

Analisi di task e utenti

Fabio Vitali
6 marzo 2000

HCI



La scelta di task rappresentativi

Il committente è in grado di specificare le funzionalità di massima del sistema, ma non l'elenco effettivo dei task da gestire.

Un incontro diretto con gli utenti permette di arrivare ad una lista di 10-12 task “tipici” che permettono di derivare la lista di funzionalità fondamentali.

Inserire presto gli utenti effettivi nella progettazione viene detto *participatory design*.



Un esempio

Supponiamo di progettare un sistema di simulazione del traffico di un ambiente urbano. Il sistema deve fornire operazioni per cambiare i parametri del traffico e vedere come cambia la simulazione

Un task potrebbe essere:

- ◆ *Un vigile urbano in riposo forzato cambia il limite di velocità di via Stalingrado verso nord tra la tangenziale e viale della Repubblica. Calcolare i flussi di traffico risultanti su viale della Repubblica assumendo una velocità su via Stalingrado tra i 30 e i 60 con incrementi di 5 kmh.*



Dettagli sull'esempio

- 1) Specifica i task, non i comandi da attivare
- 2) E' molto specifico
- 3) Descrive un'azione completa
- 4) Descrive esattamente degli utenti



1) Task, non comandi

Non viene fatto nessun assunto sul tipo di comando da attivare. Questo tipo di specifica può essere usata per confrontare alternative progettuali in maniera equa.

Se fosse stato specificato “cambia il limite di velocità selezionando un nuovo valore da un menù” avremmo predefinito il modo “corretto” per realizzare questo task, con ciò pregiudicandoci l’esplorazione di alternative.



2) Estrema specificità

Non dice soltanto cosa l'utente deve fare con esattezza, ma arriva al punto di specificare esattamente quali strade sono coinvolte.

In pratica, ci permette (ci costringe) a specificare ogni tipo di dettaglio che possa prima o poi diventare rilevante nel valutare alternative progettuali.

Nell'esempio, ci dice che sarà utile poter dividere le strade in tronconi separati e valutare l'effetto del traffico su ogni troncone.

Il caso dei nomi di file lunghi.



3) Azioni complete

La descrizione del task non richiede solo di cambiare la velocità su Stalingrado, o calcolare il flusso di traffico su Repubblica, ma dice che l'utente vuole fare entrambi.

Questo ci permette di valutare come aspetti verosimilmente diversi dell'interfaccia lavoreranno insieme.

Le liste di requisiti dell'ingegneria del software tradizionale è solo una lista di piccole azioni che il sistema deve saper eseguire.

Come queste azioni si sommano per realizzare task completi e significativi non viene di solito considerato.

Partire dalla gestione dell'input e dell'output!



4) Descrizione esatta degli utenti

Come si dettagliano esattamente i task, così dovrebbero essere dettagliati gli utenti. Solo in questo modo sappiamo in qualche maniera calarci nell'atteggiamento, nella psicologia e nell'esperienza dell'utente.

Un sistema esperto di medicina, volendo essere usato dai dottori (che non hanno tempo per imparare nuovi tool) decise di creare un'interfaccia identica ai moduli su carta già riempiti dai dottori nella vita reale.

Questo tipo di decisione non sarebbe stata presa se si fosse deciso di fare un'interfaccia per tecnici medici (che per mestiere debbono imparare il sistema da usare).



Participatory design

Una caratteristica comune dei progetti informatici è una chiara distinzione tra progettisti e utenti. Nel *participatory design* gli utenti vengono coinvolti nella progettazione ad ogni stadio, e partecipano a pari livello con i progettisti alle decisioni. Si parla di *progettazione con*, piuttosto che *progettazione per* gli utenti.

Tre sono le caratteristiche rilevanti del *participatory design*:

- ◆ Lo scopo della progettazione è quello di migliorare l'ambiente di lavoro degli utenti, non di soddisfare commesse, dimostrare eccellenza tecnica, o studiare l'uso di nuove tecnologie.
- ◆ La progettazione è collaborativa, e scopi e decisioni sono raggiunti attraverso negoziazione, e non attraverso imposizione.
- ◆ L'attività è iterativa, e le idee sono provate e accomodate vedendole in funzione in pratica.



Riferimenti

- C. Lewis, J. Rieman, *Task-centered user interface design, a practical introduction*, 1994,
<ftp://ftp.cs.colorado.edu/pub/cs/distrib/clewis/HCI-Design-Book>

