

# Corso di Interazione Uomo-Macchina

**Paolo Bottoni**

DIPARTIMENTO  
DI INFORMATICA



**SAPIENZA**  
UNIVERSITÀ DI ROMA

## Lezione 15: Supporto all'implementazione

Lucidi tradotti e adattati dal materiale presente sul sito <http://www.hcibook.com/e3/resources/>

# Supporto all'implementazione

- Strumenti di programmazione
  - Livelli di servizio per programmatori.
- Stili di interazione
  - Natura dialogo utente/sistema
- Sistemi a finestre
  - Supporto base per attività utente-sistema separate e simultanee.
- Cassette attrezzi per interazione
  - Programmazione più vicina a livello percezione utente.
- Sistemi di gestione interfaccia utente
  - Controllano relazione tra presentazione e funzionalità.

# Introduzione

Impatto interazione su programmatore

Avanzamenti in codifica innalzano livello programmazione  
specifico hardware

→ specifico a tecnica di interazione

Strati di strumenti di sviluppo

- Sistemi a finestre
- Cassette attrezzi per interazione
- Sistemi di gestione interfaccia utente

# Elementi di sistemi a finestre

## Indipendenza da dispositivo

Programmazione di gestori di dispositivi terminali astratti

Modelli di immagine per uscite e (parzialmente) ingressi

- pixel
- PostScript (MacOS X, NextStep)
- Graphical Kernel System (GKS)
- Programmers' Hierarchical Interface to Graphics (PHIGS)

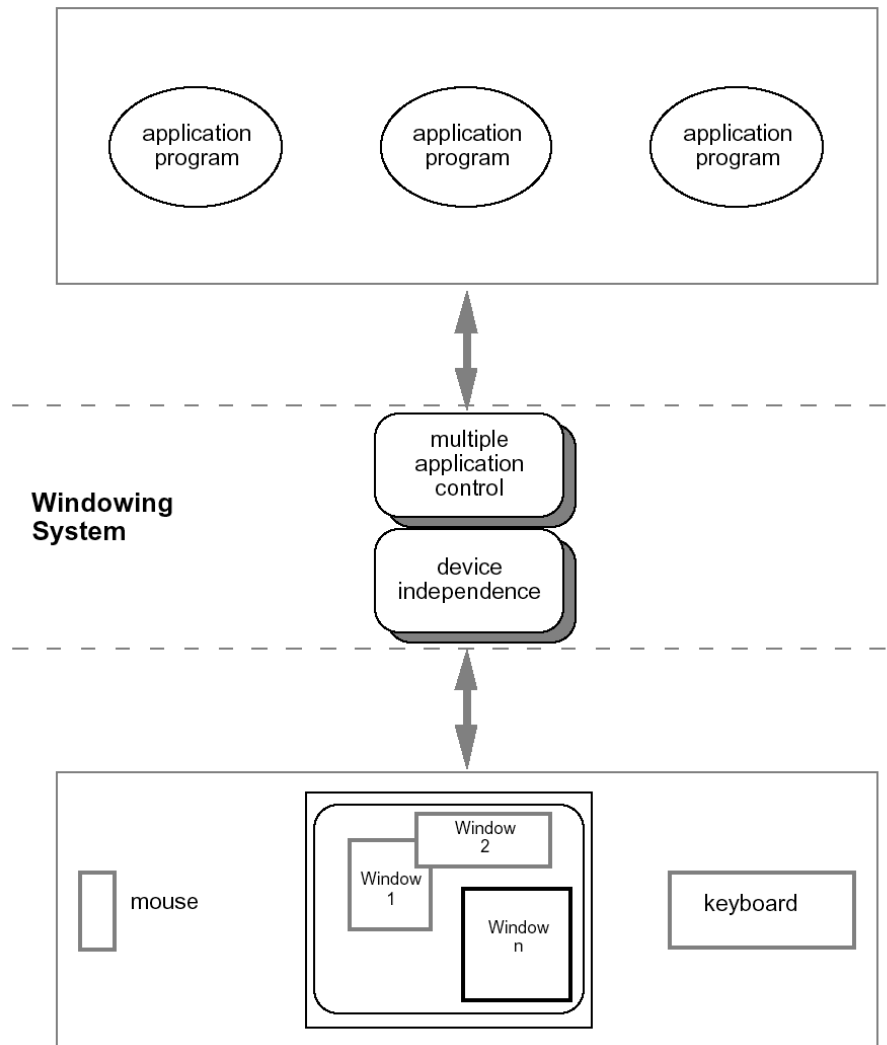
## Condivisione di risorse

Simultaneità di compiti utente

Sistema a finestre supporta processi indipendenti

Isolamento di applicazioni individuali

# Ruoli di un sistema a finestre

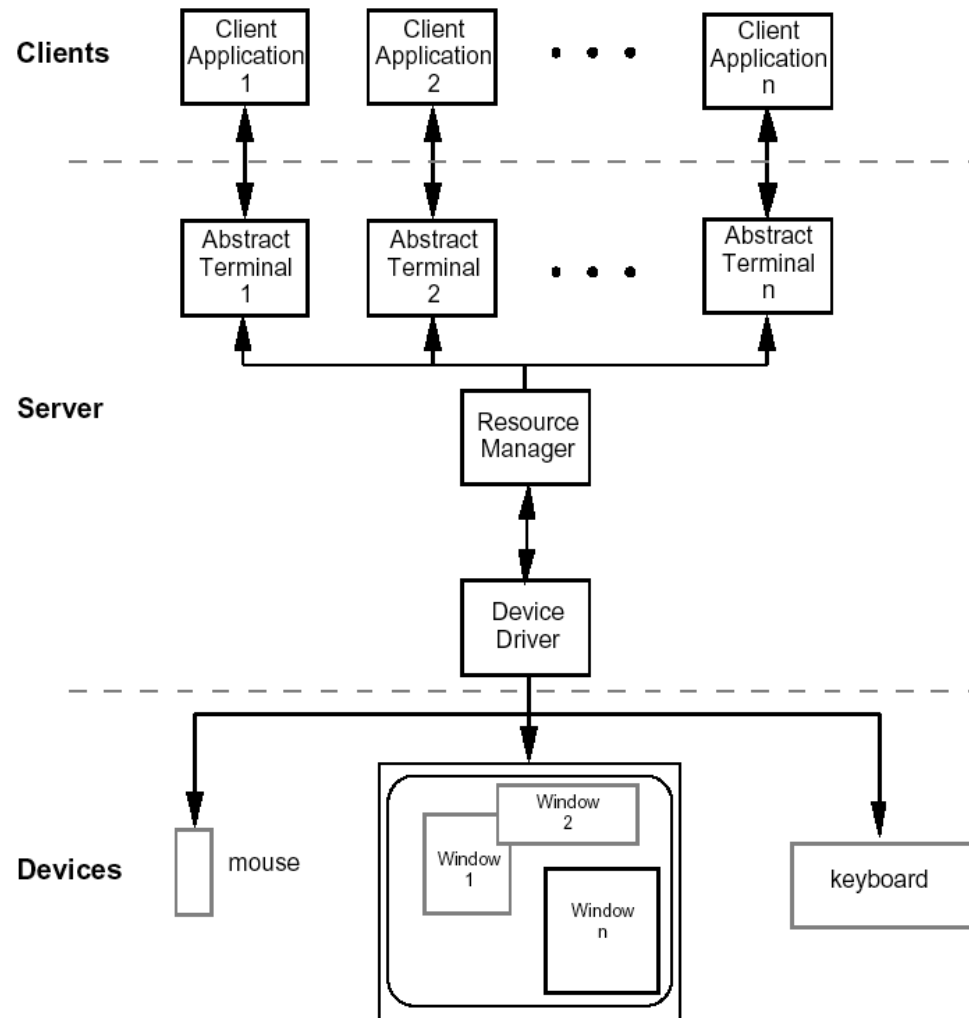


# Architetture per sistemi a finestre

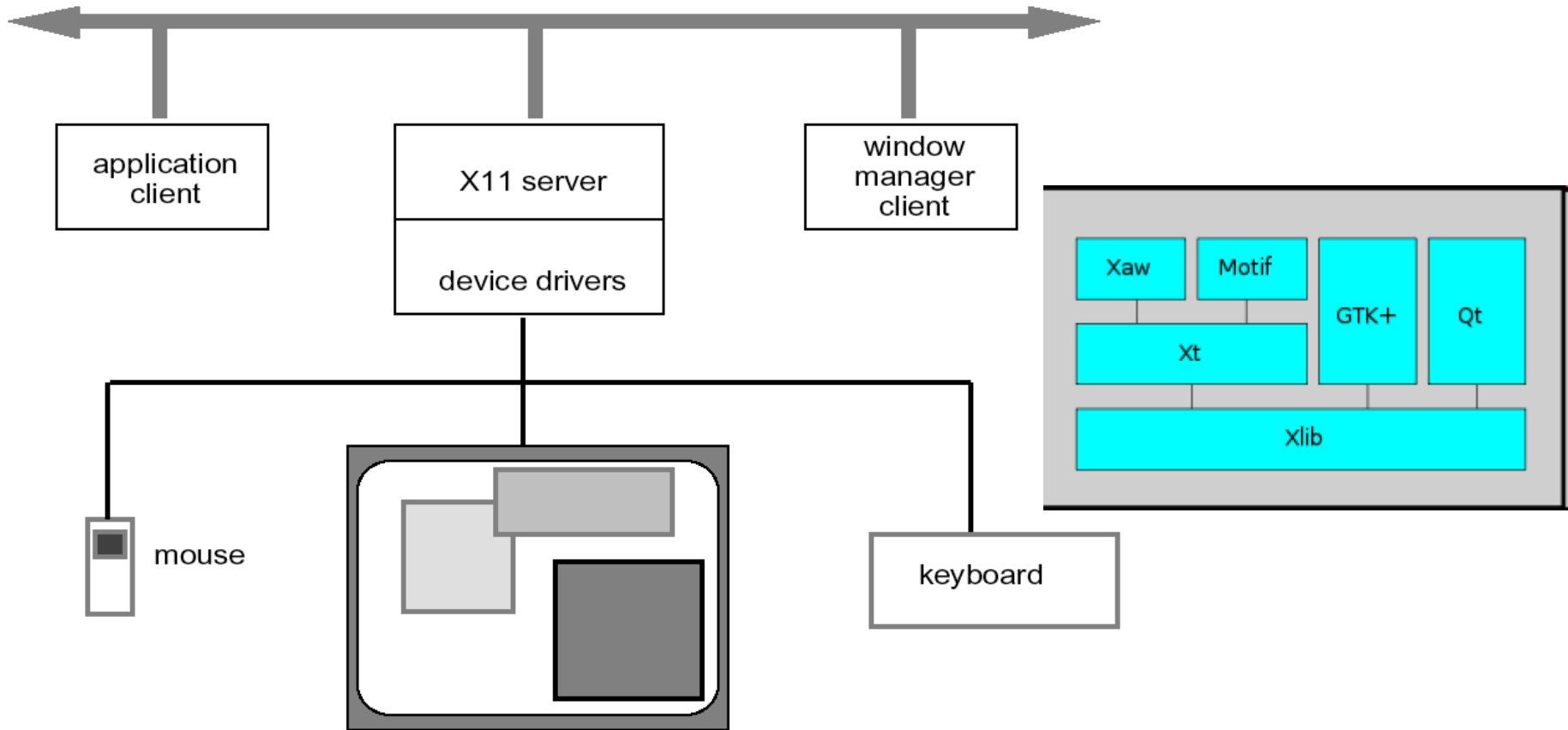
## Tre possibili architetture software

- Assumono gestore dispositivo separato
  - Differiscono per gestione applicazioni multiple
1. Ogni applicazione gestisce tutti i processi
    - Ognuna si occupa della sincronizzazione
    - Riduce portabilità applicazioni
  2. Ruolo di gestione in nucleo sistema operativo
    - Applicazioni legate a sistema operativo
  3. Ruolo di gestione come applicazione separata
    - Massima portabilità

# Architettura client-server



# Architettura X Window I



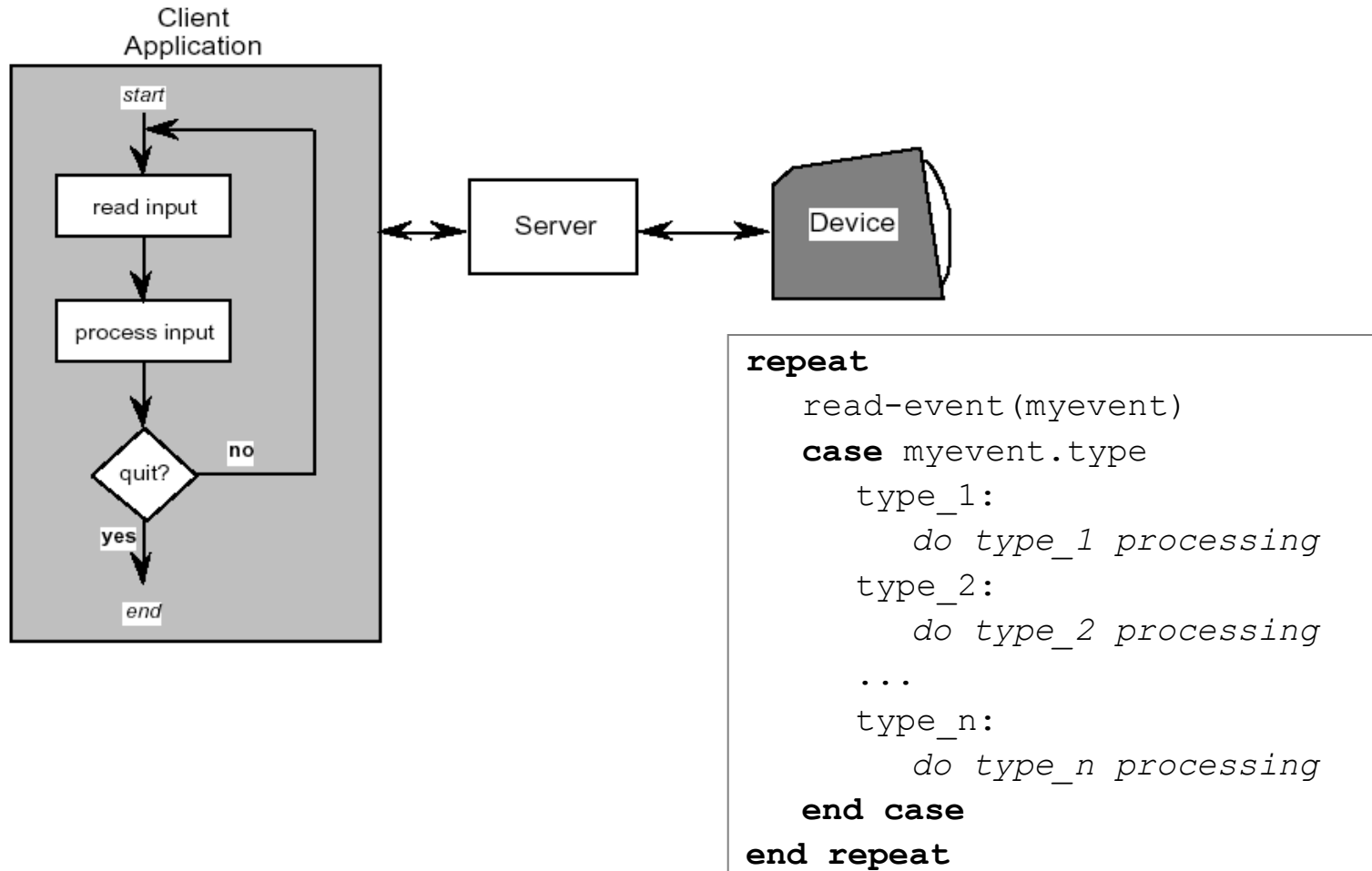


# Architettura X Window II

- Modello di gestione immagine a pixel con meccanismo di puntamento
- Protocollo X definisce comunicazione client-server
- Client separato di gestione finestre impone politiche per ingresso / uscita:
  - Come cambiare fuoco di ingresso
  - Finestre legate o sovrapposte
  - Trasferimento di dati inter-client

# Programmazione dell'applicazione - 1

## ciclo lettura-valutazione

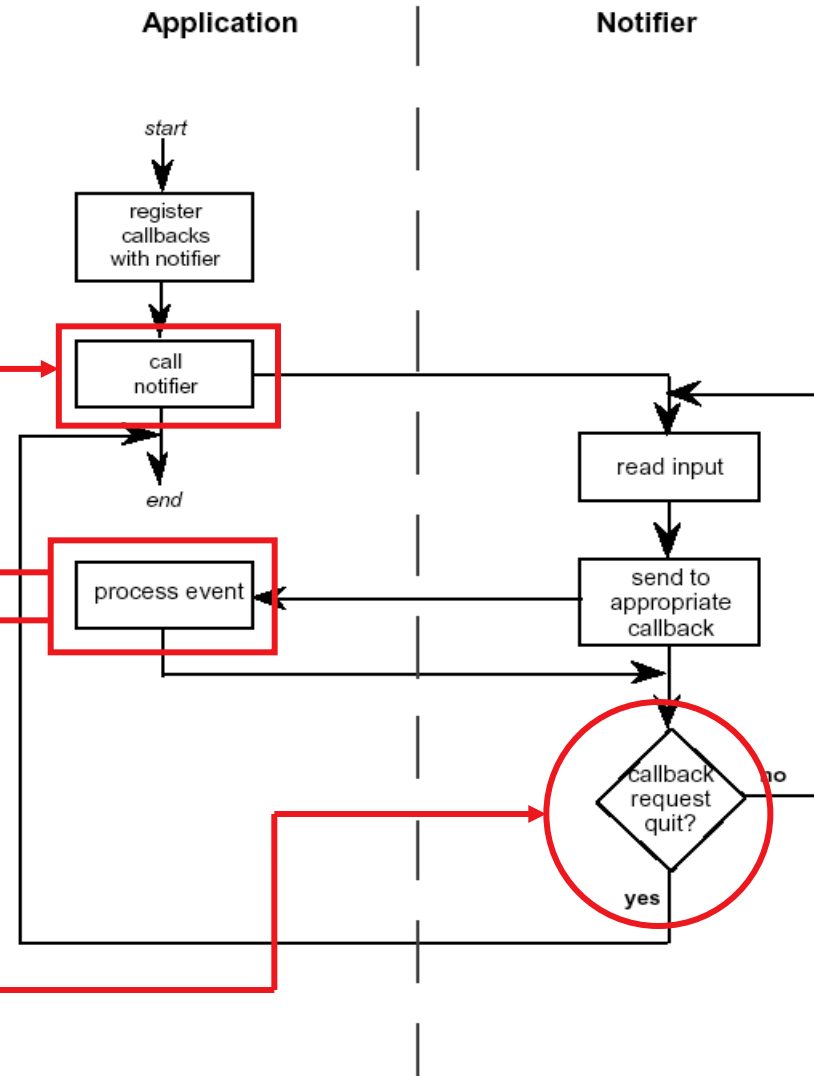


# Programmazione dell'applicazione - 1 basato su notifica

```
void main(String[] args) {  
    Menu menu = new Menu();  
    menu.setOption("Save");  
    menu.setOption("Quit");  
    menu.setAction("Save", mySave)  
    menu.setAction("Quit", myQuit)  
    ...  
}
```

```
int mySave(Event e) {  
    // save the current file  
}
```

```
int myQuit(Event e) {  
    // close down  
}
```



# Problemi di granularità

- Stile di sistema influenza interfaccia
  - Finestra di dialogo modale
    - Facile con loop di eventi (introduce altro ciclo di lettura)
    - Difficile con notifica (necessita di molte variabili di modo)
  - Finestra di dialogo non-modale
    - Difficile con loop di eventi (loop principale molto complicato)
    - Facile con notifica (basta aggiungere altro gestore)

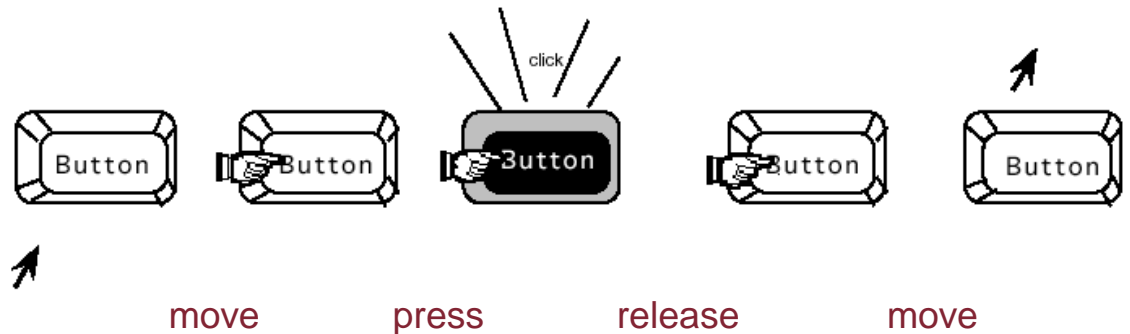
attenzione!

Se non progettato esplicitamente, accadrà da solo  
Implementazione non dovrebbe guidare progetto

# Uso di cassette degli attrezzi

## Oggetti di interazione

- Ingressi e uscite legati intrinsecamente



## Cassette attrezzi forniscono questo livello di astrazione

- Programmazione con oggetti di interazione (o tecniche, congegni, ecc.)
- Promuovono coerenza e generalizzabilità: apparenza e sensazioni simili
- Adatte a programmazione orientata a oggetti



# Interfacce utente in Java

- Cassetta attrezzi Java– AWT (Abstract Windowing Toolkit)
- Classi Java per pulsanti, menu, etc.
- Basato su notifica;
  - AWT 1.0 – necessità di specializzare classi di base
  - AWT 1.1 e oltre – oggetti callback
- Cassetta attrezzi Swing
  - Costruita sopra AWT – elementi di livello più alto
  - Usa architettura MVC

# Sistemi di Gestione Interfacce Utente (UIMS)

- UIMS aggiungono altro livello sopra cassette
  - Cassette troppo difficili per non-programmatori
- Responsabilità dell'UIMS
  - Architettura concettuale
  - Tecniche di implementazione
  - Infrastruttura di supporto
- Termini non-UIMS:
  - Sistema di sviluppo UI (UIDS)
  - Ambiente di sviluppo UI (UIDE)
    - es. Visual Basic

# UIMS come architettura concettuale

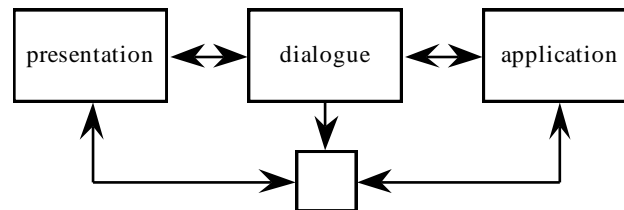
- *Separazione* tra semantica e presentazione applicazione
- Migliora:
  - portabilità – gira su sistemi diversi
  - riusabilità – componenti riutilizzate riducono costi
  - interfacce multiple – accesso a stessa funzionalità
  - personalizzabilità – da parte di progettista e di utente



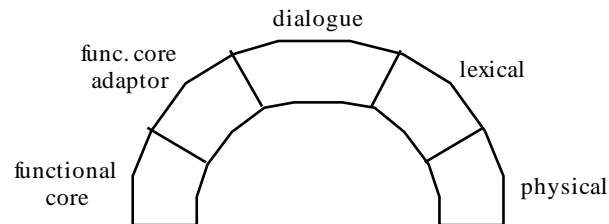
# Tradizione UIMS – strati di interfaccia / componenti logiche

- linguistici: lessicale/ sintattico /semantico

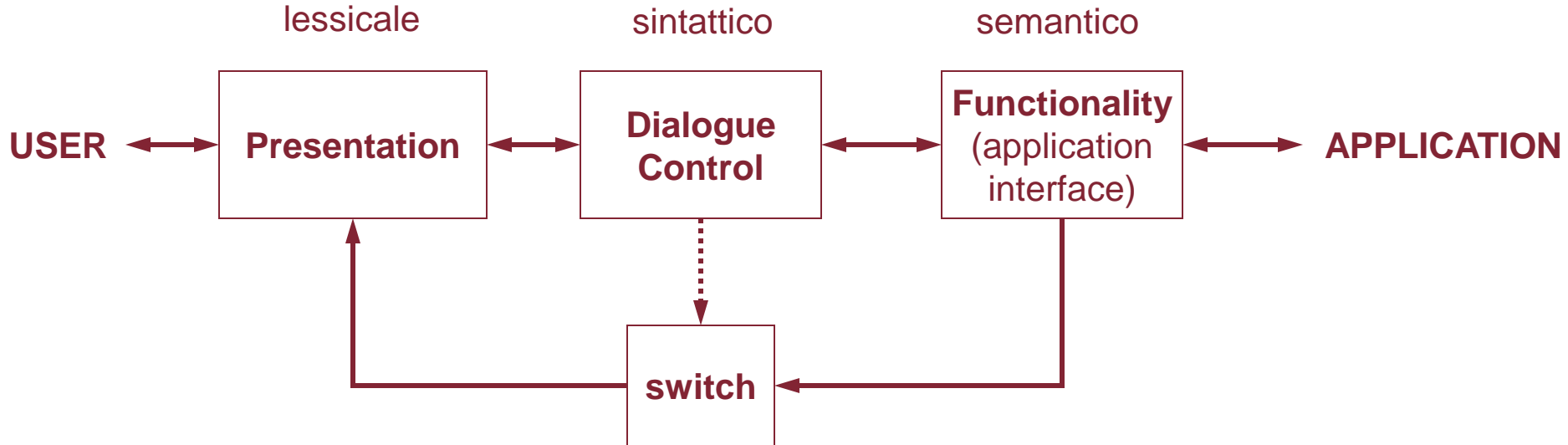
- Seeheim:



- Arch/Slinky



# Modello di Seeheim



# Concettuale vs. implementazione

## Seeheim

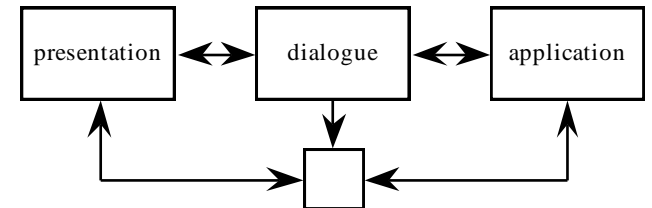
- Nasce da esperienza implementativa
- Contributo principale è concettuale
- Definisce concetti parte linguaggio "normale" UI

... a causa di Seeheim ...

... pensiamo diversamente!

es. Scatola inferiore, scambio

- Necessaria per implementazione
- Non concettuale

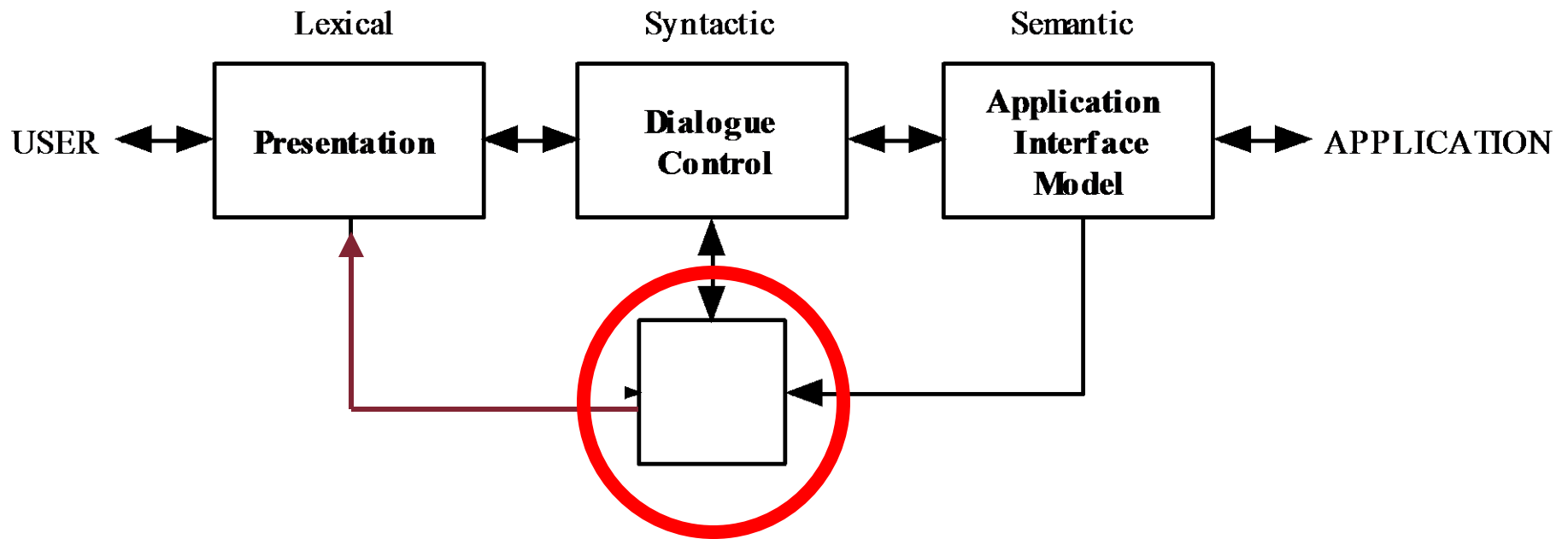




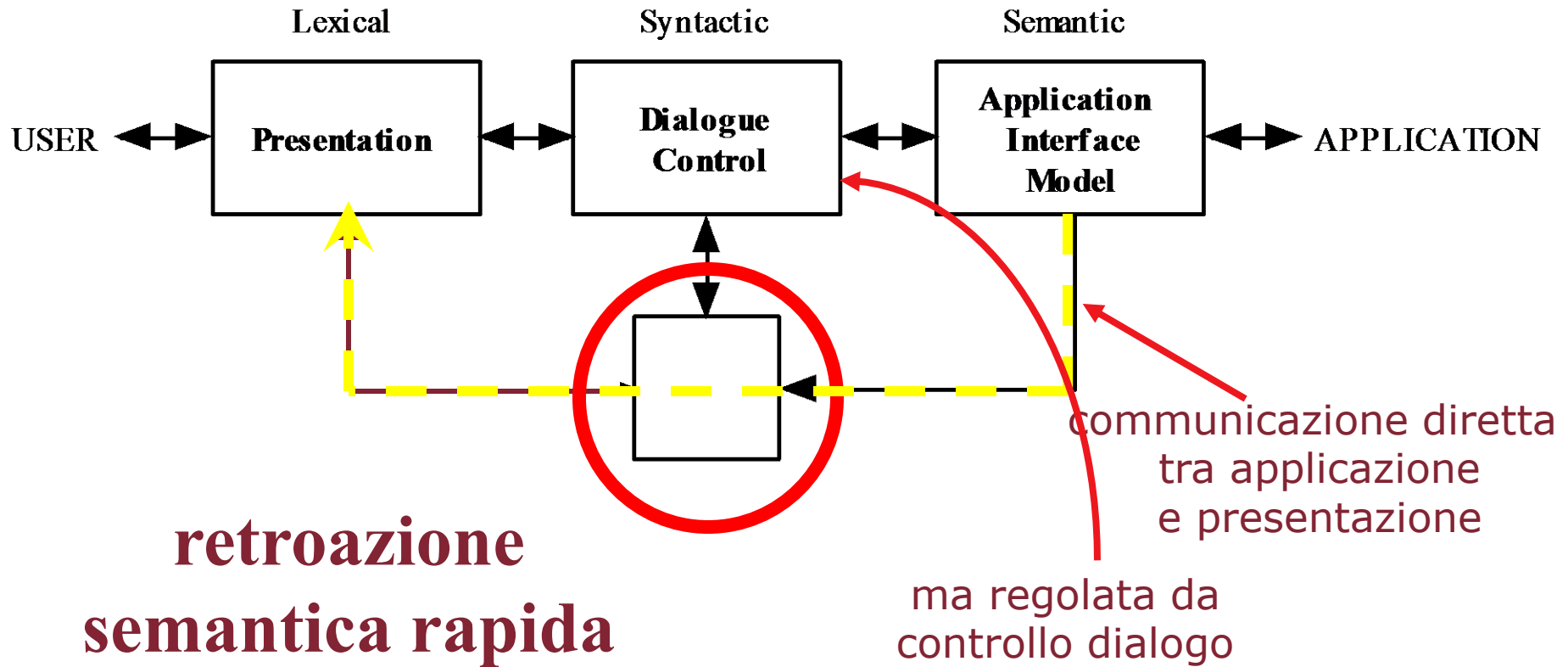
# Retroazione semantica

- Diversi tipi di retroazione:
  - lessicale – movimento del mouse
  - sintattica – evidenziazione di menu
  - semantica – variazione di valore
- Retroazione semantica spesso più lenta
  - Uso di retroazione lessicale / sintattica rapida
- Può esserci bisogno di retroazione semantica rapida
  - Disegno a mano libera
  - Evidenziare cestino o contenitore quando documento trasportato

# Cos'è questo?

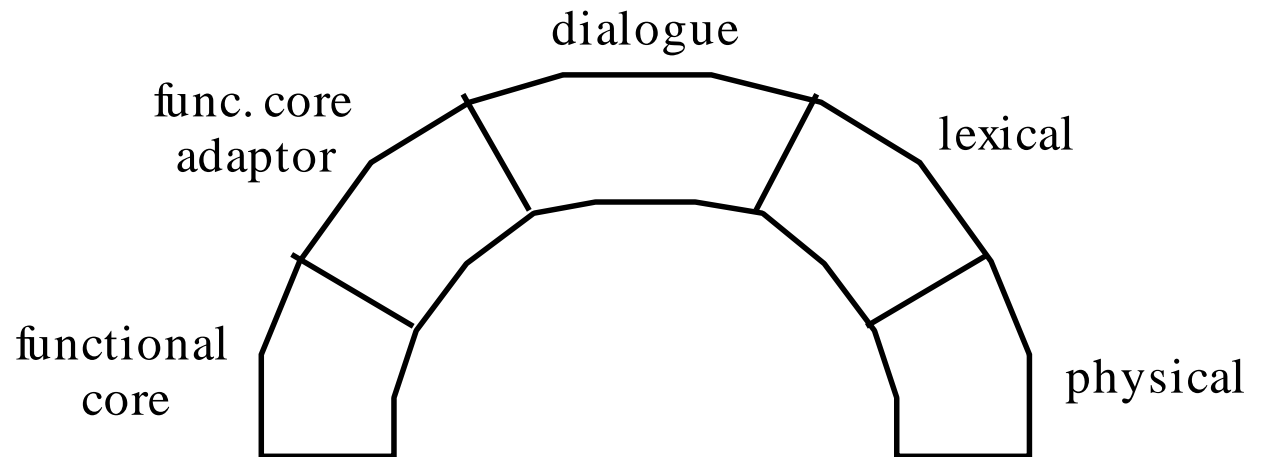


# Il bypass /switch



# Arch/Slinky

- Più strati – distingue lessicale / fisico
- Diversi strati possono essere più spessi in sistemi diversi
  - o in componenti diversi



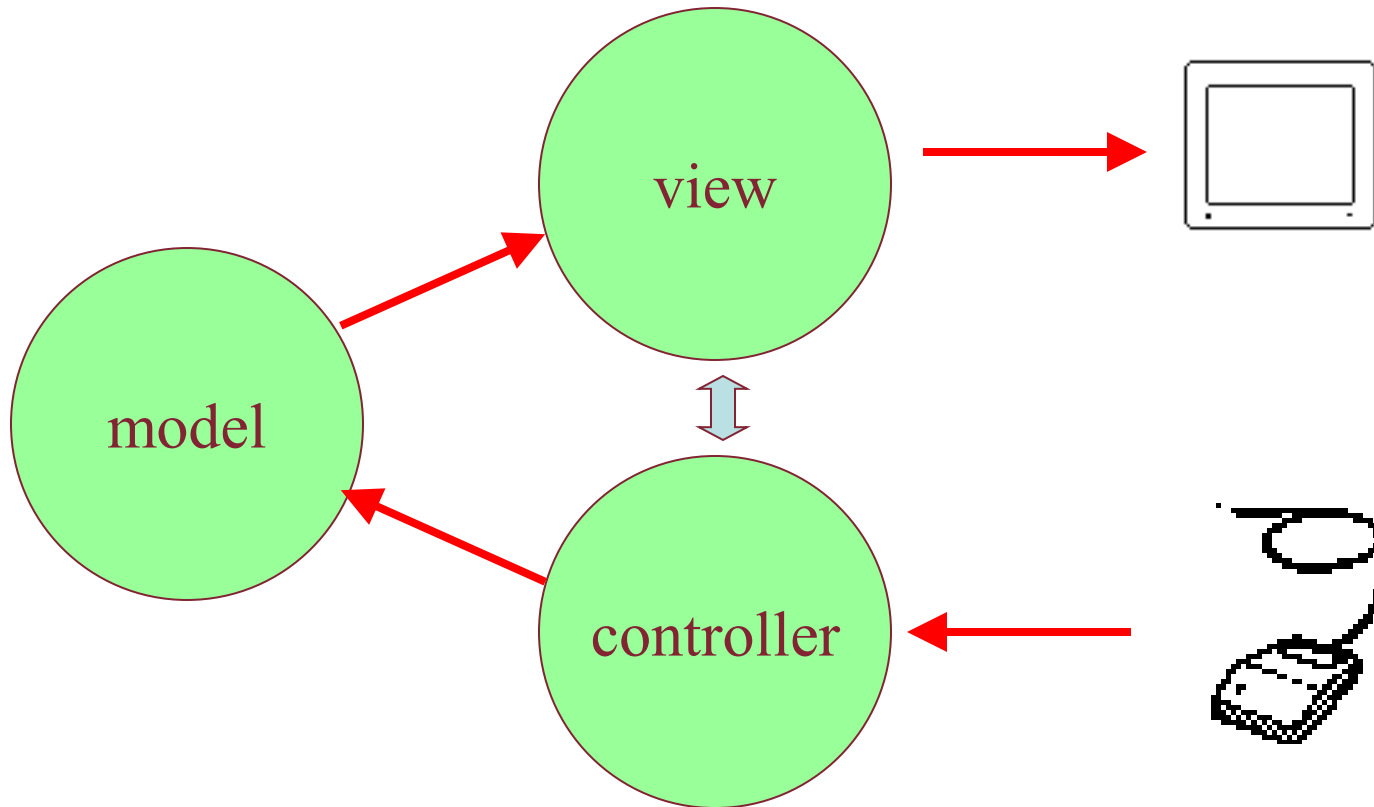
# Monolitico vs. componenti

- Modello di Seeheim ha grandi componenti
- Spesso più semplice usarne di piccoli
  - Specie se si usano cassette orientati a oggetti
- Smalltalk usa MVC – model–view–controller
  - model – stato logico interno componente
  - view – come viene mostrato su schermo
  - controller – elabora ingressi utente



# MVC

model - view - controller



# Questioni MVC

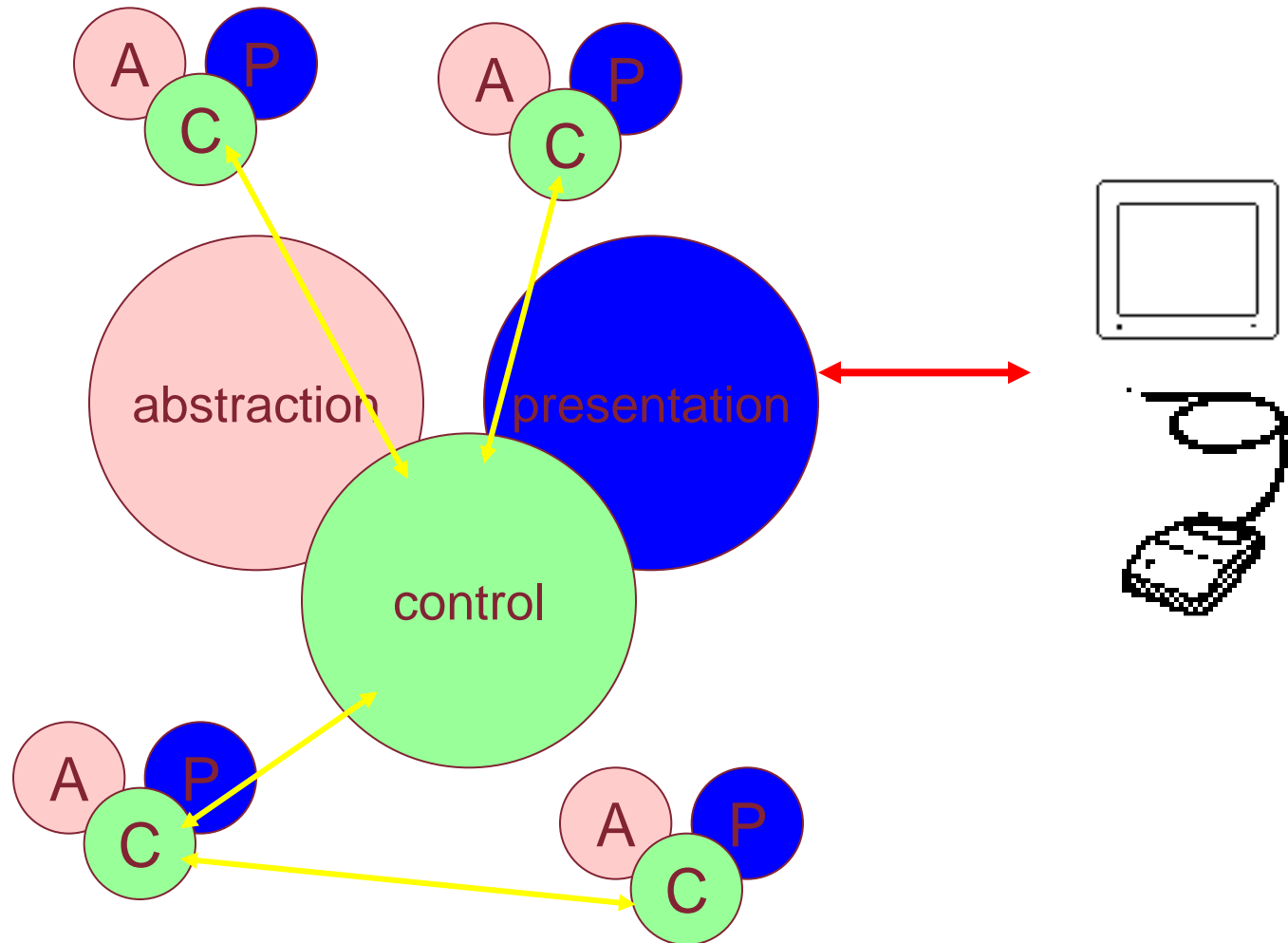
- Fondamentalmente modello a condotta:  
ingresso → control → model → view → uscita
- Ma in interfaccia grafica
  - ingresso ha significato solo in relazione a uscitaes. clic
  - occorre sapere su cosa è stato fatto clic
  - controller deve decidere cosa fare
  - ma è vista che sa cosa è mostrato in quel punto!
- In pratica controller ‘parla’ alla view
  - separazione non completa

# Modello PAC

- Modello PAC più vicino a Seeheim
  - abstraction – stato logico della componente
  - presentation – gestisce ingresso e uscita
  - control – media tra loro
- Gestisce gerarchia e viste multiple
  - Controlla parte di PAC di comunicazione fra oggetti
- Più pulita in molti modi ...  
ma MVC più usati in pratica  
(es. Java Swing)

# PAC

presentation - abstraction - control



# Implementazione di UIMS

- Tecniche per controllo dialogo
  - reti di menu
  - notazioni grammaticali
  - linguaggi dichiarativi
  - specifiche grafiche
  - diagrammi di transizione di stato
  - linguaggi di eventi
  - vincoli
- Vincoli
  - Invece di cosa *succede* dicono cosa dovrebbe essere *vero*
  - Usati in groupware ma anche interfacce per singolo utente (ALV - abstraction-link-view)

# Specifiche grafiche

- Cos'è
  - Si disegnano componenti su schermo
  - Si definiscono azioni con script o collegamenti a programmi
- In uso
  - Tecnica più popolare insieme a programmazione diretta
  - es. Visual Basic, Dreamweaver, Flash
- Locale vs. globale
  - Difficile "vedere" cammini in sistema.
  - Fuoco su cosa può essere visto su schermo.

# Deriva del controllo del dialogo

- **Controllo interno**  
(es. ciclo lettura-valutazione)
- **Controllo esterno**  
(indipendente da semantica o presentazione applicazione)
- **Controllo presentazione**  
(es. specifica grafica)

# Stili di interazione tradizionali

- Interfacce a linee di comandi
- Menu
- Linguaggio naturale
- Domanda / risposta e dialogo di interrogazioni
- Moduli e fogli elettronici
- WIMP
- Point and click



# Interfacce a linee di comandi

- Esprimere istruzioni a calcolatore direttamente
  - Tasti funzione, caratteri singoli, abbreviazioni, parole intere
- Adatte a compiti ripetitivi
- Migliori per utenti esperti piuttosto che nuovi utenti
- Offrono accesso diretto a funzionalità di sistema
- Nomi/abbreviazioni di comandi significativi!
  
- Esempio tipico: Unix

# Menu

- Insieme di opzioni mostrate su schermo
- Opzioni visibili
  - Minore richiesta di ricordare – più facile da usare
  - Si affidano a riconoscimento, nomi significativi
- Selezione attraverso:
  - Numeri, lettere, tasti freccia, mouse
- Raggruppamenti gerarchici
  - Logica raggruppamento
- Forma ristretta di sistema WIMP

Presentazione di voci da menu in altre forme.

# Linguaggio naturale

- Familiare per utente
- Riconoscimento parlato o linguaggio naturale tipizzato
- Problemi
  - Vago
  - Ambiguo
  - Difficile da realizzare bene!
- Soluzioni
  - Cercare di comprendere sottoinsieme
  - Identificare parole chiave

# Interfacce basate su interrogazioni

- Interfacce a domanda/risposta
  - Utente condotto a interagire attraverso serie di domande
  - Adatto per nuovi utenti, ma funzionalità ristretta
  - Spesso usate in sistemi informativi
- Linguaggi di interrogazione (es. SQL)
  - Usati per ritrovare informazione da una base di dati
  - Richiede comprensione base di dati e sintassi linguaggio
  - Richiede conoscenza esperta

# Riempimento di moduli

- Principalmente per introdurre o ritrovare dati
- Schermo come modulo cartaceo
- Dati messi in posti rilevanti
- Richiede
  - Buon progetto
  - Correzioni ovvie
  - Facilitazioni

Go-faster Travel Agency Booking

Please enter details of journey:

Start from: Lancaster

Destination: Atlanta

Via: Leeds

First class /  Second class /  Bargain

Single /  Return

Seat number:

# Fogli elettronici

- Primo foglio elettronico VISICALC, poi Lotus 1-2-3, oggi MS Excel
- Variazione sofisticata compilazione di moduli
  - Griglia di celle contenenti valori o formule
  - Formula può coinvolgere valori di altre celle  
es. Somma di celle in colonna
  - Utente può introdurre e modificare dati.
  - Foglio mantiene coerenza.

# Interfacce WIMP

Windows

Icons

Menus

Pointers

... o windows, icons, mice, and pull-down menu

- Stile di base per maggioranza sistemi interattivi, specie PC

# Interfacce point and click

- Usate in ..
  - Multimedia
  - Navigatori web
  - Iper testi
- Fare clic su entità
  - Icone, collegamento testuale o posizione su mappa
- Minimizzano battitura



# Elementi di interfacce WIMP

windows, icons, menus, pointers

+++

pulsanti, barre di strumenti,  
tavolozze, finestre di dialogo

# Finestre

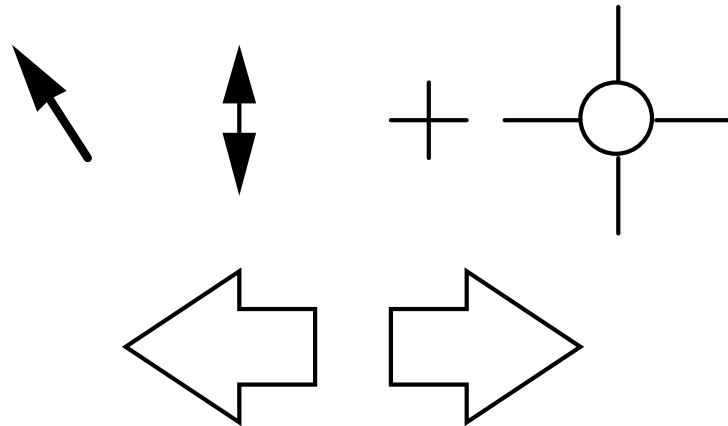
- Aree schermo che si comportano in modo indipendente
  - Possono contenere testo o grafica
  - Possono essere mosse o ridimensionate
  - Possono sovrapporsi e oscurarsi, o essere affiancate
- Barre di scorrimento
  - Utente sposta contenuto finestra verso alto, basso, sinistra, destra
- Barra titolo
  - Descrive nome finestra

# Icone

- Piccola immagine
- Rappresenta oggetto di interfaccia
  - Es. Finestra o azione
- Finestre possono essere chiuse (iconificate)
  - Riduzione permette accesso a più finestre
- Icone possono essere molte e diverse
  - Altamente stilizzate
  - Rappresentazioni realistiche
- Disposizione fisse o manipolabili

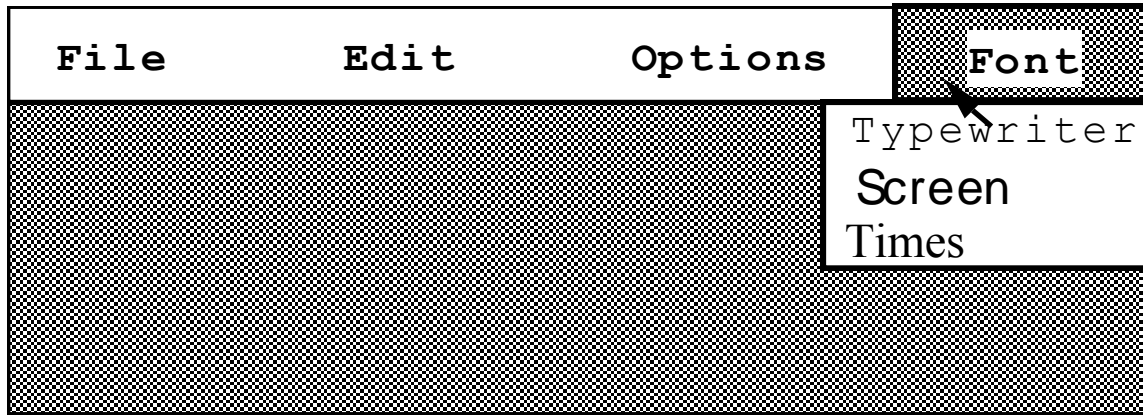
# Puntatori

- Componente importante
  - Stile WIMP basato su puntamento e selezione di elementi
- Differenza puntatore / dispositivo di puntamento
  - Es. mouse, trackpad, joystick, trackball, tasti
- Immagini grafiche per comunicare modalità



# Menu

- Scelta da offerta su schermo di operazioni o servizi
- Opzione desiderata selezionata con puntatore



problema – occupano molto spazio

soluzione – pop-up: menu appare quando necessario

# Tipi di menu

- Barra menu (normalmente) in cima a schermo, menu si apre in basso
  - Menu [pull-down](#) – mouse premuto, voci in basso
  - Menu drop-down – clic su mouse rivela menu
  - Menu fall-down – mouse si muove sulla barra!
- Menu contestuale appare in posizione
  - Menu pop-up – azioni per oggetto selezionato
  - Menu a pizza – disposti in cerchio
    - Più facile selezionare elemento (area bersaglio più grande)
    - Più veloce (stessa distanza da ogni opzione)  
... ma non molto usato



# Altre particolarità

- Menu a cascata
  - Struttura gerarchica menu
  - Selezione apre nuovo menu
  - Potenzialmente illimitato
- Acceleratori di tastiera
  - Combinatori di tasti – stesso effetto di voce di menu
  - Due tipi
    - Attivi quando menu aperto – solitamente prima lettera
    - Attivi quando menu chiuso – solitamente Ctrl + lettera

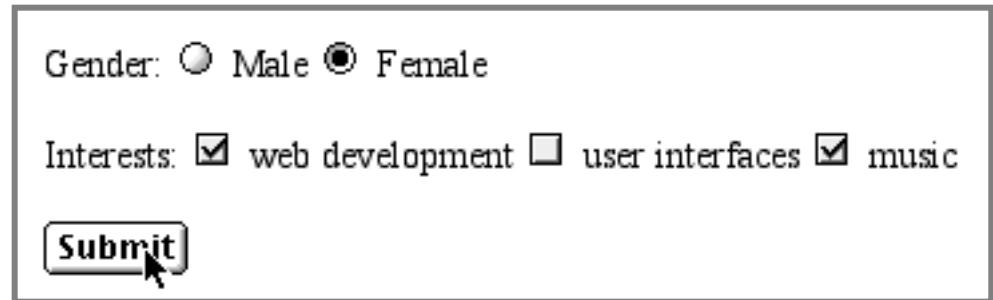
# Questioni sul progetto di menu

- Quale tipo usare
- Cosa includere in menu
- Parole da usare (azione o descrizione)
- Come raggruppare elementi
- Scelta acceleratori da tastiera



# Pulsanti

- Regioni dello schermo individuali e isolate, possono essere selezionate per invocare azione



Gender:  Male  Female

Interests:  web development  user interfaces  music

- Tipi speciali
  - radio button
    - insieme di scelte mutuamente esclusive
  - check box
    - insieme di scelte non esclusive

# Barre degli strumenti

- Lunghe file di icone
- Accesso rapido ad azioni comuni
- Spesso personalizzabile
  - Scelta di quale barra vedere
  - Scelta di quali opzioni per ogni barra

# Tavolozze e menu separabili

- Problema  
menu non dove desideratii
- Soluzione  
tavolozze – piccole finestre di azioni
  - mostrate/nascoste con opzione di menu  
es. forme disponibili in pacchetti di disegnomenu separabili e fissabili
  - menu si stacca per diventare tavolozza

# Finestre di dialogo

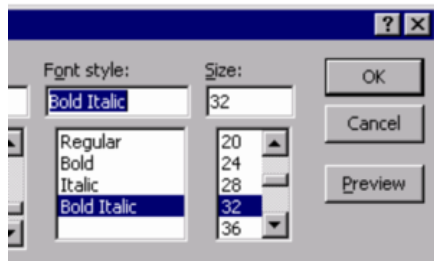
- Finestre che emergono per informare di evento importante o richiedere informazione
  - Es. Finestra salvataggio documento
- Modalità

# Comprendere e scegliere widget (congegni)

widget – parti componenti della GUI

Cosa fanno

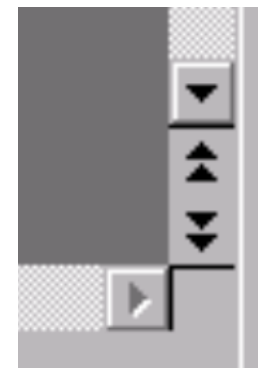
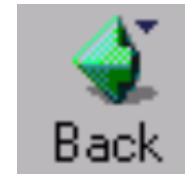
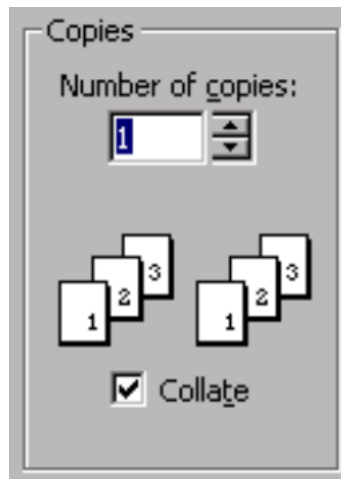
A cosa sono adatte



# Congegni?

- Elementi individuali su schermo di GUI
  - Spunte, menu, barre strumenti, pulsanti, ecc.
- Tre aspetti:
  - Apparenza      - come si presentano
  - Interazione    - come si comportano
  - Semantica      - cosa significano

# Apparenza



# Apparenza include parole

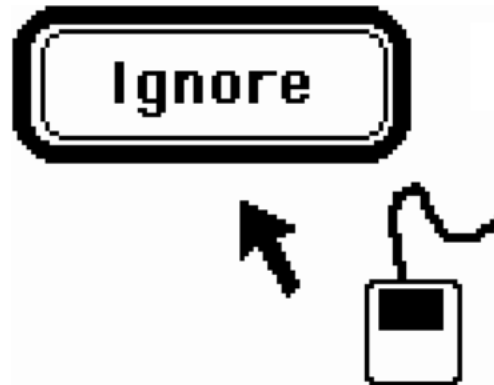
- Verbi – parole di azione
  - Abbandona, esci, rendi grassetto, rendi corsivo
- Aggettivi – parole per stato / descrizione
  - Grassetto, corsivo
- Sostantivi – solitamente forma di descrizione
  - Times New Roman, US Letter
- Attenzione ai miscugli
  - Rendi grassetto + corsivo!!?!



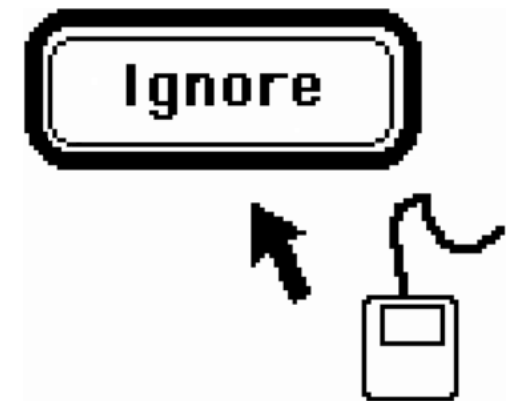
# Comportamento I



Muovi mouse su pulsante  
– evidenziazione



Muovi mouse fuori da bersaglio  
con pulsante ancora premuto  
– rimozione evidenziazione



Rilascia mouse  
– non accade nulla

# Comportamento II

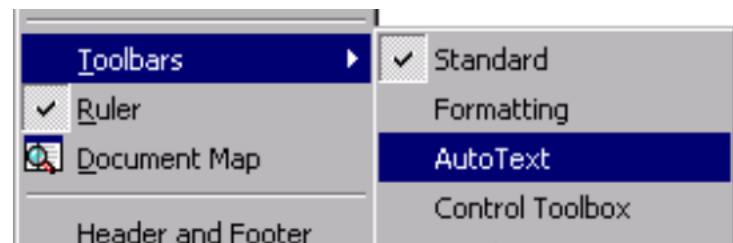
- Parti realizzate da toolkit
  - Sono corrette?
- Parti controllate da progettista
  - es. disegnare, interazione tra congegni
- Attenzione a temporizzazioni
  - es. grandi selezioni in applicazioni Windows

# Creatore GUI stabilisce semantica

- Significato azione
- Semantica solitamente definita da progettista
  - Congegni possono collegarsi direttamente a base di dati
  - Progettista stabilisce collegamenti
- Considerare separatamente
  - Prima significato - cosa si vuole che faccia
  - Poi apparenza - come farlo
- Scegliere congegno adatto a lavoro

# Cosa vogliamo

- Azioni
  - Solitamente menu, pulsanti, o barre di strumenti
- Definizione di stati / opzioni
  - Solitamente spunte, spunte esclusive, selettori
- Ma ...
  - Menu possono essere usati per stabilire stato, ecc.



# Quanti?

- Una fra diverse opzioni
  - Spunte esclusive, selezione da menu
- Zero, una o più opzioni
  - Spunte, menu a scelta multipla
- Scelta libera
  - Offrire scorciatoie a materiale recente o tipico
  - Campi testo a linea singola spesso orribili!

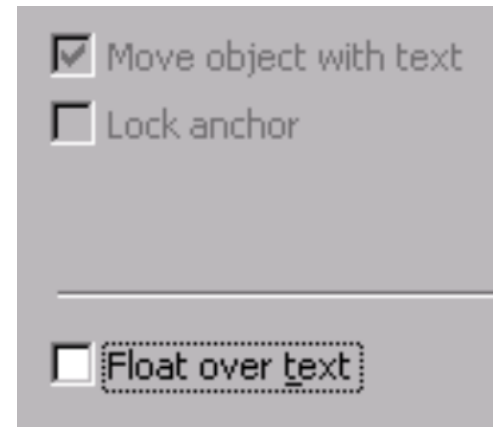
favourite colour?

red  blue  orange

- Word wrap text in autoshape
- Resize autoshape to fit text
- Rotate text within autoshape by 90°

# E inoltre

- Numerosità
  - fissata es. grassetto, corsivo, sottolineato
  - variabile es. lista di fonti
  - Scorrimento elenco telefonico ...
- Liveness (vivezza)
  - Scolorire opzioni inattive
- Interazioni dinamiche
  - Alcune scelte dipendono da altre



# Indipendenza da congegno

- Semantica azioni alto livello
- Realizzabile mediante interazioni diverse
  - Raccogliere argomenti
  - Selezionare azioni
- Concetto di Action
  - Accesso a stessa funzionalità tramite controlli diversi
  - Fornisce:
    - Stringhe per descrivere azione a utente
    - Icone per rappresentare azione
    - Stato di abilitazione/disabilitazione centralizzato.
  - Ascoltatori cambiamenti di stato dis/abilitano generazione eventi su elemento, modificano presentazione.

# Aspetti di definizione di sistema interattivo visivo

	<b>Componenti singole</b>	<b>Coordinamento fra componenti</b>
<b>Rappresentazione</b>	Identificazione tipo di componente.	Progetto layout
<b>Comportamento</b>	Reazioni ad azioni utente e cambi stato	Propagare eventi e sincronizzare viste



# Identificazione tipo componente I

- Rappresentazione dominio
  - Dati da rappresentare
    - collezioni o elementi singoli?
      - se collezioni: ordinate, non ordinate?
      - se elementi singoli: di che tipo? (es. numerici, booleani, stringhe, enumerato)
        - » se numerici: qualsiasi valore o range limitato?
  - Processi da attivare

# Identificazione tipo componente II

- Gestione interazione
  - Elementi di interazione generica
  - Elementi di navigazione in interfaccia
  - Supporto a comportamenti specifici

# Comportamenti da supportare:

- Attivazione di processi
- Interrogazione di dati
- Valorizzazione di dati
- Modifiche di caratteristiche di presentazione
- Creazione ed eliminazione di elementi

# Politiche di coordinamento (del comportamento)

- Politiche di abilitazione e disabilitazione
- Propagazione attivazioni
- Attivazioni concorrenti

# Politiche di coordinamento (della presentazione)

- Disposizione complessiva interfaccia
- Vincoli reciproci fra elementi
  - spostamenti coordinati
  - relazioni ammissibili fra tipi di elementi
- Coerenza con altri tipi di interfaccia
- Trasformazioni globali presentazione