



Laboratorio di Sistemi Interattivi

Lezione 2: Introduzione alla modellazione di sistemi interattivi



Sistemi Interattivi
Università La Sapienza

1

Sommario

- Motivazioni
- Elementi di definizione di un sistema interattivo
- Modellazione orientata agli oggetti
- Viste sul modello
- Lo strumento UML



Sistemi Interattivi
Università La Sapienza

2

Modellare per progettare

- Identificare i confini del sistema
- Identificare le possibili condizioni in cui si può trovare il sistema
- Identificare le possibili risposte a stimoli esterni
- Prevedere le possibili evoluzioni

Modellare è porsi delle domande
Progettare è tentare delle risposte



Sistemi Interattivi
Università La Sapienza

3

Perché modellare il software

- Necessità di identificare componenti del progetto
- Necessità di valutare la rispondenza del progetto ai requisiti
- Necessità di mantenere il software nel tempo



Sistemi Interattivi
Università La Sapienza

4

Perché modellare il software interattivo

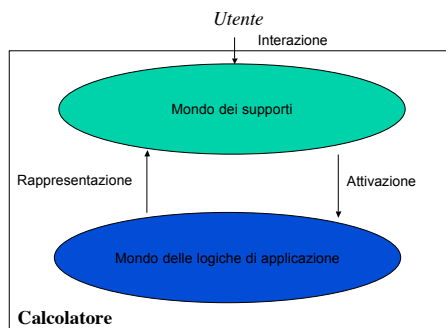
- Diversi livelli di definizione da mantenere coerenti
- Identificazione linguaggi utente e valutazione corrispondenza con i linguaggi di interazione
- Generazione automatica di codice
- Valutazione corrispondenza fra strumenti di interazione e logiche di applicazione



Sistemi Interattivi
Università La Sapienza

5

Sistemi interattivi visivi



Sistemi Interattivi
Università La Sapienza

6

Problemi

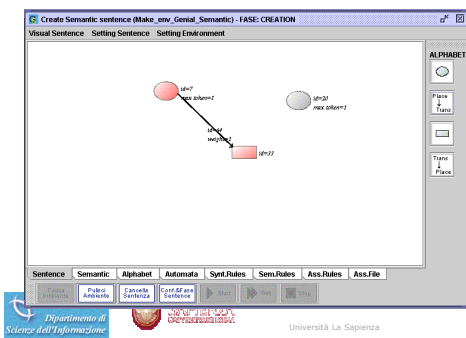
- Relazione fra supporto di rappresentazione e supporto di interazione
- Corrispondenze fra logiche di applicazione e logiche dell'interazione
- Gestione dei processi di rappresentazione e di attivazione



Sistemi Interattivi
Università La Sapienza

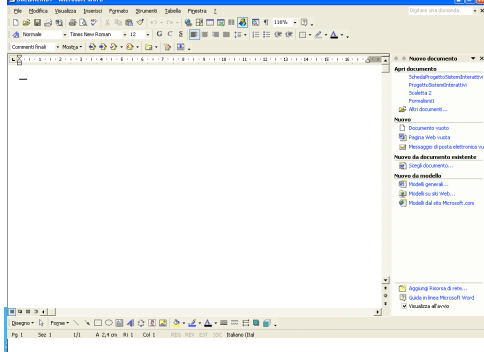
7

Un'interfaccia per un linguaggio di modellazione



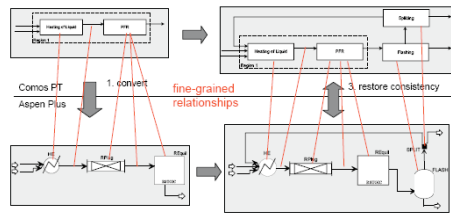
8

Un'interfaccia per funzionalità generica



9

Viste astratte e concrete

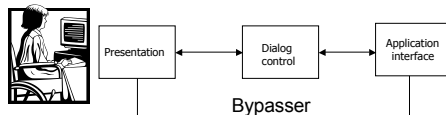


UIMS

- User Interface Management System: controlla il rapporto tra presentazione e funzionalità
- *Definisce la separazione tra la semantica dell'applicazione e la presentazione e così migliora:*
 - la portabilità - gira su sistemi differenti
 - la riusabilità - il riuso dei componenti taglia i costi
 - le interfacce multiple - per accedere alle stesse funzionalità
 - modificabilità secondo esigenze specifiche (customizability) - del progettista e dell'utente

L'interfaccia separabile

Modello di Seeheim

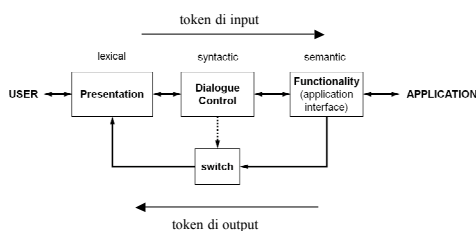


Ripercorre la divisione Lessico, Sintassi, Semantica
Controllo del dialogo monolitico
Bypasser rende descrizione formale difficile

Il modello di Seeheim

- Sviluppato nel corso del Workshop on User Interface Management Systems a Seeheim da un gruppo di lavoro i cui membri erano Jan Derksen, Ernest Edmonds, Mark Green, Dan Olsen, e Robert Spence.
- Quando questo modello fu proposto (primi anni '80), non era mai stato ancora usato come base per un UIMS (User Interface Management System) quindi non c'era stato ancora modo di valutarlo ...
- ... ma era un primo passo molto importante
- "... because of Seeheim ...
... we think differently!" (Alan Dix)

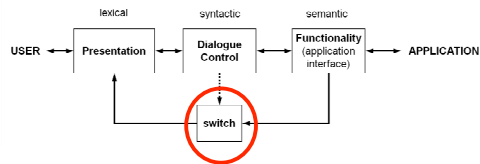
Il modello di Seeheim



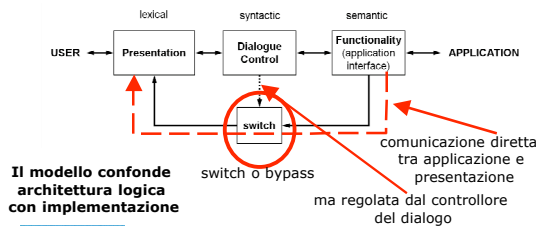
Il modello di Seeheim (continua)

- Differenti tipi di feedback:
 - lessicale – movimento del mouse
 - sintattico – evidenziazione dei menu
 - semantico – la somma dei numeri cambia su una calcolatrice
- Il feedback semantico è ovviamente più lento
- Potrebbe esserci bisogno di un feedback semantico più veloce (es. disegno a mano libera, o evidenziazione del cestino quando un elemento viene trascinato nelle sue vicinanze)

Il modello di Seeheim (continua)

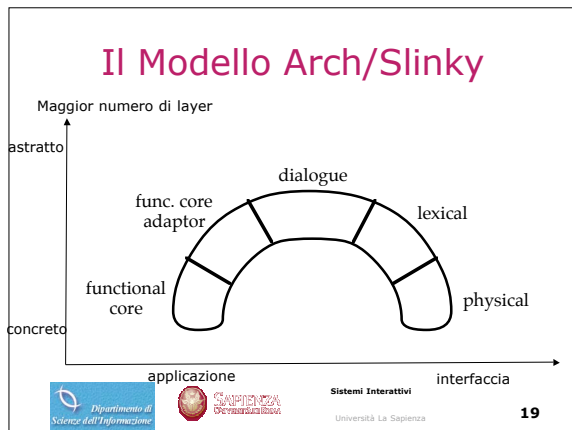


Il modello di Seeheim (continua)



Il modello di Seeheim (continua)

- Le due principali difficoltà del modello di Seeheim:
 - quando si rimpiazza un componente di presentazione, occorre riscrivere il dialogo per adattarlo alle nuove caratteristiche del nuovo componente
 - il dialogo tende ad essere basato sulla presentazione, e comunque occorre modificare la presentazione ogni volta che si modifica il dialogo



Il Modello Arch/Slinky (cont.)

- più layer! – vengono distinti i livelli lessicale/fisico
- come in una molla a 'slinky' layer differenti possono essere più consistenti (importanti) in sistemi differenti ...
- ... o in differenti componenti
- Difetto -- non si comporta meglio del modello di Seeheim nel fornire indicazioni sul preciso contenuto delle componenti o sul loro processo di progettazione.

Dipartimento di Scienze dell'Informazione

Sistemi Interattivi

Università La Sapienza

20

Modello a oggetti

- Approccio monolitico vs. approccio a componenti
- Interfaccia Utente come collezione di oggetti di interazione (widgets)
- Utente manipola oggetti direttamente
- Gli oggetti sono responsabili della trasmissione delle azioni dell'utente verso l'applicazione

User

UI

Application

Dipartimento di Scienze dell'Informazione

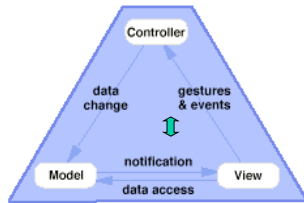
Sistemi Interattivi

Università La Sapienza

21

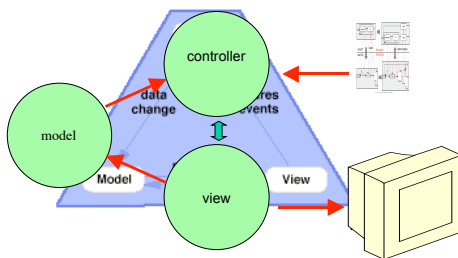
L'interfaccia separabile

Modello MVC (Model View Controller)



L'interfaccia separabile

Modello MVC (Model View Controller)

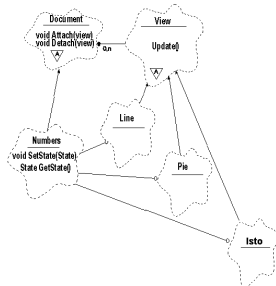


Problemi di MVC

- MVC è in buona parte un modello pipeline :
input → control → model → view → output
- Ma nelle interfacce grafiche
 - l'input ha significato solo in relazione all'output
- esempio mouse click
 - dobbiamo sapere *cosa* è stato clickato
 - il controller deve decidere cosa fare del click
 - ma è la view a sapere cosa è mostrato dove!
- in pratica il controller deve 'parlare' alla view
 - la separazione non è completa

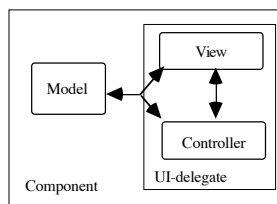
Evoluzioni di MVC

Document-View



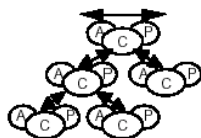
Evoluzioni di MVC

Model-Delegate



Evoluzioni di MVC

Presentation Abstraction Control



Struttura gerarchica ad agenti
Controllore media fra Abstraction e Presentation
Coordina agenti più in basso nella gerarchia

Problemi nella definizione di un sistema interattivo

	Componenti singole	Coordinamento fra componenti
Rappresentazione	Identificazione del tipo di componente necessaria.	Progetto del layout
Comportamento	Reazioni ad azioni utente e cambiamenti di stato	Propagazione di eventi e sincronizzazione viste

Identificazione tipo componente

- Rappresentazione del 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 in un range limitato?
 - Processi da attivare
 - Gestione dell'interazione
 - Elementi di interazione generica
 - Elementi di navigazione nell'interfaccia
 - Supporto a comportamenti specifici

Comportamenti da supportare:

- attivazione di processi
- interrogazione di dati
- valorizzazione di dati
- modifiche di caratteristiche della presentazione
- creazione ed eliminazione di elementi

Politiche di coordinamento (del comportamento)

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



Sistemi Interattivi
Università La Sapienza

31

Politiche di coordinamento (della presentazione)

- Layout complessivo dell'interfaccia
- Vincoli reciproci fra elementi
 - spostamenti coordinati
 - relazioni ammissibili fra tipi di elementi
- Consistenza con altri tipi di interfaccia
- Trasformazioni globali della presentazione



Sistemi Interattivi
Università La Sapienza

32
