



UNIVERSITÀ
DELLA CALABRIA

DIPARTIMENTO DI **MATEMATICA
E INFORMATICA**

Interfacce Grafiche e Programmazione ad Eventi

Carmine Dodaro

Anno Accademico 2018/2019

Regole da seguire

Quando si progetta un'interfaccia grafica è necessario considerare:

- il target dei suoi utenti: skill, esperienza, attese, ecc.
- gli utenti spesso giudicano un sistema dal sua interfaccia e non dalle funzionalità che offre
- un'interfaccia grafica progettata male può essere causa di abbandono del sistema

Ricordate sempre: Un'interfaccia grafica è come una battuta, se la devi spiegare non è così buona!

La componente umana

- Le persone hanno una memoria a breve termine limitata
- Le persone tipicamente possono ricordare 7/8 elementi informativi. Avere un'interfaccia grafica che presenta più di questi elementi crea confusione ed errori
- Le persone **sbagliano**! Quando succede i messaggi di errore devono essere appropriati
- Le persone sono diverse tra di loro: hanno capacità fisiche e mentali diverse, preferenze di interazione diverse, ecc. Una buona interfaccia grafica non dovrebbe essere basata sulle capacità di chi la realizza

I principi da rispettare

- **Familiarità:** L'interfaccia dovrebbe usare termini e concetti dettati dall'esperienza di chi utilizzerà il sistema
- **Consistenza:** Quando possibile, operazioni comparabili dovrebbero essere attivate allo stesso modo
- **Minima sorpresa:** Gli utenti non dovrebbero essere sorpresi dal comportamento del sistema
- **Riparabilità:** L'interfaccia deve sempre prevedere la possibilità di riparare agli errori
- **Guida:** L'interfaccia deve prevedere sempre messaggi significativi e semplici da interpretare
- **Diversità:** L'interfaccia deve prevedere dei meccanismi appropriati per i diversi tipi di utenti

L'uso dei colori

I colori aggiungono una dimensione extra all'interfaccia e possono aiutare un utente a capire alcune strutture informative complesse.

I colori possono essere utilizzati per evidenziare eventi eccezionali.

Gli errori comuni nell'uso dei colori includono:

- L'uso dei colori per comunicare dei significati
- L'uso eccessivo dei colori nello schermo

Regole

- Limitare il numero di colori usati
- Usare i colori per mostrare un cambio nello stato del sistema
- Attenzione all'accoppiamento di colori diversi

Progettazione

La progettazione dei messaggi di errore è importante per diverse ragioni:

- come abbiamo detto, gli utenti compiono sicuramente azioni sbagliate.
- i messaggi di errore possono cambiare la percezione sul sistema
- i messaggi devono essere educati, concisi, consistenti e costruttivi. Non bisogna tentare di essere divertenti
- i messaggi di errore devono essere orientati all'utente e non al sistema

Processo interattivo

La progettazione di un'interfaccia grafica è un processo che richiede interazioni continue tra utenti e progettisti. Le 3 attività principali sono:

- **Analisi dell'utente:** Capire cosa faranno gli utenti con il sistema
- **Preparazione di prototipi:** Sviluppare diversi prototipi da validare
- **Valutazione dell'interfaccia:** Mostrare i prototipi agli utenti e analizzare le reazioni

Criteria

- **Curva di apprendimento:** quanto è facile per un nuovo utente imparare ad usare il sistema?
- **Velocità delle operazioni:** quanto è aderente alle esigenze comuni degli utenti?
- **Robustezza:** quanto è tollerante agli errori?
- **Recupero:** quanto è facile rimediare ad un errore?
- **Adattabilità:** quanto è adattabile il sistema a diversi tipologie di lavoro?

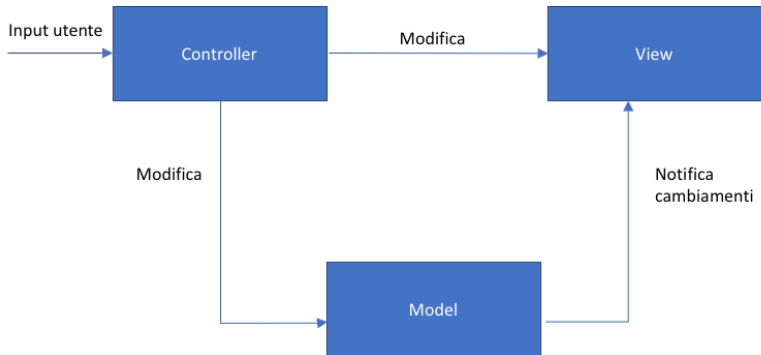
Pattern MVC

Architettura software che permette di separare le componenti principali di un'interfaccia grafica: la rappresentazione dell'informazione e l'interazione con l'utente.

MVC è l'acronimo di Model-View-Controller e l'idea è di creare tre componenti diversi:

- 1 Model: rappresenta i dati di dominio, la logica applicativa e il meccanismo di persistenza
- tante View: l'interfaccia grafica vera e propria
- tanti Controller: fanno da intermediari tra il Model e View. Gestisce le azioni degli utenti e li mappa su azioni sul modello. Quindi, in generale è responsabile della gestione degli eventi

Pattern MVC



Esecuzione

- 1 L'utente compie un'azione sulla View
- 2 Il Controller riceve l'evento di input
- 3 Il Controller notifica l'azione al Model e ne modifica lo stato
- 4 Il Model notifica alla View il proprio cambio di stato
- 5 La View legge il Model e si aggiorna