

Corso di Interazione Uomo-Macchina

Paolo Bottoni

DIPARTIMENTO
DI INFORMATICA



SAPIENZA
UNIVERSITÀ DI ROMA

Lezione 7: Pattern di Interazione

Pattern

- Emergono da sistemi di successo
- Pattern di esperienza utente
- Pattern di progetto di interazione

Pattern di esperienza utente

- Proprietà desiderabili in esperienza utente
- Tipi di pattern

Esplorazione

Soddisficiente

Scelte differite

Abituazione

Memoria prospettica

Tastiera sufficiente

Gratificazione istantanea

Cambiamenti in corso d'opera

Costruzione incrementale

Memoria spaziale

Ripetizione efficiente

Consigli di altre persone

Esplorazione

- Comprendere comportamento e possibilità senza perdersi o causare errori
- Esempi
 - Uso di diversi filtri
 - Esplorazione sito
 - Nuove funzionalità dispositivo
- Richiede
 - Undo
 - Back
 - Chiara indicazione conseguenze

Gratificazione istantanea

- Produzione di risultati da inizio attività
- Esempi
 - Creazione di documenti
 - Accesso immediato a funzionalità
- Richiede
 - Previsione azioni tipiche utente
 - Funzionalità introduttive non nascoste

Soddisficiente

- Accettazione di "sufficiente" anziché "ottimo" se risultato soddisfacente e alternative costose. Evita carico cognitivo.
- Esempi
 - Uso di metodi di accesso più lunghi, ma appresi
 - Presentazione di opzioni
- Richiede
 - Etichette facilmente identificabili
 - Disposizione elementi suggestiva scelte
 - Navigazione chiara, punti di recupero
 - Bassa complessità visiva

Cambiamenti in corso d'opera

- Utente cambia obiettivo durante interazione
- Esempi
 - Visito Amazon per consultare, finisco per comprare
 - Interruzione attività
 - Per cause esterne
 - Attività in più passi
- Richiede
 - Ambiente di navigazione ricco
 - Mantenimento dello stato raggiunto

Scelte differite

- Permettere completamenti parziali.
- Esempi
 - Informazioni di registrazione
 - Creazione di progetti
 - Creazione artistica
- Richiede
 - Non costringere utente a scelte premature
 - Indicare parti obbligatorie
 - Permettere ritorno a completamento
 - Nascondere scelte non fondamentali
 - Presentare default accettabili
 - Permettere accesso a demo

Costruzione incrementale

- Necessità di procedere con cambiamenti parziali, revisionarli, lavorare su parti diverse
- Esempi
 - Costruzione di siti
 - Scrittura di documenti
 - Processo creativo
- Richiede
 - Reazione rapida interfaccia
 - Presentazione costante modifiche stato

Abituazione

- Abitudini da uso ripetuto. Favorisce efficienza, nasconde trappole
- Esempi
 - Azioni tipiche di salvataggio, chiusura, back
 - Trasferimento di comandi fra ambienti
- Richiede
 - Coerenza fra applicazioni e entro applicazioni
 - Identificare situazioni in cui rompere abitudine

Memoria spaziale

- Utenti ritrovano elementi in base a posizione
- Esempi
 - Raggruppamenti documenti su desktop
 - Posizioni relative elementi
- Richiede
 - Costanza posizionamento
 - Scelte utente sopravanzano politiche strumento
 - Attenzione a posizioni particolari

Memoria prospettica

- Ricordarsi di ricordarsi. Pianificare usi successivi
- Esempi
 - Mantenere aperte mail a cui rispondere.
 - Impostare allarmi su agenda elettronica.
- Richiede
 - Permettere annotazioni.
 - Permettere realizzazioni flessibili di compiti.
 - Evitare ottimizzazioni non richieste
 - Tenere memoria ultime operazioni, dati modificati

Ripetizione efficiente

- Semplificare operazioni ripetitive.
- Esempi
 - Sostituzioni in testo.
 - Macro su immagini.
 - Script
- Richiede
 - Disponibilità di comando per azioni tipiche.
 - Possibilità di configurazione da parte utente.

Tastiera sufficiente

- Non costringere utente a usare mouse.
- Esempi
 - Tecnologie assistive.
 - Non interruzione flusso.
 - Compilazione campi con passaggio a successivo
- Richiede
 - Scorciatoie da tastiera
 - Utilizzo tasti freccia, tabulazioni, tasti controllo.
 - Azioni di completamento

Consigli di altre persone

- Accesso a conoscenza condivisa
- Esempi
 - Raccomandazioni Amazon
 - Siti valutazione prodotti
 - Riutilizzo codice
- Richiede
 - Integrare componenti sociali
 - Incoraggiare condivisione senza lavoro aggiuntivo
 - Organizzare eventi pubblici legati a prodotto

Pattern di progettazione: UNDO

- Problema: utenti possono fare azioni che dopo vogliono annullare
- Contesto: in molte applicazioni ripristinare stato dopo azione erronea può essere difficile, es. richiede numerose azioni
- Principio: gestione errori

UNDO: soluzione

- Utenti possono annullare loro ultime azioni
- Mantenere coda di comandi eseguiti
 - almeno ultima coppia di azioni annullabile
- Mostrare storia dei comandi
 - Utenti sanno cosa hanno fatto
- Comandi devono essere annullabili.
 - Se comando ha effetti collaterali non annullabili, avvisare utente e non accodarlo

UNDO: motivazioni

- Offrire possibilità di annullare azioni dà maggiore confidenza a utenti
 - Possono esplorare, fare errori e recuperare
 - Facilita apprendimento funzionalità
- Esempi: Adobe Photoshop, Office

Principi per supportare l'usabilità

Apprendibilità

Facilità con cui nuovi utenti possono iniziare interazione efficace e raggiungere prestazioni massimali.

Flessibilità

Molteplicità modi in cui utente e sistema scambiano informazioni.

Robustezza

Livello di supporto fornito a utente per valutare completamente con successo e comportamento diretto a un obiettivo.

Principi di apprendibilità I

Predicibilità

- Determinare effetto di azioni future basandosi su storia interazioni passate
- Visibilità operazioni
- Pattern: [ClearEntryPoints](#)

Sintetizzabilità

- Stabilire effetto azioni passate
- Onestà immediata vs. finale

Principi di apprendibilità II

Familiarità

- Quanto conoscenza a priori si applica a nuovo sistema
- Indovinabilità; carattere di offerta
- Pattern: [InputPrompt](#), [ResponsiveEnabling](#)

Generalizzabilità

- Estendere interazione specifica a nuove situazioni.
- Pattern: [GlobalNavigation](#)

Coerenza

- Somiglianza comportamento ingressi/uscite in situazioni simili o con obiettivi di compito simili.
- Pattern: [VisualFramework](#)

Principi di flessibilità I

Iniziativa del dialogo

- Libertà da vincoli imposti da sistema su dialogo di ingresso.
- Prelazione da parte di sistema o utente.
- Pattern: [ExtrasOnDemand](#)

Multiprocesso

- Supporto a interazione utente per più compiti alla volta.
- Concorrenza vs. interfolliazione; multimodalità.
- Pattern: [CardStack](#), [TwoPanelSelector](#)

Migrabilità del compito

- Passaggio di responsabilità per esecuzione di compiti tra utente e sistema.
- Pattern: [JumpToItem](#), [ResponsiveDisclosure](#)

Principi di flessibilità II

Sostitutività

- Permettere uso di valori equivalenti di ingresso e uscita uno al posto dell'altro
- Molteplicità rappresentazione; opportunità uguali
- Pattern: [ForgivingFormats](#)

Personalizzabilità

- Modificabilità interfaccia utente lato utente (adattabilità) o sistema (adattività)
- Pattern: [PropertySheet](#)

Principi di robustezza I

Osservabilità

- Capacità utente di valutare stato interno del sistema da rappresentazione percepibile, navigabilità.
- Valori di difetto; raggiungibilità; persistenza; visibilità operazioni

Recuperabilità

- Capacità utente di fare azioni correttive se errore riconosciuto.
- Raggiungibilità; recupero in avanti o indietro; sforzo commisurato.
- Pattern: [MultilevelUndo](#)

Principi di robustezza II

Rispondenza

- Percezione velocità comunicazione con sistema
- Stabilità
- Pattern: [AnimatedTransition](#)

Conformità a compito

- Grado con cui servizi supportano compiti utente
- Completezza compiti; adeguatezza a compito.