

# Lezione

## Regole di Design

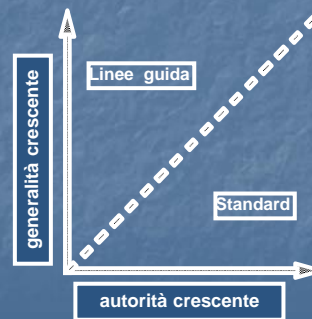
## Regole di design

Progettazione per ottenere la massima usabilità  
– lo scopo del design d'interazione

- Principi di usabilità
  - comprensione generale
- Standard e linee guida
  - direttive per il design
- I design pattern
  - sintetizzano e riusano le conoscenze di design

# Tipi di regole di design

- principi
  - regole di design astratte
  - bassa autorità
  - alta generalità
- standard
  - regole di design specifiche
  - alta autorità
  - applicabilità limitata
- linee guida
  - minore autorità
  - applicabilità più generale



Interazione uomo-macchina – A. Dix, J. Finlay, G. D. Abowd, R. Beale

Copyright © 2004 – The McGraw-Hill Companies srl

# Principi di Usabilità

- ❖ I principi di usabilità sono un mezzo più generale per comprendere l'usabilità
- ❖ guidati teoricamente da conoscenza psicologica, computazionale e sociologica

# Principi di Ausilio all'Usabilità

## ➤ Capacità di apprendimento o *learnability*

- ❖ facilità con cui nuovi utenti possono iniziare un'interazione effettiva e ottenere massime prestazioni

## ➤ Flessibilità

- ❖ la molteplicità di modi in cui l'utente e il sistema scambiano informazioni

## ➤ Robustezza

- ❖ il livello di sostegno fornito all'utente nel determinare un comportamento di successo rispetto ai suoi goal

# Principi di learnability

## **Predicibilità**

- ❖ determinare l'effetto di azioni future sulla base della storia dell'interazione passata

## **Sintetizzabilità**

- ❖ assestare l'effetto di azioni passate
- ❖ onestà immediata vs. ritardata del sistema interattivo (es. la scrivania visuale)

## **Familiarità**

- ❖ come la conoscenza precedente si applica al nuovo sistema
- ❖ intuitività; caratteristiche intrinseche degli oggetti visuali che migliorano la familiarità del sistema

## **Generalizzabilità**

- ❖ estendere conoscenza specifica di interazione a nuove situazioni

## **Consistenza**

- ❖ similitudini nel comportamento input/output che nascono da situazioni o obiettivi simili. E' un concetto relativo a diverse proprietà dell'interazione
- ❖ consistenza nel nominare i comandi
- ❖ consistenza nell'invocazione dei comandi ecc

## Un esempio

**Task:** *Stampa 3 copie del documento corrente*  
dell'applicazione Microsoft Word

Suddivisione del task in azioni

**Azione 1:** Seleziona il menù *F*ile

**Azione 2:** Seleziona la voce *S*tampa

**Azione 3:** Seleziona la stampante

**Azione 4:** Specifica le pagine da stampare

**Azione 5:** Specifica il numero di copie  
(... altro)

**Azione 6:** Invio stampa

## Learnability: un esempio (cont.I)

all'inizio del task

*predicibilità:* in questo punto è supportata perché l'utente sfrutterà la propria conoscenza della storia delle precedenti interazioni, per *prevedere* che la scelta del menù *F*ile offrirà come risultato un elenco di voci, che include quella relativa alla stampa del file.

dopo l'azione 2

*sintetizzabilità:* l'utente sarà in grado di valutare le conseguenze delle interazioni precedenti sullo stato corrente, perché la selezione della voce *S*tampa provocherà l'apertura di una finestra di dialogo, nella quale l'utente dovrà specificare tutti i dati relativi alla stampa. *Onestà immediata*

prima dell'azione 6

*familiarità:* l'utente sfrutterà la propria esperienza con i pulsanti del mondo reale per comprendere che *ok* è un pulsante che va premuto per far sì che il task vada a compimento. Il concetto di *affordance* di un oggetto: la forma ricorda come l'oggetto andrebbe manipolato.



## Learnability: un esempio (cont.II)

### all'inizio del task

*generalizzabilità*: l'utente ha già esperienza con un'azione simile all'interno di un'altra applicazione (stampa diapositive di Powerpoint).

*consistenza*: rispetto alla sequenza di azioni da fare per task simili

rispetto ai nomi utilizzati per le singole voci

rispetto al modo in cui il sistema risponde alle azioni

rispetto al look and feel delle varie schermate

## Principi di flessibilità

### Iniziativa nel dialogo

Assenza di vincoli imposti dal sistema sul dialogo input

**dialogo system pre-emptive vs. dialogo user pre-emptive**

Massimizzare la predominanza dell'utente e minimizzare quella del sistema

### Multithreading

*Un thread di un dialogo è quella parte del dialogo utente-sistema relativa a un dato task utente.*

capacità del sistema di supportare l'interazione con l'utente su più task contemporaneamente

### Principi correlati:

❖ concorrente vs. interfogliato;

❖ multimodalità

Es. un sistema a finestre: supporta in modo naturale un dialogo *multi-threaded* che è interfogliato tra diversi task che si sovrappongono.

## Principi di flessibilità (cont.)

### Migrabilità di un task

passaggio di responsabilità tra utente e computer per l'esecuzione di un task

Es. Il controllo dello spelling è fatto meglio se in maniera cooperativa tra utente e sistema

### Personalizzazione

modificabilità dell'interfaccia utente da parte dell'utente (*adattabilità*) o del sistema (*adattività*)

## Principi di flessibilità (cont.)

### Sostituibilità

concedere di sostituire tra loro valori di input e di output equivalenti

➤ per l'input: la possibilità di immettere come valori dei margini di pagina, espressioni o valori equivalenti, anche in unità di misura diverse.

➤ per l'output: *molteplicità nella rappresentazione* (ogni rappresentazione fornisce una diversa prospettiva dello stato del sistema).

➤ pari opportunità: nessuna distinzione tra input e output sull'interfaccia.

Es. un applicativo di disegno in cui l'utente traccia una linea e il sistema ne calcola la lunghezza opp. viceversa l'utente dà le coordinate e il sistema traccia la linea

## Flessibilità: l'esempio

**Task:** *Stampa 3 copie del documento corrente dell'applicazione Microsoft Word*


*iniziativa nel dialogo:* è quasi sempre un dialogo *user pre-emptive*. Il sistema prende il sopravvento solo per segnalare problemi con la stampa. Una finestra di dialogo modale che segnala:

Impossibile stampare il documento  
Nome file "xxxx", stampante "yyy", ora di avvio "12:00"  
Per accedere alla coda di stampa, fare click sull'icona della stampante e scegliere *risoluzione problemi* sul menù.

## Flessibilità: l'esempio (cont.I)

*multi-threading:* come per tutti i sistemi a finestre questo principio è supportato. L'utente può passare a eseguire un altro task all'interno di un'altra applicazione mentre sta eseguendo una delle azioni da 2 a 5. Inoltre può passare a un altro task, come la modifica del documento, dopo che l'ultima azione è stata compiuta e mentre il sistema sta rispondendo mandando in stampa il documento.

*migrabilità del task:* no, in questo caso è sempre a carico dell'utente

*sostituibilità:* dell'input - il "numero di copie" si può scrivere editando il campo di testo oppure si può inserire scorrendo le freccette 

*personalizzabilità:* non relativamente al task specifico. In generale, l'utente può ad esempio personalizzare l'interfaccia decidendo di aggiungere il pulsante 'stampante' alla barra degli strumenti. L'interfaccia è adattabile ma non adattiva.

# Principi di robustezza

## Osservabilità

Capacità dell'utente di valutare lo stato interno del sistema dalla sua rappresentazione percepibile

- ❖ esplorabilità o navigabilità;
- ❖ default;
- ❖ raggiungibilità;
- ❖ persistenza;

## Recuperabilità

capacità dell'utente di intraprendere azioni correttive una volta rilevato un errore

- ❖ recupero in avanti
- ❖ recupero indietro
- ❖ sforzo commisurato

# Principi di robustezza

## Risposta

come l'utente percepisce la comunicazione con il sistema, in termini del **tempo** che il sistema impiega ad esprimere cambiamenti di stato

## Conformità dei task

il grado con cui i servizi offerti dal sistema supportano tutti i task degli utenti

- ❖ completezza dei task;
- ❖ adeguatezza dei task



# Robustezza: l'esempio

**Task:** *Stampa 3 copie del documento corrente* dell'applicazione Microsoft Word

*osservabilità:*

quando la stampa è avviata, sull'interfaccia compare l'icona della stampante con un contatore delle pagine via via stampate. Anche sull'interfaccia del sistema operativo (es. Windows) compare un pulsante con l'icona della stampante, che può essere premuto per visualizzare lo stato del processo di stampa. (*persistenza* e *navigabilità*). Inoltre tutti i campi della finestra di stampa presentano dei valori di *default* che suggeriscono all'utente il tipo di input che il sistema si aspetta. Infine l'utente può spostarsi tra gli stati osservabili del sistema, finché l'azione finale di avvio stampa non è eseguita (*raggiungibilità*).

*ripristinabilità:* collegata alla raggiungibilità. La *recuperabilità all'indietro* è sempre garantita dal tasto 'annulla' finché non si compie l'azione di avvio stampa. Dopo, l'utente può intervenire dalla finestra di gestione dei processi, per annullare la stampa dalla coda e ripetere il task dall'inizio (*recuperabilità in avanti*).

*velocità di risposta:* immediata per tutte le azioni che compongono il task

*conformità dei task:* è riferita a tutti i task che il Word consente di eseguire. Quello dell'esempio è un task *adeguato* perché è rappresentato sull'interfaccia come capito dall'utente e supportato dal sistema. L'utente comprenderà facilmente come il task possa essere eseguito con il paradigma di interazione che ha a disposizione.

## Ricapitolando...

### Principi di Usabilità

Massimizzare i benefici di un buon progetto astruendo le proprietà generali e rendendole ripetibili così da poter guidare il progetto di nuovi sistemi interattivi

**Il successo nell'usabilità richiede sia un tocco di creatività (nuovi paradigmi) che una pratica decisa e basata su principi.**

# Gli standard

- stabiliti da organismi nazionali o internazionali per assicurare la conformità da parte di una vasta comunità di progettisti, gli standard richiedono una teoria di base corretta e una tecnologia che cambia lentamente
- standard hardware più diffusi di quelli software: autorità elevata e basso livello di dettaglio
- ISO 9241 definisce l'usabilità come efficacia, efficienza e soddisfazione con cui determinati utenti eseguono i compiti *in un determinato contesto d'uso*

## Gli standard (*cont.*)

- **ISO 9241-11 *Ergonomics of Human System Interaction -Guidance on Usability*** (1998)

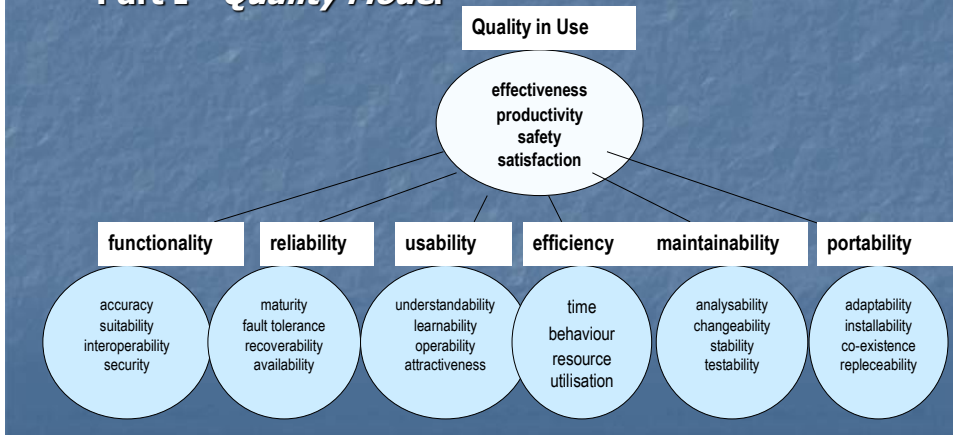
**Usability** is "*the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use*"

- *Usability of a software product* is one of the factors contributing to **quality in use**.

## Gli standard (*cont.*)

- ISO/IEC-9126 (2001) standard on *Software Engineering — Product Quality*

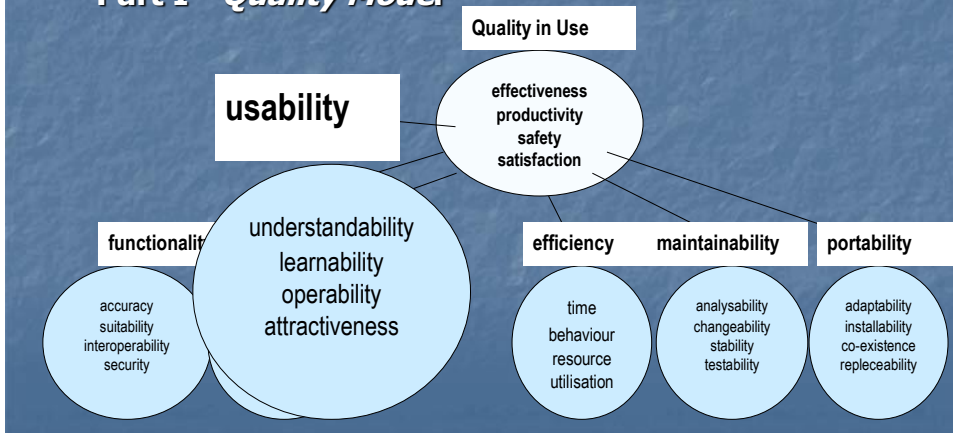
### Part I - *Quality Model*



## Gli standard (*cont.*)

- ISO/IEC-9126 (2001) standard on *Software Engineering — Product Quality*

### Part I - *Quality Model*



## Gli standard (*cont.*)

- **ISO/IEC-9126 (2001) standard on *Software Engineering — Product Quality***

In **Part I - *Quality Model***:

- **Usabilità è *the capability of a software product to be understood, learned, used and attractive to the user, when used under specified conditions***
- Qui si sottolinea il fatto che l'usabilità di un prodotto software non può essere valutata indipendentemente dal particolare contesto in cui il prodotto è usato.

## Le linee guida

- più suggestive e generali
- molti manuali e rapporti contenenti linee guida
- linee guida astratte (principi) applicabili durante le prime attività del ciclo di vita
- linee guida dettagliate (guide di stile) applicabili durante le attività successive del ciclo di vita
- comprendere le motivazioni delle linee guida aiuta a risolvere i conflitti



# Le regole d'oro e le euristiche

- Regole di design di granularità grossa
- Liste di controllo utili per un buon design
- Meglio progettare sulla base di euristiche che farlo a caso!
- Diversi tipi di euristiche; es.:
  - 10 euristiche di Nielsen (vedi Capitolo 8)
  - 8 regole d'oro di Shneiderman
  - 7 principi di Norman

## Le 8 regole d'oro di Shneiderman

1. Preservare la coerenza
2. Consentire agli utenti abituali di usare comandi rapidi
3. Offrire un feedback informativo
4. Progettare dialoghi provvisti di chiusura
5. Offrire una prevenzione e una gestione semplice degli errori
6. Permettere un'inversione semplice delle azioni
7. Supportare il controllo interno
8. Ridurre il carico della memoria a breve termine

## Regola d'oro no. 1: Coerenza

- Sequenze di operazioni simili dovrebbero essere effettuate sempre con lo stesso tipo di azioni
- Usare stesse convenzioni e terminologia per i prompt, menù, colori, etc.
- Limitare il numero delle eccezioni

## Regola d'oro no. 2: Snellimenti

- Con l'aumento della frequenza d'uso, ridurre il numero delle interazioni e aumentare la velocità
- Permettere l'uso di macro e abbreviazioni
- Abbreviare i tempi di risposta e aumentare la velocità di visualizzazione

## Regola d'oro no. 3: Feedback Informativo

- Ad ogni azione dell'utente dovrebbe corrispondere un feedback del sistema
- Il responso corrispondente ad azioni frequenti dovrebbe essere modesto, mentre azioni occasionali o primarie dovrebbero dare luogo a responsi più dettagliati
- Mostrare esplicitamente i cambiamenti, con presentazione degli oggetti di interesse

## Regola d'oro no. 4: Chiusura

- Organizzare le sequenze di azioni in gruppi, prevedendo feedback informativo alla fine di ciascun gruppo di azioni
- Dare all'utente la sensazione di poter scaricare la mente alla fine di una sequenza di azioni per dedicarsi interamente al task successivo

## Regola d'oro no. 5: Errori

- Progettare l'interfaccia in modo che sia quanto più difficile commettere degli errori
- In presenza di un errore, il sistema dovrebbe riconoscerlo ed offrire istruzioni semplici, costruttive e specifiche per risolverlo
- Azioni errate dovrebbero lasciare immutato lo stato del sistema, oppure il sistema stesso dovrebbe fornire informazioni su come ripristinare lo stato

## Prevenire gli errori

- La frequenza degli errori da parte anche di utenti esperti è sorprendentemente alta
- Migliorare i messaggi di errori: è sperimentalmente provato che ciò aumenta la possibilità di usare il sistema con successo
- Aiutare gli utenti ad evitare gli errori seguendo 3 possibili tecniche:
  - ❖ Coppie corrispondenti corrette
  - ❖ Sequenze complete
  - ❖ Comandi corretti



## Correggere coppie di comandi corrispondenti

- La mancata chiusura di una parentesi aperta può essere prevenuta usando un editore "smart"
- VANTAGGI: Evita gli errori e la necessità di gestirli, il testo non si trova mai in una forma sintatticamente scorretta
- SVANTAGGI: Potrebbe essere preferito un approccio meno rigido e meno restrittivo

## Raggruppare sequenze di steps in singole azioni

- Una fonte di errori è il mancato completamento di una fissata sequenza di passi.
- Si cerca di evitare questo problema raggruppando sequenze di passi in singole azioni
- A volte l'utente ha bisogno di accedere alle operazioni atomiche piuttosto che alla sequenza di passi raggruppati: può essere preferibile lasciare all'utente stesso la possibilità di definire delle "macro"
- La scelta delle sequenze da raggruppare può essere fatta in base ad uno studio sull'uso effettivo del sistema e sul tipo di errori ricorrenti

## Correzione automatica dei comandi

- Altra causa di errori è l'uso di un linguaggio di comandi: errori di battitura, di nomi di comandi sbagliati, nomi di file inesistenti, combinazioni di tasti sbagliate, etc.
- Il completamento automatico dei comandi può evitare questo tipo di errori, limitando inoltre l'uso della tastiera
- Possibile svantaggio: non sempre la correzione automatica completa il comando nella maniera voluta
- Lo stesso principio si applica alla manipolazione diretta: il sistema può presentare all'utente solo le azioni ammissibili, e l'utente seleziona quelle che desidera, per esempio per mezzo del mouse

## Regola d'oro no. 6: Reversibilità

- Rendere le azioni reversibili il più possibile
- Incoraggiare l'utente a esplorare opzioni sconosciute dando la sensazione di poter tornare indietro senza danno
- Scegliere opportunamente le unità di azioni reversibili

## Regola d'oro no. 7: Controllo

- Dare all'utente esperto la sensazione di essere il responsabile del sistema, che non fa altro che rispondere alle sue azioni
- Evitare reazioni sorprendenti, sequenze di azioni ripetitive, difficoltà ad ottenere le informazioni richieste o ad eseguire le azioni desiderate
- Rendere l'utente il promotore delle azioni invece che il ricettore di esse

## Regola d'oro no. 8: Memoria

- La memoria umana di breve termine può elaborare un limitato numero di informazioni ( $7 \pm 2$  elementi di informazione)
- Mantenere il display semplice, ridurre la frequenza di spostamenti di finestre, dare il tempo necessario per allenarsi
- Fornire, se necessario, accesso on-line alle forme sintattiche, abbreviazioni di comandi, codici



## I 7 principi di Norman

1. Bisogna usare sia la conoscenza presente nel mondo sia la conoscenza mentale.
2. Si deve semplificare la struttura dei compiti.
3. Si rendano le cose visibili: bisogna gettare un ponte sul golfo dell'esecuzione e sul golfo della valutazione.
4. Le corrispondenze vanno chiarite.
5. Si sfrutti il potere dei vincoli, sia naturali sia artificiali.
6. Bisogna progettare gli errori.
7. Quando tutto il resto non ha successo, si creino degli standard.

## I design pattern HCI

- Un approccio per riutilizzare le conoscenze sulle soluzioni di progettazione di successo
- Hanno avuto origine in architettura: Alexander
- Un pattern è una soluzione parzialmente specificata a un problema ricorrente all'interno di un contesto specifico.
- Esempi
  - Light on Two Sides of Every Room (architettura)
  - Go back to a safe place (HCI)
- I pattern non esistono isolati, ma sono collegati ad altri pattern in *linguaggi* che consentono di creare design completi



# I design pattern HCI (cont.)

Un Design Pattern HCI cattura l'essenza di una soluzione di successo a un problema di usabilità ricorrente nei sistemi interattivi. Esso consiste di

- un nome (*name*),
- un grado di validità (*ranking*),
- un esempio tangibile (*image*),
- un contesto (*context*),
- una descrizione del problema (*problem statement*),
- esempi (*examples*),
- la soluzione (*solution*),
- uno schizzo (*sketch* o *diagram*),
- riferimenti (*references to other patterns*)

## Borcher's patterns to interaction design > interactive music exhibits

4.2 HCI Pattern Language 139

**H13 IMMERSIVE DISPLAY \***




Figure 4.31: CAVE in the Ars Electronica Center Linz.

...you have decided to create an exhibit that several people can experience simultaneously—COOPERATIVE EXPERIENCE (ED). Now you need to find a way to design the visual output of such a system.

◇◇◇

Typical usage scenarios of standard computer systems often involve only one human interacting with the computer at any time, and the system is only a small part of the real environment of the user. But exhibits are usually visited by groups of people, and when users interact with them, they are ready to immerse themselves into the world of the exhibit.

The CAVE installation in the Ars Electronica Center in Linz uses wall-size projections all around the visitors to immerse them into a virtual reality. Special glasses synchronize with these displays to create a three-dimensional impression.

Virtual Vienna uses a rear-projected display screen of about

CAVE: 3-D walls

Virtual Vienna: panorama

140 4 A Pattern Language for Interactive Music Exhibits

1.6 m width, with the users standing at the same distance to the screen. This fills most of the optical viewing field when looking at the screen, and helps people to feel like they are actually standing at the place whose panorama is being displayed.


Personal Orchestra: large projection

Personal Orchestra uses an even bigger display area of about 2.3 m width, again with a corresponding viewing distance. This conveys the impression of actually standing in front of the Vienna Philharmonic in a far better way than it would on a small computer monitor.

With these systems, this large display not only immerses the main user into the experience, it also allows several bystanders to at least observe the exhibit in action, which many may find already sufficient without becoming an active user.

Therefore:

Prefer a single exhibit with a large-scale display, with a minimum of 1.5 m in display width, over several similar stations with smaller displays, and over other output devices that shield a single user from his co-visitors, such as head-mounted displays. Design for a viewing distance that roughly equals the width of the display.



◇◇◇

If you hide the display technology, it can become a "magic image"—INVISIBLE HARDWARE (H14)... ..

## Borcher's patterns to interaction design > interactive music exhibits

4.2 HCI Pattern Language 139

**H13 IMMERSIVE DISPLAY \***




Figure 4.31: CAVE in the Ars Electronica Center Linz.

...you have decided to create an exhibit that several people can experience simultaneously—COOPERATIVE EXPERIENCE (H3). Now you need to find a way to design the visual output of such a system.

Typical usage scenarios of standard computer systems often involve only one human interacting with the computer at any time, and the system is only a small part of the real environment of the user. But exhibits are usually visited by groups of people, and when users interact with them, they are ready to immerse themselves into the world of the exhibit.

The CAVE installation in the Ars Electronica Center in Linz uses wall-size projections all around the visitors to immerse them into a virtual reality. Special glasses synchronize with these displays to create a three-dimensional impression.

Virtual Vienna uses a rear-projected display screen of about

CAVE: 3-D walls

Virtual Vienna: panorama

4 A Pattern Language for Interactive Music 140

**examples**

1.6 m width, with the users standing at the same distance to the screen. This fills most of the optical viewing field when looking at the screen, and helps people to feel like they are actually standing at the place whose panorama is being displayed.


*Personal Orchestra:* large projection

*Personal Orchestra* uses an even bigger display area of about 2.5 m width, again with a corresponding viewing distance. This conveys the impression of actually standing in front of the Vienna Philharmonic in a far better way than it would on a small computer monitor.

With these systems, this large display not only immerses the main user into the experience, it also allows bystanders to at least observe the exhibit in a way many may find already sufficient without becoming an active user.

Therefore:

**Prefer a single exhibit with a large-scale display, with a minimum of 1.5 m in display width, over several similar stations with smaller displays, and over other output devices that shield a single user from his co-visitors, such as head-mounted displays. Design for a viewing distance that roughly equals the width of the display.**



If you hide the display technology, it can become a "magic image"—INVISIBLE HARDWARE (H14). ...

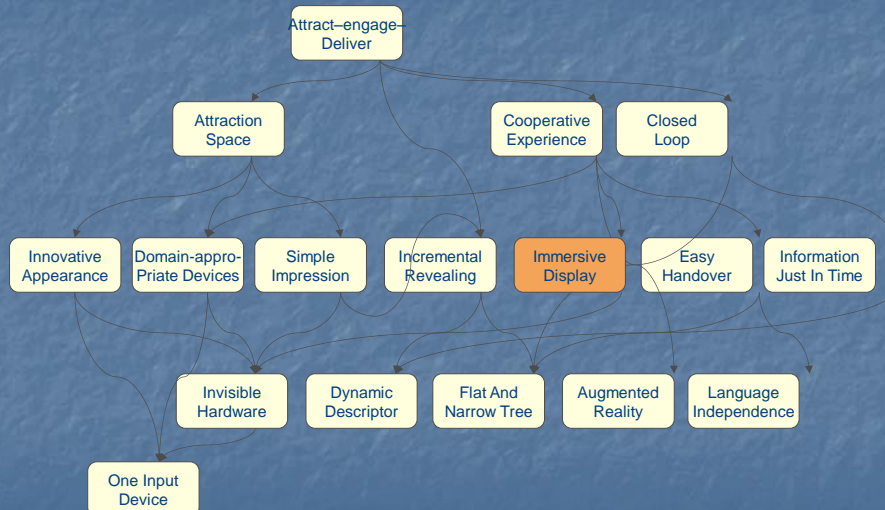
**solution**

**diagram**

**references**

[Borcher, 2001] Ashraf Gaffar, Daniel Sinnig 2003

## Borcher's patterns to interaction design > interactive music exhibits



[Borcher, 2001] p. 104 Ashraf Gaffar, Daniel Sinnig 2003

## Tidwell's Common ground > *Go back to a safe place*

**Contesto:** L'artefatto permette all'utente di muoversi attraverso spazi (come nel pattern *Navigable Spaces*), o passi (come in *Step-by-Step Instructions*), attraverso una *Narrativa* lineare, o stati discreti; l'artefatto ha anche uno o più checkpoint in quell'insieme di spazi.

**Problema:** Come può l'artefatto rendere la navigazione facile, comoda e psicologicamente sicura per l'utente?

[Tidwell, 1998] [http://www.mit.edu/~jtidwell/language/go\\_back\\_to\\_a\\_safe\\_place.html](http://www.mit.edu/~jtidwell/language/go_back_to_a_safe_place.html)

## Tidwell's Common ground > *Go back to a safe place*

**Esempi:** - il pulsante "Home" di un Web browser

- ritornare all'inizio di un capitolo in un libro fisico o una rivista
- la proprietà di "revert" di qualche applicazione software

**Forze:** - un utente che esplora un artefatto complesso o tenta molte operazioni di cambio stato, può letteralmente perdersi.

- Un utente potrebbe dimenticarsi dove si trova se smette di usare l'artefatto mentre era nel mezzo di un'attività e non vi torna per un po' di tempo.
- Se un utente capita in uno spazio o stato non desiderato egli vuole uscirne in maniera sicura e predicibile.
- Un utente sarà più incoraggiato a usare un artefatto se viene rassicurato sul fatto che può uscire facilmente da uno spazio o stato indesiderato.
- Fare backtracking in un lungo percorso di navigazione potrebbe essere tedioso.

[Tidwell, 1998] [http://www.mit.edu/~jtidwell/language/go\\_back\\_to\\_a\\_safe\\_place.html](http://www.mit.edu/~jtidwell/language/go_back_to_a_safe_place.html)

## Tidwell's Common ground > *Go back to a safe place*

**Soluzione:** Fornire un modo per tornare indietro a uno dei checkpoint che l'utente ha attraversato

- il checkpoint potrebbe essere una home page, un file o uno stato memorizzato, l'inizio logico di una sezione di narrativa o un insieme di passi. Idealmente potrebbe essere qualsiasi spazio o stato che l'utente sceglie di dichiarare come checkpoint.



**Contesto risultante:** Go Back One Step è un'aggiunta naturale a questo pattern, e spesso lo si trova insieme. Per uso non narrativo, la Interaction History è un altro pattern utile, quasi al punto di rendere non necessario il Go Back to a Safe Place: potrebbe aiutare realmente un utente che si è perso a capire dove si trova. O ricordare a un utente che ha interrotto l'interazione dove si trovava e cosa stava facendo.

[Tidwell, 1998] [http://www.mit.edu/~jtidwell/language/go\\_back\\_to\\_a\\_safe\\_place.html](http://www.mit.edu/~jtidwell/language/go_back_to_a_safe_place.html)

## I design pattern HCI (cont.)

- Caratteristiche dei pattern
  - colgono la prassi della progettazione non la teoria
  - colgono le proprietà essenziali comuni di un buon progetto
  - rappresentano la conoscenza della progettazione a vari livelli: sociale, organizzativo, concettuale e dettagliato
  - incorporano valori e possono esprimere ciò che è umano nel progetto dell'interfaccia
  - sono intuitivi e leggibili e quindi possono essere usati per la comunicazione tra tutti gli stakeholder
  - un linguaggio di pattern dovrebbe essere generativo e perciò utile nello sviluppo di design completi



# Riepilogo della Lezione

## Principi di usabilità

- il design ripetibile per l'usabilità cerca di ottenere il massimo vantaggio da un buon progetto astruendo le proprietà generali che possono determinare un design significativo
- Il successo della progettazione per l'usabilità richiede sia capacità d'introspezione creativa (nuovi paradigmi) sia una certa pratica basata su principi

## Utilizzo delle regole di design

- Standard, linee guida e design pattern indirizzano l'attività di progettazione

# Riferimenti

- Alan Dix, Janet Finlay, Gregory Abowd, Russell Beale, "Human-Computer Interaction", 3ª Edizione, McGraw-Hill, Cap. 7.

