### Modello di interazione di Norman

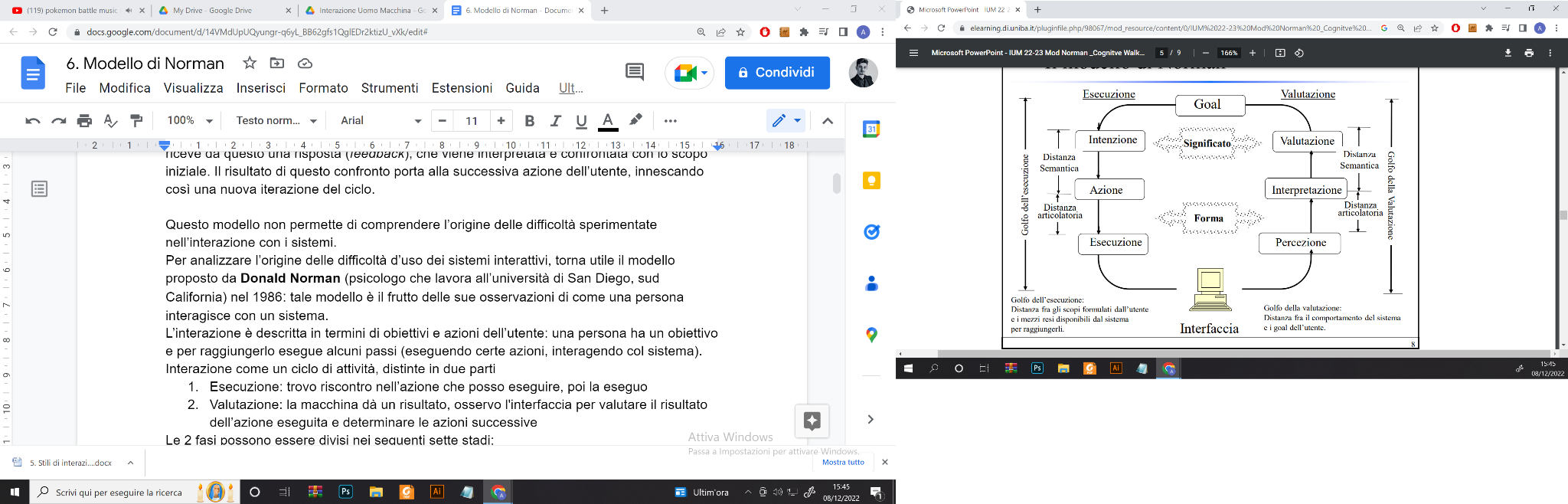
Il modello più semplice d’interazione tra un sistema e il suo utilizzatore è rappresentato dal *ciclo di feedback*: l’utente, per raggiungere il proprio scopo, fornisce un input al sistema e riceve da questo una risposta (*feedback*), che viene interpretata e confrontata con lo scopo iniziale. Il risultato di questo confronto porta alla successiva azione dell’utente, innescando così una nuova iterazione del ciclo.

Questo modello non permette di comprendere l’origine delle difficoltà sperimentate nell’interazione con i sistemi.

Per analizzare l’origine delle difficoltà d’uso dei sistemi interattivi, torna utile il modello proposto da **Donald Norman** (psicologo che lavora all’università di San Diego, sud California) nel 1986: tale modello è il frutto delle sue osservazioni di come una persona interagisce con un sistema.

L’interazione è descritta in termini di obiettivi e azioni dell’utente: una persona ha un obiettivo e per raggiungerlo esegue alcuni passi (eseguendo certe azioni, interagendo col sistema), collocati in un ciclo di attività, distinte in due parti

1. *Esecuzione*: sulla base dell’obiettivo, l’utente formula un piano d’azione, poi lo esegue
2. *Valutazione*: dopo aver eseguito un’azione, la macchina dà un risultato e l’utente osserva l'interfaccia per valutare tale risultato e determinare le azioni successive



Le 2 fasi possono essere divise nei seguenti sette stadi:

1. **Definire l’obiettivo** da raggiungere
2. **Formare l’intenzione**: decidere cosa si intende fare per raggiungere l’obiettivo
3. **Identificare la sequenza di azioni**: guardando l’interfaccia in modo da identificare se ci sono delle funzionalità per eseguire azioni corrispondenti all’intenzione e specificarle.
4. **Eseguire l’azione**: eseguire le azioni pianificate

La schermata (in generale, lo stato del sistema) cambierà, quindi

1. L'utente innanzitutto **percepisce il cambiamento** (relativo allo stato del sistema): quest’interazione non è solo visiva e non è vincolata al solo tipo di interfaccia visuale
2. **Interpretare il nuovo stato del sistema**: si elabora l’osservazione per conferirle un senso
3. **Valutare il risultato rispetto a obiettivi e intenzioni**: se lo scopo iniziale è stato raggiunto allora l’interazione ha avuto successo (passo positivo verso “l’obiettivo a lungo termine”), altrimenti l’utente deve formulare un nuovo obiettivo e ripetere il ciclo.

*Nota: Il modello è indipendente dallo stile dell’interazione; azioni complesse dovranno essere scomposte in azioni sufficientemente semplici, ciascuna delle quali comporterà il passaggio attraverso i sette stadi.*

* *“I sette stadi costituiscono un modello approssimativo, non una teoria psicologica completa. [...] Devono esserci numerose sequenze e l’intera attività può durare ore o anche giorni. C’è anche un continuo anello di retroazione, in cui i risultati di un’attività sono usati per indirizzarne altre”.*

**SCOPO:**

1. Formare lo scopo (*che scopo voglio raggiungere?*)

**ESECUZIONE**

1. Intenzione: formare l’intenzione significa “cosa intendo fare per raggiungere lo scopo?”
2. Azione: specificare un’azione significa “quale azione disponibile nell’interfaccia devo compiere per raggiungere lo scopo?”
3. Eseguire l’azione

**VALUTAZIONE**

1. Percepire lo stato del mondo (che significa? in realtà è cambiato lo stato del calcolatore) → percepire il nuovo stato del sistema
2. Lo interpreto (il risultato): cosa significa il risultato che il sistema ha fornito?
3. Valuto il risultato: ho raggiunto il mio scopo?

Il modello di Norman permette di individuare con grande chiarezza i momenti in cui possono presentarsi dei problemi: nel percorrere gli stadi dell’azione è possibile incontrare difficoltà nel passare da uno stadio all’altro, cioè nell’*attraversare i* ***golfi*** *che li separano*.

**Golfo dell’esecuzione:** distanza tra gli scopi formulati dall’utente e i mezzi resi disponibili dal sistema per raggiungerli.

* E’ maggiore all’aumentare della difficoltà (sulla base della formulazione dell’intenzione (fase 2)) nel trovare le azioni permesse dall’interfaccia del sistema per metterla in pratica (fase 3).
* Tra queste due fasi può non esserci un mapping o potrebbe esserci difficoltà nel trovarlo
* Corrisponde alla differenza tra il modello mentale dell’utente (come egli si aspetta che funzioni il sistema) e il modello delle azioni realizzate dal sistema.
* L’interazione è efficace se il golfo di esecuzione è piccolo

**Golfo della valutazione**: è la distanza fra il comportamento del sistema e i goal dell’utente.

* Lo sforzo per interpretare lo stato fisico del sistema (il risultato) e per determinare fino a che punto questo corrisponda alle aspettative o alle intenzioni dell’utente.
* Se lo sforzo richiesto dall’utente per interpretare la presentazione è minimo, questo golfo è piccolo e l’interazione è efficace.

Tra le varie fasi descritte intercorrono due tipi di distanze:

* **Distanza di tipo semantico**: di significato, tra l’informazione presentata dal sistema e l’informazione ricercata dall’utente, oppure tra l’intenzione e le modalità di esecuzione
* **Distanza articolatoria**: l’esecuzione di un’azione può richiedere dei passi inutili, che rendono l’attività da compiere non immediata (es. devo stampare un documento e ho una sola stampante e il comando di stampa richiede ogni volta di selezionare la stampante). Inoltre, si identifica anche nella distanza tra un’azione intesa (il suo significato) e il modo in cui questa viene eseguita, ovvero la forma fisica per l’esecuzione dell’azione.

Nelle fasi di *esecuzione*

* Es. distanza semantica
  + intenzione = “disegnare un rettangolo”
  + azione → utilizzare tool rettangolo per disegnarlo (bassa distanza semantica)
  + OPPURE utilizzare tool linea per disegnare il rettangolo lato per lato (alta distanza semantica)
* Es. distanza articolatoria:
  + intenzione = “spostare un elemento dalla schermata A alla schermata B”
  + azione → selezionare un elemento, andare dal menu “modifica>muovi” e selezionare la schermata B di arrivo (alta distanza articolatoria)
  + OPPURE → trascinare col cursore del mouse l’elemento dalla schermata A alla schermata B (bassa distanza articolatoria) quando la forma fisica per l’esecuzione di un’azione non mappa direttamente con il suo significato (cliccare una casella del menu per muovere qualcosa, oppure muovere direttamente questo qualcosa).

*Lezione per il progettista*: rapporto forma fisica - significato (per l’utente) → affordance; i tipi di azioni e le sequenze devono essere significative per l’utente.

Nelle fasi di *valutazione*

* Es. distanza semantica
  + il sistema stampa dei numeri in formato matriciale ma l’utente è interessato alle somme di tali numeri riga per riga e colonna per colonna: se i totali non sono stampati allora l’utente dovrà manualmente effettuare i calcoli (distanza semantica); se sono stampati allora la distanza semantica è bassa, perché l’informazione presentata corrisponde a quella cercata dall’utente.
  + esempio: stampare tutti i dati tra i quali l’utente deve selezionare solo quelli aventi una certa proprietà presenta una distanza semantica più alta che stampare direttamente solo i dati che presentano tale proprietà
* Es. distanza articolatoria:
  + Esempio: per capire numeri stampati in una tabella, l’utente effettua uno sforzo ben maggiore rispetto alla visualizzazione di diagrammi contenenti gli stessi numeri (importanza della data visualization)
* Esempio: mostrando una barra di percentuale è più immediato interpretare a che punto si è dalla fine di un certo processo rispetto al caso in cui si mostri solo la percentuale. Anche in questo caso lo sforzo dell’utente sarà minore con un’informazione di tipo visuale.

Lezione per il progettista (per fase di valutazione)

* il problema della rappresentazione percepibile
* informazione necessaria vs informazione inutile

**Cognitive walkthrough**: è una tecnica di ispezione di usabilità focalizzata sulla valutazione di un progetto rispetto alla facilità di apprendimento (utile per WWW)

* L’origine è il code walkthrough in ingegneria del software: i revisori esaminano il codice per controllare certe caratteristiche
* È basato sulla teoria di “exploratory learning” (Polson & Levis 1990)
  + Gli utenti, nel processo di soluzione di problemi, non amano consultare manuali ma preferiscono esplorare le funzionalità del sistema:

1. Iniziano con una grossolana descrizione del compito che vogliono effettuare
2. Esplorano l’interfaccia e scelgono le azioni che ritengono possano effettuare il compito (o una parte di esso)
3. Osservano le reazioni dell’interfaccia per vedere se le loro azioni hanno avuto l’effetto desiderato
4. Determinano quali azioni effettuare successivamente

I 4 punti sono analoghi al modello di Norman; notiamo che il cognitive walkthrough è simile alla valutazione euristica ma molto più dettagliato, pertanto si applica solo a pochi compiti principali. Questa tecnica insegna che anche nell’effettuare una valutazione euristica è importante tenere a mente il modello di Norman

Per effettuare un cognitive walkthrough, c’è bisogno di:

1. Una descrizione abbastanza dettagliata del prototipo del sistema
2. Una descrizione del compito che l’utente deve eseguire sul sistema
3. Una lista scritta completa delle azioni necessarie per completare il task con il prototipo
4. Un’indicazione di chi sono gli utenti e che tipo di esperienza e conoscenza il valutatore può assumere che abbiano

**Esecuzione**: il valutatore procede all’analisi della sequenza di azioni per valutare il sistema e “raccontare una storia credibile sulla sua usabilità”. Per ogni azione cerca di rispondere alle seguenti domande:

1. L’utente, in base alle sue esperienze e conoscenze, è in grado di individuare con facilità l’azione corretta da eseguire (intention-action match)?
2. L’utente è in grado di notare che l’azione corretta è disponibile?
3. L’utente ottiene un feedback nello stesso punto in cui ha eseguito l’azione?
4. Può interpretare correttamente la risposta del sistema, cioè capire di aver scelto l’azione giusta o quella sbagliata?
5. Può valutare i risultati in modo appropriato, cioè determinare se questi soddisfano il suo obiettivo iniziale?

Queste domande guida, in modo simile che per una valutazione euristica, fanno sì che il processo sia dettagliato e sistematico.

**Risultati**: il metodo trova incongruenze tra la concettualizzazione del task del progettista e quella dell’utente, problemi nel progetto di widget dell’interfaccia (menu, icone, bottoni) e di feedback di azioni.

E’ utile produrre moduli standard di valutazione per il walkthrough, che riportino:

* domande guida
* data e ora del walkthrough
* nomi dei valutatori
* per ogni azione, un modulo per rispondere a tutte le domande precedentemente viste
* Per ogni risposta negativa, si riempie un foglio a parte che riporta il problema di usabilità, la sua severità e un'eventuale descrizione della soluzione.

Il walkthrough può essere un processo individuale o di gruppo. Se è di gruppo, una persona scrive i risultati, gli altri partecipano secondo le loro conoscenze (progettisti, esperti del mercato).