



**Dipartimento di Informatica  
Università degli Studi di Bari**

## Progetto VICE

---

### Usabilità dei sistemi di e-learning

---

M.F. Costabile, R. Lanzilotti, T. Roselli, C. Ardito, V. Rossano

RAPPORTO TECNICO: N.01 ANNO 2003

---

Unità Operativa  
Dipartimento di Informatica  
Università degli Studi di Bari

Responsabile  
Prof.ssa M.F. Costabile

**Indice**

<b>1. Introduzione</b>	<b>3</b>
<b>2. Learner-Centred Design vs User-Centred Design</b>	<b>3</b>
<b>3. Progettazione learner- centred</b>	<b>6</b>
<b>4. Usabilità del software LC</b>	<b>6</b>
<b>4.1 Il concetto di usabilità</b>	<b>7</b>
<b>4.2 L'usabilità per l'e-learning</b>	<b>8</b>
<b>4.3 Valutazione di usabilità per sistemi di e-learning</b>	<b>9</b>
<b>5. Euristiche per l'e-learning</b>	<b>10</b>
<b>6. Analisi di utenti di sistemi LC</b>	<b>12</b>
<b>Bibliografia</b>	<b>12</b>

## 1. **Introduzione**

Il lavoro dell'unità di Bari è essenzialmente rivolto all'evoluzione di una metodologia di Learner-Centred Design (LCD). In particolare, si occupa della valutazione di sistemi di e-learning, non solo per quanto riguarda l'usabilità delle interfacce utente ma anche per quanto riguarda l'efficacia dell'apprendimento.

Il lavoro del primo anno prevede:

1. analisi dello stato dell'arte su LCD
2. analisi di utenti e attività di lavoro che il software deve supportare
3. studio degli utenti mentre analizzano sistemi di e-learning già disponibili. Ciò allo scopo di capire la difficoltà che incontrano, i problemi delle piattaforme attuali e trarre indicazioni per un progetto più efficace. In questo modo si possono identificare le caratteristiche più critiche per tali sistemi di cui tener conto per definire metodi di valutazione di usabilità specifici per e-learning
4. studio di possibili tool di scaffolding da implementare, negli anni successivi, nell'interfaccia dei sistemi di e-learning, in collaborazione con gli educational experts.

In questo documento si descrive il lavoro effettuato fino alla fine di settembre 2003 relativamente ai punti precedenti. In particolare, riguardo al punto 1, nella sezione 2 si riporta un confronto tra caratteristiche di LCD e User-Centred Design e in sezione 3 si descrivono le principali caratteristiche di LCD che risultano dall'analisi dello stato dell'arte. Con l'obiettivo di identificare metodi di valutazione specifici per l'e-learning, nella sezione 4, dopo aver riportato le classiche definizioni di usabilità, si discute l'usabilità per l'e-learning e si propone di specializzare la tecnica di ispezione SUE [Mat02] alla valutazione di sistemi di e-learning. Alcune euristiche specifiche per questi sistemi sono descritte nella sezione 5. Stiamo effettuando studi con utenti che utilizzano sistemi già disponibili allo scopo di raffinare tali euristiche e di identificare i pattern di valutazione tipici dell'ispezione SUE, come descritto in sezione 6.

## 2. **Learner-Centred Design vs User-Centred Design**

I tool LC si rivolgono ai bisogni dei discenti: persone che sono usualmente novizie in un dato work practice, siano essi studenti o adulti che lavorano ed hanno bisogno di migliorare la loro formazione.

Ci sono varie differenze tra la progettazione di software user-centered e quella di software di formazione LC. Nel caso di generico software user-centered, lo sforzo di apprendimento da parte dell'utente è solo nella comprensione delle funzionalità del tool. Nel caso di software LC i discenti, oltre ad imparare il tool con il quale lavorano, devono anche apprendere un nuovo work practice.

Nel tradizionale progetto user-centered, gli utenti sono considerati esperti nel work practice, essi hanno bisogno dei computer solo per eseguire i loro compiti in maniera semplice ed efficiente [May99]. Contrariamente, i discenti sono novizi in un work practice e hanno bisogno di strumenti che tengano conto della loro mancanza di conoscenza e di supporti che li aiutino ad apprendere un nuovo work practice.

I discenti si possono caratterizzare nel seguente modo [Sol96]:

1. Non posseggono una quantità significativa di conoscenza nel work practice, cioè non condividono la conoscenza delle attività, della terminologia e di tutto ciò che riguarda il work practice.
2. Sono eterogenei: i discenti non necessariamente condividono la stessa cultura, la stessa formazione di base, la stessa esperienza, gli stessi stili di apprendimento. Per questo motivo i tool LC hanno bisogno di considerare un'ampia diversità nella popolazione di utenti.

Nel tradizionale approccio user-centered, il problema centrale è il *gap* concettuale tra l'utente e il tool, ciò che Norman modella con i concetti di "golfo di esecuzione" e "golfo di valutazione" [Nor86]. Questi golfi non possono essere ignorati nell'approccio LC, dal momento che i discenti certamente devono essere capaci di usare i tool progettati per i loro bisogni. Tuttavia, nell'approccio

LC, il problema centrale non è il gap concettuale fra utente e tool, ma piuttosto è il gap concettuale tra il discente e il modello di conoscenza realizzato da un esperto del work practice (“golfo di esperienza”). Un discente ha bisogno di capire che tipo di attività deve fare, gli argomenti che riguardano la pratica, la conoscenza necessaria a completare le attività di lavoro e così via. In altre parole, il discente ha bisogno di sviluppare un modello concettuale corretto e appropriato del lavoro che deve svolgere. La “dimensione” del golfo di esperienza descrive quanto un discente è lontano dalla conoscenza del work practice. La dimensione del golfo di esperienza è proporzionale alla quantità di cambiamento concettuale del modello del work practice del discente.

I tool LC possono supportare il discente nella fase di apprendimento attraverso l’uso di scaffolding. Gli scaffold sono caratteristiche del software che supportano l’esecuzione di un dato compito o attività da parte di qualcuno che è novizio in quel compito o attività. È dunque necessario sviluppare una teoria per il progetto di scaffold efficaci.

Soloway sottolinea che, come nel progetto di sistemi user-centered si può essere guidati dalla “teoria delle azioni” di Norman [Nor86], nel progetto LC è necessario sviluppare una teoria che aiuti a capire la nozione di “acquisire esperienza” e poter quindi sviluppare principi che siano alla base di un progetto LC. Le teorie di apprendimento costruttiviste e socio-costruttiviste possono aiutarci a descrivere la natura dell’esperienza che si sta considerando e a definire un modello che aiuti i discenti a superare il golfo dell’esperienza [Sol94, Sol96a].

L’approccio costruttivista riconosce che l’apprendimento non è un semplice processo passivo di trasferimento d’informazione dall’esperto al novizio, al contrario è un processo attivo, che considera un approccio “learning by doing”, dove i discenti devono manipolare cognitivamente il materiale che stanno apprendendo per creare collegamenti cognitivi dal nuovo materiale alla loro propria conoscenza [Pia54, Pap93].

L’approccio socio-costruttivista riconosce che l’apprendimento avviene quando i discenti lavorano in un contesto che permette loro di vedere e capire la cultura professionale del loro lavoro [Bro89]. In base a queste due teorie, un tool LC utilizzato nel suo contesto reale dovrebbe concettualizzare (o visualizzare) i work practice in modo tale che il discente possa impegnarsi e capire le attività, il linguaggio, i termini e tutto ciò che concerne il work practice.

I discenti hanno bisogno di aver accesso e partecipare ad attività simili a quelle che gli esperti svolgono in un determinato work practice [Bel96]. Mancando ai discenti la conoscenza del work practice, si ha la necessità di un supporto aggiuntivo (scaffolding) che li guidi e/o li aiuti ad eseguire le loro nuove attività. Naturalmente, i discenti non possono usare gli stessi tool che usano gli esperti data la differenza del loro livello di conoscenza. Così un progettista LC deve progettare tool modellati sui tool degli esperti, ma semplificati per poter essere idonei ai discenti. Questi tool dovrebbero permettere ai discenti di partecipare alle attività in maniera il più simile possibile a quella degli esperti e permettere la costruzione di artefatti che verranno usati nel work practice. A mano a mano che il discente acquisisce esperienza nel work practice, lo scaffolding dovrebbe “attenuarsi” o scomparire. Un semplice esempio di scaffolding potrebbe essere costituito dall’uso di una guida in linea che, con messaggi sia testuali che vocali, possa guidare il learner, o il gruppo se è in atto una situazione di apprendimento cooperativo, nell’esecuzione di un compito intervenendo sempre meno, con una invasività che va via via diminuendo se il discente o il gruppo acquisisce competenza, ovvero il sistema gliela riconosce.

Nella tabella 1 vengono sintetizzate le dimensioni che caratterizzano il progetto LC.

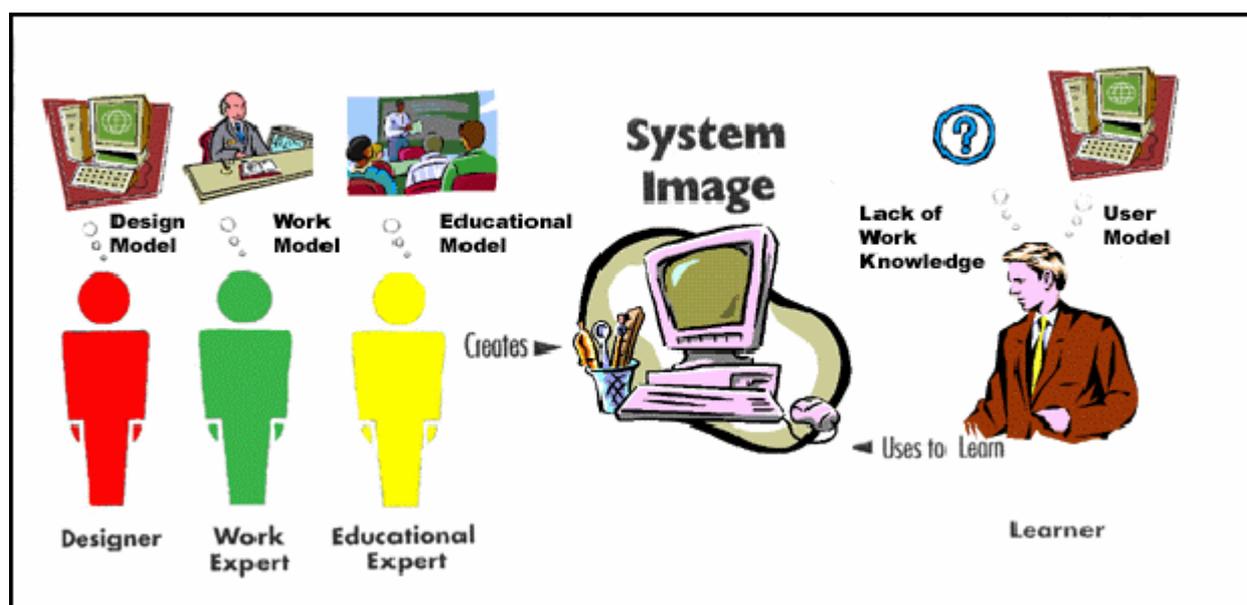
<b>Dimensioni</b>	<b>Learner-Centred Design</b>
Utenti	Discenti: novizi in un dato work practice
Obiettivo principale	Progettare tool che supportano i discenti nello sviluppare una migliore conoscenza di un nuovo work practice
Gap concettuale da considerare	<i>Golfo di esperienza</i> fra discente e work practice (senza ignorare il golfo di esecuzione e di valutazione)
Approccio teorico per colmare il gap concettuale	Applicare teorie di apprendimento (costruttiviste e socio-costruttiviste)

<p>Requisiti di un tool LC</p>	<ol style="list-style-type: none"> <li>1. Supportare i discenti nella partecipare ai nuovi work practice, incorporando nel tool scaffolding relativi al work practice</li> <li>2. Supportare i diversi tipi di discenti: livelli di esperienza, stili di apprendimento, formazione di base, età, ecc.</li> <li>3. Attenuare e progressivamente eliminare lo scaffolding a mano a mano che il discente acquisisce esperienza nel work practice</li> </ol>
<p>Implicazioni per il progetto</p>	<p>Progettare tool che supportano i discenti nell'eseguire task autentici del work practice e che espongono gradualmente i discenti alla cultura (tool, task, linguaggio) del work practice</p>

**Tabella 1:** Dimensioni del progetto LC

Un team di progetto LC ha bisogno di avere una conoscenza dettagliata del work practice, cioè i progettisti hanno bisogno di avere un buon *work model*, una connessione tra il work practice e l'esperienza necessaria per impegnarsi nelle attività di tale work practice. Un buon work model da solo, comunque, non è sufficiente per un buon tool LC. Poiché il discente è novizio, il work model non può essere presentato così com'è, perché il discente non conosce il lavoro. Per risolvere questo problema, il team di progetto ha bisogno di concettualizzare il work model in maniera tale che il discente possa apprendere. Per questo motivo, il team di progetto ha anche bisogno di un *educational model* che descriva come il work model dovrebbe essere concettualizzato per meglio facilitare la costruzione di conoscenza del discente. Di conseguenza, il team di progetto LC ha bisogno di *work experts* (professionisti di un work practice) per studiare e analizzare il work practice e per creare un buon work model, e di *educational experts* (insegnanti, ricercatori) per distinguere i tipi di comunicazione con i discenti e guidarli nell'effettuare il cambiamento concettuale da discente a esperto.

Lo sviluppo dei work model e degli educational model è vitale. La collaborazione dei work expert e degli educational expert è necessaria nel processo di sviluppo di un tool LC per assistere il progettista, che ha la responsabilità sia di progettare un sistema usabile sia di realizzare le indicazioni dei work expert e degli educational expert (Figura 1).



**Figura 1:** Modelli concettuali e ruoli nel team Learner-Centered Design

### 3. **Progettazione Learner- Centred**

Certamente molte delle tecniche tradizionali del progetto di software possono essere utilizzate in un progetto LC. Tuttavia, abbiamo anche bisogno di metodi specifici per l'approccio LC, che ci aiutino ad identificare e valutare le strategie di scaffolding che possono essere utilizzate.

Nel tradizionale approccio user-centered, i requisiti del software definiscono quali funzioni il software deve avere per aiutare gli utenti a completare il loro lavoro in modo efficiente. Nel nuovo approccio LC, abbiamo bisogno di considerare altri tipi di requisiti. Precisamente, un "requisito LC" è un'area identificata in cui il discente ha bisogno di supporto per una certa attività di lavoro [Qui01]. Per determinare i requisiti di un sistema LC, il team di progetto ha bisogno ancora di guide e metodi che lo aiutino loro a determinare:

- fino a che punto il discente può impegnarsi nell'attività
- perché un discente ha bisogno di supporto per impegnarsi nell'attività
- che tipo di conoscenza manca al discente
- che tipo di supporto potrebbe essere utile al discente per impegnarsi nell'attività

La collaborazione con gli *educational experts* può aiutare a identificare metodi per definire requisiti per LCD in modo più formale e metodologico.

Molte delle tecniche tradizionali del progetto user-centered possono essere utilizzate in LCD. Tuttavia, abbiamo bisogno di metodi specifici per LCD, che ci aiutino a scoprire le difficoltà che i discenti incontreranno, quando lavoreranno nel nuovo work practice.

Preece et al. [Pree94] descrivono due importanti fasi per il progetto del software:

1. la fase di tipo concettuale: i progettisti determinano le strategie per individuare i requisiti del software;
2. la fase di tipo fisico: i progettisti determinano le istanze fisiche delle strategie concettuali del software.

Nell'approccio learner-centered, nella fase di tipo concettuale si definiscono anche le strategie concettuali dello scaffolding che saranno utilizzate per individuare i requisiti del tool LC; nella fase di tipo fisico si implementano le strategie concettuali dello scaffolding nel tool LC in modo tale che i discenti possono capire ed eseguire l'attività supportata.

Si presenta, quindi come una necessità di fatto, la definizione e applicazione di linee guida e di standard che permettano la realizzazione di uno strumento non solo usabile dal punto di vista dell'organizzazione, reperibilità e accessibilità dei contenuti, ma anche efficace dal punto di vista didattico. Bisogna sviluppare strategie efficaci di scaffolding che siano da guida rispetto alle seguenti problematiche:

1. Che tipo di scaffolding utilizzare per i diversi discenti
2. Che tipo di scaffolding può essere utile per attrarre e motivare i discenti nel loro percorso di apprendimento; a questo scopo, i ricercatori informatici sono supportati da psicologi, insegnanti, che cercano di fornire strategie utili che supportino la motivazione del discente.
3. Quale è il modo migliore di rimuovere lo scaffolding: ci sono ancora molti problemi aperti riguardo la rimozione dello scaffolding, riguardo chi dovrebbe eliminare lo scaffolding, se il discente, l'insegnante, il sistema o una combinazione dei tre; quale è il meccanismo migliore per rimuovere gli scaffold, se le preferenze del discente o il controllo automatico del sistema, e così via.

### 4. **Usabilità del software LC**

I tool LC hanno bisogno di essere usabili, un team LC certamente ha bisogno di osservare i principi di usabilità quando progetta software didattico, poiché certamente ad un tool non usabile mancheranno alcuni benefici che potrebbero essere indispensabili per i discenti. L'approccio learner-centered ha però anche bisogno di metodi di valutazione e di criteri che valutano i differenti effetti cognitivi risultanti dall'uso della tecnologia [Sal91].

L'approccio user-centered si focalizza sull'efficienza, affinché gli utenti possano completare i loro compiti facilmente. Se un tool LC è troppo user-centered potrebbe essere troppo facile per il discente completare il suo compito. In questo caso, il discente potrebbe non essere forzato ad

applicare lo sforzo cognitivo necessario per imparare il nuovo work practice, secondo quelli che sono i principi della teoria costruttivista. Per esempio, se consideriamo i software “wizard”, essi non si possono definire una buona tecnica LC, perché questi software fanno il lavoro che in realtà dovrebbe essere fatto dai discenti. I progettisti hanno bisogno di conoscere meglio come progettare tool usabili che permettono ai discenti di impegnarsi attivamente in un work practice.

In particolare, la valutazione di un tool LC deve avvenire in modo diverso dalla valutazione di un software user-centered. Nei tradizionali sistemi user-centered, i progettisti si focalizzano sulla valutazione dell'uso del tool e considerano criteri di valutazione, quali il numero di errori che si commettono utilizzando il tool, il tempo impiegato per eseguire un task, ecc. Nella valutazione dei sistemi LC, c'è bisogno di valutare i singoli scaffold e le strategie di scaffolding, ma anche l'efficacia pedagogica totale del software, in modo tale da verificare quanto bene un discente apprende il work practice dopo aver usato il software.

#### 4.1 L'usabilità del software

Attualmente è ampiamente riconosciuto che l'usabilità è un fattore cruciale della qualità delle applicazioni interattive. Differenti concetti sono usualmente associati alla nozione di usabilità. La definizione fornita da J. Nielsen si basa su una descrizione dettagliata degli attributi dell'usabilità [Nie93]. Nielsen propone un modello nel quale l'usabilità è presentata come uno degli aspetti caratterizzanti l'accettabilità del sistema da parte degli utenti finali. L'usabilità non è una proprietà mono-dimensionale del sistema, ma è caratterizzata da cinque attributi: *facilità d'apprendimento*, cioè la facilità di apprendere le funzionalità e il comportamento del sistema; *facilità d'uso*, cioè il livello di produttività raggiungibile, dopo l'apprendimento del sistema; *facilità di memorizzazione*, cioè la facilità di ricordare le funzionalità del sistema, in modo che l'utente casuale può ritornare al sistema dopo un periodo di inattività, senza aver bisogno di capire nuovamente come utilizzarlo; *basso livello di errori*, cioè la capacità del sistema di aiutare gli utenti a non commettere errori durante l'uso, e nel caso si verificassero, dare la possibilità all'utente di risolvere facilmente; *soddisfazione dell'utente* che valuta quanto l'utente gradisce il sistema. Quest'ultimo attributo non deve essere sottovalutato in quanto un sistema gradevole da usare aumenta la produttività dell'utente.

Un'altra definizione è fornita dallo standard ISO 9126 (“Software Quality Product Evaluation: Quality Characteristics and Guidelines for their use”) e riflette le prospettive dell'ingegneria del software sulla qualità dei prodotti software [ISO92].

Nell'ISO 9126, la qualità generale di un prodotto software è decomposta in due categorie di criteri:

1. i criteri che definiscono il concetto di *qualità in uso*, in termini di qualità del prodotto che sono esterne al prodotto software e che sono direttamente collegate al contesto d'uso, al profilo utente e alle motivazioni per cui il prodotto è usato. Questo concetto è fortemente collegato alla “qualità del prodotto dal punto di vista dell'utente”. In questi criteri si definiscono fattori come: l'*efficacia* (la completezza e l'accuratezza con cui un utente raggiunge i suoi obiettivi); la *produttività* (il rapporto tra risorse e efficacia); la *sicurezza* (la quantità di rischio per la salute dell'utente); la *soddisfazione* (il livello di contentezza che un utente ha durante e dopo l'uso del sistema).
2. i criteri che definiscono le *proprietà intrinseche* di un prodotto, che possono essere analizzate prima che il prodotto sia rilasciato e indipendentemente dalla situazione d'uso. I fattori che vengono inclusi in questi criteri sono: la *funzionalità* (la quantità di funzioni contenute in un prodotto rilasciato); l'*affidabilità* (la capacità di un prodotto di mantenere i suoi livelli di performance sotto condizioni stabilite per un determinato periodo di tempo); l'*usabilità* (la misura di quanto un prodotto è adatto, facile e pratico da usare); l'*efficienza* (la quantità di operazioni -e più generalmente, risorse- da utilizzare per raggiungere un obiettivo); la *manutenibilità* (la misura in cui un prodotto è facile da testare, da modificare e estendere); la *portabilità* (abilità di spostare un prodotto da una piattaforma ad un'altra, il livello di conformità agli standard).

Più specificatamente, nello standard ISO 9126 l'usabilità è ulteriormente suddivisa in cinque sotto-caratteristiche: *comprensibilità* (understandability), la capacità intrinseca del prodotto software di rendere di mostrare agli utenti la sua adattabilità ai vari compiti che devono essere svolti nel contesto d'uso; *apprendibilità* (learnability), la capacità intrinseca del prodotto software di aiutare l'utente ad apprendere facilmente le sue funzionalità; *operabilità* (operability), la capacità intrinseca del prodotto software di rendere possibile agli utenti l'esecuzione e il controllo delle sue funzionalità; *attrattività* (attractiveness), la capacità intrinseca del prodotto software di essere gradevole agli utenti; *conformità* (compliance), la capacità del prodotto software di aderire a standard, convenzioni e linee guida dell'usabilità.

Questi fattori relativi all'usabilità sono collegati a quelli definiti in un altro standard, ISO 9241 (Ergonomic Requirements for Office Work with Visual Display Terminals) [ISO97], che contiene delle linee guida per il progetto dell'interfaccia utente e fornisce requisiti e consigli che possono essere utilizzati durante il progetto e la valutazione di interfacce utente. L'ISO 9241 rivede e completa la definizione di usabilità data nell'ISO 9126. In questo standard, l'usabilità è intesa come strettamente dipendente da particolari circostanze in cui un prodotto è usato, cioè la natura degli utenti, i compiti che devono svolgere, e l'ambiente sociale e fisico in cui lavorano. Nella parte 11 di questo standard l'usabilità è definita come: "the extent to which a product can be used by specified users achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use". Per effectiveness (efficacia) s'intende accuratezza e completezza con cui utenti specifici raggiungono determinati obiettivi in specifici ambienti. Efficiency (efficienza) si riferisce alle risorse spese in relazione all'accuratezza e alla completezza degli obiettivi raggiunti. Satisfaction (soddisfazione) è definita come comfort e accettabilità del sistema da parte degli utenti. L'usabilità è quindi intesa come un obiettivo di alto livello del progetto di un sistema. Possiamo concludere che entrambi i concetti di qualità d'uso e di usabilità, come definito nell'ISO 9241, includono gli aspetti più significativi generalmente associati all'usabilità dalle comunità HCI.

## 4.2 L'usabilità per e-learning

L'usabilità è relativa all'interfaccia di un sistema interattivo, per mezzo della quale l'utente si relaziona con il *software*. Va dunque sottolineato che l'usabilità ha senso solo in presenza di un utente e di una relazione d'uso e non esiste nel prodotto in sé. Compito degli studi di usabilità è fare in modo che il modello mentale di chi progetta il *software* (*design model*), corrisponda il più possibile al modello mentale del funzionamento del *software* così come se lo costruisce l'utente finale (*user model*). L'utente è messo al centro del processo di analisi e sviluppo.

Nel caso specifico dell'e-learning, progettare un'interfaccia "usabile" significa comporre in un unico disegno metafore di interazione, immagini e concetti usati per veicolare funzioni e contenuti sullo schermo, creando un sistema di navigazione che non disorienti l'utente. Un software per formazione dovrebbe rappresentare un'esperienza gratificante per il discente dovrebbe possedere le seguenti caratteristiche:

- essere interattivo e fornire notevoli informazioni di ritorno (*feedback*);
- avere obiettivi specifici e procedure stabilite;
- essere motivante;
- comunicare una continua sensazione di sfida (non così difficile da creare un senso di frustrazione, né così facile da annoiare);
- dare la sensazione di un coinvolgimento diretto;
- fornire strumenti appropriati al compito e all'utente;
- evitare distrazioni e fattori di disturbo che interrompano il flusso del lavoro.

Da qui la necessità di costruire interfacce chiare e coerenti, che permettano una navigazione semplice ed efficace, che mantengano quello che promettono e non mettano mai in situazioni da cui non si sappia come uscire. In altre parole, nella costruzione di un percorso didattico che sia efficace e motivante bisogna puntare alla semplicità e concentrarsi sulle esigenze e sugli obiettivi degli utenti, invece che su "effetti speciali" fini a se stessi. Sarebbe opportuno fornire linee guida operative per la progettazione e, secondo quanto definito nel "Human-Centred Design Process for

Interactive Systems” (ISO 13407) [ISO98], ogni alternativa di progetto deve essere valutata il prima possibile con i potenziali utenti del prodotto stesso.

L'obiettivo del produttore di sistemi di e-learning è quello di assicurare che i prodotti *software* siano caratterizzati da: brevi tempi di apprendimento, rapida esecuzione dei compiti, percentuale di errore tendente a zero, facilità nel ricordare le istruzioni di base, alta soddisfazione dell'utente.

Data la natura remota dell'e-learning, l'usabilità gioca un ruolo vitale per il suo successo. Un sistema di e-learning poco usabile ostacola l'apprendimento del discente, il quale sarebbe costretto a spendere più tempo nel cercare di capire come funziona il software piuttosto che nel cercare di capirne il contenuto. La valutazione di usabilità dei sistemi di e-learning potrebbe essere usata per isolare i problemi di apprendimento in particolari software di e-learning e proporre metodi per correggere questi errori [Won03].

L'usabilità di un sistema di e-learning è un concetto difficile da definire, date le diverse dimensioni e fattori che essa include. Studi recenti hanno dimostrato che i maggiori produttori di sistemi di e-learning non fanno test di usabilità sui loro prodotti, probabilmente perché i clienti non hanno modo di valutare il grado di usabilità del prodotto [Fel02]. D'altra parte, malgrado l'enorme quantità di denaro spesa sui prodotti per l'e-learning, nessun produttore di software verifica se il corso che ha sviluppato sia usabile o comunque utile.

Tradizionalmente, i test di usabilità offrono un modo per garantire che un prodotto o un pezzo di software *funzioni*, cioè verificano se questo prodotto ha senso ed è facile da usare per gli utenti che ci devono interagire. In realtà, il primo ostacolo che si incontra nel verificare l'usabilità di un sistema di e-learning è quello di stabilire un insieme di principi, basati su risultati di ricerca accademica o industriale, in base ai quali stabilire se il sistema davvero *funziona* oppure no. Ma ciò ancora non esiste.

Gli standard e i test di usabilità, che hanno trovato la loro rispettabilità per l'e-commerce ed altre applicazioni, hanno bisogno di essere rivisti prima di poter essere applicati all'e-learning: per esempio, gli strumenti per rendere navigabile e attraente un sito per prenotare un volo non possono essere applicati direttamente ad un corso di biologia on-line.

Norman afferma che per l'e-learning, “L'apprendibilità del contenuto è più importante dell'usabilità”. E continua: “Molte tecniche utilizzate dalla comunità HCI, quali il learner-centered, il progetto iterativo, e la valutazione possono essere usate per favorire l'apprendimento.”[Qui02].

Valutare l'apprendimento può portare i professionisti dell'usabilità fuori dalla loro originaria area di lavoro. Per rendere il loro lavoro efficace, essi dovrebbero collaborare con esperti di pedagogia, in modo da poter stilare una serie di linee guida pedagogiche che siano di aiuto alla valutazione dell'efficacia pedagogica del software.

### 4.3 Valutazione di usabilità per sistemi di e-learning

Tra i vari approcci di valutazioni di usabilità, i metodi di ispezione sono molto popolari. Il loro vantaggio principale è comunque il risparmio di costi: essi non coinvolgono utenti né richiedono attrezzatura speciale o disponibilità di laboratori [Nie93, Nie94]. Inoltre, gli esperti possono scoprire un'ampia gamma di problemi e possibili errori in un sistema complesso in un intervallo di tempo limitato. Per questo motivo, i metodi di ispezione sono stati ampiamente utilizzati negli ultimi anni, specialmente negli ambienti industriali [Nie94a], dato che l'industria è ampiamente interessata a metodi efficaci che possono fornire buoni risultati pur essendo poco costosi.

I metodi d'ispezione di usabilità coinvolgono solo i valutatori esperti, che ispezionano l'interfaccia utente con lo scopo di trovare possibili problemi di usabilità, fornendo giudizi basati sulla loro conoscenza, e facendo raccomandazioni per risolvere i problemi e migliorare l'usabilità dell'applicazione. I metodi di ispezione di usabilità sono più soggettivi, avendo una forte dipendenza dagli ispettori 'skills'. Tra i metodi di ispezione, possiamo includere la valutazione euristica, cognitive walkthrough, l'ispezione formale di usabilità, le ispezioni sulla base di linee guida aziendali o di standard [Nie94].

La valutazione euristica è il metodo più informale; coinvolge l'esperto di usabilità che analizza gli elementi di dialogo dell'interfaccia utente per valutare se sono conformi ai principi di usabilità, usualmente detti euristiche, da qui il nome del metodo. Essa prescrive di richiedere l'analisi del

sistema da parte di un piccolo numero di esperti i quali valutano l'interfaccia, utilizzando una lista di principi di usabilità riconosciuti, le euristiche.

Alcune ricerche hanno mostrato che la valutazione euristica è una tecnica di valutazione dell'usabilità molto efficiente, con un alto rapporto costo benefici, e per questo è riferita come uno dei cosiddetti metodi di usabilità economici.

D'altra parte, la valutazione euristica, ed in generale i metodi di ispezione, hanno come svantaggio la soggettività e la forte correlazione tra qualità dei risultati ed esperienza del valutatore. Inoltre, un'altra limitazione è che tali tecniche si concentrano sulla valutazione delle interfacce grafiche, vale a dire su aspetti comuni alla maggior parte delle applicazioni interattive. Molti valutatori sono eccellenti nell'analizzare solo certe caratteristiche di un'applicazione interattiva; spesso, però, essi trascurano altre caratteristiche strettamente dipendenti dalla categoria specifica di un'applicazione e/o dal dominio applicativo.

Allo scopo di identificare tecniche di valutazione di usabilità efficaci, sistematiche e contenute nei costi, negli ultimi anni è stata definita una metodologia di valutazione che è stata utilizzata soprattutto per applicazioni ipermediali.

SUE (Systematic Usability Evaluation) è una metodologia di valutazione dell'usabilità, che combina metodi di ispezione con metodi empirici in modo sistematico, sfruttando le caratteristiche migliori di entrambi e riducendo gli svantaggi [Mat02]. Il processo di valutazione considera attributi di usabilità più specializzati, in grado di catturare le caratteristiche di un'applicazione. SUE prescrive che ogni processo di valutazione inizi dall'ispezione dell'applicazione. Viene proposta una nuova tecnica di ispezione che si basa sull'uso di Abstract Task (AT), cioè pattern di valutazione che indicano gli aspetti dell'applicazione su cui focalizzare l'analisi.

Dopo l'ispezione, se necessario, può anche essere effettuato un test con utenti reali per analizzare in modo più dettagliato eventuali aspetti problematici dell'applicazione. Il progetto del test utente è guidato quindi dai risultati dell'ispezione, essendo così più focalizzato ed efficace.

Nella fase preliminare devono essere prese un certo numero di decisioni e devono essere eseguite due importanti attività:

- la definizione di un insieme di criteri da verificare durante la valutazione.
- la definizione di un insieme appropriato di AT da applicare durante la fase di ispezione.

I **criteri** o **principi di usabilità** sono gli attributi in cui l'usabilità può essere scomposta, così da essere meglio analizzata.

Gli *Abstract Task* sono modelli (pattern) di valutazione che forniscono una descrizione dettagliata delle attività che devono essere eseguite dai valutatori durante l'ispezione. Una lista di Abstract Task rappresenta una guida sistematica su come esplorare le caratteristiche rilevanti dell'applicazione, permettendo la standardizzazione dell'attività di ispezione. Molti valutatori sono eccellenti nell'analizzare solo certe caratteristiche di un'applicazione interattiva; spesso, però, essi trascurano altre caratteristiche strettamente dipendenti dalla categoria specifica di un'applicazione e/o dal dominio applicativo. Sfruttare un insieme di Abstract Task pronti per l'uso permette soprattutto a valutatori con poca esperienza di ottenere buoni risultati.

Seguendo l'approccio proposto da SUE, stiamo sviluppando sia dei criteri di usabilità (o euristiche) che degli AT specifici per i sistemi di e-learning.

## 5. **Euristiche per l'e-learning**

Al fine di identificare delle euristiche per e-learning stiamo effettuando un'analisi accurata delle caratteristiche di tali sistemi in modo da identificare quelle che possono essere più critiche ai fini dell'usabilità.

Le euristiche più famose, ma anche quelle più utilizzate nelle valutazioni di usabilità, sono le dieci euristiche di Jacob Nielsen [Nie93]. Queste possono essere usate per valutare software di e-learning, ma noi riteniamo che sia necessario sviluppare un insieme di euristiche specifiche per questi tipi di software.

Il nostro approccio di estendere le euristiche di Nielsen per renderle adatte ad uno specifico dominio applicativo, come già illustrato nella descrizione dell'ispezione basata su AT è stato adottato da altri gruppi di ricerca relativamente ad altre categorie di applicazioni [Bak02, Man03,

Som03]. Ad esempio, sono state identificate euristiche per la valutazione di sistemi groupware [Bak02] e euristiche per valutare sistemi con display di ampie dimensioni, quali ad esempio quelli usati nelle fiere o altre esposizioni [Som03].

Lo stesso Nielsen ha definito sette euristiche specifiche per l'e-learning, che vengono descritte di seguito [Nie]:

1. **Indicare lo stato del sistema.** Dare all'utente una chiara indicazione di dove essi si trovano. Dare un titolo a tutte le pagine, usare le barre dei menu, nominare i link in modo esplicito, identificare se i link portano a pagine HTML, oppure a finestre pop-up o PDF, includere la data dell'ultimo aggiornamento.
2. **Adeguare il contenuto all'utente.** Capire quale è il target degli utenti. Usare profili utenti per sviluppare il progetto educativo e lo stile.
3. **Dare il controllo di navigazione agli utenti.** Se gli utenti hanno visitato una pagina, ma non ricordano come potervi accedere un'altra volta, allora il progetto è inadeguato. Usare una barra di navigazione per indicare pagine importanti o tool del courseware. Fare in modo che le pagine più usate possano essere raggiunte facilmente. Evitare l'utilizzo di grafici vaghi quando il semplice testo è adeguato.
4. **Essere coerenti e seguire gli standard.** La coerenza assicura un ambiente prevedibile ai discenti. Creare un aspetto grafico statico al software. Per esempio, se le note di un argomento sono sempre presentati in una tabella gialla, i discenti sapranno facilmente come trovare aiuto. Servirsi di convenzioni standard già utilizzate.
5. **Usare un progetto minimalista.** Optare per un progetto chiaro e pulito. Sviluppare e seguire un architettura d'informazione per il software. Evitare confusione. Considerare di avere un designer grafico che riveda il sito di e-learning.
6. **Prevenire gli errori.** Cercare di essere esente da errori. Controllare se ci sono errori di scripting e se i link sono validi. Assicurarsi che la sottomissione dei form funzioni appropriatamente, che i tool interattivi funzionino e che gli attachment siano effettivamente scaricabili. Includi chiare istruzioni sulla sequenza di azioni da seguire. I consigli degli studi on-line e gli avvisi sul corso durante il corso sono particolarmente utili ai nuovi discenti on-line, che hanno bisogno di essere guidati.
7. **Aiutare gli utenti a riconoscere, diagnosticare e correggere gli errori.** I discenti probabilmente incontreranno dei problemi. Fornire documentazione, e strumenti per risolvere potenziali problemi, collegamenti al supporto tecnico e contatti per le informazioni. Le FAQ sono utili. Un progetto solido e le facilitazioni con buoni principi di usabilità assicurano che i discenti abbiano un'esperienza di apprendimento di successo e piacevole.

In base alla nostra esperienza di lavoro su sistemi di sistemi di tutoring intelligenti [Cos03a, Cos03b, Cos01, Ros95, Ros02] abbiamo formulato quattro euristiche che andrebbero aggiunte a quelle di Nielsen:

- **Scegliere accuratamente il contenuto informativo.** Fornire ai discenti informazione adeguata e coerente con il dominio applicativo del sistema. Evitare di includere contenuto che potrebbe forviare la conoscenza del discente dal nuovo work practice.
- **Organizzare il contenuto informativo secondo metodologie di apprendimento appropriate.** Organizzare il contenuto delle singole pagine scegliendo la collocazione e la combinazione dei media che meglio consenta di veicolare l'informazione da insegnare, senza distrarre l'utente.
- **Curare con attenzione il percorso di apprendimento.** Organizzare l'informazione e intervallarla con opportune aree di auto-valutazione, in modo da consentire al sistema di raggiungere gli obiettivi didattici che si propone.
- **Fornire strumenti di supporto all'apprendimento.** Prevedere tool di supporto (scaffolding) che aiutino il discente a prendere decisioni quando si trova in una situazione di difficoltà e/o non sa come procedere.

## 6. *Analisi di utenti di sistemi LC*

L'obiettivo dell'unità di Bari è approfondire la conoscenza riguardo le problematiche che presentano l'uso dei sistemi di e-learning da parte di utenti finali in una situazione reale, in modo da individuare più dettagliatamente quali sono i punti critici e, quindi, definire sia i criteri da usare nella nostra tecnica di valutazione sia i pattern di valutazione.

In questa fase stiamo effettuando un'osservazione diretta di un gruppo di utenti che utilizzano un sistema di e-learning in una situazione reale. I partecipanti allo studio sono dieci studentesse del Master ProFAD (Progettisti Formazione a Distanza) che si sta tenendo in questo periodo presso il Dipartimento di Informatica dell'Università degli Studi di Bari.

Poiché una semplice osservazione raramente è sufficiente per determinare se il sistema risponde ai requisiti degli utenti, la tecnica che abbiamo adottato è il *thinking aloud* che consiste nell'osservare gli utenti mentre usano il sistema e descrivono cosa pensano che stia succedendo, perché eseguono certe azioni e cosa stanno cercando di fare. In questo modo si riesce ad esplorare il processo di decisione seguito dall'utente nell'esecuzione di un task. Generalmente viene chiesto agli utenti di eseguire un insieme predeterminato di task, mentre il valutatore annota le azioni eseguite dall'utente e i suoi commenti.

Alla fine dell'osservazione, si procederà ad intervistare i partecipanti, in modo da raccogliere altre informazioni riguardanti l'usabilità dal punto di vista dell'utente finale.

L'intervista risulta essere un mezzo flessibile ed efficace poiché consente di osservare gli utenti, di spiegare meglio le domande se queste sono difficili, di dettagliarle in base alle risposte dell'utente, così da ottenere informazioni più dettagliate e specifiche; inoltre, le domande possono essere ripetute se sono state fraintese e gli utenti possono dilungarsi nella risposta e fornire, di conseguenza, maggiori informazioni.

### **Bibliografia**

- [Bak02] Baker, K., Greenberg, S., and Gutwin, C. Empirical development of a Heuristics evaluation methodology for shared workspace groupware. In ACM Conference on Computer Supported Cooperative work (CSCW '02), New Orleans, LA, November 2002.
- [Bel96] Belamy, R.K.E. Designing educational technology: Computer-mediated change. In B.A. Nardi (Ed.), context and Consciousness: Activity Theory and Human-Computer Interaction. Cambridge, MA: MIT Press, 1996.
- [Bro89] Brown, J.S., Collins, A., and Duguid, P. Situated cognition and the culture of learning. Educational Researcher, Vol. 18, pp. 32-42, 1989.
- [Cos01] Costabile M.F., De Angeli A., Roselli T., "Experimental Evaluation of an Intelligent Tutoring Hypermedia", Proc. of PC-HCI 2001, Patras, Greece, December 7-9, 2001, Typorama Publications, pp. 135-139.
- [Cos03a] Costabile M.F., De Angeli A., Roselli T., Lanzilotti R., Plantamura P. Evaluating the Educational Impact of a Tutoring Hypermedia for Children. Information Technology in Childhood Education Annual, to appear.
- [Cos03b] Costabile M.F., De Angeli A., Roselli T., Lanzilotti R., Plantamura P., "Does Hypermedia Really Work for Tutoring Children", IEEE Multimedia, April-June 2003. pp. 65-69.
- [Fel02] Feldstein, M. What is "usable" e-learning?. E-learn Magazine, Vol. 2002, Issue 9, September 2002.
- [ISO92] ISO (International Organization for Standardization), ISO 9126: Software Quality Product Evaluation: Quality Characteristics and Guidelines for their use, 1992.
- [ISO97] ISO (International Organization for Standardization), ISO 9241: Ergonomics Requirements for Office Work with Visual Display Terminal (VDT) - Parts 1-17, 1997.
- [ISO98] ISO (International Organization for Standardization), ISO 13407: Human-Centered Design Process for Interactive Systems, 1998.

- [Man03] Mankoff, J., Dey, A., Hsieh, G., Kients, J., Lederer, S., and Ames, M. Heuristic Evaluation of ambient display. In ACM Conference on Human Factors and Computing Systems, (CHI '03), Ft. Lauderdale, FL, 2003.
- [Mat02] Matera, M., Costabile, M.F., Garzotto, F., and Paolini, P. SUE Inspection: an Effective Method for Systematic Usability Evaluation of Hypermedia". IEEE Transactions on Systems, Man and Cybernetics- Part A, Vol. 32, N. 1, pp. 93-103, 2002.
- [May99] D.J. Mayhew, "The Usability Engineering Lifecycle", San Francisco, CA: Morgan Kaufman Press, 1999.
- [Nie] <http://www.du.edu/~rdreylin/usability.html>
- [Nie93] Nielsen, J. Usability Engineering. Academic Press, Cambridge, MA, 1993.
- [Nie94] Nielsen, J., and Mack, R.L.. Usability Inspection Methods, John Wiley & Sons, New York, 1994.
- [Nie94a] Nielsen, J.. Guerrilla HCI: Using Discount Usability Engineering to Penetrate Intimidation Barrier, In Cost-Justifying, Usability (R.G. Bias, D.J. Mayhew, Eds) Academic Press, Also available at the URL [http://www.useit.com/papers/guerrilla\\_hci.html](http://www.useit.com/papers/guerrilla_hci.html), 1994.
- [Nor86] Norman, D.A. Cognitive engineering", in D.A. Norman and S.W. Draper (Eds.), User Centered System Design. Hillsdale, NJ: Lawrence Erlbaum Associates, 1986.
- [Pap93] Papert, S. The children's machine: Rethinking school in the age of the computer. New York: Basic Books, 1993.
- [Pia54] piaget, j. The construct of reality in the child. New York: Basic Books, 1954.
- [Pree94] Preece, J., Rogers, Y., Sharp, H., and Benyon, D. Human-Computer Interaction. Reading, MA: Addison-Wesley, 1994.
- [Qui01] Quintana, C., Carra, A., Krajcik, J., and Elliot, S. Learner-Centred Design: Reflections and new Directions. Human-Computer Interaction in the New Millennium. In Carroll (Ed.) ACM Press, New York: Addison-Wesley, 2001.
- [Qui02] Quigley, A. Usability-tested e-learning? not until the market requires it. Special to eLearn Magazine. Volume 2002 , Issue 2, February 2002.
- [Ros95] Roselli, T., Artificial Intelligence can Improve Hypermedia Instructional Technologies for Learning. ACM Computing Surveys, Vol. 27, No. 4, pp.624-626, December 1995.
- [Ros02] Roselli, T., Faggiano, E., Plantamura, P., Rossano, V. A WWW-Based Cooperative Learning System and its Effects on Students' Achievement. International Conference on computers in Education (ICCE 2002), Auckland, New Zealand, December 3-6, pp. 283-287, 2002.
- [Sal91] Salomon, G., Perkins, D.N., and Globerson, T. Partners in cognition: Extending Human Intelligence with Intelligent Technologies. Educational Researcher, pp. 2-9, April 1991.
- [Sol94] Soloway, E., Guzdial, M., and Hay, K.H. Learner-Centered Design: the Challenge for HCI in the 21st Century, Interactions, Vol. 1, N. 2, 1994.
- [Sol96] Soloway, E., and Pryor, A.. The Next Generation in Human-Computer Interaction. Communication con ACM, Volume 39, n. 4, 1996.
- [Sol96a] Soloway, E., Jackson, S.L., Kleim, J., Quintana, C., Reed, J., Spitulnik, J., Stratford, S.J., Studer, S., Eng, J., and Scala, N. Learning Theory in practice: Case studies in learner-centered design. Human Factors in computing systems: CHI'96 Conference Proceedings. Reading, MA: Addison-Wesley, 1996.
- [Som03] Somervell, J., Wahid, S. and McCrickard, D.S. Usability Heuristics for Large Screen Information Exhibits, In Proc. Of Human-Computer interaction – Interact '03, Zurigo, Svizzera, pp. 904-907, 2003.
- [Won03] Wong, B., Nguyen, T., Chang, E., Jayaratna, N. Usability Metrics for e-Learning. Accepted to Workshop on Human Computer Interface for Semantic Web and Web Applications. Part of the International Federated Conferences (OTM '03).

