



UNIVERSITÀ  
DEGLI STUDI  
DI BRESCIA

# Interazione Persona-Calcolatore

## Fenomeni legati alla HCI

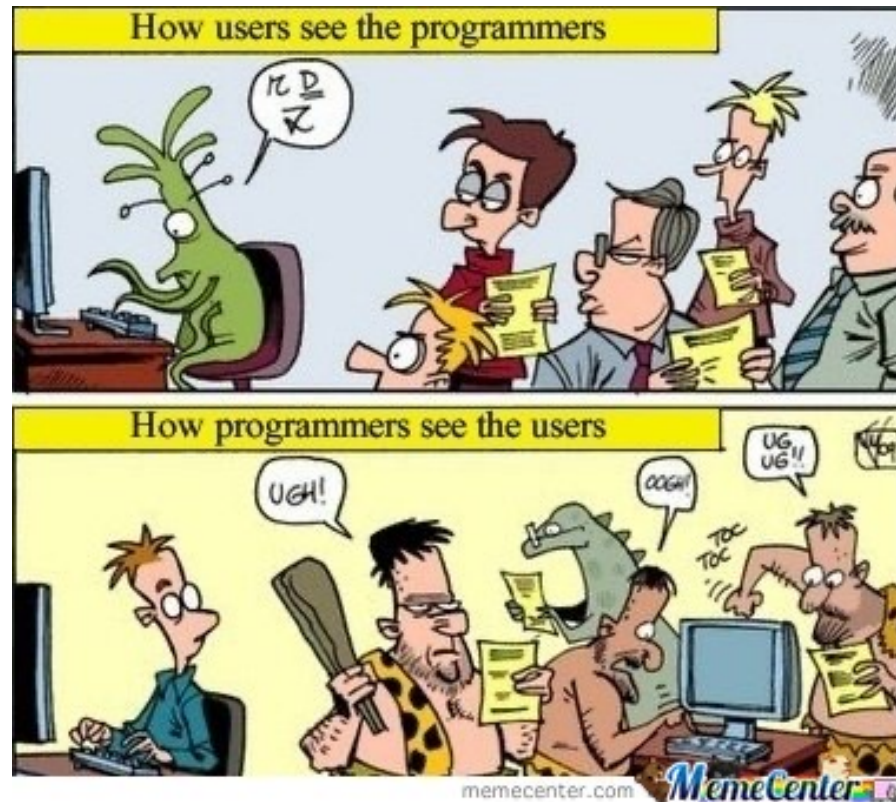
Prof.ssa Daniela Fogli

Dipartimento di Ingegneria dell'Informazione

# Fenomeni che ruotano attorno all'HCI

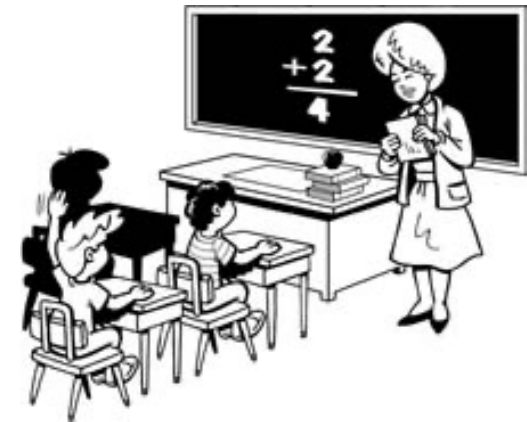
- Divario comunicazionale (*communication gap*)
- Influenza nascosta delle tecnologie (*grain*)
- Diversità degli utenti
- Conoscenza tacita/Informazione implicita
- Co-evoluzione

# Divario comunicazionale (e non solo)



# Progettisti vs Utenti

- Molti progettisti/sviluppatori progettano i sistemi e le interfacce utente secondo il **loro** punto di vista
- Non si preoccupano di capire (**prima**) le pratiche lavorative, le notazioni, il linguaggio degli utenti
- Ma pretendono (**dopo**) di “educare gli utenti”



# ... e gli utenti reagiscono così



Well, it was like the spaceship had landed, and these outer space creatures [trainers] got off, and started talking to us about how we were going to do our job, because nobody understood what they were saying.  
Now, they're talking about notifications, material numbers, document control, material masters – you know, that wasn't in any of our language."

Topi et al., 2005, Identifying usability issues with an ERP implementation

# Divario comunicazionale

- Progettisti e utenti hanno **diversa cultura, differenti approcci all'astrazione, differenti strategie** di costruzione e aggiornamento dei modelli mentali e **diversi modi di ragionare**
- Ogni qualvolta gli utenti non partecipano attivamente al progetto del sistema nasce un rischio di **divario comunicazionale fra utenti e sistema**

# Le differenze fra utenti e progettisti

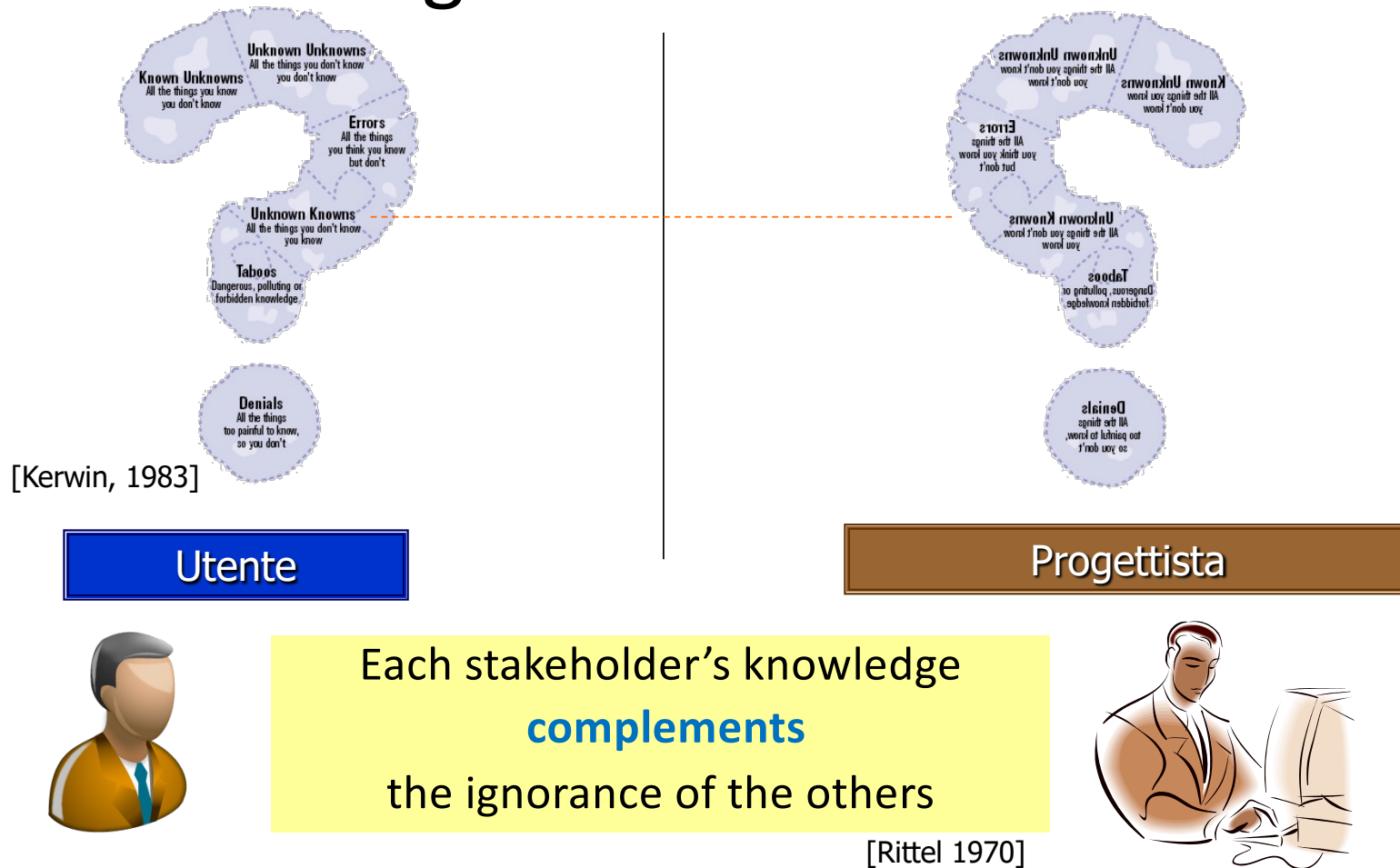
	<b>Utenti</b>	<b>Progettisti</b>
<b>Ragionamento e memorizzazione</b>	Euristico Basato su esempi e analogie	Algoritmico/Deduttivo Basato su modelli
<b>Documentazione, rappresentazione della conoscenza</b>	Notazioni tradizionali di dominio su carta	Notazioni specialistiche per calcolatore
<b>Diverse capacità di articolazione</b>	Tipiche del dominio Uso di strumenti tradizionali	Per l'uso di sistemi di calcolo

# Conseguenze per il progetto

- Conoscere l'utente
- Progettare un corretto modello concettuale  
(progettazione/valutazione iterativa del modello concettuale)
- Coinvolgere l'utente nel progetto sfruttando la 'simmetria dell'ignoranza'



# Simmetria dell'ignoranza



# Influenza nascosta delle tecnologie (Grain)

- Influenza nascosta delle tecnologie o **grain**: tendenza di uno strumento a forzare (implicitamente) il comportamento dei progettisti/programmatori/utenti (*“Each programming paradigm [and system architecture] has a grain, a tendency to push you towards certain kinds of interface”* [Dix et al. 2004])
- Il progettista costruisce sfruttando sistemi hardware e software (il dispositivo fisico, il sistema operativo, l'ambiente di sviluppo)
- Ogni strumento favorisce lo svolgimento di certe attività a scapito di certe altre

# Esempi

- **Content Management Systems:** facilitano lo sviluppo di un sito web ma, scelto un template, impongono una certa struttura al sito
- **Interazione sul singolo pixel:** alcune librerie grafiche rendono difficile gestirla

# Esempio: interazione con tutta la figura geometrica



Se clicco qui  
seleziono  
l'immagine

# Conseguenze per il progetto

- Nel progetto e sviluppo sia **dell'interazione** che del **codice** occorre verificare e valutare ogni strumento usato o sviluppato rispetto all'attività dell'utente
  - Scelta in base agli **obiettivi** tenendo conto dei **vincoli** dello strumento

# Varietà/Diversità degli utenti

- Gli utenti non sono una popolazione uniforme ma costituiscono comunità caratterizzate da diverse culture, scopi e compiti
- Ogni comunità sviluppa propri **dialetti, notazioni, conoscenza e abilità**
- Gli utenti sono **diversi anche all'interno di una stessa comunità**
- Varietà dipende da
  - “storia” dell'utente (competenze, cultura, conoscenza) e abilità fisiche/cognitive specifiche
  - distribuzione geografica
  - compiti da svolgere e contesto delle attività
  - ruolo nell'organizzazione
  - ...

# Conseguenze per il progetto

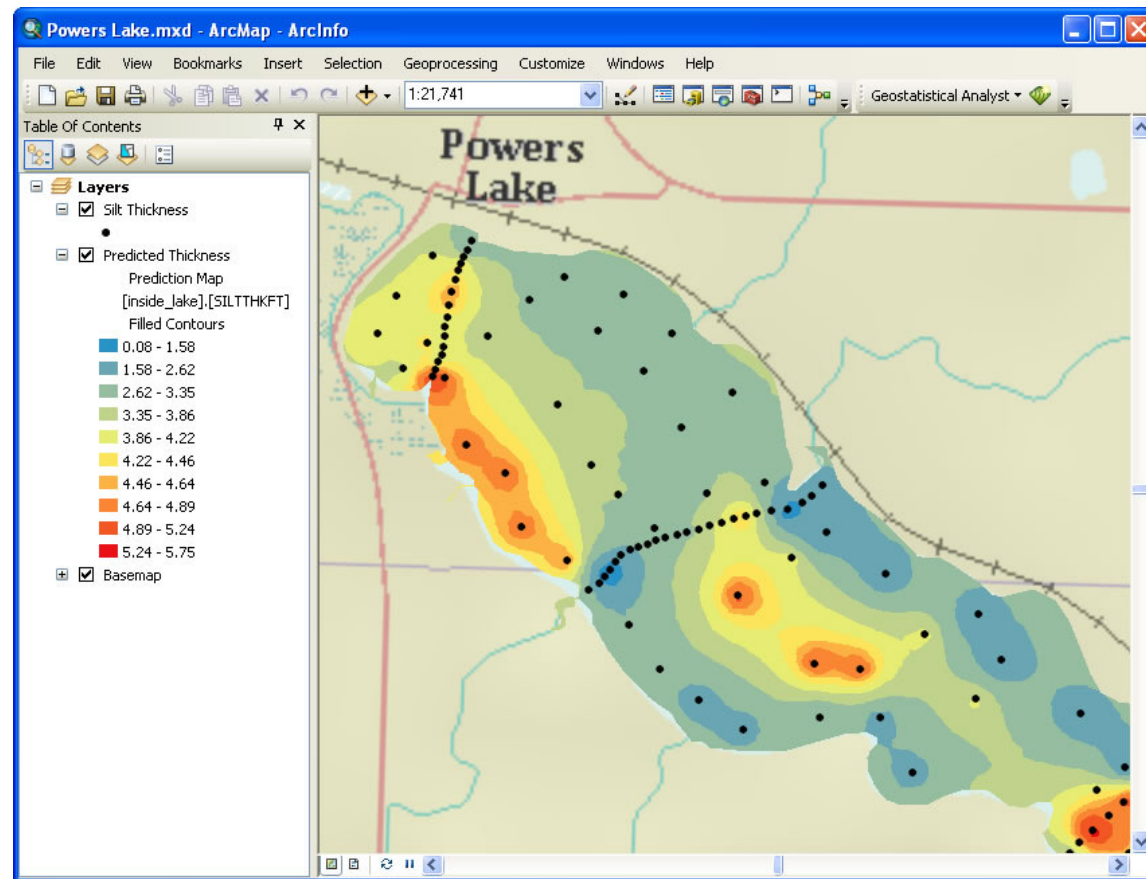
- I sistemi devono:
  - essere sviluppati con tecniche informatiche **modulari** (componenti)
  - essere **personalizzabili** alla comunità (customization)
  - essere **adattabili e adattivi** (rispetto al singolo utente)

# Conoscenza tacita & Informazione implicita

- “We know more than we can tell” [Polany, 1967]
- “Tacit knowledge consists often of habits and culture that we do not recognize in ourselves” [Wikipedia]
- Usiamo **conoscenza tacita** nell’azione e nella decisione
- Usiamo **conoscenza tacita** nel comporre e interpretare i messaggi
- I documenti espressi nella notazione degli utenti e gli strumenti usati, incorporano e rendono disponibile **informazione implicita**
- L’informazione implicita dipende dalla conoscenza tacita dell’autore, dal contesto e dall’attività



Un esperto sa leggere le immagini e capire i testi...

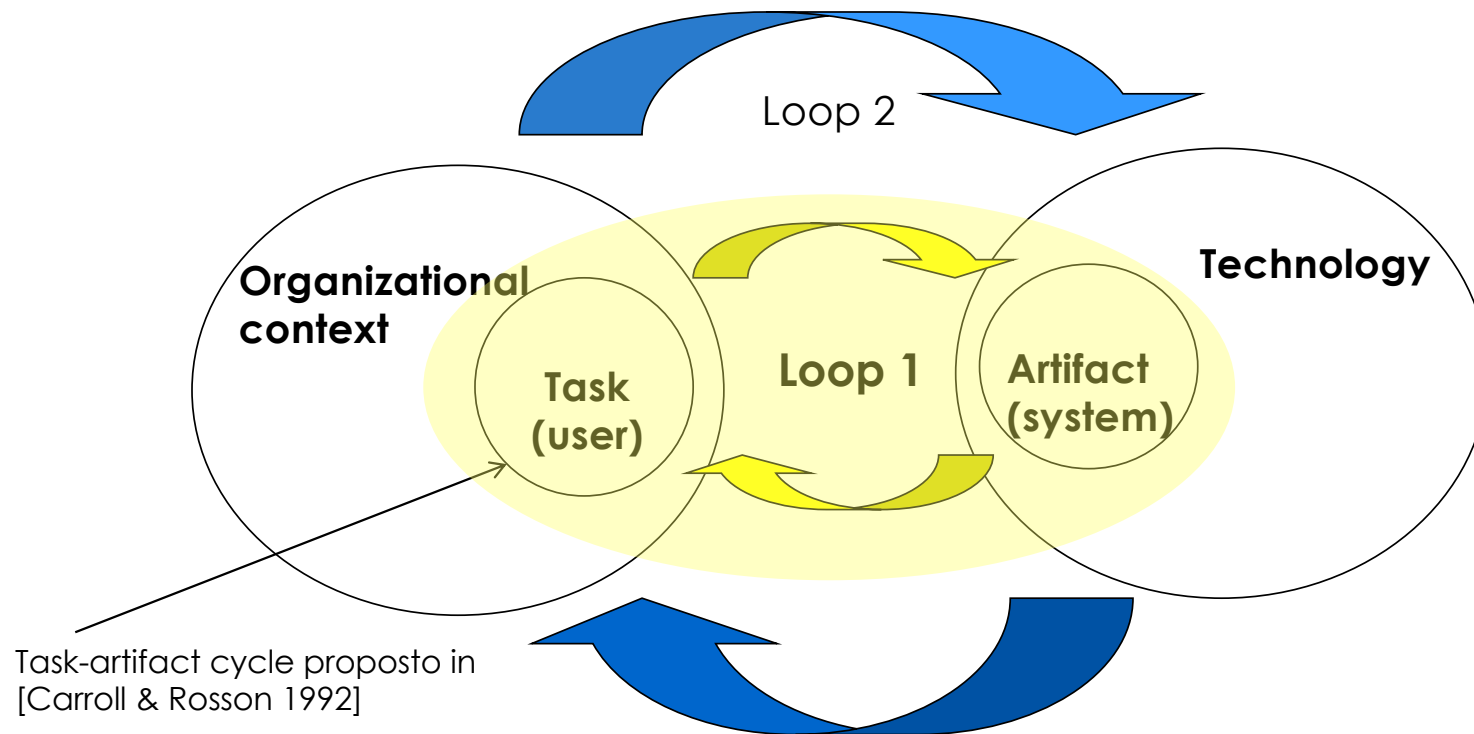


# Conseguenze per il progetto

- Il **modello concettuale** deve basarsi su *conoscenze, notazioni, linguaggio, pratiche lavorative dell'utente*
- Durante il progetto occorre verificare se l'utente percepisce le **strutture presenti** nei messaggi visuali e attribuisce ad esse lo stesso **significato** attribuito dal progettista (e quindi dal programma)

# Co-evoluzione

*“The individual is a moving target. Design for the individual of today, and the design will be wrong tomorrow [...]. This is because as individuals gain proficiency in usage, they need different interfaces than were required when they were beginners” [Norman 2005]*



Adattato da [Bourguin & al. 2001]

# Co-evoluzione: ciclo interno

- *“Using the system changes the users, and as they change they will use the system in new ways”* [Nielsen 1993] ... e di conseguenza i progettisti evolvono il sistema per adattarlo ai nuovi usi
- La co-evoluzione deriva da:
  - ***Creatività dell'utente:***  
Esempio: spreadsheet (fogli elettronici) nati per trattare dati numerici, poi usati per trattare anche dati non numerici...
  - ***Comportamenti ricorrenti:***  
Esempio: nelle applicazioni per ricerca di treni, voli, etc. gli utenti solitamente usano il luogo dove vivono come luogo di partenza...

# Co-evoluzione: ciclo esterno

- Introducendo o cambiando la **tecnologia** cambierà il modo in cui le persone **lavorano e i loro ruoli** (es. pratiche via email, flusso di lavoro regolato dal sistema, ruolo dei commessi negli enti pubblici, ...)
- Questo cambierà l'**ambiente** nel suo aspetto **fisico** (es. PC mobili anziché fissi, tablet al letto del paziente...) e **sociale** (es. tutti in casa che guardano il proprio smartphone), **organizzativo** (es. chi scrive una email di sabato/domenica/25 dicembre si aspetta che il destinatario risponda subito)
- Ma anche, se **cambia l'organizzazione**, potrebbe richiedere l'introduzione di una **nuova tecnologia** o la sua evoluzione (es. lavoro da casa richiede sistemi di teleconferenza, dati in cloud con maggiori requisiti di sicurezza, possibilità di monitorare il lavoro svolto, ...)

# Conseguenze per il progetto

- Il sistema progettato **evolve nel tempo**
- Una volta prodotto non finisce l'attività del progettista
- Occorre progettare il sistema perché possa evolvere – sia adattabile alle nuove esigenze degli utenti ... *magari da parte degli utenti stessi*
  - Necessità di una progettazione (**meta-design**) che permetta la **personalizzazione** da parte dell'utente (**end-user development - EUD**)
  - Con EUD, l'utente contribuisce allo sviluppo del sistema scegliendo, combinando, modificando, creando gli oggetti e gli strumenti di interazione