

Progettazione Interfaccia Utente

Alberto Gianoli

Univ. Ferrara - Corso di Laurea in Informatica

Dove?

- ✧ Pressman, cap. 14

- ✧

Le regole d'oro

- ❖ 3 regole d'oro per guidarci nella progettazione della UI
 1. lascia che il controllo sia nelle mani dell'utente
 2. limita la necessità per l'utente di usare la propria memoria
 3. usa una interfaccia uniforme per tutta l'applicazione
- ❖ queste regole generali si traducono in un insieme di principi che è bene rispettare quando si disegna una UI
- ❖ Mai dimenticare che “A common mistake people make when trying to design something completely foolproof is to underestimate the ingenuity of complete fools” (D. Adams)

Controllo nelle mani dell'utente

- ❖ (fatto vero) durante raccolta requisiti di un grosso progetto, si chiede ad un utente che interfaccia vuole: “.. desidero un sistema che mi legga nel pensiero, sappia quello che voglio fare prima che ne abbia bisogno, deve rendermi facile farlo. Tutto qui”
- ❖ sorridete? Scuotete la testa? La richiesta non è sbagliata: l'utente vuole qualcosa che reagisca alle sue esigenze, lo aiuti a ottenere il risultato
- ❖ L'utente vuole controllare lui il computer, non essere lui sotto il controllo del computer
- ❖ Quando poniamo dei vincoli su una user interface, a chi stiamo semplificando la vita: al programmatore o all'utilizzatore?

Controllo nelle mani dell'utente

- ❖ Definire la modalità di interazione in modo da non costringere l'utente ad azioni inutili o indesiderate
- ❖ Offrire sempre una interazione flessibile
 - ❖ i gusti sono gusti: p.e. c'è chi preferisce scrivere e chi vuole scegliere da un menù
- ❖ Ogni azione deve poter essere interrotta o annullata
- ❖ Prevedere modalità d'uso abbreviate (macro o altro) per utenti esperti se serve svolgere ripetutamente certe azioni
- ❖ Nascondere all'utente casuale i dettagli tecnici
 - ❖ non c'è ragione per cui un utente debba conoscere dettagli interni, o digitare comandi del sistema operativo della macchina

Controllo nelle mani dell'utente

- ❖ Progetta il sistema in modo che consenta la manipolazione diretta degli oggetti che compaiono sullo schermo
 - ❖ l'utente "sente" di essere in controllo se può manipolare quello che gli serve come se fosse un oggetto fisico

Limitare il ricorso alla memoria

- ❖ Più cose l'utente deve ricordare, maggiori saranno anche le probabilità di errore nelle interazioni con il sistema
- ❖ Bisogna non fare molto affidamento sulla memoria dell'utente
- ❖ Ridurre la necessità di memoria a breve termine
 - ❖ se operazione complessa, la UI dovrebbe ridurre la necessità di ricordare le azioni fatte e i risultati ottenuti finora
- ❖ Definire delle impostazioni predefinite di validità generale
 - ❖ dovrebbero essere adatte per l'uso medio, ma poter essere modificate (e resettate)
- ❖ Definire scorciatoie intuitive
 - ❖ short-cut mnemonico associato all'operazione

Limitare il ricorso alla memoria

- * L'aspetto visivo della UI deve essere una metafora del mondo reale
 - * p.e. è un sistema di pagamento? Usa la metafora del libretto degli assegni o del registratore di cassa: l'utente "sfrutta" operazioni già note e si orienta rapidamente
- * Fornire le informazioni in modo progressivo
 - * le informazioni relative a una operazione, un oggetto, ... devono partire da un alto livello di astrazione e raffinarsi quando l'utente manifesta il suo interesse
 - * p.e. in word prima dite che volete sottolineare una frase, poi scegliete il tipo di sottolineatura

Rendere uniforme la UI

"If it looks like a duck, and quacks like a duck, we have at least to consider the possibility that we have a small aquatic bird of the family anatidae on our hands."

Douglas Adams

"Le cose che sembrano differenti dovrebbero comportarsi in modo differente; le cose che sembrano uguali dovrebbero comportarsi nello stesso modo"

- ❖ Tutte le informazioni visuali vanno organizzate secondo uno standard di progettazione che deve essere mantenuto in tutte le situazioni di visualizzazione
- ❖ I meccanismi di input devono essere un insieme limitato e usato uniformemente nell'applicazione
- ❖ I meccanismi di navigazione da operazione a operazione devono essere definiti e implementati in modo uniforme

Rendere uniforme la UI

- ❖ In ogni istante deve essere evidente il contesto in cui ci si trova
 - ❖ in molti casi è anche utile che l'utente sappia come il contesto da cui proviene e le alternative per eseguire una transizione verso una nuova operazione
- ❖ Se possibile, mantenere l'uniformità all'interno di una famiglia di applicazioni
 - ❖ un gruppo di applicazioni (o prodotti) conviene implementino le stesse regole in modo che l'interazione risulti uniforme
- ❖ Se esistono modelli interattivi preesistenti e ben consolidati (cioè gli utenti sono abituati a quelli) non apportarvi modifiche se non per motivi molto importanti
- ❖ p.e. ALT-S salva i file: se lo usate per qualcos'altro, aspettatevi il caos

Modelli per l'analisi e il design delle UI

- * Di una interfaccia utente bisogna considerare 4 viste
 - * il software eng. crea il modello del design (vista centrata sul progetto)
 - * l'esperto "ergonomico" (o i sw. eng.) crea il modello utente (vista centrata sull'utente)
 - * l'utente finale sviluppa un modello mentale (vista centrata sulla percezione)
 - * i programmatori creano un modello dell'implementazione (vista centrata sull'implementazione o "immagine del sistema")
- * Quasi sempre queste viste differiscono, anche in modo sostanziale e i vincoli che ne derivano sono contrastanti
- * Compito del progettista di interfacce è arrivare a un compromesso tra le varie esigenze

Modelli di Utente

- ❖ Per fare una UI efficace, bisogna sapere cosa aspettarsi dagli utenti in termini di conoscenza del sistema o dell'ambito
- ❖ Tipicamente possiamo catalogare l'utente in 3 categorie
 - ❖ **principiante**: nessuna conoscenza sintattica (uso dell'interfaccia) e scarse conoscenze semantiche sull'uso del sistema (senso dell'applicazione: comprensione delle funzionalità)
 - ❖ **utente casuale**: ragionevole conoscenza semantica dell'applicazione, conoscenza sintattica limitata
 - ❖ **utenti costanti**: buona conoscenza semantica e sintattica, sindrome dell'utente evoluto (vogliono shortcut e interazioni abbreviate)

Immagine e percezione

- * La “percezione del sistema” è l’immagine mentale che l’utente finale si crea nella propria testa
 - * dipende molto dal tipo di utente
 - * dipende molto dalla familiarità con il dominio dell’applicazione
- * La “immagine del sistema” comprende sia la manifestazione del sistema (l’aspetto e il comportamento della UI), sia tutte le informazioni di supporto che descrivono sintassi e semantica del sistema
- * L’utente si trova a proprio agio con il sw e lo usa in modo efficace quando il modello mentale dell’utente e quello dell’implementazione tendono a coincidere
 - * il progetto deve tenere conto dell’input proveniente dal modello di utenza

Il processo di progettazione della UI

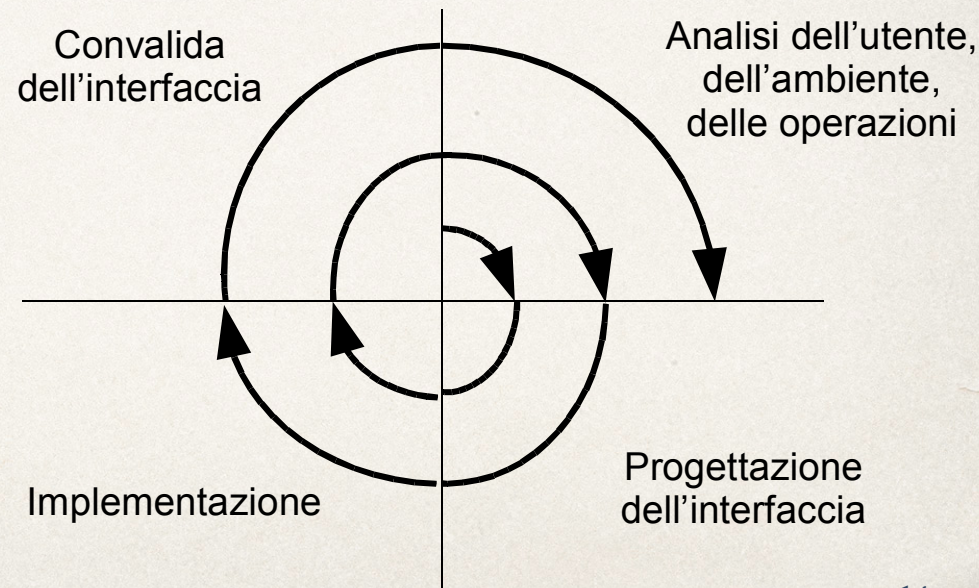
- ❖ Il processo di analisi e progettazione è iterativo e preferibilmente segue un modello a spirale (come modello sviluppo a spirale) in cui si distinguono 4 fasi

1. analisi e modellazione degli utenti, delle operazioni e dell'ambiente

2. progettazione dell'interfaccia

3. costruzione dell'interfaccia (implementazione)

4. validazione dell'interfaccia



Analisi dell'Utente e dell'Ambiente

- * E' sempre meglio comprendere il problema prima di tentare di progettare una soluzione. Per la UI questo significa:
 - * Analizzare gli utenti
 - * livello di abilità
 - * conoscenza generale del settore
 - * disponibilità ad accettare il sistema
 - * Per ogni tipologia di utenza vanno individuati i requisiti dell'interfaccia cercando di capirne la percezione
 - * Dell'ambiente in cui verrà usata l'interfaccia va considerato
 - * dove deve essere situata fisicamente l'interfaccia
 - * la situazione in cui l'utente userà l'interfaccia (in piedi / seduto, fermo / in movimento, ...)
 - * vincoli ambientali (luce, rumore,..) / richieste ergonomiche

Analisi e modellazione delle operazioni

- * L'obiettivo è rispondere a queste domande
 - * quale lavoro verrà svolto dall'utente e in quali circostanze? quali operazioni principali e secondarie verranno svolte dall'utente mentre svolge il proprio lavoro?
 - * quali specifici oggetti del dominio del problema verranno manipolati dall'utente durante lo svolgimento del lavoro?
 - * quale è la sequenza di operazioni di lavoro (workflow)?
 - * quale è la gerarchia delle operazioni?
- * E' molto simile a quanto già fatto in fase di analisi del sistema, solo applicato alla UI

Analisi e modellazione delle operazioni

Use case

- ❖ nell'analisi delle operazioni lo usiamo per mostrare come l'utente svolge alcune operazioni specifiche
- ❖ di solito scritto in modo informale
- ❖ vanno estratte le operazioni, gli oggetti e il flusso generale delle interazioni

Analisi e modellazione delle operazioni

Elaborazione delle operazioni

- * possiamo applicare la decomposizione funzionale in due modi
 - a) dato che il sistema di solito sostituisce una attività manuale, dobbiamo comprendere le operazioni normalmente svolte e tradurle in operazioni simili (non necessariamente identiche) da implementare nella UI
 - b) in alternativa, si studiano le specifiche di una soluzione computerizzata e si determinano un insieme di operazioni che troveranno posto nel modello utente, nel modello di progettazione, nella percezione del sistema
- * L'elaborazione delle operazioni è utile, ma attenzione: solo perché si è elaborata l'operazione in un modo, non vuol dire che non vi sia un altro modo per farla

Analisi e modellazione delle operazioni

Elaborazione degli oggetti

- ❖ Invece di pensare alle operazioni, partiamo dallo use case e estraiamo gli oggetti con cui abbiamo a che fare
- ❖ Cataloghiamo in classi questi oggetti
- ❖ Definiamo gli attributi di ogni classe e, dalle azioni che facciamo sugli oggetti, definiamo un elenco di operazioni
- ❖ Non vogliamo una implementazione letterale di queste operazioni, ma i dettagli dell'operazione

Analisi e modellazione delle operazioni

Analisi del workflow

- ❖ Se la UI è usata da più utenti che hanno ruoli differenti può essere necessario applicare l'analisi del workflow
- ❖ Il metodo migliore per farlo è costruire degli activity diagram in cui siano ben chiari i ruoli coinvolti (partendo dagli use cases)
- ❖

Analisi e modellazione delle operazioni

Rappresentazione gerarchica

- ❖ Una volta definito il workflow, per ogni ruolo si può definire una gerarchia di operazioni
- ❖ Di solito la gerarchia deriva da una elaborazione progressiva di ciascuna operazione identificata per l'utente
- ❖ Es: l'operazione "ritiro contante" del bancomat è formata da "identificazione" e "specificare la somma", ma "identificazione" è composta da "riconoscimento carta" e da "verifica PIN"

Design della UI

- ❖ Esistono vari modelli di design, ma tutti hanno una combinazione di questi passi
 - ❖ usa le informazioni dell'analisi per definire gli oggetti della UI e le relative azioni (operazioni)
 - ❖ definisci gli eventi (azioni degli utenti) che cambiano lo stato della UI; crea un modello di questo comportamento
 - ❖ rappresenta ogni stato della UI così come si presenterà all'utente finale
 - ❖ indica il modo in cui l'utente interpreta (o dovrebbe interpretare) lo stato del sistema sulla base delle informazioni fornite dalla UI

Definizione degli oggetti e delle azioni

- * Dalla descrizione delle operazioni si può ricavare lista oggetti e azioni mediante analisi grammaticale
- * Gli oggetti si distinguono in “di destinazione”, “di origine” e applicativi
 - * es.: se trascino l'icona di un file sull'icona della stampante per stamparlo, il primo è un oggetto di origine, mentre il secondo è un oggetto di destinazione
 - * un oggetto applicativo non può venire manipolato direttamente ma solo attraverso azioni indirette
- * Una volta individuati gli oggetti e le azioni più importanti, va specificata la posizione sull'interfaccia
 - * se per l'interfaccia si usa una metafora del mondo reale, questo è il momento di specificarla

Problemi di design

Tempo di risposta in termini di durata e variabilità

- ❖ la durata non deve essere né troppo lunga né troppo corta
- ❖ conviene una bassa variabilità, anche se il tempo medio di attesa è lungo

Sistema di help per l'utente

- ❖ integrato: contestuale allo stato del sistema
- ❖ esterno: manuale consultabile anche online
- ❖ I problemi per implementare un help system sono
 - ❖ disponibile per tutte le funzioni e in ogni fase dell'interazione?
 - ❖ in che modo l'utente può richiedere aiuto?
 - ❖ che aspetto usare (finestra, balloon, ..)?
 - ❖ come uscire dall'aiuto
 - ❖ come strutturare le informazioni di aiuto?

Problemi di design

Messaggi di errore

- ❖ comunicano all'utente situazioni patologiche
- ❖ i messaggi dovrebbero descrivere il problema con un linguaggio comprensibile all'utente
- ❖ i messaggi dovrebbero fornire informazioni utili a risolvere il problema
- ❖ i messaggi dovrebbero indicare eventuali effetti negativi dell'errore
- ❖ i messaggi dovrebbero essere accompagnati da effetti audio/video (beep, blink, colore diverso, ...)
- ❖ i messaggi non dovrebbero colpevolizzare l'utente

Problemi di design

Menu e comandi

- ❖ Oggi l'uso di interfacce a finestre ha ridotto l'uso della riga di comando, ma soprattutto tra utenti esperti si preferisce quest'ultima
 - ❖ si dovrà prevedere un comando per ogni opzione del menu?
 - ❖ che forma dovranno avere i comandi? (posso usare Fxx, ALT-P, ...)
 - ❖ quanto è difficile l'apprendimento dei comandi?
 - ❖ l'utente può personalizzare o abbreviare i comandi?
 - ❖ i menu sono autoesplicativi nel contesto della user interface=
 - ❖ i menu sono coerenti con la funzione indicata nel titolo?

Problemi di design

Accessibilità

- ❖ settore che sta avendo sempre più importanza (vedi W3C WAI)

Internazionalizzazione

- ❖ troppo spesso le interfacce sono pensate per una lingua e adattate per gli altri casi
- ❖ funzionalità di localizzazione per personalizzare la UI per un determinato mercato
- ❖ esistono varie indicazioni di internazionalizzazione che affrontano problemi generali di progettazione e problemi non banali di implementazione

Presentazione delle informazioni

Informazione statica / dinamica

- ❖ Fattori da considerare
 - ❖ l'utente è interessato a una informazione precisa o a una relazione tra dati? l'informazione è testuale o numerica? i valori relativi sono importanti?
 - ❖ quanto velocemente cambiano i dati? il cambiamento deve essere comunicato immediatamente?
 - ❖ che tipo di azione deve corrispondere al cambiamento dei dati?

Presentazione delle informazioni

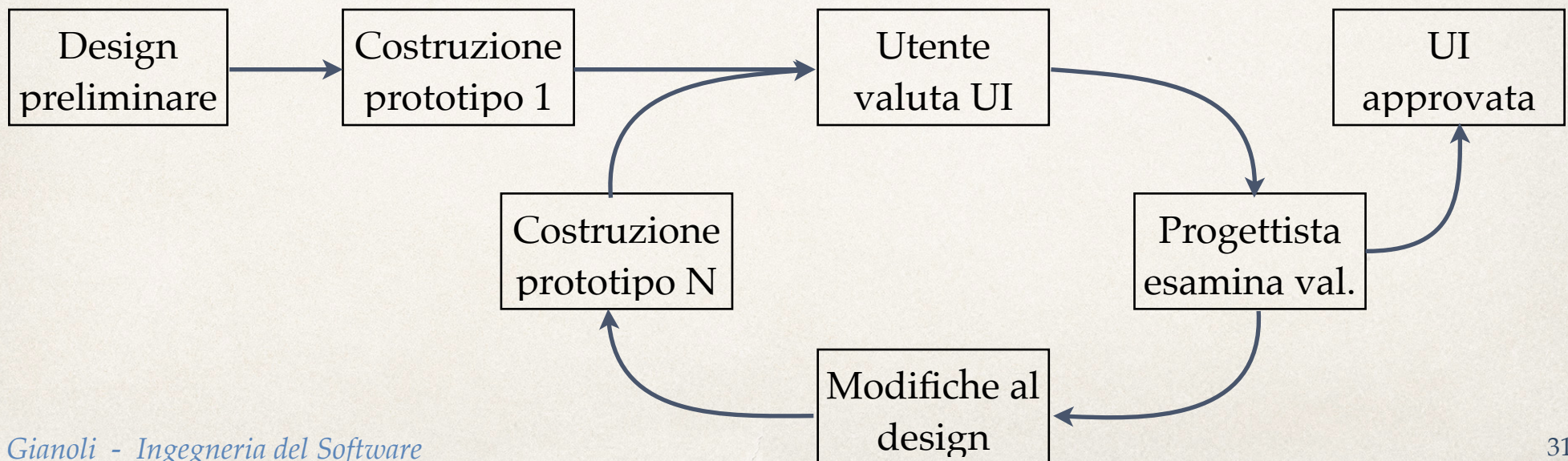
- ❖ Presentazione digitale dei dati
 - ❖ rappresentazione compatta; può comunicare valori precisi
- ❖ Presentazione analogica dei dati
 - ❖ con una occhiata fornisce una approssimazione dei valori
 - ❖ si possono mostrare i valori relativi ed evidenziare i valori fuori norma

Strumenti di implementazione

- ❖ Esistono strumenti che consentono la creazione di prototipi di interfaccia in modo da consentire in modo semplice l'approccio iterativo allo sviluppo
- ❖ I sistemi di sviluppo di interfacce utente utilizzano componenti e oggetti di base per
 - ❖ gestire dispositivi di input (mouse, tastiere)
 - ❖ convalidare l'input degli utenti
 - ❖ gestire gli errori e i messaggi associati
 - ❖ fornire messaggi di help
 - ❖ gestire finestre o campi di scorrimento interni
 - ❖ stabilire le connessioni tra il software applicativo e l'interfaccia
 - ❖ consentire la personalizzazione dell'interfaccia

Valutazione del design

- ❖ Quando si è creato un prototipo operativo della UI, occorre valutarlo per determinare se risponde alle esigenze. Vari metodi
 - ❖ informale: un utente usa il sistema e offre il proprio parere
 - ❖ formale: gruppi di utenti usano il sistema e vengono usati metodi statistici per valutare i questionari di valutazione



Valutazione del design

- ❖ La fase di valutazione può essere abbreviata se si effettuano delle valutazioni già prima di fare il prototipo di UI
 - ❖ la lunghezza delle specifiche necessarie a descrivere i requisiti della UI è un indice della difficoltà di apprendimento degli utenti
 - ❖ il numero di operazioni degli utenti e il numero medio di azioni per ogni operazione è un indice del tempo di interazione e dell'efficienza globale
 - ❖ il numero di azioni, operazioni, stati del sistema è proporzionale al ricorso alla memoria da parte dell'utente
 - ❖ lo stile, le funzionalità di help, il modo di gestione degli errori sono una indicazione della complessità della UI e del livello di accettazione da parte degli utenti