

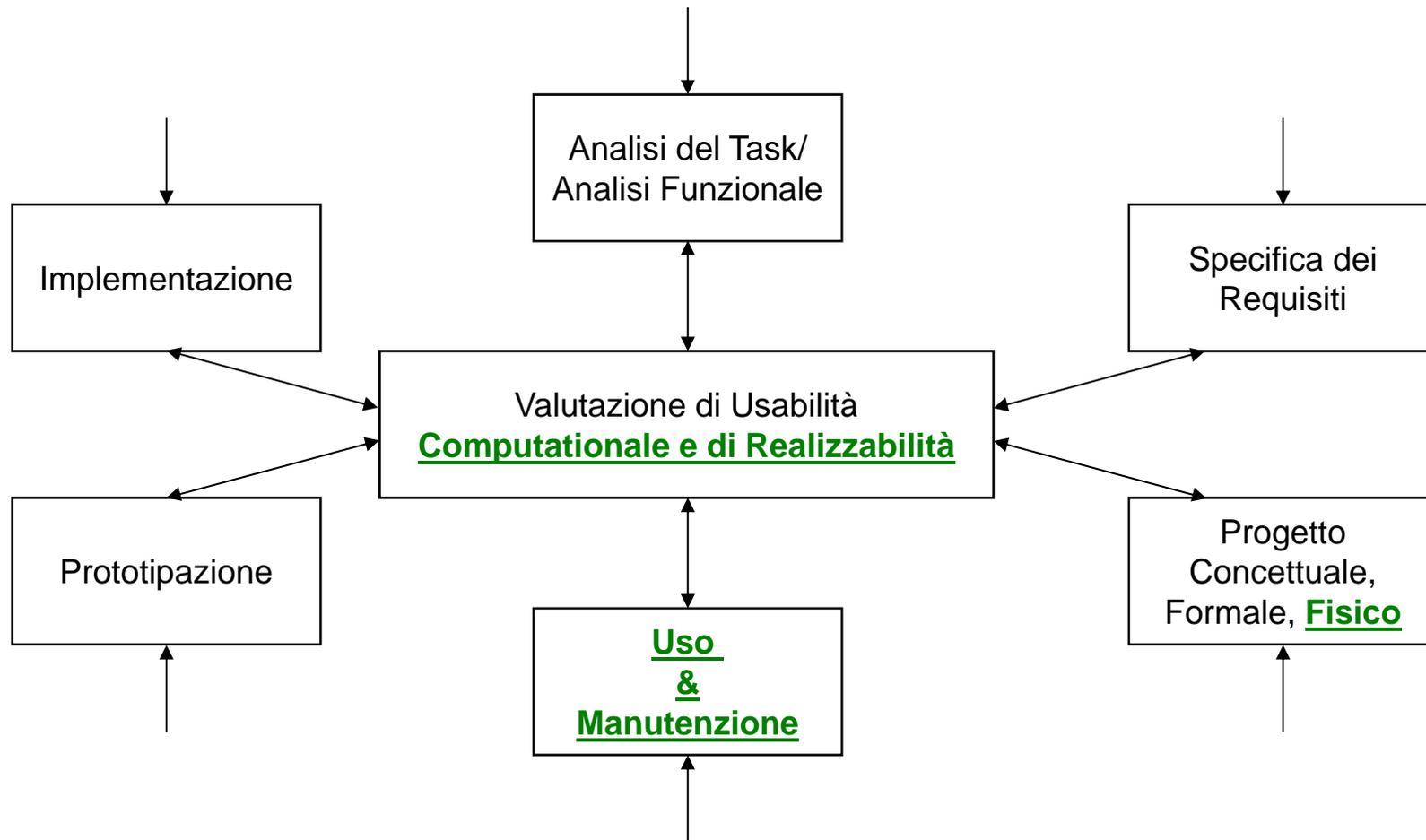
INTERAZIONE UOMO-MACCHINA

Ciclo di Vita a Stella
Progetto Concettuale, Formale, Fisico

Barbara Rita Barricelli
Stefano Valtolina

Il ciclo di vita a stella riadattato

2

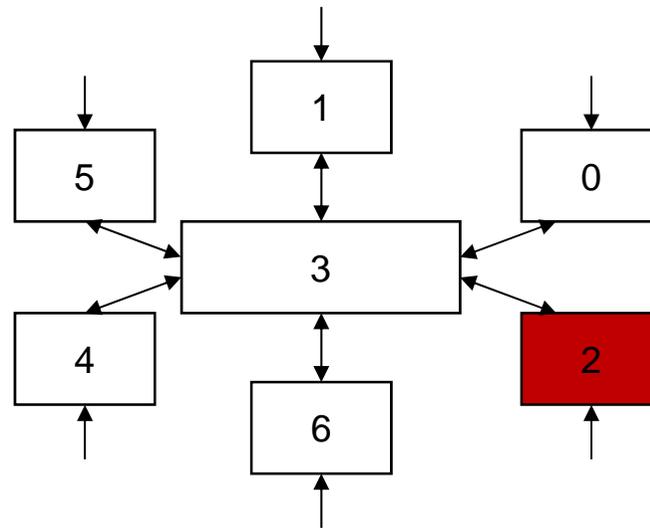


Adattato da Hartson&Hix [HH93] in [BBM99b]

Progetto Concettuale, Formale, Fisico

3

- Progettazione **parallela** (2) la vostra scelta non è mai abbastanza buona
- **Analisi competitiva** (2)



Progetto del sistema interattivo

4

- **Analisi/specifica dei requisiti:** ‘cosa, per chi, da chi’
- **Progetto:** ‘come’, sia negli aspetti computazionali che in quelli fisici e cognitivi
- **Input:** documenti di specifica dei requisiti, di descrizione dei compiti (inclusi profili degli utenti, prototipi, analisi dei sistemi concorrenti etc).
- **Output:** specifica del sistema: modello concettuale, formale, fisico. Chi deve implementare il sistema deve capire cosa deve fare. Chi dovrà usarlo dovrà capire come usarlo

Il progetto: 3 “fasi”

5

- **Progettazione concettuale:** riguarda la definizione di un modello concettuale che cattura quello che il prodotto farà e come lo farà
→ progetto di **metafora e stile di interazione**
- **Progettazione formale:** riguarda la specifica formale del dialogo fra utente e sistema → con **diverse notazioni** (es. statechart)
- **Progettazione fisica:** si occupa di definire i dettagli dell'interazione (menu, icone, layout schermate, etc)
- Il progetto definitivo emerge per iterazioni successive
- Naturalmente secondo il ciclo a stella – con **valutazioni e verifiche** ogniqualvolta necessario

Come si arriva al modello concettuale

6

- Solitamente **coinvolgendo** l'utente nel progetto (dipende dal ruolo degli utenti nel progetto, vedi dopo)
- Identificazione dei **bisogni e analisi dei compiti**:
raccogliere quante più informazioni possibili sugli utenti, il loro lavoro, e il contesto di quel lavoro
- **Specificazione dei requisiti** a partire dai bisogni e dai compiti identificati

Il progetto dell'interazione (interaction design)

7

- Creare **esperienze d'uso** che migliorino e potenzino il modo in cui le persone lavorano, comunicano, interagiscono
- **Interaction design** come “la progettazione di spazi per la comunicazione e l'interazione umana” Winograd (1997)
- “Every one is motivated by the needs of people and the desire to create habitable information systems” Reenskaug (2003)
- Differenza rispetto a **ingegneria del software** → occorre tener conto **dell'interazione con l'uomo**
 - un'analogia: la differenza fra **architetti** e **ingegneri civili**

Progetto del sistema interattivo: caratteristiche

8

- **Ruolo degli utenti:**
 - tradizionale
 - user-centered
 - partecipativo
- **Organizzazione:**
 - progetto in parallelo
 - progetto iterativo
- **Stile del progetto:**
 - strutturato (sequenziale)
 - olistico

Progetto di sistemi interattivi

9

- Progetto di sistemi interattivi → **interdisciplinare**
- **Almeno 3 discipline**
 - ▣ Ingegneria del software
 - ▣ Interazione Uomo-Macchina
 - ▣ Dominio applicativo
- Migliorare la **qualità del prodotto** attraverso il miglioramento della **qualità della comunicazione** fra gli operatori (soggetti coinvolti nel e dal progetto)

Come superare il gap comunicazionale

10

- **Coinvolgere l'utente nel progetto**
 - ▣ Non parlare solo con i manager e i “proxy-users” (persone che interpretano la parte degli utenti)
 - ▣ ... ma coinvolgere gli **utenti “veri”** per tutta la durata del progetto, in modo da tenere in considerazione i loro bisogni, scopi e attività
- **Obiettivo: usabilità, utilità e ... significatività**
(“meaningfulness”... pensare ai sistemi di intrattenimento)

Gradi di coinvolgimento: un continuum

11

- Dalla semplice **informazione sugli sviluppi** del progetto tramite strumenti vari di comunicazione
- Al **test finale** con gli utenti
- Alla **osservazione delle loro attività** (studi etnografici)
- Agli **approcci partecipativi** (e.g. tramite workshop)
- Al **metadesign**
 - ▣ Progettare sistemi che possano essere usati dagli utenti per **adattare i propri sistemi o crearne di nuovi**
 - ▣ Utenti coinvolti nella progettazione **“a tempo di uso”**
 - ▣ Riconoscere che l'utente evolve da **consumatore a produttore** di conoscenza

User-centered design

12

- Approccio al progetto centrato sull'utente
 - ▣ Ci si focalizza fin dall'inizio sulle **caratteristiche degli utenti e sui loro compiti**
 - ▣ Vengono consultati **esperti del dominio** applicativo
 - ▣ Vengono studiati **utenti futuri/probabili**
 - ▣ **Test con gli utenti**
 - ▣ **Design iterativo**
- Un problema: “**one-way communication**”
 - I progettisti estraggono informazioni dagli utenti con interviste, questionari, osservazione del loro lavoro (attenzione: la conoscenza è spesso tacita), esperimenti

Participatory design

13

- Approccio **partecipativo** al progetto
 - “... the ultimate users of the software make effective contributions that reflect their own perspectives and needs, somewhere in the design and development lifecycle of the software” [Muller et al. 1997]
- **Rappresentanti degli utenti** vengono coinvolti nel progetto (detti esperti di dominio) e divengono membri della **squadra di progetto**
- Gli scenari di attività, i prodotti e gli strumenti in opera vengono **osservati, elaborati e raffinati dalla squadra di progetto**, ogni membro guarda il progetto da un punto di vista (stakeholder)
- “**Two-way communication**” tra tutti gli interessati

Due problemi comuni a chi sviluppa sistemi con approcci partecipativi

14

- Comunicazione fra i membri della squadra di progetto
(**simmetria dell'ignoranza**)
- **Documentazione** per avere memoria del progetto

Comunicazione

15

- Fra i membri della squadra di progetto:
 - Ingegneri del software
 - Esperti di interazione uomo-macchina
 - Utenti
- Tipicamente gli utenti non capiscono il **gergo tecnico** usato dagli ingegneri del software
- Gli ingegneri del software hanno una **vaga conoscenza** di concetti, metodi, terminologia usata dagli utenti
- Occorre un **metodo per semplificare la comprensione supportando la comunicazione** (es. scenari, mockup)

Documentazione

16

- Si vuole creare una “**memoria**” per:
 - permettere il **riuso degli strumenti** (o parte degli strumenti prodotti)
 - evitare di **ripetere gli stessi errori**
 - introdurre **nuovi membri nel team di progetto**
 - **supportare gli utenti finali** nell’uso del sistema
 - **istruire nuove persone** del dominio applicativo ad usare i sistemi sviluppati

Towards Producersage

17

- **Futures for User-Led Content Production¹**
- Bruns Axel, *Blogs, Wikipedia, Second Life, and Beyond – From Production to Producersage*, Peter Lang (June 2, 2008)

¹Dr Axel Bruns - Creative Industries Faculty, Queensland University of Technology:
Towards Producersage: Futures for User-Led Content Production

User-Led Content Production

18

- Problema emergente in diversi contesti e domini
 - open source software development
 - online publishing:
 - blogs
 - open news – e.g. [Slashdot](#), [Indymedia](#), [OhmyNews](#)
 - knowledge management
 - wikis – e.g. [Wikipedia](#)
 - [Google Earth](#)
 - multi-user gaming:
 - e.g. [The Sims](#), [Everquest](#), [Second Life](#), [Spore](#)
 - creative practice
 - e.g. [Flickr](#), [ccMixter](#), [YouTube](#), [Jumpcut](#)
 - viral marketing

Caratteristiche comuni

19

- Condividere attraverso l'ambiente interattivo
 - ▣ Gli utenti **producono contenuti** – creano nuovi contenuti e li rendono direttamente **disponibili** ad altri
 - ▣ Il **coinvolgimento** è **collaborativo** – gli utenti **insieme** creano contenuti
 - ▣ Gli **artefatti digitali** non sono “**incompleti**” – sono costantemente **aggiornati** con minori e maggiori revisioni
 - ▣ Questo sviluppo di contenuti deve seguire nuove forme di **licenze per il copyright**

Cosa sta accadendo?

20

- Bisogno di:
 - prosumer (Alvin Toffler)?
 - the citizen-consumer (John Hartley)?
 - pro-am production (Charles Leadbeater & Paul Miller)?
 - customer-made products, produced by a new Generation C (Trendwatching.com)?
 - corporations harnessing the hive (J.C. Herz)?

Proodusage

21

- Oltre la **produzione**:
 - ▣ Ognuno può **editare** – l'utente diventa produttore di contenuti
 - ▣ L'uso e la produzione sono **incrementali, inestricabili e interlacciati**
 - ▣ Non esiste una stretta distinzione tra **produttore, distributore e consumatore**

Questo è il **proodusage**

Caratteristiche del Prodsusage (1)

22

□ Utenti

□ sono **produttori** :

- i progetti sono **guidati** degli utenti, o
- è necessario un loro **contributo**

□ Implica una **collaborazione** :

- con altri **utenti**, o
- con **partner istituzionali**;
- con **ruoli flessibili e intensità variabile**
- in comunità organizzate in **autonomia** e spesso **dinamiche ed eterogenee**

Caratteristiche del Prodosage (2)

23

□ **Contenuti digitali**

□ Sono sempre **“incompleti”**:

- **creazione e sviluppo** iterativa, evolutiva e guida a:
- **revisioni costanti** al fine di
- forzare differenti **direzioni di sviluppo**

□ Sono **prodotti** aventi alla base **licenze di copyright alternative**:

- **Paternità collaborative** di:
- **contributori volontari** che lavorano con **sviluppatori professionisti**
- in un rapporto in cui i singoli **contributi sono riconosciuti**

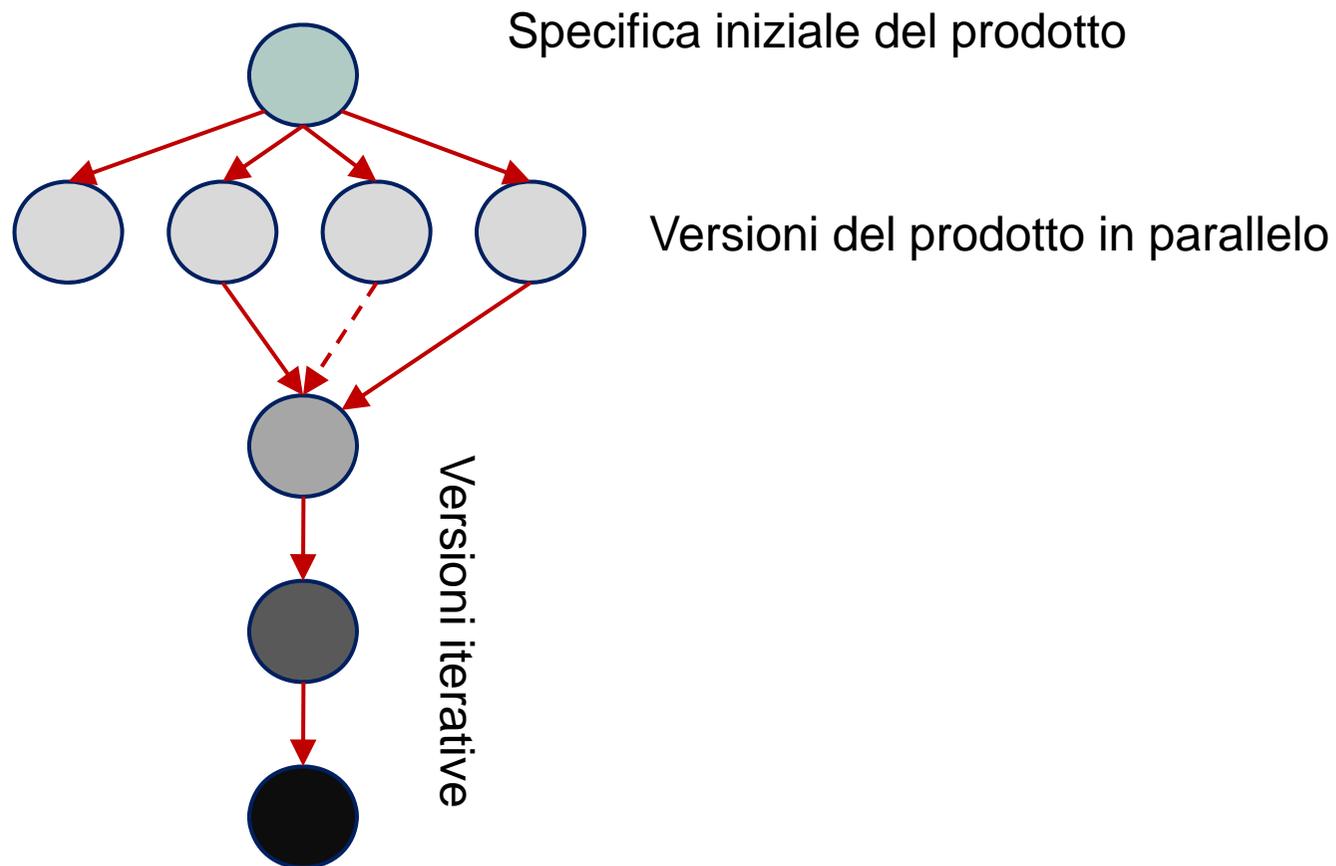
Implicazioni future

24

- **Nuovi paradigmi:**
 - ▣ Il **produsage** sta prendendo piede in ambito Web2.0, social software, ambienti collaborativi
- **Education:**
 - ▣ Necessità di nuovi **approcci** per l'insegnamento
 - ▣ focus su capacità di insegnamento **collaborativo, creativo, critico e comunicativo**
- **Società:**
 - ▣ industriale → informativa → collegata → produsage ?
 - ▣ può trasformare i cittadini da **consumatori** a **produttori** di democrazia?

Progetto in parallelo e/o iterativo (1)

25



Progetto in parallelo e/o iterativo (2)

26

- Utilità di esplorare differenti **alternative di progetto**
- **Progettazione in parallelo:** più (squadre di) progettisti affrontano il problema
 - ▣ Le squadre di progettisti **lavorano indipendentemente**
 - ▣ Se i progetti differiscono molto, è conveniente **approfondirli e ridurre i prototipi a confronto**
 - ▣ **Valutazioni di usabilità e fusione** in un progetto
 - ▣ Importante per **sistemi nuovi**
 - ▣ Se esistono più **prodotti competitivi**, analisi competitiva può funzionare da progetto in parallelo
 - ▣ E' una maniera economica di **esplorare lo spazio di progetto** con più progetti sviluppati contemporaneamente (sei mesi di ritardo possono ridurre i profitti di un terzo)

Progetto: sequenziale od olistico

27

- Entrambi danno importanza alla possibilità di **proporre e confrontare diverse alternative**
- Altre tecniche privilegiano gli aspetti **informatici-strumentali**
- **Strutturato (Sequenziale)**: separa il progetto del modello concettuale dal progetto formale e fisico.
 - ▣ **Vantaggio**: il progetto concettuale non ‘fissa’ la soluzione e mantiene aperto lo spazio delle alternative
- **Olistico**: concepisce il progetto come un processo unitario in cui la progettazione degli **aspetti fisici e strumentali** avviene di pari passo con la **progettazione del modello concettuale e formale**
 - ▣ **Vantaggio**: creare il modello concettuale in maniera che sia **comprensibile** (e discutibile) immediatamente dall’utente: il modello concettuale è trasmesso da come **l’interfaccia si manifesta fisicamente** (Norman)
 - ▣ Per mantenere **aperte le alternative**: sviluppare ‘**immagini**’, usare **mock-up** espressivi a basso costo

La progettazione fisica

28

- Definizione **concreta** di questioni dettagliate di progettazione dell'interfaccia, come fare il **design delle schermate**, quali icone, come strutturare i menu
- **Elementi di interazione** (widget): finestre, menu, icone, barre degli strumenti ... ognuno disegnato espressamente oppure scelto da una **libreria**
- Le scelte possono essere fatte secondo **guide di stile** (commerciali o aziendali) che forniscono dettami per il **look&feel dell'interfaccia**, ovvero quali widget devono essere utilizzati, a quale scopo e che aspetto devono avere

Progettare il *Layout*

29

- Due aspetti:
 - ▣ Il modo in cui il **compito viene suddiviso in diverse schermate**
 - ▣ Il modo in cui le **single schermate sono presentate**
- Il primo aspetto si può avvalere della fase di **analisi del task** che ha suddiviso il compito in sotto-compiti (una schermata per ogni sotto-task)
- **Schermate semplici**, in cui sono richieste poche scelte ed inserimenti di dati
- Attenzione che tutta **l'informazione pertinente sia disponibile al momento opportuno**
- Progettare schermate in cui **l'attenzione dell'utente** sia immediatamente guidata verso i contenuti principali (anche tramite colori, riquadri, etc.)

Progettare Menù

30

- Forniscono all'utente una scelta fra **comandi diversi** o fra diverse opzioni di uno stesso comando
- Si basano sulla **struttura dei compiti** e sulle **informazioni necessarie a compierli**: la struttura del menù deve riflettere la sequenza di azioni che l'utente esegue più di frequente
- Alcuni punti importanti:
 - ▣ nei menu a tendina e in quelli pop-up le opzioni di uso più frequente devono **essere posizionate in alto**
 - ▣ **usare raggruppamenti** (logici, sulla base degli obiettivi e dei compiti dell'utente)
 - ▣ nomi delle voci: **brevi, chiare e non ambigue**
 - ▣ attenzione allo spazio per elencare le voci che **non deve indurre l'utente a fare scelte sbagliate**

Nuove strategie di design: l'emozione

31

- Il concetto di **emozione**:
 - ▣ Stimolo (attenzione) → percezione → elaborazione (conoscenza – memoria)
 - ▣ Una **risposta** dell'organismo a determinate caratteristiche **percepite** come **stimoli** dell'ambiente, **modulata** da circuiti dedicati e che **genera** il cosiddetto **stato d'animo** (sentimento)
 - ▣ **Umore**: prolungamento contemporaneo dell'emozione e dello stato d'animo
- In questo corso: **emozione = stato d'animo** e lo valuteremo in relazione **all'interazione tra uomo e macchina**

Human-computer co-evolution

32

- “Using the system changes the users, and as they change they will use the system in new ways”



Nielsen 1993

- I sistemi interattivi sono più che monolitici **pezzi di software**
- I sistemi interattivi devono essere progettati per **evolvere** in base ai **nascenti bisogni dell'utente**
- **Idea:** permettere all'utente di personalizzare ed far evolvere il sistema a **run-time** → **End User Development**

End-User Development

33

- “End User Development is a set of **activities** or **techniques** that allow users of software systems, who are acting as **non-professional software developers**, at some point to create or modify a software artefact ”

[EUD-Net 2003]

- EUD significa
 - ▣ Attiva **partecipazione** dell'utente nel **processo di sviluppo software**
 - ▣ Possibilità da parte dell'utente di **modificare strutture del sistema**
- Nuove challenges per il **design** e **implementazione** dei sistemi

End-User Developments activities

34

- **NO:** attività che permettono all'utente di scegliere tra **alternative di comportamenti** già presenti nel sistema (sia a livello di presentazione che di strutture di interazione)
 - ▣ Es. inserimenti di **parametri**
- **YES:** attività che implicano l'uso di **paradigmi di programmazione** che permettono di **creare e modificare** strutture del sistema

Tecniche di EUD

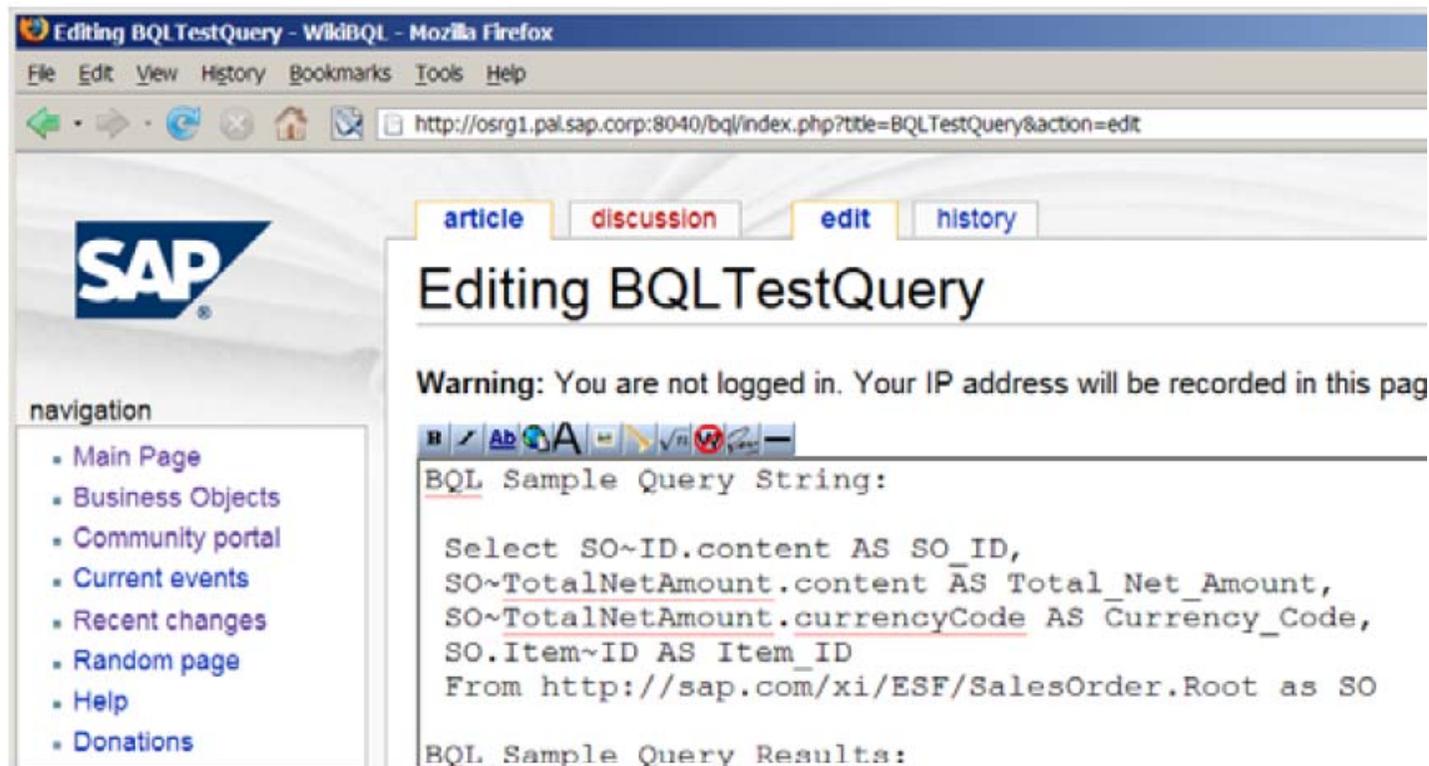
35

- Database Query Languages
- Visual Programming
- Spreadsheet Programming
- Programming By Example
- Form-based fill-in

Database Query Languages

36

- Esempio: Wiki Business Query
 - ▣ È possibile inserire **codice di query** nelle **pagine web** ed eseguirlo



The screenshot shows a Mozilla Firefox browser window with the address bar displaying `http://osrg1.pa.sap.corp:8040/bql/index.php?title=BQLTestQuery&action=edit`. The page content includes the SAP logo, a navigation menu with links like 'Main Page', 'Business Objects', and 'Community portal', and a main editing area. The main area has tabs for 'article', 'discussion', 'edit', and 'history'. The title of the page is 'Editing BQLTestQuery'. A warning message states: 'Warning: You are not logged in. Your IP address will be recorded in this pag'. Below the warning is a text area containing a BQL query string:

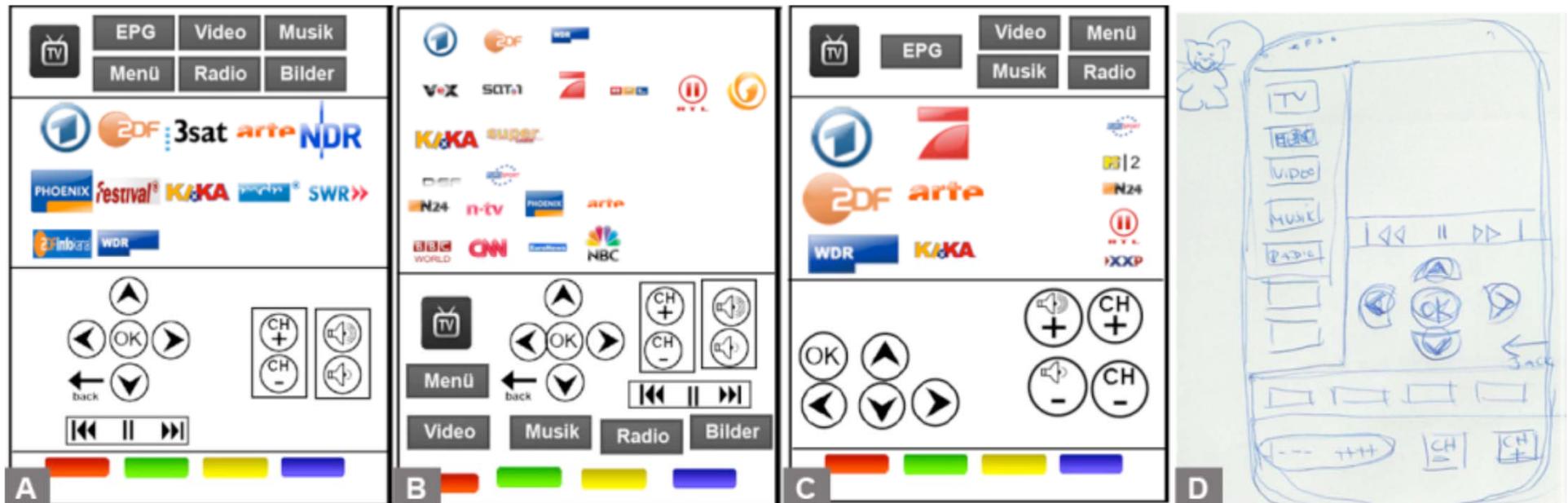
```
BQL Sample Query String:  
  
Select SO~ID.content AS SO_ID,  
SO~TotalNetAmount.content AS Total_Net_Amount,  
SO~TotalNetAmount.currencyCode AS Currency_Code,  
SO.Item~ID AS Item_ID  
From http://sap.com/xi/ESF/SalesOrder.Root as SO
```

Below the query string, the text 'BQL Sample Query Results:' is visible.

Visual Programming Systems (1)

37

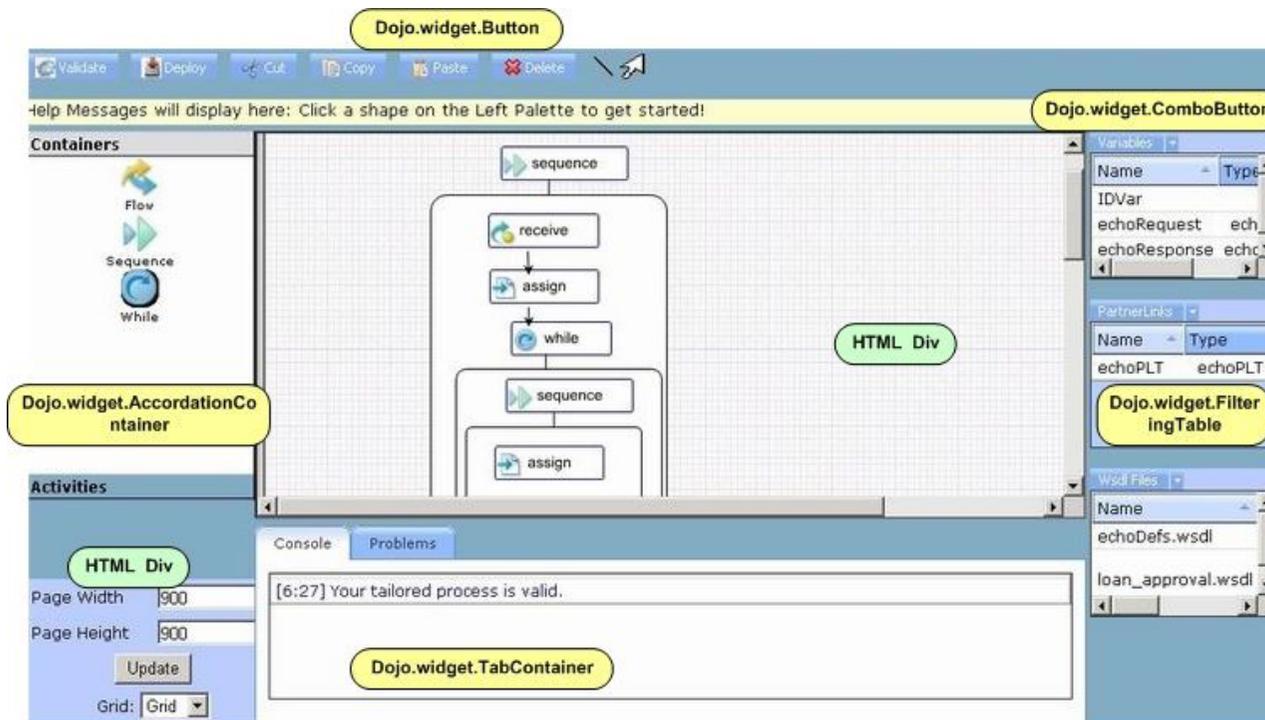
- Esempio: Paper-Based EUD
 - **Progettazione** basata sullo uso di una penna: Anoto
 - pRemote: differenti templates usati in Anoto paper e selezionati per **innescare delle funzioni**



Visual Programming Systems (2)

38

- Esempio: TailorBPEL
 - Usato a supporto di **processi di business** (linguaggio BPEL)
 - È possibile eseguire **processi a run-time**
 - Tailor: “remotely ... in an easy graphical way”



Visual Programming Systems (3)

39

- Monaco IDE supporta una **programmazione visuale e testuale** basata su Eclipse
 - Flowcharts, timing diagrams

Spreadsheet Systems

40

□ “Function Sheet”

- Aggiunge **funzioni a fogli di calcolo** usando il foglio stesso per la **definire** le funzioni
- Supporta la **ricorsione**

The screenshot shows a spreadsheet window titled "Spreadsheet". On the left, a sidebar shows a tree view with "Workbook" containing "Main" and "Functions" containing "TriArea". The main area displays a function definition in a yellow cell: `TriArea(a(B1), b(B2), c(B3))=85`. Below it, a formula bar shows the formula: `=(B4*(B4-B1)*(B4-B2)*(B4-B3))^0.5`. The spreadsheet grid has columns A through E and rows 1 through 5. Column B is highlighted in blue. Row 5 is highlighted in blue. The cell at the intersection of row 5 and column B contains the value 600.

| | A | B | C | D | E |
|---|------|-----|---|----|--------|
| 1 | a | 30 | | B1 | a |
| 2 | b | 40 | | B2 | b |
| 3 | c | 50 | | B3 | c |
| 4 | s | 60 | | B5 | Result |
| 5 | area | 600 | | | |

Programming By Example

41

- CTM (Collaborative Task Manager)
 - Crea task eseguendo **processi poco strutturati**

Form-Based Fill-in

42

- Toped
 - Definisce strutture usando delle **opzioni**

The screenshot displays a configuration interface for defining email address parts. It consists of two main sections, each with a title bar and a close button (X).

Section 1: Each Email Address has a part called the

- Rule 1:** The username has of the following characters:
 - lowercase letters
 - uppercase letters
 - digits
 - other characters:
- Rule 2:** The username has of the following characters:
 - lowercase letters
 - uppercase letters
 - digits
 - other characters:
- Rule 3:** The username is preceded by and followed by
(You can leave one of these two fields blank if it does not apply.)

+ info

+ part

Section 2: Each Email Address has a part called the

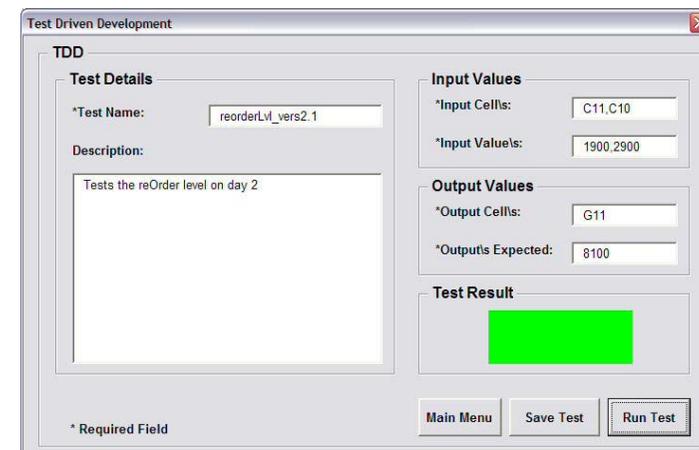
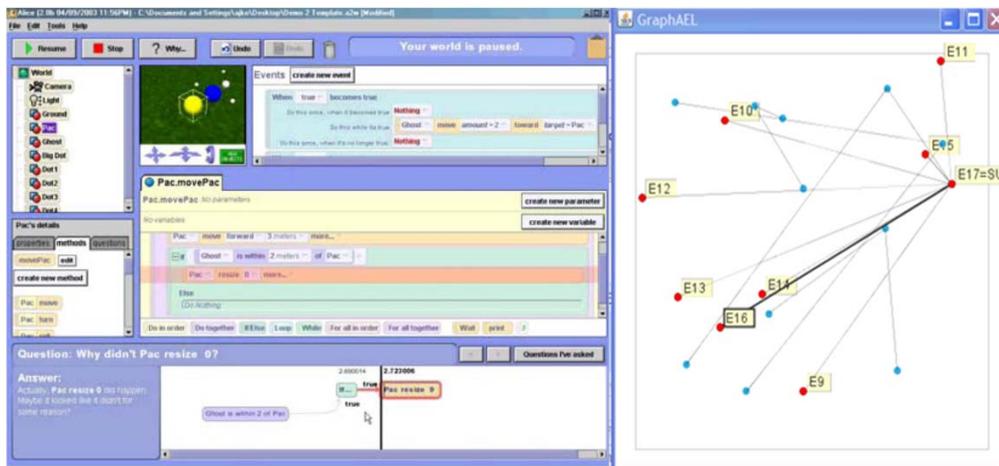
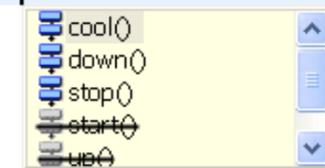
- Rule 1:** The hostname has of the following characters:
 - lowercase letters
 - uppercase letters
 - digits
 - other characters:
- Rule 2:** The hostname is preceded by and followed by

Come progettare per l'EUD

43

- Usare **paradigmi famigliari**
- Aiutare la programmazione fornendo solo **elementi “legali”**
- **Rendere visibile** tutto ciò che si programma
- **Testare la consistenza** in modo automatico
- Supportare la **collaborazione**
- **Testare** tutto ciò che si programma

```
driller.start();  
driller.down();  
driller.up();  
driller.|
```



Meta-design

44

- “Meta-design characterizes objectives, techniques, and processes for creating new media and environments allowing ‘owners of problems’ (that is, end users) to act as designers” [Fischer et al. 2004, CACM]
- “Meta-design represents an issue of how to construct socio-technical systems that allow users to cope with the emergent aspects of reality by enabling them, when needed and desired, to act as designers and be creative” [Giaccardi and Fischer, 2005]
- Il nostro punto di vista:
 - ▣ Meta-design è una tecnica che permette ad ogni **stakeholder** del team di design, usando **linguaggi e tool appropriati**, di **collaborare** nello sviluppo di sistemi software

Conseguenze del meta-design

45

- **Progettazione ed evoluzione perpetua** lungo tutto il ciclo di vita del software
- Permette all'utente di agire come **progettista** ed “**essere creativo**”:
 - **Personalizzando** il sistema per la propria comunità di utenti
 - **Confezionando su misura** il sistema in base ai propri bisogni/obiettivi

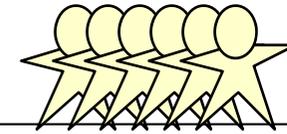
Ridistribuzione delle fasi di design

46

□ traditional design

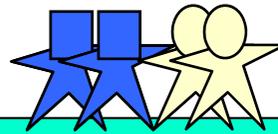


design by software professionals

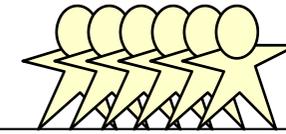


use by end users

□ participatory design



design by software professionals with end users

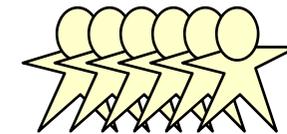


use by end users

□ meta-design & EUD



design by software professionals with end users



design by end users
use by end users