Piero Mussio
DICO- Università di Milano
mussio@dico.unimi.it

Progetto e specifica di sistemi software interattivi

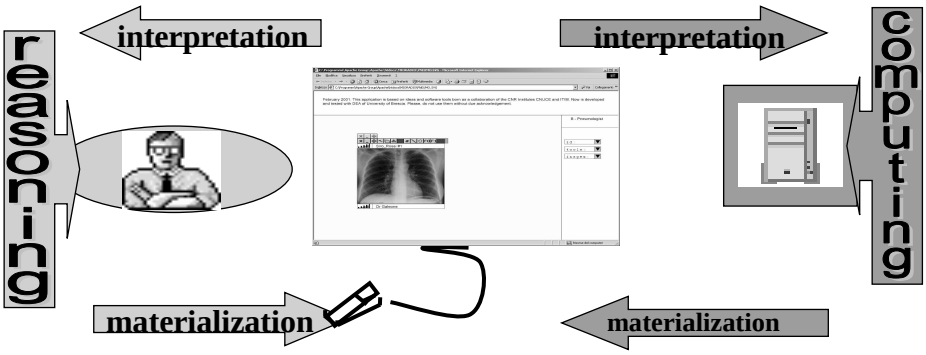
1

Outline

- Il modello a doppio ciclo
- Fenomeni caratterizzanti ium
- Controllare gli effetti dei fenomeni
- Meta-design
- L'approccio delle officine informatiche

2

HCI: a syndetic process



L'utente: il problem–owner nel dominio applicativo

Stakeholder ma non unico

La macchina: un'entità virtuale

[**PCL interaction model** Bottoni et al.95]

3

Virtual entities

- **Virtual entity (ve):** a system which exists as the result of the interpretation of a program by a computer

ve specification as a dynamic system

ve = <IF, CFSM, OF > on AZ

ve specification in terms of programs

Input events

P_{ve} = <In, Ap, Out > on Ev

The program mapping input events into elements in Az

The program acting as a digital communicant

The programs materializing the results of Ap

4

L'utente mentre svolge un'attività

…mira ad ottenere informazioni (dati e strumenti)

quando ne hanno bisogno

dove ne hanno bisogno

nel formato in cui ne hanno bisogno

al *livello di qualità* di cui hanno bisogno

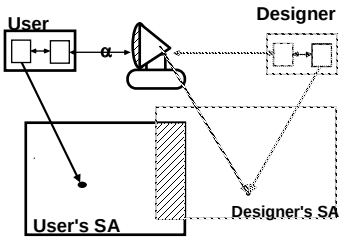
(affidabilità, validità nel tempo, astrazione, sintesi)

Alcuni fenomeni rendono complesso
il progetto e l'implementazione
di sistemi che soddisfino
queste necessità

5

1º fenomeno: il gap comunicazionale

Progettisti e utenti hanno differenti culture, conoscenze, tecniche di ragionamento e modelli mentali.



	utenti	Progettisti
Ragionamento e memorizzazione	•Euristico •Esempi & analogie •Concretezza	•Algoritmico •Deduzione astratta •Astrazione
documentazione	Notazioni tradizionali di dominio	Notazioni informatiche
Diverse capacità di articolazione	Tipiche del dominio;con strumenti tradizionali	Per l'uso di sistemi di calcolo

6

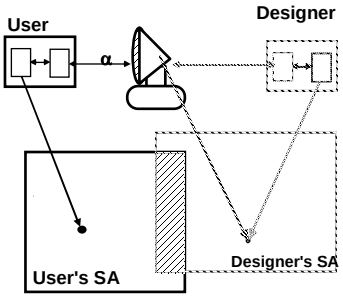
Il gap comunicazionale

- Per usare le macchine gli utenti sono costretti a esprimere i loro problemi in ‘calcolatorico ’ ed ad agire come il progettista pensa debbano agire
- Gli utenti sono esperti di dominio e ‘posseggono ‘ il problema (problem owner)
- Gli utenti hanno propri punti di vista sul dominio, sul contesto esull’attività e li descrivono nei propri linguaggi (sono stakeholder)

7

1º fenomeno: il gap comunicazionale

Conseguenze per il progettista :



-Conoscere l’utente

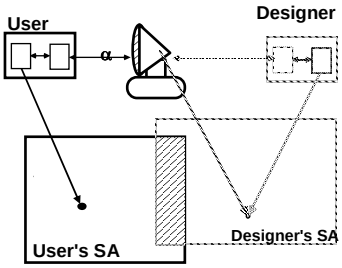
-Progetto del modello *concettuale*

-*Valutazione* del modello concettuale

8

2º fenomeno:
Varietà degli utenti

Gli utenti non sono una popolazione uniforme ma costituiscono comunità caratterizzate da diverse culture, scopi e compiti.



Le comunità si formano per ragioni:

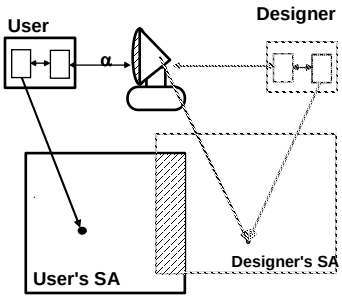
- linguistico culturali (skills, cultura)
- (dis)-abilità specifiche (fisiche,cognitive)
- dispersione geografica (↔ cultura di base)
- organizzative: compiti e contesti
- etc.

Ogni comunità
sviluppa propri dialetti, notazioni, conoscenza e abilità.

9

2º : varietà degli utenti

Conseguenze per il progettista :



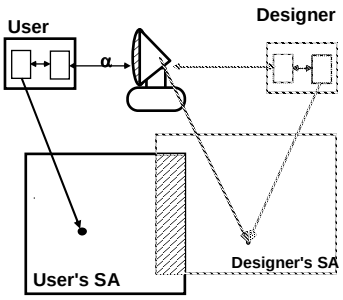
I sistemi debbono essere sviluppati

- con tecniche informatiche modulari (componenti)
- debbono essere *adattabili* alla comunità
- debbono *personalizzabili* e *adattivi*

10

3º fenomeno:
L'influenza nascosta delle tecnologie

Il Progettista costruisce sfruttando sistemi hardware e software (il microprocessore, il sistema operativo, interpreti e compilatori)



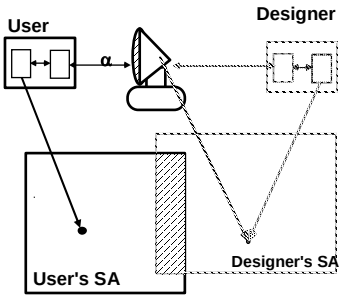
Ogni strumento favorisce lo svolgimento di certe attività a scapito di certe altre.

Se il modello concettuale non chiarisce dettagliatamente lo stile di interazione, spesso si propone uno stile che rende difficile l'attività dell'utente

11

3º fenomeno: influenza nascosta

Conseguenze per il progettista :



Il modello concettuale deve precisare ogni aspetto dell'interazione.

Nello sviluppare (codifica e interazione) occorre valutare ogni strumento sviluppato rispetto all'attività dell'utente (ciclo di vita a stella)

12

4º fenomeno :la co-evoluzione

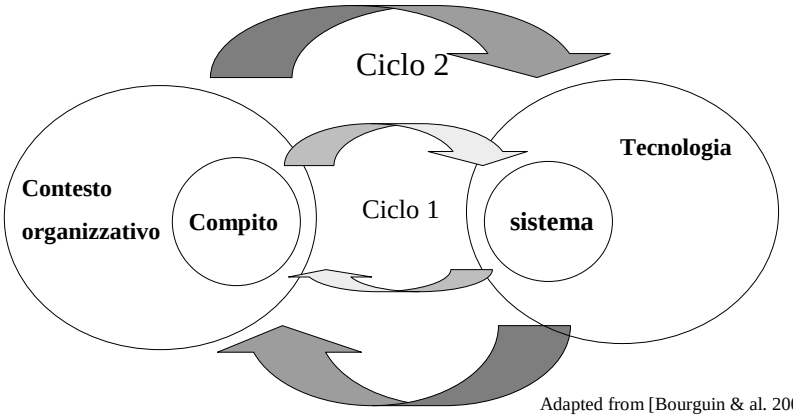
• Uno strumento non viene semplicemente aggiunto a quelli usati nello svolgimento di un'attività umana, ma la costringe ad evolversi. [Tikhomirov 1981]

• usare il sistema cambia i suoi **utenti**, e come cambiano, gli utenti useranno il sistema in nuove maniere *Nielsen, Usability Engineering*

Questi nuovi usi del sistema determinano un'evoluzione dell'ambiente e spingono i progettisti ad adattare il sistema e la tecnologia all'utente ed all'utente che è evoluto.

13

Un fenomeno interessante:la co-evoluzione



Adapted from [Bourguin & al. 2001]

Non necessariamente cicli virtuosi: diversi fenomeni possono inquinarli

14

Che fare?

Imparare a gestire la co-evoluzione studiando i fenomeni che la caratterizzano, individuando le richieste e i valori da preservare in modo da determinare uno sviluppo virtuoso delle tecnologie e dell'organizzazione

15

Conseguenze per il progettista :

- Il sistema progettato evolve nel tempo
- Una volta prodotto non finisce l'attività del progettista (P)
- P è il responsabile dell'evoluzione del sistema
- P deve progettare il sistema perché possa evolvere – sia adattabile alle nuove esigenze dell'utente (tecniche informatiche)

16

5º fenomeno :conoscenza tacita & informazione implicita

Gli utenti operano come competenti esperti del dominio in quanto sfruttano anche conoscenza pratica, generalmente tacita.

I documenti espressi nella notazione degli utenti e gli strumenti usati, portano informazione implicita

L'informazione implicita è elicitabile (ma in generale non resa esplicita) solo da chi possiede conoscenza tacita ed in proporzione alla conoscenza posseduta.

17

Conseguenze

- I messaggi scambiati e gli strumenti virtuali resi disponibili debbono permettere agli utenti di sfruttare la conoscenza tacita posseduta
- Uso di notazioni che evolvano quelle tradizionali dell'utenza
- P deve progettare un sistema che sia adattabile alla comunità degli utenti e che il singolo utente possa specializzarlo e personalizzarlo alle proprie necessità e a quelle del compito eseguito nel contesto.

18

To overcome the pathologies (1/2)

Recognize

- **users**
  - as ***experts***!!! in their domain, not in CS
  - as co-authors
  - as a non-homogenous community
- **interactive machine**
  - as a virtual, interactive, evolutive, dynamic system
- **<user- machine>**
  - as a syndetic co-evolutive system
- **interactive machine design**
  - as multidisciplinary open-ended activity

19

To overcome the pathologies (2/2)

..and develop

Design environments that:

- allow users to interact through their visual notations (formalized as necessary) and with tools familiar to them (widget resembling their tools in shape, dynamic and interaction behaviour)
- make abstract Computer Science concepts concrete to users and allow users to follow their learning and reasoning strategies (e.g. programming by example)
- support user-system co-evolution

20

Controllare e sfruttare i fenomeni..

…bilanciando gli effetti richiede competenze interdisciplinari

1.Gap:	HCI per individuare & SE per rimediare
2.Variabilità dell'utenza:	HCI & SE sfruttano la labilità.
3.Influenze nascoste :	HCI per individuare & SE per rimediare
4.Co-evoluzione:	HCI & SE per customizzare; Utenti per adattare
5.Conoscenza tacita e informazione implicita:	Utenti per individuare la notazione adatta; tutti per implementarla

21

Controllare e sfruttare i fenomeni.

- I diversi esperti hanno diversi punti di vista su uno stesso processo e sui problemi che gli utenti incontrano (sono stakeholders)
- I diversi esperti usano diversi linguaggi e notazioni
- Sfruttare la labilità del segno virtuale per presentare gli stessi contenuti secondo i diversi punti di vista e le diverse notazioni

22

Controllare e sfruttare i fenomeni.

L'utente come esperto:

indispensabile nell'individuare i problemi di usabilità del sistema

necessario come l'informazione implicita può essere incorporata nel sistema

inaffidabile nel suggerire soluzioni di usabilità ed ingegneria del software

23

Ruolo dell'utente nello sviluppo del software

Metodi

User-centered design: fornisce informazioni sui requisiti, su come si svolgono i compiti, sui casi d'uso

Participatory design: l'utente è membro attivo del gruppo di progetto

Meta-design: l'utente diventa progettista dei suoi tools e agisce come un progettista

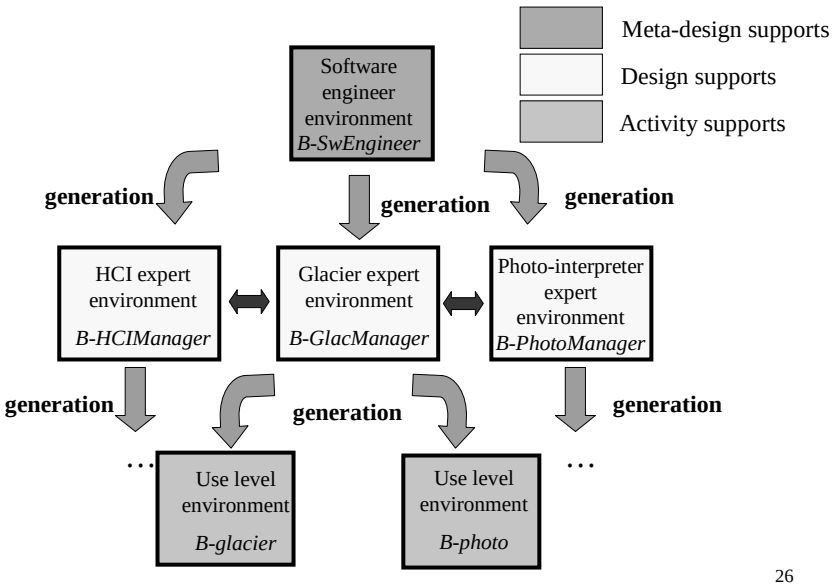
24

Meta-design

- "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]
- Meta-design is a technique which provides all stakeholders with suitable languages and tools to support their reasoning about the development of user work systems. [Mussio et al. 2004]

25

A meta-design hierarchy in an environmental case



26

Software Shaping Workshops (1/2)

- SSWs are software environments in which the users find all and only the tools required to perform their specific tasks, in a representation and exhibiting behaviors that are familiar to them. [Costabile et alt. 2002]
- Users work in analogy to artisans, using *virtual tools* that resemble real ones. **[Fresta et alt. 2000]**
- A SSW gives its users: a view on problem, a context; a set of constraints; a set of resources

27

Software Shaping Workshops (2/2)

- *Application workshop* is a SSW used by a community of domain-experts to perform its daily tasks, properly designed for its specific needs
- *System workshop* is a SSW used to generate and update other workshops, it is designed for the specific needs of a designers' community in a context, e.g. software engineers, HCI experts, experts of a specific domain designing a SSW for that domain
- *Use level and design level:* one network of SSWs

28

The case study

- Domain: mechanical engineering
  ETA Consulting, a company producing systems for factory automation

- ETA requests
  - Creating automation systems that are usable for and tailorable by ETA clients
  - Having software tools that supports ETA personnel and ETA clients (*d_experts*) in design, development, test, and use of the automation systems

- The involved stakeholders (users' community representatives)
  1. *professional software developers*
  2. ETA *d-experts*: electronic &mechanical engineers
  3. HCI *experts*
  4. Client *d-experts* (owners of the problems): testers, assembly-line operators,…

29

Need of environments for different d-experts

- Different d-experts collaborate to get a common goal

- They use same data but interpret them and materialize them in different ways

- Common case: team composed by members of different sub-communities (and companies), each with different expertise

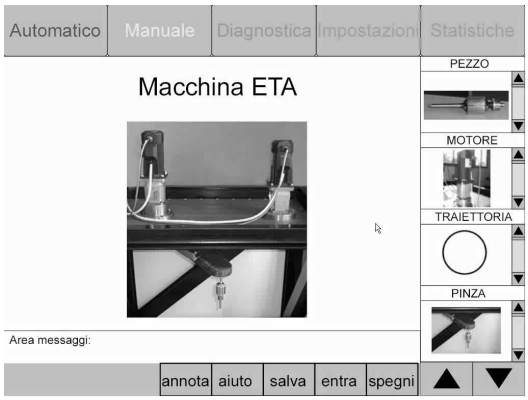
- Each sub-community needs a software environment (SSW) suitable to manage its own view of the activities to be performed

30

Application workshop for robot control: SSW for assembly-line operator

The operator equips the robot, simulates behavior, controls operations

- ✓using his/her language
- ✓organizing data according to his/her culture
- ✓using images expressive to him/her
- ✓permitting actions that resemble those performed in the traditional workplace
- ✓not requiring computer science-oriented actions



This display interpretation requires operator tacit knowledge

31

Who may develop such SSW?

- The expertise of mechanical engineers and operators may be exploited to create SSW which is correctly perceived and understood by their users
- Mechanical engineers know users' notations, share their culture, skills, tacit knowledge….
- They are not aware of tool grain and HCI problems

Mechanical engineers

- **acting as domain experts**
- **using high level customized evolutive languages**
- **supported by a team of SE and HCI experts**

32

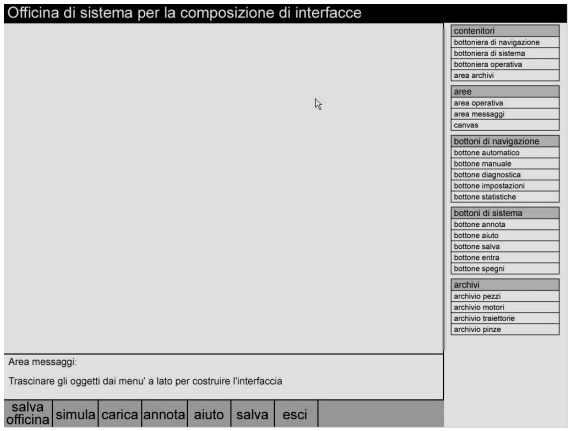
SSW for the mechanical engineer

As domain expert

produces (programs): the assembly line workshops

in the context of design workshop, using the resources of the environment,

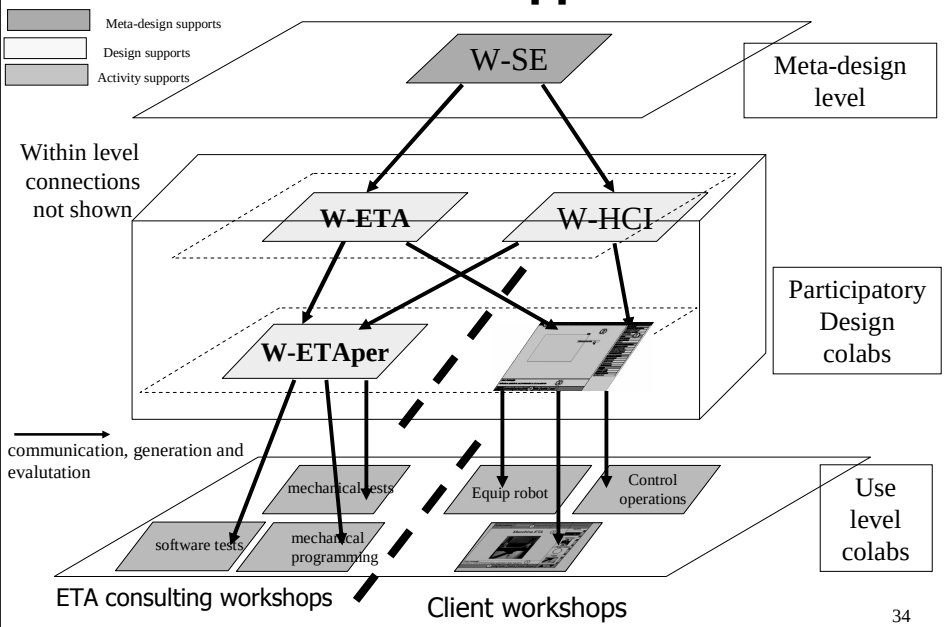
being constrained by the ETA guidelines of Interface construction



This display conveys *d-expert's implicit information* by the traditional design language of the client

33

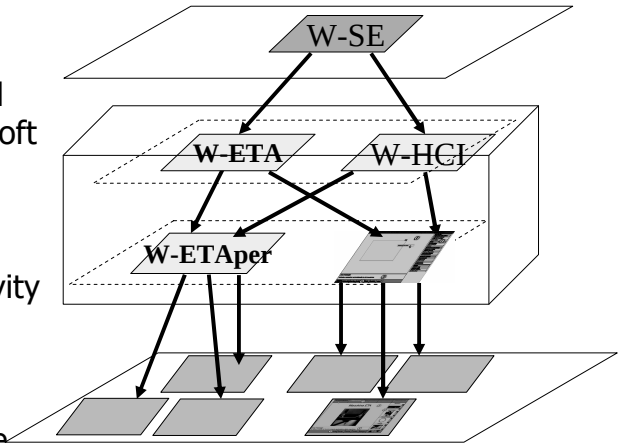
SSWs in ETA application



34

How the team operates in time

- evolutionary prototype
- initial state:
  locally developed system of hard-soft tools
- continuous developmental activity
- Fuzzy boundaries among design, development and use

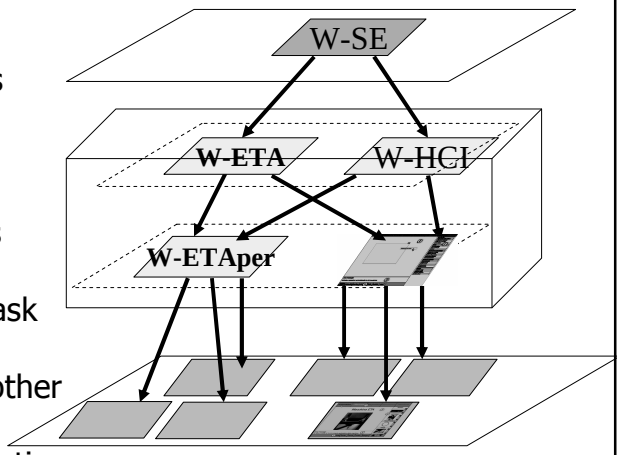


35

The single expert

Views the whole activity through his/ her eyes

- Operates through customized notations and tools
- communicates with other team members on the same level to achieve a common task
- communicates with other levels to achieve usability and co-evolution



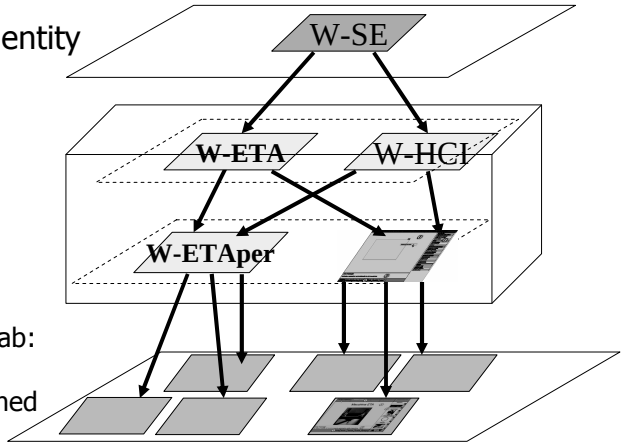
36

Co-workers in each co-lab

Operate on the same entity

In ETA design colab:
Operated entity:
Application Workshop

In ETA and Client use colab:
Operated entity:
The machine to be governed



At any level the activity performed is defined as a developmental process

- interacting with environment
- evolving from an initial state to an (unpredictable) final one

37

Altri approcci

- Approcci pratici da parte di imprese
- SER model: Seeding, Evolutionary Growth, Reseeding
  - Sviluppo non continuo (legato a diverse tecniche di specifica e implementazione)
  - Progettazione parziale (under-design)
  - Gli utenti come progettisti, non è chiaro come partecipino al progetto gli altri stakeholder

38

Conclusions

- The metadata methodology is aimed at allowing
  - designers to create systems adequate for a user community, on the basis of users' knowledge, languages, and notations
  - domain-experts users, not expert in Computer Science, to use and tailor their systems and to perform EUD activities
- Control effects of phenomena characterizing HCI

- Gentle slope of complexity: SSWs offer different levels of complexity, going from simply setting parameters up to extending the system by programming new components
- Users work at anytime with a system suitable to their specific needs
- The formal specification of SSWs is based on the PCL HCI model in which interaction is a syndetic two- interpretations- materializations process

39