Corso di Laurea in
Informatica e Comunicazione Digitale

a.a. 2006 - 2007

Interazione Uomo-Macchina

Fiorella de Rosis

Programma

- ✰ Introduzione ai metodi per la progettazione di interfacce
- Analisi e caratterizzazione dell'utenza e dei compiti svolti
- ***Principi di usabilità***
- Linguaggi grafici per l'interazione
- Metodi formali per la specifica dell'interazione
- Metodi di valutazione
- Adattamento dell'interazione all'utente e al contesto
- Interazione in linguaggio naturale e multimediale
- Agenti Animati

3. Principi di Usabilità

- Quando una interfaccia si può dire 'usabile'
- Principi per la progettazione di interfacce usabili
- Metodi di valutazione dell'usabilità

3.1 Definizione di Usabilità*

*"**Usability** is the effectiveness, efficiency and satisfaction with which specified users can achieve specified goals in a particular environment;*

- ***effectiveness** is the accuracy and completeness with which Users achieve specific goals;*
- ***efficiency** is the accuracy and completeness of goals achieved in relation to resources expended;*
- ***satisfaction** is the comfort and acceptability of using the system".*

(*Definizione ISO, International Institute of Standard Office)

Secondo la definizione dell'ISO, una Interfaccia si può dire 'Usabile' quando è:

- *facile da imparare*, per chi usa il sistema la prima volta
- *veloce da usare*, per chi ha già esperienza nell'uso del sistema
- *efficace* nel raggiungere gli obiettivi che il sistema si pone
- e *piacevole* da utilizzare per gli utenti

Esercizio 3.1

Quali degli aspetti di usabilità che hai visto sono, a tuo avviso, più importanti:

- nella fase di *apprendimento* dell'uso di un sistema
- nella fase di *uso* del sistema, da parte di Utenti '*occasionali*'
- nella fase di *uso* del sistema, da parte di Utenti '*frequenti*'
- nei sistemi il cui uso è *opzionale* (non obbligatorio)

Per i nostri Casi di Studio, quali sono gli aspetti più importanti, secondo te?

Come Costruire una Interfaccia Usabile?

Traducendo in modo appropriato, nell'Interfaccia,
i risultati della
Task Analysis
e dell'
Object Analysis

e, se gli Utenti hanno esigenze diverse,
progettando eventuali forme di *adattamento*.

Partiamo da un Esempio

Un prodotto Microsoft per produrre immagini
(Paint)

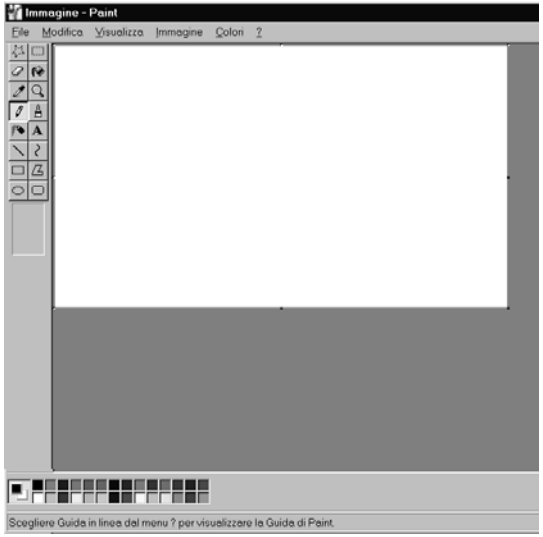
Proviamo a fare il 'processo inverso':
quale task analysis e object analysis ci sono,
dietro l'interfaccia di un prodotto commerciale?

Main Window di Paint

Toolbar in alto: *menu di bottoni*: task che consentono di manipolare il file-immagine, l'immagine o sue parti

Toolbar verticale: *menu di icone:* i task che consentono di 'costruire' l'immagine

Due Toolbar in basso: tavolozza dei colori e Barra di stato



Task Associati alla Main Toolbar di Paint

File

- Nuovo
- Apri
- Salva
- Salva con nome
- ---
- Anteprima di Stampa
- Imposta pagina
- Stampa

operazioni sul file-immagine

Modifica

- Annulla
- Ripeti
- Taglia
- Copia
- Incolla
- Cancella area selezionata
- Seleziona tutto
- ---
- Copia su
- Incolla da

operazioni su parti dell'immagine

Visualizza

- Casella degli strumenti
- Tavolozza
- Barra di stato
- Barra degli strumenti di testo
- ---
- Zoom
- Visualizza bitmap

operazioni sulla finestra di lavoro

Colori

- modifica colori

operazioni sulla proprietà 'colori'

Immagine

- Capovolgi/ruota
- Allunga/inclina
- Inverti colori
- Attributi
- Cancella immagine

operazioni sull'intera immagine

Oggetti e Funzioni in Paint

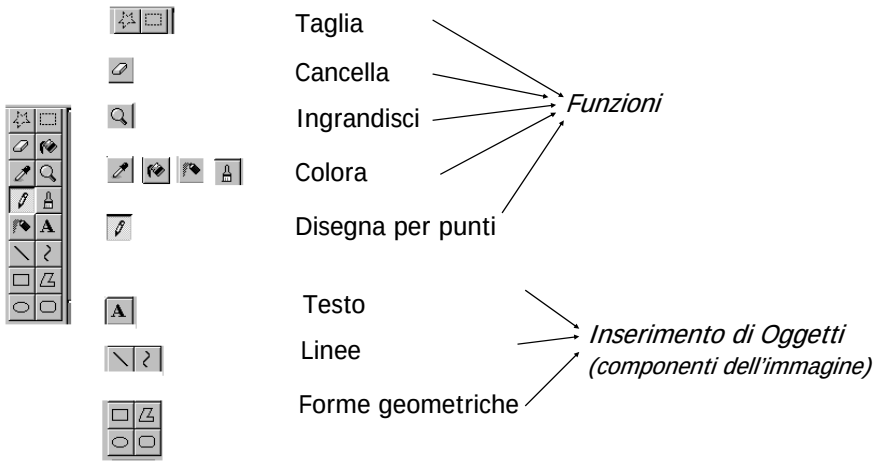
Oggetti:

- La *finestra di lavoro*, con i suoi diversi control (part-of)
- Il *file*
- L'*immagine*
- Le diverse *componenti dell'immagine forme* e *testo* (part-of) con i loro *attributi* (posizione, forma, colore, ...)

Funzioni:

- visualizza, non visualizzare (per i control nella finestra di lavoro)
- crea, apri, salva, stampa, invia (per i file)
- modifica orientamento (capovolgi, ruota), forma (allunga, inclina), dimensioni (attributi) o colori dell'immagine
- taglia, copia, incolla, cancella (per le componenti dell'immagine)

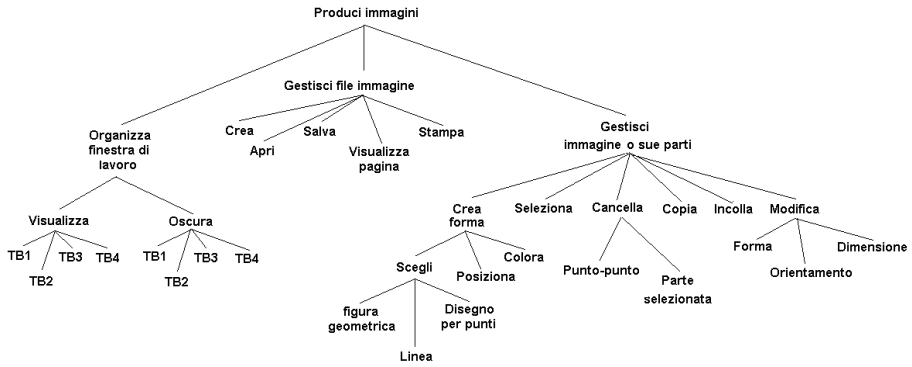
Costruzione dell'Immagine



Quale Logica nel Progetto di Paint?

- *Metafora: il 'tavolo da disegno'* (toolbar verticale)
- *Flessibilità* (task più frequenti con comando da tastiera)
- *Organizzazione dell'informazione in un' unica finestra:*
  con tutti i task sempre disponibili, in ogni fase dell'interazione
- *Organizzazione dei simboli nello spazio:*
  gerarchia dei task ⇒ organizzazione di toolbar
  somiglianza nel significato ⇒ vicinanza
- *Scelta dei simboli:* sia le funzioni che gli oggetti a cui queste si applicano sono rappresentati con 'icone-oggetto':
  una gomma per 'cancellare'
  una lente d'ingrandimento per 'ingrandire'
  un pennello per 'colorare'

Quale Gerarchia di Task in Paint?



Esercizio 3.2

Prova a ricostruire e a rappresentare, con le UAN, alcune relazioni d'ordine fra task in Paint.

Esempio:

Crea forma:= ((Scegli_figura_geometrica | Scegli_Linea | Disegna_per_punti) Posiziona Colora)*

Ci sono esempi di task concorrenti, interrompibili o in indipendenza d'ordine, in Paint?

3.2 Come Costruire Interfacce Usabili?

Ci sono modi diversi,
per tradurre User, Task e Object analysis
in una interfaccia.

Come garantire
che questa traduzione produca una interfaccia 'usabile'?

Quali criteri applicare,
per giudicare se una interfaccia è usabile?

Le otto 'Regole d'Oro' di Shneiderman

- ***Strive for consistency***: use consistent sequences of actions in similar situations, identical terminology in prompts, menus etc, consistent commands throughout.
- ***Enable frequent users to use shortcuts***: abbreviations, special keys hidden commands and macro facilities reduce the number of interactions and the pace of interaction.
- ***Offer informative feedback***: for every operator action, there should be some system feedback.
- ***Design dialogs to yield closure***: the informative feedback at the completion of a group of actions gives the operator the satisfaction of accomplishment and the indication that the way is clear to prepare for the next group of actions.

Le otto 'Regole d'Oro' di Shneiderman

- ***Offer simple error handling:*** the user should not have to retype the entire command, but only need repair the faulty part.
- ***Permit easy reversal of actions:*** this relieves anxiety, since the operator knows that errors can be undone; the units of reversibility may be a single action, a data entry or a complete group of actions.
- ***Support internal locus of control:*** users should be made the initiators of actions rather than the responders.
- ***Reduce short-term memory load:*** displays should be kept simple, frequent window motion should be reduced, online access to command syntax forms etc should be provided.

Proprietà delle Interfacce 'Usabili'

- *Naturalezza*
- *Completezza*
- *Consistenza*
- *Non complessità*
- *Non ridondanza*
- *Assistenza*
- *Flessibilità*

3.2.1. Naturalezza

"Un dialogo è naturale quando non richiede all'Utente di alterare in modo significativo il suo modo comune di eseguire un task".

Il sistema deve comportare il numero minimo possibile di cambiamenti, nel lavoro svolto dall'Utente: deve permettergli di concentrarsi sul lavoro da svolgere, non sull'uso del sistema.

Ad esempio:

i form da compilare devono rispettare il più possibile la struttura dei moduli che l'Utente compilava nella fase precedente all'automazione

Nota: una definizione 'superata' di naturalezza...!!

'Naturalezza' nell'Esempio Medico

Input dei dati relativi alle prescrizioni:
Modulo 'standard' del SSN:

codice				
nome assistito	età			
indirizzo	ASL	prov	sesso	esenzione
prescrizione-richiesta-proposta:	suggerita (S)		integrativa (I)	ricovero (R)
.....				
.....				
	data		timbro e firma del medico	
.....				
.....				

3.2.2. Completezza

"Un dialogo è completo quando permette di effettuare tutte le funzioni previste nella fase di progettazione".

L'interfaccia deve permettere di accedere a tutti task previsti, per ogni categoria di Utente, in ogni fase dell'interazione in cui l'accesso al task è necessario;

ma, inoltre, non deve contenere 'percorsi chiusi' (dead ends),

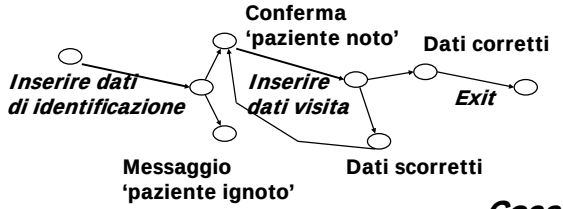
cioè situazioni in cui è impossibile completare il task iniziato oppure accedere a nuove funzioni.

Le verifiche di completezza sono facilitate dalla rappresentazione della dinamica del dialogo sottoforma di 'diagrammi di transizione fra gli stati'.

'Non Completezza' nell'Esempio Medico

Consideriamo il task '*Gestire l'archivio delle cartelle*'.

Supponiamo che l'Utente considerato sia la Segretaria, che deve inserire, alla fine della giornata, i dati relativi alle visite eseguite su diversi pazienti, non sapendo se la cartella relativa ad ognuno di questi pazienti esiste già nell'archivio.



Cosa manca da questo schema che illustra la dinamica del dialogo?

Esercizio 3.3

Fai qualche esempio di 'non completezza',
in una delle applicazioni che conosci.

3.2.3. Consistenza

"Un dialogo è consistente quando l'esperienza acquisita dall'Utente con l'uso di una parte del sistema non viene contraddetta da cambiamenti ingiustificati introdotti in altre parti."

Consistenza *visiva:* la collocazione degli oggetti grafici nello spazio deve essere guidata da regole che valgono per tutte le fasi dell'applicazione

funzionale: task simili devono essere svolti in modo simile

Consistenza *interna:* tra le diverse funzioni svolte da un sistema

esterna: tra le stesse funzioni, svolte da diversi sistemi all'interno di un ambiente (es, Windows)

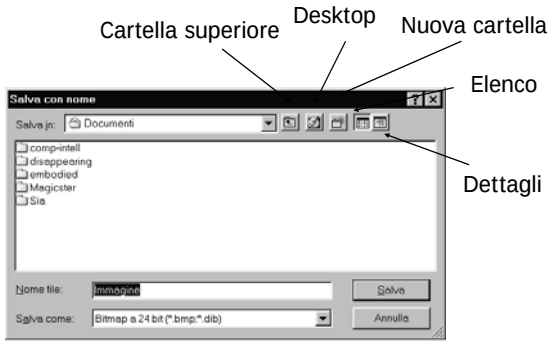
Esempi di inconsistenza che abbiamo visto nella prima Unità

Inconsistenza interna

Inconsistenza esterna

Consistenza interna in Paint

Nei task di 'gestione del file-immagine', le pop-up window che si aprono in corrispondenza delle funzioni di 'apri', 'salva con nome', ecc
hanno tutte lo stesso aspetto:



Nota che questa toolbar permette di definire la 'cartella di lavoro' (nuovo oggetto) e come mostrare i file in essa contenuti

'Non Consistenza' nell'Esempio Medico

Esempi di '*non consistenza visiva*':

- rappresentare la funzione 'trova la cartella di un paziente'
  con icone diverse,
  in diverse fasi dell'interazione;

- collocare il bottone di 'Exit' in posizioni diverse, in diverse fasi;
- ordinare gli elementi di un menu in modo diverso, in diverse fasi;
- assegnare nomi in inglese o in italiano, agli elementi di un menu, in diverse fasi.

'Non Consistenza' nell'Esempio Medico

Esempi di '*non consistenza funzionale*':

- eseguire la funzione 'uscire dal task salvando i dati introdotti', mediante un solo comando (esempio, bottone 'Quit') o mediante due comandi successivi, (esempio, bottone 'Save' + bottone 'Quit'), in fasi diverse;
- imporre un ordine prefissato / libero nell'inserimento degli stessi dati, in fasi diverse.

Esercizio 3.4

Consistenza esterna:
il task 'Salva con nome' in Paint è coerente (funzionalmente e visivamente) con quello di Word?

Fai qualche esempio di inconsistenza visiva
e nella esecuzione task
(interna o esterna),
in uno dei software che conosci

3.2.4. Non Complessità

"Un dialogo è non complesso quando consente di effettuare le funzioni previste con un 'ragionevole' grado di efficienza".

Complessità *visiva:* è funzione del numero di oggetti presenti contemporaneamente in un display

Complessità *funzionale:* dipende dal numero di operazioni elementari che devono essere eseguite per completare un task

'Complessità' nell'Esempio Medico

Confronta i due modi seguenti di compiere il task di 'Ricerca della cartella di un paziente':

- *click sull'icona "Ricerca di una Cartella"* *, nella finestra principale*
- apertura di una finestra 'dati di identificazione', che contiene cognome, nome, data di nascita, numero della cartella;
- *inserimento di 'cognome, nome e data di nascita (oppure di 'numero cartella'); conferma;*
- visualizzazione della prima pagina della cartella.

... oppure...

'Complessità' nell'Esempio Medico (segue)

... oppure

- *click sul bottone "Gestione dell'Archivio", nella finestra principale;*
- apertura di una nuova finestra, con icone 'Ricerca di una Cartella', 'Creazione di nuova cartella', ecc;
- *click sull'icona "Ricerca di una Cartella";*
- apertura di una finestra con la domanda: "Conosci il numero della cartella"?
- *Risposta si/no*
- apertura di una finestra con i soli dati d'identificazione scelti.

Nota il trade-off fra complessità visiva e funzionale!

Esercizio 3.5

Confronta la complessità funzionale di uno stesso task con due tool diversi in uno stesso ambiente.

Ad esempio:
apri un nuovo documento
registralo con un nome
scrivi una riga di testo
inserisci una immagine
salva
stampa
esci

con WinWord e con WinPad

3.2.5. Non Ridondanza

"Un dialogo è non ridondante quando

- *richiede all'Utente di fornire soltanto il minimo indispensabile d'informazione necessaria per attivare ciascuna delle funzioni desiderate;*
- *presenta all'Utente soltanto il minimo di informazioni necessarie*"

'Ridondanza' nell'Esempio Medico

Inserimento dei dati di identificazione seguenti:
Numero di Cartella: 00156789
Data di Nascita ed eta'
Richiesta di conferma per operazioni 'non rischiose' (esempio, "Vuoi davvero salvare questi dati?")

Visualizzazione di informazioni non necessarie:
tutti i dati di identificazione, anziché quelli strettamente necessari, nel display dei dati sulla visita.

Esercizio 3.6

Fai qualche esempio di errori di ridondanza, in un software che conosci.

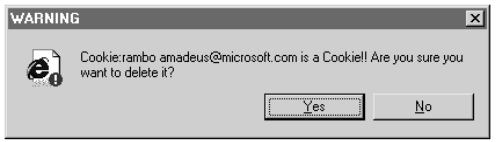
3.2.6. Assistenza

"Un dialogo deve assistere l'Utente fornendogli diversi tipi di aiuto durante l'esecuzione" :

- elenco delle istruzioni disponibili
- messaggi di errore chiari
- richiesta di conferma di quello che il sistema sta per fare (soprattutto se l'operazione è, per qualche ragione, 'pericolosa')
- avviso su quello che il sistema sta facendo (soprattutto per operazioni che richiedono tempi di esecuzione lunghi)

Esempi di messaggi di errore poco chiari

riprendiamo qualche esempio della prima Unità



'Assistenza' nell'Esempio Medico

- Indicazione della funzione svolta da ogni icona, su passaggio del mouse, per rendere comprensibile l'elenco dei comandi effettuabili
- Messaggio del genere "*L'archivio delle cartelle è complesso: l'operazione di ricerca può richiedere un certo tempo*"
- Messaggi del genere: *"Il numero della cartella è errato: deve contenere almeno 7 caratteri"*, oppure *"Non esiste nessun Mario Rossi nell'archivio, ma c'è una Maria Rossi: sei sicuro di non aver sbagliato?"...*
- Il tipico messaggio: *"Sei sicuro di voler cancellare questa cartella?"*

Nota:

tutti i messaggi devono sempre chiudersi su un comando di 'conferma', che verifica che l'Utente abbia effettivamente letto il messaggio!

Esercizio 3.7

Fai qualche esempio di assistenza insufficiente,
in un software che conosci.

3.2.7. Flessibilità

"Una interfaccia è flessibile quando è in grado di essere efficace, efficiente e confortevole anche per Utenti con caratteristiche diverse".

Quando gli Utenti previsti hanno caratteristiche (e quindi, esigenze) diverse,

l'interfaccia deve essere capace di *riconoscere* il tipo di Utente con cui sta interagendo ed *adattare* il suo comportamento di conseguenza.

Vedremo in seguito come è possibile realizzare forme diverse di adattamento nell'interazione.

'Flessibilità' nell'Esempio Medico

In questo Sistema informativo, alcune funzioni vengono eseguite frequentemente (es: la ricerca e l'aggiornamento delle singole cartelle), altre con minore frequenza (es: le statistiche periodiche sull'attività svolta). Alcune richiedono competenze mediche (ad esempio, la codifica delle diagnosi o l'inserimento di dati relativi alla visita), altre soltanto competenze di uso di sistemi computerizzati (che una Segretaria può acquisire facilmente).

Le competenze menzionate possono variare nel tempo e a seconda della tipologia di medici e l'organizzazione dell'ambulatorio.

Realizzare un unico sistema che possa essere utilizzato in entrambe le situazioni organizzative descritte nella Unità 1 richiede la costruzione di un sistema flessibile.

Esercizio 3.8

Descrivi in che modo, nel progettare l'interazione con il Sistema per il tuo Caso di Studio, hai tenuto delle esigenze di:

- *Naturalezza*
- *Completezza*
- *Consistenza*
- *Non complessità*
- *Non ridondanza*
- *Assistenza*

Fai esempi precisi per ciascuna delle caratteristiche elencate.

Indica, inoltre, se e come il sistema che hai progettato contiene elementi di *flessibilità.*

Per Approfondimenti

Le ‘Linee-Guida’ da seguire nella progettazione delle interfacce stabilite dall’International Standard Office sono reperibili (non nei dettagli) su web.

Sullo Standard ISO 9241 per ‘Ergonomic requirements for office work with visual display terminals’:

http://web.tiscali.it/userware/standardiso9241.htm

Sullo standard ISO 13407 per ‘Human-Centered Design Processes’:

http://www.usabilitynet.org/tools/r_international.htm