

# INTERAZIONE UOMO-COMPUTER

## UN'INTRODUZIONE

L'introduzione di tecnologie informatiche in ogni area delle attività umane sta rendendo l'informatica sempre più una disciplina orientata a supportare la comunicazione con gli utenti. Negli ultimi venti anni si è assistito a una crescente necessità di capire come progettare l'interazione di sistemi informatici con gli utenti in modo da ottenere sistemi facili da usare. Questo articolo fornisce un'introduzione ai concetti che caratterizzano gli obiettivi, i metodi e gli strumenti della disciplina che affronta queste problematiche.

### 1. INTRODUZIONE

L'interazione uomo-computer (*Human-Computer Interaction*) è la disciplina che studia metodi e tecniche per la progettazione e lo sviluppo di sistemi interattivi che siano usabili, affidabili e che supportino e facilitino le attività umane. Essa si è sviluppata velocemente negli ultimi anni con la penetrazione crescente dei dispositivi informatici in ogni attività umana. Infatti, il sempre maggior uso di applicazioni informatiche richiede una progettazione che sappia tenere conto dei vari possibili contesti di uso, degli obiettivi degli utenti e delle nuove tecnologie di interazione. L'informatica diventa sempre più una disciplina interattiva e orientata alla comunicazione con utenti. Per esempio, il successo del Web è dovuto alla facilità con cui permette di comunicare informazioni al mondo intero. A questo scopo sono stati sviluppati metodi, modelli, tecniche, in grado di misurarsi con queste nuove problematiche. Dovendo trattare dell'interazione tra due sistemi molto diversi, quello umano e quello informatico, con quello umano particolarmente complesso, l'interazione

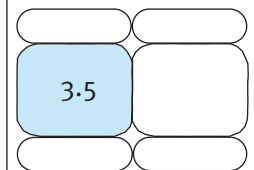
uomo-computer è intrinsecamente interdisciplinare per poter cogliere i vari aspetti che possono essere rilevanti. Lo scopo di questo articolo è di consentire un'agevole comprensione di quali sono i concetti di base, le caratteristiche e gli aspetti più significativi di questa area e gli approcci più promettenti. A questo scopo, si forniscono brevi cenni riguardanti le prime esperienze storiche più significative e una descrizione dell'area disciplinare in modo da capire meglio cosa sia e quali siano le questioni più importanti affrontate. Tra queste vi è certamente l'usabilità, un concetto complesso a molte dimensioni, dove il peso di ciascuna dimensione dipende anche dal tipo di dominio applicativo che si sta considerando.

### 2. CENNI STORICI

I primi calcolatori avevano poco di interattivo con gli utenti. La figura 1 mostra un esempio che risale alla fine degli anni '60 - inizi anni '70, con un operatore avente a disposizione un pannello di controllo alquanto essenziale che aveva, principalmente, funzioni di de-



Fabio Paternò





**FIGURA 1**  
Esempio di interazione con i primi calcolatori elettronici.

*bugging* anche se poteva essere usato per inserire direttamente dei comandi nel sistema. In sostanza, la nascita dell'interazione con i computer coincide con la nascita degli schermi grafici e con la possibilità di interagire con essi. La tesi di dottorato nel 1963 al MIT di Ivan Sutherland fornisce probabilmente la prima interfaccia utente grafica interattiva con SketchPad [9]. Questo sistema consentiva la manipolazione di oggetti grafici tramite una penna ottica: si potevano creare elementi grafici, spostarli, ricevere un *feedback* grafico, e cambiare gli attributi. Molte delle tecniche di interazione grafica vengono introdotte negli anni '70 nel laboratorio di ricerca della Xerox Parc. Nel 1981 appare sul mercato il primo sistema commerciale con il supporto della manipolazione diretta: *Xerox Star*. Verrà seguito da *Apple Lisa* nel 1982 e dal *Macintosh* nel 1984. Nel 1983, Ben Shneiderman, all'Università del Maryland definisce i principi che caratterizzano la manipolazione diretta [8]. Gli schermi grafici portarono anche all'introduzione di dispositivi che facilitavano l'interazione con essi. Il più conosciuto è il *mouse*. Esso fu inventato nei primi anni '60 da Douglas Engelbart. Altri dispositivi furono introdotti presto come la penna ottica, la tavola digitalizzatrice. Insieme a schermi e mouse fu ben presto chiara la necessità di avere a di-

sposizione tecniche che consentissero di rendere più efficiente il lavoro degli utenti e, quindi, furono introdotti i sistemi a finestre che consentivano di interagire con varie applicazioni associate a diverse finestre contemporaneamente. I primi sistemi a finestre furono sviluppati alla Xerox PARC e furono lo *Smalltalk* e *InterLisp* a metà degli anni '70 e furono adottati dallo Xerox STAR e Apple Lisa. L'Apple Macintosh rese poi queste tecniche disponibili al grande pubblico nel 1984 con uno schermo a 9 pollici, bianco e nero, 512 per 342 pixel. In quel periodo, al MIT nasce X Window System [7] che consente tramite la sua architettura *client/server* di avere un sistema a finestre, interattivo con una notevole flessibilità e portabilità in ambienti distribuiti. A consacrare il successo delle interfacce grafiche si ha, nel 1985, la prima versione di MS-Windows. Si affermano così le interfacce WIMP (*Window Icon Menu Pointer*) che diventano gli ambienti con cui, ancora oggi, si interagisce principalmente con i computer. Nel frattempo, alcuni ricercatori cercavano di identificare i concetti e i metodi più rilevanti per questa disciplina emergente: ad esempio, nel 1983, Card, Moran e Newell [1] introducono il modello di processore umano che fornisce una rappresentazione semplificata di come l'essere umano percepisce ed elabora gli stimoli esterni e il metodo GOMS (*Goals, Operators, Methods, Selection rules*) per predire la *performance* umana durante l'interazione con un computer. L'evoluzione tecnologica ha caratterizzato anche quest'area. Per esempio, gli schermi grafici: attualmente vi è una varietà impressionante che va da piccoli schermi che possono stare in telefoni o orologi a grandi schermi che coprono porzioni significative di pareti. Un numero crescente di oggetti, anche di uso quotidiano, vengono arricchiti di funzionalità interattive (come automobili, frigoriferi, orologi). Parallelamente, gli utenti si sono evoluti con la tecnologia. Quando, nel 1979-1980, chi scrive, si immatricolò al corso di laurea di Scienze dell'Informazione, non aveva mai visto prima un computer. Inoltre, i suoi primi programmi erano registrati su schede perforate che bisognava portare al centro di calcolo per l'esecuzione. Oggi, suo figlio (nato nel 1997) ha imparato prima a interagire con un personal computer che a leggere, sfruttando

la memoria visiva per ricordare i bottoni da selezionare, per esempio per poterlo spegnere.

### 3. L'USABILITÀ

Il principale obiettivo di questa disciplina è l'usabilità. Questo è un concetto complesso, a molte dimensioni (come, per esempio, rilevanza, efficienza, facilità di apprendimento, sicurezza, flessibilità ecc.), dove il peso di ciascuna dimensione dipende anche dal dominio applicativo che si considera. Infatti, se si prova a considerare applicazioni appartenenti a domini diversi si vede subito come cambia l'importanza di alcune sue componenti. Si consideri, ad esempio, un'applicazione per impiegati di banca, un video gioco e un sistema di controllo di traffico aereo. Nell'applicazione bancaria l'obiettivo è svolgere le varie pratiche in modo efficiente e rapido, quindi l'interfaccia deve essere immediata per un utente che accede in modo continuo. In un video gioco è importante il coinvolgimento emotivo, il divertimento, ed è, quindi, importante creare effetti che possono coinvolgere l'utente. Nel sistema di controllo di traffico aereo, non interessa il coinvolgimento emotivo ma lo svolgimento delle attività in modo da evitare errori perché un errore può mettere a repentaglio la vita di molte persone. Comunque, definizioni generali di usabilità esistono. Quella più riconosciuta è dello standard ISO 9241 (Requisiti ergonomici per lavoro di ufficio con terminali), in cui viene definita come la misura in cui un *prodotto* può essere usato da specifici *utenti* per raggiungere specifici *obiettivi* con efficacia, efficienza e soddisfazione in un specifico *contesto di uso*. Dove per *efficacia* si intende l'accuratezza e completezza con cui gli utenti possono raggiungere i loro obiettivi; l'*efficienza*, invece, è l'insieme di risorse spese in relazione all'accuratezza e completezza degli obiettivi raggiunti (per esempio, il tempo richiesto); la *soddisfazione* è, infine, il comfort e l'accettabilità del sistema per gli utenti e le altre persone influenzate dal suo uso. Sono tanti i motivi per cui l'usabilità è importante:

■ aumenta l'efficienza degli utenti, questo significa per le aziende e le organizzazioni un aumento di produttività;

■ si riducono gli errori, quindi, aumenta la sicurezza nell'interazione con applicazioni o servizi informatici;

■ si riduce il bisogno di addestramento (che è anche esso un costo per le aziende e gli enti);

■ si riduce il bisogno di supporto degli utenti, che quindi accettano più volentieri l'uso di applicazioni informatiche;

■ infine aumenta le vendite, infatti è evidente che quando due sistemi hanno funzionalità simili gli utenti scelgono quello che consente un più facile accesso.

È possibile identificare delle misure quantitative di usabilità [4]. Per esempio il tempo che l'utente ha impiegato per completare un compito, il numero di compiti che l'utente ha completato in un intervallo di tempo, il rapporto tra interazioni corrette ed errori (in questi casi si considera errore un'azione che non serve per raggiungere l'obiettivo corrente), il numero di errori, il numero di compiti che sono stati svolti dall'utente, il numero di compiti che non sono stati svolti dall'utente (anche se erano supportati dal sistema), il numero di volte che l'utente non è riuscito a risolvere un problema, il rapporto tra gli utenti che hanno scelto la migliore strategia e quelli che non la hanno scelta, la quantità di tempi morti, il numero di volte che l'utente si è distratto. Questo tipo di indicatori è particolarmente significativo nel caso di applicazioni dove è ben chiaro quali sono gli obiettivi dell'utente.

Un concetto distinto è quello di *accessibilità*. Un sistema informatico è *accessibile* se può essere usato da tutti, comprese le persone disabili. Esempi di problemi di accessibilità nascono quando si comunica informazione solo tramite attributi grafici che non possono essere percepiti da utenti ciechi. Ne è un esempio indicare con il colore rosso i campi obbligatori in un modulo interattivo o fornire informazioni tramite un grafico senza alcun testo alternativo. L'accessibilità solleva importanti questioni che coinvolgono molte persone con inabilità permanenti o legate a particolari situazioni. L'uso di Internet e di tutte le informazioni correntemente disponibili in formato digitale, specialmente quando recuperabili *on-line*, rappresentano una opportunità importante per la società. Accedere all'informazione tutta-

via non basta. Infatti, anche se un sistema è accessibile per utenti disabili, esso può ancora non essere sufficientemente usabile per tali utenti. L'obiettivo dell'usabilità è di rendere l'esperienza degli utenti più efficiente e soddisfacente. Ovviamente, in questa prospettiva, un sistema non può essere usato se non è accessibile. Per questo l'accessibilità è una pre-condizione per l'usabilità. Tuttavia, queste questioni sono spesso affrontate da due comunità diverse che si concentrano essenzialmente solo su uno dei due aspetti (per esempio il W3C, il consorzio che definisce gli standard per il Web, ha definito delle linee guida per l'accessibilità ma non per l'usabilità). C'è, quindi, la necessità di integrare questi due aspetti per ottenere sistemi informatici per tutti.

#### 4. L'INTERAZIONE

Per capire che cos'è lo *human-computer interaction* e l'usabilità è opportuno far riferimento al modello di Norman [5] il cui scopo è identificare le fasi principali nell'interazione utente, fornendo così una indicazione utile e strutturata degli aspetti principali da considerare quando si progettano interfacce. Questo modello fornisce una valida struttura logica, anche se semplificata, per la progettazione e valutazione. Vengono identificate sette possibili fasi per descrivere l'interazione uomo-calcolatore:

- Formulare l'obiettivo
- Formulare l'intenzione
- Identificare l'azione
- Eseguire l'azione
- Percepire lo stato del sistema
- Interpretare lo stato del sistema
- Valutare il risultato rispetto all'obiettivo.

**FIGURA 2**  
Il ciclo  
di interazione  
di Norman



Per esempio, un obiettivo può essere una certa modifica di una lettera. Tale obiettivo generale può essere decomposto in intenzioni più specifiche, come modificare le conclusioni. Questo richiede alcune azioni specifiche, come modificare alcune parole che possono essere eseguite tramite selezione delle parole interessate e immissione delle nuove parole. Una volta che le azioni sono eseguite, il sistema mostra il risultato, l'utente lo può guardare e magari accorgersi che sono state modificate anche alcune parole che, invece, andavano bene: capisce, quindi, che il suo obiettivo non è ancora stato completamente raggiunto e deve essere eseguita una nuova interazione. Norman colloca le sette fasi nel contesto del ciclo di un'interazione, al cui interno identifica il "golfo dell'esecuzione" (la differenza tra le intenzioni dell'utente in termini delle azioni da svolgere e le azioni effettivamente consentite dal sistema) e il "golfo della valutazione" (la differenza tra le rappresentazioni fornite dal sistema e quelle che si aspetta l'utente). Nelle interfacce a bassa usabilità, dove i compiti da svolgere sono malamente supportati, i due golfi possono essere utili per identificare le discrepanze tra quello che l'utente vorrebbe fare e quello che può fare (golfo dell'esecuzione) e tra quello che il sistema presenta e quello che l'utente si aspetta (golfo di valutazione) come è possibile vedere dalla figura 2. In entrambi i casi, è possibile identificare una distanza cognitiva indicata dalla quantità e qualità di informazione da elaborare da parte dell'utente allo scopo di colmare il divario nel golfo considerato.

Per capire meglio questi concetti vari esempi possono essere fatti. Un esempio abbastanza intuitivo è il gioco che consiste nell'avere i numeri da 1 a 9 inizialmente disponibili a ciascuno dei due giocatori. I giocatori giocano uno alla volta. A ogni turno i giocatori scelgono uno dei rimanenti numeri (rendendolo indisponibile). Se un giocatore possiede tre numeri la cui somma è 15 ha vinto. Innanzitutto bisogna capire il problema. Entrambi i giocatori condividono un obiettivo comune: "vincere il gioco". Vi è però anche un altro obiettivo: "se a un certo punto io non posso vincere allora non devo



consentire all'altro giocatore di vincere". Una possibile strategia è scegliere un numero tra i numeri rimanenti cercando di vincere ma anche verificare se l'altro giocatore sta per vincere. Quindi l'attività "in background" è ricordare i numeri che il giocatore ha già preso, ricordare i restanti numeri (e quelli presi dall'avversario) e ricordare a chi tocca giocare. Si può notare che il gioco diventa non banale. Si supponga allora di dover progettare un'interfaccia utente che renda più agevole fare questo gioco. Una prima soluzione è rappresentata da un'interfaccia del tipo riportato in figura 3.

Come si può notare è chiaramente evidenziato a chi tocca giocare e quali sono i numeri già selezionati (in rosso) e quelli disponibili (in verde) e chi li ha selezionati. Tuttavia, i giocatori devono ancora fare conti di una certa complessità per capire che numero scegliere evitando che l'avversario alla mossa successiva possa fare una scelta per cui vinca il gioco. Vi è una distanza cognitiva notevole tra scegliere le azioni idonee e l'obiettivo dell'utente. Una interfaccia che limita questo carico cognitivo, e quindi è più usabile, è quella rappresentata in figura 4. L'idea è che i giocatori ricevono un'interfaccia sostanzialmente diversa: una matrice 3 x 3 dove posizionare, un giocatore, delle X, e l'altro, delle O. Assumendo che alla matrice corrisponda una numerazione come quella indicata dalla matrice piccola a sinistra, il gioco diventa come il Tic Tac Toe (meglio conosciuto in Italia come "Tris") per cui lo scopo dei giocatori è di posizionare tre elementi allineati in fila o in diagonale. Si vede subito che capire se l'avversario sta per vincere diventa una cosa molto intuitiva, rilevabile a colpo d'occhio e tale da non richiedere elaborazioni particolarmente complicate.

Si nota, quindi, come un principio fondamentale è capire gli utenti e i compiti che intendono svolgere. Tali compiti sono attività necessarie per raggiungere un obiettivo, dove l'obiettivo può essere o la modifica dello stato di un'applicazione (per esempio, aggiungendo nuovi dati) o un accesso a delle informazioni contenute nelle applicazioni [6]. L'interfaccia utente deve consentire lo svolgimento di tali attività nel modo più im-

mediato ed intuitivo possibile. A questo scopo, una fase importante nella progettazione è l'analisi dei compiti, che mira a identificare quali sono i compiti più rilevanti e le loro caratteristiche. Per capire questo, è importante coinvolgere nella progettazione l'utente finale e tenere presente come svolge correntemente le attività. A tal proposito, possono essere utili varie tecniche: interviste, *workshop*, questionari, osservare gli utenti nel contesto usuale, considerare la documentazione esistente e i metodi di apprendimento utilizzati. L'analisi dei compiti può essere fatta a diversi livelli di granularità, anche a seconda degli obiettivi. Essa si può focalizzare su una persona che interagisce con una applicazione tramite calcolatore o si può allargare a tutto il processo all'interno del quale questa attività si svolge e che può coinvolgere diverse persone. Il risultato è l'identificazione dei compiti più importanti per l'applicazione considerata, i relativi problemi, attributi e preferenze dell'utente. Questo è un punto centrale nella progettazione. Si veda un esempio che con-



**FIGURA 3**  
Interfaccia per il gioco che consiste nella scelta tra numeri a somma 15



**FIGURA 4**  
Interfaccia più intuitiva per il gioco

sidera l'accesso a un orario ferroviario tramite due sistemi diversi.

Nel primo caso (Figura 5), l'accesso all'orario non è evidenziato così bene come nel secondo (Figura 6). Ma grosse differenze si notano, poi, quando si devono specificare i parametri per identificare gli orari di interesse. Nel primo caso, c'è una semplice richiesta di stazione di partenza, di arrivo, giorno e data. Tuttavia, la seconda soluzione fa emergere tutta una serie di casistiche che nella prima non vengono affrontate in modo chiaro. Per esempio, si vuole riferire l'orario alla partenza o all'arrivo. Inoltre, se uno vuole andare in un posto, molto proba-

bilmente vorrà anche tornare e, quindi, sarebbe più efficace ed efficiente richiedere le informazioni relative a entrambe le tratte con un'unica interrogazione. È evidente, quindi, come la seconda soluzione sia più efficace ed efficiente in quanto è stata fatta un'analisi dei compiti più accurata che ha fatto emergere l'importanza del compito di ricevere informazioni relative agli orari nell'applicazione considerata e ha fatto emergere tutta una serie di compiti secondari, che danno la possibilità agli utenti di raggiungere i loro obiettivi (in questo caso, soddisfare dei bisogni informativi) in modo più completo e articolato.

FIGURA 5

Esempio di accesso agli orari dei treni

FIGURA 6

Esempio di accesso più efficace agli orari dei treni

## 5. LA PROGETTAZIONE

La progettazione di interfacce è orientata alla comunicazione con gli utenti finali. Essa ha alcune problematiche in comune con la progettazione di oggetti, manifesti o *clip* televisivi, edifici. In tutti questi casi, si tratta di progettare forme e spazi nel contesto di uno specifico compito o problema [3]. A questo scopo, bisogna evitare di considerare solo aspetti funzionali interni, trovare soluzioni che abbiano un fondamento generale, evitando quindi di basarsi solo sull'intuizione del progettista e trovando un giusto equilibrio tra metodo e intuizione. Lo scopo è di selezionare gli elementi attentamente, definire soluzioni chiare, economiche, convincenti che si possano operare immediatamente, che possano essere più facilmente assimilate, comprese e ricordate, attirano immediatamente l'utente sugli aspetti importanti e consentono di raggiungere gli obiettivi rapidamente e senza errori. Tante volte le soluzioni semplici sono le più usabili. Si pensi, ad esempio, a quei siti i quali, quando vi si accede, fanno partire delle animazioni che non forniscono particolari informazioni aggiuntive ma sono semplicemente un orpello che lo sviluppatore ha inserito quasi più a dimostrare le sue capacità implementative che a fornire qualcosa di utile. Si pensi allo stato d'animo di un utente che magari accede tramite connessione *modem* lenta e si trova a dover subire l'animazione, magari senza la possibilità di saltarla, per lunghi minuti per i quali dovrà pagare una bolletta salata, e magari per sbaglio, una volta entrato nel sito, seleziona la pagina *home* e si ritrova nuovamente l'animazione che diventa sempre più odiata. Questo non vuol dire che le animazioni non debbano mai essere inserite, ma, quando si inseriscono, devono in qualche modo fornire qualcosa di aggiuntivo e non essere un elemento fine a se stesso. Per esempio, possono essere utili per capire fenomeni dinamici che evolvono nel tempo. Quindi, una progettazione efficace è in grado di ridurre gli elementi alla loro essenza, regolarizzarli in modo da favorire la loro interpretazione e combinarli in modo da sfruttarli al massimo. Un esempio di massimo sfruttamento è la barra orizzontale in alto delle finestre che contengono le applicazioni in am-

bienti PC Windows (ma anche in altri): quanti compiti può supportare! Difficilmente si possono indovinare tutti. Sono tanti: indicare il nome del file associato all'applicazione, il tipo di applicazione, se la finestra è quella correntemente selezionata, la possibilità di minimizzare o massimizzare le dimensioni della finestra, la possibilità di chiuderla e la possibilità di spostarla sullo schermo. Il tutto grazie a una piccola striscia rettangolare, estremamente poco intrusiva!

Un altro aspetto importante nella progettazione di interfacce utenti è come strutturare e organizzare la loro presentazione. A questo scopo, vi sono una serie di tecniche comunicative che mirano ad aiutare l'utente a scorrere l'interfaccia, interpretare gli elementi e trovare quello che cerca. Quello che il progettista deve cercare di fare è di raggruppare elementi, creare gerarchie, rappresentare relazioni, indicare ordinamenti tra elementi e, alla fine, trovare un equilibrio complessivo. Raggruppare elementi è utile per indicare quelli che sono semanticamente più collegati tra di loro, creare gerarchie serve per evidenziare gli elementi più importanti (per esempio, quelli che vengono acceduti più frequentemente), le relazioni servono per far capire come uno o più elementi possono influenzare altri elementi e, infine, vi possono essere vari tipi di ordinamenti logici o temporali tra gruppi di elementi. Far cogliere nel modo più immediato queste relazioni semantiche tra elementi dell'interfaccia è compito del progettista. Si veda ad esempio una pagina Web di un quotidiano ad alta diffusione (Figura 7) per capire come queste tecniche possono venire usate. Come si vede c'è una chiara gerarchia tra gli elementi. L'informazione che al momento è ritenuta più importante è evidenziata al centro con una immagine grande di accompagnamento e un titolo con *font* grandi. La seconda informazione più importante del momento viene sotto e ha a disposizione meno spazio, usa un'immagine più piccola e *font* più piccoli. Ai lati vi sono raggruppamenti di informazioni che hanno tipologie simili come i servizi di Repubblica.it e 24 ore (che contiene le ultime news). In alto, vi è un esempio di elemento (Ricerca nel sito) che è in relazione con gli altri, in quanto, se viene selezionato, fa cambiare la pagina per con-



**FIGURA 7**  
Esempio di tecniche comunicative applicate ad un'interfaccia Web

sentire una ricerca su informazioni correlate presenti nel sito. Sempre nella zona in alto, vi è la possibilità di selezionare un insieme di elementi associati a varie sezioni che sono ordinate logicamente tra di loro (politica, cronaca, economia ecc.). Si vede, quindi, come tutte le varie relazioni semantiche vengono evidenziate con tecniche di progettazione di interfacce utenti per consentire agli utenti di percepirla nel modo più immediato.

In generale, vi sono vari modi di considerare un sistema interattivo. Un possibile modo è quello di considerare i compiti da eseguire per raggiungere gli obiettivi dell'utente e gli oggetti logici che vanno manipolati per il loro svolgimento. Questa è una visione logica del sistema che può essere discussa tra le varie persone coinvolte nella progettazione (utente finale, committente, progettista di interfacce, sviluppatori software). Si può avere un'altra vista, che è sempre logica ma è più focalizzata sull'interfaccia, ovvero considerare le presentazioni e le interazioni che ne fanno parte e come muoversi da una presentazione all'altra. Le interazioni sono identificate in base alla loro semantica (i risultati che consentono di ottenere). Per esempio, si può dire che in un certo punto si ha bisogno di una selezione, ma senza specificare il tipo di modalità richiesta per realizzarla (che possono essere ad esempio selezione grafica, vocale, o tramite un gesto). Vi è, poi, una possibile descrizione più concreta dove si specificano le

modalità e le tecniche di interazione che si vogliono usare. Per esempio, si può dire che in un sistema desktop grafico la selezione avviene tramite una lista con una barra di scorrimento. Infine, si ha l'implementazione, che può essere in HTML, Java ecc.. Quando si progetta, il livello di astrazione del punto di partenza può cambiare a seconda dei casi. Certe volte si identificano i compiti da supportare e quelli sono il punto di partenza per ottenere, tramite raffinamenti successivi, l'implementazione. In altri casi, si parte da una certa implementazione che esiste e si cerca di capire se effettivamente quella è la migliore per supportare le attività dell'utente.

## 6. INTERFACCE ED ADATTAMENTO

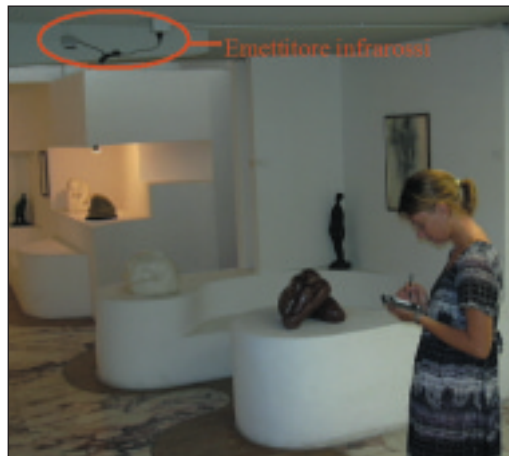
La ricchezza delle tecnologie informatiche moderne consente molti usi dei sistemi interattivi. Diventa spesso necessario che le interfacce utenti si sappiano adattare al contesto di uso, che si può considerare da tre punti di vista: quelli relativi all'utente, al dispositivo e all'ambiente circostante. Per quanto riguarda l'utente, aspetti importanti sono gli obiettivi e i relativi compiti, le preferenze e il livello di conoscenza del dominio applicativo e delle modalità di interazione. Nel dispositivo usato per l'interazione, è importante considerare le modalità supportate, l'ampiezza e la risoluzione dello schermo, le capacità e ve-



**FIGURA 8**

*Esempio di interfacce adattabili*

locità di connessione con altri dispositivi. Infine, l'ambiente ha vari aspetti che possono influenzare le modalità di interazione come il livello di rumore e di luce corrente, gli oggetti che sono disponibili. Le interfacce utenti si trovano, quindi, a doversi adattare a questi aspetti per avere una maggiore usabilità. Vi sono due tipologie di adattamento: l'*adattabilità*, ovvero la capacità di modificare aspetti su richiesta esplicita dell'utente in accordo a opzioni predefinite, l'*adattività*, ovvero la capacità del sistema di modificare aspetti dinamicamente senza richiesta esplicita dell'utente. Mentre l'*adattabilità* consente essenzialmente di scegliere le modalità di interazione con un'applicazione tra un insieme predefinito, l'*adattività* implica sistemi che le cambiano dinamicamente rispetto al contesto d'uso. Questo da una parte implica una maggiore flessibilità, ma dall'altra implica che nuove problematiche di usabilità si possono generare se questi cambiamenti avvengono in modo non facilmente comprensibili per l'utente. A questo punto, ci si potrebbe domandare che cosa si può adattare in un'interfaccia utente. Vi sono tre tipologie di aspetti che si possono adattare: le presentazioni (nella scelta del layout, degli attributi grafici ecc.), il comportamento dinamico (nella scelta delle modalità di navigazione, dell'abilitazione e disabilitazione delle tecniche di interazione ecc.), il contenuto dell'informazione fornita. Nella figura 8 è possibile vedere un esempio di interfaccia adattabile. A seconda della tipologia di utente che si è selezionato, si attivano diverse modalità di accesso all'applicazione. Nel caso di un turista, si atti-



**FIGURA 9**

*Palmare usato come supporto per la visita a musei. La posizione del visitatore è rilevata con dispositivi ad infrarosso*

vano le possibilità di accedere a informazioni generiche e alla mappa della città e del museo dalle quali gli utenti selezioneranno gli elementi di interesse per cui riceveranno informazioni. Nel caso di uno studente che ha qualche conoscenza di base, allora si possono attivare liste di elementi sugli aspetti di maggior interesse di cui l'applicazione è in grado di fornire informazioni. Alla fine si vede l'interfaccia di un esperto che ha la possibilità di comporre richieste dettagliate, conoscendo già le informazioni di base. Sempre restando nello stesso ambito si può vedere un esempio di interazione che si adatta al dispositivo ed all'ambiente. In questo caso si considera l'utente che è dentro un museo e usa una guida palmare come supporto per la sua visita (Figura 9). La guida cerca di essere il meno intrusiva possibile, fornendo molta informazione in modo vocale, in modo da consentire all'utente di poter apprezzare gli oggetti che sono nel museo,

mentre essa fornisce informazione aggiuntiva. Il canale visivo è usato dalla guida palmare per fornire informazione utile a capire dove ci si trova, quali altri elementi di interesse sono nelle vicinanze, a controllare parametri della guida, ad accedere a video che forniscono informazioni su argomenti correlati che non sono nel museo, a mostrare come raggiungere un elemento di interesse.

La soluzione che è stata adottata per il Museo del Marmo di Carrara [2] fornisce un supporto che dipende anche dalla locazione dell'utente e tale locazione è rilevata automaticamente. Questo è ottenuto tramite dispositivi ad infrarossi che sono collocati sul soffitto all'entrata di ogni stanza e che emettono verso il basso un segnale che contiene un identificatore della stanza (Figura 9). In realtà, ogni dispositivo è composto di emettitori multipli di segnali ad infrarossi per aumentare la facilità di rilevazione del segnale. Quando il dispositivo rileva il segnale, identifica la stanza in cui si entra e automaticamente emette un feedback sonoro, e mostra sullo schermo dove l'utente si trova, dopodiché alla prima selezione mostra la mappa della nuova stanza con icone per ogni opera d'arte (Figura 10). Vi sono icone per ogni tipologia e l'utente selezionando un'icona riceve informazione aggiuntiva in modo vocale e la possibilità di accedere a video su argomenti correlati (se esi-

stono). Questa soluzione è resa possibile dalla disponibilità di memorie da 1 Gbyte per palmari che consentono di registrare informazioni multimediali anche molto ricche. Questo permette di limitare l'interazione con l'esterno al rilevamento di segnali che consentono di individuare in che ambiente si trova l'utente. Un'altra possibile soluzione sarebbe stata di rilevare l'opera d'arte più vicina e attivare automaticamente il commento vocale corrispondente. Il limite di questa soluzione è che rischia, in alcuni casi, di essere troppo invadente e fornire commenti non desiderati, ad esempio quando ci si trova vicino a un'opera ma si osserva un'altra che si trova più distante.

## 7. LE INTERFACCE MULTI-DISPOSITIVO

Attualmente, una delle principali problematiche che ha un impatto sulle interfacce utenti è la continua immissione sul mercato di nuove tipologie di dispositivi interattivi: si va da telefoni digitali interattivi a schermi a muro molto ampi, passando da palmari, telefoni UMTS, tablet PCs, per citarne alcuni. Interagire con servizi interattivi diventa sempre di più un'esperienza multi-dispositivo. Diventa, quindi, importante capire quali nuove problematiche vengono introdotte in questo contesto. Una prima cosa da capire è che non è possibile fare tutto tramite tutti i dispositivi. Vi sono delle caratteristiche dei dispositivi che li possono rendere adatti a supportare dei compiti ma sono inadeguati per altri. Per esempio, chi scrive non userebbe mai (e tanto meno pagherebbe) un servizio che gli consenta di usare un telefono (anche con uno schermo un po' più grande di quelli tradizionali) per vedere un film o una intera partita di calcio in quanto l'esperienza che avrebbe sarebbe alquanto angusta e non gli consentirebbe di apprezzare una serie di dettagli che sono fondamentali in questo tipo di attività. Viceversa se resta bloccato nel traffico automobilistico e vuole trovare un itinerario alternativo, allora la speranza di aiuto gli può essere data da un dispositivo mobile. In altri casi, le attività supportate possono essere le stesse attraverso dispositivi diversi, ma le modalità cambiano. Per esempio, una preno-



**FIGURA 10**

*Rilevazione dell'utente ed indicazione di informazione dipendente dalla nuova posizione*

tazione alberghiera fatta tramite cellulari con accesso Web o WAP mira a essere essenziale, si comunicano date di arrivo e partenza e si lascia un recapito, viste le limitate possibilità, mentre attraverso un sistema desktop si può comodamente fornire una serie di informazioni aggiuntive, ad esempio, per esprimere le proprie preferenze in termini di stanze, pasti ecc.. Oppure il sistema desktop può presentare moduli di prenotazione ampi dove i vari campi possono essere riempiti in diversi ordini mentre il cellulare può imporre una certa sequenza nel fornire i parametri della richiesta a causa dello schermo limitato. Vi sono poi attività attraverso un tipo di dispositivo che possono abilitare o disabilitare attività attraverso un altro. Per esempio, si può fare una prenotazione aerea tramite un sistema *desktop* e questo abilita la possibilità di accedere tramite il cellulare ad informazioni *real-time* relative al volo prenotato. Vi sono, poi, anche attività che restano le stesse indipendentemente dal dispositivo, per esempio fare un *login* resta più o meno lo stesso attraverso diversi tipi di dispositivi. Nell'adattarsi al tipo di dispositivo bisogna anche considerare le modalità supportate perché queste influenzano le possibilità di interazione. Lo svolgimento dei compiti può essere influenzato dalla modalità di interazione disponibile: un insieme di *input* può richiedere interazioni separate attraverso un dispositivo grafico, mentre tali informazioni possono essere fornite tramite una interazione singola usando un'interfaccia vocale. Vi sono delle differenze intrinseche tra le varie modalità. Per esempio, il canale vocale è più adatto per messaggi brevi, per segnalare eventi, azioni immediate, per evitare sovraccarico visivo e quando gli utenti sono in movimento. Invece, il canale visivo è più utile per messaggi complessi o lunghi, per identificare relazioni spaziali, quando devono essere eseguite azioni multiple, in ambienti rumorosi o con utenti stazionari. Quando un sistema supporta modalità multiple (per esempio, l'interazione grafica e vocale) lo spazio delle possibili tecniche implementative è ampio e deve considerare diversi modi per combinare le modalità: complementare (entrambe le modalità sono usate in modo sinergico per completare l'interazione), assegnamento (una specifica

modalità deve essere usata per realizzare un certo scopo), ridondanza (più modalità sono usate per ottenere lo stesso effetto), equivalenza (è possibile scegliere tra più modalità per ottenere lo stesso effetto).

## 8. CONCLUSIONI E TENDENZE EVOLUTIVE

Questo articolo ha fornito un'introduzione all'affascinante mondo dello *human-computer interaction* spiegandone gli obiettivi e i concetti fondamentali, e mostrando esempi di applicazione. Un'area di ricerca che è caratterizzata da un'elevata interdisciplinarietà e che negli ultimi anni ha assistito a una vera esplosione di interesse e che si è evoluta sostanzialmente. Tuttavia, questa evoluzione sembra dover continuare nei prossimi anni sulla spinta della evoluzione delle tecnologie di interazione e dei requisiti degli utenti in costante cambiamento. La continua introduzione di nuovi dispositivi interattivi informatici nelle nostre case, uffici, auto, luoghi di commercio e di turismo implica la necessità di progettare una usabilità pervasiva che sappia garantire modalità di fruizione soddisfacenti nei diversi possibili contesti di uso. Questo apre la possibilità di creare in futuro servizi migratori; ovvero servizi interattivi che seguono l'utente nei suoi spostamenti e che si adattano ai nuovi dispositivi disponibili nei nuovi ambienti in cui l'utente si viene a trovare, in modo da garantire livelli di usabilità elevati, consentendo di continuare l'interazione al punto dove si era lasciata con il dispositivo nell'ambiente precedente. Un esempio è una persona che si sta iscrivendo a un congresso tramite un sistema desktop, si accorge che è tardi e allora prende il suo PDA e continua l'iscrizione mentre esce dall'ufficio, fino a che entra nell'auto da dove può completarla usando un sistema di interazione vocale, il tutto senza mai dover rifare le operazioni che erano state effettuate tramite i dispositivi usati precedentemente e con interfacce che si adattano al nuovo dispositivo usato. Il livello di multimodalità supportata dai nostri sistemi aumenterà in modo significativo per molte ragioni. Alcune tecnologie stanno sostanzialmente migliorando, come quelle relative all'interazione vocale che mostrano una crescente capacità di interpreta-

zione dell'input umano e che cominciano a essere supportate, in modo stabile, anche per interazione via Web. Anche le tecnologie per rilevare la presenza dell'utente si stanno diversificando e migliorando. Il miglioramento delle tecniche di riconoscimento di forme ed elementi nelle immagini sta aumentando le possibilità dell'interazione tramite gesti, dove

a seconda del gesto riconosciuto sono attivate funzionalità diverse. Queste e altre possibilità hanno l'obiettivo di rendere l'interazione con i computer simile a quella tra esseri umani e quindi condurre all'affermazione del paradigma dell'interazione naturale, che garantisce l'usabilità in quanto estremamente immediato e spontaneo.

## La valutazione dell'usabilità

La valutazione di usabilità può essere svolta con diversi scopi. Vi possono essere obiettivi precisi come volere che un utente sia in grado di svolgere un compito con un certo numero di interazioni o in un certo periodo di tempo oppure che in una sessione media il numero di errori sia minore di un certo valore.

Vi sono vari metodi che vengono considerati per la valutazione di usabilità:

- **Valutazione basata su ispezione:** in questi casi un esperto valuta il prototipo o l'implementazione finale dell'interfaccia utente secondo dei criteri pre-definiti che possono essere una serie di proprietà che debbono essere soddisfatte (come fornire feedback continuo dello stato dell'interazione) o indicazioni di aspetti da considerare simulando l'interazione dell'utente (come avviene con il *cognitive walkthrough*).
- **Valutazione basata su test degli utenti in laboratorio:** in questo caso, si utilizzano laboratori attrezzati con videocamere che memorizzano le sessioni degli utenti, in un ambiente che cerca di essere il meno intrusivo possibile.
- **Valutazione basata su feedback degli utenti:** in questo caso si raccoglie il feedback in modo informale tramite questionari, workshop, focus group, interviste.
- **Valutazione remota:** l'utente e il valutatore sono separati nel tempo e/o nello spazio, per esempio vengono creati automaticamente *file di log* delle interazioni con gli utenti che vengono poi analizzati tramite strumenti specifici.
- **Valutazione basata su modelli (simulazione):** si crea un modello per predire e analizzare le modalità di svolgimento dei compiti in una certa interfaccia.
- **Valutazione basata su osservazione degli utenti sul campo:** si osservano per lunghi periodi gli utenti mentre interagiscono con il sistema nel loro ambiente quotidiano.

Scegliere il metodo per la valutazione dell'usabilità può dipendere da vari fattori come il numero e il tipo di utenti a disposizione. Può essere utile combinare più metodi. Per esempio, partendo con un'analisi degli utenti e dei compiti, creando poi dei prototipi che possono essere soggetti a valutazione euristica e poi a test empirici fino a che non raggiungono risultati soddisfacenti.

## Integrare Usabilità e Accessibilità

Se manca un'integrazione tra accessibilità ed usabilità si rischia di avere sistemi che consentono l'accesso anche ad utenti disabili ma con difficoltà. Per meglio comprendere queste problematiche possiamo prendere l'esempio dell'interazione di utenti ciechi attraverso *screen reader*, dispositivi che convertono in formato vocale tutta l'informazione che è sullo schermo. Per facilitare l'interazione hanno dei comandi che permettono, ad esempio, di accedere alla lista di link o frame che sono in una pagina Web. È emerso che quando vengono applicate solo le guideline dell'accessibilità (per esempio, quelle del W3C) diversi problemi di navigazione possono ancora emergere per utenti che interagiscono tramite *screen reader*:

- **Mancanza di contesto nella presentazione:** leggendo tramite lo *screen reader* l'utente può perdere il contesto complessivo della pagina e leggere solo piccole porzioni di testo. Per esempio, quando si muove da un link ad un altro con il tasto di tabulazione, l'utente cieco legge il testo del link tramite il dispositivo Braille o lo ascolta tramite il sintetizzatore vocale (per esempio, ".pdf", "more detail" ecc.) ma non conosce il testo che c'è prima e dopo.
- **Sovraccarico di informazione:** la porzione statica della pagina (link, frame ecc.) può sovraccaricare la lettura attraverso lo *screen reader* perché l'utente deve leggere, anche accedendo a pagine diverse, nuovamente questa parte rallentando in questo modo la navigazione.
- **Eccessiva sequenzializzazione nella lettura della informazione:** i comandi per navigare e leggere possono vincolare l'utente ad acquisire il contenuto della pagina sequenzialmente. Quindi, è importante introdurre meccanismi per facilitare l'identificazione di parti precise nella pagina. Ne è un esempio la pagina dei risultati generati da un motore di ricerca. Solitamente, nella parte alta di tali pagine vi sono parecchi link, pubblicità, il bottone di ricerca, altri bottoni e così via e i risultati della ricerca appaiono di seguito. Per superare questi problemi vi è la necessità di identificare criteri di progettazione che, oltre che a garantire l'accessibilità, permettano anche una elevata facilità di uso quando ai sistemi accedono utenti disabili.

## Scenari e analisi dei compiti

Due tecniche importanti per l'analisi dei bisogni dell'utente e la progettazione di sistemi usabili sono gli scenari e l'analisi dei compiti. Gli scenari servono a capire il comportamento degli utenti nella vita reale. Descrivono specifici utenti in specifiche circostanze. Solitamente tre-quattro scenari descrivono gli utenti standard. È una tecnica economica in quanto scrivere scenari non richiede sforzi eccessivi, può essere limitata quando ci sono tanti tipi di utenti o tante possibili interazioni. Si tratta di una descrizione compatta e informale di uno o più specifici utenti, che interagiscono con una specifica interfaccia, per raggiungere uno specifico risultato, in specifiche circostanze. È utile per catturare il contesto in cui un'applicazione viene usata, sollecitare una discussione utile per identificare requisiti, catturare episodi importanti da studi sul comportamento degli utenti, fornire un contesto per eseguire la valutazione. Possono essere annotati con un'indicazione delle conseguenze positive e negative derivanti dalla loro occorrenza. Nell'analisi dei compiti si identificano le attività che devono essere supportate, gli oggetti che servono per eseguire le attività, le conoscenze richieste per svolgere i compiti, la possibile allocazione dei compiti tra sistema e utente. Possono essere usate varie tecniche per supportare l'analisi dei task: interviste o workshop; questionari, osservazione degli utenti, analisi di come le attività sono eseguite, analisi della documentazione esistente e delle modalità di formazione. Da un'analisi dei compiti si possono derivare dei veri e propri modelli che definiscono le relazioni semantiche e temporali tra i vari compiti. Tali descrizioni possono essere usate in modo complementare con gli scenari, perché gli scenari sono descrizioni informali ma dettagliate di uno specifico uso in uno specifico contesto, i modelli di task descrivono in modo concettuale le possibili attività e le loro relazioni. Gli scenari possono supportare lo sviluppo di modelli di task indicando parte delle attività da considerare e, viceversa, vi possono essere scenari derivati da specifiche sequenze di compiti da svolgere. In generale, i modelli di task possono essere utili per comprendere un dominio applicativo, memorizzare il risultato di una discussione interdisciplinare, progettare una nuova applicazione consistente con il modello concettuale dell'utente, analizzare e valutare l'usabilità di un'applicazione esistente, aiutare l'utente durante una sessione con il sistema di *help* in linea, documentare un'applicazione interattiva.

### Bibliografia

- [1] Card S., Moran T., Newell A.: *The Psychology of Human-Computer Interaction*. Lawrence Erlbaum Associates, 1983.
- [2] Ciavarella C., Paternò F.: The design of a handheld, location-aware guide for indoor environments. *Personal and Ubiquitous Computing*, Vol. 8, n. 2, p. 82-81, Springer Verlag, May 2004.
- [3] Mullet K., Sano D.: *Designing Visual Interfaces*. Prentice Hall, 1995.
- [4] Nielsen J.: *Usability Engineering*. Academic Press.
- [5] Norman D.: *The Design of Everyday Things*. Basic Books, 1998.
- [6] Paternò F.: *Model-Based Design and Evaluation of Interactive Applications*. Springer Verlag, 1999.
- [7] Scheifler R., Gettys J.: The X Window System. *ACM Transactions on Graphics*, Vol. 5, n. 2, April 1986.
- [8] Shneiderman B.: Direct Manipulation, a Step Beyond Programming Languages. *IEEE Computer*, Vol. 16, n. 8, August 1983, p. 57-69.
- [9] Sutherland I.E.: *Sketchpad--A Man-Machine Graphical Communication System*. Proceedings of the Spring Joint Computer Conference, Detroit, Michigan, May 1963, and MIT Lincoln Laboratory Technical Report #296, January 1963.

FABIO PATERNÒ è Primo Ricercatore e Responsabile del Laboratorio di Ricerca su Interfacce Utenti nei Sistemi Informativi dell'Istituto di Scienze e Tecnologie dell'Informazione del CNR di Pisa. È stato coordinatore scientifico di vari progetti europei nell'ambito della progettazione di sistemi interattivi e relativi strumenti software di supporto. I suoi principali interessi correnti di ricerca sono metodi e strumenti per la progettazione, sviluppo e valutazione di sistemi interattivi usabili dipendenti dal contesto.  
fabio.paterno@isti.cnr.it