammissibili anche senza analisi particolarmente raffinate (e con un atteggiamento

di cautela), il risparmio "sociale" è certamente significativo. Ovviamente, questo dato non è necessariamente d'interesse per il produttore o il venditore, che considerano in primo luogo costi e benefici in relazione al suo bilancio.

Certamente, tuttavia, è possibile effettuare altri esperimenti concettuali, per quantificare benefici che più direttamente impattano sul conto economico del produttore. Le voci che potrebbero essere considerate sono numerose, a partire dai benefici ottenibili già in fase di progettazione fino ai benefici risultanti da una migliore accoglienza del prodotto da parte del mercato, per esempio:

- Riduzione dei costi complessivi di progettazione e sviluppo, derivanti da minori rifacimenti o modifiche del prodotto nelle fasi finali del progetto, dovuti all'utilizzo di processi di progettazione iterativi, che permettono di anticipare i problemi di usabilità – e le loro soluzioni. È noto, infatti, che i costi delle modifiche di un prodotto sono tanto maggiori quanto più tali modifiche avvengono nelle fasi finali del progetto, o addirittura dopo il suo rilascio agli utilizzatori.
- Riduzione dei costi di manutenzione, per gli stessi motivi.
- Riduzione dei costi di supporto all'utente, in termini di formazione all'uso e documentazione tecnica più semplici e quindi meno costose e, soprattutto, di supporto post-vendita (es. help desk e assistenza tecnica).
- Maggiore soddisfazione dell'utente, con conseguente miglioramento dell'immagine del prodotto e della credibilità del fornitore sul mercato e, di conseguenza, un incremento dei volumi di vendita.

In conclusione, non è difficile condurre esperimenti concettuali anche piuttosto raffinati nell'ambito di prodotti specifici, per sostenere la tesi che una buona usabilità, effettivamente, ripaga. Esistono peraltro, in letteratura, numerose ricerche che, nell'ambito di specifici prodotti, forniscono dati convincenti a supporto di queste affermazioni. Tuttavia è convinzione di chi scrive che tali quantificazioni non colgano il punto essenziale. I benefici dell'usabilità vanno ben oltre i risparmi strettamente quantificabili sul conto economico delle aziende. In un contesto che si fa sempre più complesso, come accennato nell'introduzione, l'usabilità è un attributo importante, che migliora la qualità della vita degli utenti e riduce l'entità e le conseguenze del *digital divide*. Perseguire l'usabilità significa cercare di costruire un mondo a misura d'uomo e non delle macchine, in cui si viva e si lavori meglio, che minori sprechi di risorse e con più soddisfazione.

Ripasso ed esercizi

1. Spiega che cosa s'intende per ingegneria dell'usabilità.
2. Spiega quali sono, a tuo parere, i rapporti fra l'ingegneria del software e l'ingegneria dell'usabilità.
3. Quali sono i vantaggi e gli svantaggi del modello tradizionale di progettazione e sviluppo "a cascata"?
4. Quali sono i vantaggi e gli svantaggi del modello di progettazione e sviluppo per prototipi successivi?
5. Spiega che cosa s'intende con "ciclo compito-artefatto".
6. Quali sono le principali caratteristiche del modello di progettazione human-centred proposto dall'ISO 13407?
7. Che cosa s'intende per user-centred design?
8. Analizza l'interfaccia utente di Facebook, e identifica qualche semplice miglioramento auspicabile, quantificandone, in modo approssimato, il rapporto costi/benefici di tale intervento.

Approfondimenti e ricerche

1. Per approfondire i modelli dei processi di progettazione e sviluppo definiti nell'ambito dell'ingegneria del software puoi consultare uno dei molti libri introduttivi all'argomento, per esempio il classico testo di I.Sommerville, *Ingegneria del software* (Ottava edizione), Pearson Addison-Wesley, 2007.
2. Lo standard ISO 13407 si trova in rete in http://www.iso.org. Purtroppo è accessibile soltanto a pagamento.
3. Leggi la nota di Norman su *Human-Centred Design Considered Harmful*, citata nel presente capitolo (disponibile in rete, in htto://www.jnd.org).
4. La letteratura disponibile sull'analisi dei costi e benefici dell'usabilità è vasta, a partire dall'importante libro a cura di R.Bias e D.J.Mayhew, *Cost Justifying Usability* (Morgan Kaufmann, 1994, seconda edizione nel 2005). Anche in rete esiste abbondante materiale. Puoi approfondire l'argomento, per esempio, partendo dal sito della

UPA http://www.upassoc.org, oppure da http://www.usabilityfirst.com dove ci sono sezioni dedicate all'argomento, oppure cercando in rete "usability ROI", dove ROI sta per "Return Of Investment". Nel sito di Jakob Nielsen http://www.useit.com ci sono interessanti note sull'argomento, in relazione all'usabilità dei siti web. Tieni comunque presente che molto materiale presente in rete ha scopo promozionale, e che spesso le metodologie utilizzate nell'analisi costi/benefici sono approssimate. Può essere utile leggere le considerazioni di buon senso nell'articolo di Daniel Rosenberg, *The miths of usability ROI*, in *ACM Interactions*, vol.5, 11 (settembre-ottobre 2004).

5. Puoi divertirti a condurre esperimenti concettuali sul ritorno degli investimenti sulla usabilità di siti web di e-commerce utilizzando lo "usability ROI calculator" disponibile online in http://www.usereffect.com/topic/new-tool-usability-roi-calculator.

7. I requisiti

Sintesi del capitolo

In questo capitolo vengono descritti gli obiettivi del documento di specifica dei requisiti di prodotto, la cui produzione costituisce la fase iniziale di qualsiasi processo di progettazione. L'attività di definizione dei requisiti è scomposta in tre fasi principali. Nella prima, di esplorazione, i requisiti vengono raccolti con modalità diverse (interviste individuali, questionari, focus group, osservazioni sul campo, suggerimenti spontanei degli utenti). Nella seconda, di organizzazione, le osservazioni così raccolte vengono organizzate in una forma coerente, risolvendo eventuali conflitti, nel documento di specifica dei requisiti. Viene proposta una scaletta generale del documento dei requisiti, nel quale rivestono un ruolo particolarmente importante la descrizione degli scenari d'uso principali del prodotto, e la descrizione di tutti i casi d'uso. Nella terza fase, il documento così prodotto è sottoposto a revisione da parte di tutti gli stakeholder del sistema, e approvato dal committente.

Che cosa sono i requisiti di prodotto

Tutti i processi di progettazione bene organizzati dovrebbero iniziare con la stesura di un documento di requisiti.

Un *requisito* (dal latino *requisitus*, richiesto) è una proprietà richiesta, oppure auspicabile, del prodotto. Il documento dei requisiti ha allora lo scopo di accogliere in forma organica una descrizione di tutte le proprietà desiderate. Dalla sua formulazione dovrebbe essere chiaro se un requisito esprime una proprietà obbligatoria, oppure soltanto suggerita o auspicabile. Per esempio, per un sito web di commercio elettronico potremmo identificare, fra gli altri, i seguenti quattro requisiti:

- Il sito *deve* permettere all'utente di inserire nel carrello d'acquisto i prodotti di cui sta valutando l'acquisto. Il carrello *deve* poter contenere almeno 15 prodotti contemporaneamente.
- Ogni scheda prodotto contenuta nel catalogo *deve* contenere una fotografia a colori del prodotto, il suo nome, il nome del produttore, il prezzo e una descrizione sintetica ma completa, 5 righe di testo al massimo.
- L'intero processo di acquisto di un prodotto *dovrebbe* richiedere al massimo 5 minuti.

Nel terzo esempio, l'uso del verbo *dovrebbe* segnala chiaramente che il requisito è auspicabile, ma non obbligatorio. Come si vede dagli esempi, i requisiti possono essere di vario tipo. Alcuni, detti *requisiti funzionali* (*functional requirements*), descrivono le funzioni che il sistema deve realizzare (come nel primo esempio). Altri, detti *requisiti non funzionali*, descrivono proprietà che il prodotto dovrà possedere (come negli altri esempi). Lo scopo della definizione dei requisiti è individuarli e descriverli nel modo più specifico e meno ambiguo possibile. Lo standard ISO 13407 suggerisce che, per identificare i requisiti rilevanti, in un processo di progettazione human-centred, si considerino i seguenti aspetti:

- le prestazioni richieste al nuovo sistema in relazione agli obiettivi operativi ed economico/finanziari;
- i requisiti normativi o legislativi rilevanti, compresi quelli relativi alla sicurezza e alla salute;
- la comunicazione e la cooperazione fra gli utenti e gli altri attori rilevanti;
- le attività degli utenti, inclusa la ripartizione dei compiti, il loro benessere e le loro motivazioni;
- le prestazioni dei diversi compiti;
- la progettazione dei flussi di lavoro e dell'organizzazione;
- la gestione del cambiamento indotto dal nuovo sistema, incluse le attività di addestramento e il personale coinvolto;
- la fattibilità delle diverse operazioni, comprese quelle di manutenzione;
- La progettazione dei posti di lavoro e la interfaccia uomo-computer.

I requisiti vengono prodotti da persone che lavorano in stretto contatto con il committente per individuarne i bisogni in relazione al sistema da realizzare (o da migliorare, se si tratta di una riprogettazione). Possono essere stesi direttamente dai progettisti, o da persone che non necessariamente saranno coinvolte nel progetto successivo.

È importante non confondere l'attività di stesura dei requisiti con l'attività di progettazione. Quando specifichiamo i requisiti di un prodotto, non stiamo progettando, ma *stiamo ponendo dei vincoli all'attività di progettazione*, che seguirà. In sostanza, lo scopo del documento è di indicare *che cosa* deve essere realizzato e *perché*, non *come* deve essere realizzato.

Il processo di definizione dei requisiti

La fase di definizione dei requisiti può essere suddivisa in tre attività fondamentali, che possiamo chiamare *esplorazione*, *organizzazione* e *revisione* (Figura 114).

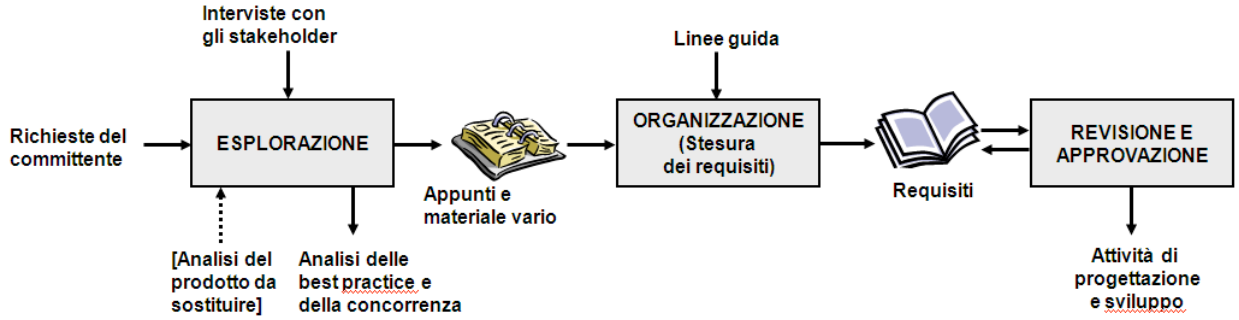


Figura 114. Il processo di definizione dei requisiti

Nell'*esplorazione*, le persone incaricate di produrre il documento dei requisiti raccolgono il maggior numero possibile d'informazioni sugli obiettivi e sulle necessità riguardo al sistema da costruire. Abbiamo usato il termine "esplorazione" per segnalare che, nella pratica, spesso questi obiettivi e necessità sono noti allo stesso committente in forma piuttosto vaga. I consulenti avranno quindi il compito importante e delicato di "esplorare" i diversi aspetti del problema, per mettere a fuoco o scoprire bisogni e priorità (in inglese si usano i termini *elicitation* o *discovery*).

Come indicato nella Figura 114, le informazioni vengono raccolte da fonti diverse. In primo luogo, dal committente, cioè colui che ha avviato il progetto e che ne costituisce il riferimento principale. In secondo luogo, dalle interviste con gli *stakeholder* del prodotto, cioè tutti coloro che, in un modo o nell'altro, hanno qualche interesse nel prodotto, o la cui attività sarà influenzata, direttamente o indirettamente, da esso.[94] Infine, dall'analisi della concorrenza, cioè di quei prodotti con i quali il prodotto in costruzione dovrà confrontarsi e competere. Se si tratta di un progetto di miglioramento di un prodotto esistente, informazioni importanti saranno ricavate anche dall'analisi dei suoi pregi e difetti.

Durante quest'attività, vengono raccolti appunti e materiale informativo vario, che dovranno successivamente essere riesaminati, selezionati e organizzati. Questo è lo scopo della successiva attività di *organizzazione* (o stesura dei requisiti), indicata sempre in Figura 114. L'obiettivo principale di questa fase è costruire un *documento di specifica dei requisiti*, condiviso e approvato dal committente. Questo sarà il riferimento principale per tutte le attività successive del progetto. Lo scopo di questo documento è di descrivere, nella forma più completa possibile, le richieste del committente e i vincoli che dovranno essere rispettati nelle fasi successive del progetto. Si analizza il materiale raccolto nella fase di esplorazione, lo si riordina, si risolvono eventuali contraddizioni (le persone intervistate potrebbero avere idee molto diverse su ciò che occorre fare), e si produce una prima bozza del documento dei requisiti. Il redattore dovrà ricorrere a

[94] La parola inglese *stakeholder* denota gli azionisti o, più in generale, tutti coloro che hanno qualche interesse in un'impresa. Il termine è di uso corrente nell'interaction design.

tutta la sua esperienza e competenza, per produrre un documento che tenga conto, per quanto possibile, dei punti di vista di tutti gli intervistati, ma che li integri in una proposta organica e coerente e che, soprattutto, sia in accordo con le priorità indicate dal committente. È lui infatti che, in quanto referente principale del progetto, avrà l'ultima parola, in caso di dubbi o conflitti.

Nella fase di *revisione e approvazione*, la bozza del documento dei requisiti così prodotta verrà poi presentata al committente per la sua approvazione. Di solito, sarà necessario effettuare diversi aggiustamenti e revisioni del documento, prima che questo possa essere considerato sufficientemente consolidato e stabile per procedere alla successiva fase di progettazione. In un processo iterativo, come già più volte ricordato, il documento dei requisiti non potrà mai, comunque, considerarsi finale: in ogni momento successivo alcuni aspetti potranno essere rivisti e modificati, sulla base delle nuove informazioni acquisite in corso di progetto.

La fase di esplorazione

La fase di esplorazione, nella quale vengono raccolti i requisiti, presenta spesso notevoli difficoltà. I problemi sono sostanzialmente di tre tipi:[95]

- Problemi di ambito. Generalmente, i "contorni" del sistema da progettare non sono ben definiti, ed esiste sempre il rischio di ampliare eccessivamente il campo di esplorazione. D'altro canto, restringendo troppo i temi da approfondire si rischia di tralasciare aspetti che potrebbero rivelarsi importanti nelle fasi successive. Inoltre, nelle conversazioni con gli stakeholder si è spesso tentati di abbozzare delle soluzioni di progetto. Questo è sbagliato: in questa fase ci si deve limitare alla sola raccolta dei requisiti, lasciando le attività di progettazione alle fasi successive, quando tutti i requisiti saranno stati individuati e organizzati in un insieme coerente.

- Problemi di comprensione. Questi avvengono a vari livelli. Da un lato, gli utenti hanno spesso una comprensione solo parziale dei loro bisogni, e una conoscenza piuttosto limitata delle possibilità offerte dalla tecnologia. Dall'altro, chi raccoglie i requisiti ha spesso una conoscenza limitata del dominio del problema, e utilizza un linguaggio differente da quello degli utenti e degli altri stakeholder. Ogni interlocutore tende a tralasciare gli aspetti che per lui sono ovvi, ma che potrebbero non esserlo affatto per interlocutori differenti.

- Problemi di conflitto. Stakeholder diversi possono avere punti di vista diversi sul sistema che dovrà essere progettato. Questi punti di vista potrebbero essere fra loro incompatibili: questi conflitti dovranno essere fatti emergere con chiarezza, e in qualche modo risolti nel documento dei requisiti finale.

- Problemi di volatilità. I requisiti evolvono nel tempo. Infatti, il contesto del sistema può mutare anche molto rapidamente e in modo inaspettato. Questi cambiamenti possono riguardare le condizioni di mercato, ricambio del management o ristrutturazioni dell'organizzazione committente, acquisizioni o vendite societarie, evoluzioni della tecnologia, e così via. Tutti questi fatti possono modificare in modo rilevante le priorità dei diversi requisiti, o addirittura modificarli completamente nel corso del progetto.

Le tecniche principali che possono essere utilizzate, nella fase di esplorazione, per la raccolta dei requisiti sono riassunte nella tabella di Figura 115, e descritte brevemente nei paragrafi che seguono.

Interviste individuali

La tecnica normalmente più usata è quella delle *interviste individuali* con il committente e i principali stakeholder del prodotto, perché permette di analizzare i singoli problemi in profondità. Gli intervistatori formulano le loro domande in colloqui individuali (faccia a faccia o telefonici) con ciascuno stakeholder, e raccolgono le risposte, annotando esigenze, suggerimenti, desideri e lamentele. Per ottenere la massima sincerità, di solito si garantisce agli intervistati che le loro opinioni verranno riportate solo in forma anonima.

[95] Cfr. anche M.G.Christel, K.C.Kang, *Issues in Requirements Elicitation*, Technical Report CMU/SEI-92-TR-012, settembre 1992 (disponibile in rete).

La scelta di chi intervistare va fatta con cura. Occorre prevedere un numero di interviste compatibile con le risorse e il tempo disponibili, ma senza tralasciare nessuna persona che possa avere qualcosa d'importante da dire sul prodotto in progettazione. Dovranno pertanto essere intervistati rappresentanti di ciascuna categoria di stakeholder. Poiché il committente è il referente principale del progetto, le sue indicazioni dovranno avere la massima attenzione. Sarà lui che stabilirà gli obiettivi principali, i tempi di realizzazione e il budget. Sarà lui che indicherà le persone da intervistare e sarà lui che revisionerà e approverà il documento dei requisiti finale. In caso di conflitto fra proposte alternative, sarà lui a decidere quale dovrà essere preferita.

Le interviste individuali possono essere più o meno strutturate. Le *interviste non strutturate* sono di carattere esplorativo, e assomigliano a delle conversazioni sugli argomenti d'interesse. L'intervistatore pone domande aperte, lasciando all'interlocutore la decisione se rispondere in modo breve o approfondito. È utile effettuare queste interviste sulla base di un canovaccio preparato in anticipo, in modo da essere sicuri di non tralasciare alcun aspetto rilevante. L'intervistatore potrà comunque orientare il colloquio diversamente da quanto pianificato, per esplorare eventuali aspetti non previsti inizialmente che emergessero nella conversazione. Le *interviste strutturate* prevedono, invece, un insieme di domande predefinite, come avviene nei questionari di cui tratteremo più oltre. A differenza dei questionari, esse sono comunque realizzate da un intervistatore, in colloqui individuali con gli intervistati. Le interviste strutturate sono utili soprattutto quando gli obiettivi del colloquio siano stati bene identificati, e sia possibile definire un insieme di domande molto specifiche, che richiedono risposte precise. Di solito queste domande sono poste in forma identica a tutti gli intervistati; in questo modo, le risposte possono essere sottoposte ad analisi statistiche. Le *interviste semi-strutturate* contengono sia domande libere, con carattere esplorativo, sia domande specifiche.

Condurre bene un'intervista può non essere facile e richiede esperienza. Occorre evitare di influenzare l'intervistato, formulando le domande in modo che non contengano implicitamente già la risposta. È necessario, inoltre, concentrarsi sui problemi e non sulle soluzioni: si dovrà sempre ricordare che l'obiettivo è quello di identificare i requisiti, e non di effettuare scelte di progetto. Queste dovranno essere fatte in seguito, a fronte di un quadro completo e organico dei vincoli esistenti. In ogni caso, l'intervistatore dovrà evitare di usare termini tecnici, cercando di parlare nel linguaggio dell'intervistato. In molti casi ci si accorgerà ben presto che è necessario chiarire bene il significato di alcuni termini, che possono essere usati dagli intervistati con accezioni particolari. Ogni organizzazione sviluppa col tempo un proprio gergo, che può creare fraintendimenti con interlocutori esterni. Può essere quindi conveniente approfittare delle interviste per definire un sintetico *glossario*. Cioè una lista dei termini più importanti utilizzati nel progetto, con le loro definizioni in relazione allo specifico contesto. Questo glossario, allegato ai requisiti, permette di stabilire una base di conoscenza comune fra gli stakeholder del prodotto e il gruppo di progetto.

Tecnica	Serve per	Vantaggi	Svantaggi
Questionari	Rispondere a domande specifiche.	Si possono raggiungere molte persone con poco sforzo.	Vanno progettati con grande accuratezza, in caso contrario le risposte potrebbero risultare poco informative. Il tasso di risposta può essere basso.
Interviste individuali	Esplorare determinati aspetti del problema e determinati punti di vista.	L'intervistatore può controllare il corso dell'intervista, orientandola verso quei temi sui quali l'intervistato è in grado di fornire i contributi più utili.	Richiedono molto tempo. Gli intervistati potrebbero evitare di esprimersi con franchezza su alcuni aspetti delicati.
Focus group	Mettere a fuoco un determinato argomento, sul quale possono esserci diversi punti di vista.	Fanno emergere le aree di consenso e di conflitto. Possono far emergere soluzioni condivise dal gruppo.	La loro conduzione richiede esperienza. Possono emergere figure dominanti che monopolizzano la discussione.
Osservazioni sul campo	Comprendere il contesto delle attività dell'utente.	Permettono di ottenere una consapevolezza sull'uso reale del prodotto che le altre tecniche non danno.	Possono essere difficili da effettuare e richiedere molto tempo e risorse.
Suggerimenti spontanei degli utenti	Individuare specifiche necessità di miglioramento di un prodotto.	Hanno bassi costi di raccolta. Possono essere molto specifici.	Hanno normalmente carattere episodico.
Analisi della concorrenza e delle best practices	Individuare le soluzioni migliori adottate nel settore di interesse.	Evitare di "reinventare la ruota" e ottenere vantaggio competitivo.	L'analisi di solito è costosa (tempo e risorse)

Figura 115. Le principali tecniche utilizzate nella fase di esplorazione dei requisiti

Questionari

I *questionari* permettono di raccogliere informazioni in forma strutturata, elaborabili con metodi statistici. Essi possono essere distribuiti ai destinatari in vari modi. Per esempio, si possono predisporre dei questionari compilabili online, generando delle pagine web contenenti le domande del questionario. È così possibile raggiungere una popolazione potenzialmente molto ampia di utenti, anche se, di solito, il tasso di risposta (*redemption*) è piuttosto basso. Esistono numerosi strumenti software (alcuni anche gratuiti, reperibili in rete), che permettono, da un lato, di costruire facilmente il questionario e, dall'altro, di elaborare i risultati e produrne una visione di sintesi attraverso grafici e diagrammi.

Una tecnica molto usata nei questionari destinati a raccogliere le opinioni degli utenti è la cosiddetta scala di Likert.[96] Il questionario è composto da una serie di affermazioni, collegate alle opinioni su cui si vuole indagare, per ciascuna delle quali sono possibili cinque risposte: *completamente d'accordo*, *d'accordo*, *incerto*, *in disaccordo*, *in completo disaccordo*. A ciascuna risposta è associato un numero compreso fra 1 e 5. Con questi valori si potrà calcolare la media delle risposte a ciascun gruppo di affermazioni correlate a uno stesso argomento.

[96] La tecnica fu ideata nel 1932 dallo psicologo americano Rensis Likert, con lo scopo di fornire uno strumento semplice per la misurazione di opinioni e atteggiamenti, ed è molto usata nella ricerca sociale.

Focus group

I *focus group* sono interviste di gruppo, che hanno lo scopo di mettere a fuoco uno specifico argomento e di far emergere i diversi punti di vista dei partecipanti o, a volte, un punto di vista condiviso fra tutti. Sono normalmente condotti da un animatore che guida la discussione e un osservatore che esamina le dinamiche di relazione del gruppo e prende appunti. La conduzione di un focus group non è compito banale e richiede esperienza. È necessario infatti evitare che il gruppo "sfugga di mano". Quando emergerà il leader naturale, tenderà a monopolizzare la discussione e a trascinare il gruppo sulle sue posizioni. Il conduttore dovrà evitare che l'incontro diventi un'occasione di sfogo di malumori e critiche poco attinenti al tema, o di promozione di scopi personali. Occorre fare in modo che tutti possano esprimere le loro idee e abbiano adeguato spazio nella discussione e che non sorgano conflitti fra i conduttori e i membri del gruppo, che potrebbero danneggiare lo svolgimento successivo del progetto.

Osservazioni sul campo

Non sempre gli utenti sono in grado di spiegare in dettaglio quali sono le modalità di uso desiderate per il prodotto nella loro attività quotidiana. Potrebbero anche avere un'immagine distorta di come si comportano nelle varie situazioni d'uso reali. Questo non deve stupire: normalmente un utente non ha interesse a conoscere in dettaglio la natura e la frequenza dei compiti che svolge quotidianamente, li svolge e basta. Uno studio sul campo per apprendere come gli utenti si comportano nella realtà può quindi essere molto istruttivo e riservare alcune sorprese. Purtroppo questo non è facile, può essere molto costoso, considerando anche la possibile varietà delle diverse tipologie di utenti. Gli studi sul campo vengono fatti con i metodi della etnografia, che abbiamo discusso a pag. 110.

Suggerimenti spontanei degli utenti

Queste informazioni sono preziose per una corretta evoluzione del prodotto e dovrebbero sempre essere sistematicamente raccolte e classificate. Una fonte molto importante d'informazioni di questo tipo è costituita dai *forum* di discussione relativi ai vari prodotti, che di solito esistono sul Web. È possibile, inoltre, attivare un sito sul quale gli utenti segnalano spontaneamente miglioramenti desiderabili, e "votano", con tecniche ormai abituali sul Web, i suggerimenti proposti.

Analisi della concorrenza e delle best practice

Un'altra attività importante nella fase di esplorazione dei requisiti è l'analisi dei prodotti concorrenti, cioè di quei prodotti con i quali il nostro prodotto dovrà confrontarsi e competere. L'analisi della concorrenza potrà essere più o meno ampia, in funzione del numero e della complessità dei prodotti esaminati e del livello di approfondimento dell'esame. Per certi settori, può essere molto complessa e costosa. Si dovrà esaminare un certo numero di prodotti, per individuarne le caratteristiche più importanti e, soprattutto, i punti di forza e di debolezza: ciò permetterà di meglio contraddistinguere il prodotto in costruzione in rapporto ad essi e definirne, come si dice, la sua *value proposition*, cioè il valore specifico e distintivo che dovrà fornire ai suoi utenti. Inoltre, quest'analisi permetterà d'individuare le *pratiche migliori* adottate dai prodotti del settore, dalle quali trarre spunti per la formulazione dei requisiti. È utile effettuare quest'analisi proprio all'inizio del progetto; infatti, durante le interviste di raccolta dei requisiti si potranno ottenere utili commenti sulle soluzioni adottate da altri e sulla loro applicabilità nel contesto corrente.

Scenari d'uso

Una tecnica molto utile per aiutarci a immaginare un nuovo prodotto, e a individuarne correttamente i requisiti, è quella d'ipotizzarne dei possibili *scenari d'uso*. Uno scenario d'uso è una narrazione, in linguaggio comune, di una possibile storia dell'uso del sistema da parte di uno specifico utente, fittizio ma in qualche modo tipico, e descritto in modo molto realistico. L'esempio che segue riporta un possibile scenario d'uso del sito Web di un cinema multisala.

> Marco è uno studente universitario. È appassionato di cinema, anche se le sue possibilità economiche sono molto limitate. Sceglie i film da vedere con molta cura e preferisce vederli dalle prime file. Però gli capita spesso che il posto gli sia assegnato d'autorità dal computer della biglietteria, senza possibilità di scelta. Questo succede anche nel multisala vicino a casa sua. Per questo motivo, quando ha saputo che il cinema ha un nuovo sito Internet che permette, agli utenti registrati, di scegliere personalmente il posto, si è subito registrato. Ora, quando vuole andare al cinema, Marco si collega al sito e

procede velocemente con l'operazione di prenotazione che è accessibile direttamente dalla home page. Inserisce nome utente e password e il sistema autorizza l'operazione fornendo come risposta le diverse opzioni di scelta. Marco ora può scegliere tra i titoli dei film in programmazione, il giorno della settimana e l'ora. A questo punto gli viene presentata la mappa della sala cinematografica, nella quale sono indicati i posti liberi (in verde) e quelli già prenotati (in rosso). Marco finalmente può scegliere il posto che preferisce facendo clic sulla figura e, dopo averlo confermato, avrà un resoconto dell'operazione, che gli sarà anche inviato con un messaggio di posta elettronica. La sera, almeno 15 minuti prima dell'inizio della proiezione, Marco si presenta alle casse del multisala con un documento d'identità. La cassiera procede a stampare i biglietti prenotati, che Marco paga. A questo punto Marco potrà accomodarsi nella sala cinematografica e vedere la proiezione del film direttamente dalla poltrona prescelta.

L'impiego degli scenari d'uso è molto utile nella progettazione di un prodotto. Durante la definizione dei requisiti, serve principalmente come mezzo di comunicazione con i diversi stakeholder e, in seguito, con i progettisti e gli sviluppatori. L'ideazione di storie d'uso tipiche e concrete è, infatti, un modo molto efficace per collocare il prodotto, ancora da progettare, nei suoi possibili contesti d'uso reali. Ognuno di noi, infatti, tende ad assumere dei "sistemi di riferimento" che considera ovvi e che quindi non ritiene necessario esplicitare o spiegare. Poichè i sistemi di riferimento dei nostri interlocutori non sono necessariamente identici ai nostri, è facile che nascano fraintendimenti ed equivoci che, nella progettazione di un prodotto complesso, possono essere molto dannosi. Equivoci nella fase di definizione dei requisiti produrranno un prodotto con caratteristiche diverse da quelle desiderate: è bene che emergano e siano chiariti al più presto. Gli scenari d'uso sono uno strumento molto efficace per questo scopo.

Inoltre, quando progettiamo un prodotto, siamo portati inevitabilmente a considerare noi stessi come utenti tipici: tendiamo quindi a modellare il prodotto sui nostri bisogni, abitudini e preferenze. Questo è sbagliato, perché gli utenti "veri" del prodotto avranno normalmente bisogni, abitudini e preferenze diverse. D'altro canto, è molto facile cadere in questa trappola: scrivere uno scenario vissuto da personaggi dotati di una loro specifica identità, ci aiuta a considerare un prodotto in modo più oggettivo. Pertanto, è molto importante che i protagonisti di uno scenario siano persone concrete, anche se fittizie, che rappresentano in qualche modo degli "utenti archetipi" del sistema. Devono essere dotati di una precisa identità; altrimenti, se pensiamo agli utenti come semplici ruoli astratti (per esempio, "studente universitario"), il rischio di mancare di concretezza e di perdere di vista le esigenze degli utenti reali è molto alto.

Ai questi utenti archetipi che le cui storie raccontiamo negli scenari d'uso si dà spesso il nome di *personae* (il plurale della parola latina *persona*). Una persona non dovrebbe essere inventata, ma creata a partire da una ricerca etnografica (pag.110), come sintesi di varie caratteristiche presenti negli utenti reali, e descritta in un *profilo* che ne elenca obiettivi, competenze, comportamenti, preferenze, ambiente sociale, stile di vita. In una parola, tutto ciò che si ritiene rilevante per caratterizzarne il rapporto con il prodotto che dovrà essere progettato. La Figura 116 mostra un esempio di alcune personae rappresentate su supporti di cartone. Queste rappresentazioni, tenute sulle scrivanie dei progettisti e accompagnate dal loro profilo, contribuiscono a ricordare costantemente a chi il progetto è destinato.



Figura 116. Sagome di cartone rappresentanti le personae di uno scenario[97]

[97] Tratto dalla presentazione di S.Mulder, *The User is Always Right – Making Personas Work for Your Site*, in http://www.slideshare.net/MulderMedia/the-user-is-always-right-making-personas-work-for-your-site.

Come si vede nell'esempio del cinema, è opportuno che, nella formulazione di uno scenario d'uso, sia riportata una storia completa, che non si limiti, quindi, alla pura interazione con il sistema, ma che ne consideri il contesto complessivo. In questo modo è facile che emergano dei requisiti impliciti, che potrebbero altrimenti essere facilmente trascurati. Così, la storia di Marco ce ne descrive la motivazione principale (la possibilità di scegliere il posto al cinema) e ci mostra le azioni compiute da Marco dopo aver completato la transazione al computer. Tutto questo aiuta il redattore dei requisiti a non trascurare aspetti importanti e a porre la giusta enfasi sugli aspetti chiave. Anche i progettisti ricaveranno utili informazioni dall'esame degli scenari d'uso. Per esempio, chi, in seguito, progetterà il sistema, comprenderà meglio il motivo per cui le funzioni per la selezione del posto a sedere debbano essere particolarmente flessibili e usabili.

Naturalmente, la storia deve "mettere in scena" situazioni tipiche. Per esempio, lo scenario appena visto potrebbe essere giustificato da un'indagine presso gli spettatori che abbia mostrato che la scelta del posto al cinema è importante per un numero rilevante di persone, e non solo per l'ipotetico utente Marco. Durante le interviste, si potrà chiedere agli intervistati d'immaginare gli scenari d'uso che ritengono più tipici. Dall'approfondimento di questi scenari potranno emergere requisiti che altrimenti sarebbero trascurati. A volte, intervistato e intervistatore discuteranno scenari alternativi. Si potranno chiedere, per esempio, se nel caso del cinema multisala l'affollamento del sabato sera possa creare delle difficoltà nel ritiro dei biglietti prenotati, e come si possano evitare code. Queste analisi, che a volte, come in questo caso, non coinvolgono direttamente le funzioni del prodotto, potrebbero suggerire soluzioni alternative più convenienti.

La storia di Marco è piuttosto semplice, perché coinvolge un solo caso d'uso del sistema: l'acquisto del biglietto. Altri scenari, più complessi, possono coinvolgere più casi d'uso, come lo scenario relativo ai sistemi d'intelligenza ambientale visto a pag.55, o il seguente:

> Marco, studente universitario, ha sempre con sé il suo netbook. Salito sul treno per rientrare a casa dalle lezioni in università, apre il suo netbook per rivedere gli appunti presi a lezione. La carica della batteria gli permette di lavorare per tutta la durata del tragitto. Grazie alle dimensioni dello schermo e della tastiera, Marco riesce a leggere e a scrivere agevolmente, tenendo il netbook sulle ginocchia: ciò gli permette di concludere la sistemazione degli appunti della lezione di Diritto durante il viaggio. Arrivato a casa, Marco collega il netbook alla corrente, per ricaricare la batteria mentre corregge gli appunti di Psicologia. Conclusi anche questi, prima di cena si connette al forum del corso e pubblica i suoi appunti in rete, per metterli a disposizione dei suoi compagni di corso.

Come indicato dalle frasi sottolineate, questo scenario coinvolge tre casi d'uso del netbook: *Correggi gli appunti*, *Ricarica la batteria*, *Pubblica sul forum*.

Lo scenario seguente, ancora più complesso, è tratto da un documento dei requisiti per un sistema di gestione delle cartelle cliniche.[98] Anche qui, le frasi che identificano i casi d'uso del sistema sono sottolineate.

> Andrea è un medico pneumologo all'ospedale …omissis… di Milano. La sua attività lavorativa si suddivide tra ambulatorio e reparto. Andrea in ambulatorio ha ogni giorno n appuntamenti prefissati e attraverso un archivio dell'ambulatorio recupera le cartelle cliniche dei pazienti che hanno appuntamento quel giorno, così può iniziare a visitarli nell'ordine di arrivo.
>
> La visita può essere una prima visita o una visita di controllo. Nel primo caso Andrea raccoglie la storia clinica e scrive l' anamnesi del paziente e in base alle patologie e sintomi rilevati cerca di capire quale problema affligge il paziente. Se il caso in questione è semplice o richiede esami che possono essere effettuati direttamente in ambulatorio in modo da poter visualizzare immediatamente i risultati Andrea può prescrivere fin da subito la terapia più adatta alla circostanza. Se il caso è più complesso e richiede esami più specifici (molte volte si effettuano in altri ambulatori o strutture) allora il medico prescrive gli esami che il paziente deve fare e gli fissa un nuovo appuntamento in cui potrà vedere gli esiti di tali esami appena richiesti.
>
> Se la visita è un controllo, Andrea legge la storia clinica del paziente, valuta i documenti portati dal paziente (valori degli esami che aveva richiesto) e segue le indicazioni che aveva segnato sulla cartella clinica nella visita precedente. Anche in questo caso, se il caso in questione lo permette prescrive la terapia da seguire oppure fissa un nuovo appuntamento.

[98] Requisiti tratti dalla tesi di laurea magistrale in Informatica di Ignazio Gentile, Università degli Studi di Milano Bicocca, A.A.2008-2009.

Andrea, come detto in precedenza, oltre all'ambulatorio lavora anche nel reparto di pneumologia dell'ospedale. Ogni giorno effettua un giro visite dei pazienti ricoverati. In tal modo visita ogni paziente, vede la diagnosi d'ingresso e gli esami effettuati dal paziente. Se ritiene opportuno stabilisce altri esami diagnostici e prescrive, cambia, o conferma la terapia. Se la clinica del paziente e i valori degli esami lo permettono Andrea può decidere anche di dimettere il paziente. Un caso particolare sono le urgenze e le emergenze. Qui il medico deve intervenire prontamente e cercare di stabilizzare il paziente. In questa fase la diagnosi è poco accurata e ci si basa più sulla clinica manifestata dal paziente. Altra attività che Andrea può svolgere sono le visite a parere, in questo caso il medico effettua consulenze in altri reparti dell'ospedale che hanno richiesto il suo intervento.

Gli scenari d'uso possono essere molti utili, ma scegliere quelli realmente significativi da inserire nel documento dei requisiti non è facile. Il rischio maggiore è quello di scadere nell'aneddotica, raccontando storie poco rilevanti per la comprensione dei requisiti del prodotto, che quindi non saranno di alcuna utilità nelle successive fasi di progettazione. Nel caso di prodotti di nuova concezione, destinati a innovare i comportamenti degli utenti, si realizzano a volte degli scenari d'uso con delle riprese video. Un esempio molto famoso, è il video del *Knowledge Navigator*, realizzato dalla Apple nel 1987 (Figura 117). Esso mostrava un possibile scenario d'uso di un personal computer del futuro (il "futuro", allora, era collocato nell'anno 2010) basato sul concetto di *agente*. Nel video, un professore universitario parlava con un aiutante sintetico, rappresentato sul monitor con sembianze umane, per raccogliere, in rete o facendosi aiutare da altri agenti remoti, i dati necessari per la stesura di un articolo scientifico.[99]



Figura 117. Knowledge Navigator (Apple, 1987)

I casi d'uso

Nel capitolo 5, a pag.119, è stata introdotta la nozione di caso d'uso. La descrizione dei casi d'uso del sistema costituisce un capitolo molto importante del documento di specifica dei requisiti. Pertanto, le prossime pagine saranno dedicate ad approfondire questo argomento.

Come abbiamo già visto, un caso d'uso (*use case*) può essere definito come un insieme di interazioni finalizzate a uno scopo utile, fra uno o più *attori* e il sistema. Esso ha un *nome* che, come abbiamo già visto, di solito è composto da un

[99] In rete, per esempio su http://www.youtube.com, esistono numerose copie di questo video. È molto interessante confrontare questo design concept con le caratteristiche dell'iPad, il prodotto - anch'esso molto innovativo - lanciato effettivamente dalla Apple sul mercato nel 2010, l'anno in cui, nelle ipotesi di oltre due decadi prima, avremmo dovuto disporre delle tecnologie necessarie a realizzare il Knowledge Navigator. A riprova del fatto che è sempre molto, molto difficile fare previsioni sul futuro.

verbo, alla terza persona singolare o all'infinito, e da un complemento, oppure da una frase che descrive sinteticamente lo scopo dell'interazione. Per esempio, in un sito web di e-commerce:

- Acquista prodotto
- Modifica il profilo utente
- Ricercare un prodotto nel catalogo
- Inserire un nuovo prodotto in catalogo
- Modificare i dati di un prodotto.

Un caso d'uso viene *invocato* (cioè iniziato) da un attore per un particolare scopo e si conclude con successo quando tale scopo è raggiunto. Ogni attore non è una persona concreta, come negli scenari che abbiamo appena descritto, ma rappresenta un particolare *ruolo* nell'interazione con il sistema (vedi pag.81). Per esempio, nel nostro sistema di e-commerce, potremmo avere tre attori, denominati Utente, Utente registrato e Amministratore del sistema. Ciascuno di questi attori invocherà casi d'uso specifici per il suo ruolo. Per esempio, l'Amministratore del sistema potrà Inserire un nuovo prodotto nel catalogo, mentre l'Utente potrà soltanto Ricercare un prodotto nel catalogo, senza poterne modificare la descrizione. Se un caso d'uso coinvolge più attori, quello che persegue lo scopo che il caso d'uso deve soddisfare sarà considerato l'*attore principale*. Solitamente, ma non sempre, esso è quello che dà inizio al caso d'uso con la sua invocazione.

Diagramma dei casi d'uso

Nel documento dei requisiti, è consigliabile aggiungere all'elenco dei casi d'uso un diagramma riassuntivo, detto *diagramma dei casi* d'uso (*use case diagram*), che mostra le relazioni fra gli attori e i casi d'uso del sistema (Figura 118).

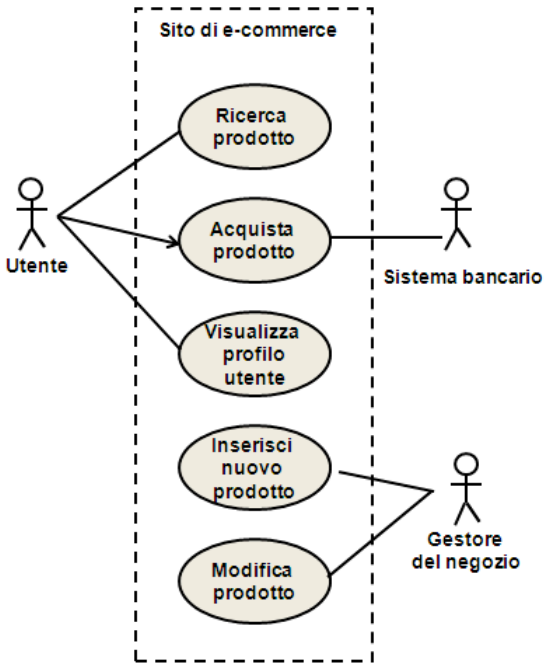


Figura 118. Un diagramma dei casi d'uso

In questi diagrammi, gli attori sono rappresentati con omini stilizzati e i casi d'uso con ellissi. Il nome dell'attore viene indicato sotto l'omino e quello del caso d'uso dentro l'ellissi (Figura 119).

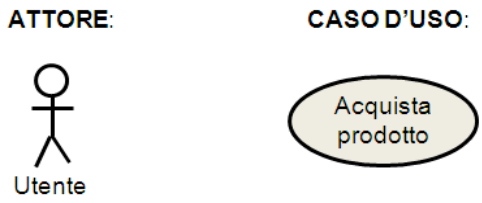


Figura 119. Rappresentazione grafica di attori e casi d'uso

Gli attori coinvolti in un caso d'uso non devono necessariamente essere umani: possono anche essere dei sistemi con i quali il caso d'uso interagisce, per inviare o ricevere dati, o per richiedere dei servizi. Così, in Figura 118, il caso d'uso Acquista Prodotto coinvolge l'attore Sistema bancario, per gestire il pagamento attraverso carta di credito. Anche gli attori non umani sono tradizionalmente rappresentati con degli omini stilizzati: per indicare che non si tratta di persone in carne e ossa, nel diagramma dei casi d'uso viene allora convenzionalmente riportata, sotto il nome dell'attore, la dicitura convenzionale <<sistema>>.

L'associazione fra un attore e un caso d'uso è rappresentata da un segmento che li congiunge e può indicare una (o più) situazioni fra le seguenti:

- l'attore esegue il caso d'uso;
- l'attore fornisce informazioni al caso d'uso;
- l'attore riceve informazioni dal caso d'uso.

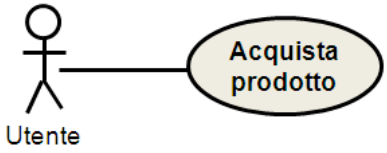


Figura 120. Rappresentazione della relazione fra un attore e un caso d'uso

A volte, al posto dei segmenti, si utilizzano delle frecce per indicare che l'attore esegue (o, come si dice, *invoca*) il caso d'uso:

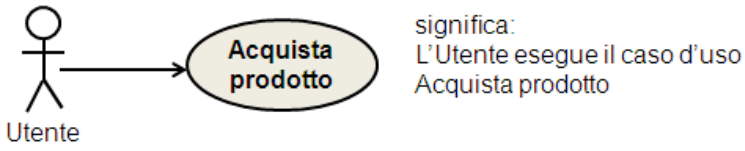


Figura 121. Rappresentazione con un arco orientato

Si noti che la direzione della freccia non specifica la direzione del flusso dei dati (che possono fluire in un senso o nell'altro), come si potrebbe supporre per analogia con altri tipi di diagrammi d'uso comune. Per evitare possibili fraintendimenti, si consiglia quindi di limitare l'uso delle frecce ai soli casi in cui possano sussistere delle ambiguità su chi invoca che cosa. Nel diagramma di Figura 118, poiché il caso d'uso Acquista prodotto è associato a due attori, è stata usata la notazione a freccia per indicare che è l'utente umano, e non il sistema, che lo invoca. Negli altri casi, non sussistendo ambiguità, sono stati usati dei segmenti non orientati.

Un diagramma che rappresenta tutti i casi d'uso di un sistema si chiama *diagramma di contesto* del sistema, perché indica i "confini" dello stesso e tutti gli attori che lo utilizzano. In questo caso, il sistema si indica con un riquadro che circonda i casi d'uso e ne riporta il nome, come nel nostro esempio.

Descrizione dei casi d'uso

Nel documento dei requisiti, a ogni caso d'uso mostrato nel diagramma dovrebbe essere associata una descrizione, per far comprendere al lettore di che cosa si tratta. Essa dovrebbe essere espressa in un linguaggio semplice e informale, comprensibile a chiunque. Infatti, come abbiamo visto, il documento dei requisiti viene utilizzato non solo dai progettisti del sistema, ma anche dagli stakeholder che contribuiscono alla loro stesura. Questi, nella grande maggioranza dei casi, non hanno le competenze tecniche necessarie per leggere delle descrizioni in linguaggi troppo formalizzati. Si usa quindi il linguaggio naturale, avendo l'accorgimento di utilizzarlo nel modo meno ambiguo possibile e senza ridondanze.

Anche se non sono state definite delle regole standard, si è consolidata la prassi di descrivere un caso d'uso specificandone i diversi scenari d'uso. Questi scenari, a differenza di quelli esemplificati nella sezione precedente, non fanno riferimento a utenti concreti, con nome e cognome, che operano in specifici contesti, ma agli attori identificati nel diagramma dei casi d'uso, in situazioni generiche, senza riferimento ad alcun contesto specifico. Si tratta, per così dire, di scenari d'uso "astratti" e ridotti ai minimi termini, che servono esclusivamente a chiarire il significato che il redattore del documento dei requisiti attribuisce ai vari casi d'uso. Alcuni di questi scenari permettono di raggiungere lo scopo, altri portano alla conclusione del caso d'uso senza che lo scopo sia raggiunto. Per esempio, nel caso di Acquista prodotto, la carta di credito potrebbe non essere valida, e il caso d'uso si concluderebbe senza alcun acquisto. Questi scenari alternativi presentano delle differenze, ma sono accomunati dal fatto che l'utente ha sempre lo stesso scopo, nel nostro caso, acquistare un prodotto. Può darsi che l'acquisto non vada a buon fine, ma l'intento rimane.

La forma più comune della descrizione di un caso d'uso è esemplificato nella Figura 122, che fa riferimento al solito Acquista prodotto del sito web di e-commerce. Viene descritto prima lo *scenario principale di successo*, detto anche *corso d'azione base*, cioè quello che è ritenuto più frequente o importante, e che si conclude con il successo del caso d'uso: l'utente raggiunge il suo scopo. Esso descrive il flusso principale degli eventi d'interazione, espresso come sequenza di *passi* numerati, ciascuno dei quali corrisponde a un'interazione fra uno o più attori e il sistema.

Nome: Acquista prodotto

Attori: Utente, sistema bancario

Scenario principale:

1. Il cliente ricerca nel catalogo il prodotto da acquistare
2. Il sistema chiede di completare l'ordine di acquisto
3. Il cliente fornisce i dati richiesti, compreso l'indirizzo di consegna
4. Il sistema presenta il conto finale, comprese le spese di spedizione
5. Il cliente accetta e fornisce le informazioni per il pagamento con carta di credito
6. Il sistema autorizza l'acquisto
7. Il sistema conferma la vendita
8. Il sistema invia al cliente una e-mail di conferma.

Estensioni:

3a. Il cliente è preregistrato:

1. Il sistema visualizza le preferenze memorizzate: indirizzo di consegna e informazioni per il pagamento con carta di credito
2. Il cliente può accettare le preferenze memorizzate o ridefinirle
3. Il sistema presenta il conto finale, comprese le spese di spedizione
4. Il cliente accetta, quindi si prosegue dal passo 6

5a Il cliente non accetta e rinuncia all'acquisto

6a. Il sistema non autorizza l'acquisto:

1. Il cliente inserisce nuovamente le informazioni sulla carta di credito e riprova,
oppure rinuncia all'acquisto

Figura 122. Descrizione del caso d'uso *Acquista prodotto*

Seguono poi gli altri scenari, detti *scenari alternativi*. Essi sono descritti come *estensioni* di quello principale, cioè specificando solo le differenze con esso, senza ripetere tutto, per non appesantire troppo la descrizione.

Per indicare un'estensione si scrive la *condizione* che determina il verificarsi di una sequenza d'interazioni alternativa rispetto a quella dello scenario principale, specificando poi le differenze. Prima di tutto si scrive il numero del passo in cui si può verificare la condizione, con una breve descrizione della stessa; quindi si aggiungono dei passi numerati espressi nello stesso stile dello scenario principale. Alla fine si indica da quale passo si rientra nel flusso di eventi base. Nel nostro esempio, le estensioni sono tre, descritte come alternative, rispettivamente, dei passi 3, 5 e 6, corrispondenti alle tre condizioni Il cliente è preregistrato, Il cliente non accetta e rinuncia all'acquisto, Il sistema non autorizza l'acquisto.

Ogni passo di uno scenario dovrebbe essere espresso con una frase semplice, che faccia chiaramente capire chi lo sta eseguendo. Non si dovrebbero mai indicare i dettagli delle azioni di un attore, ma il loro scopo. Inoltre, non si dovrebbero mai descrivere i particolari dell'interfaccia utente: bisogna sempre ricordare che si stanno specificando i requisiti di un sistema, e che le scelte di come realizzarlo devono essere lasciate ai progettisti, che interverranno nelle fasi successive. Per lo stesso motivo, il sistema viene visto come una "scatola nera" e il suo funzionamento interno non

viene considerato. In sostanza, un caso d'uso specifica *chi* (l'attore o gli attori), *che cosa* (l'interazione) e *perché* (lo scopo), senza entrare nel merito del funzionamento interno del sistema.

Se un caso d'uso ne richiama un altro, il nome di quest'ultimo viene sottolineato. Si usa la sottolineatura, perché suggerisce un collegamento ipertestuale, e chiunque lo può capire. Così, nella Figura 122, il primo passo richiama il caso d'uso *Ricerca nel catalogo il prodotto da acquistare*, che viene descritto a parte, con la stessa tecnica. Si dice allora che questo caso d'uso *è incluso* nel precedente.

L'inclusione può essere utile per esprimere con un singolo passo una sequenza di passi più elementari, che renderebbe pesante la descrizione dello scenario, oppure, più spesso, per raccogliere "a fattor comune" una sequenza di passi che si ripetono più volte nello stesso o in diversi casi d'uso.

Nella descrizione dei casi d'uso non è mai consigliabile scendere a un livello di dettaglio troppo basso, decomponendo i casi in sottocasi e questi in sotto-sottocasi. Nel documento dei requisiti è conveniente mantenersi a un livello ancora piuttosto generale. Ciò che interessa è dare al lettore un'immagine abbastanza chiara dei diversi casi d'uso, che permetta di comprendere con ragionevole precisione e senza ambiguità ciò che il sistema deve fare e non *come* deve farlo. Le descrizioni troppo lunghe e dettagliate finiscono per non essere neppure lette, il che le renderebbe ben poco utili. Sarà poi compito delle successive attività di progettazione decomporre ogni caso d'uso nei compiti che lo compongono, e questi nelle azioni elementari che l'utente dovrà effettuare.

Alistair Cockburn, autore di un libro sui casi d'uso, dà le seguenti indicazioni:[100]

> *Molti si sentono colpevoli se lo scenario principale di un caso d'uso è breve, così lo allungano per arrivare a quindici, o anche trentacinque righe. Personalmente, io non ho mai scritto uno scenario principale più lungo di nove passi. Non che il nove sia un numero magico; il fatto è che, quando ho individuato i sotto-goal a un giusto livello e ho eliminato i dettagli che riguardano la progettazione, mi restano sempre meno di nove passi. A volte, lo scenario principale di un caso d'uso può essere anche di soli tre passi.*
>
> *Il valore maggiore di un caso d'uso non è nello scenario principale, ma nei comportamenti alternativi. Lo scenario principale può occupare da un quarto a un decimo della lunghezza totale di un caso d'uso, perché ci possono essere molte alternative da descrivere. Se lo scenario principale fosse lungo trentacinque passi, l'intero caso d'uso occuperebbe dieci pagine, e sarebbe troppo lungo da leggere e da comprendere. Se lo scenario principale contiene da tre a nove passi, la descrizione complessiva potrebbe essere di solo due o tre pagine, il che è più che sufficiente.*
>
> *Se potete evitare di includere troppi dettagli dell'interfaccia utente, i casi d'uso saranno molto più facili da leggere. E i casi d'uso leggibili possono in effetti venire letti. Casi d'uso lunghi e illeggibili vengono soltanto firmati – di solito con sgradevoli conseguenze sul progetto, alcuni mesi più tardi.*

Non bisogna confondere gli scenari d'uso introdotti a pag. 149 con i casi d'uso: sono due cose diverse, che perseguono obiettivi differenti. Gli scenari d'uso che abbiamo introdotto in precedenza hanno lo scopo di illustrare situazioni tipiche di uso del sistema, per farne comprendere la portata e fare emergere eventuali requisiti impliciti. Sono storie tipiche molto concrete dell'uso del sistema, raccontate con particolari che permettano di comprenderne le motivazioni e il contesto. Spesso raccontano situazioni che coinvolgono più casi d'uso. Anche quando lo scenario riguarda un singolo caso d'uso, come la prenotazione del biglietto del cinema (pag.149), esso ne descrive una specifica istanza, e lo arricchisce d'informazioni aggiuntive. Nell'esempio, ci venivano descritte le motivazioni di Marco a effettuare la prenotazione online, e le sue azioni dopo la prenotazione, fuori dal sistema: il ritiro dei biglietti alla cassa, il pagamento, e così via.

La descrizione dei casi d'uso costituisce un passo logicamente successivo alla creazione degli scenari d'uso. In questo passo, s'identificano tutte le interazioni che, a un certo livello di astrazione e dal punto di vista dei vari attori coinvolti, dovranno avere luogo con il sistema, e li si descrive cercando di tener conto delle principali alternative. La descrizione di un caso d'uso non fa riferimento a personaggi concreti, ma a ruoli astratti incarnati dai vari attori, e contiene solo informazioni sull'interazione che questi hanno con il sistema, senza alcuna informazione aggiuntiva sul contesto. Sono,

[100] Il brano è tratto dalla nota del 2002 *Use cases, ten years later*, disponibile in rete. Il libro citato è Alistair Cockburn, *Writing Effective Use Cases*, Addison-Wesley, 2001.

per così dire, collezioni di scenari ridotti ai minimi termini, che hanno lo scopo di fissare gli aspetti principali del flusso dell'interazione con il sistema, che dovranno poi essere ulteriormente sviluppati nella fase di progettazione.

Generalizzazione, inclusione, estensione di casi d'uso

Durante l'organizzazione del documento dei requisiti, per effettuare la descrizione dei casi d'uso del sistema conviene partire dalla formulazione di un primo elenco, che verrà via via modificato per raggiungere un livello di dettaglio soddisfacente. Se i casi d'uso risultano troppo dettagliati, si passa a un livello di astrazione maggiore; se sono troppo generali, li si decompone ulteriormente. Così, in un supermercato online, Fa la spesa è troppo generale, e lo si decompone, per esempio, in Ricerca prodotto e Acquista prodotto. Al contrario, Fornisce i dati della carta di credito potrebbe non essere considerato un caso d'uso a sé stante, perché troppo dettagliato, ma soltanto un passo di Acquista prodotto. Non esistono regole fisse per determinare il livello di astrazione corretto: sarà la sensibilità dell'estensore del documento a guidarlo nella scelta. Il criterio, come già detto, è quello di ottenere una descrizione sufficientemente dettagliata da essere utile per far capire di che cosa si parla, ma non così dettagliata da scoraggiarne la lettura. I casi d'uso così definiti saranno raccolti nel diagramma dei casi d'uso del sistema, per avere una visione generale, prima di procedere alle singole descrizioni.

Alcuni casi d'uso possono rivelarsi casi particolari di altri casi d'uso. Per esempio, in un negozio online che vende libri e CD musicali, i due casi d'uso Acquista libro e Acquista CD potrebbero essere considerati casi particolari del caso d'uso più generale Acquista prodotto. Allora, si dice che Acquista prodotto è una *generalizzazione* di ciascuno degli altri due casi d'uso. Al contrario, si può dire che Acquista libro (o Acquista CD) è una *specializzazione* di Acquista prodotto.

La generalizzazione può essere rappresentata graficamente nel diagramma mediante una freccia con la punta a triangolo, come nella Figura 123. La stessa notazione grafica può essere usata anche per rappresentare la relazione di generalizzazione fra attori. Per esempio, sempre in Figura 123, Cliente è indicato come una generalizzazione di Cliente privato e di Cliente società. In pratica, si è deciso di differenziare i clienti in due categorie (persone fisiche e persone giuridiche) perché il sistema dovrà trattarle in modo differente.

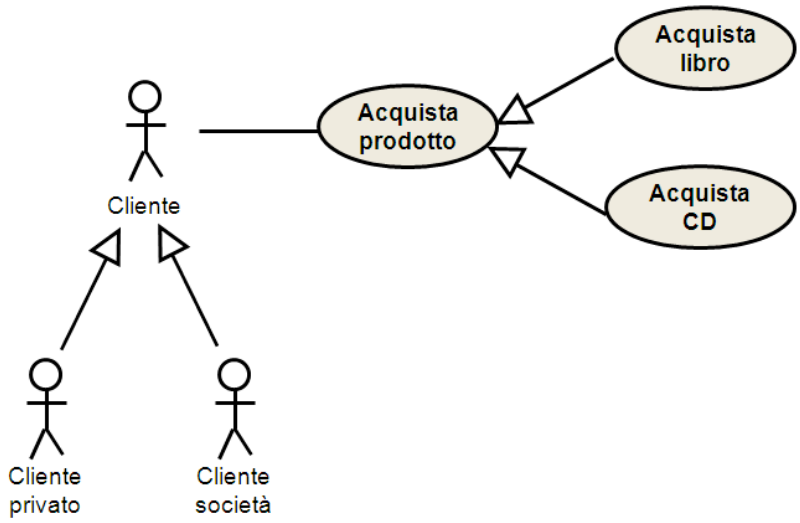


Figura 123. Rappresentazione grafica della generalizzazione fra attori e fra casi d'uso

Quando un caso d'uso ne utilizza un altro, incorporandone il comportamento, si dice che lo *include*. Se un caso d'uso è incluso più volte, conviene dargli un nome e descriverlo separatamente. Questo permette di non duplicarne la descrizione nel documento dei requisiti.

Per rappresentare graficamente l'inclusione, si traccia una freccia tratteggiata dal caso d'uso includente al caso d'uso incluso, con la etichetta <<include>>. Per esempio, in Figura 124 Acquista prodotto e Verifica stato ordini includono entrambi il caso d'uso Autenticazione. Infatti, per effettuare entrambe le operazioni l'utente dovrà fornire le proprie credenziali d'accesso al sistema, ed essere da questo riconosciuto. Autenticazione sarà quindi descritto separatamente e richiamato nella descrizione dei casi d'uso che lo includono, sottolineandone il nome nei passi che lo invocano, come abbiamo visto nell'esempio in Figura 122.

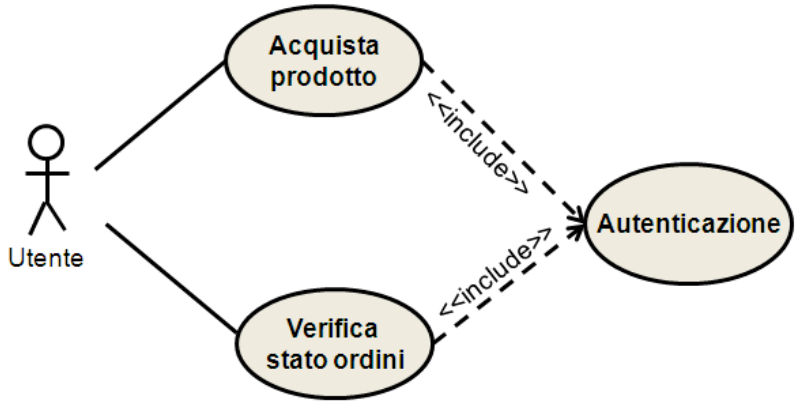


Figura 124. Rappresentazione grafica dell'inclusione fra casi d'uso

Infine, si dice che un caso d'uso *estende* un altro caso d'uso quando il suo comportamento *può* (ma non necessariamente deve) essere richiamato all'interno del primo, come una sua variante. Si noti che il caso d'uso esteso è definito indipendentemente dal caso d'uso che lo estende.

La estensione viene rappresentata graficamente come in Figura 125, dove il caso d'uso Help on line estende i casi d'uso Acquista prodotto e Verifica stato ordini. Questo significa che Help on line può essere richiamato, in qualche scenario d'uso, dagli altri due casi d'uso menzionati.

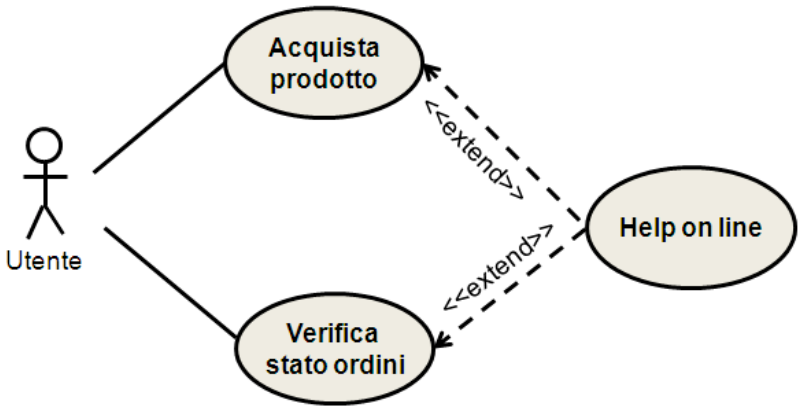


Figura 125. Rappresentazione grafica dell'estensione di casi d'uso

Queste notazioni grafiche possono aiutare a precisare meglio le relazioni fra i casi d'uso all'interno dei diagrammi. È tuttavia consigliabile non farne un uso eccessivo: notazioni troppo formali spaventano e allontanano i lettori non tecnici: è preferibile un documento di requisiti un po' meno preciso, ma letto approfonditamente e condiviso dai vari stakeholder, a un documento perfetto, ma che nessuno ha realmente compreso.

Il documento dei requisiti

Siamo ora in grado, a conclusione di questo capitolo, di individuare una possibile struttura del documento dei requisiti. Esso dovrebbe contenere, prima di ogni requisito specifico relativo al sistema da realizzare, un'approfondita analisi dell'utente e delle sue necessità. In particolare, dovrebbe coprire i seguenti temi:

- Analisi degli utenti: a quali categorie di utenti è destinato il prodotto? Quali sono le loro caratteristiche? Quali categorie vanno considerate prioritariamente?
- Analisi dei bisogni: quali sono le necessità di ciascuna categoria di utenti? Quali sono prioritari?
- Analisi del contesto d'uso: quali saranno i diversi contesti d'uso del prodotto da parte delle diverse categorie di utenti? Quali sono prioritari?

Queste analisi dovrebbero essere corredate dalla descrizione di alcuni scenari d'uso tipici, per far comprendere meglio al lettore lo scopo del prodotto. Gli scenari sono particolarmente importanti per i prodotti di nuova concezione, per i quali non esistono esperienze d'uso consolidate.
Dopo questa prima parte generale, che fornisce le motivazioni del sistema da progettare e lo colloca nel suo contesto, i requisiti dovrebbero proseguire con la descrizione dei diversi casi d'uso del sistema. Si dovrebbe inserire il diagramma dei casi d'uso, con i diversi attori individuati nella precedente analisi degli utenti, e descrivere ogni singolo caso d'uso con le tecniche esemplificate più sopra.

Una possibile struttura del documento dei requisiti è schematizzata in Figura 126.

1.Sommario
2.Generalità
- Scopo del prodotto
- Situazione attuale
- Caratteristiche degli utenti
- Contesto d'uso
- Scenari d'uso
- Fattibilità tecnologica

3. Posizionamento
- Analisi della concorrenza
- Posizionamento competitivo

4. Casi d'uso
- Diagramma dei casi d'uso
- Descrizione dei singoli casi d'uso

5. Altri requisiti
- Requisiti per l'esperienza utente
- Requisiti prestazionali
- ….

Appendici
- Glossario
- …..
- Riferimenti

Figura 126. Una possibile struttura del documento dei requisiti

Le diverse sezioni possono essere redatte sulla base delle indicazioni che seguono.

Generalità

- Scopo del prodotto. Descrive la natura del prodotto oggetto del documento di requisiti. Specifica sinteticamente gli obiettivi del prodotto (quelli generali e quelli più specifici), distinguendo quelli principali da quelli secondari, e stabilendone le priorità.

- Situazione attuale. Specifica se il progetto riguarda un prodotto di nuova realizzazione, o se si tratta di un miglioramento di un prodotto esistente. In questo secondo caso, elenca i principali punti di forza e di debolezza del prodotto attuale.

- Caratteristiche degli utenti. Specifica a quali categorie di utenti è destinato il prodotto. Descrive il profilo di ciascuna categoria: competenze, abilità, esperienze, corsi di addestramento seguiti, caratteristiche psico-fisiche, abitudini, preferenze. Specifica gli obiettivi di ciascuna categoria di utenti in rapporto al prodotto e i diversi ruoli rivestiti nei suoi confronti. In questa sezione s'identificano le classi di attori che verranno utilizzate nella successiva descrizione dei casi d'uso.

- Contesti d'uso. Descrive i diversi contesti e ambienti d'uso per le diverse categorie di utenti descritte in precedenza. Questi possono includere eventuali standard adottati, il contesto normativo (per esempio, leggi e regolamenti), le caratteristiche dell'ambiente tecnologico, l'ambiente organizzativo (per esempio, struttura organizzativa, procedure di lavoro, consuetudini consolidate) e le caratteristiche dell'ambiente fisico, se rilevanti.

- Scenari d'uso. Descrive sinteticamente gli scenari d'uso tipici e significativi, che mettano in luce gli aspetti più importanti del prodotto, collocati nel loro contesto.

- Fattibilità tecnologica. Indica le tecnologie che dovranno essere utilizzate nella realizzazione del prodotto, e che lo rendono fattibile.

Posizionamento

- Analisi della concorrenza. Identifica i principali prodotti concorrenti. Per ciascuno, riporta una scheda descrittiva, con la storia, le caratteristiche rilevanti e i principali motivi di successo e d'insuccesso. Include immagini e riferimenti come opportuno. Questa sezione può essere espressa in forma sintetica, allegando eventuale materiale di dettaglio in appendice.

- Posizionamento competitivo. Specifica quali saranno i punti di forza e gli eventuali punti di debolezza del prodotto in rapporto ai prodotti concorrenti più sopra indicati.

Casi d'uso

- Diagramma dei casi d'uso. Fornisce una visione di sintesi dei casi d'uso del prodotto. Gli attori rappresentati nel diagramma devono corrispondere alle tipologie di utenti descritte nella sezione Generalità.

- Descrizione dei casi d'uso. Descrive ogni caso d'uso in forma verbale, specificando per ciascuno lo scenario principale di successo e gli scenari alternativi.

Altri requisiti

- Requisiti per l'esperienza utente. Descrive i requisiti relativi all'interfaccia utente e, più generale, alla *user experience*. Contiene esempi o figure ove necessario. Specifica come si dovrà valutare, durante il progetto, il soddisfacimento di questi requisiti.
- Requisiti prestazionali. Esprime i requisiti relativi alle prestazioni del sistema, anche relativamente alle risorse necessarie per il suo utilizzo.

- Altri requisiti. Questa sezione, che può anche essere estesa e suddivisa in ulteriori sezioni, contiene ogni altro requisito, in funzione della natura del prodotto o servizio in esame.

Allegati

- Glossario. Definisce gli eventuali termini tecnici o gergali, specifici dell'organizzazione o del prodotto, utilizzati nel documento.
- Altri allegati. Come necessario.
- Riferimenti. Contiene ogni riferimento a pubblicazioni o a materiale di supporto utile per un migliore comprensione del documento.

Ripasso ed esercizi

1. Che cosa s'intende con il termine "requisiti di prodotto"?
2. Che cosa s'intende per stakeholder di un prodotto?
3. Quali sono i principali metodi di raccolta dei requisiti?
4. Che cosa s'intende per "analisi dell'utente" in un processo di progettazione human-centred? Abbozza, come esempio, l'analisi dell'utente per il sito web di una biblioteca universitaria.
5. Che cosa s'intende per "analisi dei bisogni"? Per chiarire meglio il concetto, scrivi anche una sintetica analisi dei bisogni relativi al progetto dell'ascensore di casa tua.
6. Che cosa s'intende per "analisi del contesto"? Spiega il concetto anche con semplici esempi.
7. Che cosa sono e a che cosa servono gli scenari d'uso? Quali sono le caratteristiche di un buon scenario d'uso?
8. Qual è la differenza fra casi e scenari d'uso?
9. Scrivi tre scenari d'uso relativi all'ascensore di casa tua.
10. Costruisci il diagramma dei casi d'uso dell'ascensore di casa tua, e scrivi la descrizione di ogni singolo caso d'uso.
11. Costruisci il diagramma dei casi d'uso di una macchina erogatrice di bibite, considerando i diversi attori coinvolti, e descrivi i diversi casi d'uso con le modalità illustrate in questo capitolo.

Approfondimenti e ricerche

1. Sulle tecniche di raccolta dei requisiti esiste molta letteratura, anche in rete. Si cerchino, per esempio, le parole "requirements elicitation", "requirements analysis", "requirements engineering". Una visione più ampia di quella del presente libro, ma ancora abbastanza sintetica, si può trovare nel classico testo di J.Preece, Y.Rogers e H.Sharp, *Interaction Design* (Second Edition), John Wiley&Sons, 2007.
2. Guarda il classico video della Apple (1987) che mostra uno scenario d'uso del Knowledge Navigator, citato all'inizio di questo capitolo. In rete ne esistono numerose copie, per esempio in www.youtube.com.
3. Anche la letteratura sui casi d'uso è vasta, ma prevalentemente orientata alla progettazione di sistemi a oggetti, e non all'interaction design. Inoltre, differenti autori interpretano il concetto in modi non sempre identici. Per questo, per approfondire la nozione di caso d'uso occorre una certa cautela, altrimenti si rischia di confondere le idee. A chi volesse farlo, si consiglia vivamente di iniziare dall'articolo di Alistair Cockburn, *Use cases, ten years later*, originalmente pubblicato nel 2002 e disponibile in rete, breve ma molto utile. Si potrà poi cercare ulteriore materiale, preferibilmente legato all'interaction design. (Per esempio, in http://www.guuui.com/issues/02_04.php).
4. Per indicazioni su come costruire le personae per gli scenari d'uso, si può vedere il blog di S.Mulder: http://www.practicalpersonas.com/persona_value/ , e la sua presentazione su www.slideshare.com: *The User is Always Right – Making Personas Work for Your Site*, in http://www.slideshare.net/MulderMedia/the-user-is-always-right-making-personas-work-for-your-site , con interessanti esempi.

8.Ingegneria e creatività

Sintesi del capitolo

Questo capitolo considera gli aspetti creativi del processo di progettazione, esaminando in particolare il passaggio dai requisiti all'invenzione del design concept iniziale. S'introducono, con esempi, i procedimenti di mimesi, ibridazione, metafora, variazione e composizione, che possono essere utilizzati dai progettisti, discutendone vantaggi e svantaggi. Si accenna alla pratica di raccogliere collezioni ordinate di design pattern, che consolidano e documentano lo stato della pratica della progettazione nelle varie tipologie di prodotti interattivi (per esempio, siti web).

Dai requisiti al design concept

Supponiamo che sia stato realizzato il documento dei requisiti di un prodotto, organizzato secondo le indicazioni del capitolo precedente. Come possiamo, partendo da questo, *concepire* il prodotto? In altre parole, quali sono i processi che ci permettono di passare dalla descrizione di un insieme di bisogni e di vincoli, all'*invenzione* del sistema che tali bisogni e vincoli soddisfi nel migliore dei modi? La progettazione è arte complessa. Il lavoro del progettista non si esaurisce nell'applicazione di metodi e best practice suggeriti dall'ingegneria dell'usabilità, come le linee guida che saranno descritte nei prossimi capitoli. Queste sono importantissime, ma non bastano. A partire da uno stesso documento di requisiti, due progettisti diversi concepiranno sistemi diversi, a volte *molto* diversi. La progettazione non è un algoritmo che, dati gli stessi input, produce sempre gli stessi output. È, in misura rilevante, attività *creativa*. Costruire il "ponte" fra ciò che esiste e ciò che vogliamo che esista richiede non soltanto un'accurata conoscenza dell'utente e dei suoi bisogni o desideri (spesso inconsapevoli o latenti), e dei vincoli posti dal contesto d'uso, ma anche visione e ispirazione e, a volte, un po' di fortuna: nei prodotti del genio, l'aiuto del caso può essere determinante[101].

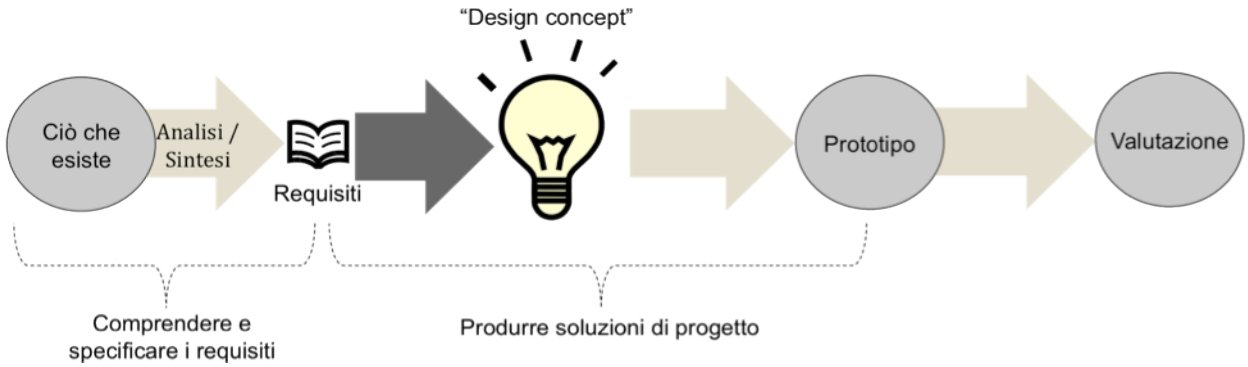


Figura 127. Dai requisiti al design concept

Il passaggio critico è rappresentato dalla freccia scura in Figura 127. Il documento dei requisiti esprime i bisogni e i vincoli emersi nella fase di esplorazione, bisogni e vincoli che possono essere soddisfatti in molti modi. Che cosa determina il passaggio da questi al *design concept*, all'idea di una loro specifica e concreta implementazione? Che cosa fa scattare la scintilla nella mente del progettista?

In sostanza, i processi che portano alla definizione di un documento di analisi dei requisiti sono processi *di analisi e di sintesi*: si tratta di ricercare, scoprire, raccogliere, selezionare e organizzare in forma coerente ciò che è nella mente

[101] Gli esempi in cui il caso è stato determinante nella individuazione di soluzioni di design di grande successo sono numerosi. L'invenzione dei Post-it, foglietti di carta colorata semi-adesivi fu, per esempio, la conseguenza imprevista del tentativo fallito di creare un adesivo molto potente. Solo in seguito si pensò di utilizzare il blando adesivo risultante per costruire dei segnalibri o dei blocchetti per note estemporanee.

degli stakeholder del prodotto che dovrà essere progettato, ovvero di estrarre le conclusioni implicite nel materiale esistente, per esempio dalle interviste effettuate e dai dati di mercato. Passare dal documento dei requisiti al design concept incarnato nei primi prototipi di un nuovo sistema interattivo coinvolge invece processi molto diversi, ciò che solitamente chiamiamo *invenzione*. La creatività, nel processo di design, è concentrata soprattutto qui. Una volta creato il design concept, si tratterà di lavorarci sopra, rifinendone le caratteristiche, perfezionandolo e producendone delle rappresentazioni adatte a guidare la successiva realizzazione, con tecniche opportune, a seconda del tipo di prodotto: un edificio, un sistema software, un mobile… Ma questi passaggi, ancora una volta, coinvolgono sostanzialmente le capacità di analisi e di sintesi del progettista, e in misura molto minore la sua creatività. Come disse del suo lavoro il grande progettista Thomas Edison quasi un secolo fa, "il genio è per l'1% ispirazione e per il 99% traspirazione".

In sostanza, ciò che si chiede al bravo progettista è un mix di abilità molto diverse: capacità di *inventare* nuove soluzioni, di *rappresentarle* utilizzando notazioni rigorose e di *valutarne* criticamente la validità. Innanzitutto, dovrà essere in grado di concepire soluzioni innovative ai problemi posti nei requisiti. Poi dovrà essere in grado di rappresentarle, utilizzando opportuni strumenti descrittivi (schizzi, diagrammi o, più in generale, prototipi). Infine dovrà essere in grado di sottoporre continuamente a valutazioni critiche e a prove d'uso le soluzioni ipotizzate, per individuarne e correggere gli eventuali punti deboli. Si comprende come questo processo sia difficilmente formalizzabile in un metodo riproducibile nelle varie situazioni, che possono essere molto diverse. Si possono però individuare alcuni procedimenti tipici del lavoro dei progettisti, che è bene conoscere. Una sintetica trattazione di questi procedimenti è lo scopo di questo capitolo.

I processi dell'invenzione

L'analisi dei processi cognitivi coinvolti nel lavoro creativo è tema molto complesso, che esula dagli scopi di questo libro. Ci limiteremo quindi, restando sulla superficie del problema, a descrivere alcuni approcci possibili a disposizione di ogni progettista nella pratica quotidiana del design e, in particolare, del design dell'interazione.

Il progettista di un nuovo manufatto, a fronte di un documento di requisiti che chiarisca lo scopo e i vincoli della soluzione richiesta, ha di fronte a sé vari percorsi, molto differenti per i risultati che producono, e per il grado di originalità e d'innovazione.

- La prima possibilità, che chiameremo *mimesi,* è quella di riprodurre, con tecnologie diverse, un prodotto già esistente, che risolve il problema. È la via più semplice, quella che non richiede al progettista alcuno sforzo creativo.
- Oppure, il progettista potrà procedere per *ibridazione*, considerando due o più prodotti esistenti, e in qualche modo fondendone le caratteristiche funzionali per creare un prodotto del tutto nuovo.
- Il procedimento più complesso, e probabilmente quello in grado di produrre i risultati più interessanti, è quello che fa uso della *metafora*. Consiste nel trasferire nell'ambito del nostro progetto soluzioni adottate in differenti domini applicativi. È un procedimento ben noto nel design dei sistemi interattivi, e che ha prodotto risultati molto importanti, come per esempio, il desktop che ha rivoluzionato, a partire dallo Star della Xerox e dal Macintosh della Apple, l'interfaccia dei personal computer.
- Un'altra possibilità è, più semplicemente, quella della *variazione*. Si tratta di progettare il nuovo sistema prendendo le mosse da un modello noto, introducendo delle *varianti* migliorative. Il modello potrà essere, secondo i casi, un prodotto concorrente oppure la versione corrente di un prodotto che si desidera migliorare. Quest'approccio può essere chiamato *evolutivo*.
- Ancora, si può considerare una collezione di prodotti esistenti, ed estrarre da ciascuno una o più caratteristiche (o, come si usa dire, *pattern*) utili nel progetto corrente. Il nuovo prodotto sarà così una sorta di *collage* di soluzioni già note e sperimentate, ma inserite in un contesto nuovo ed eventualmente reinterpretate. Chiameremo questa tecnica *composizione*. Si noti che non si tratta di un banale riuso di componenti tecnologici, ma di adottare soluzioni di design già sperimentate, che vengono fra loro armonizzate e inserite nel nuovo progetto. Chiaramente, quest'approccio può condurre a risultati più innovativi del precedente: da un *remix* intelligente di soluzioni note possono emergere prodotti sostanzialmente originali.

Come ultima possibilità, consideriamo la *creazione* pura. In questo caso, il concept del prodotto in corso di

progettazione è del tutto nuovo, e non prende alcuno spunto da prodotti esistenti. La citiamo solo come alternativa teorica, lasciando decidere agli studiosi dei processi cognitivi se questa, in pratica, possa mai verificarsi. Ne dubitiamo: nella pratica della progettazione sembra sostanzialmente impossibile individuare un concept che non sia in qualche modo associabile a prodotti o servizi preesistenti. Come nella biologia, nel vasto ecosistema degli oggetti interattivi non si trovano specie che non siano in qualche modo geneticamente correlate ad altre specie.

Tutti questi procedimenti possono essere compresenti, in varia misura, nei processi concreti di design. Esaminiamo ora più in dettaglio, anche con l'aiuto di esempi, tutte queste tecniche.

Mimesi

Mimesi, dal greco, significa semplicemente *imitazione.* Si tratta, in sostanza, di riprodurre un prodotto già esistente, tipicamente realizzandolo con tecnologie differenti (Figura 128).



Figura 128. Progettazione per mimesi

Questo procedimento non ha necessariamente una connotazione negativa, non è detto che si tratti di un plagio. Una tecnica frequente consiste nel progettare oggetti virtuali (cioè realizzati via software), che riproducono in ogni dettaglio oggetti reali ben noti. Nell'era dell'informatica molti oggetti vivono, per così dire, due vite: una vita fisica, nel mondo reale, e una vita sullo schermo del computer. Per esempio, la Figura 129 mostra la versione virtuale di un famoso modello di calcolatore scientifico della Hewlett Packard, diffusissimo fra gli ingegneri, che ne riproduceva il funzionamento, in tutti i dettagli, su uno dei primi computer Macintosh della Apple. Il fatto che si trattasse di una riproduzione fedele dell'originale aggiungeva valore al prodotto software, e per questo il logo dell'HP era messo bene in evidenza in alto a destra.

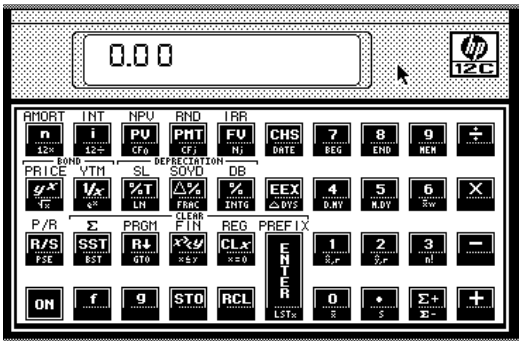


Figura 129. Il calcolatore scientifico della HP, nella sua versione virtuale per Macintosh (circa 1985)

Più recentemente, con un procedimento di mimesi l'iPhone si trasforma in un microfono di registrazione e in una bussola (Figura 130).



Figura 130. Il registratore e la bussola nell'iPhone Apple (2009)

Il procedimento della mimesi funziona bene quando le azioni che l'utente compie sull'oggetto reale hanno un corrispettivo "naturale" sulla sua rappresentazione virtuale. Per esempio, nel libro elettronico di Figura 131, l'azione di sfogliare una pagina è effettuata in modo molto naturale facendo "strisciare" da destra verso sinistra il cursore del mouse sull'angolo della pagina, così come si farebbe col dito su un libro vero.

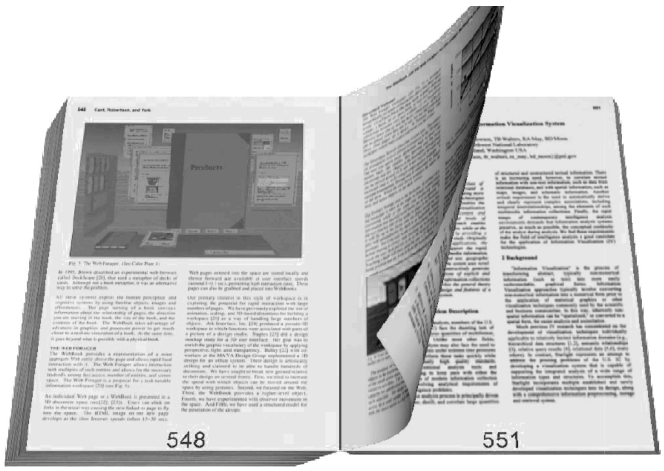


Figura 131. Un libro elettronico[102]

Non sempre questa condizione si verifica, però, ed esistono oggetti reali che mal si prestano a una loro rappresentazione virtuale. Si consideri, per esempio, il telefono virtuale rappresentato in Figura 132. L'azione di comporre il numero può essere effettuata molto naturalmente, cliccando i pulsanti della tastiera. Nel telefono vero, l'utente dovrebbe poi sollevare la cornetta. Ma come far corrispondere questa azione sul telefono virtuale? Nel prodotto indicato l'utente doveva cliccare sulla rappresentazione della cornetta. Ma questa soluzione appare, in verità, molto forzata: sollevare e cliccare sono due azioni molto diverse. E poi la cornetta virtuale, una volta sollevata, dove dovrebbe andare a collocarsi?

[102] Da L.Hong, E.N.Chi, S.K.Card, *Annotating 3D electronic books*, Proceedings CHI '95.

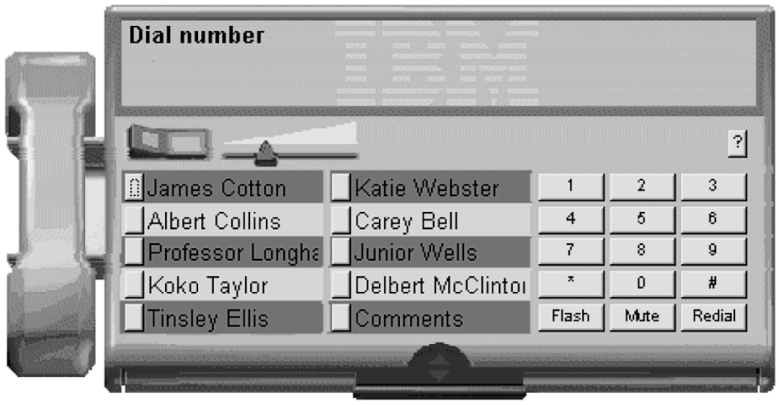


Figura 132. Un telefono virtuale (IBM Real Phone, metà anni '90)

Un oggetto reale riprodotto virtualmente può evolvere, acquisendo funzionalità che sono realizzabili soltanto nella sua versione software. Per esempio, la Figura 133 mostra un "righello elettronico" da utilizzare come accessorio di programmi di grafica. Esso può essere allungato o accorciato secondo le necessità (con un'operazione di *drag* sulla freccia posta a destra), e può cambiare l'unità di misura (con il pulsante Scale). È possibile ruotarlo di 90° cliccando sul pulsante Flip. Un righello reale non ha, evidentemente, tali funzionalità.

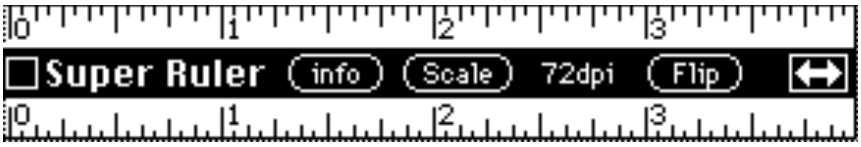


Figura 133. Un righello virtuale (per Macintosh, circa 1985)

Ibridazione

L'*ibridazione*, spiega il dizionario, consiste nell'incrociare piante o animali di specie diverse in modo da ottenere ibridi. Nel nostro caso, si tratterà di concepire un oggetto nuovo mescolando e integrando fra loro aspetti e funzioni di più oggetti diversi (Figura 134).

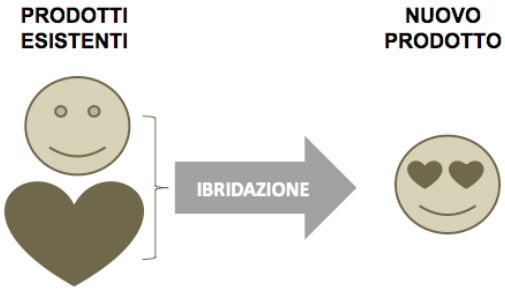


Figura 134. Progettazione per ibridazione

Così, a partire da un proiettore e da una lavagna tradizionale, s'inventa la lavagna luminosa, che fonde le due funzioni in un prodotto del tutto diverso. Gli esempi di ibridi nell'interaction design sono numerosi. La Figura 135 mostra un mouse wireless che fornisce, sul retro, un insieme di comandi per controllare una presentazione PowerPoint: puntatore laser, slide avanti/indietro, controllo volume audio.



Figura 135. Wireless Notebook Presenter Mouse 8000, di Microsoft (2006)

La Figura 136 mostra un oggetto software di Windows, costruito a partire da un orologio, un calendario, una dialogue box e una struttura a schede selezionabili attraverso linguette (*tab*).

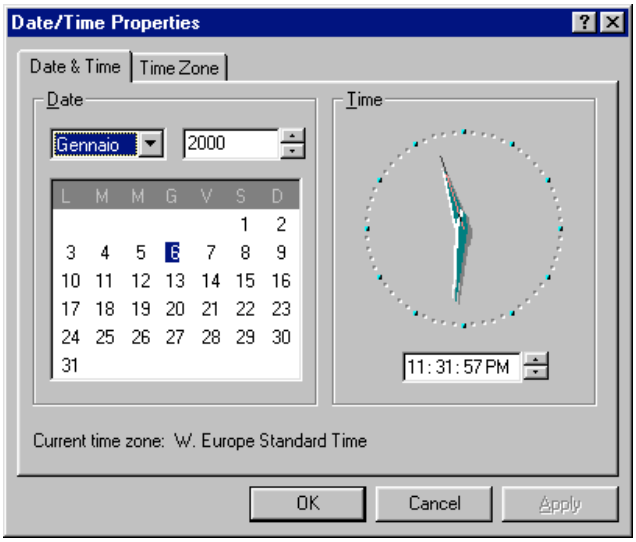


Figura 136. Orologio / calendario (Microsoft Windows, anni '90)

A volte il procedimento d'ibridazione può portare a risultati sorprendenti, come per esempio nel caso dell'I/O Brush realizzato al Media Lab del MIT, una sorta di incrocio fra un pennello e una telecamera (Figura 137).[103] Quando si passa il pennello (a sinistra in figura) su un oggetto qualsiasi, come il piatto di caramelle colorate (a destra), la telecamera incorporata nel pennello ne cattura l'immagine a colori, che è poi riprodotta passando il pennello su un touch

[103] Ryokai, K., Marti, S., Ishii, H., *I/O Brush: Drawing with Everyday Objects as Ink*. In Proceedings of Conference on Human Factors in Computing Systems (CHI 2004). Si veda anche http://web.media.mit.edu/~kimiko/iobrush/.

screen collegato (a destra, in basso). In sostanza, l'immagine ripresa dalla telecamera è la "vernice" con la quale si dipingerà poi sullo schermo. L'immagine ripresa può essere statica o in movimento: i "dipinti" realizzati col pennello ne riprodurranno la complessità e le animazioni.

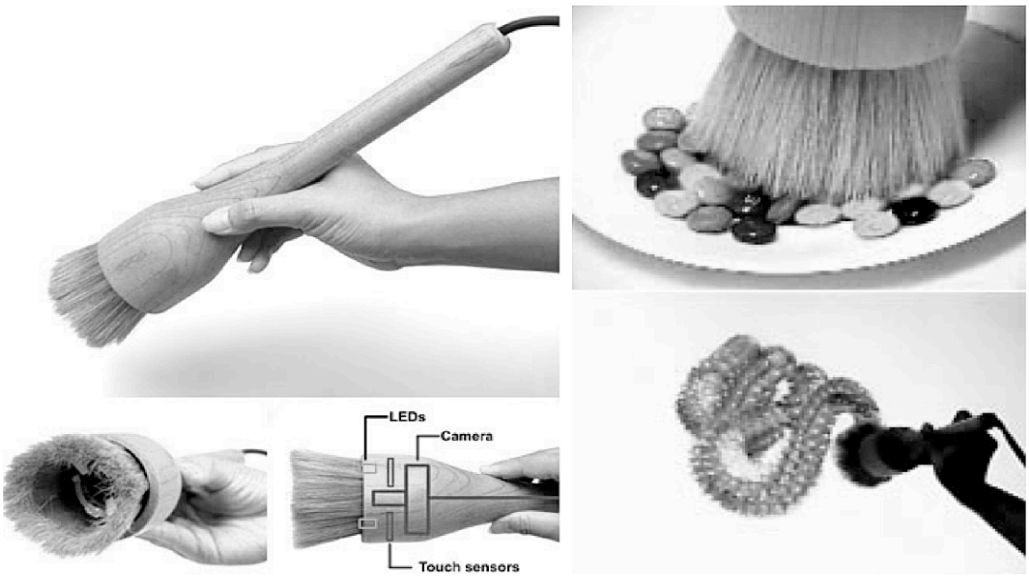


Figura 137. L'I/ O Brush del MIT Media Lab.

Che cosa si può ottenere incrociando una chitarra e un iPhone? Un risultato possibile è l'applicazione software PocketGuitar, mostrata in Figura 138.



Figura 138. L'applicazione PocketGuitar per iPhone (2009)

La tecnologia del Web permette di creare facilmente nuovi servizi online, per ibridazione (o, come si dice più precisamente in questo caso, *mashup*) a partire da servizi esistenti. La Figura 139 ne mostra un tipico esempio: un servizio online (http://www.housingmaps.com) che visualizza, su una mappa realizzata da Google Maps, le proprietà immobiliari in vendita nella località prescelta (in figura, Miami), i cui annunci sono pubblicati, ma soltanto in forma testuale, sul sito http:www.craigslist.com. Ogni pallino sulla mappa rappresenta una proprietà: cliccando sul pallino ne appare una sintetica descrizione, col link all'annuncio originale su craiglist (in figura, nella finestra sulla destra).

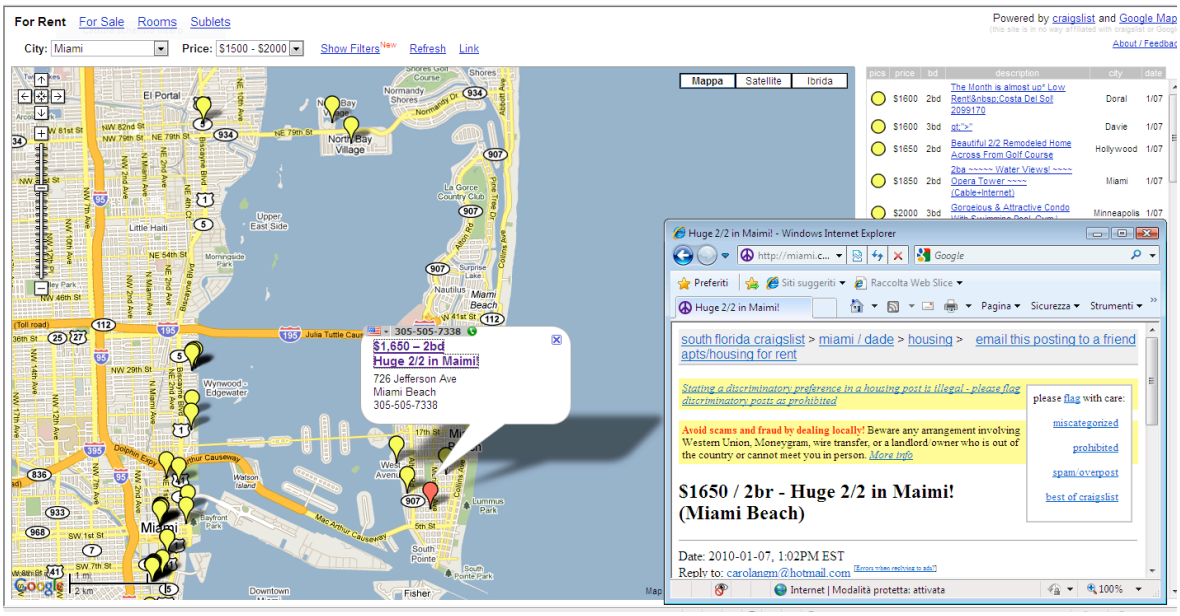


Figura 139. http://www.housingmaps.com (2009)

In questo esempio, nonostante la sua semplicità realizzativa, il prodotto ibrido fornisce un alto valore aggiunto ai dati provenienti dalle applicazioni di origine che, se non messi fra loro in relazione, sono poco utilizzabili. Così, la mappa di Miami e la semplice lista testuale delle proprietà in vendita in questa località, separatamente sono poco utili. È dal loro incrocio che l'utente può localizzare a colpo d'occhio le proposte di suo interesse.

La Figura 140 mostra un ultimo interessante esempio di ibrido. Si tratta di un'applicazione software per Mac, che fornisce una console virtuale da dj, per controllare l'esecuzione di brani musicali selezionati da iTunes. L'interfaccia riproduce fedelmente una console reale, e ne permette di realizzare i diversi effetti (*scratching*, *fading*, ecc.). In questo caso il progettista ha utilizzato entrambi i procedimenti che abbiamo analizzato: la mimesi per riprodurre la console e l'ibridazione con l'applicazione iTunes (visibile sulla destra dello schermo), che fornisce i brani musicali e le funzioni di gestione delle playlist.



Figura 140. L'applicazione djay 3 per Mac (2009)

Metafora

Il termine metafora denota una figura della retorica classica, e deriva dal greco *metaforà*, con significato di "trasporto", o "trasferimento". L'essenza della metafora è *descrivere una cosa nei termini di un'altra*. In essa, due domini semantici indipendenti vengono messi in contatto: questo fa sì che uno dei due domini venga compreso facendo riferimento all'altro. Per esempio, quando Shakespeare scrive, in *As You Like It* (2, VII):

> *È vero, il mondo è tutto un palcoscenico*
> *sul quale tutti noi, uomini e donne*
> *siam solo attori, con le nostre uscite*
> *e con le nostre entrate; ove ciascuno,*
> *per il tempo che gli è stato assegnato,*
> *recita molte parti,*
> *e gli atti sono le sue sette età*
> *....*

intende trasferire le proprietà di un "palcoscenico" (con tutto il campo semantico ad esso associato: "attori" , "uscite", "entrate", "recitare", "parti", "atti", …) alla nozione di "mondo". Così possiamo parlare del mondo utilizzando tutte le proprietà e i concetti associati al palcoscenico. La metafora consiste, in sostanza, nel mescolare fra loro campi semantici differenti, trasferendo proprietà e concetti propri di un campo semantico (il *donatore,* nel nostro caso il palcoscenico) a un altro (il *ricevente,* il mondo).

La metafora non va confusa con la *similitudine*. Quest'ultima asserisce la *somiglianza* di due campi semantici, e non la loro *identità*. Se avesse voluto fare una similitudine, Shakespeare avrebbe scritto: "il mondo *assomiglia* a un palcoscenico", e non "*è* un palcoscenico". Nella similitudine, i due concetti restano distinti, nonostante la loro somiglianza. Nella metafora, invece, i due concetti vengono identificati, nonostante la loro differenza: i due campi semantici vengono, per così dire, sovrapposti.

Nell'ambito della progettazione, il campo semantico del donatore viene trasferito all'oggetto della progettazione, arricchendolo di una nuova interpretazione (Figura 141).

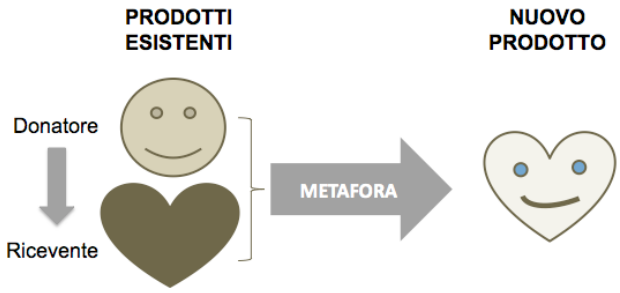


Figura 141. Progettazione per metafora

La metafora è una fonte importante di idee innovative: una volta creata l'associazione, possiamo esplorarne le conseguenze, esaminando il campo donatore per estrarne i suggerimenti. Per esempio, le due metafore:

La gamba del tavolo Il ruggire del motore

potrebbero suggerire il design di un tavolo con le giarrettiere, e lo slogan "metti un tigre nel motore", come, in effetti, è avvenuto in entrambi i casi.

Il procedimento metaforico è stato utilizzato molto spesso nell'interaction design: basti pensare alle nozioni di menu, di finestra, di desktop, di bottone, comunemente utilizzati nell'interfaccia dei personal computer. In tutti questi casi, e in molti altri ancora, dal trasferimento di concetti noti e propri di un certo dominio a domini applicativi del tutto diversi,

sono nati meccanismi nuovi, ora entrati nell'uso comune. La potenza dell'associazione metaforica può essere illustrata dal semplice esempio di Figura 142, tratto da Microsoft Word 95. Il menu muto, composto solamente da strisce colorate (il ricevente della metafora) sarebbe incomprensibile senza l'icona dell'evidenziatore (il donatore), che lo spiega senza bisogno di parole e ne chiarisce l'uso. Basta la semplice presenza di questa icona a suggerire in modo inequivocabile lo scopo del menu: l'intero campo semantico associato all'evidenziatore fisico – un oggetto entrato da tempo nell'uso quotidiano e quindi ben noto agli utenti – viene trasferito al menu. Come se non bastasse, selezionando un certo colore, il puntatore del mouse cambia forma, e assume quella dell'evidenziatore.



Figura 142. L'evidenziatore di Microsoft Word 95

La metafora si rivela uno strumento molto potente per l'interaction designer. Il campo semantico portato dalla metafora gli può suggerire concetti, soluzioni e modalità operative che possono rivelarsi molto fecondi. Si pensi alla metafora del desktop, che consiste, inizialmente, in questa semplice associazione:

lo schermo del computer "è" la scrivania dell'utente.

Quest'affermazione trasporta nel mondo dei computer l'intero mondo di concetti, oggetti, attributi, modalità operative associati all'idea di scrivania: la scrivania ha un piano (nella metafora sarà lo schermo del computer), su cui si pongono documenti, cartellette, strumenti come l'orologio, il calendario, e così via. Come su una scrivania reale posso spostare e sovrapporre documenti, così lo potrò fare sullo schermo del computer, con l'aiuto del mouse. Se voglio riporre un documento in una cartelletta, basterà spostarcelo sopra con il mouse. Esplorando le possibilità suggerite dalla metafora, il progettista ricava idee e orientamenti per il design.

Alcuni ritengono che l'uso della metafora possa anche essere utile per spiegare all'utente l'utilizzo di un sistema che non conosce ancora. Per esempio, per spiegare a chi non abbia mai visto un desktop la logica del suo funzionamento, gli si dice: "il desktop del Macintosh *è come* il piano della tua scrivania". Ma non è vero, e l'utente se ne accorge ben presto: le incongruenze sono numerose, ed egli si sente ingannato. Come possono stare le finestre sul piano di una scrivania? E il cestino della carta straccia, non dovrebbero essere per terra, accanto alla scrivania, e non sopra? E che corrispettivo hanno i menu in un ufficio reale? E perché quando "apro" un dischetto sulla scrivania ne rimane l'ombra, come nel desktop del primo Macintosh (Figura 143)? E così via. In sostanza, *si usa la metafora come similitudine*. Ma abbiamo già osservato che metafora e similitudine sono due procedimenti diversi. In ultima analisi, dire che il desktop sul computer funziona come una scrivania reale permetterà forse un marketing efficace, ma non fornirà un aiuto significativo all'utente principiante. Per conseguire una buona usabilità, il desktop virtuale dovrà allontanarsi dalla scrivania reale e, per così dire, vivere di vita propria. Il design avrà raggiunto il suo scopo se l'utente, durante l'uso, non dovrà ricorrere ad analogie con la scrivania reale, quanto all'intrinseca naturalezza e coerenza che il progettista sarà stato capace di infondere all'interfaccia. In una metafora riuscita, i due campi semantici si fondono, generando una realtà nuova.

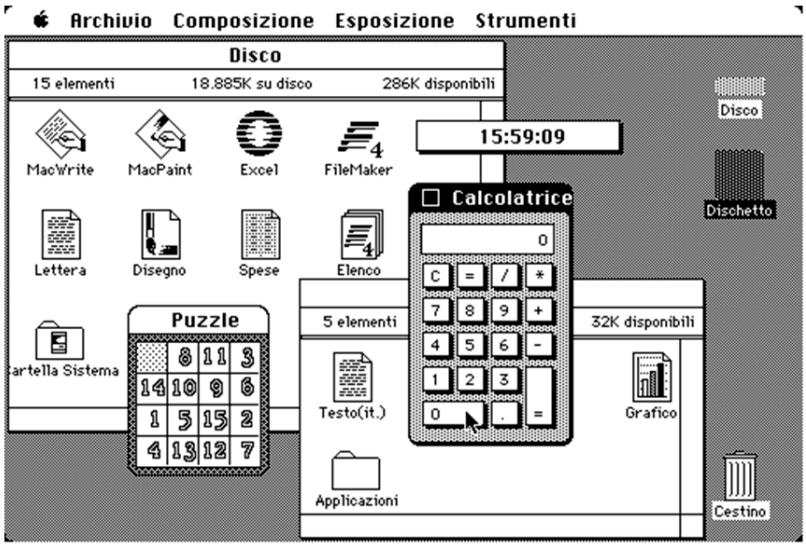


Figura 143. Il desktop del primo Apple Mackintosh (1984)

La Figura 144 mostra un altro esempio di metafora. La home page del sito web dell'aeroporto di Melbourne, molti anni fa, rappresentava uno sportello d'informazioni, con un'impiegata sorridente in attesa dei clienti. L'immagine era certamente gradevole, e la metafora suggeriva lo scopo del sito: fornire informazioni sui servizi dell'aeroporto. Questo si comprende chiaramente dai pieghevoli informativi su ristoranti, alberghi, negozi, ecc, disposti in ordine sul banco. Si dovevano cliccare per aprirli e leggerne il contenuto. Anche i quadri appesi al muro alle spalle dell'impiegata erano cliccabili, per fornire servizi più complessi, per esempio per mostrare la situazione delle partenze e degli arrivi, come nei tabelloni elettronici comunemente presenti negli aeroporti.[104]



Figura 144. Home page del sito dell'aeroporto di Melbourne (fine anni '90)

[104] Nel caso dei servizi interattivi (riquadro Services e Passenger Information), la metafora mostra qualche limite. Il campo semantico associato ai quadri sul non sembra di grande aiuto per di queste funzioni.

La Figura 145 mostra la home page della versione italiana del sito di J.K.Rowling, l'autrice dei romanzi di Harry Potter. In questo caso, la metafora della scrivania (il sito "è" la scrivania dell'autrice) non è utilizzata per facilitare all'utente la navigazione nel sito, quanto per presentargli un ambiente interessante, tutto da esplorare. Molti degli oggetti visualizzati sono cliccabili: indicandoli con il puntatore del mouse, essi si animano, e compare una breve spiegazione del loro significato. In alcuni casi questo è evidente dalla natura dell'oggetto: l'agenda 1965-2010 porta alla pagina Biografia dell'autrice, la rivista Rumours a quella delle Voci, il Daily News alla pagina Notizie, come indicato dal fumetto visualizzato in figura. In altri casi l'associazione è meno immediata, per rendere l'esplorazione più interessante. Per esempio, gli occhiali portano alla pagina Collegamenti ad altri siti, la spazzola a una pagina denominata Extra, i fermagli alle Domande frequenti. La farfalla è cliccabile, ma non porta da nessuna parte: al clic, semplicemente, vola via.



Figura 145. Home page del sito di J.K.Rowling (http://www.jkrowling.com/it, 2009)

Il procedimento metaforico è stato spesso utilizzato anche per scegliere le icone poste sui bottoni. In questo caso, il problema è di utilizzare un'immagine di piccole dimensioni, ma identificabile con chiarezza, che richiami "immediatamente" la funzione del pulsante cui è associata. Questo non è sempre di facile soluzione, soprattutto quando non ci sia lo spazio per collocare vicino all'immagine un'etichetta esplicativa (in quest'ultimo caso, l'icona assume spesso soltanto una funzione decorativa). Ci sono molti esempi di buone metafore (per esempio, le ormai classiche icone di Windows, Figura 146A) e di metafore che funzionano male, come in Figura 146B. In quest'ultimo caso, tratto da un programma degli anni '90, la funzione di molti bottoni non è identificabile a partire dalla sola icona. Infatti, che scopo potranno avere, per esempio, i pulsanti con l'icona del semaforo o con la nuvola che oscura il sole?



Figura 146. Esempi di icone

Con la diffusione delle interfacce basate su icone (per esempio nei piccoli schermi degli apparati mobili), oggi disponiamo di un'ampia collezione d'immagini la cui origine metaforica è ormai dimenticata. Esse possono essere considerate, a tutti gli effetti, elementi di un alfabeto di simboli largamente noti e condivisi, come i segnali stradali o le indicazioni simboliche nelle metropolitane o negli aeroporti. Non abbiamo bisogno di ricorrere ad alcuna metafora per interpretare molte[105] delle icone standard di un iPhone (Figura 147).



Figura 147. Le icone standard dell'iPhone (2009)

Variazione

La variazione (Figura 148) è uno dei procedimenti più frequenti nella progettazione. Una quota rilevante del lavoro del designer consiste nel progettare variazioni, in qualche senso migliorative, di sistemi esistenti. Queste potranno generare prodotti concorrenti di quelli originali, o nuove versioni evolutive degli stessi. Il progetto delle variazioni di un prodotto, a prima vista poco impegnativo dal punto di vista creativo, può produrre innovazioni sostanziali, soprattutto se si considera l'evoluzione nell'arco di più generazioni successive. L'accumularsi di modifiche anche di lieve entità può infatti portare a prodotti completamente differenti da quello di partenza.



Figura 148. Progettazione per variazione

La Figura 149 mostra le due prime versioni del programma di grafica MacPaint per il computer Apple Macintosh (progettate, rispettivamente, da Bill Atkinson nel 1983 e da David Ramsey nel 1987). Si noti il sostanziale

[105] Ma non tutte le icone dell'iPhone sono comprensibili senza la didascalia. Per esempio, perché il girasole dovrebbe rappresentare l'album fotografico?

miglioramento dei menu degli strumenti e dei pattern grafici, che nella versione 2 sono stati trasformati in due *palette* liberamente spostabili sul video, liberando spazio per la visualizzazione del foglio su cui disegnare.

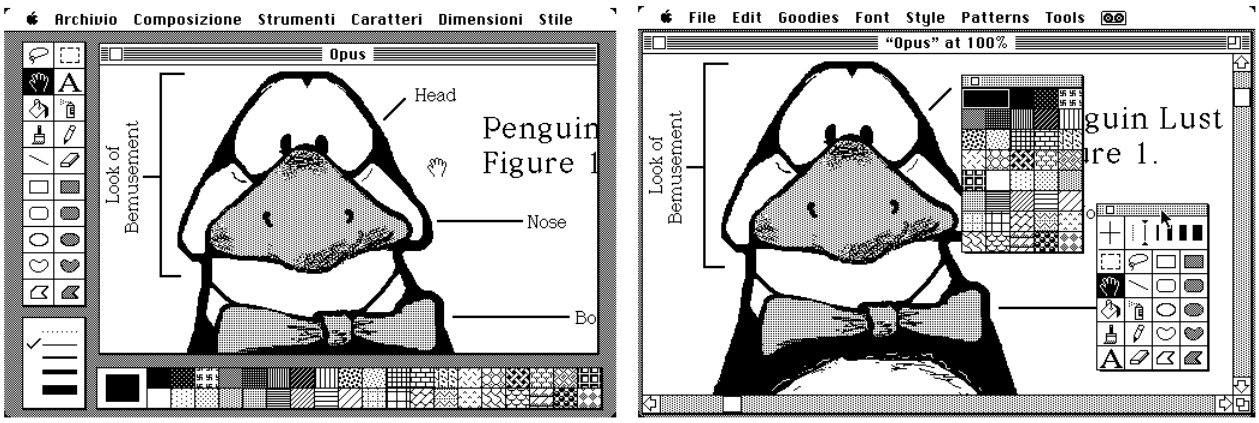


Figura 149. A sinistra: MacPaint versione 1 (Apple, 1983). A destra: MacPaint versione 2 (Claris, 1987)

La Figura 150 mostra, invece, due programmi concorrenti della versione 1 di MacPaint, prodotti intorno al 1985 da produttori diversi. In entrambi si riconosce chiaramente la fonte ispiratrice del design, che ha introdotto varianti di lieve entità, soprattutto nella scelta degli strumenti di disegno disponibili nei menu.

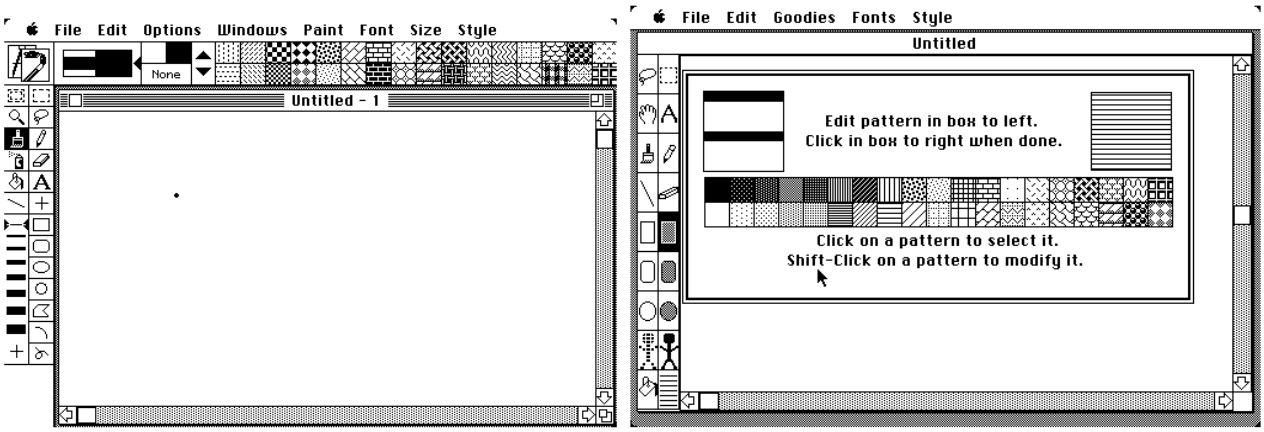


Figura 150. Due programmi concorrenti di MacPaint versione 1 (circa 1985)

I prodotti software (e di conseguenza i prodotti che contengono al loro interno una componente importante di software) sono i manufatti evolutivi per eccellenza. Le migliorie suggerite dall'esperienza d'uso del prodotto, le nuove esigenze segnalate dagli utenti, la necessità di correggere gli errori di programmazione (sempre presenti in qualsiasi sistema software di qualche complessità), mantenendo la compatibilità con il complesso ecosistema di prodotti correlati, fanno sì che i prodotti software vengano continuamente modificati.

D'altro canto, modificare un prodotto software non richiede modifiche a impianti di produzione, e le modifiche possono essere distribuite, attraverso la rete, a costi sostanzialmente nulli. Un tempo, quando non esisteva una rete globale attraverso la quale distribuire all'utenza le nuove versioni del software, questo evolveva per release discrete, e tendenzialmente abbastanza lontane una dall'altra. Il processo di distribuzione delle nuove versioni (che avveniva attraverso la spedizione fisica dei supporti magnetici, coinvolgendo spesso delle organizzazioni locali di distribuzione) era lento e costoso; era quindi conveniente non creare nuove release troppo di frequente. Oggi la situazione è completamente cambiata, ed è il prodotto software stesso, sempre connesso in rete ai server del produttore, a segnalare

ai suoi utenti la disponibilità di nuovi aggiornamenti, e a chieder loro il permesso di auto-aggiornarsi. L'aggiornamento, una volta autorizzato, richiederà solo qualche minuto, e il restart del sistema. Con le connessioni a larga banda *always-on*, il software è diventato, per così dire, un prodotto *fluido*, che si trasforma continuamente durante l'uso, per tutto il suo arco di vita. Questa caratteristica è ancora più spinta nel caso dei prodotti software che non sono installati sulle macchine dell'utente, ma che vengono gestiti direttamente dal produttore per erogare un servizio in rete, via Internet (*SaaS, Software as a Service*). In questo caso il processo di aggiornamento è potenzialmente continuo e invisibile all'utente, che utilizzerà ogni volta la versione più aggiornata del sistema, senza avere alcuna necessità di conoscere caratteristiche e frequenza degli aggiornamenti.

Tutto questo accelera ulteriormente il ritmo di evoluzione dei prodotti a elevato contenuto di software. Questi spesso sono immessi sul mercato, come si dice, in versione β, cioè quando sono ancora funzionalmente immaturi e instabili, perché non completamente collaudati. In sostanza, i produttori utilizzano gli utenti per la messa a punto dei prodotti, chiedendo a essi di segnalare sia i malfunzionamenti dovuti a errori del software, sia problemi di usabilità o suggerimenti migliorativi.

Questa tecnica è senz'altro molto sensata: gli utenti in rete sono enormemente numerosi, e coinvolgerli nel ciclo di progettazione può produrre risultati assai efficaci nel miglioramento dei prodotti. D'altra parte, queste "prestazioni" degli utenti sono spesso ricambiate con la possibilità d'uso dei prodotti a costi molto bassi, quando non sono gratuiti. Ma la spinta al cambiamento che si produce in questo modo è così accelerata, che molti prodotti, sostanzialmente, non escono mai da una versione β. Per questo fenomeno si usa il termine di *perpetual β*, concetto sviluppatosi con il software open source, e adottato più recentemente dalle applicazioni web di nuova generazione, che forniscono servizi in rete. Ciò ha portato Tim O'Reilly, attento osservatore dei fenomeni della rete, a sostenere che siamo arrivati alla fine del "software release cycle":

> *Gli utenti devono essere trattati come co-sviluppatori, seguendo le stesse procedure di sviluppo dei prodotti open-source (anche se è improbabile che il software in questione venga rilasciato con una licenza open-source). Il motto dell'open-source ("rilascia presto e rilascia spesso") si è evoluto in una posizione ancora più radicale, "la beta perpetua", dove il prodotto è sviluppato all'aperto con nuove caratteristiche inserite a cadenza mensile, settimanale o addirittura giornaliera. Non è un caso che servizi come Gmail, Google Maps, Flickr, del.icio.us e altri simili potrebbero continuare a portare la dicitura "Beta" per molti anni ogni volta.*[106]

Da un certo punto di vista, si può così affermare che il modello di sviluppo iterativo dell'ingegneria dell'usabilità tende sempre più a essere applicato non solo durante la fase di progettazione e sviluppo, ma durante tutto il ciclo di vita del prodotto. Detto in altro modo, un prodotto software tende a non uscire mai dalla fase di progettazione, fino alla sua scomparsa definitiva dal mercato.

Composizione di design pattern

La conoscenza e l'analisi delle soluzioni di progettazione adottate in altri sistemi costituisce una fonte importante di spunti per l'interaction designer. Non si tratta di copiare ciò che altri hanno concepito, ma di far tesoro dell'esperienza sviluppata in altri progetti, e di svilupparla ulteriormente adattandola a nuovi contesti. Molte soluzioni progettuali hanno una struttura o, come si dice, un *pattern* - comune, che poi s'incarna e si specializza in diversi ambiti applicativi. Più precisamente, con il termine *design pattern* si indica una soluzione generale a un problema di progettazione che si ripropone in molte situazioni, anche diverse fra loro. Non una soluzione "finita", ma piuttosto un modello, un *template* da adattare allo specifico contesto. Una parte importante del lavoro del progettista consisterà quindi nello studiare i design pattern già adottati con successo nell'ambito applicativo di suo interesse, per poterli *comporre*, adattandoli alle sue specifiche esigenze (Figura 151).

[106] Tim O'Reilly, *What is Web 2.0 – Design patterns ad business models for the next generation of software* (2005), disponibile in rete in http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html?page=1

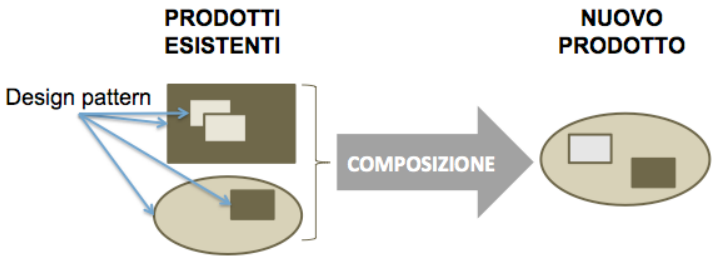


Figura 151. Progettazione per composizione

Il concetto di design pattern è nato in architettura alla fine degli anni '70, per opera dell'architetto Christopher Alexander che, affascinato dalla molteplicità e varietà di soluzioni progettuali inventate nella storia dell'architettura, si pose l'obiettivo di raccogliere in un catalogo organizzato i pattern utilizzati, da comporre poi in modo opportuno nella realizzazione di nuovi progetti:[107]

> *Ogni pattern descrive un problema che si ripresenta spesso nel nostro ambito, e quindi descrive il nucleo della soluzione di questo problema, in modo tale che la soluzione si possa utilizzare un milione di volte, senza mai rifarla nello stesso modo.*[108]

Alexander inventò anche un linguaggio, parzialmente formalizzato, per la descrizione di questi pattern. La sua opera più importante, *A Pattern Language*, contiene una collezione di 253 design pattern, dai quali l'architetto può trarre idee e indicazioni per il suo lavoro, che si tratti di progettare un centro abitato, un edificio, un singolo appartamento. La Figura 152 riporta, come esempio, il pattern 133 ("Staircase as a stage", cioè "Scala come palcoscenico"), descritto a pag.637 di questo libro. Il testo che descrive il pattern, in nostra traduzione, è il seguente:

> *Colloca la scala principale in una posizione chiave, centrale e visibile. Tratta l'intera scala come una stanza (o, se all'esterno, come un cortile). Disponila in modo che la scala e la stanza siano una cosa sola, con la scala che scende attorno a una o due pareti della stanza. Allarga il fondo della scala con finestre aperte o balaustre, e con ampi gradini, in modo che le persone che scendono lungo la scala diventino parte dell'azione della stanza mentre sono ancora sulla scala, e che le persone in basso usino naturalmente i gradini per sedersi.*

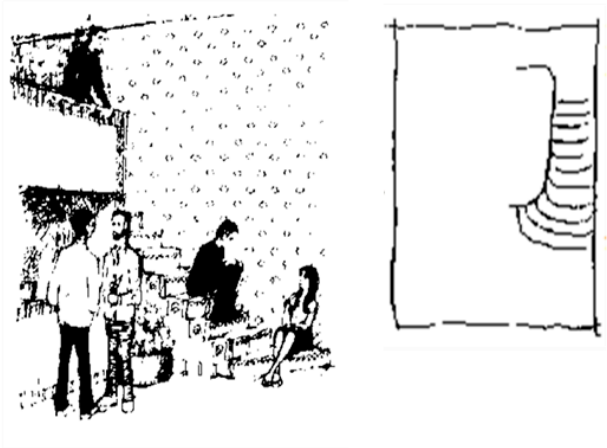


Figura 152. Un design pattern in architettura
(da Alexander, *A Pattern Language*.)

[107] C.Alexander, *The Timeless Way of Building*, Oxford University Press, 1979, e C.Alexander, S.Ishikawa, M.Silverstein, *A Pattern Language*, Oxford University Press, 1977
[108] Ibid.

Questo testo chiarisce molto bene il senso del concetto di pattern in Alexander. La scala dell'esempio non viene considerata nella sua dimensione ingegneristica, ma in quella "abitativa" (noi diremmo della "user experience"). Ciò che interessa Alexander è il rapporto – noi diremmo l'interazione – fra la scala e le persone, e i diversi modi di vivere – e di relazionarsi fra loro – che il pattern, implicitamente, suggerisce ai suoi utilizzatori. Ecco perché il concetto di design pattern, dopo essere stato adottato dall'ingegneria del software alla fine degli anni '80, ha avuto successo nella disciplina dell'interaction design.

In questo contesto esistono oggi numerose collezioni di pattern (*pattern library*), opportunamente organizzati e documentati, utilizzabili in diversi ambiti progettuali (per esempio, la progettazione di siti web). In queste collezioni, seguendo il modello di Alexander, a ciascun pattern è associata una scheda che descrive il *problema* che il pattern intende risolvere, la *soluzione* proposta, il *contesto* in cui questa può essere utilizzata, alcuni *esempi* di utilizzo del pattern, e la sua *motivazione*. Spesso vengono aggiunte considerazioni tecniche sull'implementazione del pattern, ed eventuali riferimenti alla letteratura esistente. I formati utilizzati per la descrizione dei pattern variano da caso a caso, non esiste uno standard condiviso. La Figura 153 mostra il formato delle schede descrittive di due interessanti collezioni di pattern per l'interaction design.[109]

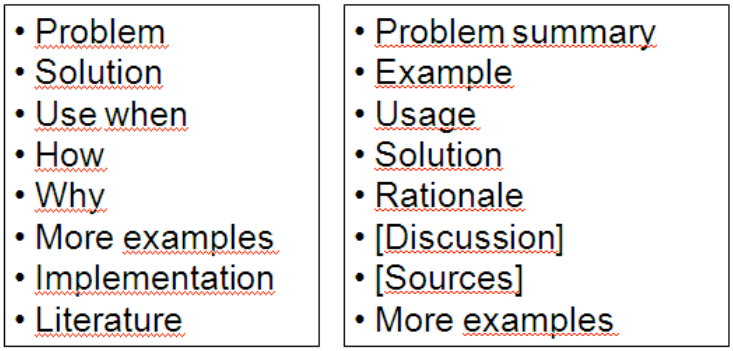


Figura 153. Due esempi di formati per la descrizione di design pattern
Fonti: van Welie (sinistra) e Toxboe (destra)

Nella raccolta di van Welie, i pattern individuati sono oltre 130, alcuni dei quali molto specifici. Essi si riferiscono al progetto di siti web. Per esempio, per le sole funzioni di ricerca in un sito, sono individuati i 13 pattern elencati in Figura 154.

Advanced search
Autocomplete
FAQ
Help Wizard
Search Box
Search Area
Search Results
Search Tips
Site Index
Site Map
Footer Sitemap
Tag Cloud
Topic Pages

Figura 154. Design pattern per le funzioni di ricerca in un sito web (van Welie)

[109] Queste due collezioni sono curate, rispettivamente, da Martijn van Welie (http://www.welie.com/) e da Anders Toxboe (http://ui-patterns.com). Esistono numerose altre raccolte di pattern per l'interaction design. Particolarmente interessante è la raccolta pubblicata da C.Crumlish, E.Malone (curatori della Yahoo! Pattern Library in rete) nel libro *Designing Social Interfaces*, O'Reilly, 2009, che raccoglie più di 100 pattern utilizzabili nella progettazione di siti web "sociali". Si veda anche http://en.wikipedia.org/wiki/Interaction_design_pattern.

Esistono anche delle raccolte di *anti-pattern*, cioè soluzioni scorrette – e quindi da evitare – a problemi che si ripropongono con una certa frequenza.

Le collezioni di pattern per il design dell'interazione sono molto utili, per diversi motivi:

- suggeriscono ai progettisti meno esperti le migliori pratiche adottate in ambiti applicativi specifici;
- raccolgono, in forma più o meno organica, lo stato della pratica corrente, cioè l'esperienza collettiva delle comunità di progetto nei vari ambiti;
- contribuiscono alla formazione di un linguaggio comune, facilitando la comunicazione fra i professionisti della disciplina;
- riducono gli sprechi di tempo e risorse dovuti alla tentazione, come si dice, di reinventare la ruota;
- facilitano l'individuazione delle soluzioni più adatte al problema specifico, contribuendo così a ridurre tempi e costi di progettazione e sviluppo;
- contribuiscono alla diffusione di "standard di fatto" ben sperimentati, con benefici effetti sull'usabilità dei sistemi.

Innovazione e comunicazione

In questo capitolo sono state indicate alcune tecniche usate nell'invenzione di prodotti interattivi. Sono differenti, ma hanno un evidente denominatore comune. Come dice Brian Arthur, nel suo libro già citato nel capitolo 1, l'invenzione di nuove tecnologie è, nella sua essenza, *associazione mentale*. I processi d'invenzione di nuove tecnologie prendono sempre le mosse da tecnologie preesistenti, variandole, associandole, scomponendole e ricomponendole secondo modalità nuove. Per esempio, riproducendo un prodotto ben conosciuto su un nuovo medium tecnologico (mimesi), o fondendo soluzioni ben note in una realtà nuova e originale (ibridazione), o trasferendo concetti da un ambito semantico all'altro (metafora) o, ancora, utilizzando soluzioni e tecnologie già sperimentate da altri come building block per costruire nuove realtà (composizione). Riprendendo ancora le parole di Brian Arthur:

> *Quando dico che l'essenza dell'invenzione è associazione mentale, non sto escludendo l'immaginazione. Anzi. Gli inventori devono avere immaginazione, innanzitutto per riconoscere l'importanza di un problema, per capire che esso può essere risolto, per individuarne soluzioni diverse, per vedere per ciascuna di esse le necessarie componenti e architetture, e per risolvere i sotto-problemi che inevitabilmente si presentano. Ma non c'è nulla di soprannaturale in questo tipo d'immaginazione. Ciò che gli inventori hanno in comune non è il "genio", o poteri speciali. Infatti non credo che esista ciò che chiamiamo geni. È, invece, la padronanza di una grande quantità di funzionalità e principi. Gli ideatori sono immersi nella pratica e nella teoria dei principi o dei fenomeni che utilizzeranno. […] Una nuova tecnologia emerge sempre da un accumularsi di componenti precedenti e di funzionalità già esistenti. Possiamo partire da questa considerazione e inquadrare la creazione con una lente grandangolare, vedendo ogni nuova tecnologia come il culmine di una progressione di apparati, invenzioni e conoscenze precedenti che conducono alla tecnologia in questione.*[110]

Forse, Thomas Alva Edison intendeva la stessa cosa quando disse, con parole più prosaiche, che "per inventare, serve una buona immaginazione e un mucchio di cianfrusaglie".

Questo processo di accumulazione e fusione di idee e funzionalità preesistenti è oggi alimentato e accelerato dal Web. Il Web è un veicolo formidabile per l'immediata circolazione di idee, esperienze e soluzioni di progetto. È la macchina più potente che l'uomo abbia mai avuto a disposizione per creare associazioni e permettere la collaborazione all'interno di comunità di persone. Prima del Web, per conoscere i prodotti dell'innovazione tecnologica occorreva muoverli – o muoversi – fisicamente. Le novità venivano presentate nei grandi eventi internazionali, e occorreva andare a vederle; per provare i nuovi software si dovevano contattare le aziende distributrici, che provvedevano a spedirli dai luoghi di produzione. Oggi – e da quei tempi "lontani" sono trascorse meno di due decadi – possiamo accedere ai prodotti software e a ogni informazione dal nostro computer di casa, spesso senza costi, e possiamo esplorare l'esistente con l'aiuto di strumenti di ricerca sempre più intelligenti. Possiamo comunicare istantaneamente con chiunque, ovunque si trovi. Possiamo creare in rete opere collettive, che nascono dalla collaborazione di migliaia d'individui. Tutti coloro che

[110] W.B.Arthur, *The Nature of Technology*, Free Press, 2009, pag. 122-124 (nostra traduzione dall'inglese).

operano nelle aree dell'innovazione – e gli interaction designer in particolare - hanno a disposizione, in rete, una gigantesca quantità di risorse alle quali attingere nel loro lavoro. Queste risorse sono di natura molto varia: dal software *open source* ai servizi di *cloud computing,* agli strumenti per la progettazione, alle recensioni dei prodotti di ogni categoria, agli articoli scientifici, alle collezioni di design pattern, alle raccolte di template per i progettisti, alle collezioni d'immagini e video di ogni tipo, fino al vasto insieme di blog gestiti da singoli progettisti che descrivono e commentano le rispettive esperienze, discutono fra loro e si segnalano vicendevolmente le risorse più interessanti praticamente in tempo reale, alle social network tematiche. Spesso questo materiale è disponibile liberamente e gratuitamente a coloro che desiderano utilizzarlo. Si tratta di un gigantesco campionario di *building blocks* – fisici e concettuali - in continua e rapida evoluzione. Il cambiamento è così rapido che la letteratura tradizionale non ce la fa più a stare al passo. Nel tempo necessario per produrre un libro a stampa, i suoi contenuti sono già obsoleti.

Un'enorme quantità di questo materiale è dedicato specificamente alla progettazione di sistemi interattivi che operano sulla rete o attraverso di essa. La rete è diventato così lo spazio principale in cui si sviluppa la tecnologia relativa ai processi connessi alla comunicazione umana. Parafrasando Brian Arthur, possiamo ben dire che *il Web si autoalimenta continuamente, creando se stesso a partire da se stesso.*

Ripasso ed esercizi

1. In che cosa consiste il procedimento d'ibridazione per la progettazione di nuove interfacce? Analizza il sistema desktop che utilizzi normalmente, e individua almeno 5 soluzioni di progetto che derivano da un procedimento di questo tipo.
2. Che cosa si intende per metafora e quali sono le differenze con l'analogia? Elenca cinque esempi di metafore.
3. Discuti l'uso del procedimento metaforico nell'interaction design. Quali sono i vantaggi e gli svantaggi, dal punto di vista del progettista e dal punto di vista dell'utente?
4. Elenca almeno dieci metafore che sono state utilizzate nella progettazione del software disponibile sul tuo computer.
5. Che cosa s'intende per design pattern?
6. Analizza la home page di tre portali web a tua scelta, e identifica le soluzioni di progetto presenti nei tre portali che a parer tuo sono riconducibili agli stessi design pattern. Dovresti cercare di riconoscere almeno 10 pattern differenti.

Approfondimenti e ricerche

1. In rete esistono numerosi interessanti "musei storici" delle interfacce utente proposte nei prodotti software a partire dai primi personal computer (per esempio, in http://www.guidebookgallery.org, aggiornato fino al 2006). Esamina uno di questi siti, e raccogli esempi d'interfacce interessanti, classificandole sulla base dei procedimenti utilizzati nella loro progettazione, descritti in questo capitolo (mimesi, ibridazione, metafora, variazione e composizione).
2. Esamina criticamente le interfacce basate su metafora da te raccolte nell'esercizio precedente, ed esprimi il tuo giudizio motivato sulla reale utilità del procedimento metaforico utilizzato per ciascuna di esse.
3. La filosofia del progetto dello Star della Xerox, il primo sistema basato sulla metafora della scrivania, è riassunta nell'articolo di Smith, Irby, Kimball, Verplank, e Harslem, *Designing the Star User Interface*, pubblicato sulla rivista Byte nel 1982, già citato negli Approfondimenti del capitolo 2 e reperibile in rete in numerosi siti. Leggi questo articolo, ed analizza la portata e le implicazioni della metafora della scrivania nel progetto dello Star.
4. Analizza i più recenti mashup di applicazioni online nel Web, e identifica le principali tipologie di ibridi che queste tecniche hanno finora prodotto. Puoi partire, per esempio, dal sito http://mashupawards.com, che segnala sistematicamente i mashup più interessanti.
5. Analizza alcune collezioni d'interaction design pattern disponibili in rete, e identifica quella che, a tuo parere, può essere più utile nella progettazione di siti web (puoi iniziare, per esempio, dalla collezione in http://ui-patterns.com, curata da Anders Toxboe, che contiene anche una vasta raccolta di screenshot interessanti, o dai link presenti nella voce "interaction design pattern" di Wikipedia). Un'altra interessante raccolta è la Yahoo! Design Patterns Library, in http://developer.yahoo.com/ypatterns/. C.Crumlish e E.Malone, curatori di questa libreria, l'hanno poi sviluppata nel libro *Designing Social Interfaces*, O'Reilly, 2009, già citato, che raccoglie

più di 100 pattern utilizzabili nella progettazione di siti web "sociali".

6. Individua qualche interessante blog che indica risorse per il design di applicazioni web (puoi iniziare, per esempio, dal blog *Tools for ideation*, in http://konigi.com).
7. Il libro di W.Brian Arthur, *The Nature of Technology – What it is and how it evolves*, Free Press, 2009 è dedicato a un'ampia discussione sui processi che determinano l'evoluzione della tecnologia. Esamina i contenuti di questo capitolo alla luce dei concetti esposti in questo libro.

9. I prototipi

Sintesi del capitolo

Questo capitolo approfondisce le caratteristiche e le finalità dei prototipi, nell'ambito dei processi di progettazione centrato sull'utente considerati nel capitolo 6. In particolare, dopo la definizione di prototipo secondo l'ISO 13407, ne viene fornita una semplice classificazione. Si sottolinea l'opportunità di utilizzare strumenti descrittivi adeguati (fra cui storyboard e, soprattutto, diagrammi di vario tipo, per esempio gli statechart dello standard UML) per specificare in dettaglio l'interazione fra utente e sistema, al fine di individuare eventuali problemi. Si descrivono quindi alcune tecniche utili per realizzare i prototipi nelle varie fasi del processo di progettazione: all'inizio (prototipi usa-e-getta: di carta, wire-frame e ipertestuali), nelle fasi intermedie e nelle fasi finali. Si forniscono vari esempi di prototipi dei diverse tipi.

Che cos'è un prototipo

Il termine deriva dal greco *prototipos,* che potremmo tradurre con "primo modello" (da *proto*, primo e *tipos*, modello). Seguendo il già citato standard ISO 13407, possiamo definire, infatti, un prototipo come:

> *una rappresentazione di un prodotto o di un sistema, o di una sua parte, che, anche se in qualche modo limitata, può essere utilizzata a scopo di valutazione.*

Questa definizione è molto ampia, e comprende oggetti di natura e di complessità molto diverse. Così, un prototipo non deve necessariamente essere un sistema funzionante, spesso può essere utile anche un semplice modello "finto" (*mock-up*). Per esempio, Jeff Hawkins, l'inventore del Palm Pilot, il primo *PDA* di successo, inizialmente tenne con sè un modellino in legno dello strumento, ovviamente non funzionante, fingendo di tanto in tanto di inserirvi o di leggervi delle informazioni.[111] Questo per meglio comprendere l'esperienza di portare sempre con sé un oggetto di questo tipo. Un altro esempio, di natura molto diversa, è il prototipo del Knowledge Navigator, realizzato con un video dalla Apple nel 1987, che abbiamo già citato a pag. 162 (Figura 117). In questo caso, il prototipo non è reale, ma solo visualizzato.

Come abbiamo visto nel capitolo 6, lo scopo principale dei prototipi è quello di coinvolgere gli utenti in tutte le fasi del progetto, fino dalle fasi iniziali. I benefici di questo approccio sono molteplici. Secondo l'ISO 13407:

- rende le decisioni di progetto più esplicite, permettendo, tra l'altro, ai progettisti di comunicare meglio fin dall'inizio del processo;
- consente ai progettisti di esplorare numerosi design concept prima della scelta finale;
- permette di incorporare nel progetto i feedback degli utenti, fin dalle prime fasi del ciclo di progettazione;
- rende possibile valutare numerose varianti del progetto e progetti alternativi;
- migliora la qualità e completezza delle specifiche del progetto.

Tipi di prototipi

Un prototipo è, dunque, un modello approssimato o parziale del sistema che vogliamo sviluppare, realizzato allo scopo di valutarne determinate caratteristiche. Queste possono essere molto varie: definire lo scopo di un prototipo è l'arte di identificare i problemi di progettazione più critici. Nelle attività di prototipazione ci si dovrebbe concentrare su quegli aspetti per i quali esistono più soluzioni possibili, fra le quali i pro e i contro si bilanciano, oppure per i quali i rischi conseguenti a una cattiva progettazione siano più elevati.

Poiché i gruppi di progetto per i sistemi interattivi sono spesso multidisciplinari, e coinvolgono persone con professionalità e priorità diverse, il termine stesso di prototipo viene usato in modo non univoco. Per esempio, un

[111] Cfr. Bergman, E. & Haitani, R., *Designing the PalmPilot: A Conversation with Rob Haitani*, in E.Bergman, *Information Appliances and Beyond*, Morgan Kaufmann, 2000.

programmatore di software potrebbe chiamare prototipo il codice di un nuovo algoritmo di cui valutare le prestazioni, mentre il designer della carrozzeria di una nuova automobile chiamerà prototipo un modello dell'auto in scala, fatto di legno. Ciò che realmente importa nella preparazione di un prototipo, in ultima analisi, è il suo *scopo*.

La tabella di Figura 155 mostra una possibile classificazione dei prototipi, sulla base del loro *scopo*, delle loro modalità d'uso, fedeltà, completezza funzionale e durata della loro vita.

Scopo	Ruolo	Serve a valutare il ruolo del prodotto nella vita del suo utente
	Interfaccia	Serve a valutare le modalità d'interazione fra utente e prodotto
	Implementazione	Serve a valutare aspetti tecnici relativi alla realizzazione tecnica del prodotto
Modo d'uso	Statico	È una rappresentazioni statica del prodotto (es.storyboard, diagrammi di vario tipo)
	Dinamico	È una rappresentazione dinamica (ma non interattiva) del prodotto, es.: video
	Interattivo	Permette agli utenti di effettuare prove d'uso del prodotto, anche se semplificate e approssimate
Fedeltà	Alta fedeltà	Assomiglia in tutti gli aspetti al prodotto finale
	Bassa fedeltà	Assomiglia alla lontana al prodotto finale
Completezza funzionale	Orizzontale	Fornisce tutte le funzioni del prodotto finale, anche se in versione semplificata o limitata
	Verticale	Fornisce solo alcune funzioni, realizzate in dettaglio
Durata	Usa e getta	Non viene conservato dopo l'uso
	Evolutivo	Viene fatto evolvere fino al prodotto finale

Figura 155. Classificazione dei prototipi

Dal punto di vista del loro scopo, possiamo classificare i prototipi in tre grandi categorie[112]:

- prototipi che servono a valutare il *ruolo* del prodotto nella vita del suo utente (*role prototype*);
- prototipi che servono a valutare l'*interfaccia* del prodotto, intesa come l'insieme delle modalità di interazione fra utente e prodotto (*look&feel prototype)*;
- prototipi che servono a valutare aspetti tecnici relativi all'*implementazione* del prodotto, per esempio particolari algoritmi utilizzati dal software *(implementation prototype)*.

Questa distinzione raramente può essere netta, poiché spesso un prototipo presenta contemporaneamente più aspetti. Ruolo, interfaccia e implementazione possono quindi essere considerati come le tre dimensioni dello "spazio" nel quale possiamo collocare ogni prototipo, e non come tre categorie separate (Figura 156). Per esempio, il Knowledge Navigator di cui si è parlato più sopra può considerarsi essenzialmente un prototipo di ruolo, con qualche aspetto, sia pure non approfondito, d'interfaccia, ma senza alcun aspetto implementativo. Pertanto, in figura, dovrebbe essere collocato nell'area indicata dal cerchio.

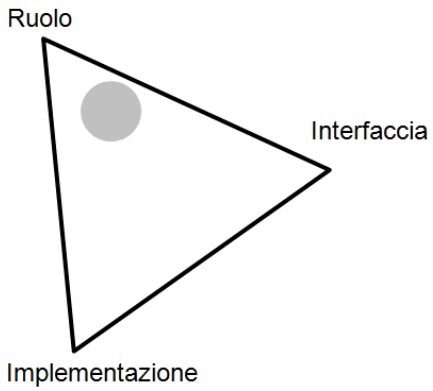


Figura 156. Lo "spazio" dei prototipi in relazione al loro scopo

Un'altra possibile classificazione dei prototipi è relativa alla loro modalità d'uso: un prototipo può essere allora *statico*, *dinamico* o *interattivo*. Nel primo caso, come nell'esempio del Palm Pilot, consisterà semplicemente in una rappresentazione statica del prodotto: una serie d'immagini, un modello tridimensionale, oppure anche una rappresentazione che permette di valutare "a tavolino" il funzionamento dinamico del prodotto, come nel caso di un flow-chart o di uno story-board. Nel secondo caso, il funzionamento dinamico del prodotto potrà essere mostrato mediante un video, come nell'esempio del Knowledge Navigator. Tuttavia, è evidente che i prototipi più utili per convalidare l'usabilità di un sistema saranno di solito quelli interattivi, che consentono agli utilizzatori di interagire col sistema in corso di progettazione, per sperimentarne l'uso - anche se in modo parziale o limitato - e individuarne, così, pregi e difetti. Un prototipo interattivo aiuta a chiarire i requisiti di progetto, che spesso sono espressi in forma vaga. Permette di osservare le reazioni dell'utente durante l'uso del sistema e di sperimentare soluzioni alternative, rapidamente e, in molti casi, a costi contenuti.

Nella pratica corrente, a volte ci si accontenta di realizzare prototipi dinamici, consistenti in una semplice sequenza d'immagini (per esempio, una serie di slide PowerPoint), che il progettista mostra all'utente in sequenza, simulando scenari d'uso tipici. Questo approccio, in realtà, non permette di valutare la usabilità di un sistema, e non dovrebbe mai sostituire l'interazione vera. Quando il progettista ci spiega, nella simulazione, come interagiremo con il sistema, mostrandocene via via la sequenza di schermate, segue un canovaccio già predisposto, che lui conosce bene. Ci presenta

[112] Cfr. S.Houde, C.Hill, *What do Prototypes Prototype?* in *Handbook of Human - Computer Interaction* (2nd Ed.), M. Helander, T.E. Landauer, P. Prabhu (ed.), Elsevier Science, Amsterdam, 1997. Anche in http://www.viktoria.se/fal/kurser/winograd-2004/Prototypes.pdf .

un'interazione ideale, preconfezionata, che non ci permette di prefigurare le difficoltà che avremo nell'uso reale, quando saremo soli con il prodotto e dovremo decidere quali azioni compiere, sulla base delle indicazioni disponibili a ogni istante. Saranno sufficienti le indicazioni che vedremo sullo schermo per suggerirci, ogni volta, il comportamento corretto? Ritornando alla metafora di Norman (pag.58), sarà facile superare i golfi dell'esecuzione e della valutazione? Saremo in grado di correggere con facilità eventuali azioni sbagliate? È molto difficile poter valutare l'usabilità di un sistema soltanto analizzando una sequenza d'immagini statiche, oppure assistendo a una simulazione condotta da altri. L'esperienza d'uso, del "metterci le mani sopra" non può essere rimpiazzata dalla sua semplice narrazione.

Come indicato nella tabella di Figura 155, quale che sia la loro finalità e il loro livello di interattività, i prototipi possono essere ulteriormente classificati in base alla loro *fedeltà* al prodotto finale, alla loro *durata* e alla loro *completezza*:

- Fedeltà al prodotto finale

  I prototipi che "assomigliano" in tutti gli aspetti al sistema finale si dicono *ad alta fedeltà* (*hi-fi prototype*). Quelli che gli assomigliano poco, *a bassa fedeltà* (*lo-fi prototype*). Questi ultimi possono essere realizzati, per esempio, con carta, cartone o legno, come il prototipo del Palm Pilot sopra citato. I prototipi a bassa fedeltà sono normalmente oggetti semplici, economici e molto facili da realizzare, ma non per questo meno utili, come vedremo fra breve.
- Completezza funzionale

  Questa distinzione riguarda il numero e la completezza delle funzionalità realizzate nel prototipo. Un *prototipo orizzontale* fornisce molte funzionalità, ma realizzate in modo schematico. Un *prototipo verticale*, al contrario, realizza compiutamente un insieme limitato di funzionalità (Figura 157). Con un prototipo orizzontale, se interattivo, si può provare l'intera interfaccia, anche se in modo approssimativo. Infatti, l'utente non potrà utilizzare nessuna funzionalità per intero: di ciascuna esisterà, per così dire, solo l'involucro esterno, o comunque una bozza rudimentale. Fornirà, quindi, un'immagine completa delle caratteristiche del prodotto, ma nessuna di esse sarà realizzata nei dettagli.
- Durata

  Un'altra importante distinzione riguarda la durata della vita del prototipo. Se il prototipo, dopo la sperimentazione, non viene conservato, esso si dice *usa e getta* (*throw-away prototype*). Se, invece, viene conservato e viene fatto evolvere o comunque integrato nel prodotto finale, si dice *prototipo evolutivo*. Normalmente, i prototipi a bassa fedeltà sono di tipo usa e getta: il modello di legno del Palm Pilot del nostro esempio non evolverà certamente nel prodotto finale dopo essere stato utilizzato. I prototipi ad alta fedeltà, di realizzazione normalmente più costosa, vengono spesso fatti evolvere nel prodotto finale.

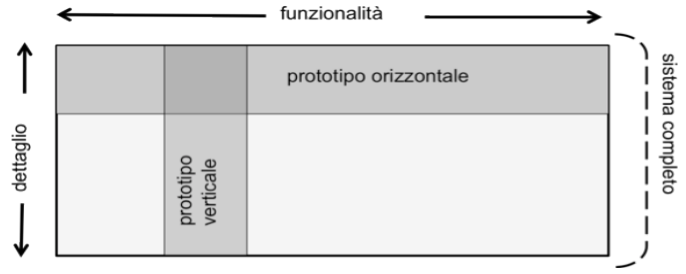


Figura 157. Classificazione dei prototipi rispetto alla completezza funzionale

In definitiva, nella realizzazione di un prototipo molte scelte sono possibili. Prototipare significa individuare di volta in volta degli obiettivi prioritari di sperimentazione, e individuare le modalità più utili per raggiungerli, costruendo un modello parziale del prodotto ed effettuandone, in qualche modo, una valutazione. Concentrando la nostra attenzione su specifici aspetti del sistema in corso di progettazione, ne trascureremo necessariamente degli altri. In definitiva, significa fare dei compromessi. In un processo di progettazione ben condotto, i diversi prototipi ci permetteranno di valutare, via via, aspetti diversi e complementari del nostro sistema.

Nonostante quest'ampio ventaglio di possibilità, nella pratica della progettazione human-centred è utile considerare, in primo luogo, quei prototipi che permettono di valutare il prodotto in rapporto con i suoi utenti. Quindi, facendo ancora una volta riferimento alla tabella di Figura 155, i prototipi di ruolo e di interfaccia, in particolare interattivi, a bassa o ad alta fedeltà. Particolarmente importanti, in un processo di sviluppo iterativo, sono inoltre i prototipi costruiti nelle prime fasi del progetto (*prototipi iniziali*), di cui tratteremo più ampiamente nel seguito.

Schizzi, storyboard e diagrammi

La definizione dello standard ISO 13407, che abbiamo ricordato all'inizio di questo capitolo, considera prototipi anche le rappresentazioni (statiche) del sistema sulla carta. Un'interpretazione così ampia della nozione di prototipo può non essere condivisa: a nostro parere, sarebbe meglio considerare come prototipi solo quelle rappresentazioni che, in qualche modo, permettono di *provare* l'interattività del sistema, e non soltanto di descriverla. Tuttavia, anche le rappresentazioni statiche hanno grande importanza pratica per il progettista, che le usa per descrivere e definire i dettagli di quello che si propone di fare. Pertanto, ne descriviamo qui di seguito le principali tipologie, lasciando alle sezioni successive un approfondimento sui prototipi interattivi.

Schizzi

Quasi sempre lo sviluppo di un'idea di progetto parte da uno schizzo, anche molto approssimativo, sulla carta. Per esempio, la Figura 158 mostra alcuni schizzi iniziali del progetto di un orologio con funzioni di cellulare, realizzato da alcuni studenti. Le immagini, appena abbozzate, servono solo a fissare le idee. Verranno poi organizzate in forma più strutturata nei prototipi successivi, che permetteranno di effettuare i primi test con gli utenti (per esempio, prototipi di carta, come vedremo più oltre).[113] Nella realizzazione di questi schizzi, i progettisti non seguono di solito metodi precisi, ma cercano di visualizzare rapidamente sulla carta le prime ipotesi di lavoro. Per esempio, nel disegno di Figura 158, le frecce indicano, anche se in modo non sistematico, alcune possibili sequenze dell'interazione (cioè passaggi del prodotto da uno stato all'altro, per esempio a seguito della pressione, da parte dell'utente, di un pulsante).

Nei gruppi di progetto che coinvolgono più persone, spesso questi schizzi sono realizzati su lavagne di grandi dimensioni, in sessioni di *brainstorming* durante le quali i partecipanti suggeriscono, discutono e modificano, in modo totalmente libero, soluzioni diverse. La proposta finale, condivisa, servirà come base per la realizzazione di un prototipo vero e proprio, nel processo iterativo di progettazione.

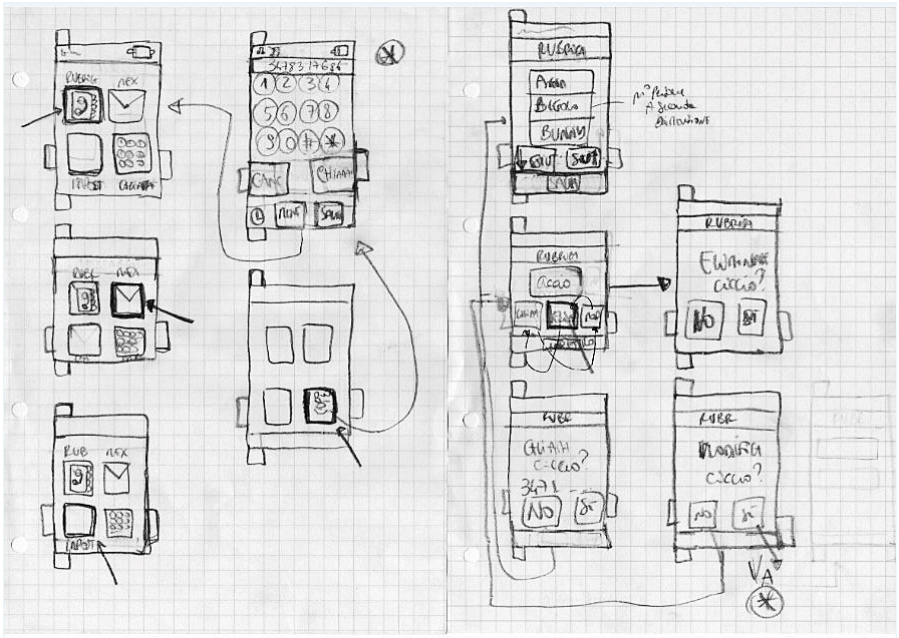


Figura 158. Schizzo iniziale per un orologio da polso con funzioni di cellulare

[113] Per altri esempi interessanti di schizzi si veda, per esempio, http://woorkup.com/2009/12/28/10-beautiful-sketches-for-website-prototypes/.

Storyboard

La tecnica dello *storyboarding*, introdotta nell'industria cinematografica dagli anni '30 del secolo scorso, consiste nel realizzare una serie di disegni che illustrano, inquadratura per inquadratura, ciò che verrà girato sul set di ripresa (Figura 159). Accanto ai disegni possono essere indicati i movimenti della macchina da presa, o brani del dialogo, o altre annotazioni. La sua funzione principale è quella di supporto alla progettazione del film: aiuta il regista a trovare il modo migliore per visualizzare una scena, e a comunicare le sue idee ai membri della sua *troupe* (il direttore della fotografia, lo scenografo, il tecnico delle luci, …) o alla produzione. Nella progettazione degli spot pubblicitari, lo storyboard viene usato anche per comunicare al cliente le varie proposte alternative prima della realizzazione.

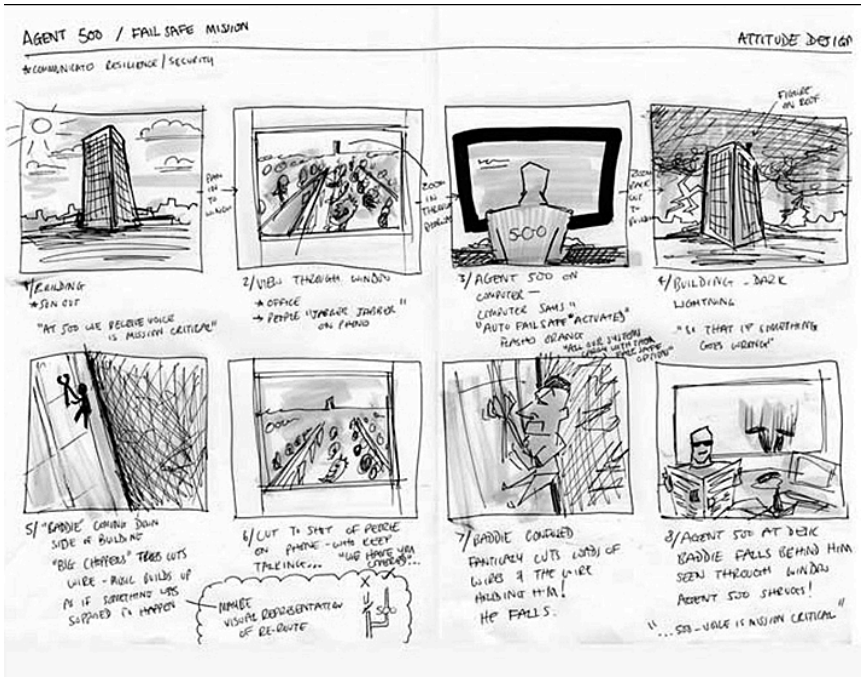


Figura 159. Un esempio di storyboard per uno spot pubblicitario (da http://www.attitudedesign.co.uk)

La tecnica dello storyboarding può essere adottata anche nell'interaction design, per rappresentare una *storia d'uso*: la sequenza degli stati del sistema durante una particolare interazione con l'utente. Per esempio, la Figura 160 mostra lo storyboard di una possibile sequenza di navigazione all'interno del sito web di un negozio di CD musicali. In questo caso gli schizzi che rappresentano le diverse schermate sono solo abbozzati, poiché l'obiettivo del progettista era quello di mostrare un possibile percorso di navigazione per la selezione di un CD a partire dalla scheda dell'artista. Selezionando la voce "Artisti" nel menu principale, passando attraverso un elenco alfabetico degli artisti disponibili, si arriva alla scheda dell'artista desiderato, che ne elenca la produzione discografica. Da questa, si raggiunge la scheda che descrive il CD selezionato.

Per l'interaction designer lo storyboard risulta utile solo in alcuni casi, soprattutto in fase di definizione dei requisiti, per esempio nella preparazione di video o animazioni che presentino degli scenari d'uso (pag. 149) o in alternativa ad essi, quando non se ne vogliono affrontare i costi di produzione. L'utilizzo degli storyboard per rappresentare sequenze d'interazione, come nell'esempio di Figura 160, è meno frequente, poiché per questo scopo esistono strumenti più potenti, come gli *statechart*, che vedremo fra poco.

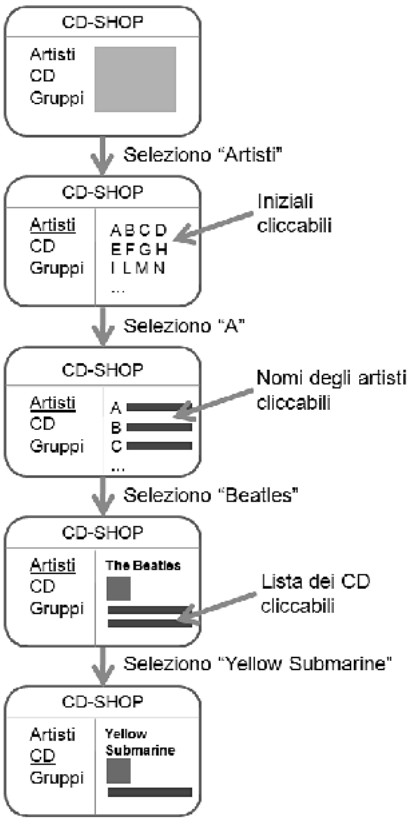


Figura 160. Storyboard di un sito web di un negozio di CD musicali

Diagrammi per macchine a stati

Per rappresentare adeguatamente sulla carta l'interazione con l'utente, ci servono strumenti più espressivi degli storyboard, che ci permettano di rappresentare *tutte le possibili sequenze d'interazione*. A questo scopo sono state sviluppate svariate notazioni, che fanno generalmente uso di diagrammi bidimensionali più o meno complessi. Ogni progettista potrà scegliere la notazione che preferisce. In questo libro ne descriveremo solo una, semplice e particolarmente comoda per questo scopo: i *diagrammi per macchine a stati,* detti anche *statechart*. Si tratta di diagrammi proposti da David Harel nel 1987 come strumento di modellazione di sistemi complessi, e in seguito adottati nel linguaggio UML come uno degli strumenti di base. [114] A differenza di altri diagrammi a stati, essi permettono di descrivere un sistema in modo gerarchico, cioè per livelli di astrazione successivi. Questo è molto importante, per mantenere entro limiti accettabili la complessità dei diagrammi, che altrimenti potrebbero diventare troppo grandi per essere facilmente gestibili.

I diagrammi per macchine a stati sono strumenti semplici ma flessibili e potenti, che servono a descrivere il comportamento di sistemi di ogni tipo. I costrutti più utili per rappresentare le interazioni utente-sistema sono descritti nell'Appendice, alla quale rimandiamo per una descrizione più completa. Qui di seguito li presentiamo informalmente, attraverso la descrizione di un esempio elementare.

Essenzialmente, uno statechart è costituito da nodi e da archi (Figura 161):

[114] *UML* (*Unified Modeling Language*) è un linguaggio visuale standardizzato comprendente varie notazioni per la modellazione dei sistemi complessi. UML è nato nella seconda metà degli anni 90, dall'integrazione delle principali metodologie di progettazione di software allora esistenti, per opera di J.Rumbaugh, G.Booch e I.Jacobson. Oggi questo linguaggio è correntemente usato nell'ingegneria del software, soprattutto per la progettazione di sistemi orientati agli oggetti.

- ogni *nodo* rappresenta uno *stato del sistema:* nel nostro caso, uno stato del dialogo con l'utente. Per esempio, potrebbe rappresentate una particolare schermata del computer;
- ogni *arco* (orientato) rappresenta una *transizione* da uno stato all'altro.

La transizione è innescata da un *evento* che normalmente (ma non sempre), corrisponde a un'azione dell'utente, e può causare l'esecuzione di un'*azione* del sistema. La transizione può essere subordinata al verificarsi di una *condizione*.

Ogni nodo è rappresentato con un rettangolo dai bordi arrotondati, contenente il nome dello stato che rappresenta. Ogni arco è etichettato con il nome dell'evento, della condizione e dell'azione associati. Evento, condizione e azione sono opzionali.

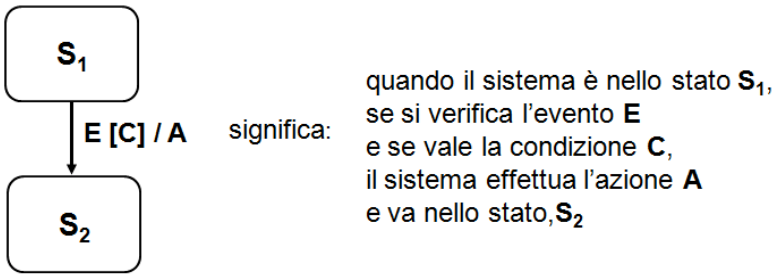


Figura 161. Gli elementi di base degli statechart

Di solito, gli eventi sono costituiti da azioni dell'utente: la pressione di un pulsante, la selezione di una voce di menu, la compilazione di uno o più campi di una form, e così via. Pertanto, per indicare un evento potremo usare semplicemente il nome del pulsante, la voce del menu o una breve frase che descriva l'azione compiuta dall'utente.

Con questi semplici diagrammi possiamo descrivere bene interazioni complesse. Per esempio, il diagramma di Figura 162 definisce l'interazione di una macchina erogatrice di bevande con il suo utente:

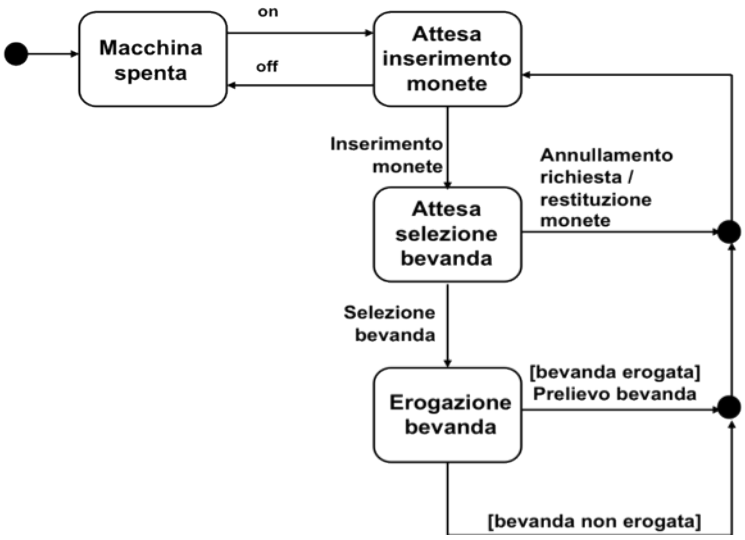


Figura 162. Statechart che definisce l'interazione con un distributore di bevande

Rappresentare con degli statechart il dialogo fra utente e sistema è un esercizio molto utile, perché costringe il progettista a esplicitare tutti i percorsi che l'utente può seguire interagendo con esso. Lo costringe a effettuare delle scelte, *prima* di realizzare il prototipo. Molto spesso questo mette in evidenza aspetti critici nella realizzazione del caso d'uso, che possono avere conseguenze importanti sull'usabilità. Per esempio, il comportamento dell'erogatrice di Figura 162 lascia molto a desiderare. Infatti, nel caso in cui la bevanda sia esaurita, la macchina si mette semplicemente in attesa di nuove monete, senza restituire quelle ricevute e senza segnalare alcunché.

Un grosso vantaggio degli statechart è che permettono di rappresentare il sistema per livelli di astrazione successivi. Questo consente di rappresentare il dialogo nei dettagli, usando tuttavia dei diagrammi di dimensioni contenute: basterà definire un diagramma separato per ogni caso d'uso individuato nei requisiti, e quindi costruire un diagramma di alto livello, che li richiama tutti, fornendo così il quadro complessivo. Si rimanda all'Appendice per i dettagli.

Gli statechart sono particolarmente utili per definire i percorsi che corrispondono a situazioni d'errore (quando, cioè, l'utente compie azioni scorrette). Questi percorsi possono essere numerosi e potremmo essere tentati di tralasciarli, rimandando le scelte concernenti il trattamento degli errori al momento della codifica dei programmi. Questa non è una buona soluzione. È molto meglio analizzare e risolvere subito tutti i problemi, per evitare che emergano nelle fasi successive, degradando l'usabilità del sistema e imponendo rifacimenti costosi. Tanto più che i diagrammi per le macchine a stati permettono di descrivere il trattamento degli errori in modo assai semplice, associando a ogni evento di errore un'azione del sistema, che visualizzi il corrispondente messaggio all'utente.

Esistono numerosi programmi che permettono di costruire e mantenere facilmente diagrammi di ogni tipo. Uno dei più noti è Visio, della Microsoft.

Prototipi iniziali

Nelle prime fasi del progetto, molte strade sono ancora aperte, ed è in genere utile esplorare più di una soluzione, prima di scegliere quella che sarà sviluppata nei dettagli. I prototipi iniziali servono proprio a questo, e sono quindi molto importanti. Essi saranno quasi sempre di tipo usa e getta, ed è opportuno che siano realizzabili molto velocemente e a costi molto contenuti. I progettisti potranno così sperimentare e valutare anche numerose soluzioni alternative. Citando ancora le parole dello standard ISO 13407:

> *Gli utenti possono essere coinvolti molto presto nel progetto, mediante l'uso di modelli statici realizzati sulla carta. È possibile presentare agli utenti le bozze delle schermate o una rappresentazione del prodotto, chiedendo loro di provarli in un contesto realistico. In tal modo si possono valutare rapidamente ed economicamente aspetti del progetto (per esempio, quanto sia facile navigare attraverso una gerarchia di menu). Per i prodotti hardware, analoghi benefici possono essere ottenuti con l'uso di modelli tridimensionali statici, costruiti con materiali semplici. Nelle fasi iniziali, anche i prototipi più rudimentali possono risultare preziosi, per esplorare soluzioni alternative. Anche se può essere utile presentare le soluzioni di progetto nel modo più realistico possibile, è consigliabile evitare di investire troppo tempo o denaro nella loro realizzazione, anche perché ciò potrebbe produrre una resistenza alle modifiche da parte dei progettisti. In un approccio human-centred, un prototipo non è semplicemente una demo per mostrare un'anteprima del prodotto agli utenti. Esso serve a raccogliere le loro reazioni, per poi utilizzarle nell'orientare le attività di progettazione successive. Quando non fosse consigliabile mostrare i prototipi agli utenti all'inizio del processo di progettazione (per esempio, per ragioni di riservatezza), le valutazioni potranno essere condotte da esperti. Queste possono essere utili e poco costose, e complementare i test con l'utente. In ogni caso, in un processo di progettazione human-centred, almeno i test finali dovrebbero essere condotti con utenti reali.*[115]

Le tecniche possibili, anche molto semplici, sono varie. Descriviamo qui quelle che consideriamo più importanti: i prototipi di carta, i prototipi wireframe e i prototipi ipertestuali.

Prototipi di carta

I prototipi più semplici che permettono di provare, anche se in modo rudimentale, l'interazione con l'utente, sono i prototipi di carta (*paper prototype*). L'interfaccia del sistema viene disegnata a bassa fedeltà su fogli di carta (o cartoncini, o *post-it*), che vengono usati per effettuare una simulazione "manuale" del sistema, con utenti-cavia. La Figura 163 mostra alcuni cartoncini utilizzati per la simulazione di un'applicazione per un iPhone. Ogni cartoncino, in grandezza naturale, rappresenta sommariamente una singola schermata . Durante la simulazione, il progettista presenta all'utente la prima schermata, e l'utente interagisce con essa simulando l'interazione (per esempio, "premendo" col dito la rappresentazione di un bottone, o fingendo di compilare un campo di input, e così via). Il progettista risponderà, in

[115] Nostra traduzione dall'inglese.

funzione delle azioni dell'utente, presentando il cartoncino con la schermata successiva, e così via. Le reazioni e le difficoltà dell'utente sono esaminate e commentate, dopodiché l'interfaccia si corregge, sempre sulla carta, e si riprova.



Figura 163. Prototipo di carta

I prototipi di carta sono poco utilizzati nella pratica, perché i progettisti tendono a non prenderli troppo sul serio. Sono considerati quasi dei giochi, non si comprende come riproduzioni così rudimentali e statiche possano dare suggerimenti di una qualche utilità. "È troppo facile, non può funzionare, proviamo qualcosa di più serio." Questo è un grande errore, perché i prototipi di carta sono realmente molto utili. Infatti, possono essere realizzati molto in fretta e a costi molto contenuti. Per esempio, il prototipo di Figura 163 rappresenta *tutte* le schermate principali di un'applicazione per iPhone non banale. Eppure, si tratta solo di 30 cartoncini, confezionati in un tempo molto limitato: il template di base è stato semplicemente fotocopiato dal manuale dell'iPhone, e poi riprodotto nel numero di copie necessario. È vero che l'interfaccia è rappresentata molto grossolonamente, a matita, che l'interazione è lenta e la user experience molto diversa da quella vera, ma nella fase iniziale del progetto non serve altro. Anzi, una maggior precisione ci farebbe sprecare inutilmente del tempo. Ciò che ci serve è una rapida simulazione del funzionamento, con uno o due utenti, che troveranno sicuramente qualcosa che non va. Infatti, le prime prove mettono spesso in evidenza difetti macroscopici. Con un prototipo di carta questi possono essere corretti in pochi minuti, e il risultato rimesso in prova con nuove simulazioni (naturalmente, con utenti diversi). La Figura 284 mostra un'immagine tratta dalla ripresa video della simulazione del prototipo di Figura 163[116]

In conclusione, è veramente conveniente realizzare i prototipi di carta come primo passo, subito dopo avere adeguatamente schematizzato i vari percorsi dell'interazione con l'uso dei diagrammi di interazione. Questi diagrammi saranno un aiuto prezioso per la persona cui è affidato il compito di gestire i cartoncini del prototipo durante la simulazione con l'utente. Infatti, senza una documentazione scritta e immediatamente interpretabile, anche la

[116] La tecnica è molto semplice, ma si presta a molteplici varianti. Esiste anche un intero libro sull'argomento: C.Snyder, *Paper Prototyping – The fast and easy way to design and refine user interfaces*, Morgan Kaufmann Publishers, 2003.

simulazione di un sistema elementare potrebbe essere piuttosto laboriosa, con il rischio di fornire all'utente risposte diverse da quelle previste.

La tecnica del Mago di Oz

Questa tecnica consiste nel realizzare un prototipo interattivo, in cui però le risposte – o parte di esse – siano fornite, se possibile all'insaputa dell'utente, da parte di un essere umano che operi, per così dire, "dietro le quinte" come, appunto, il mago di Oz della favola.[117]

Per esempio, nel prototipo di un sistema di *query*, l'utente potrebbe formulare un'interrogazione, e un esperto nascosto (il mago di Oz) potrebbe riscrivere l'interrogazione in una forma normalizzata e presentarla all'utente per la sua approvazione, e quindi fornire la risposta richiesta, simulando l'accesso a una base dati ancora inesistente. Oppure, la tecnica può essere utilizzata per realizzare prototipi iniziali di sistemi che dialogano in linguaggio naturale, per esempio per raccogliere indicazioni sui costrutti linguistici preferiti dagli utenti. Altri sistemi che si prestano bene all'uso di questa tecnica sono i risponditori automatici dei *call center*, o i cosiddetti sistemi IVR (*interactive voice response systems*), in cui l'utente richiede a voce delle informazioni (l'orario di treni o aerei, previsioni metereologiche, ecc.). e il sistema (nel nostro caso, il mago di Oz) fornisce risposte vocali a partire da *script* predisposti.

L'impiego di questa tecnica non è banale, come potrebbe sembrare a prima vista. I compiti del mago, apparentemente semplici, si rivelano spesso cognitivamente impegnativi. Affinché il prototipo risulti realistico, le risposte del mago devono essere consistenti per quanto riguarda i contenuti e i tempi di reazione. In particolare: situazioni simili devono provocare le stesse risposte, e queste devono essere conformi alle aspettative dell'utente. Per esempio, se il mago fosse troppo lento nel rispondere, l'utente potrebbe pensare di avere fornito una richiesta scorretta, o che il sistema è sovraccarico, o che si trova in uno stato di errore. In sostanza, il mago non può essere un improvvisatore: deve essere ben preparato e avere a disposizione una serie completa di supporti pronti all'uso (diagrammi per macchine a stati, schemi delle risposte, e così via). Per semplificare questi compiti può essere opportuno, in molti casi, che il ruolo del mago sia sostenuto da più di una persona: per esempio, una persona dedicata alla simulazione dell'input/output, e un'altra persona dedicata alla simulazione delle operazioni di elaborazione delle risposte.

Prototipi wire-frame

I *prototipi wire-frame* prendono il nome dai *modelli wire-frame* (letteralmente: *modelli in fil di ferro*) della grafica computerizzata.[118] Sono prototipi interattivi a bassa fedeltà, di solito usa-e-getta, nei quali la grafica è estremamente semplificata, e mostra solo i contorni degli oggetti. Permettono di sperimentare le modalità principali di interazione, prima che i dettagli della grafica siano definiti.

Per esempio, la metodologia di realizzazione di un sito web in sette fasi, di cui si è accennato a pag.136, prevede che il primo prototipo del sito (chiamato *prototipo di navigazione*) sia di tipo wire-frame. È un prototipo interattivo che permette di provare la navigazione nel sito (menu, titoli delle pagine, *bread-crumbs* ecc.), senza che sia stata definita la grafica, e prima della redazione dei contenuti. La Figura 164 mostra la home page del prototipo wireframe del sito web di un importatore di birra.[119] Si tratta quindi di un contenitore vuoto, ma completamente navigabile, il cui unico scopo è di permettere di verificare l'adeguatezza della struttura dei menu e della struttura logica delle pagine del sito.

---

[117] Il nome deriva da *Wonderful Wizard of Oz* (1900), un celebre romanzo per ragazzi dello scrittore statunitense L. Frank Baum (1856-1919). E' la storia di Dorothy, una bambina che viene trasportata da un ciclone, con tutta la sua casa, dal Kansas nel regno di Oz. Per tornare nel Kansas, Dorothy dovrà compiere una serie di imprese assegnatele da un mago che controlla il regno. Alla fine, si scoprirà che il mago di Oz non è altro che un vecchietto senza poteri, che si nascondeva dietro un paravento per simulare le sue magie. Il primo a proporre questa tecnica, e a darle il nome, è stato John F. Kelley, nella sua tesi (circa 1980).

[118] Con questo termine si indica la rappresentazione grafica di oggetti tridimensionali, disegnando soltanto i bordi dell' oggetto, il quale resta in questo modo "trasparente" e sembra, appunto, costruito con il filo di ferro. Nella grafica computerizzata, questo metodo richiede calcoli molto più semplici rispetto alla rappresentazione di superfici, ed è quindi considerevolmente più veloce.

[119] La descrizione delle sette fasi di progettazione di questo sito, sviluppata con la metodologai citata, si trova in appendice al libro R.Polillo, *Plasmare il Web*, Apogeo, 2006.

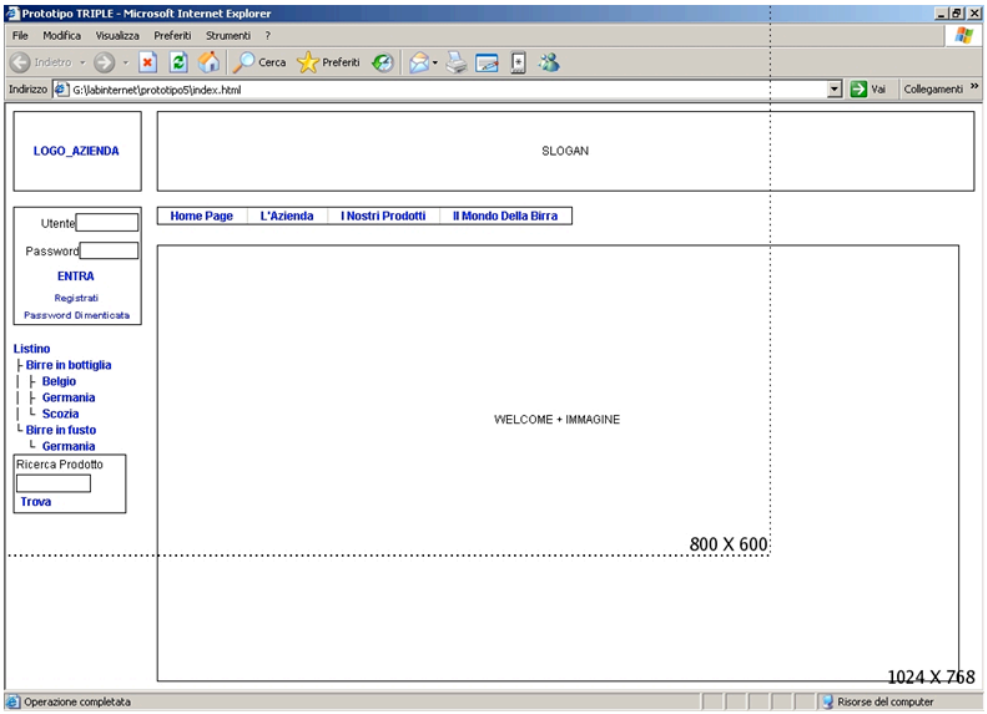


Figura 164. Prototipo di navigazione del sito di un importatore di birra

Le pagine sono infatti costituite solo da una *gabbia logica* in bianco e nero, e non contengono alcuna anticipazione sulla grafica definitiva: nessuna immagine, nessun logo o decorazione, nessun colore. Sono vuote di contenuti informativi, ma contengono i titoli definitivi e a volt il "testo finto" inserito nella gabbia logica, per mostrare gli ingombri delle varie aree logiche previste. Il fatto che questo prototipo sia il più possibile astratto e non mostri grafica e contenuti è molto importante. In questo modo, chi proverà a usare il prototipo potrà concentrarsi sulla meccanica di navigazione e sulla struttura logica delle pagine, senza essere distratto da altri elementi, come per esempio i colori o le frasi del testo. Tuttavia, la forma e le dimensioni dei menu dovranno essere rappresentate con accuratezza sufficiente per apprezzarne la leggibilità nelle diverse risoluzioni del video.

Un prototipo wire-frame di un sito ha il compito di rendere "vivo" il progetto, fino a quel momento descritto soltanto sulla carta. Ciò permette di valutarne molto rapidamente pregi e difetti: il prototipo wireframe di un sito web si può costruire in brevissimo tempo utilizzando uno strumento come DreamWeaver o PageMaker. Il fatto che questi strumenti producano delle vere pagine web è molto importante: in questo modo, infatti, si potrà controllare che il layout del sito sia compatibile con diverse risoluzioni video (in figura sono indicate le risoluzioni 800x600 e 1024x768). Gli eventuali problemi emergono con evidenza ed è facile realizzare e provare rapidamente soluzioni alternative. Data la sua semplicità, i test saranno molto rapidi e richiederanno in genere soltanto pochi minuti.

Prototipi ipertestuali

Un'altra tecnica molto utilizzata per costruire prototipi iniziali fa uso di strumenti per la costruzione di ipertesti. In questo caso, il prototipo è costituito da una serie d'immagini (*snapshot*) che rappresentano l'aspetto del prodotto in corso di progettazione. Le varie snapshot sono legate fra loro da link ipertestuali, cliccando i quali l'utente passa da una snapshot all'altra, simulando così l'interazione con il prodotto.

I prototipi ipertestuali possono essere realizzati facilmente, a costi molto limitati, con vari strumenti. Quelli più usati sono i programmi per la costruzione di presentazioni (per es. PowerPoint della Microsoft), che normalmente permettono di legare fra loro le varie slide con link ipertestuali. In questo caso:

- ogni snapshot del prodotto viene rappresentata su una slide;
- su ogni snapshot vengono realizzate aree cliccabili di forma opportuna (pulsanti, campi, ecc.), con link ad altre slide;
- cliccando su queste aree, l'utente naviga nell'ipertesto, simulando l'interazione con il prodotto.

La Figura 165 mostra un esempio di prototipo cliccabile realizzato con PowerPoint, relativo al "cellulare da polso", di cui abbiamo visto il primo schizzo in Figura 158.



Figura 165. Prototipo PowerPoint del cellulare da polso di Figura 158

Si tratta della slide iniziale dell'ipertesto, sulla quale sono state definite diverse aree cliccabili, corrispondenti ai vari bottoni virtuali dell'orologio, ipoteticamente realizzato con tecnologia touch screen.[120] Così, cliccando sull'area inferiore, comparirà una serie d'icone associate ai casi d'uso principali, disegnate su una seconda slide. Cliccando ancora sull'icona rappresentante un telefono, comparirà la slide con la tastiera numerica e i pulsanti per gestire la telefonata e la rubrica (Figura 166). I pulsanti presenti sulla destra in Figura 165 sono stati predisposti per simulare degli eventi non generati dall'utente: la ricezione di una telefonata o di un sms, la ricezione di una chiamata persa, ecc. Per esempio, premendo il pulsante "Chiamata da contatto", comparirà la slide che mostra, sullo schermo dell'orologio, il nome del chiamante e i pulsanti per accettare o rifiutare la chiamata. Anche i comportamenti condizionali possono essere realizzati con la stessa tecnica, inserendo appositi pulsanti accanto al prototipo, che determinano il percorso all'interno dell'ipertesto. Per esempio, dopo avere composto un numero di telefono cliccando sulla tastiera, si potrà simulare la condizione di "numero occupato" predisponendo una coppia di pulsanti ("Libero" e "Occupato") che permettano di scegliere il percorso desiderato nell'ipertesto.

[120] In questo prototipo, che aveva lo scopo di verificare l'impostazione generale del progetto, le dimensioni del cellulare sono più grandi di quelle reali. In questo caso, al prototipo iniziale avrebbe dovuto seguire un prototipo di dimensioni reali, anche non interattivo ma indossabile, per definire in modo preciso le dimensioni dei tasti, dei caratteri e del display, e verificare la effettiva usabilità dell'oggetto.



Figura 166. Prototipo PowerPoint del cellulare da polso di Figura 165

I vantaggi dell'uso di prototipi ipertestuali di questo tipo sono evidenti. Innanzitutto, i prototipi sono facili da realizzare e da modificare, e la simulazione non richiede un mago di Oz, e risulta più realistica e fluida. Inoltre la grafica del prodotto finale può essere simulata con un buon livello di dettaglio.

Esiste tuttavia anche qualche svantaggio. In primo luogo, questi prototipi permettono solo interazioni semplici, di tipo *point & click.* Interazioni più complesse non sono realizzabili a costi ragionevoli, e dovranno quindi essere simulate in modo approssimativo, o addirittura immaginate. Per esempio, se volessimo riprodurre le operazioni di composizione del numero di telefono nel cellulare da polso (Figura 166), dovremmo realizzare una slide per ogni diversa cifra digitata, e inserire un link a questa slide nell'area del pulsante che porta quella cifra. Il numero delle slide dovrebbe essere uguale al numero delle combinazioni possibili. Ciò non è evidentemente realizzabile. e d'altra parte non porterebbe alcun reale vantaggio nelle prove con gli utenti. Infatti, lo scopo di un prototipo iniziale è quello di convalidare l'impostazione complessiva del prodotto, e non tanto l'interazione di dettaglio con i vari controlli, che si può rimandare a un prototipo successivo, da realizzare con tecnologie più adatte. Per tornare al nostro esempio, la composizione del numero di telefono potrà essere simulata, semplicemente, cliccando in un punto qualsiasi della tastiera e facendo apparire sul display del cellulare un numero qualsiasi, sempre quello. Una slide, in questo modo, sarà sufficiente.

Inoltre, ci sono dei limiti pratici alla complessità degli ipertesti realizzabili, superati i quali il prototipo diventa poco gestibile da chi lo sviluppa. L'esperienza di uso di PowerPoint per questo scopo, compiuta da chi scrive in centinaia di progetti didattici realizzati dagli studenti, suggerisce che la "soglia di ingestibilità" dei prototipi si colloca intorno alle 150 slide. Oltre questo limite, l'ipertesto diventa troppo complesso, e quindi difficilmente mantenibile con gli strumenti elementari a disposizione in PowerPoint. È allora conveniente spezzare i prototipi più complessi in ipertesti separati, ciascuno dei quali dedicato a uno specifico aspetto del sistema. Se, tuttavia, la simulazione non entra in aspetti di eccessivo dettaglio, questi limiti sono ampiamente sufficienti a realizzare prototipi di ragionevole completezza, che possono essere tranquillamente contenuti in 80-100 slide. Per esempio, quello del cellulare da polso ha richiesto poco più di 100 slide per la simulazione di tutte le principali funzioni: effettuazione e ricevimento telefonate, gestione della rubrica, invio e gestione sms, impostazione data e ora, impostazione sveglia, impostazione volume suoneria, collegamento con auricolare tramite bluetooth.

La Figura 167 mostra il prototipo di un telecomando per un apparato multifunzionale audio-video, anch'esso realizzato a scopo didattico, con PowerPoint. In questo secondo esempio, ogni slide, oltre alla immagine del telecomando, contiene una rappresentazione dello schermo del televisore e del pannello del componente controllato (in figura, il

sistema VHS). Cliccando sui vari tasti, l'immagine dello schermo e il display del pannello viene modificata di conseguenza. Nonostante l'apparente complessità del prototipo, anche in questo caso circa cento slide hanno sono state sufficienti per simulare con alcuni utenti i seguenti compiti:

1. Accendere il sistema;
2. Visualizzare diversi canali televisivi;
3. Visionare una videocassetta;
4. Registrare le immagini dell'ultimo canale selezionato su DVD;
5. Ascoltare della musica.

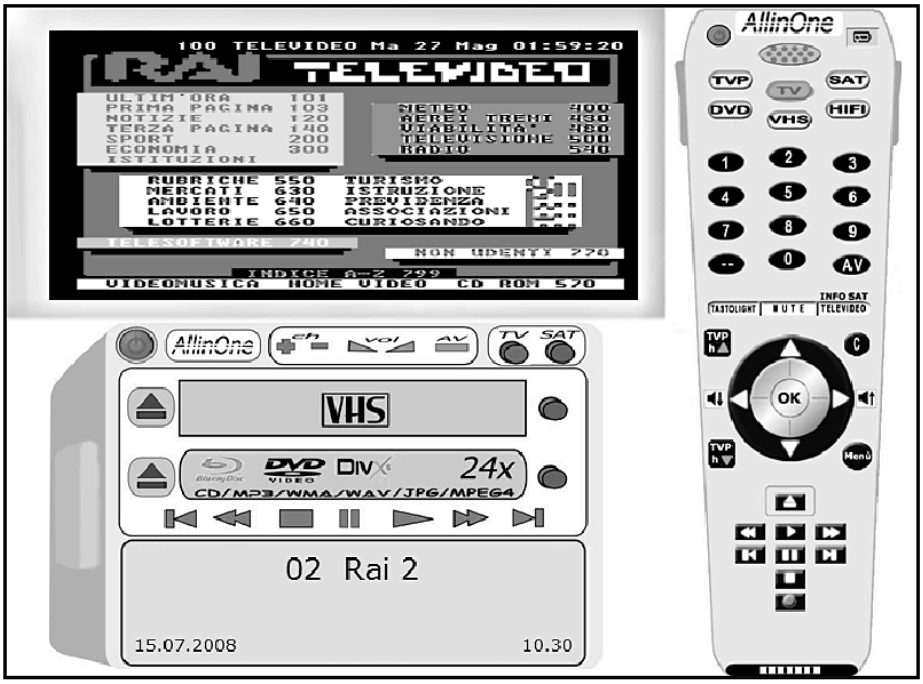


Figura 167. Prototipo PowerPoint di un telecomando per un sistema audio-video

Altri strumenti per la costruzione di ipertesti sono i generatori di pagine HTML come, per esempio, Dreamweaver della Adobe o FrontPage della Microsoft. Come abbiamo visto sopra, questi strumenti sono particolarmente adatti per la realizzazione dei prototipi di navigazione dei siti web, indipendentemente dalla tecnologia utilizzata per la realizzazione del sito finale. Sono invece sconsigliabili per la prototipazione di altri tipi di applicazioni software, perché la grafica è poco controllabile e il loro orientamento alla costruzione di siti web tende a influenzare le scelte di progetto. In pratica, è facile che il prototipo tenda ad assomigliare a un sito, indipendentemente dalla sua natura.

In ogni caso, è bene evitare di usare strumenti di prototipazione che creino difficoltà tecniche, o che il progettista non padroneggi completamente. Infatti, nella realizzazione di un prototipo tutti gli sforzi dovrebbero essere concentrati sulla concezione e sulla valutazione del prodotto, e non sulla risoluzione dei problemi tecnici posti dallo strumento utilizzato. Inoltre, strumenti troppo complessi o "invasivi" possono influenzare, con le loro peculiarità, le scelte di progetto per il sistema prototipato ("questa interfaccia è troppo complicata da prototipare con questo strumento, quindi ne scelgo un'altra").

Una soluzione molto valida in una grande varietà di sistemi è costituita dall'accoppiata prototipo di carta / prototipo PowerPoint. Inizialmente si costruisce e si sperimenta un prototipo di carta a bassa fedeltà. Quando la soluzione è abbastanza consolidata, la si realizza nuovamente ad alta fedeltà in un prototipo PowerPoint navigabile, e si effettuano

nuove prove con gli utenti. La Figura 168 mostra una scheda del prototipo di carta di un'applicazione per palmare, accanto alla slide corrispondente del prototipo PowerPoint realizzato successivamente.

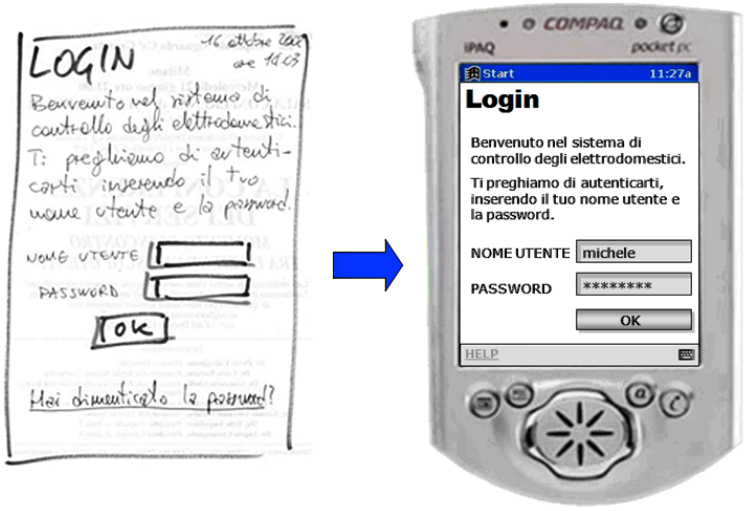


Figura 168. Dal prototipo di carta al prototipo PowerPoint

In questo progetto, pensato per un palmare, la costruzione del prototipo è stata facilitata dalla disponibilità, in rete, dell'immagine precisa del modello di palmare prescelto, che è stata quindi usata come base in tutte le slide del prototipo. Oggi esistono in rete numerose librerie d'immagini preconfezionate (chiamate *design stencil*) che possono essere usate per comporre rapidamente la grafica dei prototipi. La Figura 169, per esempio, mostra un insieme di design stencil per la prototipazione di applicazioni per iPhone.[121]

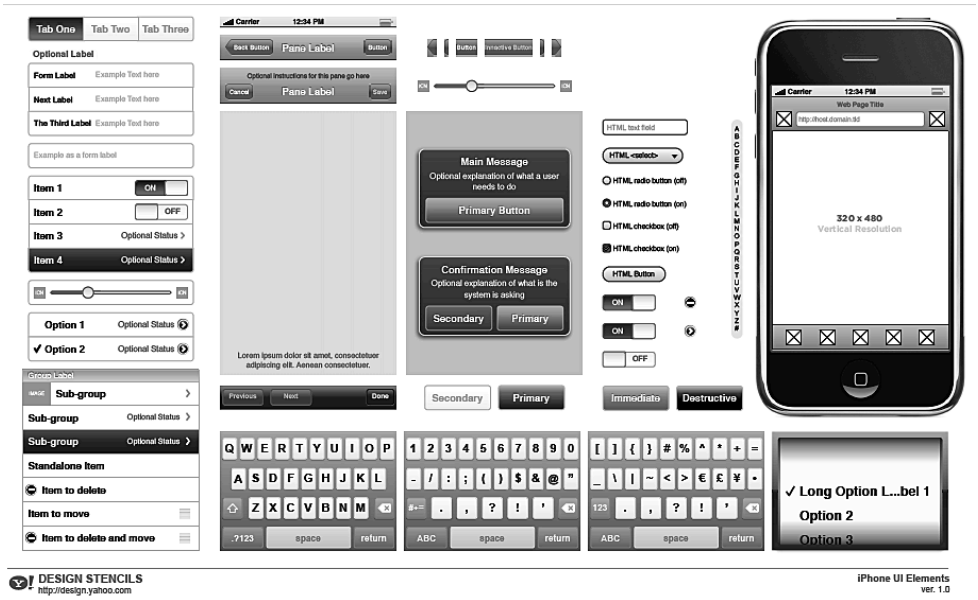


Figura 169. Set di design stencil per iPhone (dal Design Stencil Kit di Yahoo)

[121] Dal Design Stencil Kit versione 1.0 di Yahoo, in http://developer.yahoo.com/ypatterns/wireframes/ .

Prototipi intermedi

I prototipi iniziali, come abbiamo visto, sono spesso usa-e-getta: si costruiscono con le tecnologie più semplici, allo scopo di avere dei rapidi feedback sulle idee iniziali della progettazione. Quando il design concept è definito, potrà iniziare la realizzazione effettiva del sistema, attraverso un numero adeguato d'iterazioni. Da questo momento in poi, almeno per quanto riguarda i prodotti software, si cercherà di sviluppare i prototipi utilizzando le tecnologie finali. In questo modo, se tutto procede per il meglio, il sistema evolve per ampliamenti successivi e per modesti rifacimenti a partire da una base di codice iniziale.

Le strategie che guidano il processo dovranno essere definite di volta in volta, a seconda del particolare tipo di sistema. Per esempio, nel caso dei siti web, la metodologia già citata (a pag. 136 e a pag. 193) prevede due prototipi intermedi: il *prototipo di comunicazione* (che realizza, in una forma sostanzialmente finale, la grafica del sito, che tuttavia è ancora vuoto di contenuti) e il *prototipo funzionale* (che ne realizza le funzioni interattive). La Figura 170 mostra un esempio di prototipo di comunicazione: come si vede, la "cornice" del sito, con intestazioni, grafica, menu, è completa, ma le pagine sono ancora prive di contenuti.



Figura 170. Il prototipo di comunicazione di un sito web (http://www.disco.unimib.it , 2004)

I prototipi intermedi permettono di provare specifici aspetti del prodotto, ma non ancora le sue funzioni complessive, che potranno essere esercitate soltanto alla fine del processo.

Prototipi finali

Se il processo è stato condotto bene, nelle fasi finali potremo avere una ragionevole certezza che le prove d'uso condotte sui diversi prototipi hanno guidato la progettazione in modo corretto. Nelle prove d'uso dei prototipi finali non dovrebbero quindi esserci troppe sorprese. Le prove finali però sono molto importanti, perché è solo quando il sistema è sostanzialmente finito che si potranno condurre prove complete per i compiti (veri e non simulati) per i quali è stato costruito.

Pensiamo ancora una volta, per fissare le idee, al caso di un sito web (per esempio di e-commerce). Il prototipo finale è, in sostanza, il sito completo, con i suoi contenuti definitivi, le funzionalità collaudate, e con una base di dati reali o, comunque, significativi. Su questo sito si potranno effettuare prove sofisticate, in scenari d'uso completi e di notevole complessità. Alle tecniche utilizzabili per condurre questi test di usabilità è dedicato il capitolo 14, al quale rimandiamo il lettore per ogni dettaglio.

Ripasso ed esercizi

1. Che cos'è un prototipo e qual è il suo ruolo nella progettazione human-centred?
2. Che cosa s'intende per prototipo ad alta o a bassa fedeltà?
3. Spiega le differenze fra prototipo usa e getta e prototipo evolutivo, e fra prototipo orizzontale e verticale.
4. Costruisci lo statechart che descrive l'interazione con l'ascensore di un palazzo uffici (separatamente per i comandi esterni e interni all'ascensore).
5. Descrivi l'utilizzo dei prototipi di carta e descrivine i vantaggi.
6. Realizza il prototipo di carta dei comandi esterni e interni dell'ascensore di cui all'esercizio precedente, e verificane l'usabilità con una prova d'uso con due utenti.
7. In che cosa consiste la tecnica del mago di Oz? In quali casi è utile?
8. Che cosa sono e a che cosa servono i prototipi wire-frame nella progettazione di siti web?
9. Realizza con PowerPoint (o strumento analogo) un prototipo navigabile dei comandi dell'ascensore di cui all'esercizio precedente, e provalo con due utenti, diversi dai precedenti.
10. Spiega vantaggi e svantaggi nell'uso di PowerPoint come strumento di prototipazione nella progettazione di un sistema interattivo.
11. Che cosa sono i design stencil?

Approfondimenti e ricerche

1. Cerca in rete e sperimenta un programma che sia adatto alla costruzione e manutenzione di statechart. I programmi per la gestione di diagrammi bidimensionali sono numerosi, a partire da Visio della Microsoft e dai suoi concorrenti gratuiti.

2. Cerca in rete esempi di prototipi di carta, e prototipi ipertestuali, e classificane le diverse tipologie. Suggerimento: cerca su www.youtube.com, www.slideshare.com e su Google-immagini (parole chiave: "paper prototype", "paper prototyping", "lo-fi prototype"). Su www.youtube.com esistono diversi prototipi realizzati da studenti dell'autore di questo libro (parole chiave: "interazione uomo macchina", "polillo", "corso", "prototipo")

3. Cerca in rete qualche collezione di design stencil per comporre rapidamente prototipi accurati dal punto di vista grafico, e verificane le possibilità. Puoi iniziare, per esempio, dal Design Stencil Kit di Yahoo, in http://developer.yahoo.com/ypatterns/wireframes/.

4. Esistono numerosi tool per la prototipazione dell'interfaccia utente di sistemi interattivi. Cerca in rete qualche esempio e sperimentane l'uso, valutandone vantaggi e svantaggi rispetto ai programmi come PowerPoint, discussi in questo capitolo. Per esempio, puoi iniziare da http://uidesign-usability.blogspot.com/2007/03/top-10-simulation-tools-for-ui.html.

10. Principi e linee guida

Sintesi del capitolo

Questo capitolo descrive i principi che il progettista dovrebbe seguire nella progettazione di sistemi interattivi usabili. Dopo un chiarimento su che cosa si intende per principi, linee guida, standard e standard di progetto, vengono richiamati brevemente gli standard elaborati dal comitato tecnico dell'ISO che si occupa degli standard relativi alla *Ergonomics of human-system interaction*. Fra questi, riveste notevole importanza il documento ISO 9241-110, che definisce sette principi molto generali, e una serie di raccomandazioni dirette al progettista di sistemi interattivi. Il cuore del capitolo è dedicato alla descrizione di questi principi e delle relative raccomandazioni, con l'ausilio di numerosi esempi.

Principii, linee guida, regole di progetto, standard

Per aiutare il progettista di sistemi interattivi usabili, è utile fornirgli delle indicazioni che si siano dimostrate valide in progetti simili al suo. Alcune saranno di tipo positivo: "per ottenere questo risultato, puoi adottare questa soluzione". Altre di tipo negativo: "in questa situazione, evita di fare questo". Queste indicazioni possono essere più o meno *generali* (alcune sono applicabili in ogni situazione, altre in casi specifici) o più o meno *coercitive* (alcune sono semplici suggerimenti, che possono essere seguiti a discrezione del progettista, altre sono vincolanti, e devono essere seguite obbligatoriamente).

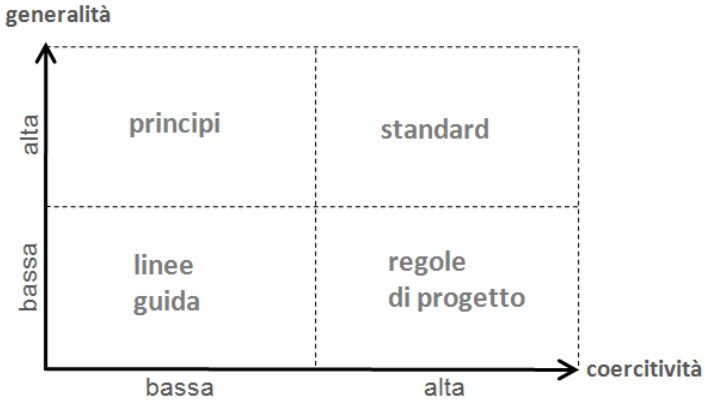


Figura 171. Classificazione delle indicazioni per il progettista

Queste indicazioni possono essere classificate in quattro grandi categorie, in funzione del loro livello di generalità e coercitività, come indicato in Figura 171:

- *Regole di progetto*

  Sono le regole che *devono* essere applicate nell'ambito di uno specifico progetto. Hanno bassa generalità (possono essere anche molto specifiche), ma alta coercitività: sono imposte dal committente, e vincolanti per il progettista. Di solito sono regole piuttosto dettagliate, che specificano le modalità d'interazione con un certo sistema, per esempio per renderlo consistente con altri sistemi della stessa organizzazione. Esempi tipici sono le regole che definiscono l'apparenza grafica dell'interfaccia con l'utente: quali font devono essere usati, quali formati sono ammessi per i campi di input usati nei dialoghi, quali forme, dimensioni e colori dei pulsanti, e così via.
- *Standard*

  Sono norme di tipo generale, emesse da organismi internazionali, che definiscono le regole da applicare nei progetti di determinate classi di sistemi. Sono vincolanti per tutti i progetti che dichiarano di essere conformi allo standard. Sono emessi da enti di standardizzazione, come per esempio l'ISO (International Standard Organization), o da altri enti indipendenti, che hanno l'obiettivo di definire delle raccomandazioni da seguire per certe categorie di sistemi. Un esempio è il W3C (World Wide Web Consortium), le cui raccomandazioni costituiscono degli *standard di fatto*

per le tecnologie del Web. La conformità a uno standard deve essere valutabile in modo preciso, e quindi gli standard devono essere formulati con estrema cura, per evitare ogni ambiguità. Ovviamente, dovendo essere applicabili a intere classi di sistemi, gli standard sono notevolmente meno specifici delle regole di progetto.

- *Linee guida*

  Sono delle raccomandazioni per il design dell'interazione di specifiche classi di sistemi, espresse in modo più o meno generale, a seconda dei casi. Sono spesso corredate di esempi e motivazioni. Non sono vincolanti, sta al progettista decidere sull'opportunità di applicarle caso per caso. Particolarmente importanti sono le linee guida per l'interfaccia utente di applicazioni relative a una specifica piattaforma, per esempio Apple[122], Windows[123], Gnome, e così via. Il loro scopo principale è quello di garantire un'elevata uniformità della user experience di tutte le applicazioni sviluppate per la specifica piattaforma. Altre sono indipendenti dalla piattaforma (*cross-platform guidelines*). Le linee guida in circolazione sono numerose, e sono spesso raccolte in documenti voluminosi, contenenti diecine o centinaia d'indicazioni. Per esempio, le linee guida per il design di siti web elaborate dal Department of Health and Human Services statunitense contiene oltre 200 linee guida, classificate in base alla loro rilevanza e al livello di evidenza scientifica.[124]

- *Principi*

  Sono indicazioni generali per la progettazione di interfacce utente usabili, basate su evidenze scientifiche o sul generale consenso. Derivano dalla conoscenza degli aspetti fisiologici, psicologici e sociali degli utenti e dall'esperienza accumulata nella pratica della progettazione dei sistemi usabili. Sono indipendenti dalla tecnologia, e sono espressi spesso in forma molto generale. Particolarmente importanti, per la loro generalità, sono i principi del dialogo con i sistemi interattivi descritti nel documento ISO 9241-Part 110, che tratteremo diffusamente nel seguito di questo capitolo.

Gli standard della human-system interaction

L'ente principale responsabile della preparazione degli standard è l'ISO (International Standard Organization, www.iso.org), un'associazione non governativa di enti nazionali di standardizzazione di oltre 160 paesi, con sede a Ginevra. I prodotti principali dell'ISO sono dei documenti chiamati "International Standard" (IS), che vengono redatti da appositi comitati tecnici (*Technical Commitee, TC*) rappresentativi degli interessi delle parti coinvolte (produttori, venditori, utenti, organizzazioni dei consumatori, laboratori di prova, governi, professionisti, organizzazioni di ricerca, ecc.).

Il processo di definizione di uno standard internazionale è piuttosto lungo e complesso, per garantire la massima trasparenza del processo, l'apertura a tutti i contributi, la coerenza tecnica all'interno del sistema degli standard e, soprattutto, il più ampio accordo fra tutti gli enti coinvolti, rappresentativi di interessi diversi. Uno standard internazionale dovrebbe, infatti, rappresentare le conoscenze e le pratiche sulle quali si raccoglie il massimo consenso fra gli esperti dei vari paesi. Pertanto, esso viene elaborato in diverse fasi, attraverso un processo formalizzato che può durare anche diversi anni: a partire da una prima proposta, vengono realizzate numerose bozze, sottoposte al commento e all'approvazione delle varie parti coinvolte, fino al Draft International Standard (DIS) che deve essere approvato per votazione dal 75% dei membri con diritto di voto.

Gli standard che interessano l'ingegneria dell'usabilità fanno capo al Technical Committee TC 159 – Ergonomics, a sua volta suddiviso in quattro sotto-comitati (*Sub-Committee, SC*):

---

[122] *Apple Human Interface Guidelines*, reperibili in rete all'indirizzo http://developer.apple.com/documentation/UserExperience/Conceptual/AppleHIGuidelines.

[123] *Windows User Experience Interaction Guidelines,* reperibili in rete all'indirizzo http://msdn.microsoft.com/en-us/library/aa511258.aspx.

[124] U.S. Department of Health and Human Services & U.S. General Services Administration, *Research-based Web Design and Usability Guidelines,* scaricabile dalla rete all'indirizzo http://usability.gov/guidelines. Si tratta di un documento molto importante, realizzato con la collaborazione di un ampio gruppo di esperti, a partire dal 2000 e più volte aggiornato. Ogni linea guida è descritta da una scheda così composta: *Guideline*, *Comments*, *Sources*, *Example*, *Relative importance*, *Strenght of evidence*. Quest'ultima assume i seguenti cinque valori: 5: *strong research support*; 4: *moderate research support*; 3*: weak research support*; 2: strong *expert opinion support*; 1: *weak expert opinon support*. La letteratura a supporto di ciascuna linea guida è citata nella sezione *Sources*.

TC 159/SC 1 - General ergonomics principles;
TC 159/SC 3 - Anthropometry and biomechanics;
TC 159/SC 4 - Ergonomics of human-system interaction;
TC 159/SC 5 - Ergonomics of the physical environment.

Nell'ambito di questo libro, il sotto-comitato di maggiore interesse è il TC 159/SC 4, che dichiara la seguente missione:

> *Standardizzazione ergonomica della interazione fra i sistemi (spesso basati su computer) e le persone che li progettano, fabbricano, usano e mantengono. Le aree di standardizzazione comprendono l'ergonomia dell'hardware (inclusi gli strumenti di input, i monitor e gli apparati interattivi), l'ergonomia del software (inclusa la progettazione del dialogo e l'interaction design) e i metodi e processi dello human-centred design (inclusa l'ingegneria dell'usabilità e i metodi di progettazione partecipativa).*[125]

Gli standard elaborati dal TC 159/SC 4 sono numerosi. La rapida evoluzione delle tecnologie d'interazione e la maturazione delle esperienze nella costruzione di sistemi interattivi fa sì che i diversi standard debbano essere periodicamente rivisti e aggiornati. A volte i documenti già emessi vengono riscritti ex novo, o integrati con nuovi standard, di argomento più specifico. Queste attività di evoluzione e aggiornamento, unite alla lentezza intrinseca del processo di standardizzazione, creano inevitabilmente una situazione piuttosto complicata, per cui non è facile avere il quadro completo della situazione. Ulteriori difficoltà derivano dal fatto che i documenti ISO non sono in genere liberamente accessibili, ma devono essere acquistati singolarmente, a prezzi molto elevati.

I principali standard prodotti dal TC 159/SC 4 sono i seguenti:

- ISO 13407

  Si intitola *Human-centred design processes for interactive systems*, ed ha lo scopo di aiutare coloro che hanno la responsabilità di gestire i processi di progettazione di hardware e software a pianificare in modo adeguato le attività di progettazione human-centred. Ne abbiamo parlato diffusamente nel capitolo 6, e in particolare a pag. 132 e seguenti.

- ISO 9241

  Si tratta dello standard principale relativo alla human-computer interaction. È molto ampio, ed è composto da numerosi documenti separati, in evoluzione da una ventina d'anni.

  Inizialmente, questo standard trattava essenzialmente gli aspetti ergonomici dei terminali video utilizzati per il lavoro di ufficio, e aveva infatti il titolo di *Ergonomic requirements for office work with visual display terminals (VDTs)*. Col tempo, i suoi obiettivi si sono ampliati, e ora tratta le problematiche di usabilità dei sistemi interattivi in generale. Pertanto, è in corso un processo di revisione dell'intero standard, rinominato di recente, più genericamente, *Ergonomics of Human System Interaction*. In questa revisione, lo schema di numerazione dei documenti dell'ISO 9241 è stato ridefinito. Nella nuova numerazione, i documenti sono organizzati in serie tematiche:

  Serie 100: Software ergonomics;
  Serie 200: Human system interaction processes;
  Serie 300: Displays and display related hardware;
  Serie 400: Physical input devices - ergonomics principles;
  Serie 500: Workplace ergonomics;
  Serie 600: Environment ergonomics;
  Serie 700: Application domains - Control rooms;
  Serie 900: Tactile and haptic interactions.

  Al documento iniziale di ciascuna serie (identificato col numero n00), che costituisce l'introduzione alla serie, segue un numero di documenti variabile in funzione delle esigenze dell'argomento.
  Questo schema sostituisce quello originale, nel quale i documenti erano denominati "Parti" e numerati da 1 a 17. In questa fase di transizione dalla vecchia organizzazione alla nuova, la situazione risulta abbastanza confusa, perché

[125] Da http://www.iso.org, nostra traduzione dall'inglese.

convivono standard con la vecchia e con la nuova numerazione. I documenti pubblicati alla data di stesura di queste pagine (compresi quelli nello stato di DIS) sono elencati in Figura 172. A questi si aggiungono numerosi altri documenti in bozza. La situazione aggiornata si trova sul sito dell'ISO, nella pagina del TC/159 SC 4.[126]

- ISO 14915
  Si intitola *Software ergonomics for multimedia user-interfaces,* ed è composto da tre documenti: *Part 1: Design principles and framework*; *Part 2: Multimedia navigation and control*; *Part 3: Media selection and combination.*
  Questo standard descrive principi e raccomandazioni per la progettazione di sistemi multimediali. In particolare, tratta l'interfaccia utente di applicazioni che incorporano, integrano e sincronizzano media differenti, statici (testo, grafica o immagini) o dinamici (audio, animazioni, video o media correlati ad altre modalità sensoriali).

Oltre agli International Standard, l'ISO può produrre altri tipi di documenti, con un livello di consenso inferiore: ISO Technical Specification (ISO/TS), ISO Public Available Specification (ISO/PAS) e ISO Technical Report (ISO/TR). In particolare, il TC/159 SC 4 ha prodotto questi documenti aggiuntivi:

- ISO/TR 16982: *Ergonomics of human-systems interaction – Usability methods supporting human-centred design;*
- ISO/PAS 18152: *Ergonomics of human-systems interaction – Specification for the process assessment of human-system issues;*
- ISO/TR 18529: *Human-centred lifecycle process descriptions.*

Part 1: General introduction
Part 2: Guidance on task requirements
Part 4: Keyboard requirements
Part 5: Workstation layout and postural requirements
Part 6: Guidance on the work environment
Part 9: Requirements for non-keyboard input devices
Part 11: Guidance on usability
Part 12: Presentation of information
Part 13: User guidance
Part 14: Menu dialogues
Part 15: Command dialogues
Part 16: Direct manipulation dialogues
Part 17: Form filling dialogues
Part 20: Accessibility guidelines for ICT equipment and services
Part 100: Introduction to standards related to software ergonomics
Part 110: Dialogue principles
Part 129: (DIS) Guidance on software individualization
Part 151: Guidance on World Wide Web user interfaces
Part 171: Guidance on software accessibility
Part 210: Human-centred design for interactive systems
Part 300: Introduction to electronic visual display requirements
Part 302: Terminology for electronic visual displays
Part 303: Requirements for electronic visual displays
Part 304: User performance test methods for electronic visual displays
Part 305: Optical laboratory test methods for electronic visual displays
Part 306: Field assessment methods for electronic visual displays
Part 307: Analysis and compliance test methods for electronic visual displays
Part 308: Surface-conduction electron-emitter displays (SED)
Part 309: Organic light-emitting diode (OLED) displays
Part 400: Principles and requirements for physical input devices
Part 410: Design criteria for physical input devices
Part 420: (DIS) Selection procedures for physical input devices
Part 910: (DIS) Framework for tactile and haptic interaction
Part 920: Guidance on tactile and haptic interactions

Figura 172. Documenti dell'ISO 9241 pubblicati al marzo 2010

[126] http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_tc_browse.htm?commid=53372

Come si intuisce da questo sommario elenco, gli standard emessi dal TC/159 SC 4 sono di tipologie diverse. Alcuni sono *standard di processo*: definiscono, cioè, le caratteristiche di determinati processi di lavoro (per esempio, nel caso dell'ISO 13407, il processo di progettazione dei sistemi interattivi). Altri sono *standard di prodotto*: specificano, cioè, le caratteristiche che devono possedere determinate categorie di prodotti. Appartengono a questa categoria vari documenti dell'ISO 9241, come per esempio la Parte 4, che stabilisce i requisiti ergonomici delle tastiere. Molti documenti, inoltre, hanno lo scopo di *definire la terminologia* da usarsi in un determinato ambito. Anche questo è un compito importante dei comitati di standardizzazione, che permette ai vari operatori di utilizzare un linguaggio comune in modo non ambiguo. Il compito può sembrare banale, ma è reso molto complicato dalla necessità di mantenere una coerenza terminologica fra i numerosi documenti, che evolvono separatamente lungo l'arco di molti anni.[127]

I principi del dialogo secondo la ISO 9241-110

Fra i numerosi documenti che compongono l'ISO 9241, vogliamo qui occuparci della Parte 110, *Dialogue principles*. Si tratta di uno documento breve, ma molto importante dal punto di vista concettuale. Esso descrive sette *principi del dialogo*, ovvero sette caratteristiche che ogni dialogo fra un utente e un sistema interattivo dovrebbe possedere. In questo documento, il termine *dialogo* viene usato, come abbiamo già visto, nel senso ampio di "l'interazione fra un utente e un sistema interattivo, intesa come sequenza di azioni dell'utente (input) e di risposte del sistema (output) per raggiungere un obiettivo". Le azioni dell'utente includono quindi l'immissione di dati, ma anche tutte le azioni di navigazione e di controllo del sistema. Anche il termine *sistema interattivo* è usato in modo generale, come "una combinazione di elementi hardware e software che ricevono degli input da un utente umano, e gli comunicano degli output, allo scopo di supportare l'esecuzione di un compito".

I sette principi del dialogo sono i seguenti[128].

- *Adeguatezza al compito (suitability for the task)*

  Un sistema interattivo è adeguato al compito se supporta l'utente nel completamento del compito, cioè quando la funzionalità del sistema e il dialogo sono basati sulle caratteristiche del compito, piuttosto che sulla tecnologia scelta per eseguirlo.
- *Auto-descrizione (self-descriptiveness)*

  Un dialogo è auto-descrittivo se agli utenti risulta evidente, in ogni momento, in che dialogo si trovano, a che punto si trovano all'interno del dialogo, quali azioni possono compiere e come queste possono essere eseguite.
- *Conformità alle aspettative dell'utente* (*conformity with user expectations*)

  Un dialogo è conforme alle aspettative dell'utente se corrisponde alle necessità dell'utente, prevedibili in base al contesto e a convenzioni comunemente accettate.
- *Adeguatezza all'apprendimento* (*suitability for learning*)

  Un dialogo è adeguato all'apprendimento se supporta e guida l'utente nell'apprendimento del sistema.
- *Controllabilità* (*controllability*)

  Un dialogo è controllabile se l'utente è in grado di iniziare e tenere sotto controllo la direzione e i tempi dell'interazione fino al raggiungimento dell'obiettivo.
- *Tolleranza verso gli errori* (*error-tolerance*).

  Un dialogo tollera gli errori se, nonostante evidenti errori negli input, i risultati desiderati possono essere ottenuti senza o con minime azioni correttive. La tolleranza per gli errori si ottiene attraverso a)- il controllo degli errori (controllo dei danni); b)- la correzione degli errori e c)- la gestione degli errori, per fronteggiare gli errori occorsi.

---

[127] Questo obiettivo non sempre viene raggiunto, come si può vedere, per esempio, dalle diverse definizioni del concetto stesso di usabilità che sono state date, negli anni, nei diversi documenti ISO.

[128] Il documento è in lingua inglese, la formulazione dei principi è in nostra traduzione. Ci riferiremo alla versione ISO 9241-110:2006, che sostituisce la precedente ISO 9241-10:1996. Rispetto alla versione del 1996, la versione del 2006 non contiene modifiche sostanziali: i principi sono gli stessi, ma le definizioni sono più precise e articolate. Sono state inoltre aggiunte raccomandazioni ed esempi, e indicazioni molto utili sugli altri standard correlati.

- *Adeguatezza all'individualizzazione* (*suitability for individualization*)
  Un dialogo è adeguato all'individualizzazione se l'utente può modificare l'interazione e la presentazione dell'informazione per adattarle alle proprie necessità e capacità individuali.

Questi principi sintetizzano, secondo il punto di vista adottato nello standard, gli elementi chiave che determinano la usabilità di un sistema interattivo (Figura 173). Il documento, tuttavia, non intende precludere la possibilità di altre formulazioni, affermando esplicitamente che "ci possono essere modi diversi di identificare gli aspetti chiave, che conducono a un diverso insieme di principi".

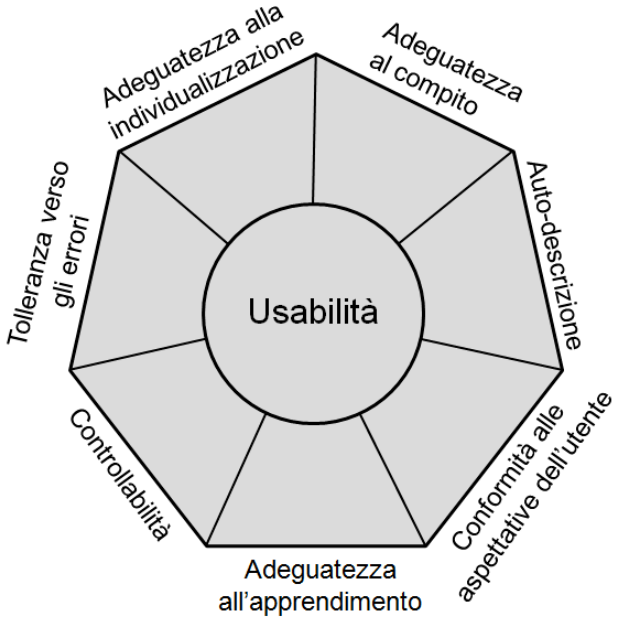


Figura 173. I sette principi del dialogo secondo l'ISO 9241

I principi elencati non sono fra loro indipendenti e possono esserci delle sovrapposizioni. In determinate situazioni essi possono essere in conflitto, e sarà quindi compito del progettista individuare il compromesso più opportuno in relazione alle specifiche caratteristiche del sistema in corso di progettazione, per ottimizzarne l'usabilità. Questo può non essere facile, perché la priorità da assegnare ai diversi principi può variare a seconda della specifica applicazione, delle caratteristiche degli utenti e dello stile di dialogo utilizzato. In effetti, lo standard afferma esplicitamente che l'ordine in cui i principi sono elencati non ha alcun particolare significato.

I principi dell'ISO 9241-110 sono importanti non solo dal punto di vista teorico-concettuale, ma anche dal punto di vista pratico. In sostanza, essi definiscono un *modello di qualità*[129] che può essere utilizzato per valutare l'usabilità di un sistema, o per confrontare l'usabilità di due sistemi simili. A questo scopo, occorrerà esaminare ciascun sistema e attribuire un "voto" al grado di applicazione di ogni principio. Sarà poi possibile visualizzare i voti ottenuti in un diagramma a stella, come nella Figura 174 (in cui si è usata una scala da 0 - voto minimo - a 4 - voto massimo) ottenendo il *profilo di qualità* del sistema. Nel diagramma di sinistra vediamo il profilo di un sistema che ha ricevuto il massimo dei voti nell'auto-descrizione e nell'adeguatezza all'apprendimento, ma un voto pessimo nell'adeguatezza all'individualizzazione. Nel diagramma di destra sono messi a confronto due sistemi.

[129] Un modello di qualità è un insieme di caratteristiche sulla base delle quali si valuta la qualità di un sistema. Queste caratteristiche vengono scelte in funzione degli obbiettivi del sistema. In questo caso, l'obbiettivo è la usabilità, e le sette caratteristiche sono quelle indicate in Figura 173.

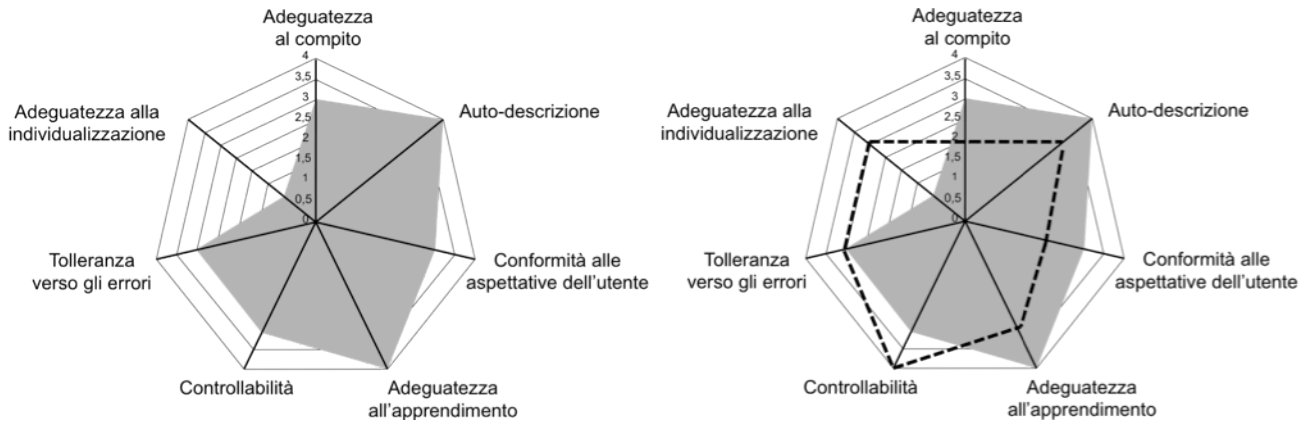


Figura 174. Profilo di qualità di un sistema basato sui sette principi del dialogo dell'ISO 9241 (sinistra) e confronto di due profili di qualità (destra)

Per ciascuno dei sette principi indicati, lo standard ISO 9241-110 fornisce una lista di raccomandazioni che dovrebbero essere tenute in considerazione nella progettazione di un sistema. Questa lista non pretende di essere esaustiva. Inoltre, le raccomandazioni non sono fra loro indipendenti, e ci sono diverse sovrapposizioni. Pur con questi limiti, si tratta d'indicazioni importanti che, se seguite con attenzione, possono evitare i problemi di usabilità più comuni. Nel seguito, per ciascun principio, descriviamo alcune linee guida che possono orientare i progettisti nel design dell'interazione, ispirandoci liberamente alle raccomandazioni dello standard, e integrandole con esempi e indicazioni tratte dalle buone pratiche consolidate nel design dell'interazione. liberamente.

Adeguatezza al compito

In sostanza, questo principio afferma che le funzionalità e le modalità d'interazione del sistema devono essere modellate sulle caratteristiche del compito che l'utente deve compiere con il supporto del sistema, e non sulla tecnologia o delle caratteristiche del sistema stesso. In pratica, il dialogo dovrà essere progettato a partire dai casi d'uso identificati nell'analisi dei requisiti. Per ogni caso d'uso, dovrà permettere all'utente di svolgere i compiti e le azioni necessarie nella sequenza più naturale per lui, e non per il software che gestisce l'interazione. Questa è l'essenza stessa dello user-centred design, e forse non è un caso che lo standard abbia elencato questo principio per primo.

Alcune linee guida coerenti con questo principio generale sono le seguenti.

- Dialologo adeguato al compito

  I passi del dialogo dovrebbero essere adeguati al compito: dovrebbero essere inclusi tutti i passi necessari ed evitati i passi non necessari. In particolare, il dialogo dovrebbe assegnare al sistema tutte quelle operazioni che possono essere automatizzate, senza caricare inutilmente l'utente di compiti che possono essere agevolmente svolti in modo automatico. Le operazioni assegnate all'utente dovrebbero essere da questi eseguibili con facilità, senza richiedere sforzi cognitivi eccessivi o abilità motorie particolari.
- Informazione adeguata al compito

  Questa linea guida è un caso particolare della precedente, e richiede che il sistema presenti all'utente tutte le informazioni utili per lo svolgimento del compito. Ciò può essere fatto in molti modi, ma le soluzioni migliori sono quelle in cui l'informativa all'utente varia in funzione del contesto. In questo modo, si trasmettono all'utente solo le informazioni utili nello svolgimento di un particolare passo del dialogo, e si evita di sovraccaricarlo con informazioni che gli serviranno in momenti diversi, anche se temporalmente vicini.

  Per esempio, un sito di commercio elettronico, in cui il processo di acquisto si sviluppa in più fasi, all'utente vengono fornite di volta in volta soltanto le informazioni necessarie per lo svolgimento della fase corrente. La Figura 175 mostra la prima fase (Scelta del viaggio) dell'acquisto di un biglietto ferroviario sul sito di Trenitalia. Il processo di prenotazione avviene in cinque fasi, e l'applicazione segnala all'utente in quale fasi si trova, evidenziandone graficamente il nome nella parte alta dello schermo. In questa fase all'utente è richiesto, correttamente, di indicare soltanto il treno desiderato, dopodiché potrà procedere alla fase successiva.



Figura 175. Acquisto del biglietto di un treno (www.trenitalia.it, 2009)

È molto importante che vengano sempre tenute in considerazione le limitazioni della memoria umana a breve termine, descritte nel capitolo 4 (pag.91). Nell'esempio di Figura 176, tratto da una vecchia versione di Microsoft Word, il messaggio indica all'utente come può vedere le parti del suo testo che non sono state esaminate dal sistema di controllo ortografico del word processor (text set to no proofing). Il messaggio sovraccarica inutilmente la memoria a breve termine dell'utente. Infatti, per vedere quale testo non è stato esaminato dal controllore ortografico, l'utente dovrebbe memorizzare la lunga sequenza di operazioni necessarie: click Edit/Replace, click More, click Format, click Language, choose (no proofing). Dovrebbe poi premere OK per riuscire ad accedere ai menu di Word e compiere tali operazioni. Ma così facendo il messaggio scompare, ed è probabile che la sequenza di operazioni da svolgere sia stata già parzialmente dimenticata: la sequenza da ricordare è lunga, e per eseguirla l'utente dovrà compiere altre attività cognitive (per esempio, per cercare le voci di menu indicate). Come già sappiamo, questo può facilmente mettere in crisi la nostra memoria a breve termine.

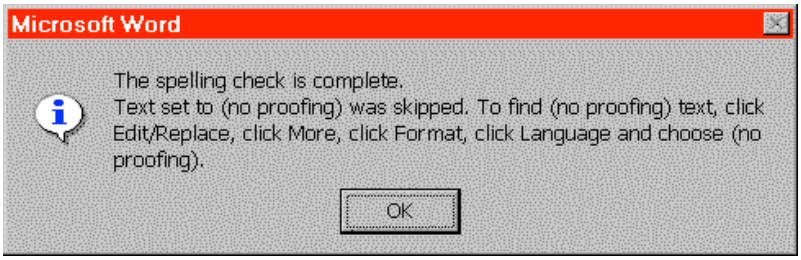


Figura 176. Sovraccarico della memoria a breve termine (da Microsoft Word 1997)

Altri esempi analoghi di sovraccarico della memoria a breve termine sono mostrati nel capitolo 4, e nel capitolo 11, Figura 210 e Figura 214. Questi esempi andrebbero considerati con attenzione, poiché errori di questo tipo vengono commessi di frequente dai progettisti meno esperti.

- Dialogo essenziale

  Il sistema dovrebbe evitare di presentare all'utente informazioni ridondanti. La comunicazione dovrebbe essere breve, diretta ed essenziale. Se l'informazione è fornita attraverso testi scritti, questi dovrebbero essere redatti con i criteri di massima comprensibilità, secondo i principi del *plain language* che tratteremo nel capitolo 13. Occorre sempre evitare di duplicare l'informazione. Ogni duplicazione genera dubbi nell'utente e ne distoglie l'attenzione dal compito principale. Per esempio, la Figura 177 mostra un errore frequente nei siti web: la duplicazione del menu principale nella home page. In questo caso, nella home page e in tutte le altre pagine del sito è visibile un menu verticale di cinque voci, in alto a sinistra. A scopo puramente decorativo, nella sola home page questo menu è

duplicato nella parte centrale della pagina, e ridisegnato in forme tondeggianti. I due menu presenti in home page sono equivalenti: ogni coppia di voci omonime (per esempio, La Compagnie) porta alla stessa pagina del sito. Ma l'utente non lo sa per certo, e avrà il dubbio che le due voci conducano a pagine diverse. Per risolvere questo dubbio, sarà costretto a provare a cliccare i diversi link e confrontare i risultati.



Figura 177. Ridondanza dei menu nella home page di un sito (http://www.bruno-verdi.com)

- Dispositivi di input e output adeguati al compito

  La tecnologia mette a disposizione molti possibili dispositivi di input e di output per realizzare il dialogo con l'utente: tastiera, mouse, schermi tattili, voce, video, audio, stampa, ecc. Quelli utilizzati dovrebbero essere scelti in funzione del compito specifico, e non viceversa. A volte il compito può richiedere, per una migliore efficienza, l'utilizzo contemporaneo di più dispositivi (*multi-modalità*). Per esempio, in un'applicazione di grafica, l'utente potrebbe utilizzare la mano destra per disegnare, muovendo lo stilo sulla tavoletta grafica, mentre con la mano sinistra potrebbe attivare i diversi comandi premendo i vari tasti funzione (o viceversa, se mancino). In questo caso, la scelta delle combinazioni di tasti dovrà essere tale da agevolare questa modalità di interazione.

- Formati di input e output adeguati al compito

  I formati dei dati di input e output dovrebbero essere adeguati al compito. Per esempio, un programma finanziario dovrebbe accettare valori con non più di due decimali, quando essi siano espressi in Euro. Un altro esempio significativo è costituito dai numeri di telefono. Abbastanza spesso, i siti web degli alberghi statunitensi forniscono soltanto il numero di telefono gratuito (con prefisso 800). Ma questo è utilizzabile soltanto dagli Stati Uniti: chi volesse prenotare telefonicamente una stanza dall'Europa non lo può fare.

- Default tipici

  Una corretta impostazione dei valori di default può semplificare molto il dialogo. Essi dovrebbero essere impostati in modo da riflettere le scelte più comuni. Per esempio, in un distributore di biglietti ferroviari, la stazione di partenza di default potrebbe essere quella in cui ci si trova. In questo modo si semplifica il compito dell'utente, che nella maggior parte dei casi potrà semplicemente specificare solo la stazione di destinazione. L'usabilità e l'efficienza migliorano, e il distributore sarà in grado di servire, nello stesso tempo, un maggior numero di utenti. Lo stesso accorgimento potrà essere usato anche nei sistemi di prenotazione o di calcolo d'itinerari via Web,

impostando il valore di default del punto di partenza sulla base del domicilio fornito dall'utente in fase di registrazione al sito.

Ancora, quando in Photoshop si crea un nuovo documento, le sue dimensioni vengono automaticamente impostate alla dimensione dell'elemento che è stato oggetto dell'ultima operazione di copia, nell'assunzione che l'utente desideri poi "incollare" tale elemento nel documento appena creato, per modificarlo.

Un'adeguata impostazione dei valori di default è particolarmente importante in quelle funzionalità che richiedono all'utente di specificare molti parametri, come per esempio nei programmi di grafica più sofisticati per le operazioni di stampa o di salvataggio di file nei vari formati. In questo caso l'utente meno esperto potrebbe affidarsi ai valori di default anche quando non conosce il significato di tutti i parametri, con la garanzia che l'operazione richiesta verrà comunque portata a termine, almeno nelle situazioni più comuni, in modo corretto.

- Compatibilità con i documenti

  Questa indicazione è particolarmente importante nel progetto di sistemi informativi aziendali. Essa si applica quando un documento cartaceo è la fonte dei dati che devono essere immessi nel sistema, o quando né è la destinazione. In entrambi i casi è conveniente che il layout delle informazioni sul video e sul documento siano congruenti. Nel primo caso, per rendere più scorrevoli le operazioni di immissione dei dati, senza che ciò richieda all'operatore di ricercare ogni volta il dato richiesto all'interno del documento. Nel secondo caso, per evitare all'utente la complessità di gestire gli stessi dati visualizzati in modo diverso.

Auto-descrizione

In sostanza, questo principio richiede che il sistema comunichi all'utente, in ogni momento, che cosa egli possa fare e come, e che cosa sta accadendo. Ricordando il modello di Norman descritto nel capitolo 3, ciò ha lo scopo di ridurre il golfo dell'esecuzione e il golfo della valutazione. Nella maggior parte dei casi, questa comunicazione utilizza il canale visivo, con informazioni testuali o grafiche visualizzate su un monitor. Possono però essere utilizzati anche altri canali, per esempio quello auditivo, con l'uso di messaggi acustici di vario tipo.

Alcune linee guida da seguire nella progettazione di un dialogo autodescrittivo sono le seguenti.

- Guida per l'utente

  Ogni passo del dialogo dovrebbe fornire all'utente tutte le informazioni utili per proseguire. Le informazioni sono sostanzialmente di tre tipi: istruzioni operative ("che cosa puoi fare ora"), informazioni sullo stato del sistema ("ora sto facendo questo") e informazioni di feedback ("ciò che hai fatto ha avuto questo effetto"). La guida all'utente non dovrebbe, comunque essere eccessivamente vincolante: egli dovrebbe essere sempre libero di modificare lo svolgimento del dialogo secondo le sue esigenze, in modo flessibile. Per esempio, nel processo di acquisto del biglietto ferroviario di Figura 175, l'utente dovrebbe essere sempre in grado di tornare alle fasi precedenti, per modificare liberamente alcune scelte già effettuate, senza dover riprendere l'intero dialogo dall'inizio. Questa flessibilità è così importante ai fini dell'usabilità di un sistema che l'ISO 9241 la sancisce dedicandole uno dei sette principi di base (controllabilità).
- Interazione evidente

  Questa linea guida è legata al concetto di affordance introdotto a pag. 63. Raccomanda, in sostanza, che i dialoghi siano progettati in modo che l'interazione con il sistema risulti evidente.

  La Figura 178 mostra la home page di un vecchio sito del Mulino Bianco, che viola questa indicazione. La parte centrale della pagina, destinata a contenere il menu principale, è muta, ma passando il mouse su ciascuna delle tre aree tonde attorno al logo centrale appaiono le voci di un menu. Si scopre così che ciascuna area tonda corrisponde a un menu: per conoscere lo scopo del sito è necessario "esplorare" la pagina col mouse. La figura mostra ciò che appare quando il mouse tocca l'area di sinistra. La home page "a riposo" non mostra alcun menu, né è possibile vedere i tre menu contemporaneamente. Un analogo problema di usabilità esiste nel sito di J.K.Rowling, già discusso a pag. 174 (Figura 145), e nel sito di Figura 222 a pag. 253.



Figura 178. Home page del sito del Mulino Bianco (2003)

In molti casi questo si può ottenere adottando modalità operative già note all'utente in altri contesti. Per esempio, da molti anni i player di musica digitale utilizzano un'interfaccia molto simile, mutuata dagli apparati dell'elettronica di consumo, come negli esempi di Figura 179.

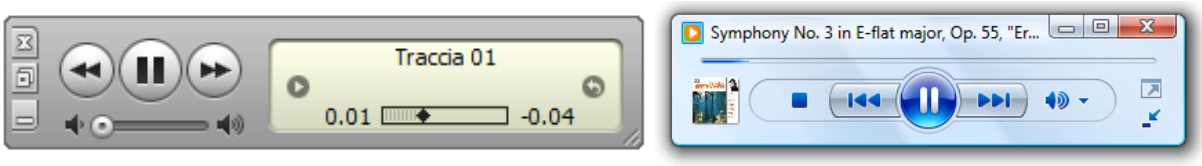


Figura 179. Il mini-player di iTunes della Apple (sinistra) e di Windows Media Player della Microsoft (destra)

Alcune modalità d'interazione, di per sé non ovvie, possono essere ritenute ormai acquisite, considerando la loro larga diffusione. Per esempio, da tempo ci siamo abituati alla classica interfaccia dei cellulari Nokia, in cui i due tasti situati subito sotto lo schermo vengono utilizzati per confermare una delle due opzioni presentate in corrispondenza. Così, il prototipo di Figura 180 può ragionevolmente ritenersi di utilizzo evidente.



Figura 180. Scommesse da cellulare (prototipo didattico)

- Descrizione dell'input atteso

  Anche questa raccomandazione è un caso particolare della prima. Ogni volta che il sistema richiede un input all'utente, dovrebbe fornirgli informazioni adeguate su ciò che si aspetta, e sul suo formato. Quindi, sono da evitare dialoghi che contengono campi liberi, come il seguente:

  DATA DI NASCITA: _____

  che andrebbero sostituiti con input guidati, per esempio:

  DATA DI NASCITA: __/__/____ (gg/mm/aaaa)

- Stato visibile

  Questa è una delle raccomandazioni più importanti. L'utente dovrebbe essere sempre informato sullo stato del sistema, sia quando questo è in attesa di input, sia quando ha elaborato l'input che gli è stato appena fornito. Se l'utente non conosce lo stato in cui si trova il sistema, non può prevedere come risponderà alle sue azioni. Un sistema "muto", che non rende chiaramente visibile il suo stato, genera incertezza e stress, anche quando va tutto bene. La regola del "silenzio assenso" non dovrebbe essere mai applicata nei sistemi usabili. Questo vale anche quando dialoghiamo con un interlocutore umano: se non ne conosciamo lo stato d'animo non sappiamo come comportarci. Di questo parleremo ancora nel capitolo 11, a proposito delle interfacce modali (pag.239).

- Formati descritti

  Il sistema dovrebbe fornire all'utente ogni informazione sui formati e sulle unità di misura utilizzati.

- Manualistica minima

  Durante l'interazione, si dovrebbe minimizzare la necessità di consultare manuali d'uso o altre informazioni presenti fuori dal sistema. Tutta l'informazione necessaria dovrebbe essere consultabile in linea, senza uscire dal sistema, preferibilmente con sistemi di help evoluti di natura contestuale. Ricordiamo, a questo proposito, le considerazioni sui manuali d'uso già fatte a pag.71.

Conformità alle aspettative

Questo principio afferma che il dialogo deve essere conforme a ciò che l'utente si aspetta, in relazione allo specifico contesto d'uso del sistema, e alle convenzioni comunemente adottate. Si tratta quindi di un principio di natura molto generale, che può essere applicato a molte situazioni diverse. Le implicazioni sono numerose: vediamone alcune.

- Linguaggio familiare

  Il sistema dovrebbe utilizzare un linguaggio che l'utente conosce bene. Questa raccomandazione è molto importante, perché la maggior parte dei dialoghi con i sistemi interattivi avviene attraverso il linguaggio. Un suo uso improprio può quindi compromettere gravemente l'usabilità. Tutti gli aspetti del linguaggio usato dovrebbero essere considerati con grande attenzione: vocabolario, sintassi e semantica. Per i sistemi che si rivolgono a utenze internazionali è particolarmente importante la qualità delle traduzioni nelle diverse lingue. Alla comprensibilità dei testi dedicheremo alcune pagine del capitolo 13.

- Aderenza alle convenzioni

  Il dialogo dovrebbe seguire le convenzioni comunemente adottate nello specifico contesto. Per esempio, in un sito web di un paese occidentale, il menu di navigazione orizzontale è allineato a sinistra, e le voci più frequentemente usate sono incontrate per prime, leggendolo da sinistra a destra. In un sito in lingua araba, i menu orizzontali sono allineati a destra, e le voci si susseguono in ordine inverso. Per esempio, la Figura 181 mostra la home page del sito della emittente televisiva Al-Arabiya, di Dubai. La versione in lingua araba, a destra, è concepita per essere letta da destra a sinistra, mentre nella versione in lingua inglese il testo, il titolo della pagina, il menu principale, i titoli delle varie sezioni e il sommario verticale degli articoli seguono le convenzioni comunemente adottate nel mondo occidentale.



Figura 181. www.alarabiya.net: versione in lingua inglese (sinistra) e araba (destra)

- Organizzazione abituale

  La struttura del dialogo e l'organizzazione dei dati dovrebbero permettere all'utente di effettuare le operazioni secondo le modalità a lui consuete. Per esempio, le merci in vendita in un supermercato online dovrebbero essere raggruppate come normalmente lo sono in un supermercato reale.

  Per seguire questa indicazione, il progettista deve tenere presente che le abitudini degli utenti si formano e si modificano rapidamente, e spesso in modo inconsapevole. Un esempio molto interessante è costituito dai layout tipici delle pagine web. Anche se la grafica dei siti web è molto variabile da sito a sito, le aspettative degli utenti per quanto riguarda la posizione dei singoli elementi si sono consolidate molto rapidamente.

In uno studio del 2001[130], a un campione di utenti venne chiesto di mostrare la posizione in cui si aspettavano di trovare alcuni tipici elementi di una pagina web: link interni ed esterni al sito, link alla home page, motore di ricerca interno, carrello per gli acquisti, campo per la registrazione al sito e così via. Nonostante la ancora giovane età del Web, si trovò che gli utenti avevano già sviluppato un sistema di aspettative comuni ben definite. Nella Figura 182, il tono di grigio indica il numero di volte che una certa area del monitor è stata indicata dagli utenti: più il colore è scuro, maggiore è il numero delle persone che ha indicato quella posizione. Per esempio, gli utenti si aspettavano di trovare il link alla home page nell'angolo in alto a sinistra dello schermo oppure, ma con frequenza minore, nel centro del piede della pagina. Ci si aspettava di trovare l'help in alto a destra, come il carrello degli acquisti. E cosi via, come indicato in figura.

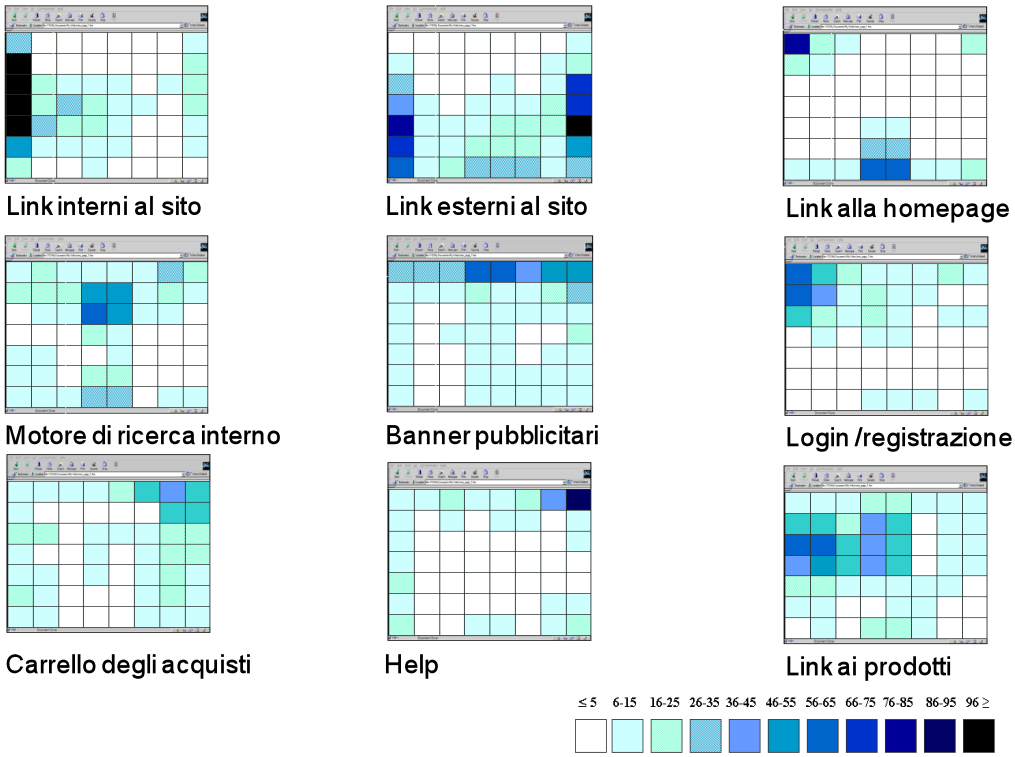


Figura 182. Le aspettative degli utenti sul layout delle pagine web, in uno studio del 2001 (M.Bernard)

- Dialogo consistente

  Le aspettative dell'utente si formano anche nell'ambito di uno stesso sistema. Se questo si comporta in un certo modo in una data situazione, l'utente si aspetterà un comportamento analogo in situazioni simili. Pertanto, tutti i dialoghi realizzati da uno stesso sistema dovrebbero avere aspetto e comportamento consistenti. Per esempio, i bottoni o le voci di menu che servono per attivare le stesse funzioni dovrebbero sempre trovarsi nella stessa posizione. Anche piccole variazioni, come nell'esempio di Figura 183, denotano scarsa attenzione alla coerenza e andrebbero evitate. In questo caso i danni non sono gravi, ma esistono molte situazioni in cui piccole incongruenze nella grafica possono compromettere seriamente l'usabilità di un sistema. Un esempio tipico è quello dei menu che si modificano durante la navigazione. La Figura 184 mostra alcuni menu tratti

[130] M.Bernard, *Developing schemas for the location of common web objects*, Usability News, 3.1 2001, in http://psychology.wichita.edu/surl/usabilitynews/31/web_object.asp . L'indagine è stata ripetuta qualche anno dopo da A.D.Shaikh e K.Lenz, in *Where's the Search? Re-examining User Expectations of Web Objects*, Usability News, 8.1 2006, in http://www.surl.org/usabilitynews/81/pdf/Usability%20News%2081%20-%20Shaikh2.pdf. Come facilmente prevedibile, si sono rilevati diversi cambiamenti nelle aspettative degli utenti, dovuti alla evoluzione del Web nei cinque anni trascorsi dalla prima indagine.

dal sito del film *The Story of Us*. Il menu principale (a) occupa una buona parte della home page. Selezionando la voce The Marriage compare la pagina con la trama del film. In questa pagina, il menu principale è posto sulla sinistra, per lasciare spazio al testo. Questo menu (b) è diverso da quello in home page: non solo è verticalizzato, ma la voce selezionata (The Marriage) è eliminata. Se ora clicchiamo, in (a) o in (b) la voce The Neighborhood, compare una nuova pagina, sulla quale il menu è ancora diverso (c), perché la voce selezionata non vi compare, mentre compaiono tutte le altre. Il menu (d), ancora differente, è quello della pagina Home Movies. Il menu, che dovrebbe costituire per l'utente, per così dire, un'ancora stabile e sicura, si trasforma continuamente, costringendolo a una faticosa e continua rimappatura delle voci[131].

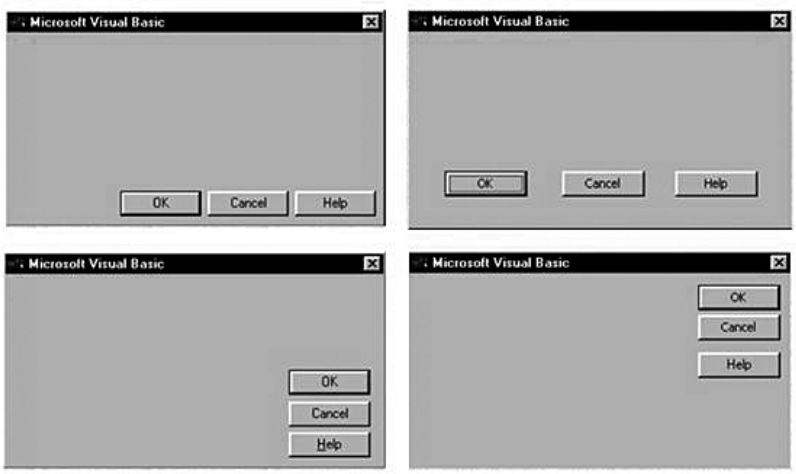


Figura 183. Inconsistenza della posizione degli stessi bottoni (MS Visual Basic 5.0)

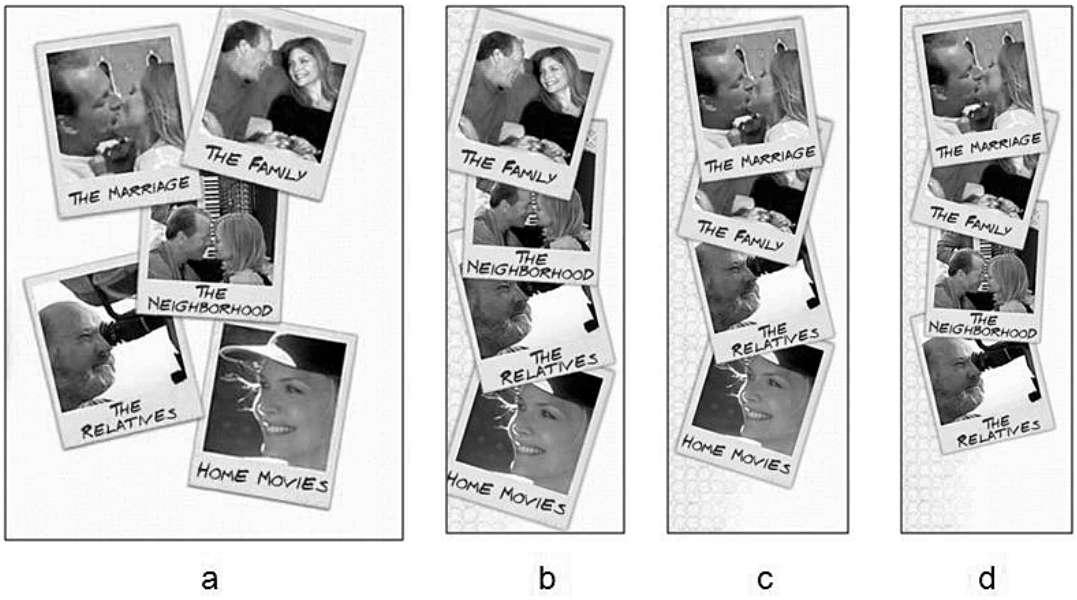


Figura 184. Menu che si trasformano (http://www.thestoryofus.net, 1999)

[131] L'esempio, in realtà, è ancora più complesso. Passando col mouse sopra le schede che rappresentano le voci del menu, i titoli cambiano. Per esempio, The Family diventa The Cast, e la pagina selezionata (intitolata The Family/The Cast) riporta informazioni sugli attori, e non sui personaggi del film, come si sarebbe immaginato. L'effetto combinato del cambiamento di posizione e di testo nelle voci del menu è sconcertante.

Anche l'incoerenza all'interno di una stessa famiglia di applicazioni può essere dannosa. La Figura 185 mostra i pannelli per la definizione degli attributi di un carattere in Microsoft PowerPoint, Word e Excel (nella versione 2007). Al di là di minime differenze, le funzioni sono identiche, ma impaginazione, etichette e dimensioni dei campi sono notevolmente diverse. In questo caso, una maggiore attenzione alla consistenza del dialogo avrebbe non solo migliorato l'usabilità complessiva delle applicazioni di Microsoft Office, ma anche ridotto i costi di sviluppo, permettendo di riutilizzare lo stesso codice di software per tutte le applicazioni della suite.

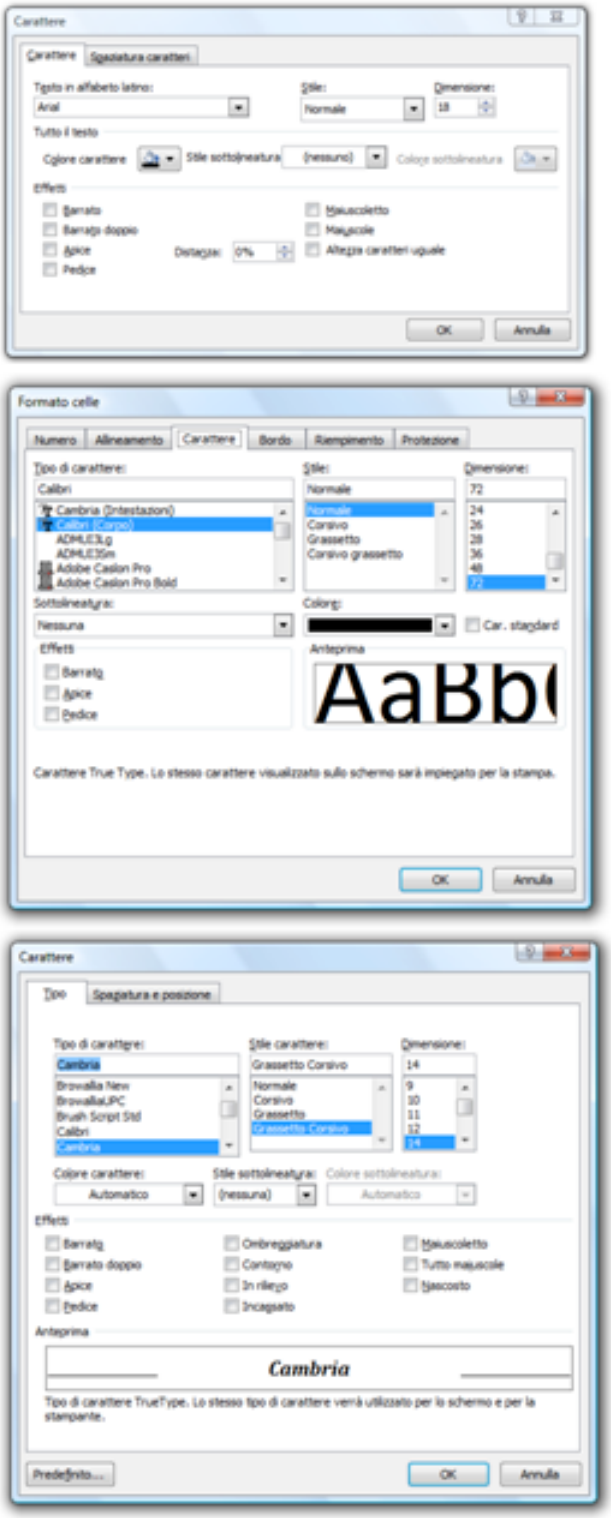


Figura 185. Form di definizione attributi di carattere. Dall'alto in basso: Microsoft PowerPoint 2007, Excel 2007, Word 2007

- Feedback conforme alle aspettative

  Si è già discussa la necessità che il sistema fornisca un adeguato feedback a ogni azione dell'utente. Secondo il modello di Norman, lo scopo è di permettere all'utente di superare agevolmente il golfo della valutazione (pag.20). Perciò, è utile che il feedback sia ben comprensibile e specifico: l'utente dovrebbe essere in grado di interpretarlo senza fatica. Meglio ancora, dovrebbe essere formulato nel modo che l'utente si aspetta. Come abbiamo già visto nel capitolo 3, importante è la sua tempestività: solo così l'utente lo può porre facilmente in relazione con l'azione cui si riferisce. Infatti, risposte non immediate possono essere interpretate come eventi indipendenti da ciò cui si riferiscono: a volte bastano pochi secondi di ritardo per peggiorare significativamente l'usabilità del sistema.

  Come indicazione di larga massima, si può affermare che tempi di risposta fino a 0,1 secondi sono percepiti come immediati, tempi di risposta da 0,1 a 1 secondo non sono percepiti come immediati, ma non sono sufficientemente lunghi da compromettere la continuità cognitiva del compito che l'utente sta svolgendo, e non richiedono quindi che il sistema fornisca particolari feedback. Invece, 10 secondi costituiscono il limite massimo per mantenere l'attenzione dell'utente focalizzata sul compito in corso. Per tempi più lunghi, l'utente desidererà probabilmente dedicarsi ad altri compiti in attesa che l'elaborazione termini, ed è quindi opportuno che il sistema fornisca un feedback adeguato indicando lo stato di avanzamento dell'elaborazione.[132]

- Tempi di risposta conformi alle aspettative

  L'utente si forma delle aspettative sul tempo di esecuzione delle elaborazioni richieste. Se questo dovesse deviare sensibilmente da queste aspettative, dovrebbe esserne preventivamente informato. Se la risposta del sistema ritarda troppo, o se - peggio ancora - il sistema resta "muto", può sorgere il dubbio che si sia bloccato per qualche errore. In questi casi, l'utente spesso rinuncia, e interrompe l'operazione, anche se questa fosse correttamente in corso. Ciò capita di frequente sul Web, quando la pagina richiesta tarda ad apparire.

  Le operazioni che richiedono tempi lunghi sono quindi particolarmente critiche, e il sistema dovrebbe sempre indicare chiaramente che cosa sta facendo, e quanto manca al suo completamento. Nemmeno l'esempio in Figura 186 è pienamente soddisfacente, poiché l'utente non è in grado di trasformare il dato numerico ("devono essere ancora cancellati 30 elementi") in una previsione temporale, in presenza di elementi i cui tempi di cancellazione siano fra loro molto diversi.



Figura 186. Indicatore di progresso (da Mac OS 8)

  È utile tenere presente che il tempo percepito dall'utente non sempre coincide con il tempo reale. In altre parole, in determinate situazioni l'utente può avere la sensazione che il sistema impieghi più (o meno) tempo di quello oggettivamente trascorso. Si possono quindi adottare opportuni accorgimenti per ridurre il tempo percepito, per esempio intrattenendo l'utente durante l'attesa con animazioni divertenti o altro: le soluzioni che vengono adottate, nei siti web, sono infinite. Si è anche verificato sperimentalmente che il tempo percepito dagli utenti trascorre più lentamente nelle fasi finali dell'attesa. È stato quindi suggerito di "rallentare" all'inizio la visualizzazione del trascorrere del tempo, per accelerarla nelle fasi finali, quando l'impazienza dell'utente è massima.

[132] Si veda per es. http://www.useit.com/papers/responsetime.html.

- Messaggi adeguati al contesto

  La lunghezza e il tipo dei messaggi prodotti dal sistema dovrebbero essere adeguati al contesto. Non tutti i messaggi sono adatti a ogni situazione, anche se il loro contenuto è pertinente. Per esempio, se desidero attivare la funzione di controllo ortografico di un word processor e chiedo al sistema di help "dove si trova il controllo ortografico?", dovrei ottenere una risposta breve e appropriata, e non l'elenco di tutti i capitoli del manuale che trattano l'argomento.

- Messaggi in posizione appropriata

  I messaggi di feedback e le spiegazioni fornite all'utente dovrebbero apparire dove si trova il focus dell'attenzione dell'utente, per non interrompere il flusso dell'interazione. Per esempio, in un sistema controllato per mezzo di un telecomando, i messaggi di feedback dovrebbero apparire dove l'utente dirige il telecomando stesso, poiché è lì che guarda.

  Lo stesso principio è adottato per la digitazione dei testi in iPhone. Poiché la tastiera virtuale è molto piccola, l'utente ha bisogno di un feedback che gli confermi di avere digitato proprio il tasto desiderato. Siccome durante la digitazione l'utente guarda il tasto che sta premendo, è lì che gli viene mostrato l'ingrandimento del carattere appena digitato (Figura 187).



Figura 187. Digitazione testi sull'iPhone (dal manuale della Apple)

Una tecnica analoga è utilizzata nel doppio slider di Figura 188, che permette di ricercare dei prodotti (in questo caso, delle fotocamere compatte) in una certa fascia di prezzo in un sito di e-commerce. I prezzi minimi e massimi selezionati (in questo caso, 300,4 e 560,3 Euro) compaiono via via sui controlli che l'utente fa scorrere sullo slider, e non altrove, perché è lì che l'utente guarda durante l'operazione.

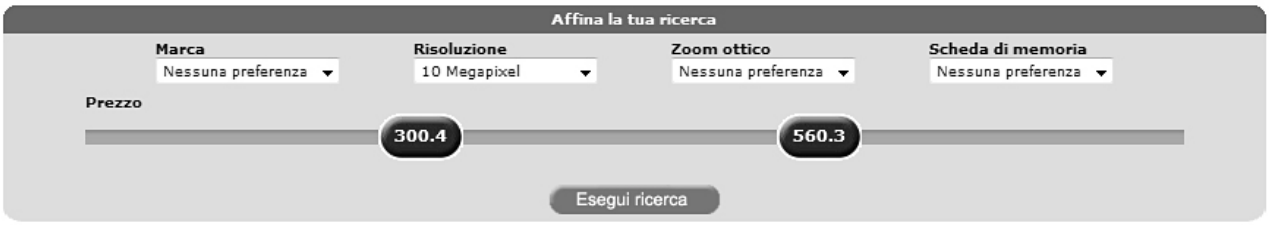


Figura 188. Da http://www.mediaworld.it (2009)

- Input in posizione attesa

  Il sistema dovrebbe richiedere l'input all'utente nella posizione in cui questi si aspetta di doverlo fornire. Questa indicazione è simile alla precedente, e si riferisce soprattutto ai dialoghi in cui l'utente deve fornire

input ripetuti.

Per esempio, durante il controllo ortografico di un testo, il word processor presenta via via all'utente le frasi da correggere, chiedendogli di accettare la correzione proposta o di confermare il testo senza modifiche. Se le frasi da correggere sono molte, la domanda viene posta ripetutamente. Per permettere all'utente di eseguire il compito con la massima efficienza, i bottoni per la conferma o la sostituzione dovrebbero apparire sempre nella stessa posizione. In tal modo l'utente può concentrarsi sul testo da correggere, senza dover spostare lo sguardo, a ogni frase, sulla posizione dei comandi. In Microsoft Word 2007 (Figura 189), correttamente, la finestra di dialogo con i comandi di conferma o sostituzione appare sempre nello stesso posto sul video, mentre il testo da correggere è fatto scorrere, in modo che la frase esaminata (evidenziata) sia sempre visibile nel contesto in cui appare. In altri sistemi, a volte questo non succede, ed è la finestra di dialogo a spostarsi di volta in volta. Questo costringe l'utente a cercare ogni volta la nuova posizione dei bottoni di conferma o di correzione, con notevole rallentamento delle operazioni e considerevole fastidio.[133]

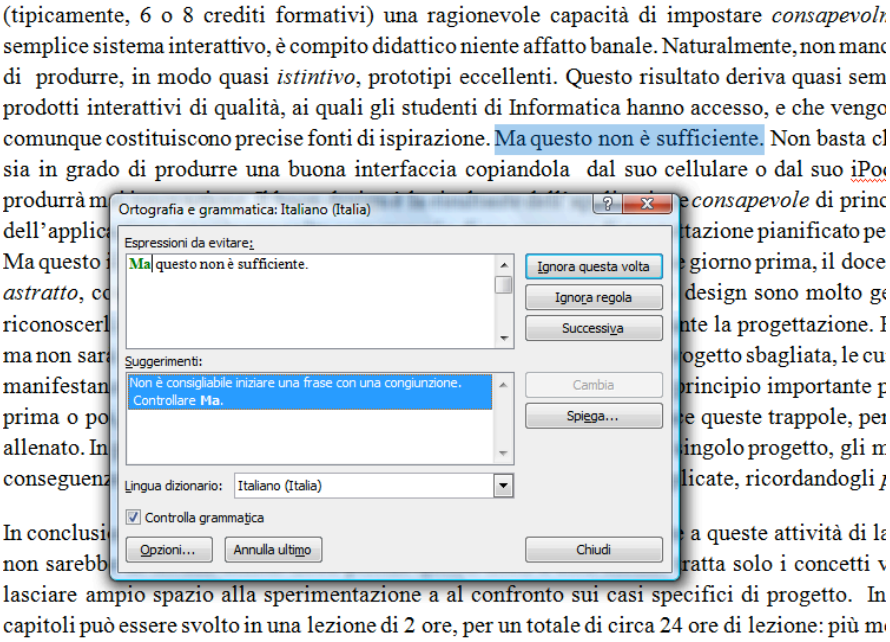


Figura 189. Controllo ortografico in Microsoft Word 2007

- Stile coerente dei messaggi

  Anche lo stile dei messaggi prodotti dal sistema è importante. L'utente si aspetta essi siano espressi in una forma coerente con il contesto e con le convenzioni correnti. In particolare, essi dovrebbero essere formulati in modo oggettivo e costruttivo, evitando qualunque connotazione negativa o enfatica. Questa raccomandazione è particolarmente importante nel caso dei messaggi di errore, che non dovrebbero mai rimproverare l'utente, anche se in modo implicito, per ciò che ha fatto. La Figura 190 mostra quattro esempi di diversa formulazione del messaggio segnalato da un web server quando l'utente cerca di accedere a una pagina inesistente del sito (errore 404). Nel primo caso, il punto esclamativo associa al testo, di per sé neutro, un tono di rimprovero – anche se lieve – che andrebbe senz'altro evitato. L'adeguatezza degli altri tre messaggi, d'identico significato ma di stile completamente diverso, non può, invece, essere valutata al di fuori dello specifico contesto. Essi possono essere considerati appropriati o del tutto fuori luogo, secondo le caratteristiche, finalità e modalità comunicative del sito in cui si trovano. All'argomento della gestione degli errori dell'utente è dedicato l'intero capitolo 11.

[133] In Word 2007, il problema sussiste invece per la funzione Trova e sostituisci, in cui il pulsante per la conferma della sostituzione viene continuamente spostato sul video, via via che il sistema mostra all'utente le occorrenze trovate.

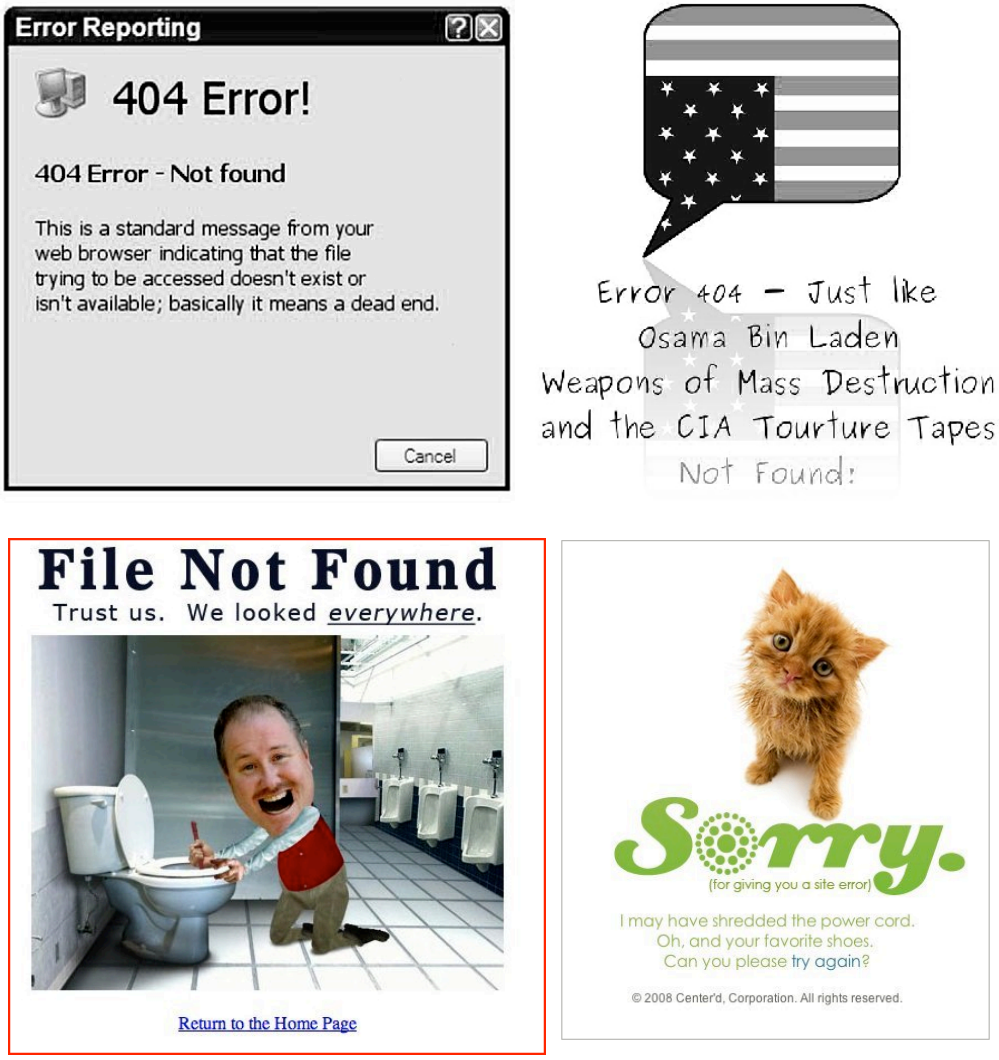


Figura 190. Quattro messaggi per l'errore 404 (pagina non trovata)
(da sinistra a destra e dall'alto in basso: www.fs.fed.us, www.planetquo.net, www.jibjab.com, www.centerd.com)

Adeguatezza all'apprendimento

Questo principio si riferisce alla *learnability*, già introdotta a pag.69. Esso auspica che il dialogo sia organizzato in modo tale da aiutare e guidare l'utente nell'apprendimento del sistema. Alcune linee guida sono le seguenti.

- Bassa soglia di apprendimento

  Ogni sistema dovrebbe essere utilizzabile, sia pure in modo elementare, anche con un livello di apprendimento minimo. L'utente inesperto dovrebbe essere comunque in grado di usare le funzioni di base con un addestramento molto limitato o, meglio ancora, senza alcun addestramento. Si è già osservato che gli utenti non amano leggere i manuali d'uso (pag.71). Di fronte a un nuovo sistema, essi preferiranno provarlo direttamente, esplorandone subito almeno le funzioni più semplici. Il sistema dovrebbe quindi facilitare questa esplorazione, e fornir loro, a tempo debito, le informazioni necessarie per un utilizzo più avanzato.

  Le operazioni richieste ai nuovi utenti per utilizzare i più diffusi servizi online sono spesso particolarmente semplici, e possono essere presi come esempio. Infatti, il fornitore del servizio ha tutto l'interesse a mantenere bassa la soglia d'ingresso, per acquisire il massimo numero possibile di utenti. L'iscrizione e l'accesso al servizio – almeno per le funzioni di base – sono spesso gratuiti, e richiedono da parte dell'utente pochi secondi: la digitazione

del proprio indirizzo di email, la definizione di una password di accesso, e poco altro. Il tutto si risolve in pochi click. Frequente è la disponibilità di un breve video che spiega le funzioni di base del sistema.

Per esempio, la home page di www.dropbox.com, un servizio di storage online, è minimalista (Figura 191). Agli utenti già registrati, propone semplicemente i due campi per inserire email e password. Ai nuovi utenti propone invece un video della durata di 2 minuti che illustra scopo e vantaggi del sistema, e un bottone per il download dell'applicazione che dovrà essere installata sul computer dell'utente. Sotto il bottone è specificato che il servizio è gratuito. Premendolo, all'utente vengono chiesti i (pochi) dati per la registrazione, quindi l'installazione avviene in modo pressoché automatico. Nella parte inferiore della home page sono riportati quattro gruppi di link (Dropbox, Community, Support, About us), com'è abituale per le applicazioni del Web 2.0. Ma poichè non servono ai nuovi utenti, non sono messi in evidenza, e sono in caratteri piccoli. Per vederli bisogna addirittura effettuare uno scroll verticale della pagina.



Figura 191. Home page di http://www.dropbox.com

Una tecnica molto utile è quella di organizzare l'interfaccia utente su due o più livelli di complessità, come nell'esempio di Figura 192, tratto da Microsoft PowerPoint 2007. L'utente può specificare gli attributi di base di un carattere di testo mediante una serie di bottoni sempre visibili sullo schermo (A). Attributi meno frequenti possono essere specificati aprendo un pannello più completo (B), mediante un apposito bottone (C). In questo modo, le opzioni sempre visibili sono in numero limitato, sono facilmente comprensibili e permettono di trattare i casi più comuni, ma il sistema è in grado di gestire anche le richieste degli utenti più evoluti, con un'interfaccia più ricca.

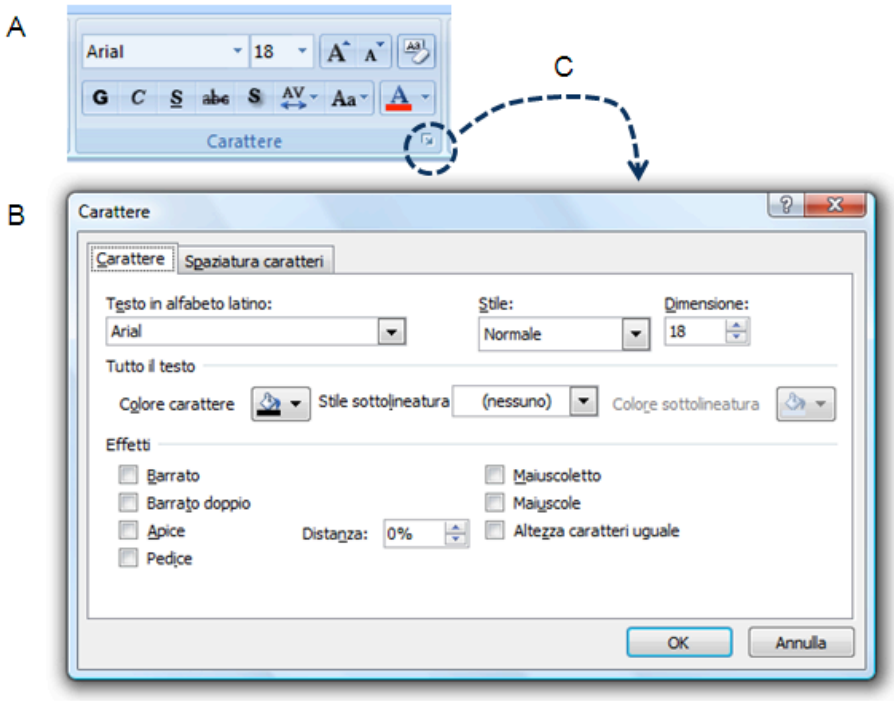


Figura 192. Organizzazione delle funzioni su due livelli (da Microsoft PowerPoint 2007)

Molto importanti, per facilitare l'accesso iniziale al sistema, sono i valori di default dei principali parametri. Essi dovranno quindi essere scelti in accordo alle modalità d'uso più comuni degli utenti inesperti.

- Aiuto alla familiarizzazione

  Il sistema dovrebbe aiutare l'utente a prendere familiarità con il dialogo, fornendo tutti gli aiuti necessari. Anche in questo caso esistono ottimi esempi fra le applicazioni online più diffuse. Per esempio, www.vimeo.com , una social network per il caricamento e la condivisione di video, suggerisce all'utente una serie di compiti iniziali per familiarizzarsi con il sistema (Figura 193).

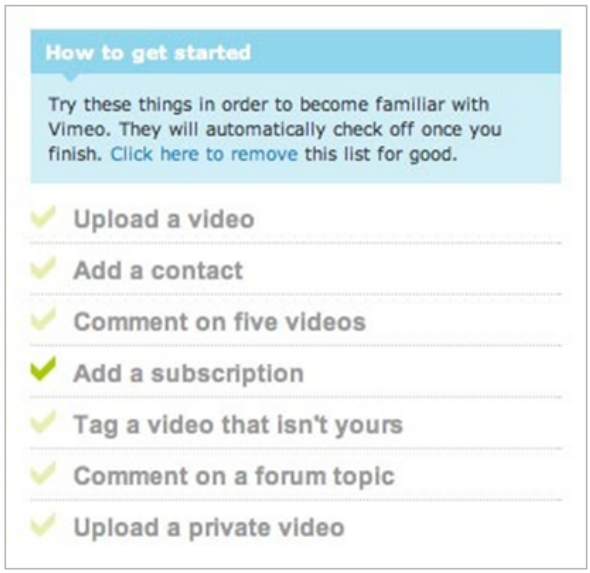


Figura 193. Aiuto alla familiarizzazione in www.vimeo.com

Anche www.digg.com (un sito molto noto che permette agli utenti di segnalare e di votare notizie presenti sul Web), accoglie il nuovo utente, a conclusione del processo di registrazione, suggerendogli alcune attività iniziali (Figura 194).

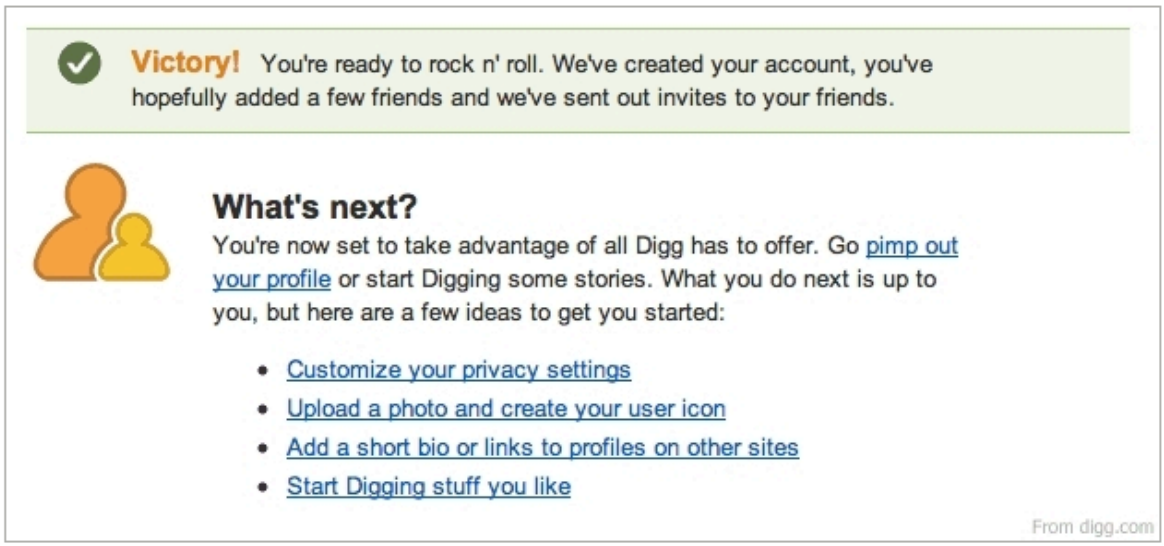


Figura 194. Aiuto alla familiarizzazione in www.digg.com

Funzionalità più sofisticate possono essere suggerite all'utente successivamente. Questo capita regolarmente nei servizi online, che evolvono continuamente, offrendo nuove funzionalità. Per esempio, Twitter utilizza dei riquadri di testo per comunicare cambiamenti nei servizi offerti, suggerimenti per l'uso del sistema, o altro. L'utente può chiudere questi riquadri se non è interessato, o approfondirne il contenuto con l'uso di appositi link o bottoni (Figura 195).

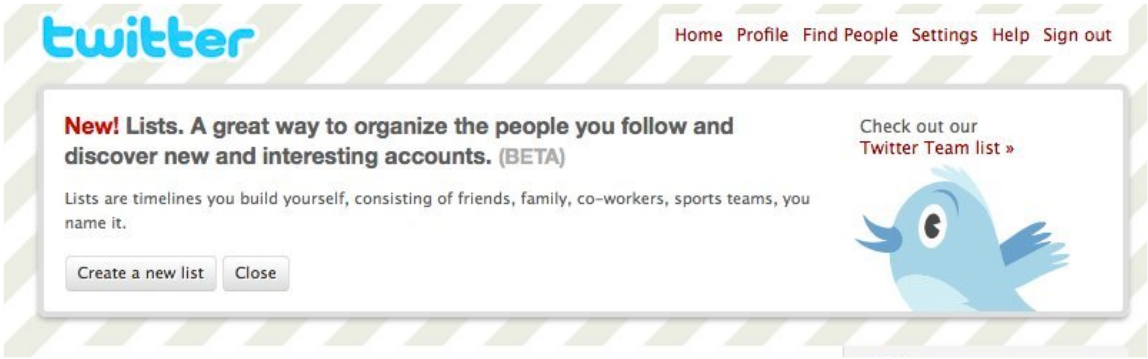


Figura 195. Annuncio di nuove funzionalità in www.twitter.com

- Aiuto online

  Negli esempi precedenti, è il sistema che suggerisce all'utente le azioni per prendere familiarità con le diverse funzioni. Tuttavia, è necessario anche dare all'utente la possibilità di chiedere aiuto quando è lui che lo desidera. Questo è lo scopo dei vari sistemi di *help online*, tradizionalmente disponibili nei prodotti software.

  Per esempio, la Figura 196 mostra come, ponendo il cursore su un controllo di Microsoft Word 2007 (in questo caso il menu a tendina che permette di modificare il colore del testo), appare un messaggio che ne spiega lo scopo.

È bene tenere presente che le esigenze di chi sta imparando sono diverse da quelle di un utente esperto e che non esiste una dicotomia netta fra principianti ed esperti. Infatti, uno stesso utente potrebbe conoscere bene certe funzioni del sistema e non avere alcuna esperienza di altre. Per tenere conto di questi aspetti, le spiegazioni e i feedback del sistema possono essere organizzati su più livelli di dettaglio. Così, nell'esempio di Figura 196, alla spiegazione elementare costituita da un'unica frase (Modifica il colore del testo) è associato un approfondimento (Per ulteriori informazioni, premere F1).



Figura 196. Spiegazione dello scopo di un bottone (Microsoft PoerPoint 2007)

Il sistema di help esemplificato in Figura 196 è del tipo più semplice: l'utente indica un oggetto presente sul video e, in sostanza, chiede al sistema: *a che cosa serve questo?* Ciò non basta: più frequentemente, l'utente desidera un certo risutato, ma non sa come fare per ottenerlo. Vorrebbe, in sostanza, che il sistema fosse in grado di rispondere a una domanda differente: *come devo fare per…?* A questo scopo sono dedicati i sistemi di help più sofisticati o, quando questi non bastano, i blog o i forum di supporto al sistema. In questo caso, la domanda viene posta direttamente al personale del supporto tecnico o agli altri utilizzatori presenti in rete. L'analisi di questi sistemi, che possono essere complessi, esula dagli scopi di questo libro. Ci limiteremo qui a ricordare che, in molti casi, ci si può limitare alla tecnica, molto più semplice, delle *FAQ* (*Frequently Asked Question*), presenti oggi in quasi tutti i servizi online.

- Feedback intermedi

  Il dialogo dovrebbe fornire dei feedback sui risultati intermedi e finali di un compito, in modo che l'utente possa imparare dai compiti portati a termine con successo. Questa tecnica è utilizzata di frequente per le transazioni che avvengono sul Web. Per esempio, quando prenota una stanza d'albergo, o acquista un prodotto, l'utente riceve delle indicazioni che gli permettono di comprendere quali fasi del dialogo ha superato e quali informazioni dovrà ancora fornire per concludere la transazione. Abbiamo già visto l'esempio dell'acquisto di un biglietto ferroviari in cinque fasi (pag.208, Figura 175).

- Modello concettuale evidente

  Il sistema dovrebbe aiutare l'utente a costruirsi un modello concettuale appropriato del sistema. In altre parole, il sistema dovrebbe mostrare chiaramente una propria logica interna, il più possibile semplice e coerente. Lo scopo è di permettere all'utente di orientarsi con facilità anche per l'esecuzione di compiti che richiedono funzioni non ancora utilizzate. Questo può essere fatto in tanti modi, ma la tecnica più frequente è quella di fornire un modello gerarchico delle funzioni del sistema, attraverso un sistema di menu. La tecnica è sicuramente adeguata per i sistemi più semplici. Quando però il numero delle funzionalità è molto elevato, può risultare difficile trovare quella desiderata all'interno di una struttura a menu molto ramificata, come per esempio quella della Figura 202 a pag.231. Una tecnica per risolvere questo problema è, per esempio, quella adottata dalle applicazioni di Office 2007 della Microsoft, in cui le numerose funzionalità sono rappresentate in forma iconica, in una serie di tab orizzontali che lasciano libera la parte inferiore dello schermo (Figura 197). Ovviamente, la tecnica funziona se le

diverse funzioni sono collocate all'interno dei vari tab secondo un criterio che l'utente consideri "naturale".

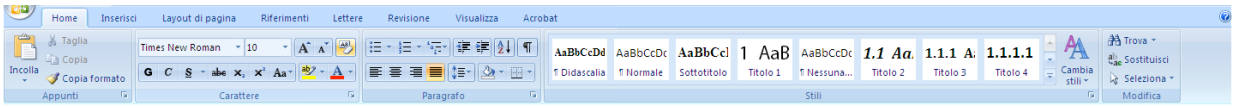


Figura 197. Il tab Home di Microsoft Word 2007

- Sperimentazione sicura

  Il modo più naturale di imparare a usare un sistema è di sperimentarne l'uso. Un sistema ben progettato dovrebbe quindi permettere all'utente di provarne le funzioni, senza che ciò produca conseguenze negative. Questo si può ottenere mettendo a disposizione dell'utente una funzione di *undo*, che consenta di annullare le conseguenze indesiderate delle sue azioni. In questo modo l'utente potrà esercitarsi a usare il sistema esplorandone le caratteristiche, ripristinando lo stato precedente nel caso di esito non desiderato. Le azioni che non potessero essere annullate dovrebbero essere preventivamente segnalate all'utente in modo chiaro, per esempio con un messaggio di avvertimento e una richiesta di conferma. Anche di questo parleremo più dettagliatamente nel capitolo 11 dedicato alla gestione degli errori dell'utente.

- Riapprendimento facilitato

  In tutti i sistemi esistono funzioni che sono usate raramente. Per esempio, in un sistema di contabilità aziendale le funzioni relative alla compilazione del bilancio sono eseguite una sola volta l'anno. Può accadere allora che l'utente dimentichi, nel lungo periodo di tempo fra un utilizzo e il successivo, come utilizzare queste funzionalità, e le debba quindi apprendere di nuovo. Il sistema dovrebbe quindi fornire un aiuto per questo riapprendimento, trattando le funzioni di uso non frequente in modo particolare.

  Ciò è particolarmente importante nel caso di funzionalità di utilizzo raro ma di elevata criticità. In un sistema di sorveglianza domestica, le funzioni che permettono all'utente di intervenire in caso di allarme (per esempio da lontano, con un cellulare) sono usate molto di rado. Infatti, se non si verificano problemi, il sistema non invierà mai alcun allarme. È facile quindi che l'utente, dopo l'iniziale sperimentazione del sistema, si dimentichi come interagire col sistema in caso di allarme. Poiché però un suo intervento errato in situazioni di emergenza può avere gravi conseguenze, il sistema dovrà ricordargli in modo molto chiaro - e rapidamente - come intervenire. Il progettista dovrà studiare questo dialogo con particolare cura, tenendo anche presente la condizione di stress in cui l'utente potrebbe trovarsi in questa situazione, facendogli commettere degli errori.

Controllabilità

In sostanza, questo principio afferma che è l'utente che deve guidare il sistema. Ciò significa che egli non deve essere costretto a seguire una sequenza rigidamente predeterminata di passi d'interazione: egli dovrebbe poter decidere di sospendere il dialogo quando lo desidera, per riprenderlo successivamente, e di fornire le informazioni richieste dal sistema nell'ordine che gli è più congeniale. Dovrebbe poter cambiare idea durante l'interazione, e modificare gli input da lui già forniti, una o più volte, senza vincoli. In breve, l'interazione deve essere controllata dall'utente, e non dal sistema.

Nelle interfacce semplificate del paradigma "scrivi e leggi" dei primi elaboratori (pag. 33), questo principio era spesso violato. Si realizzavano dialoghi del tipo domanda-e-risposta, in cui il sistema poneva una serie di domande, alle quali l'utente doveva rispondere nell'ordine stabilito, senza deroghe. Esempi tipici di questa modalità d'interazione erano i primi sistemi esperti, come quello già illustrato in Figura 21, a pag.35.

Oggi questo stile d'interazione è praticamente scomparso, e i sistemi lasciano all'utente molta più flessibilità. Per esempio, nei dialoghi guidati da form (pag. 35), l'utente può scegliere l'ordine di compilazione dei vari campi e, se lo desidera, può tornare indietro e correggere gli input già forniti. Tuttavia capita non di rado che il progettista, per ridurre la complessità del software da realizzare, introduca delle rigidità nel dialogo che ne compromettono l'usabilità. Per evitarlo, è importante seguire le linee guida seguenti.

- Tempi dell'interazione controllati dall'utente

  L'utente dovrebbe poter impiegare tutto il tempo che desidera per effettuare i vari passi del dialogo, senza che il sistema gli ponga dei vincoli.

  I time-out imposti dal sistema, che in qualche caso sono inevitabili, andrebbero introdotti solo in casi di effettiva necessità. Questa raccomandazione può essere in conflitto con esigenze di sicurezza, come avviene per esempio in alcune transazioni sul Web. Per esempio, durante l'accesso a una banca online, dopo un periodo d'inattività da parte dell'utente l'applicazione chiude automaticamente la sessione senza chiedere conferma. Infatti, l'utente potrebbe essersene andato senza effettuare il logout. Per evitare che altri utenti non autorizzati possano subentrare ed effettuare transazioni illecite, la banca preferisce chiudere d'autorità la sessione, anche se questo può comportare un grave disagio all'utente ancora presente.

  Queste indicazioni si applicano anche ai messaggi inviati dal sistema. È pratica scorretta far sì che i messaggi restino visibili solo per un tempo limitato (per esempio, alcuni secondi), e poi siano automaticamente rimossi. Infatti, non è lecito assumere che l'utente possa leggerli prima della scadenza del tempo: potrebbe essere distratto o impegnato in altre attività. Ogni messaggio inviato dal sistema dovrebbe restare visibile fino a quando l'utente non ne segnali esplicitamente l'avvenuta lettura, per esempio premendo un apposito bottone di OK.

- Proseguimento del dialogo controllato dall'utente

  L'utente dovrebbe poter decidere come proseguire nel dialogo, senza che il sistema imponga vincoli rigidi. A volte può essere opportuno permettergli di effettuare uno o più compiti secondari durante l'esecuzione del compito principale. Per esempio, quando l'utente riceve una telefonata, il cellulare potrebbe permettergli di inserire "al volo" numero e nome del chiamante nella rubrica, mentre accetta la chiamata. Oppure, durante una telefonata, l'utente dovrebbe poter accedere alla rubrica, per comunicare un numero all'interlocutore.

- Punto di ripartenza controllato dall'utente

  Se il dialogo è stato interrotto per qualche motivo, l'utente dovrebbe poter scegliere il punto dal quale riprenderlo, se ciò è compatibile con il compito. Per esempio, se l'utente deve interrompere la compilazione di un sms per rispondere a una chiamata, il sistema dovrebbe archiviarlo automaticamente in bozza, per consentirgli di riprendere da dove era stato interrotto.

  A volte il motivo dell'interruzione del dialogo è dovuto al fatto che l'utente si accorge di dover modificare qualche input da lui fornito in precedenza. Anche in questo caso, il sistema dovrebbe permettergli di ripartire dal punto desiderato, senza costringerlo a ripartire da capo. Questa importante linea guida non è seguita, per esempio, dal sistema di prenotazione voli dell'Alitalia. Il processo è suddiviso in 6 fasi, e all'utente viene correttamente indicata la fase in cui si trova (in Figura 198, la fase corrente è Acquista). Tuttavia, se l'utente vuole tornare a una fase precedente, per esempio per modificare i propri dati (forniti nella fase Dati passeggero) o cambiare la scelta del volo (effettuata nella fase Scegli volo), il sistema non glielo permette, costringendolo a tornare alla prima fase (Modifica volo) e ripetere l'intero processo dall'inizio.

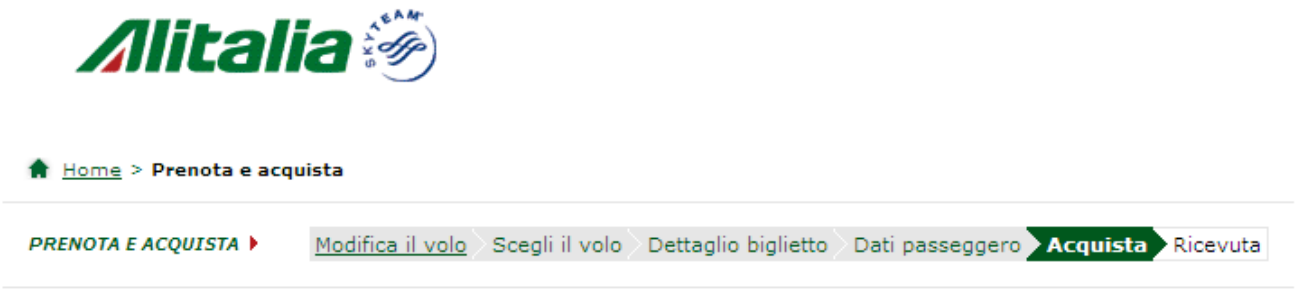


Figura 198. Fasi dell'acquisto di un biglietto aereo in http://www.alitalia.it (2009)

- Reversibilità delle operazioni

  Se le operazioni sono reversibili e se il contesto d'uso lo permette, dovrebbe essere sempre possibile annullare

almeno il passo più recente del dialogo (e, di conseguenza, annullare lo stesso annullamento).

La disponibilità di una funzione di undo (e redo) migliora sensibilmente la usabilità di un sistema, perché permette di eliminare le conseguenze di azioni errate, ed elimina l'ansia causata dal timore di commettere errori.

I sistemi più sofisticati permettono di annullare numerosi passi del dialogo. Per esempio, le applicazioni di Office 2007 permettono di annullare fino a 100 azioni precedenti (una o più alla volta) e di rieseguirle dopo l'annullamento. Abbiamo appena visto, nell'esempio precedente, che nel processo di prenotazione voli di Alitalia non è disponibile una funzione di undo per tornare a una fase precedente.

- Modalità di visualizzazione dei dati controllata dall'utente

  È utile che l'utente possa tenere sotto controllo non soltanto la sequenza dei passi del dialogo, ma anche le modalità di visualizzazione dei dati necessari al compito. Questo è particolarmente importante nel caso in cui essi siano numerosi: l'utente dovrebbe essere in grado di controllare quali e quanti dati gli vengono mostrati.

  Per esempio, un'agenda elettronica ben fatta, oltre a permettere viste diverse del calendario degli impegni (mensile, settimanale, giornaliero) potrebbe permettere di visualizzare tutti gli appuntamenti con uno specifico cliente.

- Dispositivo d'interazione scelto dall'utente

  All'utente dovrebbe essere permesso di scegliere uno qualsiasi dei dispositivi di input o di output disponibili, se compatibili con il compito.

  Per esempio, la funzione di ricerca in un sito web potrebbe essere attivata, dopo avere immesso la parola cercata nell'apposito campo, cliccando col mouse il bottone Cerca, oppure premendo il tasto Enter.

- Personalizzazione dei valori di default

  L'utente dovrebbe essere in grado di definire nuovi valori di default in accordo alle proprie personali esigenze, se compatibili con il compito.

  Per esempio, in un word processor, lo stile utilizzato come default per il testo "normale" dovrebbe poter essere modificato dall'utente.

- Disponibilità dei dati originali

  Dopo la loro modifica, i dati originali dovrebbero rimanere disponibili all'utente, se necessari per il compito.

Tolleranza verso l'errore

Nell'interazione con un sistema, anche l'utente più esperto commette inevitabilmente degli errori. È compito dei progettisti concepire sistemi che riducano al minimo la possibilità che questi errori avvengano, e la gravità delle loro conseguenze. Un dialogo si dice *tollerante verso gli errori* (in inglese, *error-tolerant*) quando fornisce i risultati desiderati anche in presenza di errori dell'utente, senza (o con minime) azioni correttive da parte sua. Questa proprietà si ottiene mediante accorgimenti diversi, che permettano di prevenire gli errori per quanto è possibile (*error prevention*), di segnalarli con chiarezza quando avvengono (*error handling*), suggerendo o effettuando automaticamente le azioni correttive appropriate (*error recovery*[134]). Si tratta di un tema molto importante per il progettista dei sistemi interattivi, al quale è dedicato l'intero capitolo 11. Pertanto, qui di seguito ci limitiamo a riassumere in forma schematica le raccomandazioni contenute nell'ISO 9241, rimandando al capitolo 11 per approfondimenti ed esempi.

- Assistenza all'utente

  Il sistema dovrebbe aiutare l'utente a evitare di commettere errori negli input da lui forniti, e a scoprire quelli che comunque vengono commessi.

- Verifica e convalida dei dati

  Prima di procedere all'elaborazione dell'input, il sistema dovrebbe verificarlo e convalidarlo.

- Prevenzione di azioni non lecite

  Il sistema dovrebbe evitare che un'azione dell'utente possa causare una caduta o uno stato indefinito del sistema. Per esempio, se si desidera stampare un documento composto da 35 pagine, il dialogo della funzione di stampa dovrebbe permettere di specificare soltanto numeri di pagina nell'intervallo compreso fra 1 e 35.

[134] Il termine inglese error recovery, comunemente usato, può essere tradotto con recupero, o ripristino, o ripresa dall'errore.

- Richieste di conferma

  Prima di eseguire azioni che possano produrre conseguenze gravi e non annullabili, il sistema dovrebbe chiedere conferma all'utente. Per esempio, quando l'utente chiede di cancellare un file.
- Spiegazione dell'errore

  Quando l'utente commette un errore, il sistema dovrebbe fornirgli una spiegazione adeguata, indicando la causa dell'errore e le modalità di correzione.
- Spiegazioni aggiuntive

  Se possibile e opportuno, il sistema dovrebbe fornire all'utente, su sua richiesta, informazioni aggiuntive sulla natura dell'errore e sulla sua correzione.

  Per esempio, a fronte di un errore il sistema potrebbe emettere un messaggio conciso, contenente un link a una spiegazione più dettagliata.
- Assistenza per il recupero

  Quando l'utente commette un errore, il sistema dovrebbe fornirgli un supporto attivo per permettergli di ristabilire la situazione corretta.
- Minimo sforzo di correzione

  I passi necessari per correggere un errore dovrebbero essere semplificati al massimo.

  Per esempio, a seguito di un input errato, il cursore viene posizionato automaticamente dove è richiesta la correzione.
- Correzione differibile

  L'utente dovrebbe poter rimandare la correzione dell'errore a un momento successivo, a meno che ciò non sia necessario per proseguire nel dialogo.

  Per esempio, l'utente dovrebbe essere in grado di completare la compilazione di un indirizzo, anche se il codice di avviamento postale è errato, rimandando l'immissione del codice corretto a un momento successivo. Infatti, per conoscere il codice corretto, potrebbe avere la necessità di consultare una fonte al momento non disponibile.
- Correzione automatica modificabile

  Quando il sistema è in grado di correggere automaticamente un errore commesso dall'utente, dovrebbe avvisarlo della correzione effettuata e permettergli di modificarla.

  Per esempio, quando il correttore ortografico di un word processor corregge una parola digitata dall'utente, questi deve essere in grado di modificare la correzione.

Adeguatezza all'individualizzazione

Come si è più volte ricordato, l'usabilità di un sistema è relativa a uno specifico utente, compito e contesto d'uso. È quindi utile che un sistema permetta all'utente di adattarne il comportamento alle proprie specifiche esigenze e capacità. Si dice allora che il sistema è *personalizzabile* o, nel linguaggio dell'ISO 9241-110, *individualizzabile*.

La personalizzazione può riguardare numerosi aspetti diversi, quali la lingua, le preferenze in relazione ai compiti da effettuare, le modalità di rappresentazione dei dati, e così via. Alcune linee guida da tenere presenti in relazione a questo principio sono le seguenti.

- Scelta di rappresentazioni alternative

  Il sistema dovrebbe permettere all'utente di scegliere fra varie forme di rappresentazione, adatte alle diverse necessità individuali. La Figura 199 mostra, per esempio, come l'utente possa scegliere il sistema di misura, la valuta, e la rappresentazione di ora, data e numeri nel sistema operativo MacOS della Apple. Queste sono solo una parte delle numerose personalizzazioni possibili, selezionabili dal menu Preferenze del sistema.

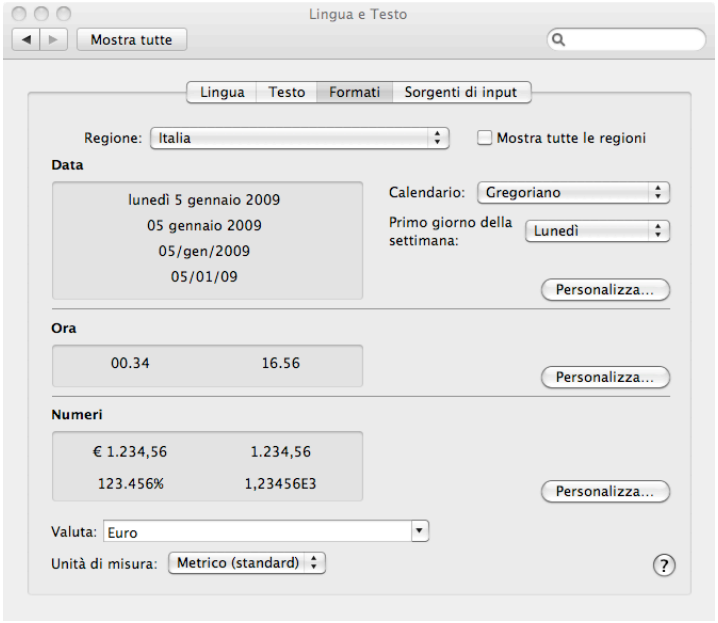


Figura 199. MacOS Finder 10.6 (2009)

Particolarmente importanti sono le possibilità di scelta di rappresentazioni alternative che rendano il sistema accessibile a utenti con disabilità (visive, auditive o di altro tipo). Queste opzioni sono oggi normalmente presenti in ogni sistema operativo, ma dovrebbero essere fornite anche in ogni applicazione di pubblica utilità.

Per esempio, un sito web permette di visualizzare i testi in modalità "alto contrasto", per facilitarne la lettura agli utenti ipo-vedenti.

- Scelta dei formati dei dati input e output

  L'utente dovrebbe poter scegliere le rappresentazioni più appropriate per il formato dei dati elaborati nello specifico contesto applicativo.

- Vocabolario personalizzabile

  In molti casi, è utile poter arricchire il vocabolario usato dal sistema, per aggiungere eventuali termini utilizzati nel contesto specifico.

  Un buon esempio è fornito da http://www.ning.com, un servizio online che permette di realizzare delle social network private. Il sistema è multi-lingue: l'amministratore di ogni social network può scegliere la lingua che il sistema userà per le voci dei menu e i suoi messaggi, fra numerose possibilità. La traduzione dei testi base in inglese, però, non è rigidamente prefissata: l'amministratore, se lo desidera, la può cambiare, semplicemente modificandola in una tabella (Figura 200).



Figura 200. Personalizzazione della traduzione in www.ning.com (2009)

- Scelta del livello delle spiegazioni

  Il livello di dettaglio e/o la forma delle spiegazioni (per esempio, nei messaggi di errore o nei testi di help) dovrebbe essere modificabile in funzione del livello di conoscenza dell'utente.

- Personalizzazione dei tempi di risposta

  L'utente dovrebbe poter modificare i parametri relativi ai tempi di risposta dei dispositivi di input e di output, per adattarli alle proprie personali esigenze.

  Per esempio, i sistemi operativi permettono normalmente di impostare vari parametri, quali la sensibilità del mouse e l'intervallo temporale accettato dal sistema per il "doppio clic", e altri, in funzione delle preferenze dell'utente. La Figura 89, a pag.102, mostra il pannello del sistema operativo del Mac che permette di regolare i parametri del mouse.

- Scelta del metodo d'interazione

  Quando appropriato, l'utente dovrebbe poter scegliere fra diverse tecniche di dialogo o metodi d'interazione.

  Per esempio, un word processor permette di salvare un documento selezionando la voce di un menu, o cliccando su un'icona, o digitando una combinazione di tasti. Un sistema per l'acquisto di biglietti aerei permette all'utente di specificare la data selezionandola su un calendario oppure, semplicemente, digitandone il valore nel campo a essa dedicato, come nell'esempio di Figura 201 .



Figura 201. Selezione della data in www.alitalia.it (2009)

- Personalizzazione del dialogo

  Se appropriato, l'utente dovrebbe poter modificare alcune componenti del dialogo, per adattarlo a specifiche necessità nell'effettuazione dei compiti.

  La personalizzazione può essere più o meno spinta. Per esempio, in un word processor l'utente può scegliere quali *toolbar* debbano essere visualizzate. In questo modo, se l'utente non ha necessità di utilizzare strumenti di composizione grafica (per esempio, perché il documento in costruzione non contiene figure), potrà evitare di affollare il video con comandi che non gli servono.

  Un altro esempio di personalizzazione, tratto dal word processor Word 2008 per Mac, è mostrato in Figura 202. L'utente può definire un proprio glossario di frasi che utilizza di frequente, che potranno poi essere inserire nel testo in costruzione semplicemente selezionandole da un menu. Riferendoci all'esempio, l'utente potrà scegliere da un menu di formule di chiusura la frase "Con i migliori saluti" e chiederne l'inserimento nel testo. La frase selezionata sarà stata da lui precedentemente inserita nel glossario utilizzando un'apposita funzione. Questo meccanismo può essere molto utile per aumentare l'efficienza della composizione di particolari tipi di testi costituiti da clausole standard ripetute di frequente, come per esempio i documenti contrattuali.

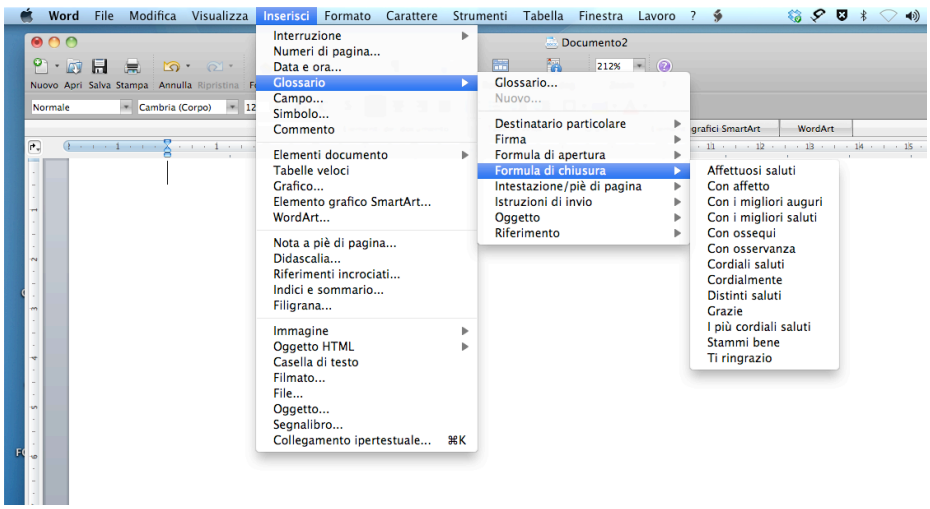


Figura 202. Microsoft Word 2008 per Mac

Lo stesso word processor permette, ancora, di personalizzare gli stessi menu dei comandi, aggiungendo o eliminando particolari voci.

- Rispristinabilità dei valori precedenti

  I sistemi interattivi più evoluti forniscono di solito ampie possibilità di personalizzazione. Questo può creare delle difficoltà all'utente, che potrebbe dimenticare quali personalizzazioni ha attivato. È quindi importante che il sistema presenti un quadro chiaro del valore dei parametri, e che l'utente possa ripristinare facilmente le impostazioni di default o quelle da lui stesso definite precedentemente. Nel caso di sistemi utilizzati da più utenti, ciascuno di essi dovrebbe poter memorizzare i parametri di personalizzazione in un proprio profilo, per poter attivare velocemente la propria configurazione.

Sintesi delle linee guida

In questo capitolo abbiamo descritto i principi dell'ISO 9241-110 e numerose linee guida per la progettazione dei dialoghi uomo-sistema, inquadrandole in questi principi. Queste linee guida, liberamente ispirate alle raccomandazioni presenti nel documento dell'ISO, sono state illustrate con numerosi esempi tratti da sistemi di varia natura e tecnologia, realizzati in periodi diversi. Esso sono le seguenti.

Adeguatezza al compito
- Dialogo adeguato al compito
- Informazione adeguata al compito
- Dialogo essenziale
- Dispositivi di input e output adeguati al compito
- Formati di input e output adeguati al compito
- Default tipici
- Compatibilità con i documenti

Auto-descrizione
- Guida per l'utente
- Interazione evidente
- Descrizione dell'input atteso
- Stato visibile
- Formati descritti
- Manualistica minima

Conformità alle aspettative dell'utente
- Linguaggio familiare
- Aderenza alle convenzioni
- Organizzazione abituale
- Dialogo consistente
- Feedback conforme alle aspettative
- Tempi di risposta conformi alle aspettative
- Messaggi adeguati al contesto
- Messaggi in posizione appropriata
- Input in posizione attesa
- Stile coerente dei messaggi

Adeguatezza all'apprendimento
- Bassa soglia di apprendimento
- Aiuto alla familiarizzazione

- Aiuto online
- Feedback intermedi
- Modello concettuale evidente
- Sperimentazione sicura
- Riapprendimento facilitato

Controllabilità

- Tempi dell'interazione controllati dall'utente
- Proseguimento del dialogo controllato dall'utente
- Punto di ripartenza controllato dall'utente
- Reversibilità delle operazioni
- Modalità di visualizzazione dei dati controllata dall'utente
- Dispositivo d'interazione scelto dall'utente
- Personalizzazione dei valori di default
- Disponibilità dei dati originali

Tolleranza verso l'errore

- Assistenza all'utente
- Verifica e convalida dei dati
- Prevenzione di azioni non lecite
- Richieste di conferma
- Spiegazione dell'errore
- Spiegazioni aggiuntive
- Assistenza per il recupero
- Minimo sforzo di recupero
- Recupero differibile
- Recupero automatico modificabile

Adeguatezza all'individualizzazione

- Scelta di rappresentazioni alternative
- Scelta dei formati di input e output
- Vocabolario personalizzabile
- Scelta del livello delle spiegazioni
- Scelta del metodo d'interazione
- Personalizzazione del dialogo
- Ripristinabilità dei valori precedenti
- Personalizzazione dei tempi di risposta.

Si tratta di principi e linee guida del tutto generali, che non dipendono dalle specifiche tecnologie d'interazione utilizzate, che possono essere molto diverse (Figura 4, pag.14). Il sistema ci può trasmettere informazioni attraverso il senso della vista o dell'udito, oppure (ma più raramente) generando sensazioni tattili. A sua volta, l'uomo può comunicare utilizzando le mani, attraverso l'uso di tastiere o altri dispositivi di manipolazione, la voce (attraverso sistemi di riconoscimento vocale) oppure, anche se più raramente, con i movimenti del proprio corpo, che il sistema può rilevare attraverso sensori opportunamente collocati nello spazio o perfino con lo sguardo (utilizzando apparati di eye-tracking).

Ogni dispositivo possiede caratteristiche specifiche, e interagisce con il sistema percettivo, cognitivo o motorio dell'utente in modo diverso. Sarebbe quindi utile introdurre delle raccomandazioni specifiche per ciascuno di questi dispositivi, allo scopo di orientare il progettista verso le soluzioni di progetto più corrette dal punto di vista dell'usabilità. Questo richiede, da una parte, l'analisi delle caratteristiche tecnologiche e funzionali dei diversi dispositivi e, dall'altra, lo studio delle abilità umane coinvolte nel loro utilizzo.

Tutto ciò ci porterebbe lontano, ed esula dagli scopi di questo libro, che ha carattere introduttivo. Ci limiteremo, pertanto, a fornire qualche indicazione generale solo per quanto riguarda la comunicazione visiva, per il suo ruolo privilegiato rispetto ad altri canali di comunicazione. A questo argomento è dedicato il capitolo 12 e, per quanto riguarda l'uso del testo, il capitolo 13.

Ripasso ed esercizi

1. Descrivi le differenze fra principi, linee guida, regole di progetto e standard.
2. Quali sono i principali standard prodotti dal Technical Committee 159/SC 4 dell'ISO?
3. Riassumi i sette principi del dialogo secondo la ISO 9241.
4. Esamina, alla luce delle linee guida descritte in questo capitolo, l'interfaccia interna ed esterna dell'ascensore di casa tua. A tuo parere, il progetto di queste interfacce è coerente con queste linee guida? In caso contrario, quali sono i difetti e con quali conseguenze dal punto di vista dell'usabilità?
5. Che cosa significa, secondo l'ISO 9241, che un dialogo è controllabile? Quali sono, a tuo parere, le motivazioni che hanno portato gli autori dello standard a inserire la controllabilità fra i principi di base del dialogo?
6. Esamina le differenti possibilità di personalizzazione previste dal sistema di word processing che utilizzi normalmente. Quali sono? Quali potresti utilizzare e come, in funzione delle tue specifiche necessità e abitudini?
7. Esamina i diversi sistemi di help del word processor che usi di solito, e riassumine per punti le caratteristiche salienti. Li ritieni adeguati? Perché?
8. Esamina criticamente l'elenco delle linee guida riportato a pag.232. Ritieni che ci siano delle sovrapposizioni e delle ridondanze? Quali? Ne proporresti una semplificazione? Quale?

Approfondimenti e ricerche

1. Scarica il documento *Research-based Web Design and Usability Guidelines*, citato in nota a pagina 202, che contiene oltre 200 linee guida per l'usabilità dei siti web. Confronta queste linee guida con quelle discusse nel presente capitolo, e associa ciascuna linea guida ad uno dei sette principi del dialogo dell'ISO 9241. Suggerimento: dovrebbe essere sufficiente confrontare l'elenco delle linee guida di pag.232 con l'elenco delle linee guida riportato in testa al citato documento, esaminando la relativa scheda descrittiva solo in caso di dubbio.
2. Le linee guida per la progettazione delle interfacce utente del Macintosh e di Windows Vista sono contenute, rispettivamente, nei documenti *Apple Human Interface Guidelines*, e *Windows User Experience Interaction Guidelines,* reperibili in rete agli indirizzi:
   http://developer.apple.com/documentation/UserExperience/Conceptual/AppleHIGuidelines e
   http://msdn.microsoft.com/en-us/library/aa511258.aspx.

   Esamina questi documenti, in particolar modo le sezioni relative a icone, menu, windows, messaggi, controlli e layout.
3. Il libro di A.Dix, J.Finlay, G.Abowd e R.Beale, *Interazione uomo macchina*, (ed.italiana McGraw-Hill, 2004) descrive un insieme di principi per la progettazione di buone interfacce, diversi da quelli dell'ISO 9241. Questi principi, che sono chiamati nel libro "regole di design", sono dettagliatamente descritti nel Cap. 7 dell'edizione italiana (pagg.253-284), organizzati in tre gruppi: Apprendibilità (composto da 5 principi), Flessibilità (5 principi) e Robustezza (4 principi). Esamina in dettaglio questi principi e costruisci una tabella che li pone a confronto con i principi ISO e con le relative raccomandazioni. Quale dei due insiemi è più comprensivo? Ci sono principi presenti solo in una delle due formulazioni? Quali?
4. Verifica la evoluzione degli standard prodotti dal comitato TC159/SC 4, sul sito http://www.iso.org.

11. Progettare per l'errore

Sintesi del capitolo

Questo capitolo discute la nozione di errore umano nel dialogo uomo-macchina e presenta alcune linee guida per il trattamento degli errori compiuti dall'utente, approfondendo quanto già detto nel capitolo 10. Dopo una classificazione dei principali errori in lapsus e sbagli, si descrivono le principali tecniche di prevenzione a disposizione del progettista. Si discutono poi le linee guida per una corretta diagnosi degli errori. Infine si descrivono i processi di ripristino, suddividendoli in due tipologie: forward recovery e backward recovery. Il capitolo contiene numerosi esempi.

L'errore umano

Nel capitolo 10 abbiamo elencato le raccomandazioni fornite dallo standard ISO 9241-110 per la realizzazione di sistemi error-tolerant, cioè quei sistemi che consentono all'utente di raggiungere facilmente gli obiettivi desiderati nonostante gli errori compiuti nell'interazione. Questo capitolo tratta in modo più dettagliato questo argomento che, nonostante la sua grande importanza pratica ai fini dell'usabilità di un sistema, viene spesso trascurato dai progettisti.

Anche l'utente più esperto commette degli errori. Questo è inevitabile, se si pensa che spesso, nei compiti quotidiani, c'è un solo modo di fare le cose nel modo corretto, ma molti modi di sbagliare. Pensiamo al compito di cuocere un uovo sodo: si tratta di un processo elementare, ma la lista dei possibili errori è piuttosto lunga. Possiamo romperlo per un movimento troppo brusco mentre lo prendiamo dal frigorifero, se lo immergiamo nell'acqua bollente quando è troppo freddo il guscio tende a incrinarsi, possiamo sbagliare il tempo di cottura o rovinarlo mentre lo sgusciamo. Ci può cadere per terra mentre lo portiamo in tavola. Infine, possiamo non accorgerci, prima di assaggiarlo, che si tratta di un uovo rancido.

Il progettista deve innanzitutto comprendere che *l'utente che sbaglia non è un utente sbagliato*, ed evitare di colpevolizzarlo, o pretendere da lui un'impossibile perfezione. Deve accettare il fatto che *l'utente sbaglia perché il sistema gli consente di sbagliare* e questo, in ultima analisi, è un difetto ascrivibile a cattiva progettazione. Il progettista deve allora predisporre tutti gli accorgimenti per evitare, per quanto possibile, questa eventualità e gestirla nel modo più corretto quando si verifica.

A questo scopo, deve innanzitutto comprendere meglio la natura dell'errore umano. Che cosa significa errore? Se cerchiamo nel dizionario la definizione di questo termine, troviamo, per esempio:

Atto, e effetto di allontanarsi dalla verità o dalla norma convenuta[135]

Questa definizione non è di grande aiuto, perché troppo generale. Noi siamo interessati a comprendere l'errore umano dal punto di vista psicologico e operativo, non filosofico. Un errore compiuto dall'uomo – e in particolare dall'utente di un sistema interattivo - può avere cause diverse e produrre effetti differenti: se conosciamo le cause possibili, possiamo cercare di prevenirlo; se ne conosciamo gli effetti, possiamo cercare di limitarli. Fortunatamente l'errore umano è stato ampiamente analizzato dagli studiosi di scienze cognitive. James Reason, nel suo libro *Human Error* ne dà la seguente definizione operativa:

> *"Errore" sarà inteso come un termine generico per comprendere tutti i casi in cui una sequenza pianificata di attività fisiche o mentali fallisce il suo scopo, e quando questo fallimento non possa essere attribuito all'intervento di qualche agente casuale*[136].

[135] dal dizionario Garzanti della lingua italiana.
[136] James Reason, *Human Error*, Cambridge University Press, 1990, pag.9 (nostra traduzione).

Nello stesso libro, si trova una classificazione molto utile per i nostri scopi, illustrata nella Figura 203. In questo schema, un'azione è considerata corretta quando si verificano tre condizioni: 1) l'utente aveva l'intenzione di agire, 2) l'azione è proceduta come desiderato, 3) l'azione ha ottenuto il suo scopo. Se non si verificano tutte queste tre condizioni, si hanno quattro fondamentali tipi di errori.

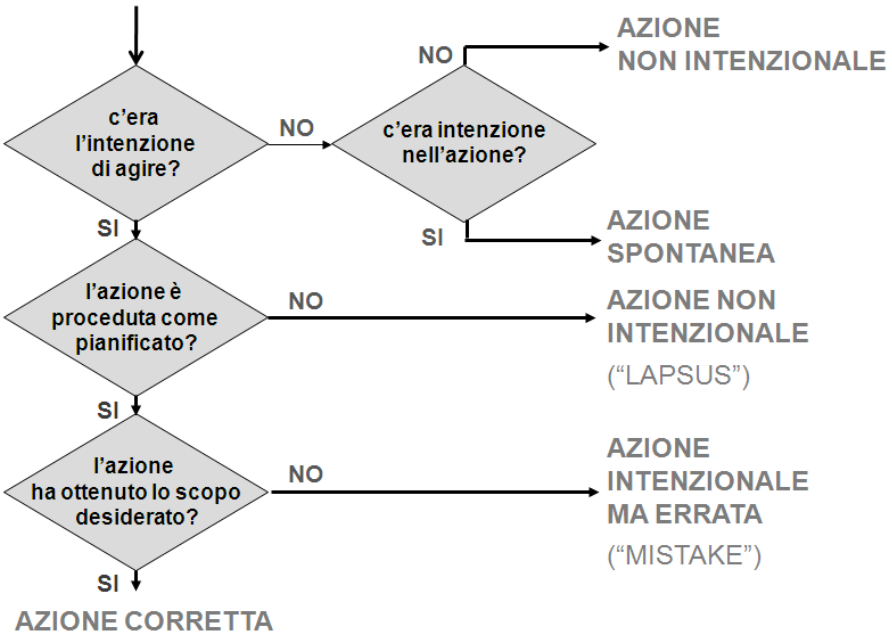


Figura 203. Classificazione degli errori

- Azione intenzionale ma errata (*mistake*)

  Per questo tipo di errore, in inglese si usa il termine *mistake*.[137] Esso si verifica quando l'utente ha agito con intenzione, l'azione si è svolta come aveva pianificato, ma non ha ottenuto lo scopo prefissato. In sostanza, l'utente ha compiuto un'azione credendo che portasse a un determinato risultato, ma così non è stato. Ad esempio, per accendere la luce in una stanza ha premuto l'interruttore sbagliato, credendo che fosse quello giusto. Aveva l'intenzione di agire (accendere la luce premendo un determinato interruttore), ha effettivamente premuto quell'interruttore, ma lo scopo non è stato raggiunto. Riprendendo il modello di Norman schematizzato in Figura 43 a pag.59, l'utente ha formulato l'intenzione corretta, ma ha specificato (ed eseguito) l'azione sbagliata.

- Azione non intenzionale (*lapsus*)

  Per questo tipo di errore si usa più propriamente il termine latino *lapsus* o, in inglese, *slip*, che significano, letteralmente, "scivolata". Si ha un lapsus quando, parlando o scrivendo, si sostituisce involontariamente una parola con un'altra *(lapsus linguae* o *lapsus calami*, nel caso della scrittura). Oppure, generalizzando, quando si compie involontariamente un'azione al posto di un'altra. Riprendendo l'esempio precedente, quando l'utente, volendo premere un determinato interruttore per accendere la luce, ha invece premuto involontariamente l'interruttore vicino. Tornando al modello di Norman, si ha quindi un lapsus quando l'utente, dopo avere formato l'intenzione corretta, e specificato un'azione corretta, ha invece eseguito l'azione sbagliata.

  I lapsus sono molto frequenti, e possono verificarsi soprattutto quando l'azione corretta e l'azione sbagliata "si assomigliano", per esempio quando due pulsanti sono fisicamente vicini. Oppure quando due compiti diversi hanno in comune una sequenza iniziale di azioni, e la sequenza finale in un caso viene eseguita di rado, e nell'altro molto spesso. Per esempio, se tutti i giorni si percorre in automobile la stessa strada per andare da casa all'ufficio, sarà facile, imboccando lo stesso percorso, "distrarsi" e arrivare davanti all'ufficio anche se la destinazione desiderata questa volta era diversa. Norman chiama questi tipi di lapsus "errori di cattura", perché una sequenza (quella più familiare) "cattura" l'altra.

  I lapsus possono essere evitati (o comunque resi poco probabili) progettando il sistema in modo che queste

[137] In italiano potremmo tradurre genericamente con *errore* l'inglese *error* e usare il termine *sbaglio* per *mistake*.

situazioni non si verifichino.

- Azione spontanea

  In questo caso, l'azione è compiuta intenzionalmente, ma senza che l'utente avesse precedentemente l'intenzione di agire. Per esempio, quando qualcuno ci lancia improvvisamente un oggetto e, quasi per un riflesso automatico, lo afferriamo al volo, o ci proteggiamo con le mani. L'azione non era prevista, ma ci siamo trovati nella necessità di compierla. Un'azione spontanea non necessariamente deve essere classificata come errore: è tale solo quando produce effetti indesiderati.

- Azione involontaria

  In questo caso, l'azione è del tutto non intenzionale. Per esempio, quando urtiamo involontariamente una persona oppure quando, mentre a tavola ci versiamo del vino, rovesciamo il bicchiere.

Per ciascuno di questi tipi di errore, gli effetti possono essere molto diversi. Se tocco inavvertitamente una persona sul tram, la cosa non ha grande importanza. Ma se tocco inavvertitamente il pallone con le mani nei pressi della porta durante una partita di calcio di serie A le conseguenze possono essere molto gravi. Come si vedrà meglio nel corso di questo capitolo, spesso non esiste una dicotomia netta fra errore e comportamento corretto.

Le strategie che il progettista deve mettere in atto per limitare la possibilità e le conseguenze dell'errore umano sono riassunte nella Figura 204. Innanzitutto, dovrà progettare il dialogo in modo tale che l'errore risulti impossibile, o comunque poco probabile (*prevenzione*). Nel caso in cui l'utente dovesse comunque commettere un errore, questo dovrebbe essere gestito. In particolare, individuato e spiegato correttamente all'utente (*diagnosi*), affinché egli possa correggere la situazione (*recovery*). Vediamo più in dettaglio le principali tecniche che possono essere utilizzate a questi scopi.

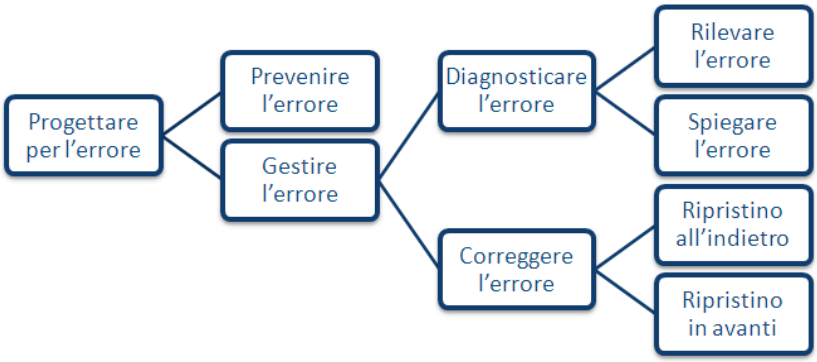


Figura 204. Progettare per l'errore

Prevenzione

Anche nei sistemi interattivi, prevenire è meglio che curare. Prevenire l'errore (*error prevention*) significa progettare il sistema in modo che la possibilità di errori da parte dei suoi utenti sia minima. In particolare, un'azione dell'utente non dovrebbe mai causare una caduta del sistema o un suo stato indefinito. Alcune tecniche molto diffuse per prevenire gli errori sono le seguenti:

- Diversificare le azioni dell'utente;
- Evitare comportamenti modali;
- Usare funzioni obbliganti;
- Imporre input vincolati;
- Non sovraccaricare la memoria a breve termine dell'utente;
- Richiedere conferme;
- Usare default inoffensivi.

Vediamole singolarmente.

Diversificare le azioni dell'utente

Questa tecnica serve a prevenire i lapsus. Si tratta di fare in modo che le azioni che l'utente deve eseguire per effettuare compiti diversi siano ben diversificate, in modo da minimizzare la probabilità che l'utente ne esegua inavvertitamente una al posto dell'altra. Per esempio, è bene distanziare fisicamente i pulsanti o le voci di menu di uso più frequente da pulsanti o voci di menu relative a comandi "pericolosi". Si tratta di una raccomandazione alquanto ovvia, che però non di rado è disattesa.

A volte questa indicazione è in conflitto con quella, altrettanto corretta, di raggruppare fra loro i comandi semanticamente correlati. La Figura 205 mostra un esempio tipico di questa situazione di conflitto. Il programma Outlook della Microsoft, come tutti i programmi di posta elettronica, associa a ogni messaggio che si trova nella mailbox di input lo stato di Read o di Unread, a seconda che esso sia stato letto o no dall'utente. I messaggi nello stato di Unread sono evidenziati visivamente (normalmente in neretto), per permettere all'utente di individuare le mail ancora da aprire. Nel menu Edit ci sono anche due comandi (Mark as Read e Mark as Unread) che permettono di modificare questi stati. Ciò è molto utile quando l'utente, dopo aver letto un certo messaggio, decide di non rispondere subito: per ricordarsi che la mail non è stata evasa, potrà usare il comando Mark as Unread, che la visualizzerà di nuovo in neretto. Il sistema fornisce anche un comando Mark All as Read, che pone *tutti* i messaggi presenti nella mailbox di input nello stato di Read. Il comando è collocato immediatamente sotto gli altri due, e può accadere di selezionarlo inavvertitamente al posto del contiguo Mark as Unread, con esiti molto fastidiosi. Infatti, il sistema non consente di annullarne gli effetti, e l'utente non avrà più modo di riconoscere i messaggi ancora da leggere da tutti gli altri. Questo capita di frequente a chi, come chi scrive, abbia l'abitudine di dare una prima scorsa alla posta per poi evaderla in un secondo tempo. Il problema potrebbe essere evitato, allontanando il comando pericoloso dagli altri due, oppure lasciandolo dove si trova ma chiedendo conferma all'utente prima di eseguirlo. Quest'ultima soluzione, nel caso specifico, è probabilmente la più corretta, perché lascia vicini tre comandi funzionalmente correlati.

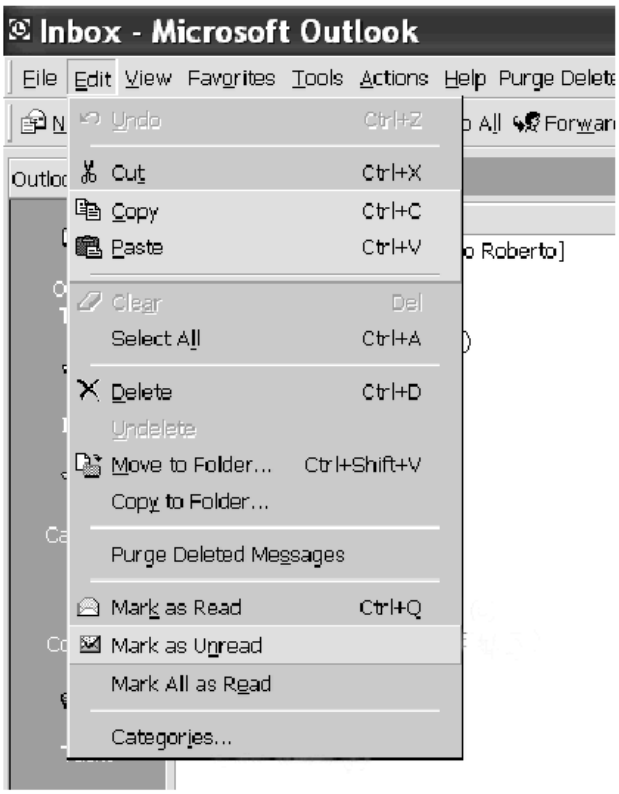


Figura 205. Un menu "pericoloso" (in Microsoft Outlook)

Evitare comportamenti modali

Chiamiamo *modale* un sistema che, a fronte di una stessa azione dell'utente, si comporta diversamente a seconda dello stato in cui si trova *e questo stato non è facilmente riconoscibile dall'utente*. Questo comportamento andrebbe sempre evitato: se l'utente non è in grado di identificare facilmente lo stato del sistema, sarà costretto a fare delle supposizioni, con un'elevata probabilità di commettere errori.

Un esempio classico è quello della funzione per il riconoscimento di una password in cui le minuscole sono considerate diverse dalle corrispondenti maiuscole. Una password contenente caratteri minuscoli, se digitata quando il sistema è nello stato di Caps Lock, viene rifiutata. In molti sistemi, però, lo stato di Caps Lock non è visibile: l'utente, di fronte a un rifiuto della password sarà perciò portato a credere di avere sbagliato a digitare, e proverà di nuovo, anche più volte. Raramente attribuirà il problema al tasto di Caps Lock, che è usato raramente (ma è "pericolosamente" vicino al tasto alzamaiuscole, che si utilizza invece molto spesso). Il problema può essere facilmente evitato rendendo ben visibile lo stato di Caps Lock, per esempio con un messaggio posto vicino al campo d'immissione, dove l'utente rivolge la sua attenzione, come nelle recenti versioni di Windows.

Un altro esempio spesso citato è vi, un text editor in passato molto diffuso in ambiente Unix. Esso poteva trovarsi negli stati di attesa carattere oppure di attesa comando. Digitando il carattere a, nel primo caso lo si aggiungeva al testo in corso di stesura, nel secondo caso il sistema entrava nello stato di append, e si metteva in attesa di un carattere successivo. Se l'utente dimenticava lo stato corrente, non era in grado di prevedere il comportamento del sistema. In effetti, lo stato era segnalato sul video, ma ai bordi, in modo scarsamente visibile perché lontano dal focus dell'attenzione dell'utente che, ovviamente, era concentrata sulla posizione del cursore nel testo o su quella delle dita sulla tastiera.

Il programma MacPaint della Apple, uno dei primi programmi di disegno, risolveva un problema analogo in modo elegante (Figura 206). L'utente poteva selezionare da una paletta lo strumento desiderato, e il sistema cambiava stato, per compiere le funzioni associate allo strumento. Il programma evidenziava lo strumento selezionato, e quindi lo stato del sistema, sulla paletta. Inoltre, con una tecnica adottata poi da molti programmi di questo tipo, il cursore assumeva la forma dello strumento prescelto (nel nostro esempio, la "matita", per tracciare linee a mano libera). In questo modo, lo stato del sistema era ben visibile proprio dove l'utente rivolge la sua attenzione, e cioè sul cursore. Questa tecnica non è usata in PowerPoint (Figura 207), dove il cursore mantiene sempre la forma di una croce, qualunque sia lo strumento selezionato. Poiché gli strumenti vengono selezionati da un menu a tendina, che subito si richiude, l'utente non ha modo di vedere lo stato del sistema, e quindi di sapere quale figura verrà tracciata muovendo il cursore.

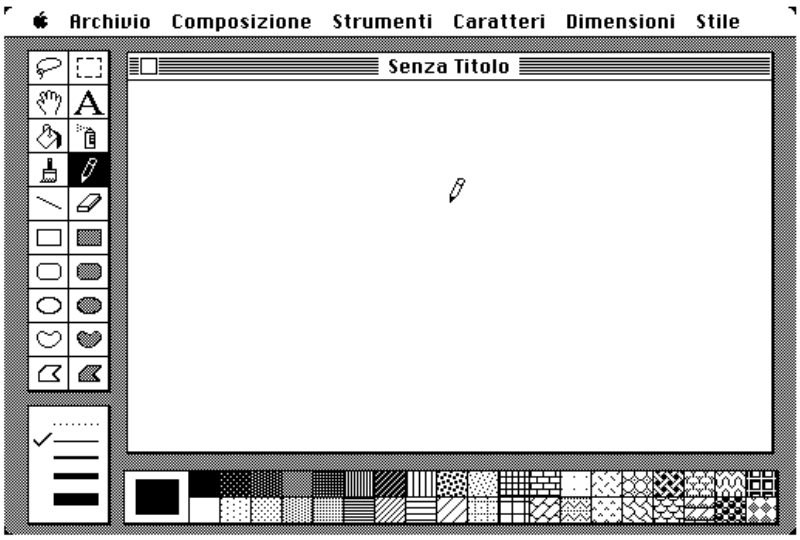


Figura 206. Cursore non modale (MacPaint, 1984)

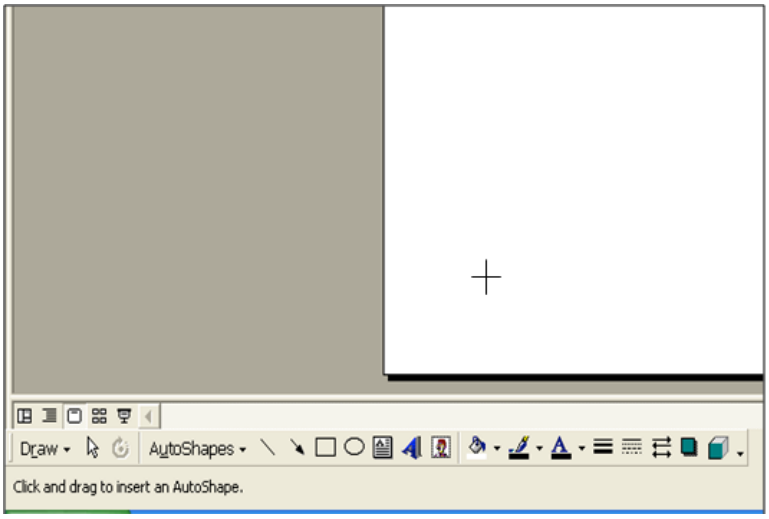


Figura 207. Cursore modale (da Microsoft PowerPoint 2003)

Usare funzioni obbliganti

Donald Norman definisce *obbligante* (in inglese: *forcing function*) una funzione in cui le azioni dell'utente sono vincolate in modo tale che la mancata esecuzione di un passaggio impedisca il successivo[138]. In tal modo, egli è obbligato a compiere le azioni nella sequenza corretta. Si tratta di una tecnica molto efficace di prevenzione degli errori.

Nella vita quotidiana incontriamo spesso delle funzioni obbliganti. A volte ci danno fastidio, perché limitano i nostri comportamenti, ma ci risparmiano problemi più gravi. Per esempio, alcune automobili emettono un segnale d'allarme quando apriamo la portiera con la chiave inserita nel cruscotto. Questo per evitare che si esca dall'auto dimenticando la chiave in macchina. Negli Stati Uniti, per un certo periodo le automobili montavano un dispositivo che impediva la partenza quando le cinture di sicurezza non erano allacciate. Il dispositivo risultò molto impopolare e fu in seguito eliminato, ma era sicuramente molto efficace per prevenire un comportamento errato del conducente.

A volte non ci rendiamo conto di utilizzare funzioni obbliganti. Per esempio, quando mettiamo in moto l'automobile con la chiave, in realtà compiamo con un solo gesto complesso tre operazioni: 1)- liberiamo il bloccasterzo, 2)- attiviamo il motorino di avviamento, 3)- mettiamo in moto. Se queste azioni fossero eseguibili separatamente (come si doveva fare un tempo), potremmo eseguirle nell'ordine sbagliato, con problemi evidenti.

Anche nei sistemi software le funzioni obbliganti sono importanti. Per esempio, a partire dal sistema Star della Xerox, il primo computer personale basato sulla metafora della scrivania, tutti i sistemi di questo tipo adottano il modello *oggetto-azione*, piuttosto che quello *azione-oggetto*. In altre parole, l'utente seleziona prima l'oggetto su cui desidera operare (per esempio, cliccando sull'icona di un file), e poi la funzione che vuole compiere (per esempio, il comando Stampa per stampare il file selezionato). Questa scelta, apparentemente arbitraria, permette di realizzare funzioni obbliganti. Infatti, selezionando prima l'oggetto su cui operare, il sistema è in grado di disabilitare tutte quelle voci di menu che corrispondono ad azioni che non hanno senso su tale oggetto. Se saltiamo il primo passaggio (la selezione dell'oggetto), il secondo non può essere eseguito, perché la voce corrispondente del menu è disabilitata. Questo non sarebbe stato possibile selezionando prima il comando e poi il suo argomento. Così, in Figura 208, che rappresenta il primo Macintosh della Apple, tutte le voci che operano su file (Apri, Stampa, ecc.) sono disabilitate, perché nessun file è stato selezionato.

[138] Cfr. Donald Norman, *La caffettiera del masochista*, Giunti, 1990, pag.171 e segg.

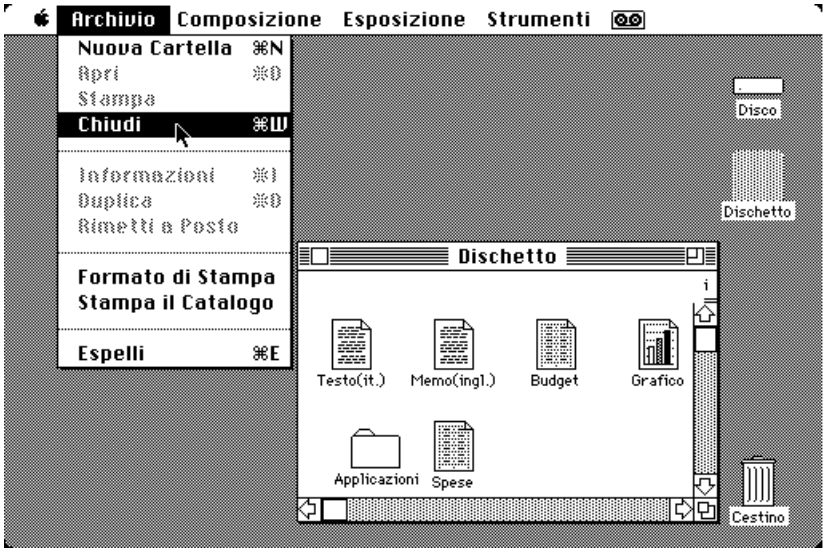


Figura 208. Finder (Apple Macintosh, 1984)

Imporre input vincolati

Questa tecnica, che generalizza quella delle funzioni obbliganti, consiste nel permettere all'utente di fornire solo valori di input corretti.

In Figura 209 vediamo diversi modi di chiedere l'immissione di una data. In (a), il sistema non pone alcun vincolo al formato della data. In (b), (c) e (d) il sistema pone vincoli via via più stringenti. Quale soluzione è in grado di prevenire meglio gli errori di digitazione? Possiamo facilmente scartare la (a), perché troppo libera, e la (b), perché la (c), più informativa, è sicuramente migliore. La (c) lascia ancora molta libertà all'utente, che può digitare date formalmente scorrette. Nella (d), invece, i valori selezionabili dai tre menu a tendina sono preimpostati, e quindi sempre corretti. Non è però ovvio quale delle due ultime soluzioni sia la migliore. Per deciderlo, non basta un'analisi "a tavolino": dovremmo fare delle prove d'uso con un sufficiente numero di utenti. Infatti, la (d), che a prima vista ci sembrerebbe la più sicura, evita sicuramente che l'utente digiti una data illecita, ma introduce un altro problema, che nell'altra soluzione non si pone. In un menu a tendina molto lungo il mouse può "scivolare" facilmente su una voce vicina a quella desiderata, soprattutto quando si deve selezionare una delle voci più in basso. Nel nostro esempio, le voci dei mesi sono soltanto 12, ma quelle dei giorni 31, un numero abbastanza elevato. Quante sono quelle degli anni? Si inizia dal 1900? E qual è l'anno impostato come default nel campo? Se fosse l'anno meno recente, la probabilità di dover selezionare una voce molto lontana sarebbe alta, e la probabilità di errore più elevata.

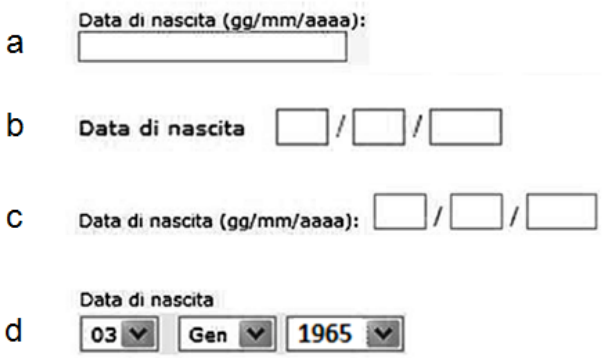


Figura 209. Esempi di input più o meno vincolati

Non sovraccaricare la memoria a breve termine

Nel capitolo 4 (pag.91) sono state brevemente descritte le caratteristiche della memoria a breve termine dell'uomo, in particolare la sua capacità limitata e la breve persistenza dell'informazione. Dialoghi che sovraccaricano la memoria a breve termine risultano faticosi per l'utente, e aumentano la probabilità che egli commetta degli errori. Tipico è il caso dei messaggi pre-registrati dei call-center. A chi telefona viene letto un elenco di servizi disponibili, che spesso è troppo lungo per essere facilmente ricordato. Quando, durante l'enumerazione delle alternative disponibili, l'utente ne identifica una che potrebbe ragionevolmente fare al caso suo, la deve memorizzare e proseguire nell'ascolto, nell'eventualità che ne vengano annunciate altre ancora più pertinenti. Arrivato alla fine dell'enumerazione, non è raro il caso che, avendo dimenticato le alternative presentate all'inizio, l'utente debba rifare la telefonata e ascoltare da capo l'elenco.

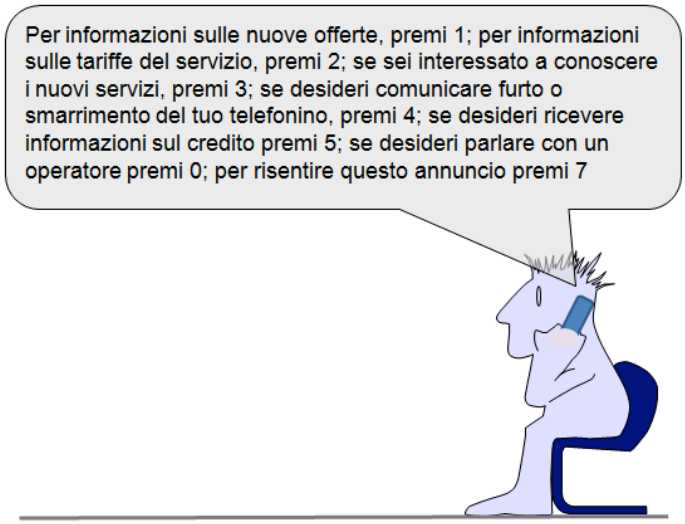


Figura 210. Sovraccarico della memoria a breve termine

Richiedere conferme

Il sistema dovrebbe sempre avvertire l'utente quando questi richiede l'esecuzione di azioni irreversibili o comunque potenzialmente pericolose, e domandare conferma.

Le richieste di conferma devono essere formulate in modo semplice e non ambiguo. Un esempio corretto è mostrato in Figura 211, in cui il messaggio è chiaro e le tre alternative ben distinte. Questo non è il caso del messaggio di Figura 211b, causata dalla richiesta di uscire da un computer game senza avere prima salvato lo stato del gioco. Sarebbe preferibile una forma più esplicita, che spiegasse meglio gli effetti delle tre alternative, senza usare negazioni. Per esempio: Warning: the game was not saved e, sui tre pulsanti: Exit without saving – Save and exit – Cancel.

Il sistema non deve costringere l'utente a conferme troppo complicate, come quella di Figura 211c: qui non ci si accontenta di richiedere all'utente un semplice clic su un pulsante di OK, ma lo si obbliga a digitare le tre lettere che compongono la parola YES. Evidentemente, il progettista ha pensato che, così facendo, si elimina la possibilità che l'utente confermi per errore, ma la soluzione può essere considerata alquanto vessatoria.

Le richieste di conferma dovrebbero essere limitate ai soli casi di operazioni per le quali l'eventuale annullamento fosse impossibile, o richiedesse molto tempo. È infatti inutile intralciare il lavoro dell'utente richiedendogli di confermare operazioni che, se effettuate per errore, potrebbero essere facilmente annullabili.

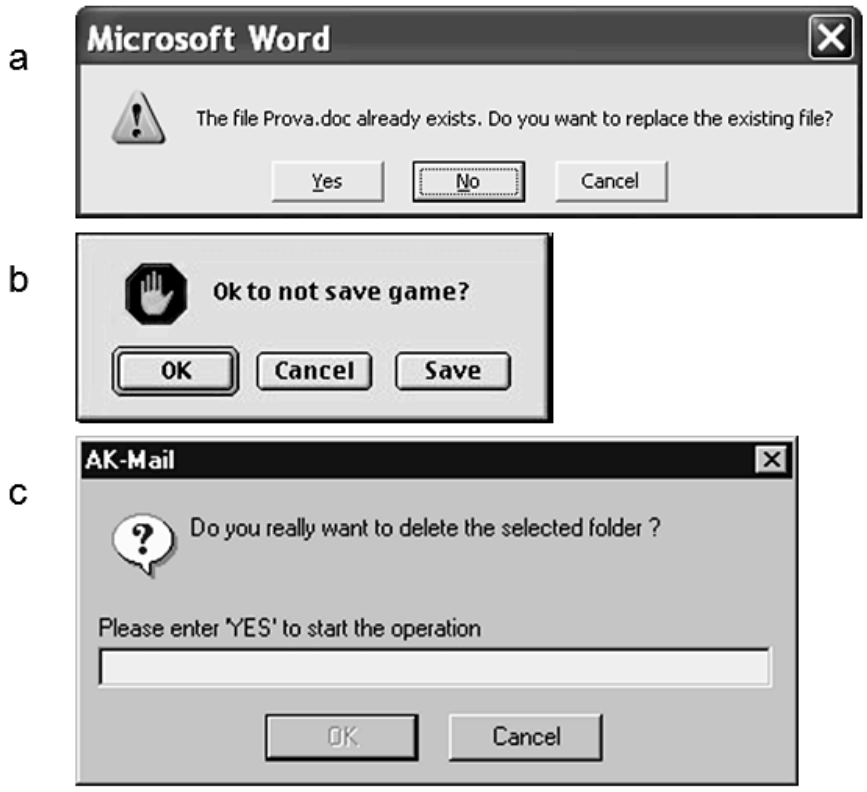


Figura 211. Alcune richieste di conferma, da sistemi degli anni '90:
a) da Microsoft Word, b) da un computer game per Apple Macintosh, c) da AK-Mail per PC

Usare default inoffensivi

Questa tecnica consiste nell'usare, per quanto possibile, valori di default inoffensivi. Il sistema, in altre parole, non dovrebbe mai intraprendere per default l'azione più pericolosa tra quelle possibili in un determinato contesto. Per esempio, la richiesta di stampa di un documento di Microsoft Word 2007 ha come default l'opzione Pagine da stampare = Tutte. Non viene richiesta alcuna conferma, nemmeno nel caso di documenti molto lunghi. A chi scrive è capitato varie volte, durante la stesura di questo libro, di avviare per errore la stampa di tutto il documento (un file di oltre 200 pagine!), invece che della sola pagina corrente, come desiderato. Un lapsus costoso, le cui conseguenze avrebbero potuto essere evitate da una semplice richiesta di conferma.

Diagnosi

Anche se il progettista adotta le tecniche di prevenzione più appropriate, resterà sempre la possibilità che l'utente commetta un errore. Pertanto, il sistema deve sempre controllare l'input e, nel caso che si riveli scorretto, dovrà fornire all'utente una spiegazione adeguata, che gli permetta di recuperare la situazione in modo rapido, senza incertezze e senza stress.

Sono tre le funzioni che un messaggio di errore ben progettato deve svolgere:

- *Allertare*, cioè segnalare che qualcosa non va;
- *Identificare*, cioè indicare che cosa non va, e perché;
- *Dirigere*, cioè spiegare all'utente i passi che deve compiere per ripristinare una situazione corretta: "ora devi fare questo".

In Figura 212 vediamo alcuni esempi di messaggi di errore. In (a), sono presenti, anche se in forma molto sintetica, le tre funzioni sopra indicate. L'utente viene allertato per mezzo di un'icona ben visibile (un grosso punto di domanda su

fondo bianco), l'errore viene correttamente identificato con un testo breve ma chiaro (The selected printer is not available), e viene indicata la possibilità di selezionare un'altra stampante: Do you want to select a different printer? Non si dice ancora come fare, si chiede soltanto di specificare le intenzioni: presumibilmente, in caso di risposta affermativa, seguiranno ulteriori indicazioni. Nel complesso il messaggio è accettabile, anche se, a ben vedere, descrive "che cosa" non va, ma non spiega il perché. Infatti, l'indisponibilità della stampante potrebbe avere cause diverse, tipicamente: 1)- l'utente ha dato il comando di stampa senza modificare la stampante di default, che non corrisponde a quella collegata al sistema, oppure 2)- la stampante selezionata è quella corretta, ma è spenta. Anche se il software non fosse in grado di discriminare fra queste due situazioni, potrebbe almeno indicare nel messaggio il nome della stampante selezionata: The selected printer: <name> is not available. In questo caso l'utente potrebbe comprendere immediatamente la causa del problema e porvi rimedio.

Il messaggio di Figura 212b corrisponde a una situazione di errore simile alla precedente (la stampante richiesta non è disponibile). La scelta del colore dell'icona di allerta (giallo) vuole correttamente segnalare che la situazione è probabilmente recuperabile. Il messaggio d'identificazione dell'errore (Printer not found) è sintetico ma chiaro. Le opzioni proposte all'utente sono però incomprensibili. L'opzione Retry non ha senso se lo stato del sistema non cambia – per esempio accendendo la stampante nel caso fosse spenta. Ma all'utente non viene data alcuna indicazione in tal senso: se egli non fa nulla per correggere la situazione, la riesecuzione del comando di stampa non può che portare allo stesso risultato. Osserviamo le altre due opzioni: qual è la differenza fra Ignore e Abort? La prima, probabilmente, corrisponde a una presa d'atto del messaggio e, implicitamente, alla richiesta di tornare allo stato precedente alla richiesta di stampa. Ma allora sarebbe stato meglio usare il termine OK, oppure Cancel, a seconda delle convenzioni utilizzate negli altri messaggi dello specifico sistema. Se questa interpretazione del messaggio è corretta, allora Abort è inutile, e non corrisponde ad alcun'altra situazione possibile. L'intestazione del messaggio (System error) è poi incomprensibile. Chi ha richiesto una stampante che non c'è: l'utente o il sistema?

L'ultimo esempio (Figura 212c) usa correttamente un'icona di colore rosso, per segnalare che è stato rilevato un errore grave. Il simbolo usato dall'icona (una croce a x, come quelle usate per segnalare i passaggi a livello) ci sembra più corretto del punto esclamativo del messaggio precedente, che possiede una certa connotazione di rimprovero. Ma la descrizione dell'errore – che identifica analiticamente la causa del problema - è espressa in linguaggio tecnico, comprensibile solo ai programmatori che hanno sviluppato l'applicazione.

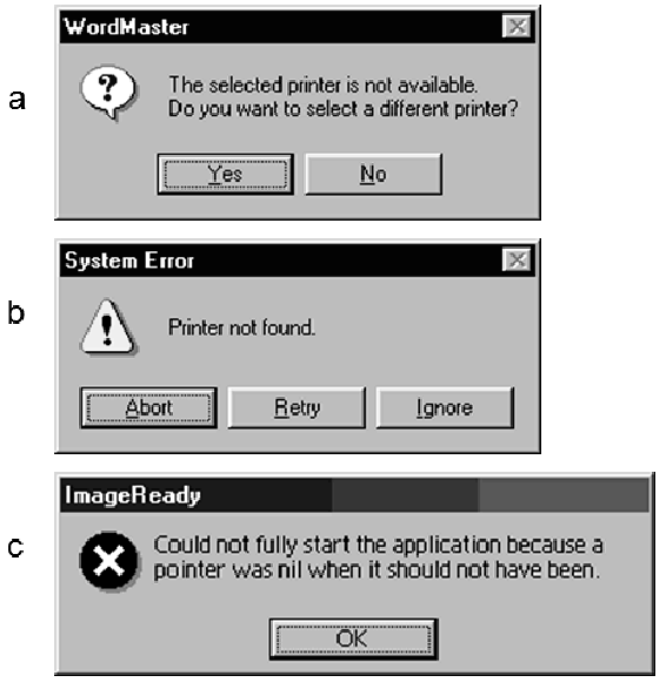


Figura 212. Messaggi di errore

Come abbiamo già osservato a pag.219, è molto importante che il messaggio non contenga alcun rimprovero (anche se sottinteso) all'utente. Al contrario, molte applicazioni web dell'ultima generazione adottano la politica opposta: il sito sembra chiedere scusa all'utente per averlo portato a commettere un errore (*self-deprecating error messages*), come nell'esempio di Figura 213, o di Figura 190 a pag.220 (esempio in basso a destra).



Figura 213.

Tutti gli esempi discussi si riferiscono a *singoli* errori. Sono frequenti, tuttavia, situazioni più complesse, in cui l'utente compie più di uno sbaglio. L'esempio più tipico è la compilazione di una form in un'applicazione web. L'utente immette i valori in diversi campi, e quindi ne chiede l'invio al sistema, che li dovrà controllare segnalando gli eventuali errori. Poiché più di un valore può essere errato, ne possono seguire messaggi di errore multipli, che devono possedere un'adeguata "granularità" ed essere facilmente associabili ai valori cui si riferiscono.

La Figura 214, tratta da un sito di commercio elettronico di molti anni fa, mostra un esempio prodotto artificialmente, introducendo a bella posta un errore in ogni campo della form. In questo caso, all'utente risulta molto difficile associare il messaggio di errore al campo cui si riferisce. Infatti, il messaggio copre interamente la form in cui sono stati commessi gli errori. I messaggi sono troppo numerosi per essere facilmente ricordati (si ricordino le limitazioni della memoria a breve termine). In pratica, l'utente è costretto a prendere appunti sulle correzioni da effettuare, prima di tornare alla form di imputazione dati.



Figura 214. Messaggi di errore multipli (da un sito di e-commerce, anni '90)

La situazione di Figura 215 è più corretta, anche se perfettibile. In questo caso, l'elenco dei numerosi errori rilevati è presentato mediante un piccolo pannello sovrapposto alla form d'immissione dati, e facilmente spostabile sullo schermo per rendere visibili tutti i campi.

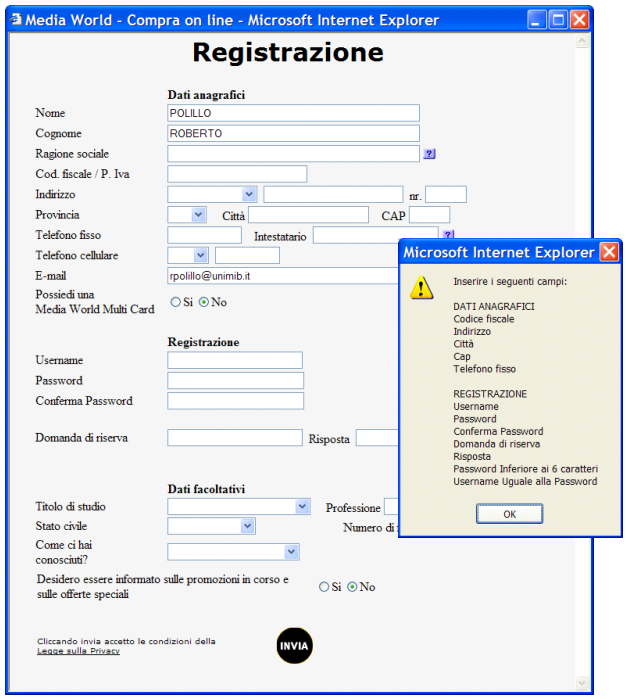


Figura 215. Messaggi di errore multipli (www.mediaworld.it, fine anni '90)

La soluzione di gran lunga più corretta è tuttavia quella esemplificata in Figura 216. In questo caso la segnalazione di errore è visualizzata immediatamente sopra il campo errato (per maggiore evidenza, il messaggio è in colore rosso vivo).

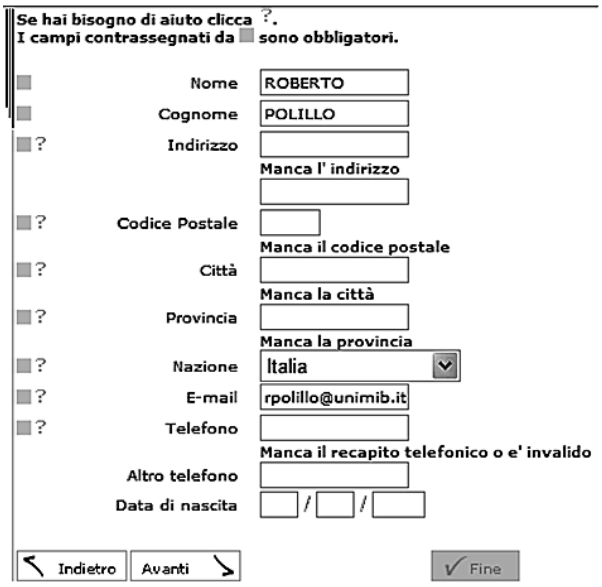


Figura 216. Tecnica corretta per la segnalazione di errori su una pagina web

La spiegazione dell'errore e di come fare per correggerlo dovrebbe essere particolarmente dettagliata e chiara quando il sistema si rivolga a utenti poco esperti. Particolarmente interessante, a questo proposito, lo stile adottato da Amazon nei suoi primi anni di esistenza, quando i siti di commercio elettronico erano ancora poco noti alla maggior parte degli utenti del Web. Le spiegazioni erano piuttosto estese, e spesso contenevano anche le motivazioni delle richieste fatte all'utente, come nell'esempio di Figura 217.



Figura 217. Da http://www.amazon.com (circa 1995)

Per evitare una verbosità eccessiva, può convenire esprimere il messaggio in forma sintetica, permettendo però all'utente di richiedere spiegazioni più dettagliate.

Correzione

Quando l'utente commette un errore, deve essere possibile correggerlo. Questo processo può essere attuato, secondo i casi, dall'utente o dal sistema, o da entrambi, in modo cooperativo. Le situazioni possibili sono schematizzate nella Figura 218.[139] A partire da uno stato iniziale del sistema, l'utente compie un'azione, per esempio l'immissione del valore di una data in un campo di una form. Se l'azione è corretta, il sistema si porta nello stato desiderato, indicato in figura come stato finale. Per esempio, la data viene accettata e registrata in un file. Se invece l'azione non è corretta (per esempio, perché la data contiene degli errori formali), il sistema si porta in uno stato di errore. A partire da questo stato, il processo di correzione può avvenire con due strategie diverse, a seconda delle situazioni. Vediamole in dettaglio.

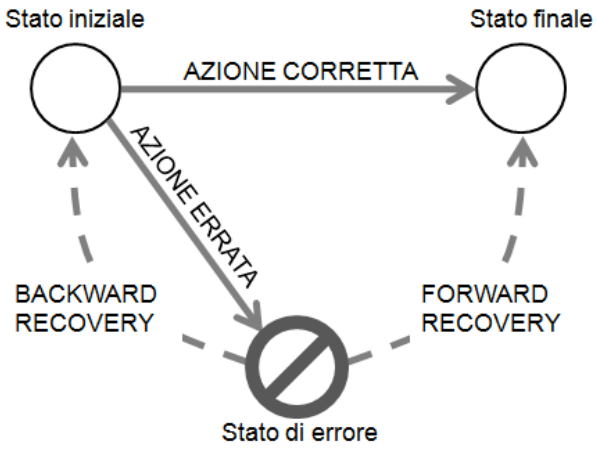


Figura 218. Ripristino dall'errore

Backward recovery

Secondo questa strategia, si tratterà di annullare le conseguenze negative dell'errore commesso, e riportare il sistema nello stato iniziale, dal quale l'utente potrà compiere, questa volta in modo corretto, l'azione che aveva sbagliato. Il processo per riportare il sistema nello stato iniziale si chiama *ripristino* - in inglese, *backward recovery.*[140] Nel caso

[139] Cfr. Francis Jambon, *Taxonomy for Human Error and System Fault Recovery from the Engineering Perspective*, *International Conference on Human-Computer Interaction in Aeronautics (HCI-Aero'98),* Montréal, Canada, 27-29 may 1998. pagg. 55-60.

[140] Recovery significa recupero, riparazione, reintegro.

elementare della data sbagliata, la backward recovery consisterà semplicemente nel cancellare dal campo di input il valore scorretto. Questo compito può essere svolto dall'utente oppure effettuato dal sistema, che sbianca il campo e posiziona il cursore all'inizio dello stesso, in attesa del nuovo input che – si spera – porterà allo stato finale desiderato.

In altri casi gli interventi di backward recovery possono essere più complessi. Per esempio, supponiamo che l'utente digiti una data formalmente corretta, ma diversa da quella che voleva immettere nel sistema e che si accorga del lapsus solo dopo avere confermato i dati. In questo caso, il sistema avrà accettato il valore, e lo avrà registrato nella base dati. La backward recovery richiederà quindi di modificare la base di dati: un'azione molto diversa da quella di prima.

La backward recovery può essere considerata un modo di "tornare indietro nel tempo". Il modo più semplice di farlo è quello di utilizzare una funzione di undo, che annulli le conseguenze dell'ultima azione. I sistemi più evoluti forniscono la possibilità di annullare le ultime n azioni dell'utente, con n anche molto grande (undo a più livelli).
La realizzazione di funzionalità di undo non è sempre possibile, o praticabile. Infatti, possono essere necessarie grandi quantità di memoria (per conservare gli stati precedenti), e gli algoritmi di ripristino possono essere molto complessi. Tuttavia, in alcune tipologie di sistemi, meccanismi di undo sofisticati sono indispensabili, come per esempio nei programmi di grafica. La Figura 219 mostra le funzioni di undo in PowerPoint 2007 (a) e Photoshop CS3 (b). In entrambi, un pannello mostra l'elenco delle ultime azioni effettuate dall'utente. In PowerPoint le azioni più recenti sono in alto, mentre in Photoshop sono in basso. In entrambi i casi è possibile annullare le ultime n azioni con un solo comando. In (a), sono state selezionate le ultime 4 azioni, che verranno annullate al rilascio del mouse. In (b), selezionando un'azione, vengono annullate tutte le azioni successive. In entrambi i sistemi l'undo può, a sua volta, essere annullato (funzione di redo).

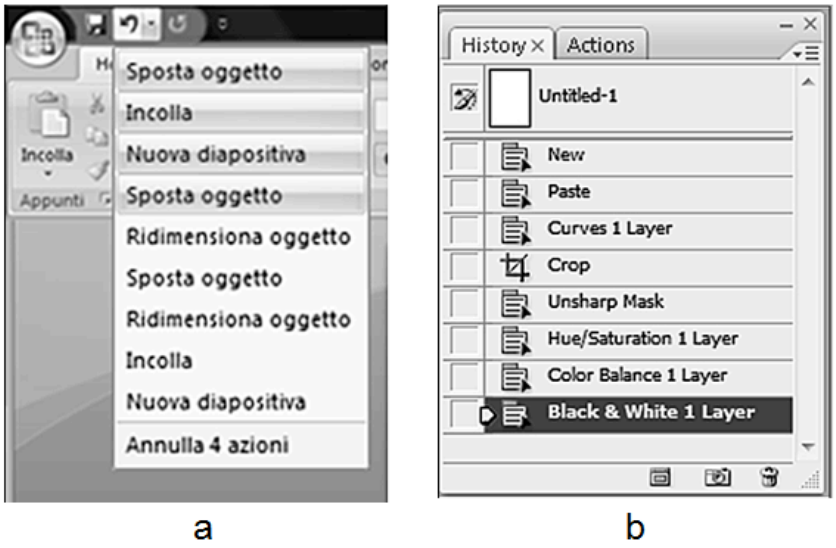


Figura 219. Undo a più livelli: PowerPoint 2007 (a) e Photoshop CS3 (b)

Forward recovery

Esistono situazioni in cui la correzione dell'errore avviene con un processo diverso. Invece di tornare allo stato iniziale e da lì ripetere l'azione, si cerca di raggiungere lo stato finale direttamente dallo stato di errore, *senza prima tornare indietro*. Questo processo si chiama *forward recovery* (Figura 218), e può essere effettuato automaticamente dal sistema, o richiedere l'intervento dell'utente. Un sistema che attua sistematicamente strategie di forward recovery si dice *error tolerant*.

La Figura 220 mostra il comportamento del motore di ricerca di Google quando l'utente compie un errore di battitura. In questo caso l'utente ha digitato nel campo di ricerca la parola chiave usibilità, invece di usabilità. Poiché la parola digitata non esiste, il sistema la corregge e la interpreta come usabilità*:* senza nulla chiedere preventivamente all'utente, porta a termine la ricerca con questa parola. L'utente è avvertito a posteriori: "Forse cercavi usabilità".

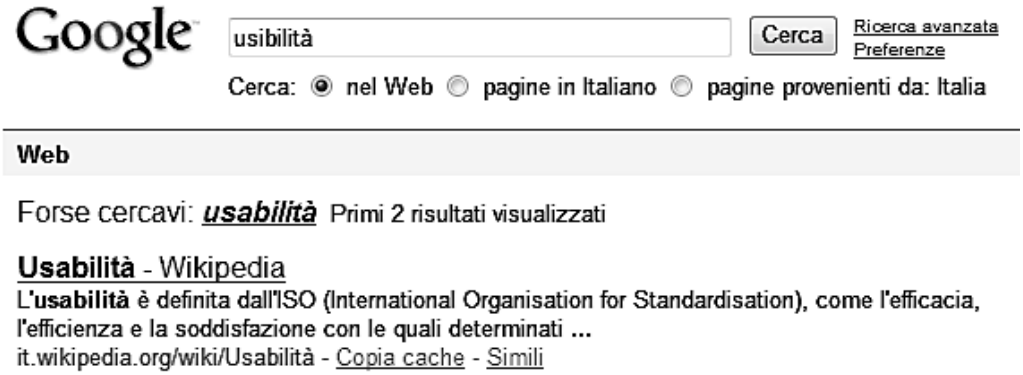


Figura 220. Forward recovery in http://www.google.com

La forward recovery può essere attuata anche dall'utente. Supponiamo di dover disegnare un quadrato, con PowerPoint o altro programma di grafica. Normalmente, occorre selezionare lo strumento "rettangolo" e, tenendo premuto il tasto Shift, tracciare la figura sul foglio visualizzato. Se l'utente si dimentica di tenere premuto il tasto Shift, la figura risultante sarà rettangolare. Per ottenere il quadrato potrà allora scegliere fra due strategie alternative: 1)- cancellare il rettangolo e ricominciare il disegno (backard recovery) oppure 2)- conservare il rettangolo e modificandolo fino a trasformarlo in un quadrato (forward recovery).

Ci sono casi in cui la backward recovery non è possibile, e l'unica strategia disponibile è la forward recovery. Se per esempio rompiamo un piatto, non saremo evidentemente in grado di ripristinarlo nella configurazione iniziale: potremo soltanto tentare di aggiustarlo incollando i vari pezzi.

I due ultimi esempi mostrano che, in molti casi, la recovery è imperfetta: si può solo arrivare a un'approssimazione dello stato desiderato. Un piatto aggiustato con la colla non è sicuramente uguale a un piatto integro, e un quadrato disegnato "a mano libera" potrebbe essere impreciso. In entrambi i casi, non si raggiungerà lo stato finale inizialmente desiderato, ma uno stato finale approssimato (Figura 221).

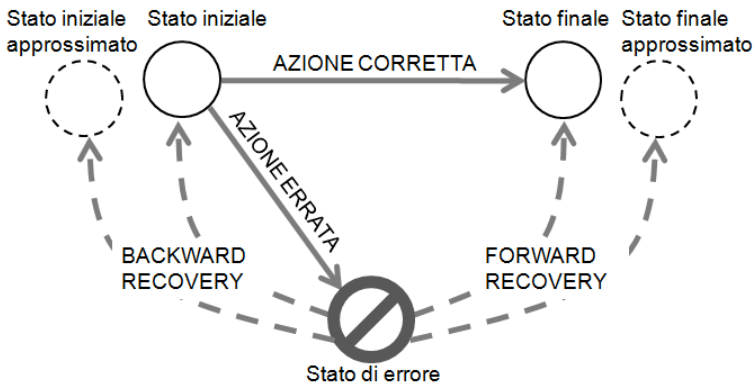


Figura 221. Ripristino imperfetto

In ogni caso, che si tratti di recovery all'indietro o in avanti, sarà bene che il progettista ricordi le raccomandazioni seguenti, che abbiamo già menzionato nel capitolo dedicato all'ISO 9241-110 (pag. 227):

- Minimo sforzo di correzione: i passi richiesti all'utente per correggere l'errore dovrebbero essere il più possibile semplici, e il sistema dovrebbe porre in atto ogni accorgimento per ridurne il numero e la complessità, svolgendo automaticamente quelle operazioni che, per loro natura, non richiedono l'intervento umano.

- Correzione differibile: l'utente dovrebbe poter rimandare la correzione dell'errore a un momento successivo. Questo, in virtù del principio di controllabilità, di cui abbiamo ampiamente trattato a pag. 225 e segg. Può darsi, per esempio, che l'utente non possegga tutte le informazioni per correggere immediatamente i dati di input che il sistema non ha accettato.
- Correzione automatica modificabile: quando il sistema è in grado di correggere automaticamente l'errore commesso dall'utente, dovrebbe avvisarlo della correzione e permettergli di modificarla. Così, per esempio, se un correttore ortografico modifica la parola digitata dall'utente, l'utente dovrebbe esserne avvertito e poterne modificare la correzione.

Conclusioni

Il senso di questo capitolo è che, nel dialogo con un sistema interattivo, non esiste una dicotomia netta fra comportamento errato e comportamento corretto. Parafrasando ancora una volta Donald Norman, tutta l'interazione uomo-macchina dovrebbe essere trattata come una procedura cooperativa fra utente e sistema, dove gli equivoci possono nascere da entrambe le parti, e devono essere risolti con chiarezza e serenità. L'errore è parte integrante del comportamento umano, e come tale deve essere previsto e accettato.

Un sistema ben progettato non deve pretendere da chi lo usa una conformità perfetta a regole precise e non modificabili, dovrà invece indicargli, nei modi e nei contesti più opportuni in funzione delle diverse circostanze, che cosa può o deve fare per raggiungere i suoi scopi, adattandosi in modo flessibile e "intelligente" alle eventuali deviazioni che, in ogni caso, saranno numerose e frequenti. Questo è il senso del principio di tolleranza verso l'errore che lo standard ISO 9241-110 richiede ai sistemi usabili. Rispetto alla prassi corrente nell'ingegneria del software, e nonostante i grandi progressi che negli ultimi anni sono stati compiuti nell'usabilità dei sistemi, tutto ciò richiede ancora un profondo cambiamento di mentalità. La prevenzione e il trattamento degli errori è una componente ineliminabile e importante del lavoro di progettazione di un sistema, e il progettista non la deve considerare un *optional*, dedicato ad alcuni utenti particolarmente inaccurati o distratti.

Ripasso ed esercizi

1. Descrivi la classificazione dell'errore umano secondo Reason, indicando un esempio per ciascuna tipologia di errore.
2. Spiega la differenza fra lapsus e mistake, e indica due esempi di ciascuna tipologia di errore, tratti dalla tua esperienza personale dell'uso di qualche sistema interattivo.
3. Che cosa significa "progettare per l'errore"?
4. Spiega che cosa si intende per comportamento modale di un'interfaccia utente, e perchè tale comportamento va evitato. Indica un esempio d'interfaccia modale, e spiega come questa interfaccia possa essere resa più usabile.
5. Che cosa s'intende per funzione obbligante? Descrivine due esempi diversi da quelli indicati nel testo.
6. Perchè, per prevenire errori, è necessario non sovraccaricare la memoria a breve termine dell'utente?
7. Quando è opportuno chiedere conferma all'utente prima di eseguire una funzione da lui richiesta, e quando non lo è?
8. Quali sono le caratteristiche di un buon messaggio di errore?
9. Quali sono le caratteristiche di un buon messaggio di errore per transazioni sul Web?
10. Spiega la differenza fra forward error recovery e backward error recovery, con esempi.
11. Valuta la gestione degli errori nel tuo telefonino, e formulane un motivato giudizio dal punto di vista dell'usabilità.

Approfondimenti e ricerche

1. Il Web è ricco di esempi di messaggi d'errore mal progettati. Costruisci una galleria di esempi di messaggi "sbagliati", raccogliendoli in categorie tipiche. Suggerimenti: cerca con Google, per esempio, "error message guidelines" e immagini con parole chiave "error message".

2. Donald Norman, nel suo libro *La caffettiera del masochista*, classifica i lapsus in varie tipologie. Dopo aver letto questa classificazione, trova un esempio di ciascuna tipologia di lapsus nell'uso dei programmi che usi abitualmente con il tuo personal computer.
3. Esamina le modalità di prevenzione e di gestione degli errori dell'utente in tre siti di commercio elettronico appartenenti allo stesso settore di mercato, e individuane pregi e difetti. Quale dei tre siti ha il miglior trattamento degli errori? Motiva dettagliatamente la tua risposta.

12. Progettare la grafica

Sintesi del capitolo

Questo capitolo descrive alcune linee guida per la progettazione d'interfacce grafiche. Dopo avere osservato che gli obiettivi che possono guidare il visual designer possono essere diversi, ci si concentra in particolare sull'obiettivo dell'usabilità. Si introducono le leggi della Gestalt, che descrivono le modalità con le quali l'apparato visivo umano segmenta il campo visivo raccogliendo in gruppi gli elementi visivi che lo compongono. Si mostra quindi, attraverso la discussione di numerosi esempi, come le leggi della Gestalt possono guidare utilmente il progettista di sistemi interattivi nella realizzazione di soluzioni grafiche di facile comprensione. Gli esempi riguardano soprattutto l'opportunità di avvicinare elementi visivi semanticamente e funzionalmente correlati, dando loro un aspetto simile e separandoli visivamente dagli altri elementi con l'uso di riquadri o cornici, e creando un ordine visivo mediante un uso consapevole del colore e degli allineamenti.

Design dell'interazione e comunicazione visiva

La comunicazione visiva ha un ruolo fondamentale nell'interazione uomo-macchina: la maggior parte delle informazioni che un sistema trasmette al suo utente sono veicolate da display video di varia forma e dimensione, dai grandi monitor che tappezzano le pareti delle control room dei grandi impianti di processo, fino ai piccoli display dei telefoni cellulari o degli altri dispositivi mobili in circolazione. L'usabilità di questi sistemi dipende quindi in modo considerevole dalla loro interfaccia grafica.

La grafica dei sistemi interattivi può perseguire obiettivi diversi: la comprensibilità delle informazioni, l'usabilità, la gradevolezza complessiva, l'originalità, la capacità di suscitare emozioni. Occorre grande chiarezza nella definizione dei requisiti del progetto, perché ciascuno di questi obiettivi richiede approcci e soluzioni differenti, che potrebbero essere fra loro in conflitto. Certamente, secondo gli scopi prefissati, i risultati della progettazione saranno molto diversi. Consideriamo la Figura 222a, che mostra la home page del sito della Fanta di qualche anno fa, all'interno del sito italiano della Coca Cola. Anche se non si capisce subito, si tratta di un menu: le varie voci appaiono, una a una, esplorando con il mouse il piede e la mano delle ragazze nelle zone indicate dal tratteggio. La Figura 222b mostra infatti la scritta che compare quando il puntatore del mouse passa sopra la zona destra del piede in primo piano. Il design è divertente e originale, ma certamente lascia molto a desiderare dal punto di vista dell'usabilità. L'utente non ha mai una visione d'insieme dei contenuti del sito: il menu è visibile sempre e solo a pezzi, e per leggerne tutte le voci occorre esplorare col mouse una diecina di aree sensibili diverse.



Figura 222. Home page di www.fanta.it (2003)

Tutto questo può non creare problemi per un sito destinato soprattutto a funzioni d'intrattenimento, ma potrebbe essere molto negativo in un sito con uno scopo diverso: tutto dipende dagli obiettivi. L'originalità del design è spesso un obiettivo prioritario per il sito web di uno stilista di moda, ma sarebbe probabilmente considerata controproducente per il sito di un'officina meccanica che voglia trasmettere un'impressione di economicità e concretezza. La capacità di suscitare forti emozioni potrebbe essere importante per la grafica di un computer game, ma certamente non per un sistema informativo aziendale.

Usabilità, comprensibilità, originalità, gradevolezza, emozione sono attributi in larga misura indipendenti. Un sistema può avere un'interfaccia comprensibile e non essere usabile, può essere gradevole ma non suscitare emozioni. Oppure può godere contemporaneamente di tutte queste caratteristiche (Figura 223).

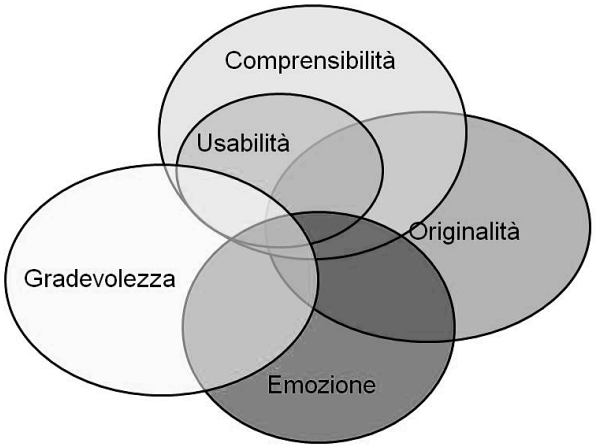


Figura 223. I diversi obiettivi della grafica di un sistema interattivo

Nei primi anni di questo secolo, a seguito della pubblicazione (nel 2000) del libro *Designing Web Usability* di Jacob Nielsen, ci fu un ampio dibattito fra i fautori dell'usabilità dei siti web, da un lato, e della libertà espressiva, dall'altro. I primi rimproveravano ai secondi di perseguire una creatività libera da ogni vincolo, spesso a discapito della facilità d'uso. I secondi obiettavano che l'usabilità perseguita ad ogni costo avrebbe, in ultima analisi, limitato la libertà di comunicazione che la rete permetteva:

> *Dietro la semplificazione della navigazione s'intravede la trasformazione della rete in una sorta di percorso prestabilito che segue strade precostituite verso destinazioni che poi sono facilmente intuibili: comprare, comprare comprare. 'Making things easy' (facilitare le cose) è il principio guida per la trasformazione della rete in un sistema di potere economico e politico rigido, automatico, inevitabile. Se riduciamo Internet a un sistema pavloviano di domande prevedibili e di risposte precostituite, la rete diverrà un congegno di produzione e distribuzione di merce e di potere.[…] Ma l'ambiguità è l'essenziale di ogni comunicazione che non sia riducibile a mera ingiunzione, ordine che proviene dal potere e al quale bisogna obbedire se non si vuole essere emarginati ed espulsi. La pretesa di una comunicazione univoca e non ambigua può rivelare una certa ignoranza della semiologia dell'interazione, o piuttosto rivela l'intenzione di ridurre l'interazione a processo precostituito. […] Internet non è un medium che deve sacrificare ogni cosa alla creazione di opportunità economiche ma una sfera di creazione nella quale si pongono delle domande estetiche, delle ricerche di significato, cioè della comunicazione vera, e non prestampata a uso e consumo di commercianti e di utenti conformisti.*[141]

Da allora il Web è molto cambiato, e le posizioni estreme si sono attenuate. Gli stessi avvocati dell'usabilità a tutti i costi hanno rivalutato in modo considerevole l'importanza degli aspetti emozionali del design. Lo stesso Donald Norman inizia il suo libro *Emotional Design* (2004) con una citazione di William Morris, uno dei padri dell'industrial design:

> *Se si vuole una regola d'oro in grado di soddisfare chiunque, eccola: non tenere in casa propria nulla che non si ritenga utile, o non si consideri bello.*[142]

Nel prologo a questo libro, Norman spiega, infatti:

> *Negli anni '80, quando scrissi "La caffettiera del masochista", non presi in considerazione le emozioni. Mi occupai di utilità e usabilità, forma e funzione, il tutto in maniera logica, senza passione – anche se gli oggetti progettati con poca cura mi fanno infuriare. Ma oggi ho cambiato idea. Perché? In parte per via*

[141] Franco "Bifo" Berardi, *Dissociare il web design dalla usabilità*, in *Web design e usabilità, un dibattito*, e-book gratuito realizzato dal forum della trasmissione RAI Mediamente, in http://www.unitus.it/virtual/e-book/e-library.htm (aprile 2001).
[142] William Morris, *The Beauty of Life*, 1880.

> *dei nuovi sviluppi scientifici nella comprensione del cervello e del modo in cui l'emozione e il processo cognitivo siano intimamente interconnessi tra loro. Noi studiosi ora comprendiamo l'importanza dell'emozione nella vita quotidiana, quanto si dimostri preziosa. Certo, l'utilità e l'usabilità sono importanti, ma senza divertimento e piacere, gioia ed eccitazione, e, sì, ansia e rabbia, paura e ira, la nostra esistenza sarebbe incompleta.*[143]

Pur consapevoli dell'importanza della creatività e dell'emozione nell'interaction design, in questo capitolo tratteremo la grafica esclusivamente dal punto di vista della usabilità, senza occuparci degli altri aspetti indicati in Figura 223, che esulano dagli argomenti di questo libro e ci porterebbero molto lontano. Da questo punto di vista, citando le parole dello standard ISO 9241-12,[144] "la presentazione dell'informazione visiva dovrebbe abilitare l'utente a eseguire i compiti percettivi (per esempio, la ricerca d'informazioni sullo schermo) in modo efficace, efficiente e con soddisfazione". Per raggiungere quest'obiettivo, prosegue ancora lo standard, è importante che, nel progettare l'informazione visiva, si considerino le seguenti caratteristiche:

- Chiarezza (*clarity*): il contenuto informativo è veicolato velocemente e accuratamente;
- Discriminabilità (*discriminability*): l'informazione visualizzata può essere distinta con accuratezza;
- Concisione (*conciseness*): agli utenti viene fornita solo l'informazione necessaria per eseguire il compito;
- Consistenza (*consistency*): la medesima informazione è presentata nello stesso modo all'interno del sistema, conformemente alle aspettative dell'utente;
- Scopribilità (*detectability*): l'attenzione dell'utente è diretta verso l'informazione necessaria;
- Leggibilità (*legibility*): l'informazione è facile da leggere;
- Comprensibilità (*comprehensibility*): il significato è chiaramente comprensibile, non ambiguo, interpretabile e riconoscibile.

Per raggiungere questi obiettivi, nella progettazione della grafica occorre considerare numerosi aspetti. Non solo le caratteristiche del sistema visivo umano e dei processi cognitivi che filtrano ed elaborano le informazioni percepite dai nostri occhi, ma anche le consuetudini – individuali e collettive – che associano alle immagini che vediamo significati e valori diversi. Il progettista di sistemi software è spesso impreparato per questo. La sua formazione, tradizionalmente, non comprende lo studio del sistema visivo umano e della comunicazione visiva. La sua sensibilità verso le arti figurative è spesso modesta o del tutto inesistente, come suggerisce la nostra esperienza trentennale con gli studenti d'informatica. Le abilità più marcate fra i progettisti di software sono infatti quelle di carattere logico-analitico, molto lontane dalla sensibilità di un visual designer. Non stupisce, quindi, che le interfacce grafiche realizzate da progettisti di origine informatica siano spesso trascurate e molto carenti dal punto di vista estetico. Un professionista dell'interaction design, qualunque sia la sua formazione, dovrebbe tuttavia essere in grado di valutare la qualità delle interfacce grafiche, e di progettare soluzioni grafiche corrette dal punto di vista dell'usabilità ed esteticamente gradevoli, almeno nelle situazioni meno impegnative. Nelle applicazioni in cui la qualità della comunicazione visiva sia ritenuta critica, potrà poi essere affiancato da uno specialista di comunicazione visiva. Questo avviene abitualmente nella progettazione dei siti web non elementari.

In questo capitolo forniremo alcune linee guida per la progettazione delle interfacce grafiche. Esse integrano le linee guida descritte nel capitolo 10, che sono di validità generale, introducendo alcuni elementi specifici che tengono conto delle caratteristiche del sistema visivo umano.

---

[143] Donald Norman, *Emotional Design*, ed.italiana Apogeo, 2004, pag.6.

[144] ISO 9241-12:1998, *Presentation of information*. Questa parte dello standard ISO 9241 contiene raccomandazioni sulla presentazione visiva dell'informazione applicabili ad ogni tipo di dialogo realizzato per mezzo di display video. Il documento contiene raccomandazioni sull'uso di vari elementi delle interfacce testuali e grafiche: finestre, aree di input e di output, raggruppamenti di elementi, liste, tabelle, etichette, campi, cursori, e così via. La descrizione è troppo dettagliata per gli scopi di questo libro, e non ne tratteremo oltre.

Le leggi della Gestalt

La *psicologia della Gestalt* (la parola tedesca Gestalt significa *forma*, *schema*, *rappresentazione*), detta anche *psicologia della forma*, è una corrente psicologica che si sviluppò tra gli anni '10 e gli anni '30 del secolo scorso. I suoi esponenti si focalizzarono soprattutto sugli studi della percezione e del problem-solving.

L'idea portante della psicologia della Gestalt è che non è corretto dividere l'esperienza umana nelle sue componenti elementari, da analizzare separatamente, perché *un insieme è più della somma delle sue parti*. In particolare, questo avviene nella percezione visiva: gli elementi che ci si presentano nel campo visivo interagiscono fra loro in modo complesso, e noi percepiamo qualcosa che è sostanzialmente diverso dalla loro semplice somma. Per esempio, quando osserviamo una fila di lampadine che si accendono e si spengono in sequenza con una certa cadenza temporale (Figura 224), noi percepiamo delle luci in movimento, anche se nulla, in realtà, si muove.



Figura 224. Luci alternate vengono percepite in movimento

La Figura 225 illustra in modo suggestivo questo principio. Si tratta di un dipinto di Salvador Dalì, che rappresenta una stanza contenente un divano, due quadri alla parete, un caminetto che regge un orologio, vista da una grande porta incorniciata da tende. L'effetto complessivo determinato dalla somma di questi elementi è però molto diverso; l'osservatore percepisce innanzitutto un viso di donna che, com'è confermato dal titolo del quadro, assomiglia all'attrice Mae West: il tutto è diverso dalla somma delle sue parti.



Figura 225. "Viso di Mae West in forma di appartamento" (Salvador Dalì, 1935)

Gli psicologi della Gestalt hanno cercato di individuare le leggi elementari che governano questi fenomeni. Nel 1923, Max Wertheimer descrisse le *leggi dell'organizzazione figurale*, in base alle quali gli elementi presenti nel campo visivo tendono a organizzarsi in unità, cioè a venire raggruppati in modi diversi, secondo la loro forma e posizione relativa. Esse sono chiamate legge della vicinanza, della somiglianza, della chiusura, della continuità di direzione, della buona forma e dell'esperienza passata, e sono descritte qui di seguito.

- Legge della vicinanza: a parità di tutte le altre condizioni, gli elementi del campo visivo che sono fra loro più vicini tendono a essere raccolti in unità.

Nella Figura 226a, formata da 36 piccoli cerchi, potremmo in teoria vedere molti gruppi diversi, formati da varie combinazioni di cerchi. In realtà vediamo relativamente pochi raggruppamenti: linee orizzonti, o verticali, o diagonali, o ad angolo. Queste configurazioni sono però piuttosto instabili, e si tramutano continuamente l'una nell'altra. Tuttavia, è sufficiente inserire piccole modifiche nella posizione dei punti, perché la figura si stabilizzi: nelle Figura 226b, Figura 226c e Figura 226d le configurazioni ci appaiono infatti molto stabili e univoche. Nella prima vediamo inequivocabilmente tre colonne verticali, nella seconda tre righe orizzontali, e nella terza quattro gruppi di forma quadrata. Poiché l'unica grandezza che varia fra una configurazione e l'altra è la distanza, mentre forma, colore e numero dei cerchi sono rimasti invariati, è chiaro che i raggruppamenti sono determinati dalla distanza relativa fra gli elementi. Ecco perché Wertheimer afferma che, *a parità di tutte le altre condizioni*, gli elementi fra loro vicini tendono a essere percepiti come facenti parte di un'unica unità.

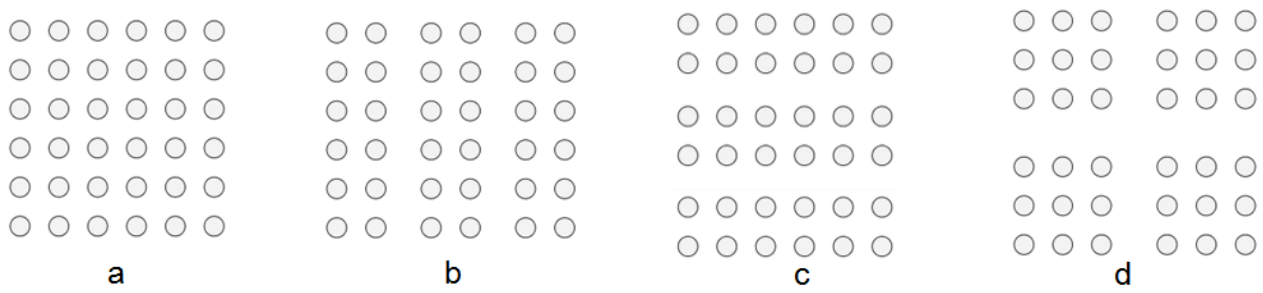


Figura 226. Legge della vicinanza

- Legge della somiglianza: a parità di tutte le altre condizioni, gli elementi del campo visivo che sono tra loro simili tendono a essere raccolti in unità.

Se nella Figura 226a, invece di spostarli, modifichiamo il colore di alcuni elementi, otteniamo una nuova segmentazione ben definita e stabile del campo visivo. Per esempio, nella Figura 227a percepiamo due gruppi ben distinti di elementi: un gruppo di tre righe orizzontali bianche, e un gruppo tre righe orizzontali nere. In Figura 227b i due gruppi di righe bianche e nere sono verticali. Se invece di modificare il colore modifichiamo la forma, il risultato è lo stesso: gli elementi si raccolgono in gruppi di analoga forma. Per esempio, nella Figura 227c il gruppo dei quadrati neri viene percepito come ben distinto dal gruppo dei cerchi neri. A parità di tutte le altre condizioni, quindi, tendono a raggrupparsi quegli elementi che hanno qualche tipo di *somiglianza*. Non solo colore o forma, come negli esempi, ma anche grandezza, orientamento o movimento verso una stessa direzione[145], come sarebbe immediato verificare con facili esempi.

Il fenomeno descritto dalla legge della somiglianza può anche essere utilizzato per porre in evidenza alcuni elementi, per *diversità* o *contrasto*. Nella Figura 227d, l'elemento grigio spicca chiaramente nel contesto degli altri elementi, tutti neri: la legge della somiglianza fa sì che esso venga isolato da tutti gli altri, in un gruppo costituito da un singolo elemento.

[145] Cioè, gli elementi che si muovono verso una stessa direzione vengono raccolti in gruppo. In questo caso, piuttosto che di legge della somiglianza si preferisce parlare di legge del moto comune.

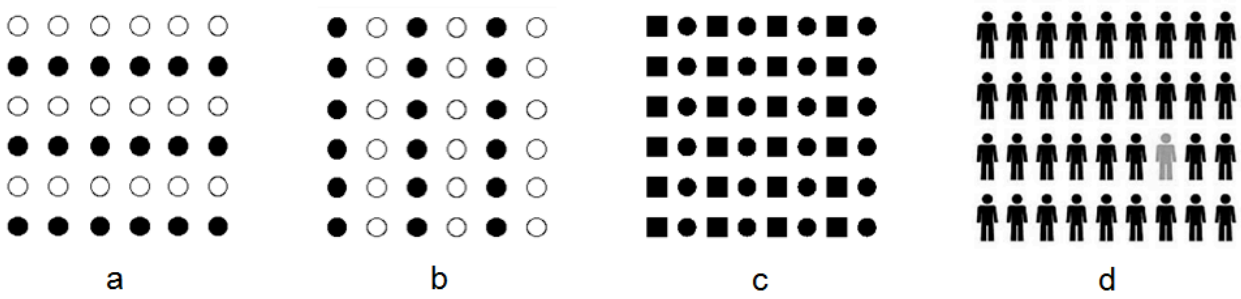


Figura 227. Legge della somiglianza

- Legge della chiusura: a parità di tutte le altre condizioni, le linee delimitanti una superficie chiusa si percepiscono come unità più facilmente di quelle che non si chiudono.

In altre parole, fra tutte le possibili organizzazioni percettive di un insieme di elementi, verrà vista preferenzialmente quella che produce figure chiuse. In Figura 228a, per la legge della vicinanza vediamo quattro coppie di linee verticali. Tuttavia, se aggiungiamo i tratti orizzontali che uniscono le linee verticali come in Figura 228b, la segmentazione del campo visivo cambia, e vediamo tre rettangoli chiusi con due linee verticali ai lati.

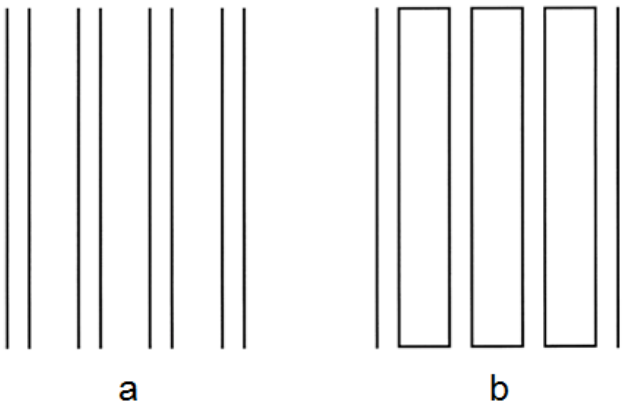


Figura 228. Legge della chiusura

Che la chiusura sia più forte della vicinanza è dimostrato anche nell'esempio di Figura 229a: anche se le "parentesi quadre" accostate (][) sono molto vicine, e quindi dovrebbero essere raccolte in gruppo, noi le associamo diversamente, e vediamo tre rettangoli chiusi. Se però eliminiamo le parentesi quadre ai due lati estremi della figura, come in Figura 229b, la situazione cambia, e la segmentazione diventa piuttosto instabile: a volte vediamo tre coppie di parentesi accostate, altre volte vediamo due rettangoli e due parentesi ai lati.

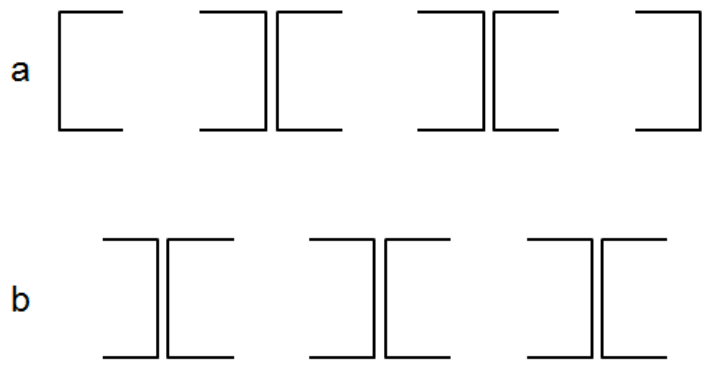


Figura 229. Conflitto fra chiusura e vicinanza

- Legge della continuità di direzione (detta anche della curva buona): a parità di tutte le altre condizioni, le linee che vanno nella stessa direzione si costituiscono in unità più facilmente delle altre.

In sostanza, il sistema visivo sembra funzionare in modo che un segmento rettilineo tenda a evitare bruschi cambiamenti di direzione e quindi, a un incrocio con altri segmenti, si unifica di preferenza con quello che continua nella medesima direzione. Per esempio, i quattro segmenti obliqui di Figura 230a si unificano in un unico segmento che sembra disposto "dietro" i tre gruppi di linee verticali, unificati per effetto, come abbiamo visto, della legge della vicinanza. Anche se questa non sarebbe, teoricamente, l'unica lettura possibile della figura, è quella più adatta a farci riconoscere gli oggetti del mondo reale, che possono venire parzialmente nascosti da altri oggetti più vicini a noi. Anche le linee curve tendono a mantenere il proprio andamento e, dopo un incrocio con altre linee, a proseguire nelle linee che meno si discostano da esso. Così, in Figura 230b, vediamo le due linee 1-2 e 3-4, e non le altre possibili (1-4 e 3-2, oppure 1-3 e 4-2).

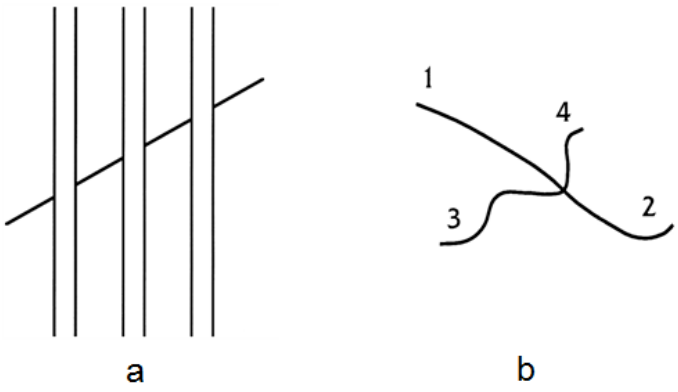


Figura 230. Legge della continuità di direzione

- Legge della buona forma: a parità di tutte le altre condizioni, il campo percettivo si segmenta in modo che risultino entità per quanto possibile equilibrate, armoniche, costituite secondo un medesimo principio in tutte le loro parti.

Questo importantissimo principio, detto anche *principio di pregnanza* o della *coerenza strutturale*, non è facilmente definibile con precisione, ma risulta chiaro dagli esempi. Nella Figura 231a, in virtù della legge della chiusura, vediamo due figure chiuse distinte, ciascuna con una propria forma definita. Se però le avviciniamo come in Figura 231b, gli elementi si raggruppano in modo completamente diverso. Le due figure, per così dire, si trasformano, e diventa praticamente impossibile vedere le due configurazioni di partenza. Questo perché, nel nuovo insieme, le linee si collegano fra loro in un modo strutturalmente più coerente: i segmenti lineari si uniscono ad altri segmenti lineari a formare un poligono, mentre le linee curve si uniscono a formare una figura tondeggiante. La tendenza alla coerenza strutturale e alla continuità di direzione ci permettono di vedere la figura in un solo modo, eliminando tutte le altre possibili segmentazioni, fra le quali anche quella di Figura 231c (una figura esterna a tratto continuo, e una interna a

tratteggio).

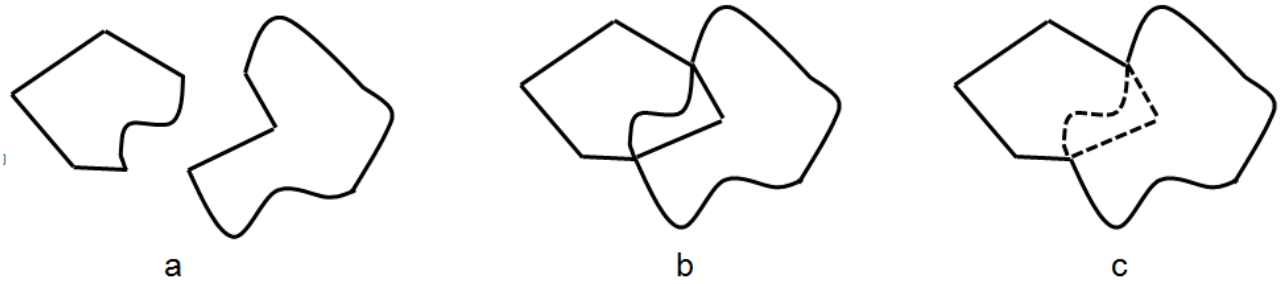


Figura 231. Legge della buona forma

La Figura 232 mostra altri esempi interessanti a illustrazione della legge della buona forma. Le configurazioni a1 e a2 potrebbero, teoricamente, essere lette come figure a due o tre dimensioni (un cubo). Ma mentre a2 appare chiaramente come un cubo, è molto difficile vedere in a1 la terza dimensione, anche se rappresenta anch'essa un cubo in visione prospettica. Questo perché la a1 è già regolare, simmetrica ed equilibrata nel piano, mentre a2 è più "regolare" se la consideriamo nello spazio.

La legge della buona forma interviene anche nei meccanismi che ci permettono di isolare le figure dallo sfondo. Consideriamo le due immagini b1 e b2, sempre in Figura 232. È più "naturale" interpretarle come cornici nere o come un quadrato bianco su un quadrato nero? In b1 prevale la prima interpretazione, mentre in b2 prevale nettamente la seconda. Infatti, nelle due situazioni, queste sono le soluzioni percettive più semplici ed equilibrate: b2 sarebbe insolita come cornice, poiché molto irregolare. Un meccanismo analogo agisce negli altri due esempi di Figura 232. In c1 vediamo prevalentemente delle bande bianche su uno sfondo nero. Infatti, sono queste le bande più regolari: fra i margini di ciascuna banda intercorre sempre la medesima distanza. Con l'interpretazione contraria, avremmo delle bande nere molto irregolari su sfondo bianco: questa interpretazione, per la legge della buona forma, viene scartata. In c2, invece, prevale l'interpretazione contraria: vediamo una banda (ancorchè molto irregolare) nera su fondo bianco. In questo caso, contano di più la minore dimensione dell'area nera rispetto a quella bianca e la sua centralità.

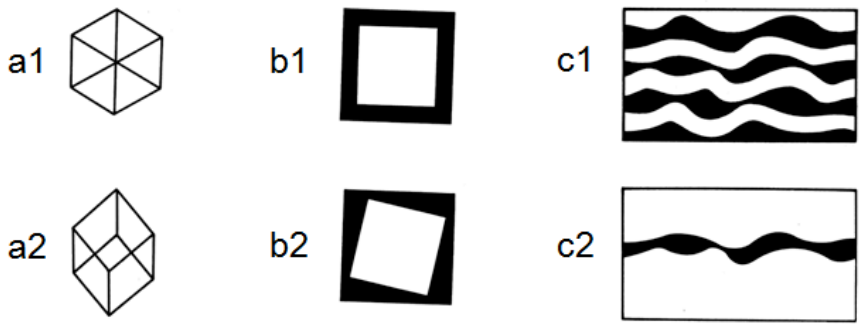


Figura 232. Legge della buona forma: altri esempi

- Legge dell'esperienza passata: a parità di tutte le altre condizioni, gli elementi del campo visivo che danno origine a una figura familiare o dotata di significato tendono a formare un'unità.

In sostanza, questo principio ci dice che l'esperienza passata orienta le nostre percezioni. Un esempio tipico è mostrato in Figura 233a, dove riconosciamo facilmente la lettera E, anche se molti tratti sono mancanti. Infatti, la familiarità con questa lettera ci permette facilmente di completare "mentalmente" la figura. Osserviamo che questo processo

d'integrazione, messo in atto dal nostro sistema cognitivo, non è sempre scontato. La Figura 233b contiene un'altra configurazione di tratti appartenente alla lettera E, nella quale tuttavia il riconoscimento è più problematico, anche se il numero di tratti è identico a quello dell'esempio precedente.

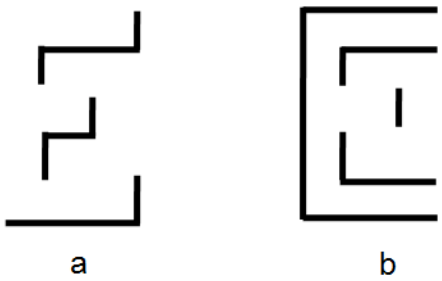


Figura 233. Legge dell'esperienza passata

Gli esempi a illustrazione delle leggi della Gestalt potrebbero essere molto più numerosi.[146] Essi ci dimostrano che, di fronte a una molteplicità di elementi presenti nel nostro campo visivo, il nostro sistema visivo "sceglie" una ben precisa interpretazione, in virtù di propri meccanismi di funzionamento. Questi non dovrebbero sorprenderci. Essi, in ultima analisi, si sono evoluti per permetterci di riconoscere nel modo migliore gli oggetti del mondo fisico che ci circonda: un oggetto tende a essere costituito da parti adiacenti (legge della vicinanza) e simili (legge della somiglianza); i suoi contorni tendono a variare gradualmente senza presentare brusche discontinuità (legge della continuità di direzione) e sono chiusi (legge della chiusura). Moltissimi oggetti hanno forme regolari e simmetriche (legge della buona forma), e dalle esperienze passate impariamo a riconoscere gli oggetti già visti (legge dell'esperienza). Un sistema percettivo che privilegia queste leggi fornirà quindi, nella maggior parte dei casi, informazioni che descrivono "correttamente" il mondo reale.[147]

Se invece gli elementi presenti nel campo visivo sono irregolari nella forma e nella distribuzione spaziale, senza simmetrie o continuità, abbiamo serie difficoltà a riconoscerne il "senso", come negli esempi di Figura 234a e b. Nella prima, l'assenza di linee chiuse e la somiglianza delle numerose chiazze nere presenti sullo sfondo bianco ci impediscono di vedere ciò che la figura rappresenta: un cane di razza dalmata visto da dietro, che annusa il terreno. In questa immagine, la legge della somiglianza gioca a nostro sfavore, associando fra loro le macchie nere del cane e del terreno. D'altra parte, l'assenza di linee chiuse ci impedisce di separare la figura del cane dallo sfondo. Nella seconda, per gli stessi motivi, risulta difficile riconoscere il viso barbuto che la figura rappresenta.[148]

[146] Il lettore interessato può riferirsi al libro *Grammatica del vedere*, di G.Kanitza, Ed.Il Mulino, 1980, dal quale abbiamo tratto molti degli esempi sopra discussi.

[147] Ciò avviene nella maggior parte dei casi, ma non sempre, come testimoniano le cosiddette *illusioni ottiche*, causate da configurazioni visive che "ingannano" i meccanismi della visione, e producono interpretazioni sbagliate. Lo studio delle diverse illusioni ottiche e delle loro cause è molto interessante, ma lo spazio a disposizione non ci consente di parlarne. D'altra parte, questi fenomeni si presentano molto di rado – o non si presentano affatto - nelle tipiche interfacce grafiche dei sistemi interattivi.

[148] Si tratta di due esempi molto citati nei testi sulla visione. Il primo è tratto dal classico testo di P.Lindsay e D.Norman, *Human Information Processing* (Academic Press, 1980); il secondo da P.B.Porter, *Another puzzle-picture*, in *American Journal of Psychology*, n.67, 1954, pp.550-551.

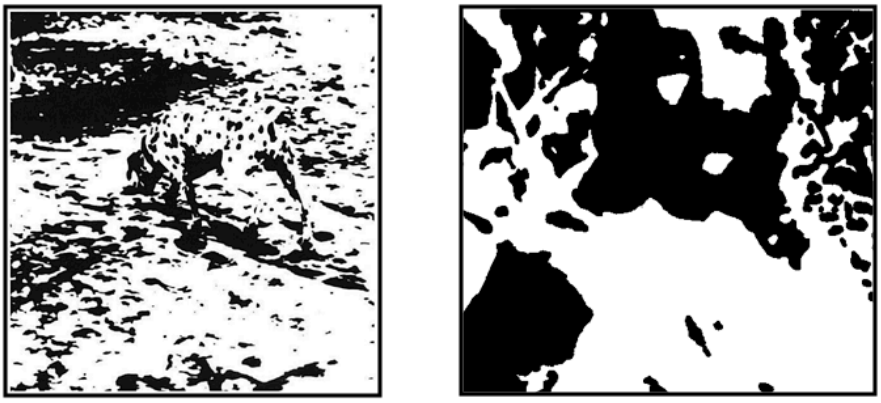


Figura 234. Figure prive di Gestalt

La conoscenza delle leggi della Gestalt è di grande utilità per il progettista di interfacce grafiche. Egli potrà sfruttare questi meccanismi a suo favore, per far sì che il sistema visivo dell'utente gli mostri le immagini presentate sullo schermo nel modo desiderato. Nel seguito di questo capitolo ne vedremo numerosi esempi.

Vicinanza

Il principio della vicinanza della Gestalt ci dice che elementi vicini verranno percepiti come appartenenti a uno stesso gruppo. Questo ci suggerisce di porre uno vicino all'altro gli elementi grafici che, dal punto di vista funzionale, sono fra loro correlati. E anche, ovviamente, di tenere lontani elementi che non hanno fra loro alcun rapporto. In questo modo il progettista sfrutta la legge della vicinanza a proprio vantaggio, facendo sì che i meccanismi della visione rafforzino la percezione dello stretto legame che unisce gli oggetti dell'interfaccia fra loro semanticamente o funzionalmente correlati.

La Figura 235 mostra la form di un'applicazione alberghiera. Contiene numerosi campi, per l'immissione dei dati degli ospiti dell'albergo. Esiste una certa correlazione fra campi vicini: le informazioni relative alla camera occupata sono raggruppate in basso, i dati per il pagamento del conto sono al centro, e così via. Ma non c'è alcun ordine visivo: i campi appaiono disposti alla rinfusa, e queste correlazioni sono quasi impossibili da individuare. Ogni volta che utilizziamo la form, la dobbiamo esplorare visivamente, e riscoprirne la logica nascosta. I meccanismi della visione non ci aiutano a coglierne il senso. L'immagine è così destrutturata ("priva di Gestalt") che nemmeno la legge dell'esperienza passata ci permetta di venirne a capo rapidamente: ogni volta che la esaminiamo è come se fosse la prima volta.

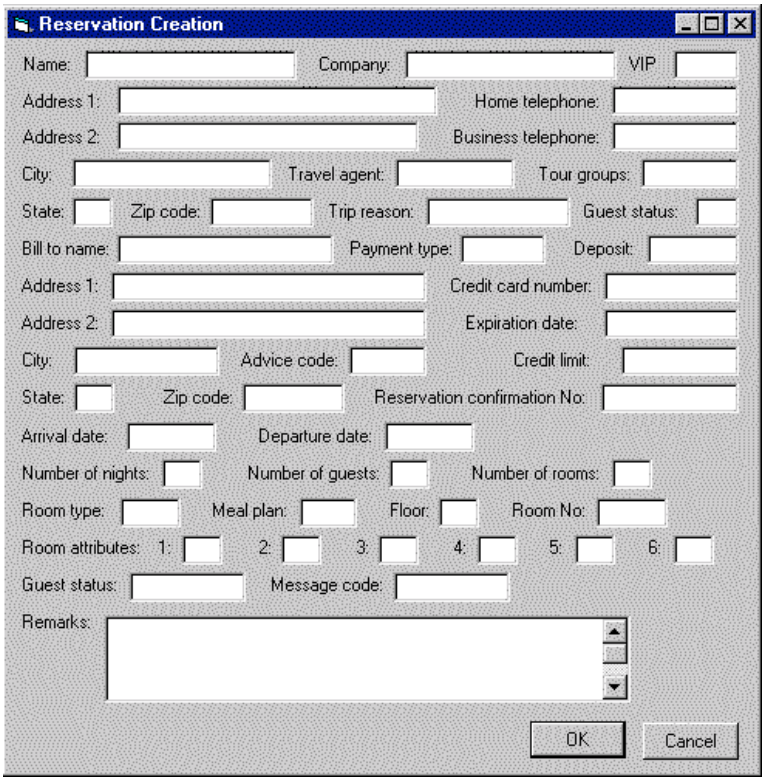


Figura 235. Form di un'applicazione alberghiera

La Figura 236 mostra gli stessi campi, riorganizzati visivamente su due form.[149] Ora, i campi sono stati chiaramente suddivisi in quattro gruppi di significato omogeneo: il gruppo con le informazioni riguardanti la prenotazione della camera, quello con i dati anagrafici dell'ospite e, nella seconda form, un gruppo con i dati per il pagamento e uno con i dati per la fatturazione. La vicinanza dei campi semanticamente correlati facilita enormemente la comprensione rispetto alla versione precedente. La correlazione fra i campi vicini è ulteriormente sottolineata dalle linee che incorniciano i diversi gruppi, per sfruttare la legge della chiusura. L'allineamento verticale delle etichette e dei campi contribuisce ulteriormente alla forte sensazione di ordine e di semplicità strutturale trasmessa dalla grafica.

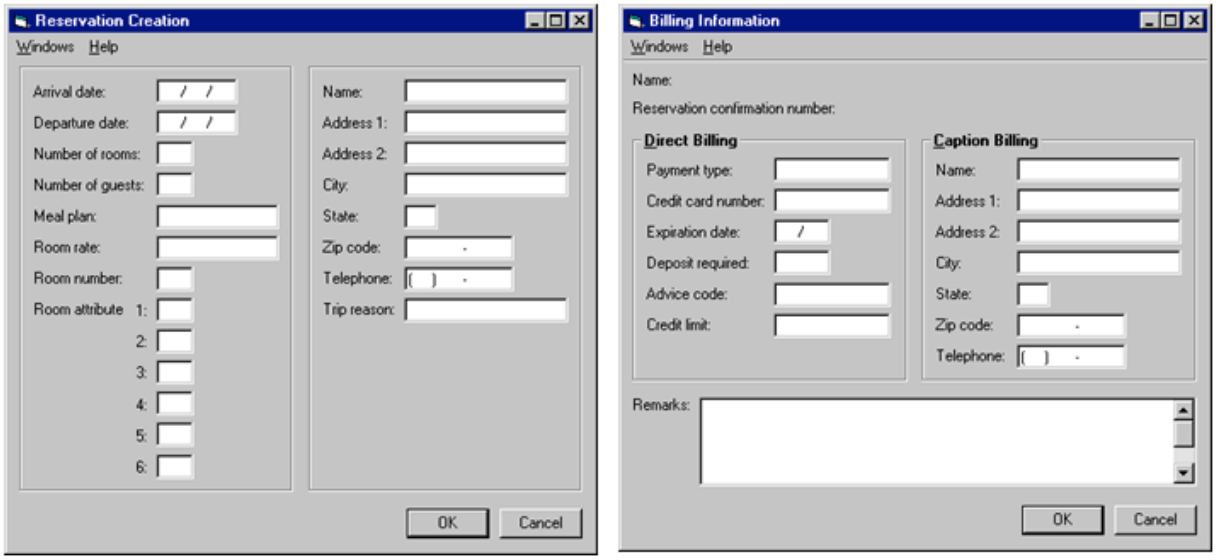


Figura 236. L'applicazione alberghiera di Figura 235 ridisegnata

[149] Nella riorganizzazione, qualche campo è stato eliminato, evidentemente perché ritenuto superfluo.

I meccanismi della visione possono facilitare la comprensione di un'immagine, ma possono anche creare serie difficoltà, se non li sfruttiamo per i nostri scopi. Il pannello di Figura 237, che permette di definire le tabulazioni dei testi in PowerPoint 2007, risulta incomprensibile. Il titolo Allineamento, che si riferisce ai quattro check-box sottostanti, è invece visivamente contiguo al grande campo bianco sulla sinistra, e gli viene quindi associato. Quest'ultimo dovrebbe essere associato al titolo Tabulazioni da cancellare il quale però, presumibilmente per un errore di software, è spostato sulla destra, e si trova quindi contiguo al titolo Tabulazioni predefinite, con cui fa gruppo (anche per la somiglianza delle parole). Anche in questo caso, ogni volta che esaminiamo il pannello, dobbiamo "lottare" con il nostro sistema visivo per interpretarlo correttamente.

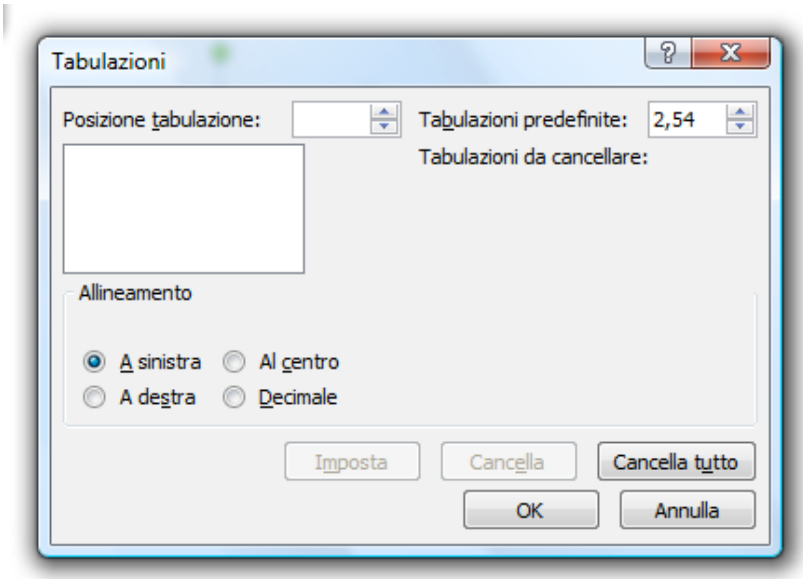


Figura 237. Da Microsoft PowerPoint 2007

Vediamo ora un esempio più complesso, legato a una situazione che si presenta frequentemente nella pratica, in molte diverse varianti. Durante la progettazione di un sistema informativo aziendale, all'inizio degli anni '90, si trattava di visualizzare su monitor la tabella di informazioni anagrafiche rappresentata in Figura 238 e costituita da cinque colonne, di cui qui non interessa il significato. Le righe della tabella erano però troppo lunghe per i monitor utilizzati, che permettevano di visualizzare righe di 80 caratteri.

Anagrafica	Conto FM	Ragione Sociale	Stato	Data St.
- __.E._____.	M2 _____.	_____	____	_____
- __.E._____.	M2 _____.	_____	____	_____
- __.E._____.	M2 _____.	_____	____	_____
- __.E._____.	M2 _____.	_____	____	_____
- __.E._____.	M2 _____.	_____	____	_____

Figura 238. Una tabella con righe lunghe, da visualizzare su monitor di 80 caratteri

La soluzione, adottata da un analista di procedure ignaro dei meccanismi della Gestalt, è assolutamente disastrosa (Figura 239). Ogni riga della tabella è stata, per così dire, ripiegata in due, e visualizzata su due righe contigue dello schermo. In questo modo, i campi Stato e Data St. si trovano disposti sotto i campi Anagrafica e Conto FM, mentre il campo Ragione Sociale, troppo lungo, è stato diviso in due *tranche*, disposte una sotto l'altra. Per permettere l'identificazione dei vari campi, è stato quindi necessario ripetere le etichette su ogni riga della tabella, non essendo più possibile porle una sola volta in testa a ogni colonna, come in Figura 238.

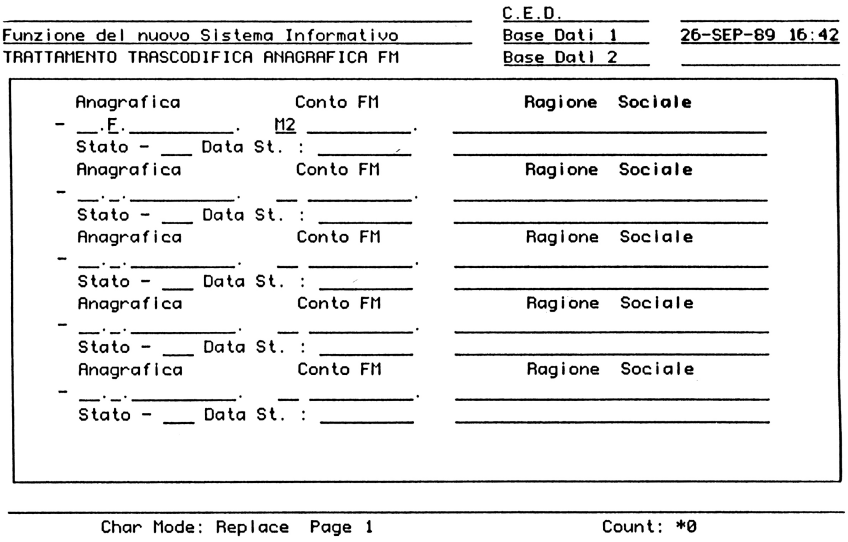


Figura 239. Rappresentazione a video della tabella di Figura 238

Tutto ciò, unito a qualche disattenzione nella progettazione del layout, rende la figura del tutto incomprensibile. Il motivo è molto semplice: le leggi della Gestalt *ci impediscono* di vederla nel modo corretto. Infatti, lo spazio sotto le righe che iniziano con Anagrafica è maggiore di quello sotto le altre righe. Pertanto, la legge della vicinanza ci costringe ad associare le righe a tre a tre, e non a due a due, e per di più in modo sfasato rispetto alla situazione corretta: i gruppi che il nostro sistema visivo ci presenta sono infatti quelli di Figura 240, che risultano del tutto privi di senso. Anche la legge della somiglianza crea seri impedimenti alla comprensione. Infatti raggruppa fra loro le etichette simili, creando un gruppo di Ragione Sociale, un gruppo di Conto FM, due gruppo di coppie di linee orizzontali contigue, ecc. Questo, in sostanza, ci fa leggere verticalmente la tabella, impedendoci, di fatto, di leggerla nel modo corretto. Tutte le volte che il sistema ci proporrà la tabella sul monitor, sperimenteremo le stesse difficoltà di comprensione. Tanto forte è l'effetto combinato dei meccanismi che abbiamo descritto, che nemmeno la legge dell'esperienza passata ci può aiutare.

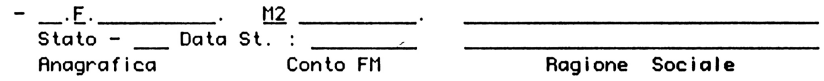


Figura 240. Associazione delle righe di Figura 239, per la legge della vicinanza

Somiglianza

Possiamo utilizzare a nostro vantaggio la legge della somiglianza dando forma o colore simili a quegli elementi grafici che sono funzionalmente o semanticamente correlati.

Nel menu di Figura 241a, tratta dalla home page di www.yahoo.it, ad ogni singola voce è associata una piccola icona colorata. In questo caso, le icone non servono a spiegare il significato della voce: le etichette sono chiare, e non c'è bisogno di spiegazioni aggiuntive; d'altra parte le figure sono piccole e non sempre riconoscibili. La loro funzione principale è un'altra: quella di associare a ogni voce di menu un pattern visivo ben riconoscibile, che non avrebbe con l'uso del solo testo. In questo modo, per la legge della somiglianza, tutte le voci vengono raccolte in un gruppo, ben differenziato dagli altri gruppi presenti sulla pagina, che è molto densa di informazioni. Analogo fine hanno le icone di Figura 241b, tratta dalla home page del sito della British Airways di qualche anno fa: esse sono utili nonostante la scarsa comprensibilità delle immagini (perché mai "Si registri ora" dovrebbe essere associato alla figura di un ombrellone sulla spiaggia?). Nel menu di Figura 241c, tratto dal pannello di controllo di Windows Vista, strutturalmente simile ai precedenti, le icone hanno dimensioni maggiori, e contengono figure ben riconoscibili. In questo caso, a differenza degli altri due esempi, costituiscono una vera alternativa al testo: in molti casi l'utente sarà in

grado di selezionare la funzione desiderata senza leggerne la descrizione testuale, semplicemente eseguendo una *visual search* sulla colonna delle icone.

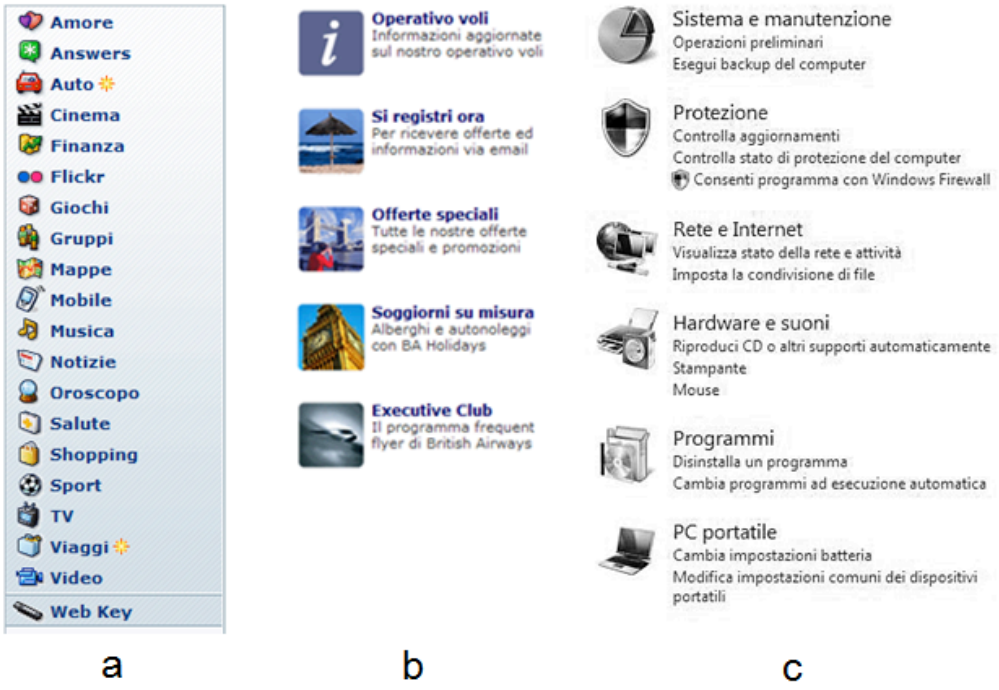


Figura 241. Menu da http://www.yahoo.it (a, 2009), http://www.british-airways.com (b, 2003) e Windows Vista (c, 2009)

Nell'esempio di Figura 242, tratto dalla home page del sito della Esselunga, le otto voci del menu principale posseggono una forma ben riconoscibile, data dalla associazione del testo e di una icona quadrata ben visibile. Stranamente, però, il visual designer ha distanziato le due righe del menu, inserendo fra l'una e l'altra quattro grandi banner, anch'essi di forma quadrata, ma molto più grandi. La legge della somiglianza e quella della vicinanza, in questo caso, operano in conflitto fra loro: la prima tende a unificare le otto voci di menu in un singolo gruppo, la seconda tende a separarle in due gruppi distinti, fra i quali si inserisce il gruppo dei banner. Questa scelta, che frammenta il menu principale e lo pone in secondo piano, non ci sembra convincente.



Figura 242. Da www.esselunga.it (2009)

Due pessimi esempi, tratti dalla raccolta *Interface Hall of Shame* sul Web, sono riportati in Figura 243 e in Figura 244. Nel primo, il progettista non ha ritenuto necessario introdurre alcun accorgimento visivo che permettesse di ripartire le informazioni in gruppi, semplificando l'immagine e dandole un significato. Ne risulta una form in cui nomi e valori dei campi si confondono: l'occhio la percorre invano alla ricerca di qualche elemento che permetta di attribuirle un senso. La figura è del tutto priva di Gestalt: la si confronti, pur con le ovvie differenze, con l'immagine del cane dalmata che si confonde con lo sfondo, già visto nella Figura 234a.

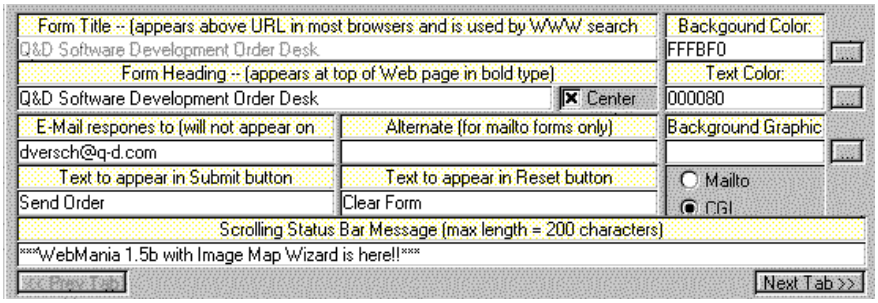


Figura 243. Da Webforms

Nell'esempio di Figura 244, invece, la legge della somiglianza ripartisce gli elementi visivi in tre gruppi ben definiti: bottoni, etichette e campi. Ma, interagendo con il sistema, ci si accorge che non tutti gli elementi che sembravano bottoni lo sono realmente. Subscriber e Contact, infatti, sono solo etichette, e quindi non cliccabili, anche se hanno la stessa forma di Save e Cancel, che sono realmente bottoni.

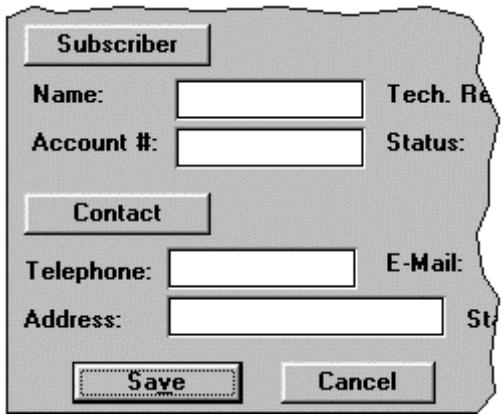


Figura 244. Titoli o bottoni?

La Figura 245 mostra un altro esempio di pessimo design. Qui, il progettista ha cercato di ridurre le difficoltà dovute all'eccessivo numero di tab associando a ciascuno di essi un'icona colorata, probabilmente allo scopo di rendere più riconoscibili le singole voci. Ma il risultato è controproducente. Le icone, che sono vivacemente colorate con una diecina di colori diversi, contrastano con il colore neutro delle scritte e, per la legge della somiglianza, non si raggruppano con le etichette contigue, ma fanno gruppo a sé. Ne risulta un guazzabuglio visivo che rende molto difficile l'identificazione della funzione desiderata.

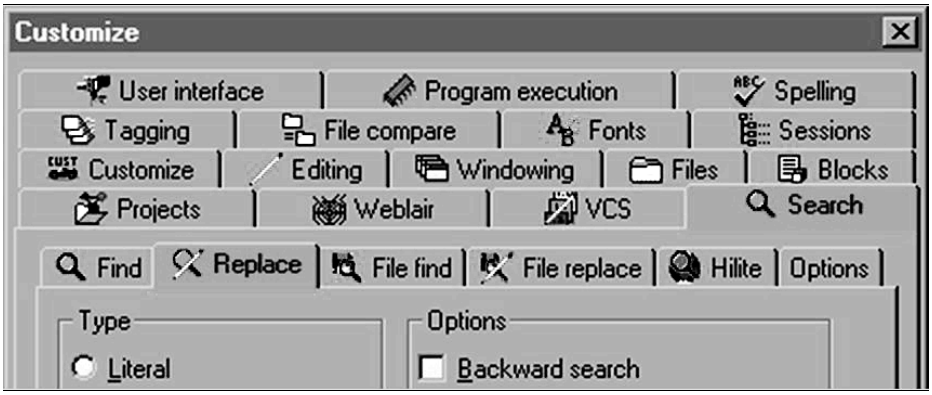


Figura 245. Da MultiEdit 8.0

Chiusura

Una tecnica molto efficace per associare visivamente più elementi consiste nel racchiuderli all'interno di una cornice chiusa.

La Figura 246 mostra tre versioni di uno stesso menu a tendina. Nella prima (a), le varie voci sono elencate in un ordine logico (New, Open e Close sono voci contigue, e così Save e Save as...), ma nessun accorgimento grafico mette in evidenza questi raggruppamenti. Nella seconda (b), è stata inserita una spaziatura per distinguere i tre gruppi, sfruttando la legge della vicinanza. Nella terza (c) è stata inserita una linea di separazione fra un gruppo e l'altro. Ogni gruppo risulta ora all'interno di un rettangolo, che lo isola dagli altri, sottolineando con evidenza molto maggiore le relazioni fra le voci di ciascun gruppo.

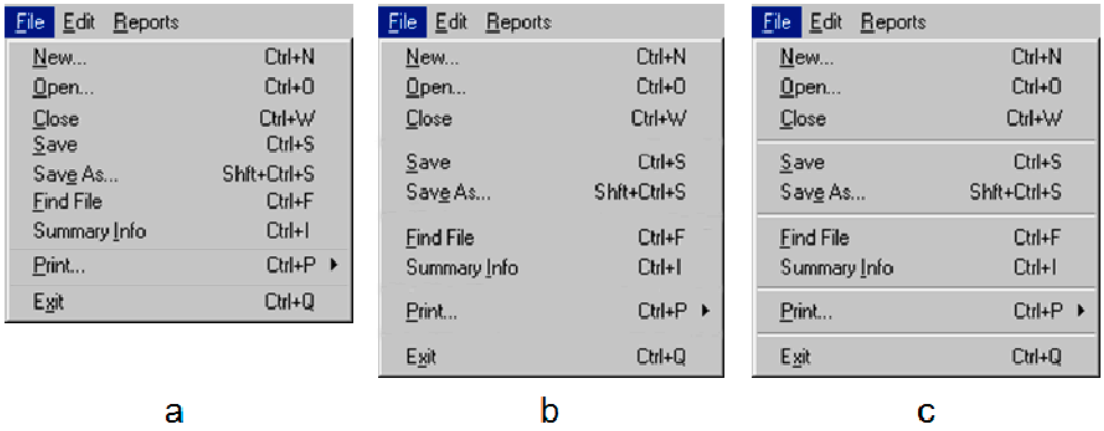


Figura 246. L'effetto dei separatori in un menu a tendina

Quando si abbia la necessità di mostrare sul monitor una grande quantità di elementi, come avviene spesso sul Web, l'uso delle cornici è spesso la tecnica più conveniente per orientare l'utente nella lettura corretta della pagina. La Figura 247 mostra la home page del sito di due compagnie aeree, British Airways e Alitalia. In entrambe, la form di prenotazione voli (sulla sinistra in entrambe) è posta in grande evidenza dal riquadro che la isola, anche cromaticamente, dagli altri elementi.



Figura 247. http://www.britishairways.com e http://www.alitalia.it (2009)

Queste due pagine web, pur nella loro semplicità – la grafica è semplice e funzionale, senza soluzioni a effetto - risultano molto chiare e leggibili: le informazioni sono ben organizzate in gruppi logici facilmente individuabili. Tutti i meccanismi della visione descritti dalle leggi della Gestalt sono stati sfruttati per facilitare l'orientamento dell'utente all'interno dei diversi gruppi di informazioni. Il risultato non deriva soltanto dalla presenza delle cornici, ma dall'uso sapiente di diversi accorgimenti: la vicinanza di informazioni correlate e il contrasto (di forma o colore) fra informazioni che non lo sono, la forte evidenza dei pulsanti principali ottenuta con colori vivaci e saturi, che richiamano i colori del logo delle due compagnie.

Ben diverso è il risultato ottenuto dai progettisti del sito di Figura 248. Le informazioni sono inserite in numerosi riquadri, che affollano la pagina. La sensazione complessiva che essa ci trasmette è di confusione. Ciò che spicca innanzitutto, per la legge della somiglianza, è il gruppo delle immagini, che rappresentano prevalentemente delle automobili. Queste però non sono disposte in modo da permetterci di distinguere, "a colpo d'occhio", le diverse aree funzionali della pagina, come negli esempi di Figura 247: infatti, "escono" dalle cornici per fare gruppo a sé. Nemmeno i titoli dei riquadri ci aiutano a questo scopo: nella maggior parte dei casi, essi sono di corpo inferiore a quello dei testi, e quindi restano, per così dire, in secondo piano. Né ci aiutano i quadratini accanto ai titoli: sono troppo piccoli e di un colore che non li mette in risalto (giallo). Ancora una volta, la legge della somiglianza opera non a favore, ma *contro* la comprensibilità del sito. La pagina ci costringe a una lettura sequenziale: per individuarne i contenuti, dobbiamo scandirla per intero, leggendo titoli e testi, e identificandone a posteriori i temi principali.



Figura 248. http://www.quattroruote.it (2009)

Allineamento

L'allineamento degli elementi visivi è una delle tecniche più importanti usate dai grafici per dare all'immagine una struttura immediatamente percepibile. Una pagina i cui elementi siano disallineati ci trasmette un'impressione di complessità che può essere molto ridotta con un maggior ordine visivo. Per ottenerlo, gli elementi dovrebbero essere inseriti in una griglia logica ben definita. Confrontiamo ancora una volta la form di Figura 235 con la sua riorganizzazione in Figura 236. L'immediata comprensibilità di quest'ultima non è solo il risultato della vicinanza dei campi semanticamente correlati e delle cornici che racchiudono i diversi gruppi di informazioni. Anche l'allineamento delle etichette e dei campi all'interno dei riquadri e, nella form di destra, dei riquadri stessi, contribuisce a trasmettere una sensazione di ordine e di semplicità. In ultima analisi, l'allineamento rende gli elementi più "simili", e quindi anche la legge della somiglianza è all'opera per facilitare ulteriormente l'identificazione dei diversi gruppi.

Anche in situazioni molto più semplici di quella di Figura 235 il mancato allineamento degli elementi visivi può creare problemi. Nell'esempio di Figura 249, il progettista aveva inteso porre in evidenza i raggruppamenti logici dei vari campi introducendo i due riquadri e ponendoli al centro della finestra. Ma l'effetto è controproducente: l'immagine è confusa, e non se ne percepisce la logica: gli elementi sembrano disposti in modo casuale. Per giunta, il gruppo di elementi che maggiormente spicca è quello dei campi di input, unificati dalla somiglianza di colore (bianco) e di forma (rettangolare). Anche in questo caso, l'organizzazione visiva degli elementi impedisce di leggerla in modo corretto.

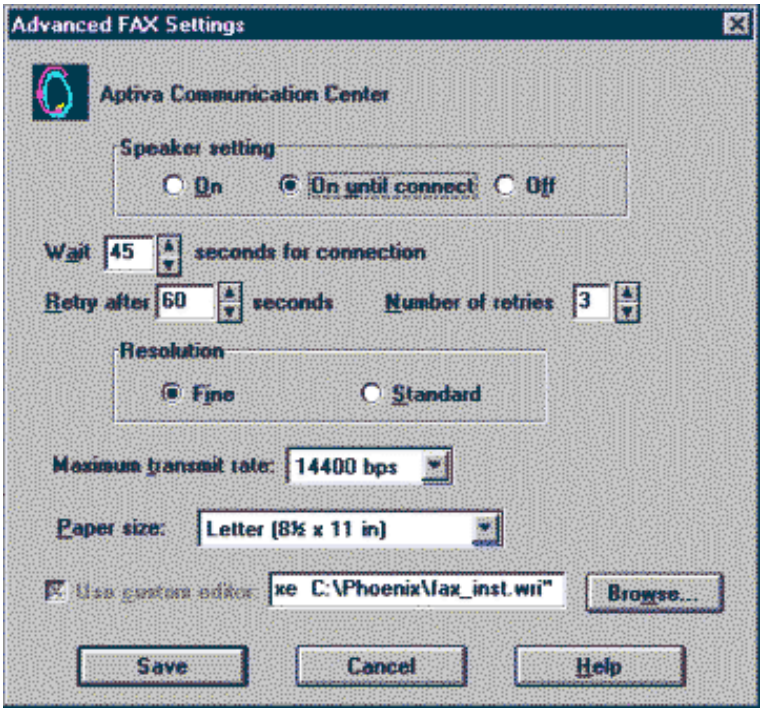


Figura 249. Da Aptiva Communication (IBM)

Colore

Il colore, se usato bene, può essere di grande aiuto alla comprensione di un'interfaccia. Quando è usato male può creare serie difficoltà. I colori andrebbero impiegati non tanto (o non solo) per rendere gradevoli le schermate, quanto soprattutto per distinguere meglio fra loro contenuti di natura diversa. Ancora una volta, possiamo sfruttare a nostro vantaggio la legge della somiglianza della teoria della Gestalt, utilizzando gli stessi colori per associare visivamente elementi fra loro correlati, o colori diversi e contrastanti per dissociare elementi semanticamente o funzionalmente lontani.

Anche se la stampa in bianco e nero di questo libro trasforma i colori in toni di grigio, l'esempio di Figura 250 dimostra chiaramente l'efficacia di questa tecnica, se usata in modo sapiente. Si tratta della mappa in tempo reale dei mercati finanziari presentata dal sito http://www.smartmoney.com. I riquadri rappresentano le principali aziende quotate in borsa, raggruppate per settori merceologici (Health care, Financial, Energy, Utilities, ecc.). L'area di ogni riquadro è proporzionale alla capitalizzazione dell'azienda, e il colore del riquadro rappresenta la variazione del valore del titolo a partire da una data scelta dall'utente. In rosso sono indicate le variazioni negative, in verde quelle positive. La luminosità del colore è proporzionale all'entità della variazione: rosso vivo = perdita consistente, rosso scuro = perdita lieve, e così via, come spiegato nella legenda in alto a destra. L'immagine ci mostra "a colpo d'occhio" la situazione complessiva del mercato, permettendoci di raggruppare visivamente le aziende in funzione dell'andamento del loro titolo (legge della somiglianza). D'altro canto, la vicinanza dei riquadri corrispondenti ad aziende dello stesso settore (legge della vicinanza) e la cornice bianca che delimita ogni settore (legge della chiusura) ci permettono di avere immediatamente una percezione qualitativa dell'andamento dell'intero settore. Infine, la disposizione spaziale regolare, realizzata ricercando i migliori allineamenti fra i riquadri e collocandoli all'interno di aree anch'esse rettangolari e per

quando è possibile allineate, contribuisce a dare un'impressione di semplicità, nonostante la grande quantità di informazioni visualizzate.

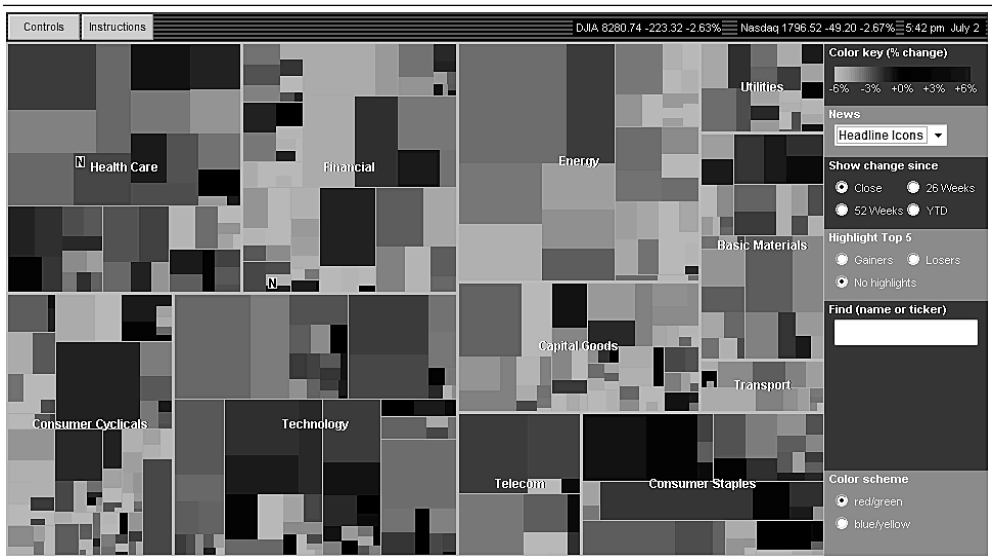


Figura 250. http://www.smartmoney.com (2009)

Un altro eccellente uso del colore è mostrato in Figura 251, che rappresenta una pagina del sito della British Airways di qualche anno fa. Per facilitare all'utente la scelta di un volo fra numerose possibilità in giorni vicini, le alternative sono visualizzate in colori diversi secondo la fascia tariffaria utilizzata.



Figura 251. Da http://www.britishairways.com

I colori possono anche essere usati per indirizzare l'attenzione dell'utente su elementi di particolare importanza. Per

questo sono particolarmente indicati i colori saturi e caldi (per esempio rosso o arancione vivo), meglio se su sfondi insaturi e freddi (per esempio verde chiaro o azzurro), come nella Figura 188 a pagina 218, dove i due controlli dello slider sono di colore rosso vivo su sfondo grigio, mentre il pulsante Esegui ricerca, di colore blu, è meno in evidenza. In Figura 252, il menu principale è messo in grande evidenza dal colore rosso vivo, anche nel contesto di pagine molto ricche di banner e di elementi colorati. La voce corrente è segnalata da una freccia nera. Essa risulta ben visibile, anche se molto piccola, per effetto del contrasto con il colore rosso vivo del menu. I menu di secondo livello (nel nostro esempio, quello relativo alla voce Servizi e Assistenza) sono invece grigio pallido: questo li separa bene, per contrasto, dal menu principale.

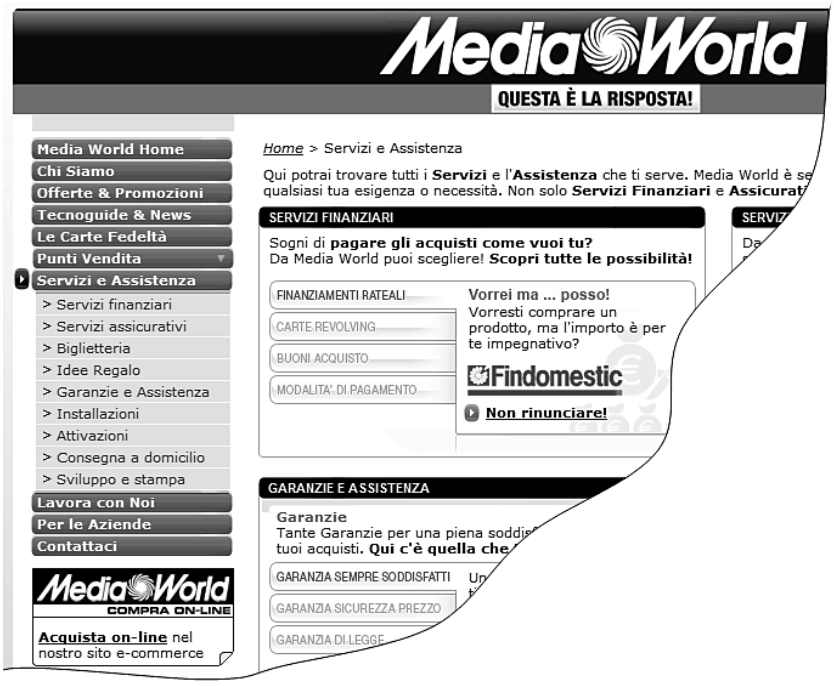


Figura 252. Da http://www.mediaworld.it

Il colore può essere usato per potenziare l'effetto della legge della chiusura, colorando i riquadri che identificano le diverse aree funzionalmente differenti di una schermata. A questo scopo si useranno prevalentemente sfondi di colori insaturi, per non risultare stancanti, e rendere i testi ben leggibili. La tecnica è particolarmente utile per i siti web con pagine molto ricche di informazioni, che è necessario differenziare visivamente.

L'uso del colore, per essere d'efficace aiuto alla comprensione, non deve mai eccedere. È consigliabile usare pochi colori contemporaneamente, evitando pagine eccessivamente variopinte. Infatti, troppa informazione equivale a nessuna informazione, e pagine troppo ricche di colori diversi tendono a sembrare complesse, anche quando in realtà non lo sono.

Un altro aspetto che richiede una certa attenzione è l'uso di colori che, presso determinati gruppi di utenti, sono associati a particolari significati. Per esempio, nei paesi occidentali il colore rosso è usato per segnalare pericolo, mentre il verde è normalmente associato a messaggi di via libera. Un cartello di STOP di colore verde a un incrocio stradale sarebbe ambiguo, e probabilmente causerebbe parecchi incidenti. Così, scelte differenti del colore dei due pulsanti in Figura 253a possono portare a conseguenze molto diverse dal punto di vista dell'usabilità del sistema. L'associazione STOP→verde e GO→rosso genererebbe probabilmente più errori da parte dell'utente dell'associazione inversa, molto più consueta. L'osservazione sembra ovvia, ma gli esempi di queste associazioni "sbagliate" abbondano. La Figura 253b mostra un esempio interessante: le parole YES e NO sono, rispettivamente, di colore verde e rosso. Il progettista ha associato i colori al significato delle etichette (NO→negazione→rosso) invece che all'effetto che si ottiene premendo i pulsanti relativi, come sarebbe stato più corretto (NO→non cancellare i record→verde). La distinzione è sottile, ma la scelta fatta dal progettista è certamente la più pericolosa.



Figura 253. Quali colori associare ai tasti?

La Figura 254 mostra un altro esempio di scelta criticabile dei colori. Nelle pagine del catalogo dei prodotti in vendita presso www.mediaworld.it, i codici colore scelti per segnalarne la disponibilità non seguono le normali convenzioni. Infatti, il verde e il rosso non contraddistinguono i prodotti disponibili ed esauriti, come ci si aspetterebbe, ma indicano, rispettivamente, le novità e i prodotti in promozione. Il nero indica un prodotto esaurito. Il giallo, invece, identifica un prodotto disponibile e non un prodotto in via d'esaurimento, come ci potremmo aspettare. Le associazioni sono per noi così innaturali da richiedere uno sforzo cognitivo abbastanza rilevante.



Figura 254. Da http://www.mediaworld.it

Tutto ciò, ovviamente, secondo le consuetudini dei paesi occidentali. In altri paesi, le consuetudini potrebbero essere molto differenti.

Un altro problema che dobbiamo tenere presente nella scelta dei colori è quello della cecità cromatica (daltonismo): come abbiamo già visto nella sezione dedicata alla visione del capitolo 4 (pag.99), non tutti gli utenti sono in grado di distinguere correttamente i vari colori, a causa di difetti nella pigmentazione dei coni della retina dell'occhio. Il problema è importante, poiché la percentuale di persone affette da disturbi nel riconoscimento dei colori è elevata (circa 1 maschio ogni 12 e 1 femmina ogni 165). I disturbi più frequenti (presenti nel 5% circa delle persone) riguardano l'incapacità di distinguere il rosso dal verde, ma ci sono altri tipi di disturbi, anche se meno diffusi. E' quindi opportuno che nell'interfaccia non ci siano informazioni importanti identificabili esclusivamente attraverso il colore. In particolare, non dobbiamo mai supporre che l'utente sia in grado di distinguere il rosso dal verde. Da questo punto di vista, l'esempio di Figura 254 è doppiamente scorretto: i codici colore, oltre che insoliti, come abbiamo già osservato, non sono distinguibili da una parte significativa degli utenti. Il sito di Figura 250, invece, correttamente consente all'utente di scegliere uno schema cromatico alternativo (con la check-box in basso a destra), che visualizza in giallo e in blu, rispettivamente, le variazioni negative e positive. Questi due colori, infatti, sono distinguibili dalla grandissima maggioranza delle persone.

Percorsi visivi

È abbastanza diffusa la convinzione che, quando esaminano una schermata, i nostri occhi seguano un percorso regolare, iniziando dalla posizione di home (l'angolo in alto a sinistra) e procedendo da sinistra a destra e dall'alto in basso, come quando si legge un testo scritto. Questa convinzione non ha alcun fondamento. In realtà, la situazione è molto più complessa, e può essere analizzata utilizzando i dispositivi di eye tracking, che sono in grado di tracciare il percorso

effettuato dal nostro sguardo (chiamato *scanpath*). Questi dispositivi mostrano che il movimento dei nostri occhi è molto irregolare: lo sguardo si fissa per un certo tempo su un determinato punto, per acquisire l'informazione visiva (*fissazione*), e quindi sposta l'asse visivo su un altro punto, con un movimento rapidissimo (chiamato *saccade*) durante il quale non viene acquisita alcuna informazione visiva. In media vengono eseguite tre-quattro fissazioni al secondo.

Gli studi effettuati con queste apparecchiature mostrano che il percorso dello sguardo, anche sulla stessa immagine, è molto variabile, e dipende non solo dalle caratteristiche dell'immagine stessa, ma anche – e soprattutto – dagli obiettivi di chi guarda. In un classico esperimento del 1967, lo psicologo russo Alfred Yarbus studiò i percorsi dello sguardo sul dipinto *Un visitatore inaspettato*, di I.E.Repin (Figura 255). Pur con apparati di eye tracking molto più rudimentali di quelli disponibili oggi, l'esperimento mostrò chiaramente che l'osservatore esaminava il quadro con percorsi visivi completamente diversi secondo la richiesta fattagli dal conduttore dell'esperimento. Gli scanpath mostrati nella Figura 255 corrispondono alle seguenti richieste del conduttore: 1)- esaminare liberamente il quadro; 2)- esaminare l'ambiente materiale; 3)- indicare l'età delle persone; 4)- indicare che cosa facevano i personaggi prima dell'arrivo del visitatore inatteso; 5)- memorizzare quali abiti indossano le persone; 6)- memorizzare la posizione delle persone e degli oggetti nella stanza; 7)- indicare quanto tempo il visitatore inatteso è stato lontano dalla famiglia.

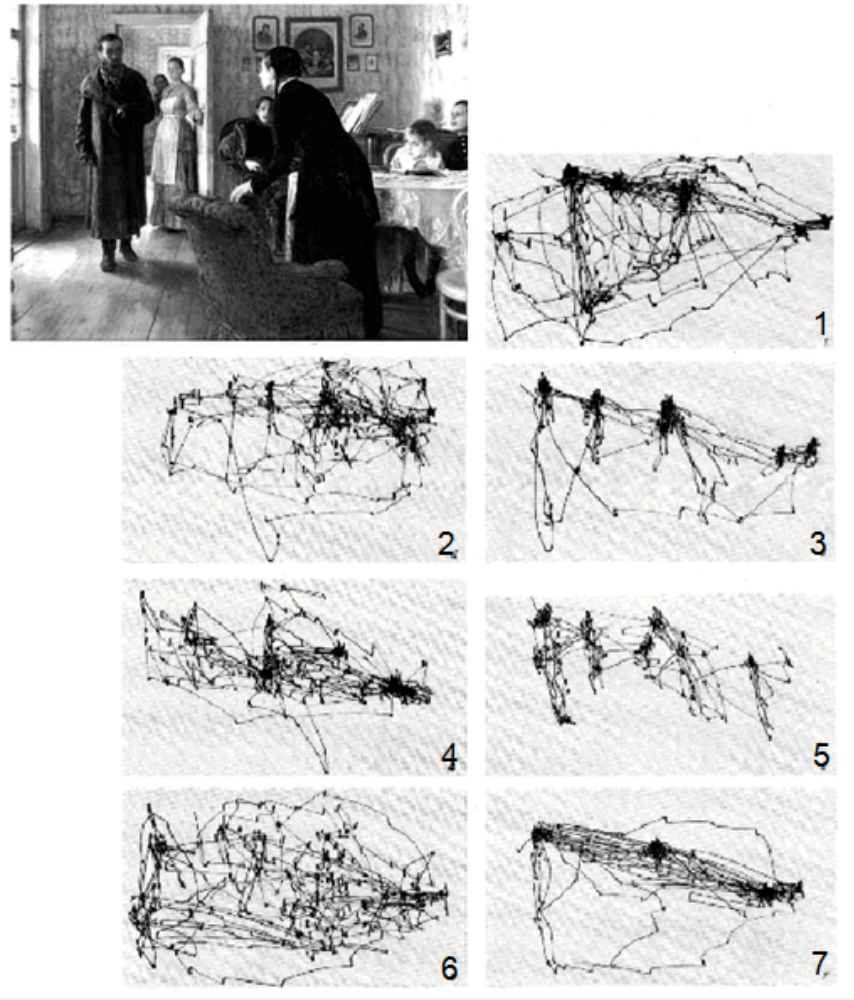


Figura 255. L'esperimento di Yarbus (1967)

Studi analoghi possono essere condotti sulle interfacce grafiche dei sistemi interattivi, tipicamente sulle pagine web. La Figura 256 mostra lo scanpath di una persona nell'esame di una pagina web. Come si vede, lo sguardo percorre la pagina irregolarmente, fissandosi sui diversi menu, sui titoli, secondo una logica che dipende dagli obiettivi (che in questo caso non conosciamo) dell'utente.

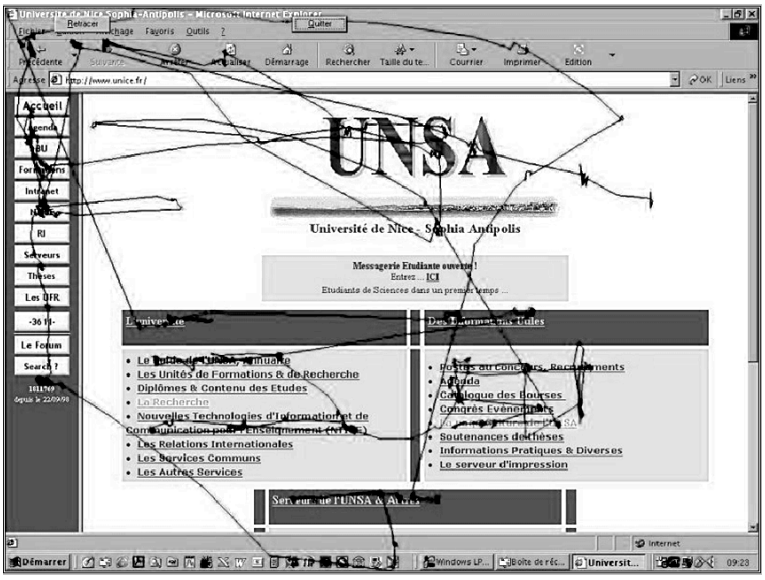


Figura 256. Scanpath di una pagina web (fonte: Université de Nice – Sophia Antipolis)

Le analisi delle interfacce con dispositivi di eye tracking possono fornirci utili informazioni per migliorarne l'usabilità. Jakob Nielsen e Kara Pernice hanno raccolto in un libro i risultati di un'ampia serie di esperimenti condotti con apparati di eye-tracking sulle pagine dei siti web, allo scopo di analizzare le relazioni fra la grafica del sito e la sua usabilità.[150] Questi esperimenti confermano quanto detto più sopra: le persone non esaminano una pagina web sempre nello stesso modo. A volte guardano inizialmente nel mezzo della pagina, perché vi sono attratti da un'immagine di loro interesse. Altre volte l'occhio si sofferma inizialmente sul logo, per sapere su che sito sono arrivati. Oppure esaminano innanzitutto l'area dove si trova il menu di navigazione orizzontale o, ancora, ignorano navigazione e figure per cercare subito qualche informazione nel testo. E così via. Indipendentemente dalla strategia seguita, lo scanpath è quasi sempre molto irregolare: l'occhio si muove qua e là sulla pagina, con percorsi spezzati del tipo di quello in Figura 256.

Sommando fra loro gli scanpath percorsi da numerosi utenti, è possibile costruire le cosiddette *heat-map*, che mostrano le aree della pagina sulle quali gli sguardi si sono, in media, maggiormente soffermati. Analizzando le heat-map di un gran numero di pagine, Nielsen e Pernice hanno osservato una configurazione prevalente a foma di F (Figura 257):

- gli utenti inizialmente tendono a esaminare, con un movimento orizzontale degli occhi, la parte superiore dell'area dei contenuti: questo rappresenta il tratto orizzontale superiore della F;
- poi, lo sguardo esplora la pagina un po' più sotto, anche qui con una scansione orizzontale, ma più breve: il tratto orizzontale corto della F;
- quindi, la pagina viene esaminata con un movimento verticale, tendenzialmente sulla sinistra: il tratto verticale della F.

Si deve comunque tener conto che si tratta di una configurazione media, ottenuta accumulando gli scanpath di numerosi utenti: i singoli percorsi individuali possono differire fra loro in modo significativo.

[150] J.Nielsen, K.Pernice, *Eyetracking Web Usability*, New Riders, 2010.

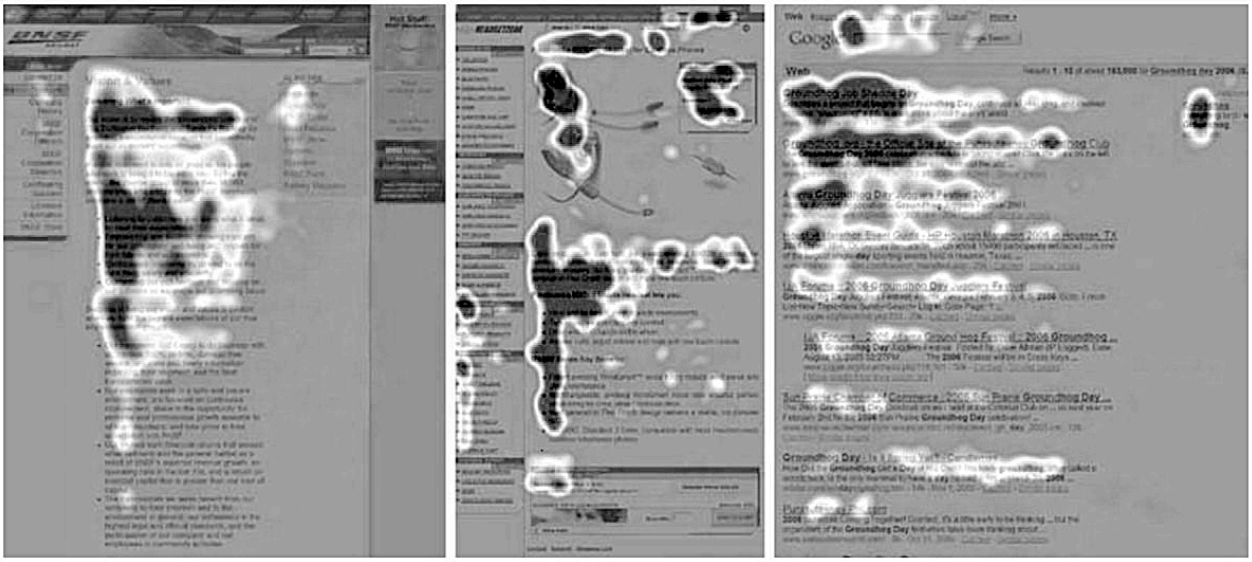


Figura 257. Heat-map di tre pagine web (da Jakob Nielsen, http://wwww.useit.com/alertbox/reading_pattern)

Analisi di questo tipo possono fornire informazioni molto utili per migliorare l'usabilità dei siti web. Studiando i cammini percorsi dallo sguardo dei diversi utenti sulle pagine, e correlandoli agli obiettivi che essi hanno nel condurre questo esame, possiamo ricavare utili informazioni per migliorarne il layout. Anche se i dispositivi di eye-tracking oggi non sono più invasivi, e sono di semplice utilizzo, si tratta tuttavia di analisi complesse, tutt'oggi utilizzate solo in casi molto particolari.

Ripasso ed esercizi

1. Riporta la formulazione dalle prime tre leggi della Gestalt, e spiega quali indicazioni possiamo trarre da esse per il visual design delle pagine di un sito web.
2. Esamina le pagine attuali del sito http://www.alitalia.it. Ritieni che i progettisti ne abbiano disegnato i layout in modo corretto, dal punto di vista delle leggi della Gestalt? Rispondi dettagliatamente per ciascuna pagina esaminata, indicando e motivando le eventuali modifiche che potrebbero migliorarne l'usabilità.
3. Confronta la versione corrente dei due siti http://www.alitalia.it e http://www.british-airways.it. Quale, a tuo parere, utilizza layout coerenti con le leggi della Gestalt?
4. Analizza la struttura dei pannelli che servono per definire le impostazioni del tuo browser. Come giudichi, dal punto di vista delle leggi della Gestalt, il layout di questi pannelli? Rispondi dettagliatamente, analizzando ogni singolo pannello, e indicando le eventuali modifiche che potrebbero migliorarne l'usabilità.

Approfondimenti e ricerche

1. Per approfondire le relazioni fra i meccanismi della visione e il design grafico, puoi usare: C.Ware, *Visual Thinking for Design*, Morgan Kaufmann, 2008, un libro molto interessante, rigoroso e ricco di esempi.

2. Esamina un insieme di siti appartenenti a una stessa categoria, a tua scelta (per esempio, siti di compagnie aeree, oppure siti della moda, siti di negozi online, ecc.), e raccogli esempi di soluzioni grafiche che, a tuo parere, sono criticabili dal punto di vista della teoria della Gestalt, spiegandone i motivi.

3. È molto importante per il progettista avere a disposizione delle linee guida per la progettazione di form usabili. In rete esiste molto materiale su questo argomento. Ricerca del materiale attendibile, e produci un sintetico elenco di linee guida che si riferiscono agli aspetti grafici dei form, citando, per ciascuna di esse, le fonti. Suggerimenti: ricerca con Google *form usability*. Un ottimo punto di partenza è l'articolo di Maurizio Boscarol: *L'usabilità dei form*, su http://www.usabile.it.

4. Cerca in rete qualche studio che analizza gli scanpath di pagine web. Suggerimento: ci sono diversi video interessanti su http://ww.youtube.com, e parecchio materiale sul sito di Jakob Nielsen, in http://www.useit.com/eyetracking.

13. Progettare il testo

Sintesi del capitolo

Questo capitolo tratta l'usabilità dei testi utilizzati nei sistemi interattivi, dal punto di vista della loro leggibilità fisica, comprensibilità e gradevolezza. Per quanto riguarda la leggibilità, dopo una sintesi dei concetti base della tipografia digitale, vengono fornite alcune linee guida per l'uso corretto dei caratteri nei testi su monitor. Per quanto riguarda la comprensibilità dei testi in lingua italiana, si introduce l'indice Gulpease, e le nozioni di vocabolario di base e di vocabolario fondamentale. Vengono quindi riassunte le principali indicazioni fornite nei manuali di stile, e descritto il particolare stile che conviene utilizzare nei testi per il Web, ricchi di link ipertestuali. Il capitolo si chiude con esempi di uso "emozionale" del testo.

L'usabilità del testo

Secondo il vocabolario il *testo* (dal latino *textus*, con significato originario di tessuto o trama) è "l'insieme delle parole che compongono uno scritto". Un testo è composto di *caratteri*. I segni grafici che sono utilizzati per rappresentare un testo sono detti *glifi*.[151] Il testo è un componente molto importante di molti sistemi interattivi, e può comparire in vari contesti: nei messaggi di errore o di avvertimento, nei sistemi di help online, nelle pagine di un sito web. In questo capitolo considereremo il testo come componente a se stante, allo scopo di individuare delle linee guida per la sua stesura, per una migliore usabilità complessiva del sistema in cui è inserito.

Innanzitutto, osserviamo che la nozione di usabilità può essere utilizzata anche per i testi. Applicando la definizione di pag.66, l'usabilità di un testo è "il grado con cui esso può essere usato da specificati utenti per raggiungere specificati obiettivi con efficacia, efficienza e soddisfazione in uno specificato contesto d'uso". Come abbiamo già osservato, questa definizione è di tipo operativo, e suggerisce di definire delle metriche opportune per misurare l'efficacia, l'efficienza e la soddisfazione raggiunte, in questo caso, dallo specifico testo.

Nel caso di un testo, efficacia ed efficienza possono essere misurate in diversi modi. Tipicamente, l'efficacia potrebbe misurare l'accuratezza e completezza con cui il testo viene "compreso" dai suoi lettori. Secondo questa interpretazione, un testo sarebbe considerato efficace quando il lettore fosse in grado di acquisirne tutti i contenuti ("completezza") in dettaglio ("accuratezza"). L'efficienza potrebbe invece essere misurata dal tempo medio impiegato dai lettori per raggiungere determinati obiettivi di accuratezza e completezza. In questo senso, un testo A sarà considerato più usabile di un testo B se, a parità di condizioni (per esempio, argomento, lunghezza, formato, ecc.) potrà essere "compreso", con lo stesso livello di accuratezza e completezza, in un tempo più breve. La soddisfazione dell'utente, potrebbe essere valutata chiedendo a un campione di soggetti di esprimere un giudizio di gradimento: ovviamente, non dei contenuti, ma del modo in cui essi sono comunicati attraverso il testo.

Come sempre, efficacia, efficienza e soddisfazione sono relative a una specifica tipologia di utenti e a un determinato contesto d'uso. Per esempio, utenti con un basso livello di scolarità sono in grado di padroneggiare un lessico più ristretto di utenti in possesso della laurea. D'altra parte, testi di tipo tecnico, facilmente comprensibili a uno specialista del settore, potrebbero non essere comprensibili ad altri lettori, anche se con buona scolarità. Infine occorre tenere presente il contesto d'uso, che è particolarmente importante quando il testo è inserito in un sistema interattivo. Infatti, per esempio, la situazione di chi legge un romanzo – o un saggio - è molto diversa da chi legge un testo di help mentre sta usando un'applicazione software. Nel primo caso, l'attenzione del lettore è concentrata sul testo che, per così dire, ne costituisce l'oggetto primario. Nel secondo caso, l'utente è concentrato sul compito che sta svolgendo con il sistema, e il testo assume una funzione collaterale, di ausilio: il contesto d'uso è completamente diverso. Anche il medium

[151] Il termine glifo deriva dal greco γλυφω, "incidere". Mentre un carattere (o grafema) è una unità di testo, un glifo è una unità grafica. Un glifo, a sua volta, è composto da *tratti* grafici. Glifi e caratteri non sono in corrispondenza biunivoca. Per esempio, la coppia di caratteri "fi" corrisponde a un unico glifo, perché i due caratteri sono graficamente legati.

utilizzato è parte del contesto: un testo di un quotidiano a stampa viene fruito in condizioni molto diverse da quelle del testo di un quotidiano online.

Bastano questi pochi esempi per comprendere che l'usabilità di un testo dipende da un grande numero di elementi. Per ricavare delle linee guida utili alla progettazione di testi usabili dovremo quindi decomporre il problema complessivo in problemi più semplici, da affrontare uno per volta. Potremo così studiare separatamente i diversi fattori che contribuiscono all'usabilità dei testi, per ricavare delle indicazioni specifiche, da ricomporre poi in un quadro complessivo che tenga conto di tutte queste analisi. Per i nostri scopi è molto utile analizzare il testo da tre punti di vista: quello della sua *legibility*, quello della sua *readability* e quello della sua *struttura para-testuale*.

Legibility

La *legibility* di un testo è la facilità con cui riusciamo a discriminare le singole lettere che lo compongono. L'analisi della legibility considera la struttura tipografica di un testo: la forma, la dimensione, il colore dei caratteri, il modo in cui essi sono disposti sulla pagina in rapporto gli uni con gli altri. In relazione a questi elementi possiamo studiare la minore o maggiore facilità con cui un lettore può distinguere un carattere dall'altro, sui differenti supporti tecnologici utilizzati (carta o monitor). Quando si analizza la legibility di un testo, non ci si occupa del suo significato, e della facilità o meno con cui il lettore può comprenderne il contenuto, ma soltanto della rappresentazione grafica e della riconoscibilità in rapporto al suo sistema visivo. La legibility può essere misurata con tecniche sperimentali, e posta in rapporto, come vedremo, alle diverse caratteristiche tipografiche del testo stesso.

Questi esperimenti possono essere realizzati in modo relativamente semplice, perché i fattori in gioco sono sostanzialmente di natura oggettiva (caratteristiche fisiche dei glifi, del medium utilizzato per visualizzarli, dell'ambiente di prova). I soggetti per gli esperimenti devono essere selezionati sulla base di una normale acuità visiva e capacità di discriminare i colori, ma non sulla base delle loro competenze linguistiche, poiché non devono comprendere il significato dei testi che vengono loro presentati. Gli studi sperimentali sulla legibility ci possono fornire indicazioni pratiche sul migliore uso delle molte soluzioni tipografiche di cui oggi disponiamo. Questi studi richiedono, da un lato, la conoscenza dei concetti e delle tecniche della tipografia digitale e, dall'altro, i metodi di chi studia la percezione umana e, in particolare, il sistema visivo dell'uomo.

Readability

La *readability* di un testo è la sua leggibilità complessiva.[152] In questo caso, ciò che importa soprattutto è la sua struttura linguistica: l'ampiezza del lessico utilizzato, la complessità delle strutture sintattiche e semantiche. Analisi di questo tipo sono ovviamente molto più complesse, poiché devono considerare la dimestichezza del lettore con il lessico e i costrutti utilizzati nel testo, e la conseguente facilità o difficoltà di comprensione. Esse sono condotte con i metodi e i concetti della linguistica. La grande difficoltà nello studio della readability deriva dal fatto che i processi cognitivi coinvolti nella lettura sono ancora poco noti. I vari livelli di elaborazione (visuale, lessicale, sintattica, semantica) sono strettamente interconnessi (Figura 258): è quindi molto difficile organizzare degli esperimenti che permettano di studiare un singolo fattore isolandolo da tutti gli altri. Nella stessa letteratura tecnica sull'argomento (soprattutto quella meno recente), le nozioni stesse di legibility e readability vengono a volte confuse; ciò fa sì che per molti esperimenti gli obiettivi non risultino chiari.

[152] In italiano non esiste un termine alternativo a *leggibilità*, che permetta di distinguere i due concetti. Per evitare ambiguità, saremo quindi costretti ad utilizzare sempre la terminologia in inglese.

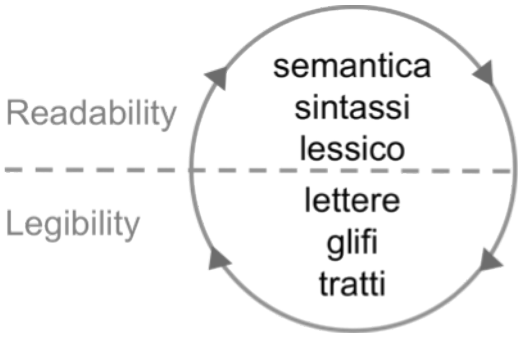


Figura 258. I processi cognitivi che sovraintendono alla lettura e comprensione dei testi sono strettamente connessi[153]

Struttura para-testuale

Come indicato dal prefisso *para*,[154] il termine *para-testo* indica tutto ciò che sta "accanto" o "di contorno" ma anche "al servizio" del testo. Analizzare la struttura para-testuale significa considerare aspetti quali l'organizzazione del testo in capitoli e di questi in sezioni, l'esistenza e la forma di titoli, riassunti, tabelle, schemi, figure, decorazioni, prefazioni e postfazioni, eccetera. In altre parole, studiare tutto ciò che sta "attorno" al testo e, in qualche modo, ne dirige o orienta la fruizione. Questi elementi sono particolarmente importanti per il progettista di sistemi interattivi, nei quali il testo è spesso immerso in un contesto para-testuale a sua volta interattivo. Tipico è il caso dei siti web, che contengono apparati di navigazione (menu, titoli delle sezioni, breadcrumb, thumbnail, ecc.) e collegamenti ipertestuali interni agli stessi testi, per permettere al lettore di muoversi rapidamente da una porzione di testo all'altra. Tutti questi elementi para-testuali non soltanto orientano la fruizione ma la rendono effettivamente possibile. Sono quindi importanti ai fini dell'usabilità complessiva del testo. Lo studio e il confronto delle varie soluzioni para-testuali possibili nei sistemi interattivi sono condotti prevalentemente nell'ambito disciplinare della human-computer interaction.

Nel seguito di questo capitolo tratteremo legibility, readability e para-testi interattivi separatamente.

La tipografia digitale

Un *tipo di carattere* o *font*[155] è un insieme di caratteri con un certo stile grafico. Esso contiene *caratteri alfabetici*, in versione minuscola (*lowercase*) e maiuscola (*uppercase*), *cifre* e *caratteri speciali*. Storicamente, le minuscole sono state introdotte più tardi delle maiuscole. Originalmente, infatti, l'alfabeto latino era scritto solo in lettere maiuscole, ben delimitate sopra e sotto da due linee ideali. Le minuscole furono introdotte in seguito, per permettere una scrittura più rapida con la penna. Le minuscole hanno altezze diverse, per la presenza di *ascendenti* (per esempio, nelle lettere b, d, t) e *discendenti* (per esempio, nelle lettere g, p, q, y).

I caratteri appartenenti a un certo font possono essere rappresentati con glifi diversi, secondo le seguenti proprietà:[156]

- *stile* (*font-style*): *normale* (*normal*), *corsivo* (*italic*) o *obliquo* (*oblique*). Di solito, lo stile obliquo è ottenuto con algoritmi che trasformano i glifi dello stile normale inclinandoli verso destra, mentre lo stile corsivo utilizza glifi disegnati appositamente;

[153] Immagine: cortesia Luca Colombo.

[154] *Para* deriva dal greco παρα, che significa "presso, accanto, oltre".

[155] Il termine inglese *font* proviene dal francese medioevale *fonte* (femminile), ovvero " fuso", in riferimento al procedimento tradizionale di stampa a caratteri mobili, nel quale i singoli caratteri venivano realizzati fondendo il metallo. In italiano, è controverso se debba essere usato al maschile o al femminile. Noi lo useremo al maschile, seguendo l'uso prevalente e le indicazioni dei dizionari Garzanti e Zingarelli.

[156] La nomenclatura in uso è molto variabile. Per i termini in lingua inglese, utilizziamo qui la terminologia del linguaggio dei CSS (Cascading Style Sheets), che può essere ormai considerato uno standard della tipografia digitale.

- *variante* (*font-variant*): *normale* (*normal*) o *maiuscoletto* (*small-caps*). Nel maiuscoletto, le minuscole sono simili alle maiuscole, ma un po' più piccole e con proporzioni leggermente diverse (Figura 259).
- *peso* (*font-weight*): i tratti dei glifi possono essere di spessore *normale* o più spessi (*bold*), in diversi gradi. La terminologia è variabile: *nero*, *neretto*, *grassetto*, *leggero*, ecc. Alla *boldness* si possono dare dei valori numerici.
- *Dimensione* o *corpo* (*font-size*): la distanza verticale fra due *linee di base* (*baseline*) contigue (vedi Figura 259). Come si vede in figura, è uguale alla distanza verticale fra il margine superiore dell'ascendente più alta e il margine inferiore della discendente più bassa, più "un po'", affinché i caratteri di due righe contigue non si tocchino, quando fra esse non s'inserisce alcuno spazio[157]. Si possono usare diverse unità di misura. La più utilizzata è il *punto tipografico*, indicato con la sigla pt. Un punto tipografico vale 1/72esimo di pollice, pari a 0,35277 mm.[158] Per esempio, i caratteri di questo libro sono di corpo 10 pt (???). Per *occhio medio* (*x-height*), si intende l'altezza delle minuscole, assunta convenzionalmente come quella della lettera x (Figura 259). Si usa la lettera x perché le minuscole, anche quelle senza ascendenti e discendenti, non hanno tutte la stessa altezza. Infatti, alle lettere tonde si danno spesso dimensioni più grandi di quelle lineari, per applicare una correzione ottica senza la quale apparirebbero al lettore più piccole delle altre. Per lo stesso motivo, l'*altezza delle maiuscole* è convenzionalmente quella della lettera E.

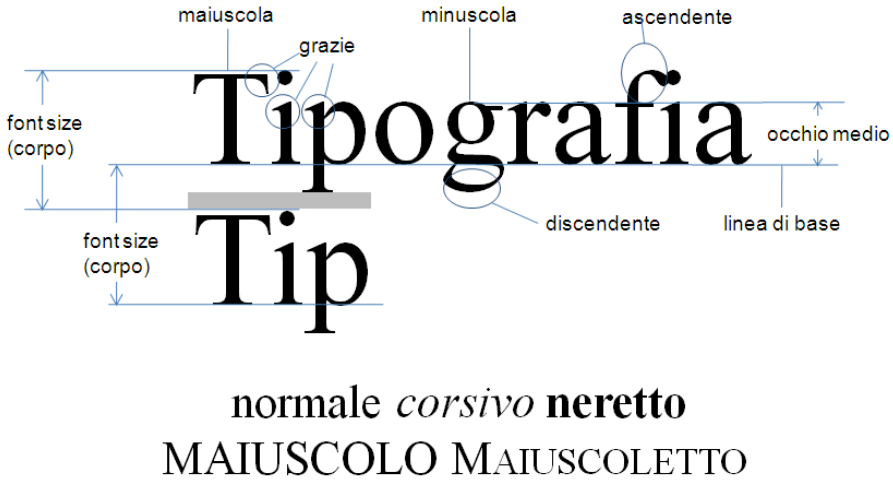


Figura 259. Terminologia tipografica (il font è Times New Roman)

Fra le proprietà del testo nella sua globalità ci sono le seguenti:

- *decorazione* (*text-decoration*), per esempio: *sottolineato* (*underline*), *cancellato* (*line-through*), *lampeggiante* (*blink*);
- *spaziatura delle lettere* (*letter-spacing*): può essere quella di default per il font utilizzato (*normal*), o una spaziatura aggiuntiva, di specificato valore;
- *spaziatura delle parole* (*word-spacing*): quella di default per il font, o una spaziatura aggiuntiva;
- *allineamento* (*text-align*): *a bandiera sinistra* (*left*), *a bandiera destra* (*right*), *centrato* (*center*), *giustificato* o "a pacchetto" (*justify*, Figura 260);
- *rientro* (*text-indent*): il valore di rientro della prima riga di ogni paragrafo;
- *interlinea* (*line-height*): è la distanza fra le linee di base di due righe di testo contigue.

[157] La definizione esatta di corpo, nella tipografia tradizionale, è la misura tra spalla superiore e spalla inferiore del carattere, cioè l'altezza totale del blocchetto di piombo che contiene il carattere.
[158] Esistono svariati punti tipografici. Questo è il Pica PostScript, quello più usato nei sistemi di desktop publishing.

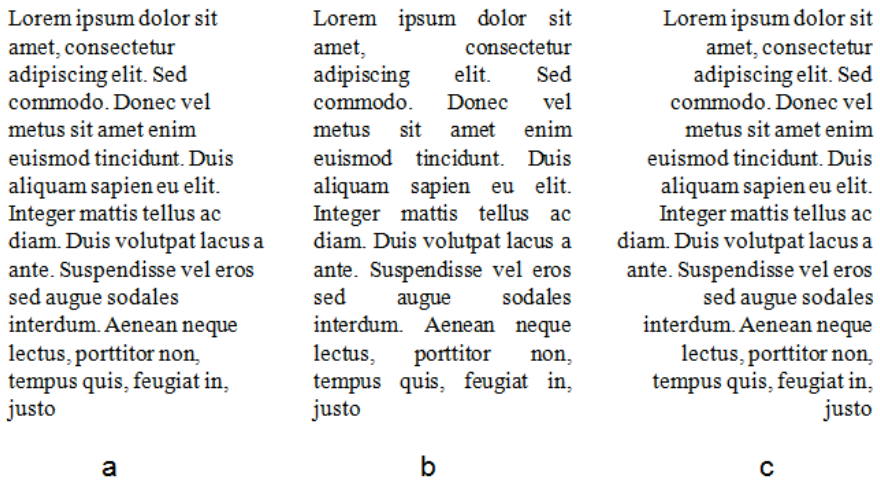


Figura 260. Allineamento a bandiera sinistra (a), destra (c), a pacchetto (b)

I font si suddividono in due categorie principali: *graziati* o *senza grazie*. I caratteri graziati (o *serif*) hanno particolari terminazioni dei tratti delle lettere, chiamati appunto *grazie* (Figura 259). L'uso delle grazie deriva dai caratteri lapidari romani, dove era molto difficile scalpellare nel marmo angoli di novanta gradi necessari a terminare le aste. Le grazie servivano allora a evitare (o nascondere) le sbrecciature. I font senza grazie sono chiamati anche *bastoni*, o *sans-serif*.

Un'altra distinzione importante è quella fra *print-font* e *screen-font*. I primi sono i font tradizionali, disegnati principalmente per la stampa. I secondi, molto più recenti, sono progettati in primo luogo per una resa ottimale sui monitor. Gli screen-font disponibili sono molto meno numerosi dei print-font, ma acquisiscono un'importanza sempre maggiore. Con l'evoluzione e la diffusione della tecnologia, infatti, il video è visto sempre più come il supporto prevalente per la visualizzazione dei documenti, e non soltanto come un comodo mezzo per controllare l'anteprima del prodotto finale, che sarà stampato su carta.

Il disegno di un buono screen-font richiede l'uso di tecniche specifiche. Infatti, le tecnologie di stampa e di visualizzazione su schermo sono molto diverse, e producono immagini con differente *risoluzione*. La risoluzione è la densità di punti elementari che compongono un'immagine, rapportata a una dimensione lineare, normalmente il pollice (*inch*, 2,54 cm). La risoluzione di un monitor è molto inferiore a quella ottenibile sulla carta stampata. Su un monitor abbiamo normalmente una risoluzione nel range 75-130 ppi (*pixel-per-inch*, pixel per pollice) circa.[159] Una stampa di buona qualità parte invece da una risoluzione di 300 dpi (*dot-per-inch*, punti per pollice[160]), utilizzata nei normali procedimenti di stampa tipografica, ma può arrivare, nelle stampanti laser commerciali, a 2400 dpi e più, molto di più. In definitiva, la risoluzione dei monitor è, oggi, circa un terzo di quella delle stampe di qualità standard: la differenza, osservata con una lente d'ingrandimento, è sostanziale, come mostra l'esempio di Figura 261.[161]

[159] Nei computer con schermo piccolo, o nei telefoni cellulari, è desiderabile una densità maggiore, in quanto questi schermi sono destinati ad essere visti da vicino. Una serie di risoluzioni standard si trova in http://en.wikipedia.org/wiki/List_of_displays_by_pixel_density#Standard-aspect_Display.

[160] Non bisogna confondere i pixel che formano l'immagine di un monitor con i punti (dot) che formano l'immagine nei procedimenti a stampa: sono cose diverse e hanno normalmente dimensioni diverse. Quindi, non necessariamente un punto sulla carta corrisponde a un pixel sul monitor. Analogamente, non bisogna confondere i punti tipografici (pt) usati come unità di misura per la dimensione dei font, con i punti (dot) prodotti dalle tecnologie di stampa. Un pt ha, come abbiamo visto, il valore fisso di 1/72esimo di pollice, mentre la dimensione di un dot dipende dalla tecnologia di stampa.

[161] La risoluzione dei monitor è destinata a migliorare in futuro, ed esistono già tecnologie non commerciali che permettono una risoluzione superiore ai 500 ppi. In ogni caso, non sarebbe molto utile realizzare monitor con risoluzioni superiori ai 300 dpi. Infatti, questa è all'incirca la risoluzione che un occhio umano di normale acuità visiva riesce a discriminare alla normale distanza di lettura da un monitor.



Figura 261. Ingrandimento della lettera "g" del font Bembo in corpo 10 pt, riprodotta a diverse risoluzioni: 72 dpi, 150 dpi, 300 dpi, 1200 dpi[162]

Il progettista di uno screen-font deve curare che le forme dei glifi, in tutte le loro varianti, si adattino in modo ottimale alla griglia di pixel che li rappresenterà sullo schermo. Quindi, dovranno avere tratti curvilinei ridotti al minimo, uno spessore dei tratti costante e consistente anche nello stile normale, spazi interni alle lettere e fra le lettere ampi e regolari e, nel caso dei font serif, grazie squadrate e non troppo sottili. Inoltre, le coppie di lettere che si possono lambire dando luogo a *legature* (come, per esempio, ft, fi, fl, ff), devono essere ben separate, e caratteri simili devono distinguersi bene anche a bassa risoluzione (o, O e 0, oppure 1, I, J, l, i). In pratica, gli screen-font vengono progettati inizialmente su una griglia, e soltanto in seguito disegnati con tratti curvilinei. Le differenze, a un'analisi non superficiale, possono essere notevoli, come in tutti i caratteri dell'esempio in Figura 262.



Figura 262. Screen-font (Trebuchet MS, in alto) e print-font (Cantoria MT) a confronto[163]

Anche se i font esistenti sono moltissimi, quelli utilizzati nei sistemi interattivi non sono molti. Nelle applicazioni web si riducono sostanzialmente a quelli presenti in tutti i principali sistemi operativi in circolazione, circa una diecina. Infatti, nella tecnologia attuale del Web, i font non sono inviati dal server al browser insieme alla pagina che li usa, ma devono essere pre-installati nel sistema client. Utilizzare font diversi da quelli già presenti significherebbe quindi costringere gli utenti a installarli prima di poter vedere la pagina.

La Figura 263 mostra alcuni dei font più diffusi.

[162] Questa figura è tratta dalla tesi di laurea magistrale in Teoria e tecnologia della comunicazione di Luca Colombo, *The New Web Typography*, Università degli Studi di Milano Bicocca, A.A.2007/2008 (relatore: Letizia Bollini).
[163] Il corpo è 14 pixel, senza anti-aliasing, ovviamente in forte ingrandimento. Cortesia di Luca Colombo, vedi nota precedente.



Figura 263. Alcuni font molto diffusi

- Il *Times New Roman* è probabilmente il font graziato più usato sulla carta stampata. Esso fu progettato da Stanley Morrison per conto del giornale londinese *The Times*, che lo adottò nel 1932 in sostituzione del font che il giornale usava in precedenza, chiamato *Times Old Roman* (da cui il nome). Aveva lo scopo di essere ben leggibile anche con caratteri di piccole dimensioni stampati sulla carta di cattiva qualità usata durante la Grande Depressione degli anni '30. Il disegno dei caratteri, alti e stretti, era specificamente concepito per ridurre i fastidiosi spazi bianchi derivanti dall'allineamento "a pacchetto" dei testi nelle colonne del giornale (Figura 260b). Il Times New Roman fu usato dal *Times* per quaranta anni. Dal 1972 fu sostituito più volte, sempre però con font di aspetto simile.
- Il *Georgia* è uno screen-font graziato, disegnato da Matthew Carter per conto della Microsoft nel 1993. Fu progettato per essere leggibile sui monitor anche in corpo piccolo, ed è molto simile al Times New Roman, rispetto al quale ha tuttavia diversi miglioramenti: le linee che compongono le lettere sono un po' più spesse, e il loro spessore varia meno all'interno di uno stesso carattere. A parità di dimensione del font, le lettere sono un po' più larghe e alte; l'altezza della x è lievemente più grande; le grazie sono più larghe e con tratti meno obliqui. Non ci sono legature e le lettere sono più "verticalizzate", per permettere una migliore resa sul monitor (Figura 264).
- L'*Arial* è un font senza grazie adatto sia ai monitor sia alla carta stampata. Fu progettato nel 1982 ispirata a Helvetica, un font disegnato nel 1957 che ebbe grande successo nel mondo della grafica e del design negli anni '70. Arial fu usato da Microsoft in Windows 3.1, ed è oggi molto diffuso sul Web.
- Il *Verdana* è uno screen-font senza grazie, diventato quasi uno standard di fatto per i testi su monitor. Progettato da Matthew Carter per la Microsoft per massimizzare la leggibilità anche in corpo piccolo (fino a 4 pt) e su monitor a bassa risoluzione, fu rilasciato nel 1996 per Windows 95. Esso possiede caratteri larghi e ben spaziati, minuscole alte e ben leggibili, ed ha il vantaggio di differenziare bene i caratteri simili, come per esempio la i maiuscola (che per questo ha le grazie), la elle minuscola e la cifra 1, che in altri font utilizzano lo stesso glifo. Verdana e Georgia sono i due screen-font per eccellenza: contrariamente alla prassi tradizionale di disegno dei caratteri, Carter progettò i glifi di Georgia e Verdana disegnandoli inizialmente come bitmap, e solo in seguito tracciandone i contorni.

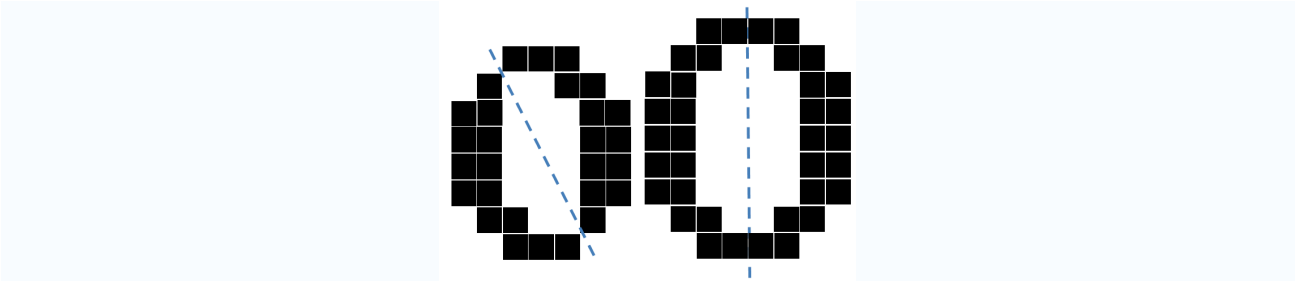


Figura 264. La lettera O in Times New Roman (sinistra) e Georgia (destra). La "verticalizzazione" dei caratteri migliora la resa sul monitor

Un font i cui glifi sono di larghezza variabile è detto *proporzionale,* mentre un font con glifi di larghezza fissa è detto *non proporzionale* (o *monospace* o a larghezza fissa). Per esempio, nei font proporzionali la "w" e la "m" sono della stessa larghezza mentre la "i" ha una larghezza inferiore. Però, in molti font proporzionali tutte le cifre hanno la stessa larghezza, per permettere l'allineamento di colonne di numeri. I font proporzionali sono generalmente considerati più eleganti e più facili da leggere e sono quindi quelli più comunemente utilizzati sulla carta stampata e sui monitor.

I font non proporzionali furono creati per le macchine per scrivere e per le stampanti a impatto, poiché lo spostamento del carrello dopo la stampa di un carattere era sempre della stessa misura. Anche i primi monitor, che utilizzavano un unico font, avevano caratteri di larghezza fissa. Oggi, i caratteri non proporzionali sono usati solo quando ci siano particolari esigenze d'incolonnamento, come, per esempio, nel codice dei programmi. In questo caso, infatti, le righe di testo devono essere rientrate con precisione a più livelli, per mostrare la nidificazione delle istruzioni. L'esempio tipico di font a spaziatura fissa è il *Courier*, disegnato nel 1955 per le macchine per scrivere dell'IBM (Figura 263).

Esistono anche numerosi font destinati a scopi particolari: quelli che riproducono la scrittura a mano libera, che contengono simboli matematici o simboli grafici "astratti" (chiamati *dingbat font*), e così via. Per esempio *Symbol*, con le lettere dell'alfabeto greco e diversi simboli matematici e *Webdings*, un dingbat font della Microsoft (Figura 265).

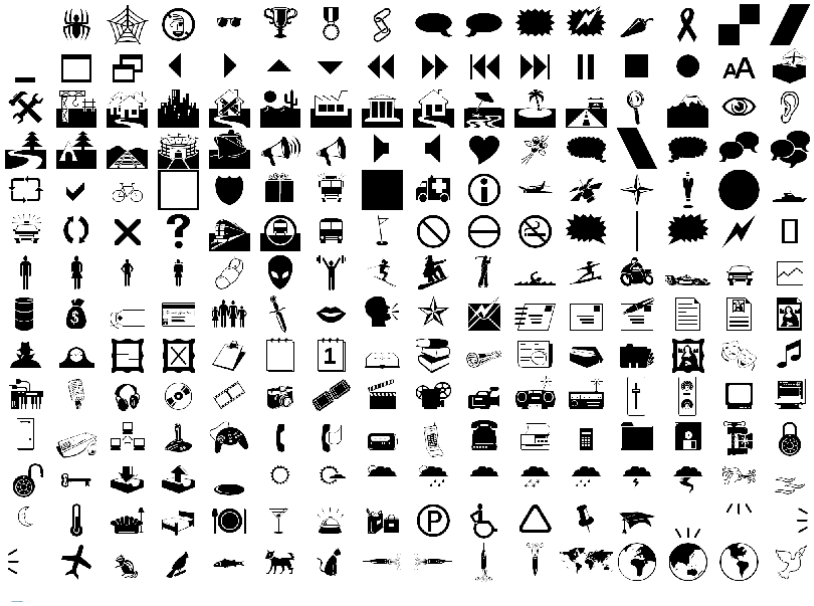


Figura 265. Il font Webdings

Legibility

Prima di esaminare i risultati degli studi sulla legibility, ricordiamo brevemente i processi che avvengono nel nostro sistema visivo durante la lettura di un testo scritto. Essi sono diversi da come intuitivamente li immaginiamo: il nostro occhio non esamina sequenzialmente il testo carattere per carattere e da sinistra a destra, come ci sembrerebbe naturale, ma riconosce le lettere di una parola (e a volte di parole contigue, se brevi) in parallelo. Inoltre, si muove, per così dire, a scatti. Questo processo può essere analizzato in modo molto accurato con apparati di *eye tracking*, con i quali si può visualizzare il percorso del nostro sguardo sul testo (*scanpath*), ed eseguire tutte le misure relative.

Durante la lettura, l'occhio esamina una parola per un tempo sufficiente a riconoscerla (*fissazione*), e quindi sposta l'asse visivo, con un movimento rapidissimo durante il quale non viene acquisita alcuna informazione, sulla parola successiva, dove avviene un'altra fissazione (Figura 266). Durante la fissazione di una parola, l'occhio ricava anche informazioni preliminari sulla parola successiva. Il movimento rapido (*saccade*) nella lettura silenziosa ha un'ampiezza tipica di circa 2 gradi (corrispondenti approssimativamente a 8 caratteri) e una durata molto breve, dell'ordine dei 20 msec. Ogni fissazione ha una durata media di circa un quarto di secondo. Nella Figura 266, le parole "sweat" e "pain" sono brevi, e vengono riconosciute nell'ambito della stessa fissazione. La parola successiva, "and", molto frequente, non viene fissata, poiché le informazioni acquisite durante la fissazione precedente sono sufficienti per il riconoscimento.[164]

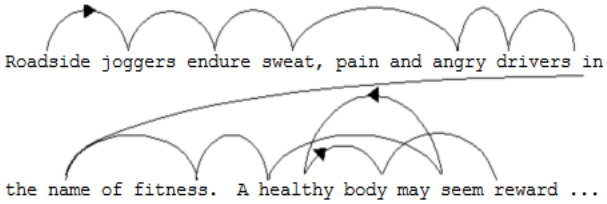


Figura 266. Fissazioni e saccadi nel processo di lettura

I dati citati rappresentano valori medi, e variano in funzione delle caratteristiche del testo. Per esempio, sulle parole difficili le fissazioni sono più lunghe, e può succedere che certi blocchetti di caratteri vengano riesaminati se, per qualche motivo, la lettura non va a buon fine, come si vede nell'esempio in Figura 266. In effetti, circa il 15% del tempo complessivo è utilizzato in queste riletture. Anche le caratteristiche tipografiche del testo influenzano i tempi di lettura. Pertanto, è molto utile eseguire delle misure sperimentali, basate su una metodologia rigorosa, che mettano in correlazione i tempi di lettura del testo con i parametri che ne definiscono le varie caratteristiche visive (per esempio, font, dimensioni, colore). Ciò permette di ricavare delle linee guida pratiche per il progettista, che può scegliere le caratteristiche tipografiche del testo in modo da facilitare il processo di lettura sulla base di fondamenta scientifiche rigorose.

Nella letteratura esistono numerose linee guida di questo tipo, ma non sempre basate su evidenze scientifiche. Ciò è comprensibile, poiché la tipografia è arte ormai antica, che precede di secoli le conoscenze che ci permettono oggi di condurre esperimenti rigorosi. In assenza di queste, grafici e tipografi hanno comunque elaborato delle indicazioni, che si sono col tempo trasformate in pratiche consolidate fra i professionisti. Occorre tuttavia essere cauti nell'applicarle: si tratta, a volte, d'indicazioni non confermate sperimentalmente e, non di rado, fra loro contraddittorie. La confusione esistente oggi su questo argomento è ulteriormente amplificata dal fatto che queste indicazioni sono riportate su numerosissimi siti Internet, senza citare la fonte e in modo acritico. Questo meccanismo di "passa-parola" contribuisce al consolidamento di vere e proprie leggende senza alcuna base reale.

Peraltro, anche le analisi sperimentali oggi disponibili spesso non sono conclusive, e sono difficili da interpretare. Questo è dovuto a cause diverse. Per prima cosa, le tecnologie sono in rapida evoluzione, ed è chiaro che esperimenti

[164] Questo è il modello del processo di lettura oggi più comunemente accettato, ma ce ne sono altri. Per maggiori informazioni si veda K.Larson, *The Science of Word Recognition* (2004), in http://www.microsoft.com/typography/ctfonts/wordrecognition.aspx , da cui è tratta la Figura 266.

condotti su monitor di tipo diverso sono difficilmente comparabili. Inoltre, i parametri che possono influenzare i risultati di questi esprimenti sono numerosi ed è difficile concepire degli esperimenti che isolino i diversi fattori. Gli esperimenti di solito misurano i tempi di scansione visiva di un testo da parte di un campione di lettori con normale acuità visiva, al variare di diversi parametri tipografici, per esempio font, dimensioni, contrasto fra carattere e sfondo, tinta, spaziature, allineamenti, e in condizioni di visualizzazione controllate. Per esempio, ai lettori usati nel test si chiede di esaminare sequenzialmente un testo alla ricerca di una o più parole date. All'inizio dell'esame viene fatto partire un cronometro, che viene fermato quando la parola cercata è riconosciuta, per misurare il tempo trascorso (tempo di reazione, o *reaction time*). In tal modo si costringe il lettore a scansionare tutti i caratteri del testo, senza analizzarne il significato. Questo è importante, per evitare che i processi cognitivi di più alto livello (Figura 258) interferiscano modificando i risultati dell'esperimento. Naturalmente, il tempo (efficienza) viene rapportato al *tasso di errore* nei riconoscimenti (efficacia). In diversi esperimenti, però, si richiede ai soggetti, dopo la lettura, di rispondere a semplici domande sul *contenuto* del testo. Ciò richiede da parte dei soggetti, oltre al puro riconoscimento dei caratteri (legibility), anche una parziale analisi lessicale o semantica (che riguarda più propriamente la readability). Ciò fa sì che questi esperimenti siano poco significativi o comunque d'incerta interpretazione.

Sfortunatamente, gli esperimenti finora condotti suggeriscono che non esistono delle regole semplici, che possano essere utilmente seguite in ogni situazione. Nel seguito di questa sezione tenteremo, senza entrare nei dettagli, di riassumere quelle indicazioni pratiche che, a oggi, sembrano ragionevolmente supportati da evidenze sperimentali.

Lettura su carta e su monitor

Da molti anni è diffusa la convinzione che la lettura a video sia più faticosa e più lenta della lettura sulla carta stampata. Questa convinzione ha origine da vari esperimenti condotti negli anni '80, e da allora frequentemente citati, i quali indicavano come la lettura su monitor fosse più lenta di circa il 25% di quella sulla carta. Ciò viene spiegato in ragione della sostanziale diversità delle due tecnologie: le caratteristiche fisiche, le possibilità di regolazione, l'angolo di lettura e, in generale, l'interazione del lettore con i due mezzi. Per esempio, sulla carta il lettore può seguire col dito o con la penna la scansione del testo durante la lettura. E naturalmente, come già ricordato, il video ha una risoluzione molto inferiore a quella della stampa.

Studi successivi hanno mostrato che questa affermazione è piuttosto dubbia. Già nel 1992, a conclusione di un'ampia rassegna della letteratura sull'argomento, A.Dillon scriveva: "Sebbene la lettura sullo schermo di un computer possa essere più lenta e occasionalmente meno accurata della lettura dalla carta, è probabile che non esista una sola variabile responsabile della differenza. È quasi certo che le cause non sono dovute né a problemi della tecnologia né di chi legge. La lettura su video può essere altrettanto veloce e accurata della lettura sulla carta. Invariabilmente, ciò che è cruciale è la qualità dell'immagine presentata al lettore."[165] Da allora, come si è visto, sono stati realizzati font specificamente progettati per il video, e la tecnologia dei monitor ha fatto significativi passi avanti. In effetti, una rassegna più recente (1997) conclude che "nonostante il fatto che questi studi mostrino differenze fra la lettura su carta e lettura online, essi sono in genere poco rilevanti o inconsistenti. Anche la scoperta che la lettura su monitor è significativamente più lenta della lettura su carta è stata messa in dubbio da esperimenti recenti che dimostrano come i miglioramenti nelle tecnologie video riducano le differenze, e possano perfino eliminarle." [166]

È necessario considerare, comunque, che l'uso reale di un sistema interattivo raramente avviene in condizioni ottimali. Il caso tipico è quello delle pagine web, che sono visualizzate su monitor di qualità molto variabile e in situazioni di fruizione diverse e fuori dal controllo del progettista. Quando leggiamo un testo a stampa, possiamo facilmente controllarne l'orientamento e spostarci per usufruire di migliori condizioni d'illuminazione. Chiaramente, con un monitor questo non è sempre fattibile. È pertanto consigliabile usare cautela nella progettazione dei testi destinati alla lettura su video, e assumere che il lettore si trovi nelle peggiori condizioni di fruizione possibili. Ciò significa, in

[165] Andrew Dillon, *Reading from paper versus screen: a critical review of the empirical literature*, *Ergonomics* 35(10), pp. 1297-1326 (1992), anche in http://dlist.sir.arizona.edu/1238/01/Ad1992.pdf

[166] K. O'Hara & A. Sellen, *A Comparison of Reading Paper and Online Documents*, CHI 97, pp. 335-342

pratica, utilizzare testi brevi e seguire le indicazioni che emergono dagli studi sperimentali sulla legibility riassunte qui di seguito.

Font

Sono stati condotti numerosi esperimenti per individuare i font che permettano la lettura più rapida. Tuttavia, se consideriamo i font più utilizzati, tralasciando quelli più ornati e fantasiosi, non sembrano esistere esperimenti conclusivi che favoriscano, dal punto di vista dei tempi di lettura, un font rispetto a un altro. I risultati mostrano differenze modeste, non meritevoli di particolare attenzione e, soprattutto, fortemente dipendenti anche da altri fattori (tipicamente, la dimensione dei caratteri).

La Figura 267 mostra il risultato di uno di questi esperimenti.[167]

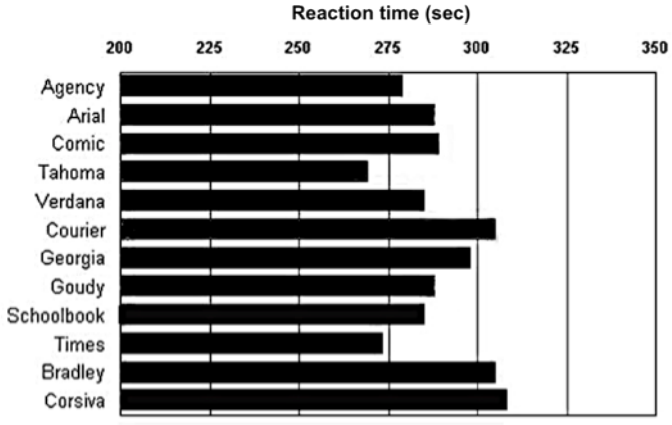


Figura 267. Tempi di lettura di un testo con diversi font (corpo 12)

Nella letteratura dell'HCI si riporta spesso la raccomandazione di preferire, per i testi sul video, font senza grazie, come Arial e Verdana a font graziati, come per esempio il Times New Roman. A questo proposito il video si comporterebbe in modo diverso dalla carta stampata, per la quale i font graziati sarebbero più leggibili.[168] In effetti, pressoché tutti i libri a stampa usano font con le grazie. La raccomandazione è spesso giustificata con il fatto che le grazie, piccoli segni anche obliqui, non sono ben riproducibili con i pixel del video, che rendono meglio forme semplici composte di linee orizzontali e verticali. Questa indicazione, pur largamente seguita nella pratica, non sembra tuttavia supportata da evidenze sperimentali conclusive, anche a causa dei metodi non sempre ottimali utilizzati negli esperimenti. Le differenze sarebbero comunque marginali, tali da non giustificare una scelta a scapito dell'altra. Ai fini pratici, sembra molto ragionevole la seguente conclusione, tratta da un'analisi recente della letteratura: "alla fine, dovremmo accettare che ogni font ben progettato e ampiamente utilizzato sia ugualmente leggibile, e che abbia più senso dibattere sull'uso di font con le grazie o senza grazie da un punto di vista estetico che da quello della leggibilità".[169]

Oggi, diversi giornali online, e non solo il New York Times, usano font graziati (per esempio, il Corriere della Sera, Le Monde, Il Sole 24 Ore, che utilizzano il font Georgia).

---

[167] M.Bernard et al., *A Comparison of Popular Online Fonts: Which is Best and When?*, *Usability News*, 4, n.1 (2001), in http://www.surl.org/usabilitynews/32/font.asp. Questo esperimento non distingue chiaramente legibility e readability: ai soggetti viene chiesto di riconoscere, in un testo, parole scambiate con altre parole di suono simile ma chiaramente fuori contesto (es. cake / fake). L'esercizio implica quindi anche processi di riconoscimento lessicale.

[168] Questa convinzione viene a volte giustificata con il fatto che le grazie, enfatizzando la "orizzontalità" dei singoli caratteri, renderebbero più fluida la scansione orizzontale del testo nella lettura.

[169] A. Poole, *Literature Review - Which Are More Legible: Serif or Sans Serif Typefaces?*, in http://www.alexpoole.info/academic/literaturereview.html (ultimo aggiornamento: 2005)

Maiuscole e minuscole

La leggibilità di un testo scritto in caratteri maiuscoli è minore di quella di un testo in caratteri minuscoli. Infatti, l'uso delle minuscole associa a ogni parola un pattern dato dalle ascendenti e dalle discendenti, che probabilmente ne facilita il riconoscimento. Questo pattern non esiste nelle parole maiuscole, come si vede dall'esempio in Figura 268. In questa figura si vede anche che la presenza di ascendenti e discendenti permette di riconoscere con facilità una frase, anche se i suoi caratteri sono visibili solo in parte. Di conseguenza, è consigliabile non comporre mai testi lunghi solo in caratteri maiuscoli. Questi possono essere utilizzati senza problemi in testi brevi, come per esempio intestazioni o titoli.



Figura 268. Leggibilità di testi in caratteri minuscoli

Corsivo, neretto e sottolineato

Il corsivo va in linea di massima evitato: sui monitor a bassa risoluzione può produrre una sgradevole scalettatura dei font (*aliasing*) per il disallineamento dei pixel lungo i tratti obliqui dei caratteri. Questo effetto può essere enfatizzato dalla presenza di grazie. Il neretto e il sottolineato possono essere utilizzati per richiamare l'attenzione su parole particolari. Come per i testi a stampa, tuttavia, è consigliabile limitarne l'uso ai casi di reale necessità, per evitare effetti visivi di eccessivo disordine. Peraltro, in accordo a una convenzione molto diffusa, molti consigliano di riservare le sottolineature ai link testuali nelle pagine web, per evitare ambiguità.

Dimensione dei caratteri

La dimensione dei caratteri è uno degli attributi che più influenzano legibility di un testo. È consigliabile usare font di dimensioni non inferiori al corpo 12. La Figura 269 riporta i risultati di un esperimento che mostra chiaramente come il tempo di lettura del testo (in ordinate, espresso in msec) peggiori significativamente nel passaggio da corpo 12 a corpo 10: l'aumento è del 30-50% circa, a seconda dell'interlinea.[170] L'esperimento mostra anche come l'effetto di un corpo piccolo possa essere parzialmente mitigato aumentando la spaziatura fra le righe. La legibility del testo in corpo 10 migliora infatti fortemente nel passaggio a interlinea 2.

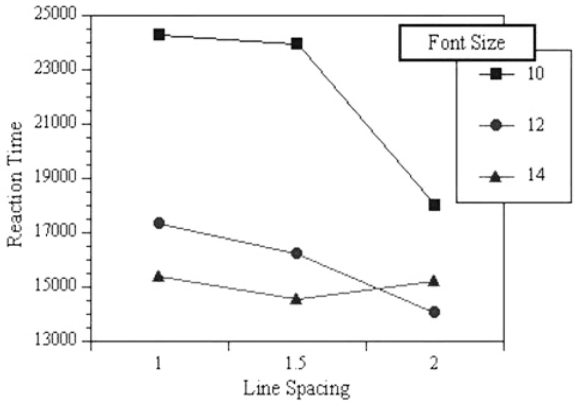


Figura 269. Tempi di lettura di un testo in funzione della dimensione dei caratteri e dell'interlinea

[170] S. Williams & L.Scharff, *The Effects of Font Size and Line Spacing on Readability of Computer Displays* , in http://www.laurenscharff.com/research/SWExp.html

Allineamenti

E' opinione corrente che l'allineamento del testo a sinistra (Figura 260a) migliori la legibility, poichè fornisce una ancora visiva per i movimenti di "ritorno a capo" dello sguardo. Tale opinione, tuttavia, non sembra supportata da conferme sperimentali.[171] Sembra però difficile negare che l'allineamento a bandiera destra risulti confuso e quindi poco gradevole (Figura 260c). L'allineamento a pacchetto andrebbe evitato per colonne di testo strette, per evitare gli sgradevoli effetti prodotti dall'inserimento di spazi bianchi per l'allineamento a destra, come chiaramente visibile in Figura 260b.

Tinta

Anche a proposito della tinta dei caratteri e del colore di fondo, circolano numerose opinioni. Ben poche, però, sono supportate da risultati scientificamente verificati. Alcuni studi mostrano che la tinta non influisce significativamente sulla leggibilità, la quale invece è influenzata da luminosità e contrasto con lo sfondo; altri studi mostrano risultati diversi.

Per quanto riguarda il colore dei caratteri, si può anche ricordare che, nell'occhio umano, immagini di colore diverso sono focalizzate su piani diversi.[172] Pertanto, è opportuno non mescolare in un testo caratteri di colori diversi. Se così facessimo, i caratteri di diversi colori – soprattutto quelli di colori lontani nello spettro come per esempio il blu e il rosso – ci apparirebbero su piani diversi, causando serie difficoltà di lettura.

Infine, è opportuno ricordare che la percentuale di persone con problemi nella visione del colore (cfr. pag.101 e pag.274) è significativa. I disturbi più frequenti, presenti nel 5% circa delle persone, consistono nella difficoltà nel distinguere il verde dal rosso. Pertanto, è indispensabile non associare al colore del testo informazioni che non siano veicolate anche con altri mezzi.

Polarità

Nemmeno un confronto fra la *polarità negativa* (cioè caratteri scuri su fondo chiaro) e la *polarità positiva* (caratteri chiari su sfondo scuro) sembra portare a risultati coerenti, anche se alcuni esperimenti suggeriscono che la polarità negativa sia preferibile. In effetti, essa è quella di gran lunga più utilizzata nei siti web con molto testo, per esempio nei giornali online.

Sintesi

In conclusione, le raccomandazioni che emergono dagli studi menzionati non sono molte, e vanno integrate da considerazioni basate sul buon senso. La tabella di Figura 270 le riassume.

Meglio evitare testi lunghi
Di solito si raccomanda di utilizzare (a video) font senza grazie (poche le conferme sperimentali)
Evitare il corsivo
Evitare testi lunghi in caratteri tutti maiuscoli
Usare preferibilmente caratteri di corpo non inferiore a 12; altrimenti usare interlinea doppia
Attenzione al contrasto fra colore del testo e colore dello sfondo
Preferire caratteri scuri su fondo chiaro
Non usare sfondi con *texture* che ostacolino la lettura
Non affiancare caratteri di tinte spettralmente lontane (problemi di messa a fuoco contemporanea)

[171] Un esperimento da noi condotto non mostra alcuna differenza nei tempi di lettura di testi allineati a sinistra o a destra, a bandiera.

[172] Ciò è dovuto al fatto che l'angolo di rifrazione di un fascio luminoso, nel passaggio da un mezzo all'altro dipende dalla lunghezza d'onda del fascio. Pertanto, quando i raggi luminosi attraversano il cristallino dell'occhio, vengono deviati di un angolo funzione della loro lunghezza d'onda, e quindi, in sostanza, del loro colore. Le immagini di diverso colore vengono quindi focalizzate sulla retina su piani leggermente diversi. L'effetto è tanto più sensibile quanto più le lunghezze d'onda differiscono.

Non veicolare le informazioni esclusivamente attraverso il colore (daltonismo, poca sensibilità al blu)

Figura 270. Linee guida per la legibility dei testi a video

Adrian Frutiger, grande font designer, scrisse una volta: "Durante tutta la mia carriera professionale, compresi che la bellezza, la leggibilità e, in una certa misura, la semplicità, sono dei concetti molto vicini fra loro: una buona lettera è quella che si annulla davanti al lettore per divenire puro veicolo tra lo spirito dello scritto e la mente di chi legge." Questa frase racchiude due concetti filosoficamente molto importanti per il designer di artefatti usabili. Il primo, che abbiamo già discusso a pag.124 a proposito dei livelli di maturità della progettazione, esprime il fatto che un artefatto maturo è "invisibile" durante l'uso. Così come una buona penna durante il processo di scrittura, un font ben progettato "scompare" all'interno del rapporto che s'instaura, durante la lettura, fra la mente di chi legge e la storia che il testo racconta. Il secondo concetto pone in relazione diretta bellezza e usabilità, almeno nell'ambito della tipografia. Afferma che i font che preferiamo sono quelli che riusciamo a leggere più facilmente. Diversi ricercatori hanno cercato di convalidare questo concetto, che ci pare intuitivamente accettabile, senza tuttavia ottenere risultati definitivi. Esiste, probabilmente, una certa correlazione fra usabilità *percepita* e gradimento estetico (ci piace ciò che ci sembra facile da usare), ma non fra gradimento e reale usabilità. In effetti, usabilità percepita e reale non sono necessariamente correlate: ciò che ci appare facile da usare non sempre lo è davvero.

Readability

Quando dallo studio della legibility di un testo si passa ad analizzarne la readability, le cose si complicano notevolmente. L'essenza della lettura non sta nel riconoscimento visivo dei caratteri che costituiscono il testo, ma nella comprensione dei suoi contenuti. Un testo è tanto più *readable* quanto più rapidamente e senza sforzo siamo in grado di comprenderne "a fondo" i contenuti. Questo richiede, da parte della nostra mente, elaborazioni complesse. Per comprendere una frase:

Nel mezzo del cammin di nostra vita
mi ritrovai per una selva oscura

dobbiamo non solo riconoscere le parole che la compongono, ma anche analizzarne la struttura. Dobbiamo comprenderne tutti gli elementi e la loro interazione: chi sta facendo cosa, dove e quando. Come arriviamo a questa comprensione? È chiaro che i fattori in gioco sono numerosi, e coinvolgono sia il livello lessicale, sia quello sintattico e semantico della frase. Inoltre la comprensione dipende fortemente dalle caratteristiche del lettore: dal suo livello di dimestichezza con il lessico e i costrutti linguistici utilizzati, dalla sua cultura generale e dalle sue conoscenze dello specifico argomento trattato.

Ancora una volta, però, possiamo formulare delle ipotesi semplificatrici. Per esempio, ipotizzare che, a parità di tutte le altre condizioni, un testo sia tanto più leggibile (readable) quanto più sia costituito da parole brevi e da frasi costituite da un numero limitato di parole. È chiaramente una semplificazione drastica del problema, che però permette di definire degli *indici di leggibilità* (*readability index*) attraverso delle semplici formule matematiche. Per calcolare l'indice di un testo, basterà quindi contarne le frasi, le parole e i caratteri (o le sillabe) che lo compongono, e applicare la formula (meglio se con l'aiuto di un computer).

Negli Stati Uniti, dagli anni '20 del secolo scorso, sono stati proposti vari indici di leggibilità per la lingua inglese. L'obiettivo principale era valutare la readability dei libri di testo scolastici. Le formule che esprimono questi indici devono, ovviamente, essere tarate sulla realtà, attraverso messe a punto sperimentali che mettano in correlazione i valori degli indici con la scolarità del lettore. Per quanto riguarda la lingua italiana, lo strumento più noto per la valutazione

della readability di un testo è l'indice Gulpease.[173] A differenza di altri, esso considera la lunghezza delle parole e delle frasi in caratteri invece che in sillabe, ed è perciò molto facile da calcolare. Il suo valore è un numero compreso fra 0 (leggibilità minima) e 100 (leggibilità massima) e si calcola mediante la seguente formula:

$$89 + \frac{300 * (numero\ delle\ frasi) - 10 * (numero\ delle\ lettere)}{numero\ delle\ parole}$$

Le costanti della formula sono state scelte in modo che:

- se il valore è minore di 80 il testo è difficile da leggere per chi ha la licenza elementare;
- se il valore è minore di 60 il testo è difficile da leggere per chi ha la licenza media;
- se il valore è minore di 40 il testo è difficile da leggere per chi ha un diploma superiore.

L'attendibilità di un indice di leggibilità non deve però essere sopravvalutata. Il semplice conteggio di caratteri, parole e frasi di un testo ci può fornire un indizio sulla sua complessità lessicale e sintattica, ma non ci può dire nulla sul suo significato, e quindi sulla sua reale comprensibilità. Frasi come "Penso, dunque sono" o "M'illumino d'immenso" sono fatte di poche parole, ma la loro comprensione non è per niente banale. Gli indici di leggibilità operano, per così dire, solo sulla superficie del testo, e non ci guardano dentro: se prendiamo un testo con un certo indice e ne mescoliamo a caso le parole, il testo conserva lo stesso indice, pur diventando ovviamente incomprensibile.

Pur con queste limitazioni, l'uso di un indice di leggibilità può fornire indicazioni utili per il miglioramento di un testo. Un indice Gulpease troppo basso ci suggerisce che, probabilmente, stiamo utilizzando frasi troppo lunghe e sintatticamente complesse. Poiché la stesura di un testo comporta di solito diverse revisioni, possiamo utilizzare questo strumento per misurare il miglioramento – in termini di semplificazione – fra una revisione e la successiva. Oppure per confrontare testi differenti (Figura 271).

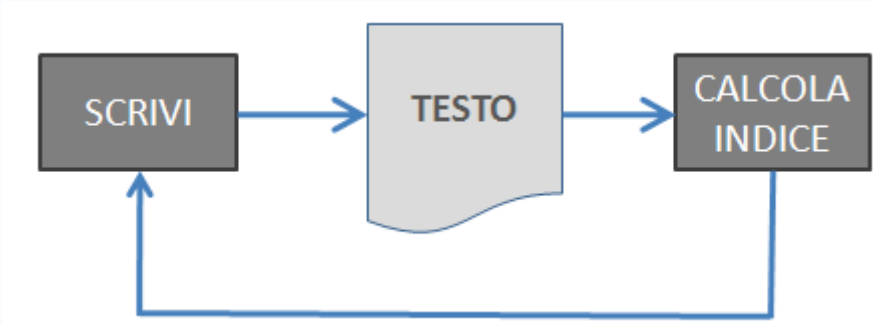


Figura 271. Uso dell'indice di leggibilità

Oltre alla complessità sintattica, possiamo considerare il *vocabolario* (o, come si dice, il *lessico*) utilizzato nella scrittura. Due testi che hanno lo stesso indice possono avere readability molto diversa, in funzione del vocabolario usato. È probabile che un testo fatto solo di parole di uso frequente e generalizzato sia più facilmente comprensibile di uno con parole insolite, tecniche o gergali, anche se con lo stesso indice di leggibilità. Si può allora studiare il vocabolario della lingua italiana e suddividerlo in insiemi di vocaboli noti a fasce via via più ampie di popolazione. Si può così costruire una rappresentazione a centri concentrici del lessico della lingua (Figura 272), che descriviamo brevemente nel seguito.

[173] Questo indice è stato sviluppato nel 1988 dal Gruppo Universitario Linguistico Pedagogico dell'Università di Roma La Sapienza (GULP), sulla base di ricerche di M. Corda Costa e T. De Mauro

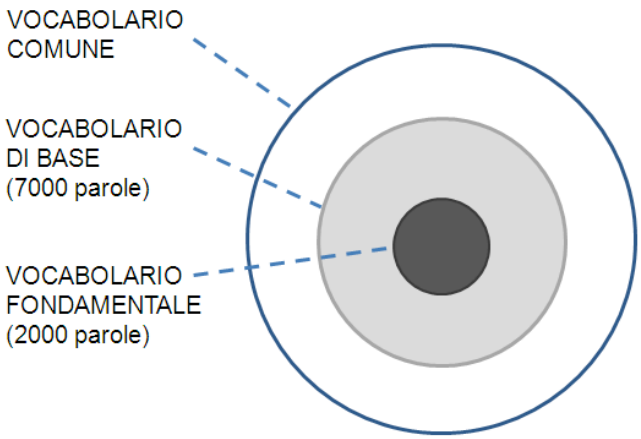


Figura 272. Struttura del lessico della lingua italiana

Se si contano tutte le parole utilizzate nella storia della lingua italiana, anche quelle presenti una sola volta in qualche testo, si arriva a un numero molto elevato (probabilmente molto oltre il milione). Se però si eliminano dal conteggio le parole in disuso, i termini utilizzati solo in certe aree geografiche e i termini tecnici utilizzati in specifici settori, il numero si riduce sensibilmente, e si ottiene il cosiddetto *vocabolario comune* della lingua. Si tratta di quell'insieme di vocaboli registrato nei dizionari generici (non specialistici) della lingua. Il numero delle parole[174] del vocabolario comune dipende, ovviamente, dai criteri di selezione usati dai redattori. Per esempio, l'edizione 2004 del Vocabolario Zingarelli della lingua italiana conta 134.000 voci, mentre vocabolari più ridotti ne contano circa 60.000 o meno.

Come indicato nella Figura 272, all'interno del vocabolario comune si possono definire altri due insiemi più piccoli. Il primo è il cosiddetto *vocabolario di base*. Si tratta di quei termini del vocabolario comune che sono largamente noti ai membri delle più svariate categorie di persone. Il linguista Tullio De Mauro l'ha definito come l'insieme di parole certamente note a chi ha frequentato la scuola di base, cioè possiede una licenza di scuola media. È un insieme composto di circa 7000 vocaboli, di cui De Mauro e i suoi collaboratori hanno costruito l'elenco. Ne fanno parte, per esempio, le parole: abbronzare, abusare, accampamento, acconto, adagiare, agonia, alunno, anticipare, arrugginire, attribuire.[175] Alla data del censimento del 2001, gli italiani con la licenza media, e quindi teoricamente in grado di comprendere il vocabolario di base, erano i due terzi circa di tutte le persone di almeno 11 anni di età.

C'è infine il nucleo più interno della sfera lessicale di una lingua. È il *vocabolario fondamentale*. Sono i vocaboli che chi parla una lingua ed è uscito dall'infanzia conosce, capisce e usa. Sono le parole di massima frequenza nel parlare e nello scrivere, disponibili a chiunque in ogni momento, sempre che conosca l'italiano. Più precisamente, sono le parole note alla generalità degli italiani che abbiano frequentato le scuole elementari, e costituiscono, all'interno del vocabolario di base, un nucleo di circa 2000 parole. Per esempio, ne fanno parte le parole: abbastanza, abitudine, accogliere, acqua, addio, affrontare, amare, assai, atmosfera, avvenimento. Nel censimento del 2001, gli italiani in possesso almeno della licenza elementare, e quindi "teoricamente" in grado di comprendere con certezza il vocabolario di base, erano il 93% delle persone di almeno 11 anni. Abbiamo detto "teoricamente", perché sappiamo che la preparazione scolastica non è affatto uniforme. Secondo un'indagine Censis del 2000, il 34% della popolazione italiana possiede "una competenza alfabetica molto modesta, al limite dell'analfabetismo", quanto alle capacità e abilità necessarie per leggere testi in prosa quali: articoli di giornale, annunci, lettere, racconti, ecc. Questa "competenza

[174] Più propriamente, dovremmo dire *lessemi*. Questo termine designa l'unità lessicale astratta, portatrice di significato. Ogni lessema può avere diverse *forme*, che chiamiamo genericamente *parole*. Così, *il* e *lo* sono due forme dello stesso lessema, ma due parole distinte.

[175] Le parole del vocabolario fondamentale e del vocabolario di base della lingua italiana sono elencate nell'appendice del libro di Tullio De Mauro, *Guida all'uso delle parole*, Editori Riuniti, pubblicato nel 1980 e da allora più volte ristampato. Il vocabolario di base non è un corpus statico, ma evolve nel tempo. Per esempio, nell'elenco di De Mauro non compare la parola *cellulare*, diventata da allora di uso diffuso.

alfabetica al limite dell'analfabetismo" si ritrova in percentuali significative anche fra chi ha fatto studi superiori, e non solo fra le persone con livello di scolarità molto basso.

La disponibilità dei vocabolari fondamentale e di base è molto importante ai fini pratici. Infatti, attraverso di essi è possibile costruire strumenti informatici che ci aiutano a semplificare un testo, segnalando i vocaboli "difficili" e proponendo dei sinonimi con un più alto grado di diffusione. In rete esistono siti che offrono un servizio di valutazione dell'indice e di classificazione delle parole utilizzate nel testo. Per esempio, la Figura 273 mostra l'analisi effettuata dal servizio www.eulogos.net sulle prime righe della Introduzione di questo libro. Il valore dell'indice Gulpease è 44. La tabella mostra l'indice di ogni frase e il suo livello di comprensibilità in relazione al livello scolastico del lettore. Riporta anche la classificazione delle parole, evidenziando in particolare quelle che non appartengono al vocabolario di base (VdB), per esempio: ingegneria, usabilità, interattivi, informatica, software, inquadramento.

Questi strumenti possono essere incorporati anche nei word processor.

Legenda per le frasi
nelle quali ogni parola è confrontata con il VdB

Grassetto: vocabolario fondamentale
Tondo: vocabolario di alto uso
Corsivo: vocabolario di alta disponibilità
Corpo e carattere diversi: non presente nel VdB

Legenda per la difficoltà
correlata al livello di scolarizzazione del lettore

---- quasi incomprensibile
+--- molto difficile
++-- difficile
+++- facile
++++ molto facile

Frase	G	Difficoltà/livello scol. Elem.	Media	Sup.
Questo libro fornisce un'*introduzione* generale all'**ingegneria** dell'**usabilità**, per la **progettazione** di prodotti facili da usare, in particolare sistemi **interattivi** ad alto contenuto di **software**.	39	----	+---	++--
È nato come libro di testo per corsi universitari di carattere **introduttivo** per le discipline dell'**interazione** uomo macchina o dell'**interaction design**, oggi presenti in quasi tutti i corsi di laurea in **Informatica**, ma può essere letto anche da chi non si occupi **specificamente** di **informatica**.	44	----	+---	+++-
Infatti, a parte una generica comprensione dei concetti generali dell'**informatica** e del **software**, il libro non **presuppone** nel *lettore* particolari **nozioni** di carattere tecnico.	42	----	+---	+++-
Può quindi essere usato senza difficoltà anche da studenti di orientamento diverso, per esempio dei corsi di laurea in Scienze della Comunicazione o Disegno Industriale, o da chiunque desideri avere un **inquadramento introduttivo** a questi argomenti.	39	----	+---	++--
La disciplina dell'**interazione** uomo macchina ha più di un quarto di secolo, ed è oggi **v**	65	+---	+++-	+++-

Figura 273. Analisi della leggibilità delle prime righe della Introduzione di questo libro (www.eulogos.net)

Un esempio di scrittura ad alta leggibilità è il mensile *Due Parole* (http://www.dueparole.it), prodotto in rete per iniziativa di Tullio De Mauro (fino al maggio 2006). I redattori di *Due Parole* scrivono utilizzando in modo consapevole e sistematico criteri di *scrittura controllata*. I criteri principali della scrittura controllata sono: la brevità dei testi, la semplicità delle frasi, l'uso del vocabolario di base (le parole che non vi appartengono vengono spiegate). Molto curata è anche l'organizzazione logico-concettuale dei testi. Si rivolge alle persone che hanno bisogno di testi informativi molto leggibili e comprensibili: studenti stranieri che seguono corsi in Italia, extra-comunitari con poca dimestichezza della lingua italiana, ragazzi italiani della scuola dell'obbligo che hanno difficoltà di comprensione della lingua italiana, soprattutto scritta.

I manuali di stile

Esistono in commercio molti *manuali di stile*, che danno utili indicazioni per la redazione dei testi. Oltre a ricordarci le numerose regole sull'uso corretto dei vari elementi che compaiono in un testo, questi manuali forniscono spesso delle linee guida per una scrittura di facile comprensione. Si tratta di regole stilistiche, che consolidano la prassi del buon uso

della lingua italiana, senza, ovviamente, convalide sperimentali. Sono le regole seguite dai redattori che correggono i testi dei libri prima di passarli in stampa. Per esempio, uno dei più noti manuali di stile per la lingua italiana,[176] propone dieci semplici linee guida, che riportiamo integralmente qui di seguito.

1. Usare parole precise

   Nei casi concreti e specifici, conviene usare termini concreti e specifici anziché termini astratti e generici. I termini astratti sono invece utili per dare generalità al discorso.

   Esempi: documento → relazione (se tale); divisione → parete (se tale); togliere → svitare (se ruotando); modificare → correggere (se sbagliato).

2. Usare parole semplici

   Senza andare a scapito della precisione, conviene usare parole semplici e correnti anziché parole ricercate e difficili. Molti concetti complessi possono essere espressi con parole semplici, senza con questo compromettere il senso dell'informazione.

   Esempi: appellativo → nome; remunerazione → compenso; conferire → dare; delucidare → chiarire.

3. Usare espressioni semplici

   Nel rispetto dello stile generale del discorso, conviene in genere usare espressioni semplici e immediate: in molti casi una sola parola può sostituire un'intera circonlocuzione.

   Esempi: allo scopo di → per; nel momento in cui → quando; in base al fatto che → poiché; fare uso di → usare.

4. Omettere le parole inutili

   Nella costruzione delle frasi, conviene omettere le parole e le espressioni che possono essere eliminate senza modificare o impoverire il contenuto della frase.

   Esempi: se è vero che → se; questo è un argomento che → questo argomento; il fenomeno, considerato nella sua natura → il fenomeno.

5. Omettere le precisazioni superflue

   Generalmente, conviene omettere le precisazioni strettamente superflue, che non aggiungono nulla al senso del discorso.

   Esempi: il successo finale del corso → il successo del corso; eliminare del tutto → eliminare; assolutamente indispensabile → indispensabile; unire insieme con → unire con.

6. Costruire periodi semplici

   In genere, conviene comporre periodi brevi e semplici, più facili da leggere e interpretare, e spesso più efficaci. I periodi complessi possono essere scomposti in sequenze di periodi semplici, logicamente correlati fra loro.

   Esempi: Per la sua complessità, la procedura è suddivisa in passi distinti, ciascuno dei quali corrisponde a una sequenza elementare di operazioni e fornisce un risultato autonomo → Per la sua complessità, la procedura è suddivisa in passi distinti. Ogni passo corrisponde a una sequenza elementare di operazioni e fornisce un risultato autonomo.

7. Tenere vicini i termini collegati

   Nella costruzione delle frasi, conviene tenere vicini fra loro i termini fra i quali esiste uno stretto collegamento logico: la vicinanza fisica aiuta a cogliere il collegamento.

   Esempi: Rimandiamo a domani la decisione, quando avremo dati più precisi → rimandiamo la decisione a domani, quando avremo dati più precisi; il testo viene composto, dopo i vari passi di revisione, nella sua forma finale → dopo i vari passi di revisione, il testo viene composto nella sua forma finale.

8. Esprimere le idee analoghe in forma analoga

   Se si deve esprimere una serie di concetti analoghi, conviene usare una forma di espressione analoga per i singoli concetti: l'analogia della forma evidenzia quella della sostanza.

   Esempi: La qualità si ottiene progettando con attenzione e con una realizzazione accurata → La qualità si ottiene progettando con attenzione e realizzando con cura; il piano di profondità controlla il beccheggio. Il rollio è controllato dagli alettoni. Con il timone si controlla l'imbardata → Il piano di profondità controlla il beccheggio. Gli alettoni controllano il rollio. Il timone controlla l'imbardata.

[176] R.Lesina, *Il nuovo manuale di stile (edizione 2.0) – Guida alla redazione di documenti, relazioni, articoli, manuali, tesi di laurea*, ed. Zanichelli, 1994.

9. Preferire la costruzione positiva a quella negativa

   Quando una frase può essere costruita in forma positiva, conviene in genere scriverla in tale forma, poiché risulta più chiara e diretta.

   Esempi: non credo che accetterò l'incarico → credo che rifiuterò l'incarico; quel treno non arriva mai in ritardo → quel treno arriva sempre in orario.

10. Usare la forma passiva in modo ponderato

    In linea di massima, nella costruzione del discorso conviene usare di preferenza la forma attiva, che generalmente risulta più chiara e diretta di quella passiva.

    Esempi: La comprensione è facilitata dalla semplicità di linguaggio → la semplicità del linguaggio facilita la comprensione; questo atteggiamento può essere interpretato dal pubblico come un segno di disinteresse → il pubblico può interpretare questo atteggiamento come un segno di disinteresse.

Queste regole di semplificazione dovrebbero essere adottate in modo particolarmente rigoroso per i testi che hanno lo scopo di comunicare informazioni o istruzioni operative. Un'area particolarmente importante è quella dei testi della Pubblica Amministrazione, per i quali da tempo è in atto un progetto di semplificazione del linguaggio. Nella *Direttiva sulla semplificazione del linguaggio dei testi amministrativi* emessa dal Ministro per la Funzione Pubblica nel maggio 2002, si dice:

> *I numerosi atti prodotti dalle pubbliche amministrazioni, sia interni (circolari, ordini di servizio, bilanci) sia esterni, devono prevedere l'utilizzo di un linguaggio comprensibile, evitando espressioni burocratiche e termini tecnici. Anche gli atti amministrativi in senso stretto, che producono effetti giuridici diretti e immediati per i destinatari, devono essere progettati e scritti pensando a chi li legge. Oltre ad avere valore giuridico, però, gli atti amministrativi hanno un valore di comunicazione e come tali devono essere pensati. Devono, perciò, essere sia legittimi ed efficaci dal punto di vista giuridico, sia comprensibili, cioè di fatto efficaci, dal punto di vista comunicativo.*[177]

In Figura 274 sono riportate le regole di scrittura del testo indicate nella direttiva sopra citata. È interessante osservare che, al punto 2, la direttiva suggerisce di utilizzare preferenzialmente il vocabolario di base di Tullio De Mauro e, al punto 10, suggerisce di usare quegli elementi para-testuali che possano facilitare la lettura, quali neretti, sottolineati, corsivi, grandezza del corpo, elenchi, ecc.

1. Scrivere frasi brevi
2. Usare parole del linguaggio comune
3. Usare pochi termini tecnici e spiegarli
4. Usare poco abbreviazioni e sigle
5. Usare verbi nella forma attiva e affermativa
6. Legare le parole e le frasi in modo breve e chiaro
7. Usare in maniera coerente le maiuscole, le minuscole e la punteggiatura
8. Evitare neologismi, parole straniere e latinismi
9. Uso del congiuntivo
10. Usare in maniera corretta le possibilità di composizione grafica del testo

Figura 274. Le regole del linguaggio amministrativo

[177] http://www.funzionepubblica.it/chiaro/direttiva.pdf. Vedi anche http://www.funzionepubblica.it/chiaro/allegato.pdf e http://www.entilocali.provincia.le.it/nuovo/files/Progetto%20di%20semplificazione%20del%20linguaggio.pdf .

I molti esempi contenuti nella direttiva e nel materiale informativo aggiuntivo chiariscono gli intendimenti del Ministero. Ne riportiamo solo due, a illustrazione delle regole 1 e 3 rispettivamente:

> *Testo originale: Qualora dal controllo dovesse emergere la non veridicità del contenuto della dichiarazione, il dichiarante decade dai benefici conseguiti sulla base della dichiarazione non veritiera, fermo restando quanto previsto dall'art.26 della legge 4 gennaio 1968, n.15, in materia di sanzioni penali.*
>
> *Testo riscritto: Chi rilascia dichiarazione falsa, anche in parte, perde i benefici descritti e subisce sanzioni penali.* (* Articolo 26, legge n.15 del 4.1.1968)*
>
> *Testo originale: Tali posizioni sono da identificare non tanto in diritti irrefragabili, il cui esercizio prescinde dall'adozione di atti permissivi dell'Amministrazione, ma in situazioni giuridiche suscettibili di trasformazione a seguito di atti di tipo suindicato.*
>
> *Testo riscritto: I cittadini che vogliono iniziare un'attività devono chiedere un'autorizzazione alle amministrazioni competenti.*

Si tratta di un linguaggio semplificato, privo di complessità non necessarie, che trasmette le informazioni al lettore nel modo più chiaro ed efficace possibile, che viene denominato spesso *plain language*. Il plain language è la lingua ordinaria, che si sforza di assomigliare a quella usata nella conversazione quotidiana; è privo di espressioni gergali, dotte, desuete o rare, e coadiuvato da un'impostazione grafica che agevola la lettura. Idealmente, il lettore dovrebbe riuscire a capire il testo alla prima lettura. È la lingua che dovrebbe essere utilizzata, in particolare, nei testi che sono parte di sistemi interattivi usabili.

Il testo nel Web

I testi per il Web hanno caratteristiche peculiari. Scrivere per il Web richiede uno stile editoriale che si adatti bene alle modalità di lettura tipiche di questo medium. Infatti, come dimostrano numerosi esperimenti, l'utente del Web non legge le pagine, ma le "scorre", un po' come se cercasse informazioni su una carta geografica. A ogni pagina presta attenzione per un tempo limitato: spesso, solo per pochi secondi. Se non trova subito l'informazione cercata, è molto probabile che rinunci e passi a un'altra pagina. Anche nei testi che gli interessano, l'utente cerca di arrivare subito al punto, sorvolando sulle frasi meno importanti.

Come si è visto a pag.274, il senso di lettura su una pagina web non è necessariamente da sinistra a destra e dall'alto in basso, ma "saltellante", senza un orientamento costante: lo dimostrano le analisi effettuate con strumenti di eye tracking. Ogni tanto clicchiamo su un link, esaminiamo il testo che ci viene presentato, poi clicchiamo ancora, spesso per tornare indietro. Secondo alcuni, non è nemmeno corretto parlare di lettura: si tratta infatti di un processo molto diverso da quello che avviene leggendo un libro; assomiglia, piuttosto, a ciò che avviene quando scorriamo le pagine di un quotidiano: lo esaminiamo per cercare le notizie che ci interessano scorrendo titoli e occhielli, spesso in modo non sistematico e non sequenziale. Quando un articolo richiama la nostra attenzione, raramente lo leggiamo per intero, da capo a fondo. Cerchiamo, invece, di estrarne il senso nel minor tempo possibile. Sul Web ci comportiamo nello stesso modo, ma abbiamo più gradi di libertà: qui un testo si espande anche "in profondità" e non solo in lunghezza e larghezza. Se una frase è un link, con un clic possiamo richiamare subito un'altra pagina e così via, di clic in clic. Queste possibilità, unite all'enorme quantità di informazioni disponibili, ci inducono a scorrere i contenuti ancora più in fretta, "saltellando" di pagina in pagina e di sito in sito. Si ricordi, a questo proposito, quanto detto a pag. 87 sulla dispersione dell'attenzione nell'utilizzo del Web.

Il testo deve allora essere organizzato in modo da non creare ostacoli a chi lo esamina in questo modo. Jakob Nielsen ha introdotto il termine *scannable text*, che possiamo tradurre con *testo da scorrere*, per indicare un testo che si può facilmente esaminare in modo rapido e scorrevole.[178] Per questo, dobbiamo operare non soltanto sul testo vero e

[178] Vedi per esempio: J.Morkes, J.Nielsen, *Concise, Scannable, and Objective:How to Write for the Web*, 1997 (http://www.useit.com/papers/webwriting/writing.html). Il sito di Jakob Nielsen contiene molto materiale sull'argomento: http://www.useit.com/papers/webwriting.

proprio, ma anche sugli elementi para-testuali (titoli, sottotitoli, link ipertestuali, ecc.). Per quanto riguarda il testo, dovremo seguire i principi già visti del plain language, esprimendo i concetti in modo conciso e strutturando il testo in paragrafi brevi, ognuno dei quali esprime un singolo concetto. Dovremo arrivare rapidamente al punto, esponendo i fatti in modo diretto, senza figure retoriche come metafore, similitudini o altro. Per quanto riguarda il para-testo, alcune indicazioni sono le seguenti:

- strutturare il testo in pagine brevi, che preferibilmente non superino le dimensioni di una schermata, per ridurre la necessità di scrolling;
- fare ampio uso di titoli e sottotitoli brevi e densi d'informazione (e non d'effetto, come nei giornali);
- mettere in evidenza le parole chiave e i concetti importanti con opportuni artifici tipografici (neretto, colore, collegamenti ipertestuali che rimandano ad approfondimenti, ecc.);
- usare le sottolineature esclusivamente per evidenziare i collegamenti ipertestuali, per non creare ambiguità;
- organizzare i contenuti per livelli successivi di dettaglio, utilizzando estesamente le possibilità associative di un ipertesto: titoli brevi, paragrafi brevi, rimandi a ulteriori approfondimenti con collegamenti ipertestuali;
- inserire i rimandi ipertestuali in modo naturale nel testo. Per esempio, non si scrive: "Per vedere le nostre offerte speciali, cliccate qui" ma, semplicemente: "Le nostre offerte speciali".

Come già osservato, a parte i collegamenti ipertestuali, l'analogia più prossima è quella con la pagina di un quotidiano, con i suoi titoli, occhielli, sommari, fotografie, didascalie, riquadri di approfondimento. È il cosiddetto stile a *piramide invertita*, come si dice nel gergo dei giornalisti. Invertita perché s'incomincia dal fondo, in altre parole dalle conclusioni, dalla sintesi, per proseguire con le spiegazioni e con i dettagli. E comunque usando sempre frasi brevi, semplici, dirette, evitando stereotipi, figure retoriche e ridondanze (Figura 275).

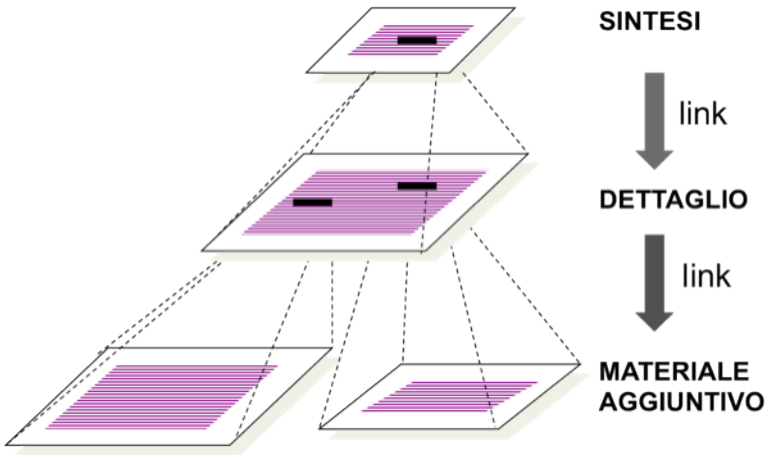


Figura 275. Stile a piramide invertita

Jakob Nielsen ha coniato il termine "micro-content" per indicare quei contenuti espressi in frasi brevi, poche diecine di caratteri al massimo, e che tuttavia contengono tutte le informazioni essenziali, senza alcuna ridondanza. Una buona pagina web è ricca di micro-contenuti. La tendenza, nella comunicazione digitale, alla frammentazione dell'informazione in "micro-contenuti" è evidente: basti pensare all'enorme diffusione degli sms e, più recentemente, al grande successo di Twitter (www.twitter.com), non solo come social network, ma soprattutto come veicolo di informazione sintetica e in tempo reale.

Come è stato notato,[179] le sei parole chiave per la letteratura del terzo millennio di cui parla Italo Calvino nelle sue *Lezioni Americane* possono anche essere considerate le sei parole chiave della comunicazione sul Web: leggerezza, rapidità, esattezza, visibilità, molteplicità, consistenza. Testi che sulla carta sarebbero considerati assolutamente stringati, letti sul Web appaiono antiquati e ridondanti. Bisogna sempre evitare le introduzioni ampie e verbose: il navigatore del Web ha fretta, e non le leggerà.

Oltre allo stile "leggero", ciò che differenzia un testo tradizionale da un testo per il Web è la presenza di link ipertestuali. Un ipertesto cambia profondamente le regole della scrittura, che si sviluppa anche in profondità, e non più soltanto in lunghezza. Questa nuova dimensione del testo introduce interessanti possibilità, ma anche notevoli rischi. Un link è un po' come una nota a piè di pagina: aggiunge nuove informazioni, ma distrae dal discorso principale. Le note a piè di pagina, però, sono quasi sempre brevi, e non portano da altre parti: l'occhio torna al testo principale quasi subito, e in genere non si perde il filo del discorso. Un link ipertestuale, invece, non sempre funziona così: può condurre a un testo che contiene altri link, che a loro volta conducono ad altri testi, e così via. Una lettura ramificata, nella quale il lettore si può smarrire facilmente (Figura 276).

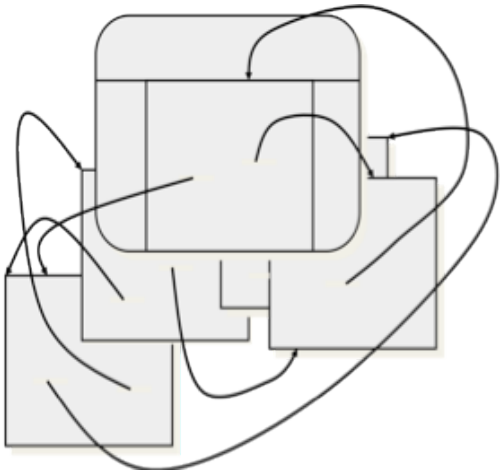


Figura 276. I link ipertestuali possono creare strutture complesse

I primi scrittori di ipertesti (negli anni 80, e quindi prima del Web), entusiasti per questo nuovo modo di scrivere (e di leggere), utilizzavano spesso i link in modo eccessivo. L'ipertesto era visto come un mezzo per mettere in piena evidenza la struttura di relazioni del sapere, una sorta di liberazione dai vincoli della scrittura sequenziale tradizionale. "Everything is interconnected", dicevano i profeti di questo nuovo tipo di letteratura. Rilette oggi, queste prime creazioni ci appaiono spesso francamente illeggibili: per esempio, non era raro imbattersi in collegamenti circolari, che mandavano il lettore in loop. In un sito web, per evitare queste difficoltà, converrà fare un uso abbastanza sobrio dei link interni al testo, per evitare di creare una struttura di navigazione aggiuntiva, sovrapposta a quella del sito, col rischio di far perdere l'orientamento al navigatore. Nei casi più comuni, conviene limitarsi ad usare i link per richiamare i dettagli nella piramide invertita, come in Figura 275.

Esistono in commercio numerosi ottimi manuali di stile specificamente orientati al Web.[180]

L'uso creativo del testo

L'uomo può comunicare in molti modi, e anche il testo può avere usi molto differenti. Si può giocare con la forma grafica del testo, usarla in modo creativo per suscitare sorpresa, divertimento, ammirazione, ripugnanza, piacere. Da questo punto di vista, il Web ha da sempre stimolato fortemente lo sviluppo della creatività dei progettisti. Gli esempi sono infiniti. La Figura 277 riporta una delle prime home page di Yahoo! (del 1995), in cui la scelta dei caratteri e dei

[179] L.Carrada, *Scrivere per Internet*, Ed.Lupetti, 2000.
[180] Per un sintetico prontuario adatto per il Web, si veda per esempio il libro di M. Grasso, *Scrivere per il Web*, Franco Angeli Editore, 2002. Il testo probabilmente più completo per quanto riguarda queste indicazioni, anche se non specificamente orientato al Web, resta comunque il *Manuale di Stile* già citato in nota a pagina 296.

colori del logo vuole sottolineare graficamente il significato della parola “yahoo” che, nella lingua inglese, significa “rude, selvaggio, sgraziato”.

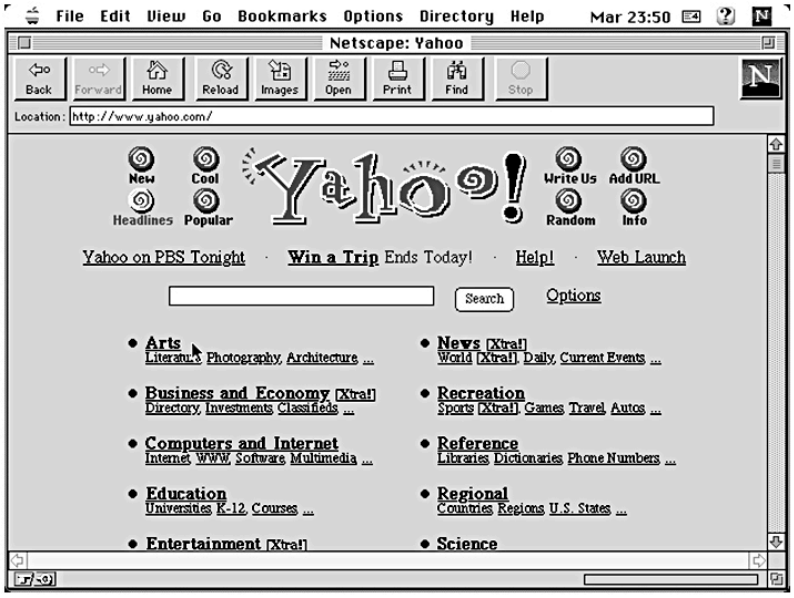


Figura 277. Una delle prime home page di Yahoo (1995)

La Figura 278 mostra un altro esempio di uso creativo del testo: la home page di un’agenzia di design, che intende promuovere la propria creatività grafica. Al centro, un elaborato menu composto da caratteri di foggia e orientamento diversi. Dal punto di vista della legibility, esso lascia molto a desiderare. I caratteri sono grandi, ma hanno orientamenti diversi, che costringono il lettore a ruotare la testa verso sinistra e verso destra. Il loro colore, rosso scuro, li rende poco distinguibili dallo sfondo nero.[181] Come se non bastasse, la parola centrale (il nome della società che possiede il sito) è spezzata su 4 righe: per ricomporla il lettore deve compiere uno sforzo abbastanza significativo. Tuttavia, con ogni evidenza, l’effetto complessivo è voluto, ed è il risultato di scelte grafiche consapevoli.

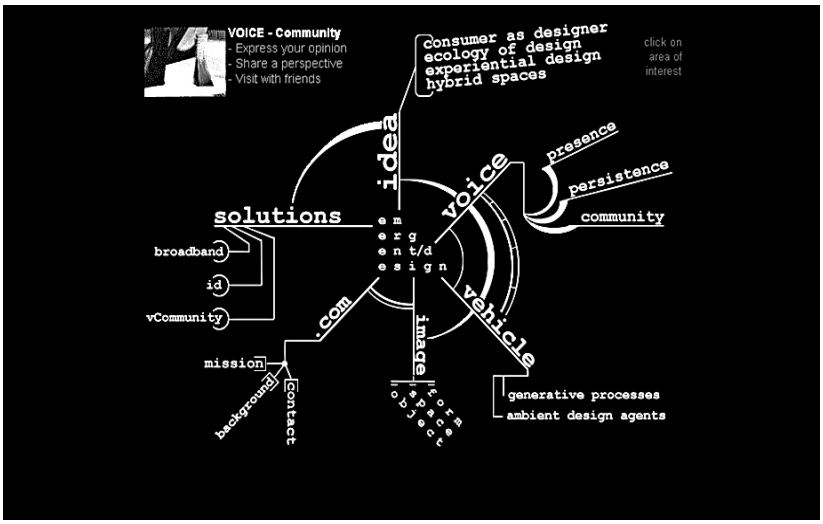


Figura 278. Home page
(www.emergent-design.com, online negli anni1999-2004)

[181] Nella riproduzione in bianco e nero il contrasto è stato artificialmente aumentato: senza questo accorgimento le scritte sarebbero risultate praticamente leggibili.

Un esempio estremo, più recente, è mostrato in Figura 279. Si tratta di una home page che mostra l'indice del contenuto del sito, in forma di cuore (di colore rosso su fondo bianco). Le differenti scritte che compongono il cuore sono, in sostanza, le voci di un menu. Il sito usa tecnologie di animazione: cliccando sopra ciascuna scritta, il cuore si anima e si disgrega, i diversi caratteri si separano e, letteralmente, volano via in direzioni diverse. La voce selezionata, però, si ricompone a formare il titolo della pagina corrispondente. Quando l'occhio, che segue quest'animazione, lascia i caratteri, la nuova pagina è apparsa sullo schermo video. La legibility è quasi nulla – ci vorrebbe moltissimo tempo a leggere in sequenza tutte le voci – ma la pagina possiede un fascino innegabile.[182]



Figura 279. La home page dell'Annual Review 2008 della British Hearth Foundation (www.bhf.org.uk/annualreview2008)

Non tratteremo oltre l'uso emozionale dei testi. Questo non significa che il progettista dei testi debba *sempre* privilegiare legibility e readability rispetto ad altre caratteristiche. Non dimentichiamo che, per la definizione adottata in questo libro, usabilità significa anche soddisfazione dell'utente. In una visione non integralista dell'usabilità, gli aspetti che contribuiscono al gradimento del prodotto da parte dei suoi utenti sono altrettanto importanti di quelli che contribuiscono all'efficacia e all'efficienza d'uso. Peraltro, diversi esperimenti hanno mostrato una certa correlazione fra piacevolezza estetica e usabilità percepita, indipendentemente dalla usabilità effettiva. A ragione, da alcuni anni, la letteratura dell'usabilità ha fortemente rivalutato gli aspetti ludici ed emozionali del design.

Ciò che importa, ancora una volta, è che il progettista definisca in modo chiaro gli obiettivi e le priorità del sistema, e li persegua in modo consapevole, evitando compromessi e deviazioni. Questo rischio è sempre presente quando non si padroneggiano bene le potenzialità espressive a disposizione, come avviene per esempio in presenza di media nuovi, per i quali non si siano ancora consolidati dei paradigmi maturi di comunicazione. Questo è accaduto nei primi anni di sviluppo del Web, in cui l'usabilità veniva sacrificata all'appeal grafico anche in siti con obiettivi puramente strumentali e nei quali, quindi, avrebbero dovuto prevalere considerazioni di leggibilità e di comprensibilità. D'altra parte, questo era accaduto anche nei primi anni dello sviluppo del libro, un mezzo di comunicazione che oggi ci sembra banale, ma che ha richiesto secoli di esperienze (e di errori) per consolidare la struttura che oggi diamo per scontata. Per esempio, consideriamo la pagina del codice miniato del quindicesimo secolo rappresentata in Figura 280. L'oggetto, certamente apprezzabile dal punto di vista estetico, per quanto riguarda l'usabilità è tutto sbagliato. I caratteri sono poco distinguibili uno dall'altro, le righe di testo sono troppo brevi in rapporto alla dimensione dei caratteri, gli elementi decorativi (le lettere iniziali di ogni frase e i decori di fine frase) creano un "rumore" eccessivo che degrada fortemente la legibility, la quantità di testo per pagina è troppo bassa, le pagine non sono numerate.

[182] In effetti, il sito è stato incluso fra i finalisti del Webby Award 2009, che premia i migliori siti web, nella categoria "Best Use of Typography" (www.webbyawards.com).



Figura 280. Libro delle Ore (Francia, circa 1440)

Ripasso ed esercizi

- Qual è la differenza fra legibility e readability?
- Esamina la home page di un quoidiano online, e identifica tutti gli elementi paratestuali ivi contenuti.
- Che cosa significa esattamente che un carattere è in corpo 12? Come si definisce l'interlinea?
- Qual è la differenza fra print font e screen font? Quali sono gli screen font più usati?
- Qual è il modello più comunemente accettato del processo di lettura?
- Riassumi li indicazioni tipografiche da seguire per ottenere una buona legibility dei testi sul video.
- Che cosa sono l'indice Gulpease e il vocabolario di base della lingua italiana?
- Analizza, alla luce delle dieci linee guida citate a pag. 295, il testo di un articolo del giornale di oggi, e semplificalo, se necessario, di conseguenza.
- Che cosa si intende per scannable text? Riassumi le indicazioni che daresti a un redattore per la scrittura di testi sul Web.

Approfondimenti e ricerche

- Per una sintetica rassegna sui modelli dei processi di lettura, puoi leggere l'articolo di K.Larson, *The Science of Word Recognition, or how I learned to stop worrying and love the bouma*, in http://www.microsoft.com/typography/ctfonts/wordrecognition.aspx.
- Analizza e confronta le scelte tipografiche dei principali quotidiani online, dal punto di vista della leggibilità, facendo riferimento alle linee guida elencate in questo capitolo. Sulla directory di Yahoo puoi trovare l'elenco dei principali quotidiani online (alfabetico, per paese o per popolarità): http://dir.yahoo.com/News_and_media/Newspapers.
- Leggi l'intervista a Matthew Carter sul design dei font Verdana e Georgia in http://www.will-harris.com/verdana-georgia.htm.

- In Windows Vista e Office 2007 Microsoft ha introdotto la ClearType Font Collection, composta da 6 famiglie di font: Calibri (sans serif, che sostituisce Times New Roman come font di default di Word e Arial come font di default di PowerPoint), Cambria (serif), Candara (sans serif), Consolas (spaziatura fissa), Constantia (serif) e Corbel (sans serif). Esamina ed esperimenta questi font, verificandone le caratteristiche sulle voci relative di Wikipedia.
- Confronta la comprensibilità di due quotidiani online in lingua italiana effettuando la valutazione dell'indice Gulpease di un campione di due brevi testi (per esempio articoli sugli stessi argomenti), utilizzando il servizio disponibile sul sito www.eulogos.net, o l'apposita funzione di Microsoft Word.
- Analizza i testi delle notizie di Due Parole (http://www.dueparole.it/) e confrontali con quelli di un quotidiano online di pari data.
- Analizza i testi del sistema di help online del tuo computer, dal punto di vista delle buone regole del plain language.

14. Valutare l'usabilità

Sintesi del capitolo

Questo capitolo descrive le due tecniche più utilizzate per valutare l'usabilità dei sistemi: le cosiddette valutazioni euristiche e i test di usabilità. Nel primo caso, la valutazione è eseguita da esperti di usabilità, con l'aiuto di regole più o meno dettagliate, che riflettono lo stato delle conoscenze del settore. Si citano le euristiche di Nielsen, molto conosciute. Nel secondo caso, i test sono effettuati da un campione di utenti, che utilizzano il sistema in un ambiente controllato, sotto osservazione da parte dei valutatori. I test di usabilità si possono classificare in formativi e sommativi; in test di compito e di scenario. Il capitolo termina con una serie di indicazioni pratiche per la preparazione e la esecuzione di un test di usabilità, e per la preparazione del rapporto di valutazione.

Verifiche e convalide

Nel modello di progettazione e sviluppo iterativi descritto in Figura 109 (pag.131), ad ogni ciclo di iterazione si effettuano dei test di valutazione dell'ultimo prototipo prodotto. Come specifica lo standard ISO 13407, "la valutazione è un passo essenziale in una progettazione human-centred. All'inizio del progetto, l'obiettivo principale sarà la raccolta di indicazioni per orientare le attività di progettazione successive. Nelle fasi più avanzate, con la disponibilità di prototipi più completi, sarà invece possibile quantificare il livello di raggiungimento degli obiettivi dell'utente e dell'organizzazione. Dopo il rilascio del sistema, sarà possibile monitorarne l'adeguatezza ai nuovi contesti di utilizzo."

I termini generici "valutazione" o "test" possono denotare due attività molto diverse:

- il controllo che il prodotto sia congruente con quanto espresso nei documenti di specifica dei requisiti. Per questo tipo di test si usa normalmente il termine *verifica* (*verification*);
- il controllo che il prodotto soddisfi effettivamente le esigenze per le quali è stato concepito. Per questo tipo di test si usa, invece, il termine *convalida* (*validation*).

Nel primo caso si tratta, come si usa dire nella lingua inglese, di verificare che il gruppo di progetto "made the things right"; nel secondo caso che "made the right things". La differenza fra verifica e convalida è sostanziale, perché non è detto che i documenti di specifica dei requisiti esprimano sempre correttamente le esigenze che il prodotto dovrebbe soddisfare. Infatti, l'estensore delle specifiche potrebbe avere male interpretato le richieste degli stakeholder del prodotto, che peraltro nel corso del progetto potrebbero cambiare.

Le attività di convalida sono molto più difficili delle verifiche. Non si tratta, infatti, di controllare la congruenza (o, come si dice, la *tracciabilità*) fra le caratteristiche del prodotto e le indicazioni contenute nel documento dei requisiti, ma di controllare che il prototipo soddisfi effettivamente le esigenze espresse (o, a volte, ancora inespresse) del committente. Per questo, la convalida non può essere condotta soltanto dal team di progetto, ma richiede necessariamente il coinvolgimento dell'utente e degli altri stakeholder del prodotto.

In questo libro ci occuperemo, in particolare, delle attività di valutazione dell'usabilità, rimandando per gli altri aspetti all'ampia letteratura esistente sul testing in generale. Per compierle si possono impiegare diverse tecniche, che rientrano in due grandi categorie:

- Valutazioni effettuate da parte di esperti di usabilità, senza alcun coinvolgimento dell'utente. Queste valutazioni prendono il nome di *ispezioni* (*inspection*). Le più note sono le cosiddette *valutazioni euristiche* (*euristic evaluation*).

- Valutazioni effettuate con il coinvolgimento dell'utente.
  Sono le più importanti e le più utilizzate. Fra queste, nel seguito saranno descritti i *test di usabilità* (*usability test*).

Valutazioni euristiche

L'aggettivo *euristico* si usa, in matematica, per denotare un procedimento non rigoroso che consente di prevedere o rendere plausibile un determinato risultato, che in un secondo tempo dovrà essere controllato e convalidato con metodi rigorosi. Nell'ingegneria dell'usabilità, si chiamano euristiche quelle valutazioni di usabilità effettuate da esperti, analizzando sistematicamente il comportamento di un sistema e verificandone la conformità a specifiche "regole d'oro" (chiamate, appunto, euristiche), derivanti da principi o linee guida generalmente accettati. In pratica, per eseguire una valutazione euristica, l'esperto di usabilità dovrebbe considerare una regola euristica alla volta, ed esaminare dettagliatamente le funzioni del sistema, per verificarne la conformità.

Le euristiche impiegate sono diverse. In letteratura se ne trovano alcune costituite da centinaia di regole dettagliate. È evidente che valutare un sistema in base a una tale quantità di regole risulta impraticabile. Si preferisce quindi, più spesso, utilizzare euristiche costituite da pochi principi guida generali. Fra queste, sono molto note le euristiche di Nielsen, costituite da dieci regole che, sebbene molto generali, permettono al valutatore di inquadrare i problemi rilevati in categorie bene individuate.

Le dieci euristiche di Nielsen, spiegate con le sue stesse parole, sono le seguenti[183]:

1. Visibilità dello stato del sistema
   Il sistema dovrebbe sempre informare gli utenti su ciò che sta accadendo, mediante feedback appropriati in un tempo ragionevole.
2. Corrispondenza fra il mondo reale e il sistema
   Il sistema dovrebbe parlare il linguaggio dell'utente, con parole, frasi e concetti familiari all'utente, piuttosto che termini orientati al sistema. Seguire le convenzioni del mondo reale, facendo apparire le informazioni secondo un ordine logico e naturale.
3. Libertà e controllo da parte degli utenti
   Gli utenti spesso selezionano delle funzioni del sistema per errore e hanno bisogno di una "uscita di emergenza" segnalata con chiarezza per uscire da uno stato non desiderato senza dover passare attraverso un lungo dialogo. Fornire all'utente le funzioni di undo e redo.
4. Consistenza e standard
   Gli utenti non dovrebbero aver bisogno di chiedersi se parole, situazioni o azioni differenti hanno lo stesso significato. Seguire le convenzioni della piattaforma di calcolo utilizzata.
5. Prevenzione degli errori
   Ancora meglio di buoni messaggi di errore è un'attenta progettazione che eviti innanzitutto l'insorgere del problema. Eliminare le situazioni che possono provocare errori da parte dell'utente, e chiedergli conferma prima di eseguire le azioni richieste.
6. Riconoscere piuttosto che ricordare
   Minimizzare il ricorso alla memoria dell'utente, rendendo visibili gli oggetti, le azioni e le opzioni. L'utente non dovrebbe aver bisogno di ricordare delle informazioni, nel passare da una fase del dialogo a un'altra. Le istruzioni per l'uso del sistema dovrebbero essere visibili o facilmente recuperabili quando servono.
7. Flessibilità ed efficienza d'uso
   Acceleratori – invisibili all'utente novizio – possono spesso rendere veloce l'interazione dell'utente esperto, in modo che il sistema possa soddisfare sia l'utente esperto sia quello inesperto. Permettere all'utente di personalizzare le azioni frequenti.

[183] Nielsen, J., and Molich, R., *Heuristic evaluation of user interfaces*, *Proc. ACM CHI'90 Conference, pagg.*249-256. Le 10 euristiche citate sono state riprese, in nostra traduzione, dal libro di J.Nielsen, *Usability Engineering*, 1994.

8. Design minimalista ed estetico

   I dialoghi non dovrebbero contenere informazioni irrilevanti o necessarie solo di rado. Ogni informazione aggiuntiva in un dialogo compete con le unità di informazione rilevanti e diminuisce la loro visibilità relativa.

9. Aiutare gli utenti a riconoscere gli errori, diagnosticarli e correggerli

   I messaggi di errore dovrebbero essere espressi in linguaggio semplice (senza codici), indicare il problema con precisione e suggerire una soluzione in modo costruttivo.

10. Guida e documentazione

    Anche se è preferibile che il sistema sia utilizzabile senza documentazione, può essere necessario fornire aiuto e documentazione. Ogni tale informazione dovrebbe essere facilmente raggiungibile, focalizzata sul compito dell'utente, e dovrebbe elencare i passi concreti da fare, senza essere troppo ampia.

La Figura 281 mostra una possibile corrispondenza fra le euristiche di Nielsen e le regole dell'ISO 9241 descritte nel capitolo 10. Si noti che più euristiche possono derivare da uno stesso principio e, d'altro canto, più principi possono giustificare una stessa euristica. Per esempio, l'euristica "flessibilità ed efficienza d'uso" può essere associata anche al principio "adeguatezza all'individualizzazione".

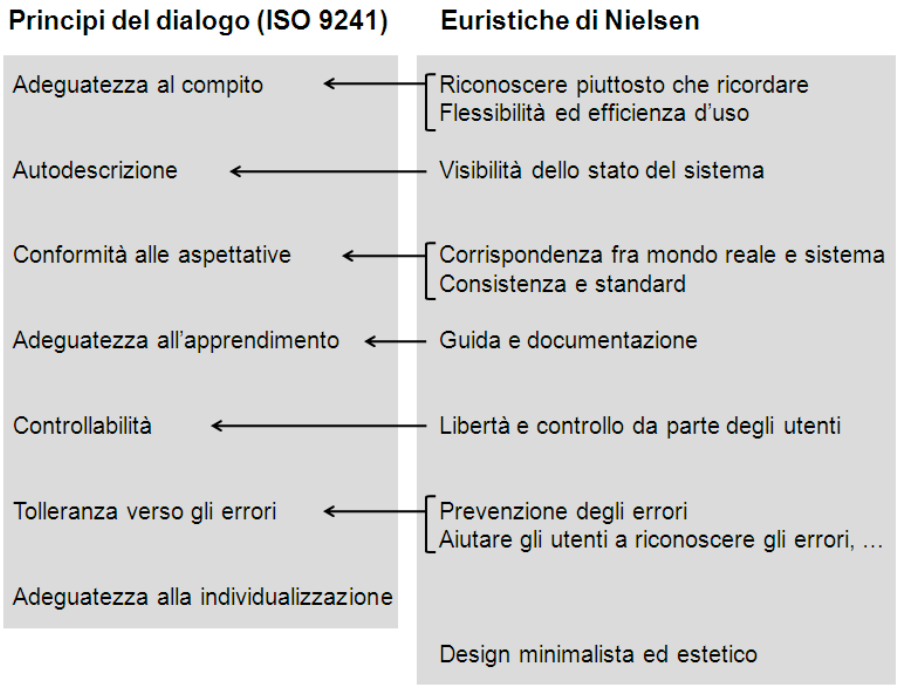


Figura 281. Confronto fra i principi del dialogo dell'ISO 9241e le euristiche di Nielsen

La valutazione euristica ha il vantaggio di essere relativamente poco costosa. Tuttavia fornisce risultati piuttosto soggettivi. Quanto più le euristiche sono generali, tanto più il risultato della valutazione dipenderà dall'esperienza, dalla sensibilità e, a volte, dalle preferenze personali del valutatore. Le esperienze condotte in molti progetti hanno mostrato che valutatori diversi tendono a trovare problemi diversi. La Figura 282 mostra i risultati della valutazione euristica di un sistema bancario effettuata da 19 valutatori, descritta dallo stesso Nielsen.[184] Il diagramma riporta in verticale i valutatori, e in orizzontale i 16 problemi di usabilità da questi individuati. Ogni quadratino nero indica un problema segnalato da un valutatore. I problemi sono ordinati lungo l'asse orizzontale: più sono a destra, più valutatori li hanno individuati. La figura mostra chiaramente che i diversi valutatori hanno trovato problemi diversi. Nessun problema è stato segnalato da tutti i valutatori, e solo 4 sono stati segnalati da più della metà dei valutatori. Alcuni problemi "facili"

[184] http://www.useit.com/papers/heuristic/heuristic_evaluation.html.

da trovare (quelli sulla destra) sono stati individuati da quasi tutti i valutatori, altri (quelli sulla sinistra) soltanto da pochi. Infine, i valutatori che hanno segnalato alcuni dei problemi più difficili da scoprire, non ne hanno invece segnalati alcuni che sono stati individuati da molti valutatori.

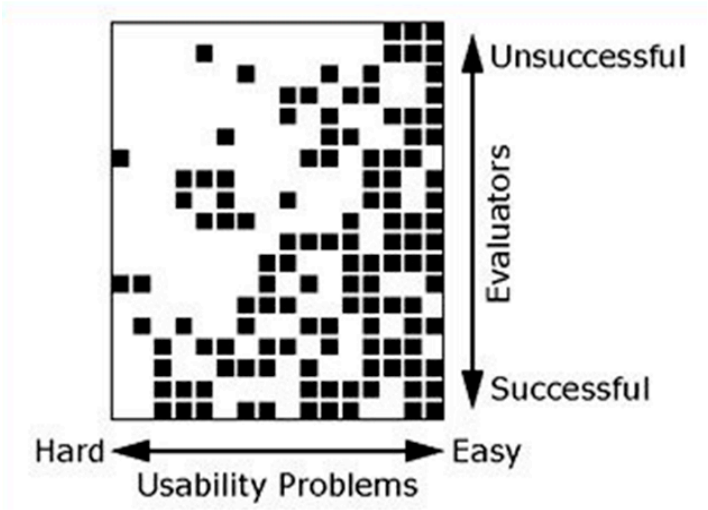


Figura 282. Risultati di una valutazione euristica di un sistema bancario (http://www.useit.com)

In sintesi, con la valutazione euristica è possibile ottenere buoni risultati solo impiegando più valutatori sullo stesso progetto, che analizzino separatamente il sistema senza comunicare fra loro. In ogni caso, questa tecnica non garantisce che vengano rilevati tutti i problemi di usabilità, e può capitare che vengano segnalati problemi che, in realtà, non esistono. Ciò è dovuto al fatto che i valutatori possono avere delle preferenze personali su come determinate funzioni del sistema debbano essere sviluppate, e non è detto che queste corrispondano a criteri oggettivamente verificabili. È anche evidente che i risultati saranno tanto più affidabili quanto più i valutatori saranno esperti nel particolare ambito in esame. Di conseguenza, la valutazione euristica può aggiungersi, ma non sostituirsi ai test con gli utenti, che devono comunque essere condotti. I due tipi di valutazioni sono complementari, e possono dare risultati diversi (Figura 283).

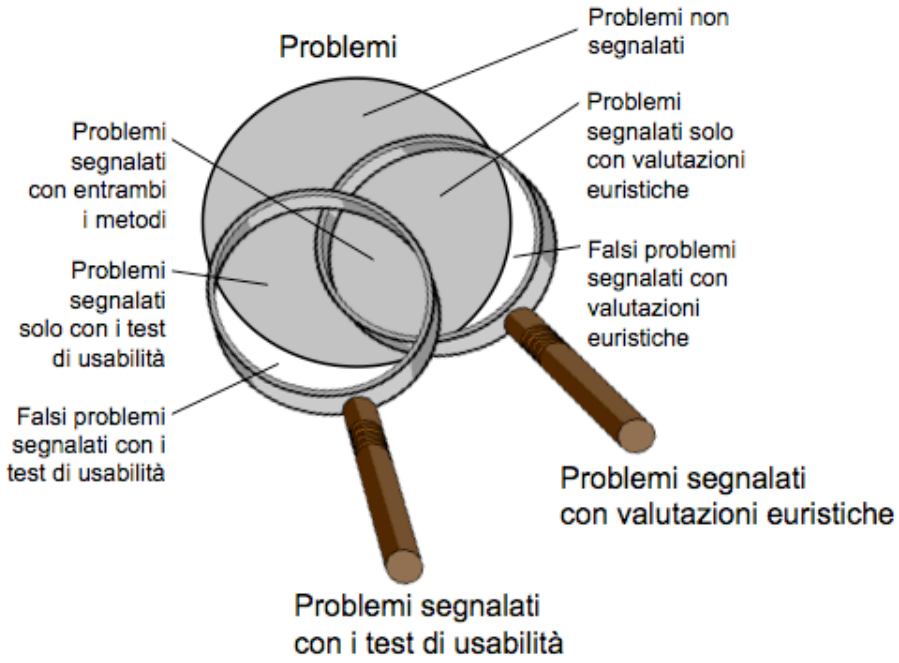


Figura 283. a) Risultati della valutazione euristica; b) confronto fra valutazione euristica e test di usabilità

Test di usabilità

Un test di usabilità consiste nel far eseguire a un gruppo di utenti dei compiti tipici di utilizzo del sistema in un ambiente controllato. Si sceglie un campione di utenti che sia rappresentativo della categoria di utenti cui il sistema si rivolge, e si chiede a tutti di svolgere, separatamente, gli stessi compiti. Chi conduce il test osserva e analizza il loro comportamento per comprendere se, dove e perché essi hanno incontrato delle difficoltà. Il test coinvolge, oltre all'utente che prova il sistema, almeno due altre persone: un *facilitatore*, che ha il compito di gestire la "regia" della prova, e uno o più *osservatori*, che assistono al test, annotando i comportamenti dell'utente che ritengono significativi. Il ruolo degli osservatori è critico: essi dovrebbero conoscere bene il sistema, e avere eseguito personalmente i compiti chiesti agli utenti. Solo in questo modo saranno in grado di interpretarne e valutarne correttamente i comportamenti. Facilitatori e osservatori potranno essere degli esperti di usabilità, oppure dei membri del team di progetto, a seconda dei casi.

Quando si usino prototipi di carta (pag.191), o tecniche con il mago di Oz (pag.193), servirà una terza persona, con il compito di simulare il sistema. Come abbiamo già osservato, si tratta di un compito piuttosto impegnativo, che richiede preparazione e attenzione; pertanto, non potrà essere svolto né da chi coordina il test né da chi ne osserva lo svolgimento. A simulare il sistema sarà molto spesso un progettista, che conosca ne bene il funzionamento previsto.

Un test di usabilità ha lo scopo di ricavare indicazioni concrete per il miglioramento del sistema. Chi lo conduce dovrà esaminare in dettaglio le operazioni svolte dagli utenti per capire dove nascono le difficoltà, da che cosa sono causate e in quale modo possano essere rimosse. Per questo, è molto utile la cosiddetta tecnica del "pensare ad alta voce" (*think aloud*), che consiste nel chiedere all'utente di esprimere a voce alta ciò che pensa mentre compie le varie operazioni. Questo è molto utile, perché permette agli osservatori di raccogliere informazioni sulle strategie messe in atto dall'utente nell'esecuzione dei compiti, e sulle difficoltà che egli incontra durante il test.[185]

L'analisi del comportamento degli utenti non può essere condotta in tempo reale durante lo svolgimento del test, ma deve essere compiuta dopo, con la necessaria tranquillità. Anche se l'utente esprime ad alta voce, durante la prova, le difficoltà che sperimenta, le cause di queste difficoltà possono non essere evidenti. Per identificarle, e comprendere come possano essere rimosse, sarà necessario riesaminare con attenzione la sequenza di azioni eseguite dall'utente. Durante il test, l'osservatore dovrà quindi prendere appunti, registrando le situazioni in cui l'utente manifesta incertezza o commette degli errori. Questi appunti saranno riesaminati in seguito, per individuare le cause del problema e studiare le correzioni più opportune.

Se possibile, è preferibile eseguire una registrazione (audio e video) della sessione di test, per rivedere in seguito tutto ciò che è avvenuto durante la prova. La tecnica più comune consiste nel riprendere con una telecamera il viso dell'utente mentre esegue il test, registrando contemporaneamente le sue parole e, con un'altra telecamera, il sistema. Le due registrazioni dovranno poi essere viste in modo sincronizzato: spesso le espressioni del viso dell'utente sono altrettanto rivelatrici delle sue parole e delle sue azioni. Le Figura 284 e Figura 285 sono tratte, rispettivamente, dalla registrazione di un test con un prototipo di carta e con un prototipo PowerPoint per un'applicazione iPhone. In entrambe, sulla sinistra appare il viso dell'utente, sulla destra le operazioni che compie sul sistema. Nella prima figura, sono stati inseriti in sovraimpressione, per comodità, i nomi dei compiti in esecuzione ("Leggi l'avviso più recente").

[185] La tecnica viene correntemente utilizzata, anche se non è esente da difetti. Essa è infatti piuttosto intrusiva, e crea un contesto sostanzialmente diverso dalla esecuzione silenziosa. Infatti, modifica il carico cognitivo dell'utente, e può portarlo a modificare la strategia di esecuzione dei vari compiti.



Figura 284. Registrazione di un test di usabilità con prototipo di carta

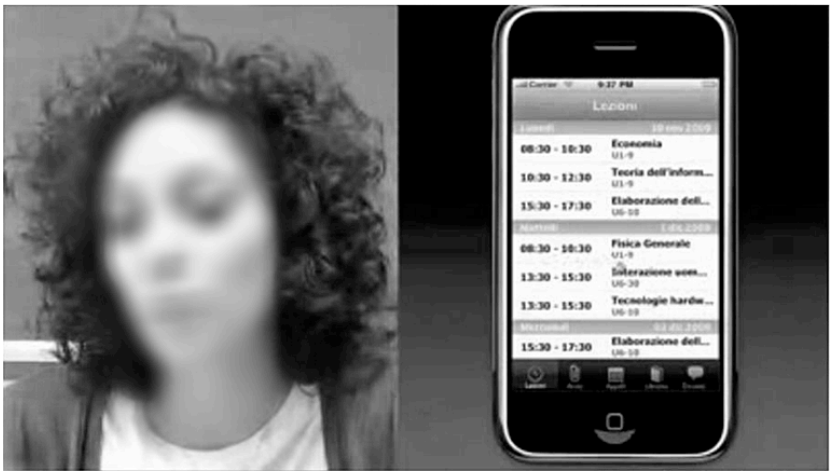


Figura 285. Registrazione di un test d'usabilità per un'applicazione su iPhone

Nel caso di sistemi eseguiti al computer, tutto questo si può fare, molto semplicemente, con una sola webcam che riprenda il viso dell'utente, un microfono e un programma in esecuzione sullo stesso computer usato per il test, che registri le immagini che appaiono sul video, in modo sincronizzato con le registrazioni audio e video. Tali programmi (ne esistono anche di gratuiti) permettono poi di esaminare in dettaglio, per così dire alla moviola, le azioni effettuate dall'utente e metterle in corrispondenza con le espressioni del viso e le parole pronunciate. La Figura 286 mostra una immagine dalla registrazione di un test di un sito web.



Figura 286. Registrazione di un test di usabilità con un programma apposito

Fino a qualche anno fa era diffusa la convinzione che per fare un buon test di usabilità fosse indispensabile usare un laboratorio appositamente attrezzato (*usability lab*). Esso è costituito da due stanze contigue: una per l'utente che prova e una per gli osservatori (Figura 287). Nella prima, l'utente esegue il test da solo; nella seconda, il facilitatore e gli osservatori lo possono vedere (non visti) attraverso un finto specchio, che permette a chi si trova nella sala di osservazione di vedere l'interno della sala prove, ma non viceversa. La sala osservazione contiene inoltre tutti gli apparati necessari per controllare la registrazione audio e video delle prove: una vera e propria sala di regia.

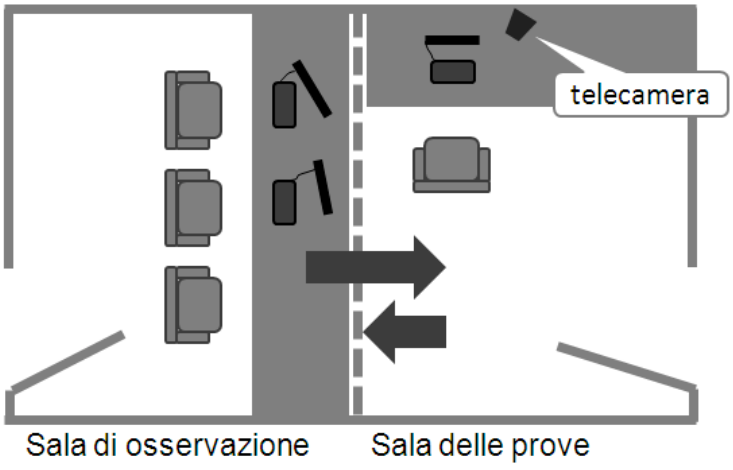


Figura 287. Usability lab

Laboratori di questo tipo sono piuttosto costosi, e sono usati dalle organizzazioni che devono condurre molti test, in modo professionale. Tuttavia, per condurre dei buoni test di usabilità non è indispensabile disporre di una struttura di questo tipo, e ci si può organizzare in modo molto più semplice. Oggi quasi ogni computer è dotato di una webcam integrata e, come abbiamo detto, esistono numerosi pacchetti software che gestiscono la registrazione senza che siano necessarie apparecchiature e competenze particolari. Utente, facilitatore e osservatori si possono semplicemente disporre attorno a un tavolo, come in Figura 288.

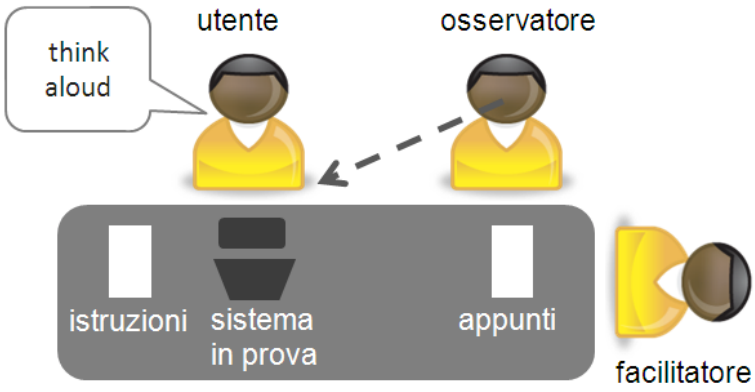


Figura 288. Test di usabilità informale

Se si dispone di una stanza riservata dove condurre il test, tanto meglio, ma neanche questo è indispensabile: la Figura 289 mostra un test di usabilità condotto con successo in un laboratorio universitario molto affollato. Ovviamente, in una situazione di questo tipo sarà più difficile fare seguire il test da più osservatori: la logistica non lo consente. In effetti, uno dei principali vantaggi di uno usability lab come quello di Figura 287 è che si può fare assistere alle prove anche qualche progettista. Vedere con i propri occhi come l'utente si blocchi tentando di usare il sistema, e sentirne i commenti – a volte esasperati - può rivelarsi veramente molto istruttivo.

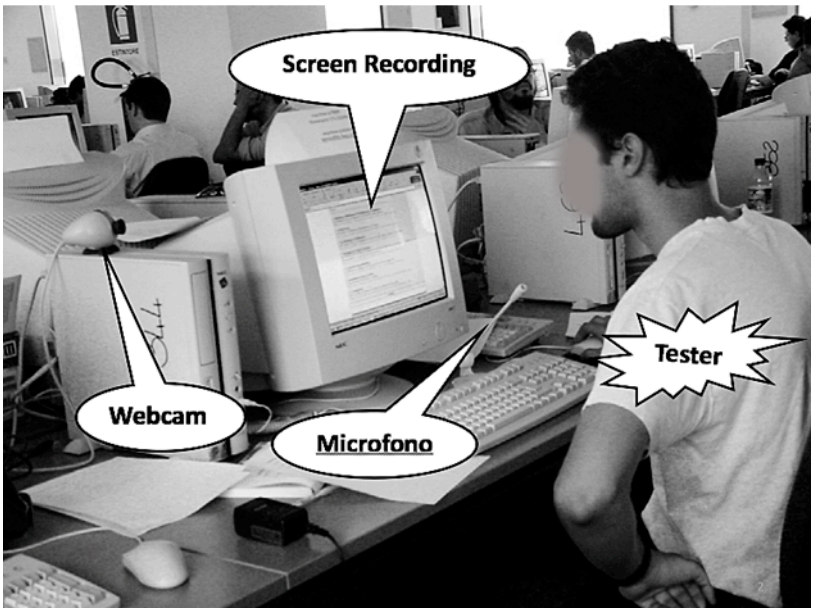


Figura 289. Test di usabilità di fortuna

Test formativi e test sommativi

Vedremo in dettaglio, più oltre, come un test di usabilità debba essere organizzato e condotto. Per il momento, segnaliamo che i test di usabilità possono essere classificati, in funzione dei loro obiettivi, in due grandi categorie: test *formativi* (*formative*) e test *sommativi* (*summative*).

I primi sono utilizzati durante il ciclo iterativo di progettazione, per sottoporre i vari prototipi a prove d'uso con gli utenti, allo scopo di identificarne i difetti e migliorarne l'usabilità. Si chiamano formativi perché, appunto, contribuiscono a "dare forma" al prodotto: il loro scopo è individuare il maggior numero possibile di problemi. Essi sono particolarmente utili nelle fasi iniziali della progettazione, quando il design concept è appena abbozzato. È spesso conveniente eseguire questi test in modo rapido e approssimativo (*quick & dirty*), facendo esercitare a un piccolo numero di utenti le funzionalità principali del sistema. Questo perché nelle fasi iniziali del progetto, quando il design concept è appena abbozzato, i test mettono in luce rapidamente i difetti macroscopici, che richiedono una parziale (o totale) riprogettazione dell'interfaccia. Quindi non conviene andare troppo per il sottile: si sprecherebbe del tempo provando "a tappeto" funzioni che, nella riprogettazione, saranno probabilmente modificate.

Inoltre, si verifica spesso un *effetto di mascheramento*: ogni problema di usabilità nel quale incappiamo monopolizza, per così dire, la nostra attenzione e ci impedisce spesso di vederne altri, soprattutto se di più lieve entità. Li scopriremo all'iterazione successiva, quando il problema iniziale sarà stato rimosso (Figura 290). Quindi si prova in fretta, si modifica rapidamente il prototipo eliminando i difetti più evidenti, e si prova ancora, e così via. Per il primo test di un prototipo iniziale di carta, 2-3 utenti sono in genere sufficienti.

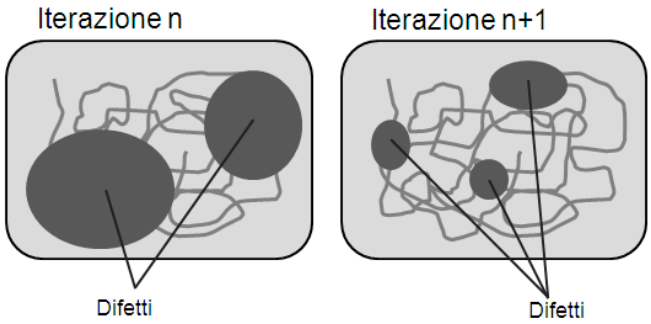


Figura 290. Effetto di mascheramento: i difetti macroscopici impediscono di vedere difetti di minore entità

Jakob Nielsen ha introdotto il termine *discount usability* (*usabilità scontata*) per indicare queste tecniche, rapide, poco costose e non troppo sistematiche per individuare i problemi di usabilità. Già in un articolo del 1993[186] aveva osservato che utilizzare molti utenti nei test di usabilità è inutilmente costoso, e che sono in genere sufficienti da 3 a 5 utenti per rilevare più del 75% dei problemi di usabilità in un sistema (Figura 291).

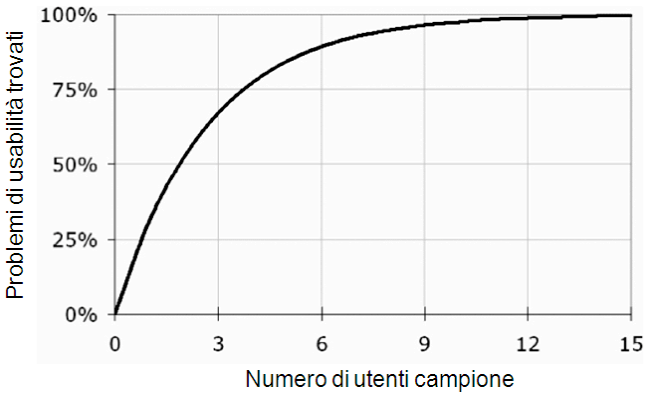


Figura 291. La "regola di Nielsen" per i test di usabilità (da www.useit.com)

[186] J.Nielsen, T.K.Landauer, A Mathematical Model of the Finding of Usability Problems", in Proceedings of the ACM INTERCHI 93 Conference, Amsterdam, Aprile 1993, pp.206-213. Per una sintesi, si veda l'Alertbox di J.Nielsen del 19 marzo 2000, Why You Only Need to Test With 5 Users, in www.useit.com, da cui è tratta la Figura 291.

Questa semplice indicazione pratica (chiamata *regola di Nielsen*) è stata in seguito criticata da diversi autori, che ne hanno contestato la validità generale. Lo stesso Nielsen l'ha parzialmente modificata, portando il numero suggerito di utenti a 5. In pratica, dice Nielsen, i primi 5 utenti metteranno in evidenza la maggior parte dei problemi di usabilità più significativi: gli utenti successivi non farebbero altro che confermare gli stessi risultati, aggiungendo ben poco di nuovo.

I test della seconda categoria si chiamano sommativi. Il termine deriva dal verbo "sommare", ed indica una valutazione più complessiva del prodotto, al di fuori – o al termine – del processo di progettazione e sviluppo. Sono test più completi di quelli formativi, che non hanno lo scopo di fornire indicazioni ai progettisti, ma di valutare in modo sistematico pregi e difetti del prodotto, o sue particolari caratteristiche.[187] Sono di solito condotti quando il sistema è completamente funzionante, per esempio per indicarne i punti deboli e valutare l'opportunità di un redesign migliorativo. Oppure per confrontarne le caratteristiche con quelle di sistemi concorrenti. Anche se non si possono dare regole fisse, un test di usabilità ben strutturato, di tipo sommativo, coinvolge di solito un numero maggiore di utenti, per esempio 10-15, o anche di più.

Qualunque sia la strategia adottata, i soggetti da utilizzare nei test dovranno sempre essere scelti con cura, affinché rappresentino utenti tipici. Per poter interpretare correttamente l'esito di ciascun test, chi lo conduce dovrà conoscere, per ciascun soggetto, il livello di esperienza nell'uso di sistemi analoghi a quello in esame. Il suo profilo dovrebbe essere classificato su due dimensioni: il livello di conoscenza del dominio applicativo del sistema, e il livello di familiarità con la tecnologia utilizzata (Figura 292). Queste due dimensioni sono largamente indipendenti. Per esempio, nel test del sito web di una compagnia aerea, alcuni utenti potrebbero avere una lunga esperienza di viaggi in aereo, ma non avere mai fatto un acquisto online. La conoscenza del profilo degli utenti impiegati nelle prove è indispensabile, per interpretarne correttamente i risultati.

Nella scelta degli utenti, si potranno scegliere rappresentanti di tutti i quadranti della figura, o solo di alcuni, a seconda degli obiettivi del test e della natura del sistema. In genere conviene scegliere utenti con caratteristiche diverse. In tal modo, è probabile che essi affronteranno i compiti assegnati con strategie differenti, esercitando aspetti diversi del sistema. In ogni caso, tutti gli utenti dovrebbero avere almeno un potenziale interesse nelle funzioni svolte dal sistema. Non ha senso chiedere a chi non è mai salito su un aereo di prenotare un volo sul sito di una compagnia aerea. Si rischierebbe di incontrare delle difficoltà che non derivano dal sistema, ma dalla poca dimestichezza che l'utente ha con il problema che gli è stato sottoposto. I risultati della prova ne saranno probabilmente inquinati. È sbagliato, per "fare numero", reclutare le persone più facilmente disponibili, senza altri accertamenti: si devono proprio selezionare dei potenziali clienti del sistema.

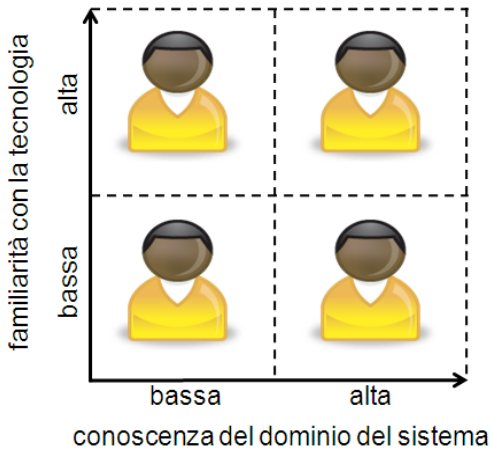


Figura 292. Le due dimensioni del profilo degli utenti

[187] Il termine "valutazione sommativa" è derivato dalla docimologia, dove si usa per indicare le verifiche praticate a conclusione dei processi didattici sulla validità delle attività effettuate, in rapporto a determinati obiettivi formativi.

Naturalmente, gli utenti per le prove non dovranno mai essere scelti all'interno del gruppo di progetto. I progettisti conoscono troppo bene il sistema per fornirci delle indicazioni significative. Anche se ci assicurassero che faranno il possibile per ignorare quello che sanno e di "mettersi nei panni dell'utente", le loro prove sarebbero inevitabilmente poco attendibili.

Test di compito e test di scenario

Rispetto alle attività condotte dagli utenti durante le prove, i test di usabilità possono essere classificati in due grandi categorie: *test di compito* e *test di scenario.*

Nei test di compito, gli utenti svolgono singoli compiti, che permettono di esercitare funzioni specifiche del sistema. Per esempio, nel caso di un sito web di un supermercato:

> Compito 1: Ricerca nel catalogo dei prodotti una scatola di pomodori pelati da 500 grammi;
>
> Compito 2: Comprane due, pagando con la tua carta di credito;
>
> Compito 3: Verifica lo stato del tuo ordine precedente.

Questi test possono essere eseguiti anche quando il sistema non è completamente sviluppato. Per esempio, per eseguire la ricerca nel catalogo dei prodotti, non è necessario che le funzioni per l'acquisto siano già disponibili. Essi sono tanto più utili quanto più i compiti riflettono i casi d'uso reali del sistema. Un errore frequente dei conduttori inesperti è quello di suggerire implicitamente agli utenti le operazioni da eseguire, dando loro, in pratica, una serie di istruzioni, invece che un problema da risolvere. L'utente deve, invece, essere posto in situazioni simili a quelle in cui si troverà nell'uso reale del sistema, quando dovrà decidere, da solo, che cosa fare. Se l'utente è troppo guidato, i problemi di usabilità, anche se presenti, non emergeranno, e il test non sarà di alcuna utilità. Per esempio, consideriamo la seguente riformulazione dei due compiti dell'esempio precedente:

> Compito 1: Registrati al sistema, fornendo il tuo indirizzo di email e una password;
>
> Compito 2: Entra nella sezione per gli acquisti online e ricerca la sottosezione dei prodotti in scatola;
>
> Compito 3: Ricerca una scatola di pomodori pelati da 500 gr e caricala nel carrello della spesa;
>
> Compito 4: Conferma l'acquisto fornendo questo indirizzo di consegna e questi dati per l'addebito su carta di credito.

Procedendo in questo modo, daremmo all'utente troppe indicazioni sul funzionamento del sistema, anche se in modo implicito. Per esempio, gli faremmo sapere che, per comprare qualcosa, deve prima registrarsi, fornendo come user-id il suo indirizzo di posta elettronica. Poi che esiste una sezione per gli acquisti online, e che questa è organizzata in sottosezioni, una delle quali riguarda i prodotti in scatola. Quindi che per comprare qualcosa bisogna caricarlo nel carrello, e così via. Tutte queste informazioni non sono invece disponibili all'utente in un contesto di acquisto reale, soprattutto se egli non ha mai utilizzato un sistema simile. In definitiva, in un contesto reale l'utente potrebbe incontrare delle difficoltà che in un test così formulato non si presentano mai.

La seconda categoria è costituita dai test di scenario. In questo caso, agli utenti viene indicato un obiettivo da raggiungere attraverso una serie di compiti elementari, senza indicarli esplicitamente. L'utente dovrà quindi impostare una propria strategia di azione. Per un test più realistico, all'utente potrà essere indicato uno scenario complessivo che definisca meglio il contesto in cui dovrà immaginare di muoversi. Per esempio, per il sito del supermercato, lo scenario proposto agli utenti potrebbe essere il seguente:

> Domani sera hai due amici a cena, ma non hai il tempo di andare al supermercato. Decidi quindi di fare la spesa online, pagando con la tua carta di credito. Collegati al sito e ordina gli ingredienti per una cena veloce e poco costosa, ma simpatica.

Come si vede da questo esempio, i test di scenario possono mettere alla prova l'utente (e il sistema) in modo molto più impegnativo dei test di compito. In particolare, permettono agli utenti di utilizzare il sistema in relazione alle proprie specifiche necessità, preferenze e abitudini. Nello scenario indicato, gli utenti terranno conto delle loro preferenze alimentari, e di quelle dei loro amici. Utilizzeranno la loro carta di credito, e chiederanno la consegna al loro vero indirizzo, verificando se questa potrà essere effettuata in tempo utile per la cena. Così, la strategia che gli utenti seguiranno per raggiungere l'obiettivo potrebbe essere molto diversa da quella prevista dai progettisti. Per questo motivo, i test di scenario possono essere molto utili per individuare eventuali carenze nell'impostazione della struttura complessiva dell'interazione, o mancanze di funzionalità utili. Dunque, si dovrebbe cercare di anticipare, per quanto è possibile, i test di scenario all'inizio del progetto, usando anche prototipi parziali o a bassa fedeltà. I test di compito permettono, invece, una verifica di usabilità più fine, perché localizzata a specifici casi d'uso. Quindi possono essere più utili quando l'architettura funzionale del sistema sia già ben consolidata, per provare l'usabilità di specifiche funzioni.

È indispensabile che le istruzioni agli utenti (l'elenco dei compiti da svolgere o la descrizione degli scenari) siano date per iscritto, nel modo più chiaro possibile. Solo in questo modo tutti gli utenti partecipanti al test si troveranno nelle medesime condizioni: le spiegazioni date a voce, inevitabilmente diverse di volta in volta, potrebbero influenzare i partecipanti in modo differente, rendendo i risultati dei test poco confrontabili.

Misure

Oltre all'osservazione qualitativa dei comportamenti degli utenti, durante un test di usabilità è utile raccogliere anche delle misure oggettive. Quelle più significative sono il tempo impiegato da ogni utente per l'esecuzione di ciascun compito, e il tasso di successo (*success rate*), cioè la percentuale di compiti che ciascuno riesce a portare a termine.

Il calcolo del tasso di successo può tener conto anche dei compiti eseguiti solo in parte. Si consideri, per esempio, il test descritto in Figura 293, in cui 4 utenti eseguono 6 compiti. Su 24 compiti, 9 sono stati portati a termine (S), e 4 sono stati eseguiti solo in parte (P). Per i rimanenti, gli utenti hanno fallito (F).

	Compito 1	Compito 2	Compito 3	Compito 4	Compito 5	Compito 6
Utente 1	F	F	S	F	F	S
Utente 2	F	F	P	F	P	F
Utente 3	S	F	S	S	P	S
Utente 4	S	F	S	F	P	S

Legenda: S=successo F=fallimento P=successo parziale

Figura 293. Sintesi risultati di un test di usabilità

In questo caso, il tasso di successo potrebbe essere calcolato così:

$$\text{Tasso di successo} = (9 + (4 * 0{,}5)) / 24 = 46\%$$

In questa formula, ogni successo parziale è stato conteggiato, convenzionalmente, come la metà di un successo pieno. Il risultato di questo calcolo ci dice che gli utenti non sono riusciti a portare a termine più della metà dei compiti assegnati. Questo dato ci dà una indicazione piuttosto significativa sulla usabilità del sistema. Esso andrebbe poi meglio interpretato con l'aiuto di informazioni più dettagliate sugli utenti e su quali compiti hanno avuto i problemi maggiori.

Come condurre un test di usabilità

Come abbiamo già detto, è preferibile che il team che conduce il test sia costituito da almeno due persone. Una avrà il compito di dirigere le attività e di interloquire con gli utenti, e nel frattempo verificare che le registrazioni, se vengono fatte, procedano correttamente. L'altra osserverà con attenzione il comportamento dell'utente durante il test, senza interferire, prendendo appunti sulle situazioni più significative. Quest'attività è molto importante, anche nel caso in cui il test venga registrato. Infatti, l'esame di una registrazione può richiedere molto tempo (diverse ore per ogni ora di registrazione). Se l'osservatore ha già localizzato durante il test i problemi principali, nell'analisi della registrazione si potrà concentrare sui punti critici, con un significativo risparmio di tempo.

Un test di usabilità viene condotto in quattro fasi successive, come indicato in Figura 294. Esaminiamole separatamente.

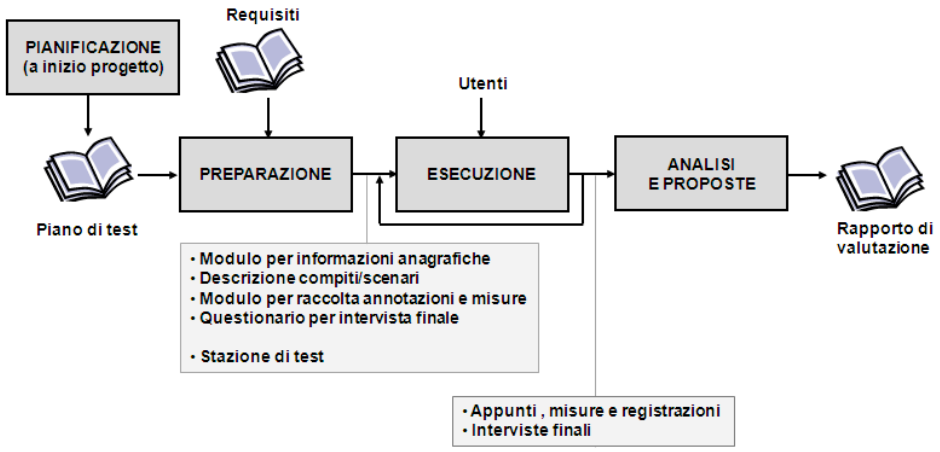


Figura 294. Le fasi di un test di usabilità

Pianificazione

L'organizzazione di un test di usabilità dipende in modo sostanziale dagli scopi da raggiungere, che dipendono dalla natura del prodotto e dalla strategia della sua realizzazione. Come abbiamo visto nel capitolo 9, i prototipi prodotti in un ciclo di sviluppo iterativo possono essere di tipo diverso, e dovrebbero essere realizzati in accordo a un piano di lavoro definito all'inizio del progetto. Il piano dovrebbe specificare la natura e lo scopo di ogni prototipo: i test su di esso dovrebbero essere progettati di conseguenza.

Anche lo standard ISO 13407 raccomanda che l'intero processo di valutazione sia pianificato in anticipo, precisando, tra l'altro, "quali parti del sistema devono essere valutate e come; quali prototipi dovranno essere realizzati, come deve essere eseguita la valutazione e con quali risorse; quali dovranno essere le interazioni con gli utenti e come dovrà essere condotta l'analisi dei risultati."

Preparazione del test

Nella fase di preparazione del test, il team di valutazione deve innanzitutto definire il numero e il profilo degli utenti campione e i compiti (o gli scenari, nel caso dei test di scenario) che si richiederà loro di svolgere. Sono decisioni molto delicate, poiché da esse dipenderà in larga misura l'utilità del test.

Nel caso di un test formativo inserito nel processo di sviluppo, si seguirà spesso la regola di Nielsen di cui abbiamo parlato più sopra. Il lavoro potrà anche essere svolto dagli stessi progettisti, se lo sanno fare, senza il supporto di un esperto di usabilità. Nel caso, invece, in cui il test di usabilità costituisca un evento a sé stante, per esempio per valutare l'opportunità di interventi migliorativi su un sistema esistente, occorrerà un'organizzazione più robusta. La durata di ogni singolo test potrà essere più lunga (ma di solito non durerà più di un'ora o un'ora e mezza al massimo per ciascun utente). Il numero degli utenti sarà maggiore, tenendo comunque presente che normalmente si ritiene più produttivo fare tanti test con pochi soggetti che pochi test con molti soggetti. In questi casi anche l'organizzazione del team di

valutazione dovrà essere potenziata. In un test di una certa ampiezza si raccolgono informazioni in grande quantità, e bisogna poi saperne trarre le dovute conclusioni. In questo caso, l'inserimento nel team di un professionista dell'usabilità è fortemente consigliabile.

Proseguendo nella preparazione del test, il team di valutazione deciderà quindi le misure da raccogliere, e predisporrà tutti gli aspetti relativi alla logistica per l'esecuzione delle prove (postazione di lavoro e di osservazione, strumenti di registrazione, e così via), in modo che queste possano avvenire senza troppi disturbi. Preparerà infine i materiali necessari allo svolgimento dei test, e in particolare:

- un modulo per raccogliere le informazioni sugli utenti: informazioni anagrafiche, livello di esperienza, competenze nell'ambito specifico del sistema;
- il testo con le istruzioni per lo svolgimento delle prove da consegnare agli utenti: la descrizione di compiti e scenari dovrebbe essere concisa, ma particolarmente chiara, per evitare chiarimenti e spiegazioni durante lo svolgimento della prova;
- la modulistica che gli osservatori utilizzeranno per raccogliere le misure relative all'esecuzione di ciascun compito, e le loro annotazioni durante il test. Questi moduli devono permettere di annotare rapidamente le informazioni utili, e quindi dovrebbero essere predisposti in funzione delle caratteristiche del test. Per esempio, si possono preparare stampe delle schermate video del sistema, su cui annotare gli aspetti critici, e una tabella come quella di Figura 293 per raccogliere le misure relative ai singoli utenti e compiti;
- un questionario per le interviste finali degli utenti, di cui parleremo fra poco.

Esecuzione del test

La fase di esecuzione del test vera e propria, se tutto è già bene organizzato e ci si limita a un test con pochi utenti, non dura in genere più di qualche ora complessivamente. Un test più ampio richiederà, al massimo, una o due giornate di lavoro.

È molto importante che, durante il colloquio di spiegazione iniziale con ciascun utente, sia chiarito molto bene che l'obiettivo della prova è valutare il sistema, e non la capacità dell'utente di svolgere bene e rapidamente i compiti assegnati. È indispensabile che il facilitatore metta ogni utente a suo agio, per ridurre al massimo lo "stress da esame" che non sarà mai del tutto eliminabile, e che potrebbe compromettere l'esito della prova. Bisogna spiegare bene che se una persona trova difficile usare il sistema, questo non avviene perché è incapace, ma perché il sistema è progettato male. A ogni utente dovrà poi essere esplicitamente garantita la riservatezza delle eventuali registrazioni che saranno effettuate, che dovranno essere visionabili esclusivamente dai team di valutazione e di progetto, a meno che egli non firmi un'esplicita liberatoria che autorizzi una diffusione più ampia.

I test devono essere condotti singolarmente, un utente alla volta. È opportuno prevedere, per ciascuno, un breve periodo di familiarizzazione con il sistema, prima del test vero e proprio. Durante lo svolgimento della prova i valutatori dovranno interferire il meno possibile: solo il facilitatore è autorizzato a parlare con l'utente, e i suoi interventi dovranno essere limitati allo stretto indispensabile, per non influenzarne il comportamento. Il suo scopo sarà esclusivamente quello di rassicurarlo in caso di difficoltà, incitandolo a proseguire con tranquillità, senza mai suggerire le azioni da compiere e senza fornire o chiedere spiegazioni. Dovrà invece, quando necessario, ricordargli il "thinking aloud", cioè di commentare ad alta voce ciò che sta facendo: che cosa si propone di fare, che cosa vede sullo schermo, come pensa di dover proseguire, quali difficoltà sta incontrando, quali dubbi ha, e così via. Questo sarà molto utile agli osservatori che prendono appunti, e che dovranno poi ricostruire, anche con l'aiuto delle eventuali registrazioni, le cause dei problemi incontrati. Il facilitatore avrà poi il compito di interrompere l'utente nel caso che, durante l'esecuzione di un certo compito, abbia incontrato difficoltà che non riesce a superare.

Al termine del test d'usabilità, è utile intervistare gli utenti sull'esperienza che hanno appena fatto. In queste interviste, che conviene condurre utilizzando un questionario appositamente predisposto, l'intervistatore chiederà, a ogni utente, quali sono, a suo parere, i punti di forza e di debolezza del sistema, gli aspetti che dovrebbero essere migliorati, e quelli che ha gradito maggiormente. Queste interviste possono fornire ulteriori indicazioni, ma non devono sostituire le prove

d'uso del sistema. Infatti gli utenti si limitano spesso a indicazioni generiche, e sono raramente in grado di individuare con precisione le cause delle loro dfficoltà.

Analisi dei risultati e proposte migliorative

L'ultima fase del test è quella in cui si analizza il materiale raccolto (appunti ed eventuali registrazioni audio/video) e si traggono le conclusioni. È la fase più delicata, e richiede tempo e grande cura. Anche una sessione di test breve, se riesaminata con attenzione, può fornire molte informazioni. Ogni gesto, ogni frase, ogni esclamazione dell'utente è un indizio importante, che va considerato e discusso dal team di valutazione, per individuarne cause e implicazioni.

Ci sono alcuni errori tipici dei valutatori poco esperti, che vanno evitati. Il primo è di limitarsi sostanzialmente a riportare i giudizi espressi dagli utenti nelle interviste successive al test. Questi sono utili, ma costituiscono solo una porzione abbastanza marginale dei risultati di un test ben condotto. Infatti spesso l'utente tende a esprimere giudizi o sensazioni di carattere generale (es.: "la fase di login è troppo complicata e mi chiede informazioni inutili"), senza essere in grado di risalire con precisione a *tutte* le cause di tali giudizi o sensazioni. Se lo fa, la sua analisi potrebbe rivelarsi sbagliata: non possiamo pretendere che l'utente sia un esperto di usabilità. Può capitare anche che l'utente, dopo una sessione di prova che ha mostrato evidenti difficoltà, esprima un giudizio sostanzialmente positivo sul sistema. Quindi il valutatore non deve mai accontentarsi dei commenti degli utenti, ma deve sempre compiere un'analisi diretta e dettagliata dei loro comportamenti, esaminando il materiale registrato o gli appunti presi durante le sessioni di prova. Il secondo errore tipico è quello di limitarsi all'elencazione di poche difficoltà macroscopiche, senza andare in profondità. Occorre, invece, elencare analiticamente tutti i problemi individuati, grandi e piccoli: solo così il test ci darà il massimo rendimento.

Il risultato di quest'analisi è un elenco dei problemi incontrati nello svolgimento di ciascun compito, descritti in modo circostanziato. A ciascun problema il team di valutazione assegna un livello di gravità, in base a considerazioni di vario tipo: il numero di volte che tale problema è stato evidenziato nei test, il livello d'esperienza degli utenti che hanno sperimentato il problema, l'effetto che il problema ha avuto sul completamento del compito (il problema ha impedito di portarlo a termine, o l'utente ha trovato comunque una soluzione o un percorso alternativo che gli ha permesso di arrivare al risultato desiderato?).

L'elenco dei problemi rilevati sarà poi riesaminato per produrre un elenco di interventi proposti per migliorare il prodotto (o il prototipo). Esse dovranno essere organizzate in diversi livelli di loro priorità, per esempio:

- Priorità 1: intervento indispensabili e urgenti

  Sono interventi per risolvere problemi bloccanti, senza i quali il sistema non può essere utilizzato per gli scopi per cui è stato progettato.
- Priorità 2: interventi necessari

  Interventi che risolvono problemi di usabilità gravi, ma che non impediscono all'utente di utilizzare il sistema. Per esempio, possono esistere dei percorsi alternativi che permettono comunque all'utente di raggiungere i suoi obiettivi.
- Priorità 3: interventi auspicati

  Interventi che risolvono problemi di usabilità di minore gravità. Per esempio, problemi che si verificano in situazioni particolari e poco frequenti, e che comunque sono superabili con difficoltà limitate.

Il rapporto di valutazione

L'esito di una valutazione di usabilità dovrebbe essere descritto in modo accurato, non solo per test di tipo sommativo, ma anche nel caso dei test effettuati durante il processo iterativo di progettazione e sviluppo. Per questo, si utilizza un documento chiamato *rapporto di valutazione*. Esso non solo descrive i risultati dei test effettuati, ma fornisce anche evidenza del fatto che essi siano stati condotti con metodi adeguati. In particolare, che gli utenti utilizzati nelle prove siano rappresentativi delle categorie identificate nei requisiti e che siano in numero adeguato, che i test effettuati siano sufficienti per fornire indicazioni attendibili nei vari casi e contesti d'uso e, infine, che siano stati usati metodi appropriati sia nella conduzione del test sia nella raccolta dei dati.

Lo standard ISO 13407 identifica tre tipi fondamentali di rapporti di valutazione, a seconda che il loro scopo sia fornire

feedback per la progettazione, o provare la conformità a specifici standard oppure, infine, fornire evidenza del raggiungimento di obiettivi human-centred, per esempio in termini di usabilità o salute o sicurezza per l'utente.
Uno schema generale per la stesura del rapporto di valutazione, un po' più semplice di quello suggerito dall'ISO 13407, può essere il seguente.

Identificazione del documento

Riportare i nomi degli autori, la data e la versione del documento.

Sommario

Riportare una sintesi dello scopo del documento e delle sue conclusioni.

Prodotto valutato

Descrivere brevemente il prodotto o il prototipo sottoposto a test, con ogni informazione che lo identifichi con precisione. Indicare le aree funzionali sottoposte al test.

Obiettivi della valutazione

Descrivere gli obiettivi specifici perseguiti nella valutazione descritta nel documento.

Metodologia utilizzata

Specificare quanti utenti hanno partecipato al test, il loro livello di esperienza e le loro caratteristiche in relazione al prodotto in esame. Specificare i compiti o gli scenari assegnati, il contesto in cui si è svolto il test e la strumentazione utilizzata. Descrivere come è stato condotto il test e da chi, quanto tempo è durato (per ciascun utente e complessivamente), quali misure sono state raccolte, il ruolo degli osservatori e come sono stati analizzati i risultati.

Sintesi delle misure

Fornire una tabella di sintesi delle misure raccolte. Per esempio, i tempi di esecuzione e la percentuale dei vari compiti che sono stati portati a termine con successo, complessivamente e per ciascun utente. Aggiungere commenti ove opportuno.

Analisi dei risultati

Descrivere analiticamente i problemi incontrati da ciascun utente durante il test, compito per compito, allegando ove opportuno degli screen shot significativi e assegnando ad ogni problema un livello di gravità. Ogni problema sarà numerato, per un più facile riferimento. Descrivere in dettaglio, se significativi, reazioni e commenti degli utenti, registrati durante le prove. Questa è la sezione principale del documento, e dovrà contenere tutte le informazioni utili a formulare i possibili interventi per rimuovere i problemi descritti, senza che sia necessario tornare a esaminare il prodotto. Consisterà in genere di molte pagine.

Sintesi delle interviste agli utenti

Sintetizzare i risultati delle interviste effettuate a ciascun utente dopo l'esecuzione del test.

Raccomandazioni

Inserire la descrizione analitica degli interventi migliorativi proposti, raggruppati per livelli di priorità (per esempio: priorità 1: interventi indispensabili e urgenti; priorità 2: interventi necessari ma meno urgenti; priorità 3: interventi auspicati). Per ogni intervento proposto si farà riferimento al problema relativo, descritto nella sezione precedente. Gli interventi saranno numerati, per una facile tracciabilità.

Allegati

Allegare i moduli anagrafici compilati dagli utenti, la descrizione dei compiti/scenari data agli utenti prima del test, e tutti i questionari compilati nelle interviste finali. Allegare anche il materiale rilevante prodotto durante il test (per esempio, le riprese video).

Test di usabilità: costi e benefici

Qualunque sia la tecnica utilizzata, i test con gli utenti sono indispensabili. Infatti, le cause delle difficoltà incontrate dagli utenti possono essere moltissime. Analizzare un sistema "a tavolino", come nelle valutazioni euristiche, anche se può permetterci d'individuare numerosi difetti, non è mai sufficiente. I problemi possono essere nascosti e verificarsi soltanto con certi utenti, in relazione alla loro esperienza o formazione. Cose ovvie per chi già conosce il sistema o sistemi analoghi possono rivelarsi difficoltà insormontabili per utenti meno esperti. Un test di usabilità ben condotto mette subito in evidenza queste difficoltà. La necessità del coinvolgimento degli utenti è affermata con chiarezza dalla stessa ISO 13407:

> *La valutazione condotta soltanto da esperti, senza il coinvolgimento degli utenti, può essere veloce ed economica, e permettere di identificare i problemi maggiori, ma non basta a garantire il successo di un sistema interattivo. La valutazione basata sul coinvolgimento degli utenti permette di ottenere utili indicazioni in ogni fase della progettazione. Nelle fasi iniziali, gli utenti possono essere coinvolti nella valutazione di scenari d'uso, semplici mock-up cartacei o prototipi parziali. Quando le soluzioni di progetto sono più sviluppate, le valutazioni che coinvolgono l'utente si basano su versioni del sistema progressivamente più complete e concrete. Può anche essere utile una valutazione cooperativa, in cui il valutatore discute con l'utente i problemi rilevati.*

In Italia i test di usabilità sono ancora poco praticati. I motivi principali sono due. Il primo è senz'altro costituito dall'insufficiente diffusione di una cultura dell'usabilità, sia presso gli utenti sia presso gli stessi progettisti La sensibilità verso questi problemi è tuttora molto bassa, e gli esperti di usabilità, nelle università e nelle aziende, sono pochi. Il secondo motivo è che – si sostiene – i test di usabilità costano troppo. Si tratta di una convinzione diffusa ma sbagliata: i test di usabilità si possono fare rapidamente e con costi veramente molto contenuti. Un test di usabilità ben strutturato, di tipo sommativo, può coinvolgere 10-15 utenti, o più. Ma, come abbiamo visto, i test di tipo formativo possono essere fatti in modo molto più snello, con ottimi risultati.

Esiste anche un terzo motivo che a volte viene addotto per non fare test di usabilità. Si sostiene, in sostanza, che i test di usabilità non ci danno dei risultati oggettivi, ma ci segnalano soltanto le risposte soggettive di determinati individui di fronte al sistema. Questa è la tipica reazione di autodifesa dei progettisti: la "colpa" dei problemi non è nel sistema, è di quel particolare utente, che non è capace di usarlo come dovrebbe. Altri utenti, più in gamba, non incontrerebbero difficoltà. Il ragionamento è insidioso, perché, apparentemente, difendibile. Più o meno, è questo: test "scientifici", con risultati statisticamente validi, dovrebbero coinvolgere moltissimi utenti: almeno molte diecine. Questo non si può fare, sarebbe troppo lungo e costoso. I test con pochi utenti non sono significativi: le persone sono troppo diverse l'una dall'altra. Perché dovremmo dar peso alle reazioni soggettive di pochi individui e avviare costose modifiche soltanto sulla base di queste reazioni?

Il fondamento di queste obiezioni è formalmente inappuntabile: un esperimento o è condotto con il necessario rigore, o è inutile: non permette di trarre alcuna conclusione valida. Ma dal punto di vista pratico non regge: un test di usabilità – a meno che non sia condotto su una popolazione vasta di utenti e con metodi statistici rigorosi, il che non succede praticamente mai – non è un esperimento scientifico, fatto per confermare determinate ipotesi. Il suo scopo è di verificare le reazioni di certi soggetti a determinati stimoli. Queste reazioni sono un fatto oggettivo, si possono vedere e registrare con la telecamera. Anche le reazioni di pochi individui ci possono insegnare qualcosa, *se opportunamente decodificate e interpretate*. Ed è soprattutto quest'analisi e interpretazione ciò che interessa, e che dà valore al test. Essa ci fornisce una comprensione migliore del nostro sistema, e di come può essere usato. Dai test di usabilità possiamo scoprire aspetti che abbiamo trascurato nella progettazione e che possiamo migliorare.

Peraltro, in un tipico test di usabilità, molto spesso il conduttore non ha bisogno, per così dire, di "leggere fra le righe". In genere, quando ci sono dei problemi, le reazioni degli utenti sono evidenti, a volte addirittura scomposte, e di significato inequivocabile. Per capire perché è necessario fare test di usabilità dobbiamo vederne qualcuno. Leggerne su

un report scritto può non bastare a convincerci. Ma altra cosa è vedere con i nostri occhi una persona in carne e ossa, che ha accettato di sottoporsi al test, e che si mostra gentile, disponibile, interessata e volonterosa e che, dopo diversi tentativi non riesce a portare a termine un compito, e allora si fa rossa in viso, balbetta frasi incoerenti, e poi abbandona sbattendo, con un gesto di stizza, il mouse contro il tavolo… Queste reazioni, nella loro specificità certamente soggettive e individuali, costituiscono comunque un dato oggettivo, che non possiamo trascurare. Le difficoltà macroscopiche emergono subito, anche con pochi utenti, e questo è il senso della "regola di Nielsen". Diverso è il caso dei problemi minori, in cui le differenze di esperienza fra i vari utenti possono contare molto. In questi casi possono essere necessari molti test e molti soggetti e, soprattutto, una buona esperienza e capacità di analisi da parte degli osservatori.

In conclusione, i test di usabilità sono parte necessaria e ineliminabile del processo di progettazione e sviluppo di un sistema interattivo. L'usabilità non è un *optional* che si possa eliminare per abbassare i costi, come gli accessori in un'auto economica. Così come non si possono eliminare i test per verificare il corretto funzionamento del software. Molto semplicemente, se il prodotto è poco usabile, o non funziona, gli utenti non lo useranno.

Altre tecniche di valutazione

In questo capitolo, coerentemente con la natura introduttiva di questo libro, ci siamo limitati a descrivere le due tecniche principali di valutazione dell'usabilità di un sistema interattivo: le valutazioni euristiche e i test di usabilità con gli utenti. Sono due tecniche di carattere generale, che hanno lo scopo di individuare i principali problemi di usabilità di un sistema. Le abbiamo descritte con finalità essenzialmente pratiche, enfatizzando le tecniche che ci permettono di raggiungere risultati utili in fretta e a costi limitati (discount usability). Indagini su aspetti più specifici andranno condotte con tecniche apposite, per esempio effettuando studi sul campo, oppure indagini mirate attraverso interviste o focus group, oppure ancora esperimenti di laboratorio con utenti scelti in modo statisticamente rappresentativo.

In particolare, la conduzione di esperimenti di laboratorio condotti con metodologie rigorose su campioni rappresentativi di utenti costituisce oggi uno strumento importante per l'avanzamento delle conoscenze sull'usabilità e sui principi della Human-Computer Interaction in generale. Possiamo affermare che la HCI possiede due anime, molto diverse ma ugualmente importanti e complementari: un'anima progettuale e un'anima sperimentale. La prima ci porta a concepire, progettare e realizzare soluzioni e strumenti nuovi; la seconda ci permette di verificarne, sperimentalmente, la validità. È quindi comprensibile che una parte rilevante degli studiosi di questa disciplina si occupi della conduzione di esperimenti condotti con tecniche scientifiche rigorose. Queste richiedono competenze specifiche, anche di statistica, che esulano dagli scopi di questo libro, e per le quali rimandiamo ai testi specializzati.

Ripasso ed esercizi

1. Descrivi sinteticamente il metodo delle ispezioni per valutare l'usabilità di un sistema, specificandone vantaggi e svantaggi.
2. Che cosa s'intende per valutazione basata su euristiche? Quali sono i vantaggi e gli svantaggi?
3. Descrivi sinteticamente il metodo di ispezione dell'usabilità di un sistema basato sulle euristiche di Nielsen.
4. Applicando il metodo di valutazione basato sulle euristiche di Nielsen, come valuteresti l'usabilità dei comandi (esterni e interni) dell'ascensore di casa tua, e perché?
5. Qual è la differenza fra i test di compito e di scenario? E fra test formativi e sommativi?
6. Che cosa è la tecnica del "thinking aloud"? Perché la si usa spesso?
7. Riassumi e motiva la regola di Nielsen.
8. Devi organizzare un test di usabilità di un nuovo modello di penna stilografica. Come lo organizzi e lo conduci?
9. Quali misure si raccolgono normalmente durante un test di usabilità? A che cosa servono?
10. Che cosa si intende per "discount usability"?

Approfondimenti e ricerche

1. Cerca su http://www.youtube.com esempi di registrazioni di test di usabilità di vario tipo (prototipi di carta, dispositivi mobili, applicazioni per personal computer). Puoi iniziare con i video prodotti dagli studenti che hanno usato questo libro, cercando le seguenti parole chiave: "test usabilità", "usability test", "polillo".

2. Per una interessante discussione retrospettiva sulla discount usability, si veda la Alertbox del settembre 2009 di Jakob Nielsen: *Discount Usability: 20 years* del settembre 2009, sul suo sito http://www.useit.com/alertbox/discount-usability.html.

3. Ricerca in rete un software gratuito per gestire i test di usabilità, e sperimentalo. Per esempio, puoi utilizzare Morae (http://www.techsmith.com), uno dei prodotti più noti. Non è gratuito, ma permette prove gratuite per un periodo limitato.

Postfazione

Questo libro ha presentato una visione introduttiva della disciplina della ingegneria della usabilità. La trattazione è stata indipendente dalla specifica natura dei sistemi da progettare, per i quali si è fatta la sola ipotesi che si tratti di sistemi che interagiscano in modo sostanziale con delle persone (escludendo, quindi, quei sistemi che controllano altri sistemi, senza significativi interventi umani), e ai quali sia richiesto un elevato grado di usabilità. Quanto detto si può applicare, per esempio, alla progettazione di sistemi informativi, di apparati di controllo di apparecchiature critiche, di apparati per uso personale, di prodotti dell'elettronica di consumo, di sistemi multi-utente di varia natura, e così via.

In questo ambito, si è osservato che, negli ultimi due decenni, le discipline tradizionali della progettazione hanno subìto un completo *cambio di paradigma*: da una visione sistema-centrica delle attività e dei processi coinvolti, a una visione fortemente utente-centrica, per la quale l'oggetto della progettazione non sono più le sole funzionalità del sistema (system design), ma anche, e in primo luogo, le modalità di interazione fra il sistema e i suoi utilizzatori (interaction design).

Questo cambio di paradigma ha profonde implicazioni su tutte le tematiche connesse alla progettazione, e in particolare sui modelli del processo di progettazione e sviluppo, sulla composizione dei team di progetto, e sulla formazione stessa dei progettisti:

- i processi di progettazione e sviluppo, qualunque siano i contesti organizzativi, le metodologie, gli strumenti e gli standard adottati, devono necessariamente essere di tipo iterativo, per inserire l'utente – e soprattutto le prove d'uso del sistema - lungo l'intero processo. *Le prove d'uso diventano una componente della attività di progettazione*;
- i team di sviluppo devono essere necessariamente multi-disciplinari, per fronteggiare la complessità e la articolazione dei problemi posti dalla forte centralità dell'utente, con tutte le problematiche connesse (ergonomiche, psicologiche, sociali);
- infine, la formazione dei progettisti – tradizionalmente di orientamento esclusivamente tecnico – deve ampliare i propri orizzonti. Un team multi-disciplinare è composto di persone con professionalità, culture, linguaggi, valori e priorità diverse, che devono riuscire a comunicare in modo armonico, nel rispetto dei contributi specifici al progetto complessivo.

Anche se questo cambio di paradigma è stato proposto quasi un quarto di secolo fa, nella quotidiana pratica progettuale molta strada deve ancora essere percorsa per una sua adozione matura e consapevole, soprattutto nel nostro Paese. Questo libro si è proposto di contribuire a tale evoluzione, fornendo del materiale didattico per corsi universitari di introduzione e sensibilizzazione a questi argomenti, soprattutto (ma non solo) diretti a studenti di orientamento informatico.

Nello scrivere questo libro non mi sono posto l'obiettivo di formare degli specialisti di usabilità. Le molte facce di questo mestiere non consentono di riassumerne i molteplici aspetti di questa professione in così poche pagine. Né consentono di adottare il solo punto di vista dell'informatico, che inevitabilmente le caratterizza, data la mia formazione. Mi sono, invece, proposto l'obiettivo, molto meno ambizioso ma che ritengo assai importante, di suggerire ai progettisti di formazione tecnica la utilità di un approccio più ampio alle proprie attività, che riconosca che i sistemi da loro progettati si rivolgono, innanzitutto, a degli utilizzatori umani.

Certamente, non penso che ogni progettista di software – o di sistemi ad alto contenuto di software - debba trasformarsi in un esperto di usabilità: sono professioni diverse, ed è giusto che ciascuno tenti di fare al meglio il proprio mestiere. Ma ritengo indispensabile che ogni progettista di software comprenda che il suo lavoro ha implicazioni ben più ampie della sola risoluzione di problemi tecnici. La formazione tecnica tende inevitabilmente alla specializzazione, uno strumento potente, ma molto pericoloso. Il rischio è che lo specialista interagisca solo con altri specialisti della sua

disciplina, e perda, a poco a poco, la capacità di comunicare con quelli di altre discipline – e di comprenderli. Il fenomeno è chiaramente percepibile solo da chi transiti attraverso contesti molto diversi. Personalmente, avendo frequentato ambienti professionali e didattici molto differenti, ho sperimentato di persona la rilevanza del fenomeno. Nell'attività didattica, mi sono trovato, dopo un *imprinting* dato da decenni di insegnamento nei corsi di laurea in Informatica, a tenere alcuni corsi per studenti di Disegno Industriale e di Scienze della Comunicazione. L'impressione iniziale di incomprensione (reciproca) è stata fortissima. Non si tratta solo di differenti basi tecniche, ma di differenti linguaggi e, soprattutto, valori. Infatti, ogni disciplina ha i suoi valori e i suoi miti. Questi costituiscono il tessuto connettivo comune che permette ai membri del gruppo di riconoscersi e di comunicare. Quando si perde questo tessuto comune, la collaborazione si fa molto più difficile.

Nella mia attività professionale, nel periodo di grande crescita del Web alla fine degli anni '90, mi sono trovato a gestire la fusione di due aziende che realizzavano siti importanti. Una era di estrazione informatica, con una forte cultura di ingegneria del software, l'altra proveniva dall'area della comunicazione di marketing e della creatività. Entrambe erano leader di mercato. In teoria, sembrava la perfetta combinazione delle competenze necessarie a progettare siti web di grande qualità. Ma non funzionò. Le due comunità erano troppo diverse, e non furono in grado di intendersi e di comunicare. La innegabile leadership, riconosciuta, di ciascun gruppo ne rafforzava valori e convinzioni, e impediva di comprendere valori e convinzioni dell'altro gruppo, altrettanto validi ma sostanzialmente differenti.

Da allora, mi sono sempre dedicato al tentativo di favorire la crescita di una cultura multi- e inter-disciplinare nella scuola.

Quando la sera è tersa, osservo il cielo.
Non finisco mai di stupirmi,
tanti punti di vista ci sono lassù
- mi ha risposto.

W. Szymborska
da Il vecchio professore, in Due Punti, ed.Adelphi

Per approfondire

Le sezioni "Approfondimenti e ricerche" in coda ad ogni capitolo contengono suggerimenti e riferimenti per approfondire gli argomenti trattati, di solito utilizzando materiale disponibile liberamente in rete. Qui, mi limiterò a indicare alcuni libri e siti web che ritengo utili sui temi di questo libro. Ho fatto una drastica selezione: le bibliografie che riportano centinaia di riferimenti dimostrano l'ampiezza della cultura dell'autore, ma servono poco a chi cerca un orientamento per proseguire nello studio. I criteri di scelta e di classificazione sono personali: aggiungerò, per ogni titolo, una breve motivazione. Non citerò articoli: quelli riferiti nel testo sono indicati nelle note dei vari capitoli, e non sono riportati qui.

Letture suggerite a complemento di questo testo

I libri seguenti, tutti di facile e piacevole lettura, possono costituire degli utili complementi al presente libro.

- D.A.Norman, *The psychology of everyday things*, Basic Books Inc., 1988. Edizione italiana: *La caffettiera del masochista – Psicopatologia degli oggetti quotidiani*, Gruppo Editoriale Giunti, 1990 e successive riedizioni.

  Un piccolo grande classico, un libro importante travestito da libro leggero. Da leggere assolutamente, in inglese o in italiano.
- A.Soro (ed.), *Human Computer Interaction – Fondamenti e prospettive*, Polimetrica Publisher, 2008

  Una collezione di articoli di inquadramento ai principali temi dell'HCI, scritti da autori italiani, a scopo didattico. Può costituire un utile complemento didattico al presente libro, su temi specifici. È anche disponibile, gratuitamente, in rete sul sito dell'editore www.polimetrica.com.
- D.A.Norman, *Emotional Design*, Basic Books Inc., New York, 2004. Traduzione italiana: *Emotional Design – Perchè amiamo (o odiamo) gli oggetti della vita quotidiana*, Apogeo, 2004.

  Quindici anni dopo *The psychology of everyday things*, Donald Norman aggiusta il tiro, e rivaluta gli aspetti emozionali del design. Facile e divertente, da leggere.
- S.Krug, *Don't Make Me Think – A Common Sense Approach to Web Usability* (Seconda edizione), Ed. italiana Tecniche Nuove - HOPS, 2001.

  La letteratura sulla usabilità dei siti web è vastissima. Questo libro, che ha avuto un notevole successo di mercato, può essere usato come una introduzione elementare all'argomento. È un testo snello e con taglio pratico, che si legge in poche ore, ma molto efficace. Come recita il sottotitolo, "un approccio di buon senso all'usabilità Web".

Introduzioni generali all'interaction design e all'ingegneria della usabilità

I seguenti libri costituiscono delle introduzioni ampie e approfondite ai temi classici dell'interaction design e della ingegneria della usabilità. Trattano sostanzialmente gli stessi argomenti, sia pure con taglio molto diverso.

- J.Preece, Y.Rogers, H.Sharp, *Interaction Design – Beyond Human Computer Interaction* (Second Edition), John Wiley & Sons, 2007. La prima edizione è stata pubblicata in Italia, con lo stesso titolo, da Apogeo nel 2004.

  Un libro di quasi 800 pagine sull'interaction design, molto utilizzato come libro di testo nelle università. Le tre autrici hanno background diversi, dalle scienze cognitive al software engineering.
- M.B.Rosson, J.M.Carroll, *Usability Engineering – Scenario-based Development of Human-Computer Interaction*, Morgan Kaufmann Publishers, 2002.

  Un altro libro di testo di livello universitario, orientato a studenti di informatica. Il libro, scritto da due docenti di computer science, ha una forte enfasi progettuale.
- S.Lauesen, *User Interface Design – A Software Engineering Perspective,* Pearson Addison-Wesley, 2005.

  Anche questo libro è orientato agli studenti universitari. È forse il testo più informatico fra quelli citati.
- J.Nielsen, *Usability Engineering*, Academic Press, 1993.

  Un testo classico, piuttosto snello e di tipo introduttivo, diretto ai progettisti. Ha avuto il grande merito di

promuovere l'ingegneria della usabilità presso un vasto pubblico, negli anni '90. È stato scritto molti anni fa, ma vale la pena darci un'occhiata.

- A.Dix, J.Finlay, G. Abowd, R.Beale, *Human-Computer Interaction* (Terza edizione), Pearson-Prentice Hall, 2004. Traduzione italiana (in versione ridotta): *Interazione uomo macchina*, McGraw-Hill, 2004.

  Anche questo libro è stato molto usato come libro di testo universitario. È una introduzione generale ai temi della interazione uomo macchina, e quindi tratta un insieme di temi più ampio dei precedenti. La versione italiana è ridotta rispetto all'originale inglese, ma resta di circa 650 pagine.
- J.A.Jacko, A.Sears (ed.), *The Human-Computer Interaction Handbook – Fundamentals, Evolving Technologies and Emerging Applications*, Lawrence Erlbaum Associates, 2003.

  Una grande (quasi 1300 pagine!) raccolta di articoli di rassegna sui principali temi della HCI. Anche se gli articoli non sono recentissimi, costituiscono un utilissimo punto di partenza (anche di tipo bibliografico) per approfondire lo studio di specifici argomenti.
- R.W.Proctor, K-P.L.Vu (ed.), *Handbook of Human Factors in Web Design*, Lawrence Erlbaum Associates, 2005. Una raccolta, simile alla precedente, di articoli di rassegna sui principali temi correlati all'ingegneria dell'usabilità dei siti web. Molto utile anche per i riferimenti bibliografici.

Introduzioni alla psicologia

I seguenti libri possono essere utili sia come inquadramento generale alle problematiche della psicologia, sia come primo livello di approfondimenti sugli argomenti di natura psicologica, che sono stati appena accennati in questo libro.

- P.Gray, *Psicologia* (Seconda edizione italiana), Zanichelli, 2004.

  Un manuale universitario molto noto, che fornisce un'ampia introduzione alla psicologia generale. È un testo molto chiaro, che può essere utile, per esempio, per un primo approfondimento sulla percezione e la memoria umana.
- G.Robinson-Riegler, B.Robinson-Riegler, *Cognitive Psychology – Applying the Science of Mind*, Pearson, 2004.

  Un manuale universitario di psicologia cognitiva, piuttosto completo.

Libri di approfondimento su temi specifici

I seguenti libri possono servire per approfondire temi specifici introdotti in questo libro.

- J R.Bias, D.J.Mayhew, *Cost Justifying Usability - Second edition: An Update for the Internet Age*, Morgan Kaufmann, 2005.

  La nuova edizione di un testo classico (del 1994) sulla valutazione del rapporto costi/benefici dell'usabilità.
- I.Sommerville, *Ingegneria del software* (Ottava edizione), Pearson Addison-Wesley, 2007.

  Un testo classico, molto utilizzato nelle università, per chi desidera un inquadramento delle problematiche dell'ingegneria del software. Non è orientato all'ingegneria dell'usabilità, ma fornisce un quadro molto completo dei metodi utilizzati nella produzione di software di ogni tipo.
- M.Fowler, *UML distilled* (Terza edizione). Ed. italiana: Pearson – Addison Wesley, 2004.

  È un testo molto breve che costituisce una ottima introduzione ai costrutti più usati del linguaggio UML. È probabilmente il libro più diffuso su questo argomento. Assume che il lettore abbia una conoscenza di base dei concetti di programmazione orientata agli oggetti.
- A.Cockburn, *Writing Effective Use Cases*, Addison-Wesley, 2000.

  Un libro molto ben fatto su come identificare e descrivere i casi d'uso. Corredato di numerosi esempi tratti da progetti reali, è una utile guida pratica per principianti e per esperti.
- I.Horrocks, *Constructing the user interface with statecharts*, Addison Wesley, 1999.

  Un libro sull'uso degli statechart per la descrizione delle interfacce utente.
- M.Pezzè, M.Young , *Software Testing and Analysis – Process, Principles and Techniques*, John Wiley, 2007.

  Un libro completo, di livello universitario, sul test del software.
- C.M.Barnum, *Usability Testing and Research*, Allyn & Bacon / Longman Publishers, 2002.

  Un libro interamente dedicato ai test di usabilità. La trattazione è generale, e non specificamente dedicata a particolari applicazioni.

- C.Snyder, *Paper prototyping – The fast and easy way to design and refine user interfaces*, Morgan Kaufmann Publishers, 2003.

  Un libro di facile lettura sull'uso dei prototipi di carta per la messa a punto dei sistemi interattivi.
- J. Nielsen, *Designing Web Usability*, New Riders Publishing, 2000. Edizione italiana: *Web Usability*, Apogeo, 2004.

  Il libro classico di Jaklob Nielsen sulla usabilità dei siti web.
- J.Nielsen, H.Loranger, *Prioritizing Web Usability*, New Riders Book, 2007. Edizione italiana: *Web Usability 2.0 – L'usabilità che conta*, Apogeo, 2006.

  Nonostante il titolo della edizione italiana, non si tratta di una revisione del libro precedente, ma di un libro completamente nuovo.
- C.Alexander, *The Timeless Way of Building*, Oxford University Press, 1979, e C.Alexander, S.Ishikawa, M.Silverstein, *A Pattern Language*, Oxford University Press, 1977.

  Due libri affascinanti, che hanno introdotto il concetto di pattern language. Non si riferiscono all'informatica, ma all'architettura. Il primo spiega filosofia e motivazioni, il secondo contiene l'intero catalogo dei pattern proposti. Una lettura molto stimolante, non solo per gli architetti.
- C.Crumlish, E.Malone, *Designing Social Interfaces*, O'Reilly, 2009.

  Un libro dei curatori della Yahoo! Pattern Library in rete, che raccoglie più di 100 design pattern utilizzabili nella progettazione di siti e applicazioni web "sociali".
- C.Ware, *Visual thinking*, Morgan Kaufmann, 2008.

  Un libro molto interessante, rigoroso e ricco di esempi, sui rapporti fra i meccanismi della percezione visiva e il visual design.
- R.Lesina, *Il nuovo manuale di stile – Guida alla redazione di documenti, relazioni, articoli, manuali, tesi di laurea* (Seconda edizione), Zanichelli, 1994.

  Un manuale completo destinato a chiunque debba scrivere o revisionare testi non letterari: relazioni, monografie, saggi, tesi di laurea. Suggerisce come trattare titoli e paragrafi, nomi, numeri e simboli, illustrazioni e tabelle, riferimenti e note. Un classico usato da generazioni di redattori professionisti.
- L.Carrada, *Scrivere per Internet*, Lupetti, 2000

  Un ottimo libro, per chi vuole sapere in fretta tutto quello che serve sulla scrittura per il Web. Scorrevole e sintetico.
- M.Grasso, *Scrivere per il Web*, Franco Angeli, 2002

  Contiene anche una sintesi delle regole editoriali più utili: quando mettere gli apostrofi, quando usare corsivi e virgolette, e così via. Snello e molto utile.
- W.B.Arthur, *The Nature of Technology – What it is and how it evolves*, Free Press, 2009.

  Un libro affascinante, di taglio filosofico, sulla natura e i processi dell'innovazione tecnologica, scritto da un grande studioso di economia.
- R.Polillo, *Plasmare il Web – Road map per siti di qualità*, Apogeo, 2006.

  Il libro descrive un processo iterativo per la progettazione e lo sviluppo di siti web, mediante prototipi successivi. È un esempio di applicazione dell'approccio descritto in questo libro.

Siti web che bisogna conoscere

- http://www.useit.com

  Il sito di Jakob Nielsen, dedicato all'usabilità dei siti web. Pubblica una rubrica settimanale (*Alertbox*) dedicata all'approfondimento di specifici argomenti relativi all'usabilità. Un sito che da anni fa opinione nel mondo del Web.
- http://www.jnd.org

  Il sito di Donald Norman, ricco di articoli, interviste, link ed altro sul buon design.
- http://www.wikipedia.org

  Le voci di Wikipedia relative alla ingegneria della usabilità sono moltissime, e di varia qualità. I testi a volte lasciano piuttosto a desiderare, ma sono comunque un riferimento importante, soprattutto per i numerosi link. Di solito, le voci in lingua inglese sono più complete di quelle italiane, che a volte ne sono una sintesi mal tradotta.
- http://www.iso.org

Il sito dell'International Standard Organization, da dove sono scaricabili tutti i documenti prodotti da questo Ente. Purtroppo, salvo rari casi, i documenti sono a pagamento, e molto costosi.

- http://www.publiaccesso.gov.it

  Il portale del CNIPA (Centro Nazionale per l'Informatica nella Pubblica Amministrazione) dedicato all'accessibilità informatica. Il sito raccoglie la normativa italiana in tema di accessibilità informatica, i documenti di approfondimento, manuali e testi di riferimento, studi e recensioni, prove di prodotti hardware e software ed esempi di siti accessibili.
- http://www.w3c.org/WAI

  Il sito della Web Accessibility Initiative del W3C, quindi il riferimento ufficiale del W3C per i problemi di accessibilità.
- http://www.webaccessibile.org

  Un sito italiano sull'accessibilità del Web, gestito da IWA (International Webmaster Association) Italy.

Il sito [dell'autore] di questo libro

- http://www.rpolillo.it

  Qui trovate tutti i link alla versione elettronica di questo libro, con tutto il materiale associato, ai miei corsi universitari e agli altri miei libri.

Appendice. Notazione per gli statechart

Introduzione

In questa appendice descriviamo un particolare tipo di diagrammi, che si rivela molto utile per descrivere il dialogo fra utente e sistema. Si tratta degli *statechart***,** inventati da David Harel nel 1987,[188] come strumento di modellazione di sistemi complessi, che abbiamo introdotto nel capitolo 9 (pag.189). Gli statechart, con il nome di *state machine diagrams*, sono stati in seguito adottati da UML (Unified Modeling Language), il linguaggio visuale standardizzato usato nell'ingegneria del software, soprattutto per la progettazione di sistemi orientati agli oggetti.[189]

UML comprende numerose notazioni differenti, che possono essere selezionate in funzione delle esigenze e delle preferenze del progettista. Anche per descrivere l'interazione fra utente e sistema, UML propone diverse alternative. La scelta degli statechart non è quindi l'unica possibile. La proponiamo perché ci sembra quella che concilia semplicità e potenza descrittiva. In questo libro seguiamo la notazione adottata da UML 2.

Gli statechart sono diagrammi a stati di tipo gerarchico. In altre parole, la loro caratteristica principale è quella di permettere di modellare un sistema descrivendone la sua evoluzione da uno stato all'altro, per livelli di astrazione successivi. Questo è molto utile per descrivere con relativa semplicità situazioni che altrimenti richiederebbero diagrammi molto complessi dal punto di vista grafico. Essi possono essere utilizzati per rappresentare sistemi di vario tipo, e non necessariamente per descrivere i dialoghi fra utente e sistema.

Nel seguito, ci limiteremo a descrivere i costrutti di uso più frequente in modo piuttosto informale, rimandando il lettore che desiderasse informazioni più complete ai testi specialistici.[190]

Stati

In un diagramma per macchine a stati, uno *stato* del sistema modellato è rappresentato da un rettangolo con gli angoli arrotondati, con il nome dello stato al suo interno:

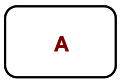


Figura 295. Rappresentazione grafica di uno stato di nome A

Transizioni

Una *transizione* è rappresentata da un segmento orientato che connette fra loro due stati. Essa indica il passaggio da uno stato all'altro.

Una transizione è contrassegnata da una etichetta composta da tre parti, ciascuna delle quali è opzionale:

evento [condizione] /attività

[188] D.Harel, *Statecharts: A visual formalism for complex systems*, in Science of Computer Programming, vol.8, n.10, pagg.231-274, 1987, reperibile anche in rete.

[189] Per una introduzione a UML, si veda, per esempio, M.Fowler, *UML distilled (Terza edizione)*. Ed. italiana: Pearson – Addison Wesley, 2004.

[190] Tutti i libri su UML hanno un capitolo sui diagrammi per macchine a stati. Tuttavia, questi diagrammi vengono trattati spesso in modo sommario. Il riferimento più completo è quello ufficiale, costituito dalla specifica dell'OMG: *Unified Modeling Language Superstructure – Version 2.0 (sezione 15)*, che si può trovare sul sito www.omg.org.

evento	(detto anche *trigger*) indica l'evento che innesca la transizione da uno stato all'altro;
pre-condizione	(detta anche *guard*) indica una condizione booleana che deve essere verificata affinché la transizione abbia luogo;
attività	indica una azione che deve essere eseguita dal sistema durante la transizione.

Così, nel seguente diagramma, quando il sistema si trova nello stato A, al verificarsi dell'evento e si passa allo stato B effettuando l'azione A, purchè sia verificata la condizione c.

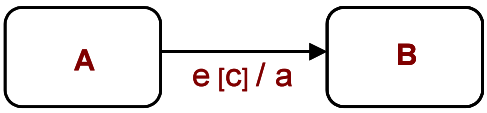


Figura 296. Una transizione da A a B

Quando il sistema si trova in un certo stato e accade un determinato evento, si può verificare una sola transizione uscente da quello stato. Di conseguenza, quando ci sono più transizioni etichettate con lo stesso evento, le condizioni associate devono essere mutuamente esclusive. Questa situazione si può rappresentare anche utilizzando un rombo, che simboleggia una scelta, come in Figura 297.



Figura 297. Il rombo rappresenta un'alternativa

Lo *stato iniziale* di un diagramma è quello indicato dalla (unica) transizione uscente da un pallino nero. Lo *stato finale* viene rappresentato invece da un pallino nero cerchiato.[191] Così, nel diagramma seguente, *A* è lo stato iniziale. Quando si verifica l'evento *e*, il sistema va nello stato finale.



Figura 298. La notazione per indicare stato iniziale e stato finale

Allo scopo di semplificare il diagramma, più transizioni possono essere connesse attraverso *giunzioni*:

[191] Il pallino nero non rappresenta uno stato del sistema. Pertanto, la freccia che da esso esce non può avere associati eventi, condizioni o azioni. Il pallino nero cerchiato, invece, è uno stato a tutti gli effetti. Pertanto, la freccia in esso entrante può avere associati eventi, condizioni o azioni.



Figura 299. Le giunzioni possono semplificare un diagramma

Il diagramma di Figura 300 descrive il funzionamento di una macchina erogatrice di bevande.

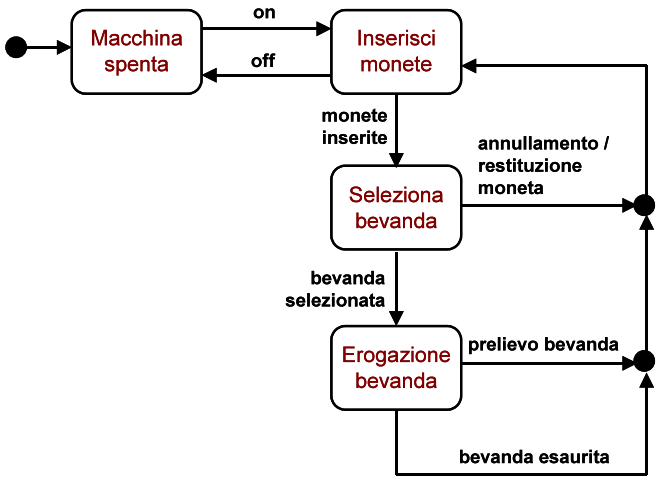


Figura 300. Diagramma a stati che modella il funzionamento di una macchina erogatrice di bevande

Transizioni interne

A volte può essere utile poter indicare che il sistema modellato effettua delle attività (eventualmente subordinate al verificarsi di certi eventi e condizioni) senza cambiare stato. Queste situazioni possono essere rappresentate scrivendo eventi, condizioni e attività all'interno dello stato, con la stessa notazione vista in precedenza: evento[condizione]/attività. In questo caso, il rettangolo arrotondato che rappresenta lo stato viene suddiviso in due sezioni, una per il nome e una per le altre informazioni.

Per esempio, in Figura 301 abbiamo raffinato l'esempio di Figura 300, indicando che l'erogatrice, nello stato Seleziona bevanda, è in grado di segnalare all'utente quali bevande sono esaurite. In questo esempio abbiamo anche utilizzato gli eventi speciali entry ed exit. Il primo si verifica automaticamente quando il sistema entra nello stato, e causa l'esecuzione dell'attività specificata (nel nostro caso: display "Seleziona bevanda"). Il secondo si verifica automaticamente immediatamente prima che il sistema esca dallo stato. L'attività specificata (nel nostro caso: display "Grazie!") verrà eseguita dopo ogni altra attività associata allo stato.



Figura 301. Uno stato con transizioni interne

Le attività interne permettono di modellare situazioni complesse senza che ciò comporti necessariamente una eccessiva proliferazione di stati nel diagramma.

Stati composti

Uno *stato composto* (o *superstato*) è uno stato che può essere decomposto in una o più *regioni*, ciascuna delle quali può contenere altri stati (detti *sottostati*). Consideriamo, per il momento, stati composti da una sola regione.

La Figura 302 mostra uno stato *A* contenente una sola regione, la quale contiene una macchina a stati che specifica, ad un livello d'astrazione inferiore, il "comportamento interno" di A.

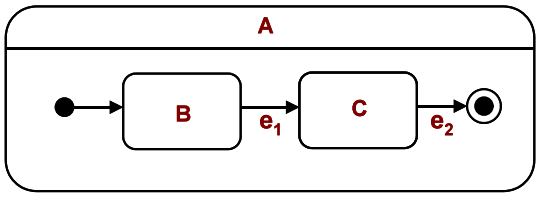


Figura 302. A è uno stato composto, B e C sono suoi sottostati.

Quando il sistema entra nello stato composto A, esso entra nel sottostato B (che è lo stato iniziale del diagramma interno), per poi transitare nel sottostato C, e quindi terminare. Se A e B hanno attività associate agli eventi speciali entry, esse vengono eseguite (prima quella associata ad A, e subito dopo quella associata a B). Analogamente per gli eventi speciali exit associati a C e ad A: all'uscita, verrà prima eseguita l'azione relativa a C, e quindi quella relativa ad A.

Per non complicare troppo i diagrammi, i sottostati di uno stato composto possono venire specificati a parte. In questo caso, per indicare che uno stato è composto, si usa la notazione seguente:

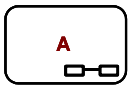


Figura 303. Il simbolo che indica che A è uno stato composto

Vediamo un esempio di stato composto. Nel diagramma di Figura 300 possiamo dettagliare a parte lo stato Inserisci monete, come inFigura 304. In tale diagramma, abbiamo supposto che tutte le bevande abbiano lo stesso prezzo e che lo stato Macchina spenta effettui, all'uscita, l'azzeramento di un contatore importo.[192]

[192] Questo può essere specificato associando allo stato Macchina spenta una transizione interna del tipo: exit / azzera importo.

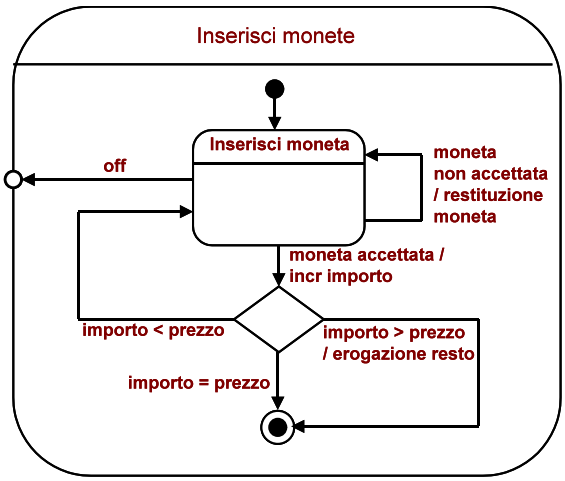


Figura 304. La descrizione dello stato composto Inserisci monete di Figura 108

Come si vede dall'esempio di Figura 305, il diagramma contenuto in uno stato composto non deve necessariamente possedere un solo punto di ingresso e un solo punto di uscita. Possiamo, infatti, definire altri *pseudo-stati di ingresso e di uscita*, usando la notazione indicata nella stessa figura.

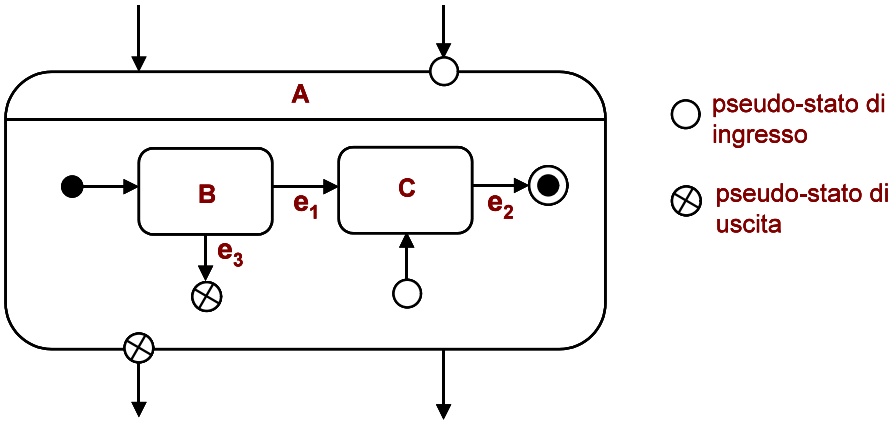


Figura 305. Pseudo-stati di ingresso e di uscita

Più pseudo-stati di ingresso e di uscita possono essere differenziati assegnando loro un nome. Il diagramma di Figura 305 può essere anche rappresentato come in Figura 306.

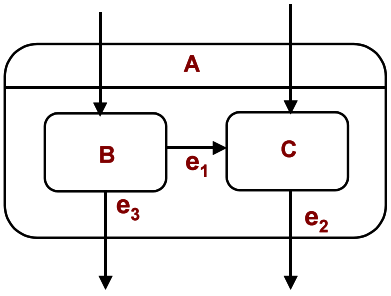


Figura 306. Una rappresentazione alternativa dello stato composto di Figura 305

Sottomacchine

Può essere necessario richiamare uno stesso diagramma da varie parti di un diagramma di più alto livello. In tal caso, si dice che il diagramma richiamato rappresenta una *sottomacchina* del chiamante. Per esempio, nella Figura 307 una sottomacchina `S` è richiamata da due stati diversi. La notazione `A:S` si può leggere "lo stato `A` è una istanza della sottomacchina `S`".

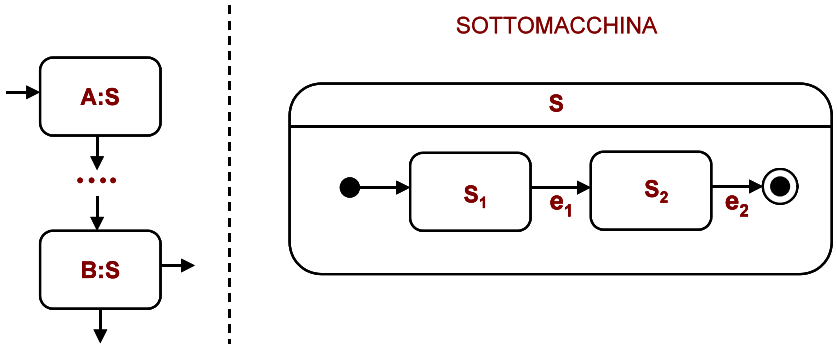


Figura 307. Una sottomacchina S richiamata da più punti di un diagramma a stati di più alto livello

Notazioni abbreviate

Per permettere di rappresentare in modo semplice anche situazioni complesse, senza dover disegnare nel diagramma troppe transizioni, si possono definire alcune notazioni abbreviate. Per esempio, possiamo rendere più compatto un diagramma in cui lo stesso evento e conduce a uno stesso stato a partire da più stati diversi, come in Figura 308.

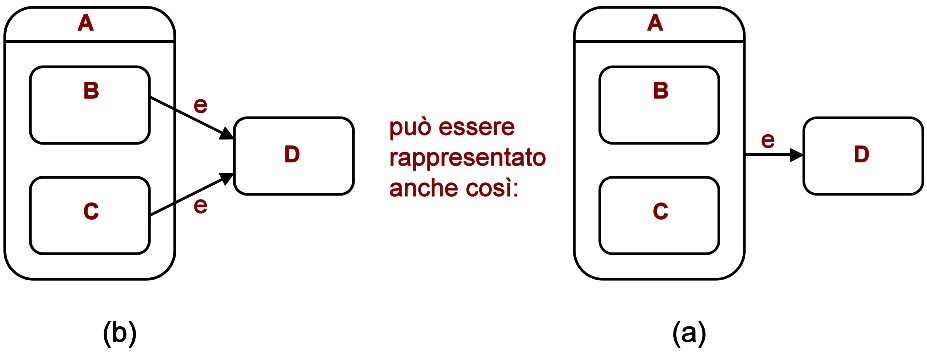


Figura 308. Il diagramma (a) può essere rappresentato, in modo graficamente più semplifice, come in (b)

È anche possibile rendere più compatti quei diagrammi in cui da uno stesso stato si raggiungono più stati diversi, come in Figura 309 (a). In questo caso, si introduce la notazione di Figura 309 (b), che utilizza uno pseudo-stato *selettore* `S`. L'evento `e` è definito (nella nota accanto al diagramma) come la disgiunzione degli eventi `e1` ed `e2`. Il selettore attiverà lo stato `A` quando `e=e1`, e lo stato `B` quando `e=e2`. Ovviamente, dovremo indicare con precisione la regola per associare ciascun evento a ciascuno stato di destinazione, come si è fatto in figura.

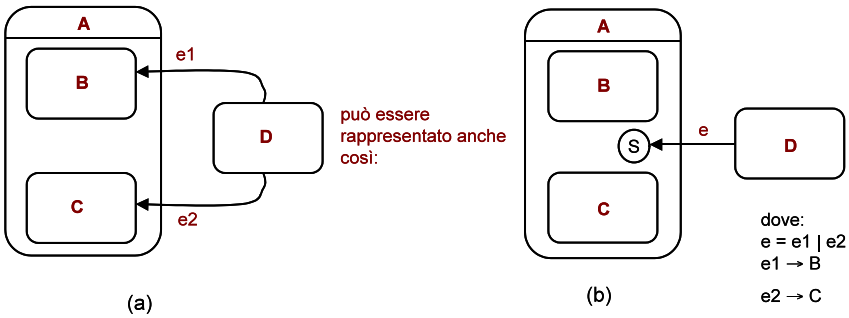


Figura 309. Semplificazione di un diagramma attraverso la introduzione di un selettore S

La notazione precedente può essere generalizzata al caso in cui la selezione di uno stato fra più stati alternativi dipende dal valore di un parametro p, come in Figura 310. In questo caso, l'evento e(p) causa la transizione allo stato D(p).

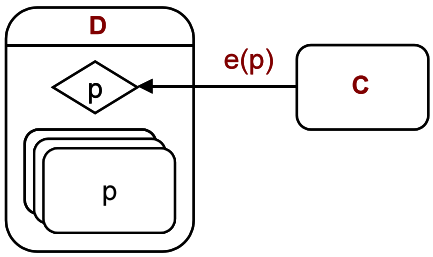


Figura 310. Attivazione di uno stato D(p) in funzione di un evento e(p), dipendente dal parametro p

Queste semplici convenzioni possono semplificare in modo significativo diagrammi complessi. Per esempio, il diagramma di Figura 311 (a) può essere rappresentato in forma compatta come in Figura 311 (b).

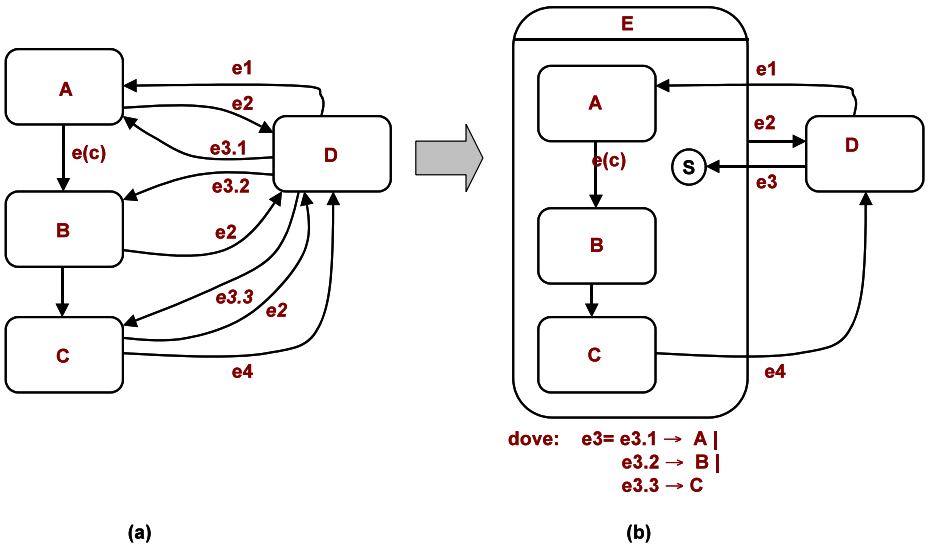


Figura 311. Il diagramma (a) può essere rappresentato in forma più compatta come in (b), introducendo lo stato composto E.

Sottostati concorrenti

Finora abbiamo considerato stati composti costituiti da una sola regione. Quando le regioni sono più di una, gli stati contenuti in una regione sono *concorrenti* a quelli delle altre. Pertanto, lo stato composto si potrà trovare contemporaneamente in più sottostati, ciascuno appartenente ad una sua diversa regione.

Le regioni di uno stato sono separate da linee tratteggiate, orizzontali o verticali, come nellaFigura 312, che mostra uno stato composto da due regioni.

Con riferimento a questa figura, quando si entra nello stato composto *A*, vengono attivati *contemporaneamente* gli stati iniziali di *tutte* le regioni (nel nostro caso, B e D). I diagrammi di ogni regione poi evolvono in parallelo.

Quando tutti i diagrammi raggiungono il loro stato finale, allora termina anche lo stato composto e, nel caso, viene eseguita l'azione associata al suo evento speciale exit).

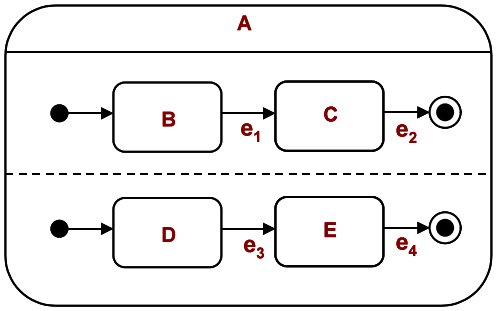


Figura 312. Uno stato composto A costituito da due regioni

La Figura 313 mostra un semplice esempio di diagramma con stati concorrenti. Esso rappresenta l'evoluzione della preparazione di un corso universitario. Per superare l'esame, occorre frequentare un laboratorio, realizzare un progetto di esame e sostenere con successo un compito scritto.

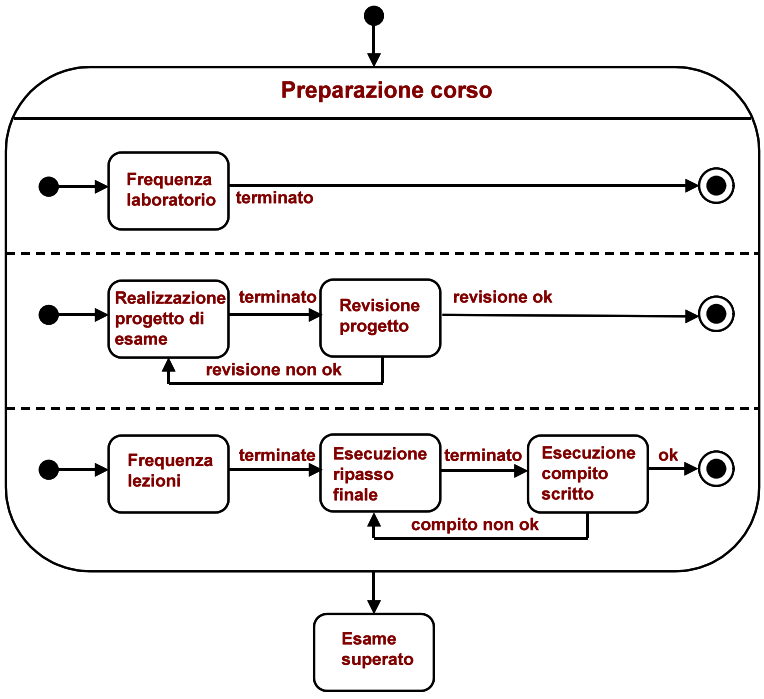


Figura 313. Macchina a stati che rappresenta la preparazione di un corso universitario

Indice analitico