

Metodologia di adattamento del modello di interazione

Survey

Nel presente documento si riassumono i contenuti del deliverable "**D3.4_2 Metodologia di adattamento del modello di interazione**" relativo all'attività *RI 3.4 "Metodologie basate sulla conoscenza per la personalizzazione dell'interazione"*, nell'ambito del terzo Obiettivo Realizzativo (OR 3) "Studio di modelli di descrizione formale dell'utente e del contesto e regole di evoluzione".

Nel presente documento si affrontano le problematiche derivanti dalla progettazione di interfacce adattate al contesto e personalizzate per lo specifico utente. Nel dettaglio si illustra il processo di adattamento ed ottimizzazione di un'interfaccia utente distinguendo l'ottimizzazione di elementi non strutturali come ad esempio colori e font, dall'ottimizzazione di elementi strutturali dell'interfaccia come disposizione dei widget e layout.

Il documento, dopo una analisi delle interfacce e delle problematiche legate alla progettazione, tratta di innovative tecniche di adattamento e sperimenta tale tecniche per l'ottimizzazione delle interfacce in SAPI.

La progettazione di una Interfaccia Utente (IU) è un processo critico, spesso costoso e complesso, solitamente guidato da principi di design e da guidelines.

Molte di queste guidelines sono spesso difficili da tradurre in codice ed un buon progettista di interfacce è spesso guidato, in larga parte, da esperienza e senso estetico. Le guidelines sono spesso utilizzate per organizzare il layout e le caratteristiche di una interfaccia. Tuttavia le attuali guidelines, come Apple's Human Interface Guidelines e Sun's Java Look and Feel Guidelines sono spesso troppo specifiche o troppo vaghe, così da non essere sempre applicabili allo specifico problema. Recenti lavori di Quiroz in IEEE Congress on Evolutionary Computation 2007, Oliver in World Wide Web Conference 2002 e Ichikawa in Genetic and Evolutionary Computation 2003 usano meta euristiche e/o tecniche evolutive per organizzare elementi strutturali di una interfaccia come menu o disposizione degli elementi ed elementi non strutturali come colori e font.

La progettazione di una interfaccia, dunque, coinvolge una serie di problemi decisionali riguardo la struttura, gli attributi e la logica di presentazione. Per esempio, quale è la migliore disposizione dei diversi widget, come suddividere i vari elementi tra i diversi frame, quale palette di colori utilizzare sono comuni problematiche che devono essere risolte in fase di progettazione dell'interfaccia.

La scelta della più appropriata soluzione tra le diverse alternative rappresenta un problema di ottimizzazione legato alla massimizzazione di diverse funzioni obiettivo, ognuna per un diverso modulo del processo di ottimizzazione dell'interfaccia. In questa ottica si rende possibile collegare la progettazione dell'interfaccia alla ricerca dell'algoritmo adottando un approccio generativo nel design e nella costruzione dell'interfaccia utente.

Questo approccio comporta notevoli aspetti positivi, tra i quali:

- Un vasto numero di alternative possono essere esplorate. Tali alternative possono spesso risultare soluzioni innovative e/o supportare proattivamente la creatività umana ed il processo di scelta della migliore interfaccia;
- Allo stesso tempo si possono considerare diversi attributi qualitativi e rispettare le guidelines, mentre per i criteri in conflitto si riesce a trovare il giusto trade-off;
- Si svincolano i progettisti dalla risoluzione dei problemi legati alla ricerca del trade-off tra specifiche contrastanti. Essi potranno dedicarsi all'impostazione di tutte le specifiche lasciando all'algoritmo la ricerca dell'ottimizzazione delle scelte e delle preferenze imposte;
- Le interfacce possono essere automaticamente adattate ad un più largo insieme di dispositivi e ad un più specifico insieme di preferenze utente.

Le tecniche generative non si limitano solo ad algoritmi genetici, ma si estendono ad altre tecniche evolutive ed altre tecniche di ricerca e meta euristiche in grado di esplorare maggiormente l'intero spazio delle soluzioni. La programmazione genetica, ad esempio, si predispone a risolvere più efficacemente problematiche di ottimizzazione di alberi in presenza di vincoli di validità e di parametri legati a ciascun albero.

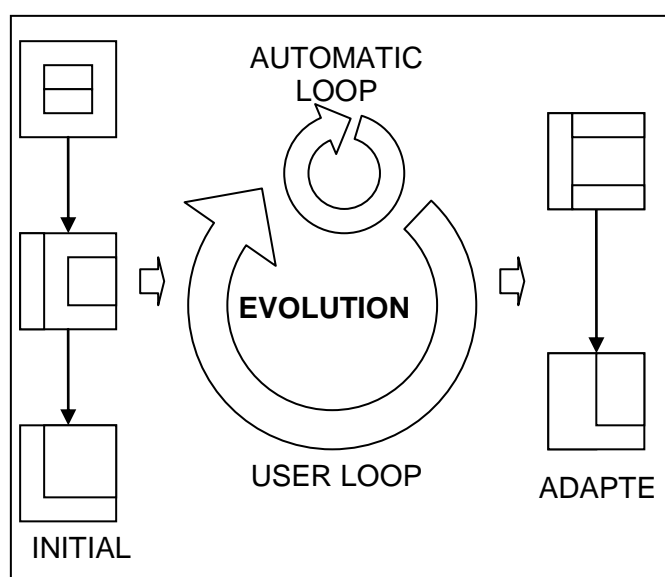


Figura 1 - Interactive Interface Evolution

La funzione obiettivo può essere implicitamente determinata dall'effettivo utilizzo dell'interfaccia. Rilevanti informazioni sono fornite dai feedback utente sia a runtime che in fase di progettazione.

In fase di progettazione i feedback utente sono forniti dal progettista dell'interfaccia il quale, selezionando determinate alternative fornisce delle informazioni che guidano la generazione di interfacce adattate.

A runtime, l'utente finale risponderà in maniera diversa alle diverse interfacce. Tenendo conto di un tempo necessario per l'apprendimento delle funzionalità e dei comandi dell'interfaccia, si valuta la bontà delle interfacce in modo da selezionare l'interfaccia opportuna che rappresenta la migliore soluzione al problema di ottimizzazione.

Tuttavia in entrambi i casi, l'utente viene inserito nel ciclo di ottimizzazione e diviene una parte integrante nella ricerca della migliore soluzione. Ciò conduce alla sperimentazione di algoritmi evolutivi interattivi (Interactive Evolutionary Algorithms) come descritti da Takagi. In particolare essi riguardano l'inserimento dell'utente nel loop del processo evolutivo, come evidenziato in Figura 1.

In questo caso l'evoluzione è data da due loop:

- Il loop più interno esegue in maniera automatica lo screening di un insieme di alternative al fine di identificare le potenziali soluzioni
- Il loop più esterno coinvolge l'utente il quale valuta un insieme più piccolo di alternative al fine di condurre l'algoritmo verso la generazione dell'interfaccia opportunamente adattata.

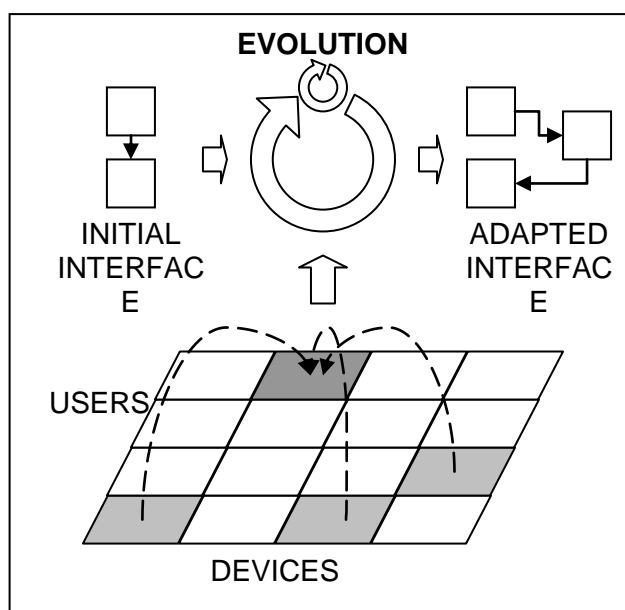


Figura 2 – Utente-dispositivo-interfaccia

Come analizzato nell'attività RI 3.1 dal titolo "Definizione di modelli di profilo utente e delle regole di inferenza/evoluzione del modello" un profilo utente identifica un insieme di preferenze, informazioni (interessi dell'utente, abilità e/o disabilità), regole ed impostazioni che sono usate da un prodotto o da un servizio per fornire all'utente un servizio adatto e personalizzato. Come già affrontato nel documento D3.3_1 "Definizione del modello di contesto e sue regole di evoluzione" si distingue il profilo dallo user context o modello del contesto inteso come l'insieme delle caratteristiche, hardware e software del dispositivo utilizzato per la fruizione del servizio, e le caratteristiche che eventualmente caratterizzano le condizioni di contorno all'applicazione. Come si osserva in Figura 2 una interfaccia deve tener conto del legame esistente tra utenti, contesto (ambiente, dispositivi di accesso, canale

trasmissivo) ed interfaccia per un adeguato adattamento. In SAPI l'adattamento richiede una chiara distinzione tra presentazione e logica dal momento che entrambi possono essere obiettivi del processo di ottimizzazione.

Ottimizzazione delle strutture gerarchiche di una interfaccia

Un primo passo per l'ottimizzazione di una interfaccia utente è stato l'ottimizzazione dei sistemi di menu. I risultati derivanti da tale studio risultano essere applicabile a problemi di ottimizzazione legati alla navigazione orizzontale e verticale di strutture gerarchiche complesse quali ad esempio barre degli strumenti, barre dei menu, menu di navigazione, etc. In altre parole, l'approccio è stato ideato in modo da essere scalabile a qualsiasi struttura che sia riconducibile ad un albero pur di distinguere due insiemi: il primo costituito da menu e sottomenu (nodi dell'albero), il secondo da action da associare ad una specifica voce (nodi foglia dell'albero).

Il sistema ha la capacità di ottimizzare i sistemi di menu nel rispetto di

1. Requisiti di accessibilità (raggiungibilità di una determinata voce del menu)
2. Guidelines internazionali (raccolta di best practices nell'organizzazione dei menu)
3. Preferenze dell'utente (organizzazione delle diverse voci secondo una lista di preferenze)

A seguito della fase di setup, in cui vengono specificati vincoli e preferenze che la struttura dei menu devono soddisfare, si istanzia l'algoritmo e si genera in maniera randomica la popolazione iniziale. Per ciascun individuo della popolazione si assegna un punteggio che varia tra 0 e 1 ed assume valore massimo quando il particolare menu soddisfa tutti i requisiti imposti.

Gli operatori genetici scelti sono:

- Selezione a torneo
- Cross-over semplice
- Mutazione

Dopo N generazioni si ottiene il migliore individuo legale, ovvero l'individuo che rispetta tutti i vincoli obbligatori e massimizza il valore della fitness. Tale valore è massimo quando sono soddisfatte tutte le preferenze specificate. Le preferenze hanno un peso diverso ai fini del calcolo della fitness e ciascuna ha associato un parametro che tiene conto dell'importanza della specifica preferenza. Si assume una scala di priorità compresa tra 1 e 5, considerando un vincolo a priorità 1 come una preferenza molto importante, mentre un vincolo a priorità 5 come un vincolo poco importante. Inoltre un vincolo con priorità 0 è considerato un vincolo che definisce la legalità della struttura generata. Dunque, da un lato i vincoli a priorità 0 consentono la ricerca di un albero legale, dall'altro i restanti vincoli indirizzano la ricerca

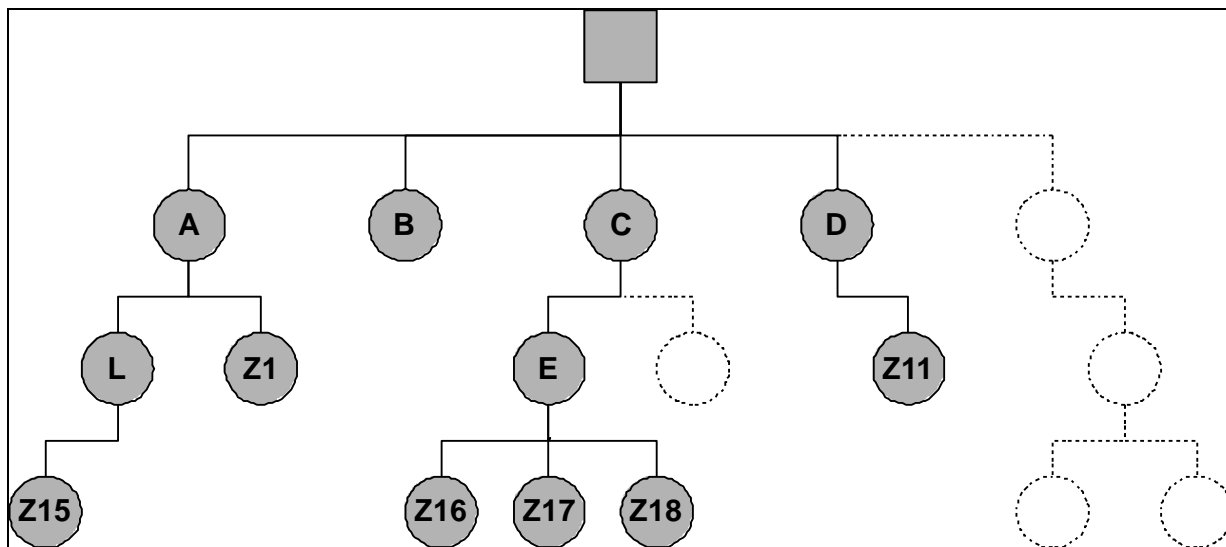


Figura 4 - Esempio di albero a cui corrisponde un sistema di menu.

Palette dei colori dell'interfaccia

Rendere accessibile una interfaccia è un obiettivo che coinvolge anche l'uso del colore. Se i colori definiti nei fogli di stile generano combinazioni tra primo piano e sfondo poco visibili, la presentazione grafica sarà inaccessibile. Poiché la grande maggioranza degli utenti riceve le informazioni attraverso la vista, ecco allora che un uso appropriato del colore diventa fondamentale ai fini dell'accessibilità.

Per evitare di dover ogni volta effettuare una analisi sperimentale e raggruppare ogni volta un team di validazione per confermare l'accessibilità dei colori scelti si progetta un algoritmo in grado di simulare la percezione dei colori secondo diversi modelli a seconda della disabilità visiva ed in grado di ottimizzare i colori forniti dal progettista dell'interfaccia.

Data una paletta di colori iniziali l'algoritmo tende ad ottimizzare la scelta dei colori garantendo il contrasto di luminosità tra i colori che sono in relazione tra di loro e senza perdere le informazioni iniziali legate alla tonalità colore. Infatti si cerca la migliore combinazione di colori che rispettando il contrasto non si allontanano di molto dai colori specificati dal progettista dell'interfaccia.

La generica paletta viene codificata all'interno di un cromosoma come si vede in Figura 5.

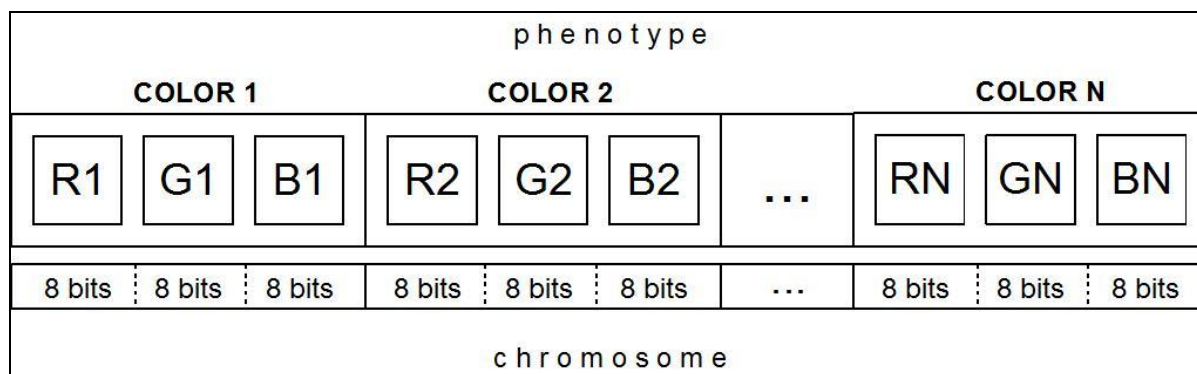


Figura 5 - Codifica

Per le relazioni tra colori codificati nel cromosoma si considerano tre modelli, mostrati in Figura 6:

- Semplice M0, nel quale ogni coppia adiacente di colori è messa in relazione.
- Medio M1, nel quale vengono messe in relazione triple di colori.
- Complesso M2, nel quale vengono create relazioni più complesse.

La funzione di fitness (Equazione 1) che si massimizza tiene conto di due fattori:

1. Il contrasto di luminanza tra ogni coppia di colori correlati
2. La distanza cromatica di ciascun colore da quello originale presente nella paletta di partenza

Equazione 1:

$$f = \left(\prod_{i=1}^n (1 - d_i) \prod_{j=1}^k c_j \right)^{\frac{1}{n+k}}$$

Dove

$$d_i = \frac{\Delta E_i}{\Delta E^*}$$

$$c_j = \frac{20 + \min(C_j - T_j, 0)}{20}$$

Il contrasto di luminanza (contrast ratio) C è definito come descritto nelle WCAG 2.0 del W3C.

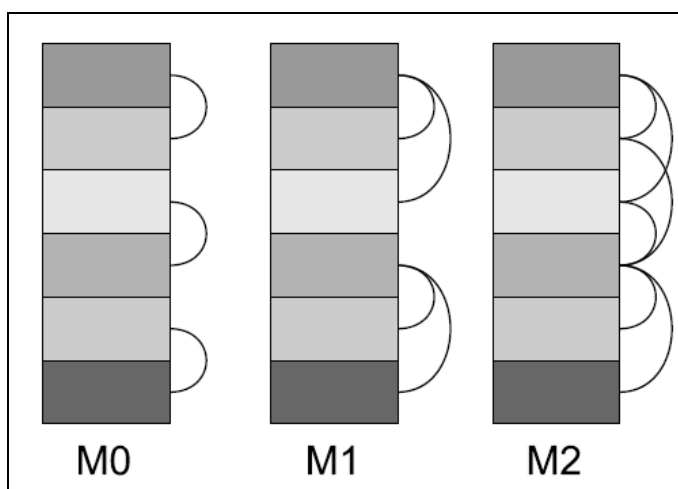


Figura 6 - Modelli di relazione tra i colori

A partire da un set di 6 colori il cui contrasto inizialmente è basso e considerando le relazioni secondo il modello M1, si ottiene un set di altrettanti colori il cui contrasto è molto buono e la funzione obiettivo è massimizzata.

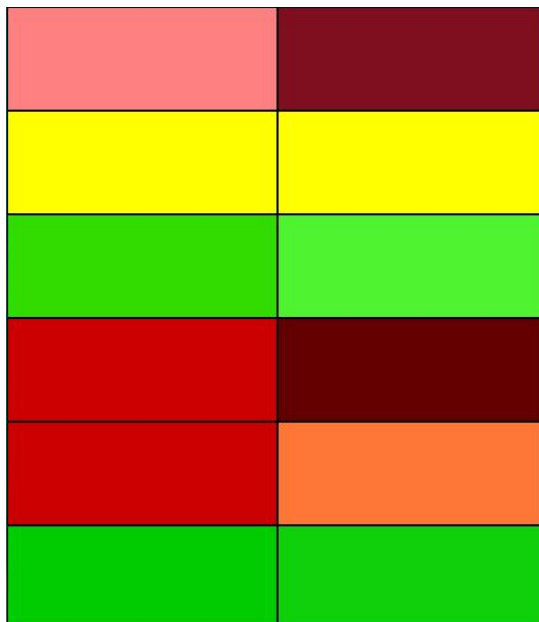


Figura 7 - Ottimizzazione di una paletta iniziale di 6 colori correlati secondo il modello M1

Questo stesso test (a partire dallo stesso set di colori e adattandoli adeguatamente a seconda del problema considerato) è stato effettuato anche per la deuteranopia e per la protanopia ottenendo i risultati mostrati in Figura 8 e Figura 9 la cui fitness assume rispettivamente valore 0,953255 e valore 0,961878.

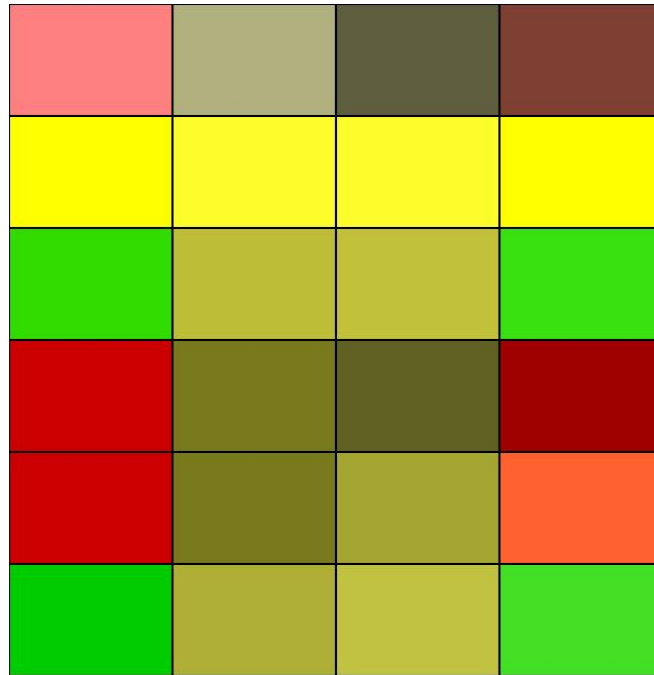


Figura 8 – Ottimizzazione della palette originale (mostrata nella prima colonna) per utenti affetti da deuteranopia. La seconda colonna mostra come la palette viene percepita dai deuteranopi, mentre la terza e la quarta mostrano le palette ottimizzate percepite da normodotati e deuteranopi.

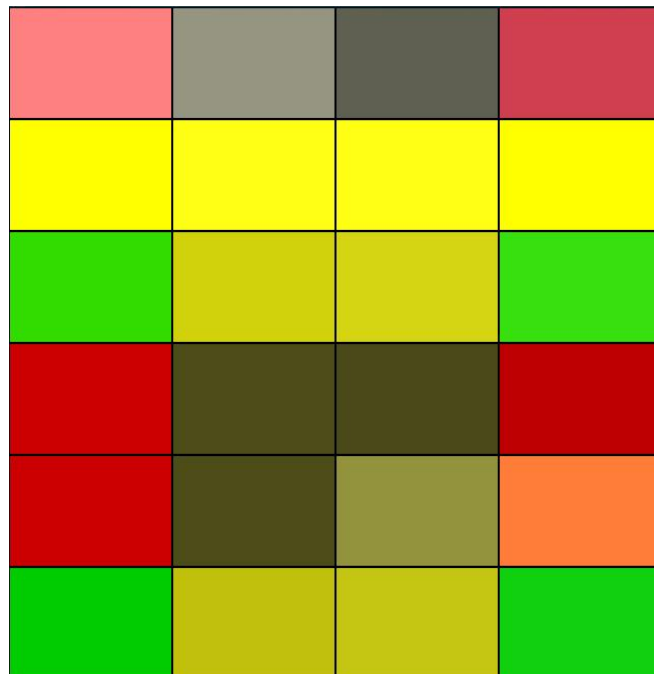


Figura 9 - Ottimizzazione della palette originale (mostrata nella prima colonna) per utenti affetti da protanopia. La seconda colonna mostra come la palette viene percepita dai protanopi, mentre la terza e la quarta mostrano le palette ottimizzate percepite da normodotati e utenti affetti da protanopia.

Conclusioni

L'approccio adottato è stato considerato innovativo e promettente dai revisori di recenti conferenze internazionali come l'ottava conferenza europea su Evolutionary Computation in Combinatorial Optimisation (EVOCOP 08) e la conferenza internazionale su Computer Human Interaction (CHI 08). Una sintesi dei lavori presentati è disponibile al seguente link: http://www.poste.it/azienda/sapi/SAPI_Pubblicazioni.html