



UNIVERSITÀ DEGLI STUDI DI ROMA
“LA SAPIENZA”

Tesi di laurea specialistica in Informatica

SWift:
SignWriting improved fast
transcriber

Studente:
Fabrizio BORGIA 697964

Relatrice:
Maria DE MARSICO
Correlatrice:
Claudia S. BIANCHINI

Anno Accademico 2009/10

Indice

1	Un mondo silenzioso	7
1	La lingua dei segni	7
2	Mancanza di una forma scritta	9
3	Errori e paradossi	10
4	Notazioni proposte	12
2	SignWriting	15
1	Cos'è il SignWriting	15
2	Le origini del SignWriting	19
3	Studio del SignMaker	21
3.1	Display del Segno	23
3.2	Menu dei Glifi	24
3.3	Pannello degli Strumenti	26
3	Sordità e Information Technology	29
1	Situazione attuale	29
2	Information Scent	30
3	Hyperlinking per le lingue dei segni	32
4	Progettazione del <i>SWift</i>	35
1	Identificazione dei Requisiti	35
1.1	Requisiti Utente funzionali	36
1.2	Requisiti Utente non funzionali	38
2	Requisiti di Sistema	39
2.1	Requisiti di Sistema funzionali	40
2.2	Requisiti di Sistema non funzionali	44
5	Storyboard del <i>SWift</i>	47
1	Prima fase di progettazione (2009.12 2010.03)	47
1.1	Interfaccia	47
2	Seconda fase di progettazione (2010.04)	50
2.1	Interfaccia	50
2.2	Interfaccia - Pulsanti	56

3	Terza fase di progettazione (2010.05)	59
3.1	Interfaccia	59
3.2	Interfaccia- Pulsanti	61
3.3	Database	63
4	Quarta fase di progettazione (2010.06 2010.8)	65
4.1	Interfaccia	65
4.2	Database	68
6	Interfaccia del SWift	71
1	L'abito fa il monaco	71
1.1	Criteri utilizzati	72
2	Analisi del SWift	74
2.1	Display del Segno	75
2.2	Menu dei Glifi	77
2.3	Pannello degli Strumenti	82
2.4	Pannello del Completamento	83
7	Logica del SWift	85
1	Linguaggi	85
2	Architettura	87
2.1	MVC	87
2.2	SWift e MVC	89
2.3	Database	99
8	Conclusioni	101
9	Glossario	103
10	Indice degli acronimi	105
11	Ringraziamenti	107
	Bibliografia	108

Introduzione

Il SignWriting improved fast transcriber (SWift), presentato in questo lavoro, è un editor per scrivere e trascrivere con l'aiuto del computer utilizzando il SignWriting (SW), che è uno dei sistemi inventati finora per rappresentare le lingue dei segni. Il SWift è stato sviluppato per tutti gli utenti del SW, quindi non esclusivamente per i sordi. L'applicazione consentirà loro di comporre il segno desiderato, aiutati da un sistema di completamento automatico che renderà più veloce e intuitivo il processo di composizione, e di salvare il segno in modo da non perdere il lavoro svolto. Il SWift ha l'ambizioso obiettivo di contribuire ad abbattere le barriere "elettroniche" che tengono la comunità dei sordi isolata dal web: essendo un editor, ed essendo sviluppato modularmente, può essere inserito in qualsiasi piattaforma desideri utilizzare il SW come lingua madre. A causa della natura del mio lavoro, prima di proseguire nella lettura è necessaria la conoscenza di alcuni termini tecnici, la maggior parte dei quali relativi al mondo dei sordi, quindi si invita, prima di continuare, a consultare il glossario presente alla fine del documento. La prima parte del mio lavoro (Cap. 1-3) spiegherà problematiche e concetti relativi al mondo dei sordi, per consentire al lettore di comprendere il contesto in cui si inserisce il SWift e le motivazioni che hanno reso necessaria la sua esistenza. E' facile intuire che, essendo il presente elaborato una tesi di Informatica e non essendo io uno studioso del mondo dei sordi, i primi capitoli saranno ricchi di citazioni di articoli e libri sui quali ho studiato nelle prime fasi del mio lavoro. La seconda parte (Cap. 4-7) illustrerà gli aspetti più tecnici della progettazione e dello sviluppo del SWift. Le conclusioni e qualche riga sugli sviluppi futuri saranno al Cap. 8. Alla fine del documento, come è d'obbligo, saranno disponibili il glossario, l'indice delle sigle utilizzate, la bibliografia e ringraziamenti.

Capitolo 1

Un mondo silenzioso

1 La lingua dei segni

“Sottilissime variazioni nell’esecuzione dei segni, quasi impercettibili per gli occhi degli udenti non abituati a questa lingua, possono apportare importanti mutamenti a livello morfologico e sintattico.”

Le persone sorde, anche se educate ad un linguaggio parlato, comunicano tra loro in una forma visivo-gestuale diversa da quella vocale degli udenti. Questa forma di comunicazione non va confusa con una semplice mimica ma è una lingua a tutti gli effetti, una lingua dei segni con caratteristiche molto complesse. Il mondo degli udenti ha ignorato a lungo questo sistema di comunicazione, a volte lo ha combattuto sottovalutandone i pregi perchè veniva giudicato incompatibile con l’acquisizione della lingua parlata. Per la prima descrizione sistematica di una lingua dei segni bisogna aspettare la metà del 1700, periodo in cui l’Abbè dell’Epèe scoprì che i suoi studenti sordi comunicavano attraverso segni naturali e decise di usare questi segni per insegnare loro la lingua parlata e scritta. Il suo successore nella scuola per sordi di Parigi fu Sicard, che studiò a lungo la LSF (Langue des Signes Française), e questa, grazie a Gallaudert fu introdotta negli Stati Uniti, diventando l’ASL (American Sign Language). Anche in Italia esisteva e veniva usata una lingua dei segni tra i sordi ma nel 1880 si tenne un congresso a Milano nel quale si privilegiò la comunicazione orale e che impedì una più ampia diffusione della lingua dei segni. In molti paesi, l’interesse per questa lingua si sviluppò negli anni ’60 grazie all’opera di Willhelm Stokoe che rintracciò nell’ASL una struttura simile a quella delle lingue vocali, infatti come nelle lingue vocali, che con un numero ristretto di fonemi danno vita a un infinito numero di parole dotate di significato, così nelle lingue dei segni, con un ristretto numero di unità minime (cheremi) si può ottenere un numero

vastissimo di unità dotate di significato (i segni).

Stokoe si concentrò principalmente sulla componente manuale della lingua dei segni, arrivando a proporre una notazione in cui un segno si può comporre secondo tre parametri:

- **il luogo** nello spazio dove le mani eseguono il segno
- **la configurazione** delle mani nell'eseguire il segno
- **il movimento** delle mani nell'eseguire il segno

“Nell’ASL sono stati identificati 19 configurazioni, 12 luoghi e 24 movimenti dalla cui combinazione si hanno tutti i segni possibili dell’ASL” (Corazza, Volterra, 2004). Un altro parametro importante che è stato individuato più tardi rispetto all’analisi originaria di Stokoe è **l’orientamento** del palmo delle mani. Alcuni segni presentano la stessa configurazione, lo stesso luogo o lo stesso movimento, ma si differenziano soltanto per l’orientamento del palmo della mano.

“Inoltre è stato appurato che l’ASL possiede una serie di regole precise di tipo grammaticale: sottilissime variazioni nell’esecuzione dei segni, quasi impercettibili per gli occhi degli udenti non abituati a questa lingua, possono apportare importanti mutamenti a livello morfologico e sintattico” (Corazza, Volterra, 2004). Analizzando l’ASL con un minimo di attenzione e sistematicità, ci si è accorti dell’esistenza di una serie di meccanismi che permettono di codificare tutte quelle informazioni che vengono espresse da alcune lingue vocali tramite gli articoli, le preposizioni, il sistema flessionale o l’ordine delle parole nelle frasi. Questi meccanismi sono fondamentalmente:

- l’uso particolare dello **spazio**
- la **modificazione sistematica del movimento** con cui viene prodotto il segno
- la produzione di **movimenti non manuali**, dal capo e degli occhi, le espressioni facciali, l’orientamento e la postura di tutto il corpo

Più di recente, gli studi di Cuxac (Cuxac, 2000) hanno dimostrato che, oltre alle componenti appena descritte, anche lo sguardo gioca un ruolo fondamentale nel determinare il significato del segno. Concludendo, è stato rintracciato nell’ASL un sistema che per molti versi presenta caratteristiche analoghe al sistema fonologico, morfologico e sintattico delle lingue vocali, anche se permane una differenza importantissima. Nelle lingue vocali gli elementi che formano una parola occorrono sequenzialmente in ordine lineare, mentre gli elementi che formano un segno occorrono simultaneamente o perlomeno si sovrappongono quindi spesso non possono venire analizzati in sequenze temporali ma piuttosto in termini di unità spaziali e di movimento che coesistono

all'interno di un unità di tempo. Va infatti tenuto presente che il linguaggio dei segni non è affatto universale come si potrebbe pensare ma vi sono tante lingue dei segni quante sono le varie comunità dei sordi. Ciascuna comunità crea e sviluppo una sua lingua dei segni con caratteristiche proprie legate al gruppo in cui viene usata e ai bisogni comunicativi che deve assolvere. (Corazza, Volterra, 2004)

2 Mancanza di una forma scritta

“Ad oggi, nessuna lingua dei segni, nel mondo, ha sviluppato autonomamente una forma scritta.”

“Da un punto di vista strettamente linguistico, un sistema di scrittura è, per ogni comunità che lo usa, un codice socialmente condiviso utilizzato per la trasmissione di testi, che consenta di superare limitazioni di spazio e di tempo” (Pizzuto *et al.*, 2006) limitazioni che, nel caso di una lingua a tradizione esclusivamente orale, diventano determinanti. I bisogni di comunicazione che possono determinare la nascita di un sistema di scrittura possono essere diversi: la necessità di trasmettere un corpus di leggi, la trasmissione di testi letterari, l'elaborazione di dizionari e grammatiche per l'istruzione. Si può dire quindi che i sistemi di scrittura sono creati per rispondere a bisogni di comunicazione, artistici e di istruzione, e sono progettati per quello. *“Di conseguenza, i sistemi di scrittura forniscono un'analisi delle strutture linguistiche sufficienti per ottenere particolari scopi, ma non un'analisi esaustiva delle strutture linguistiche: le strutture linguistiche che sono codificate in un sistema di scrittura sono quelle necessarie a veicolare i significati che devono essere comunicati in un particolare contesto e per un particolare scopo o utilizzo. Questo avviene sempre attraverso un processo sociale di elaborazione, diffusione e istituzionalizzazione del sistema di scrittura”* (Pizzuto *et al.*, 2006).

D'altra parte, i sistemi di trascrizione sono strumenti utilizzati principalmente dai ricercatori per analizzare strutture linguistiche, e sono sviluppati per rappresentare le caratteristiche linguistiche studiate dal ricercatore. I sistemi di trascrizione, quindi, non influenzano l'utilizzo (e la varietà) del linguaggio, mentre i sistemi di scrittura si.

“La storia dei sistemi di scrittura dimostra che in generale, la possibilità di trascrivere testi è subordinata alla nascita di un sistema di scrittura: i sistemi di trascrizione sono facilitati della nascita di un sistema di scrittura perché quest'ultimo provvede una rappresentazione di competenza linguistica socialmente condivisa e comunemente approvata” (Pizzuto *et al.*, 2006).

Possiamo quindi dire che la trascrizione nasce dopo la scrittura: per trascrivere serve un sistema di scrittura. La specificità delle lingue dei segni è che

attualmente non sono dotate di un sistema di scrittura, cosa che le rende paragonabili alla lingue a tradizione esclusivamente orale che, si ricorda, costituiscono le maggior parte delle lingue esistenti (Ong, 1982). Mancando, quindi, di un sistema di scrittura, sorge il problema di sviluppare un sistema di trascrizione adeguato per le lingue dei segni, e sorge, ben più importante, un dubbio: si può evitare di sviluppare un sistema di scrittura? Più precisamente la domanda, così come è stata esplicitata da Pizzuto *et al.* è: *“Siamo sicuri che, nel tentativo di sviluppare appropriati strumenti di notazione e trascrizione per le lingue dei segni, si possa aggirare lo sviluppo di una qualche forma di scrittura prima di procedere oltre?”*.

Molte lingue parlate e la maggior parte delle lingue dei segni, non avendo un loro sistema di scrittura, vengono trascritte utilizzando il sistema di scrittura di altre lingue. Per le lingue dei segni questo porta con se molti problemi e complicazioni. *“Primo tra tutti il fatto che le lingue dei segni, nei discorsi che sono sempre faccia a faccia (anche quando si comunica a distanza con tecnologie video), trasmettono l’informazione in modalità multilineare: nella comunicazione visiva/segnica”* (Di Renzo, 2008), le informazioni vengono veicolate dallo sguardo, dalle mani, dall’espressione facciale, dalla testa e dalle spalle.

Le difficoltà che sono state appena elencate hanno minato, finora, il processo di sviluppo di una forma scritta per le lingue dei segni, aggravate anche dal fatto che *“i segnanti sordi, che ovviamente devono essere i primi attori di questo processo di sviluppo, vivono in un universo diglossico in cui la loro lingua dei segni deve co-esistere con con la lingua dominante scritta e parlata da, e in iterazione con, la comunità udente circostante”* (Pizzuto *et al.*, 2006). In assenza, quindi, di una forma scritta, non è facile, per i ricercatori sordi, valutare l’appropriatezza dei vari strumenti di notazione e trascrizione che vengono proposti per la loro lingua dei segni.

3 Errori e paradossi

“Potremmo mai accettare un dizionario francese che non contiene nessuna rappresentazione delle forme proprie della lingua francese [...] ?”

Molti sforzi sono stati compiuti dalla comunità scientifico-linguistica, per venire a capo del problema della forma scritta della lingua dei segni, alcune notazioni sono state proposte, riscuotendo consensi più o meno ampi, tali notazioni verranno illustrate nel paragrafo successivo, prima, però, è bene sottolineare gli errori e i paradossi, purtroppo frequenti, in cui ci si imbatte facendo ricerca in questa direzione. *“Uno dei problemi più importanti è l’utilizzo, infelice ma purtroppo ancora molto diffuso delle cosiddette “glosse, per suddividere e “trascrivere quelli che vengono considerati i “significati di*

base dei segni, indentificati da sequenze segnate e testo (Pizzuto *et al.*, 2006). Il termine glossa è in realtà un modo sbagliato di chiamare un'operazione di etichettatura eseguita nell'ambito della ricerca sulla lingua dei segni. Le glosse, infatti, se utilizzate in modo appropriato nell'annotazione di linguaggi scritti/parlati, sono sempre una componente aggiuntiva, che non rimpiazza, ma accompagna, in un linguaggio di riferimento noto all'autore e al lettore di determinato studio, una rappresentazione indipendente del linguaggio oggetto dello studio.

L'esempio seguente¹, riportato nel già citato articolo di Pizzuto *et al.* preso da una descrizione di E.G.Pulleyblank del Yoruba (una lingua Nigeriana) illustra il punto:

ó	gbé	e	wá
lui/lei	portare	esso	venire

“Lui lo portò”

Pizzuto *et al.* continua spiegando che la prima riga fornisce una rappresentazione ortografica indipendente delle unità costituenti analizzate nella sequenza di Yoruba descritta. La seconda riga fornisce, in corrispondenza uno a uno, le glosse in Inglese per gli elementi rappresentati nella prima riga. La terza riga fornisce una traduzione in Inglese. Questo è un esempio plausibile e utile delle glosse, viste come una notazione ausiliaria che aiuta il lettore a capire (attraverso etichette scritte in una lingua conosciuta) i criteri lessicali e morfologici del Yoruba. Ma gli elementi costituenti della sequenza originale sono e devono essere rappresentati indipendentemente, altrimenti non avremmo semplicemente alcuna idea dei criteri di forma/significato del linguaggio oggetto dello studio.

Molto differentemente, nell'ambito della ricerca sulla lingua dei segni, le glosse vengono utilizzate come principale e unico mezzo per la rappresentazione dei segni in forma scritta. Ad esempio, come si legge in Pizzuto *et al.*, una resa, attraverso le glosse di una sequenza LIS (Lingua dei Segni Italiana) con un significato comparabile alla sequenza in Yoruba di cui prima potrebbe essere:

INDICAZIONE-a INDICAZIONE-b PORTARE

¹Qui tradotto dall'originale in Inglese.

Un testo (in Inglese, Italiano etc.) accompagna la rappresentazione per descrivere, ad esempio, dove sono collocati e diretti nello spazio, i segni INDICAZIONE, o se il verbo etichettato come PORTARE sia o meno collocato nello spazio, e come. Il problema è che, nel secondo caso, il lettore non ha alcun modo di ricostruire le forme della lingua dei segni che sono state prodotte. *“Non c’è una rappresentazione indipendente dei segni. Tutto ciò che abbiamo sono semplicemente etichette a parole per i significati che abbiamo assegnato a forme che semplicemente non ci sono. Di conseguenza, qualsiasi trascrizione di una lingua dei segni effettuata in questo modo non può essere definita una trascrizione nel senso appropriato della parola”* (Pizzuto et al., 2006). Il paradosso di analizzare una lingua (qualsiasi lingua) esprimendosi esclusivamente un’altra lingua è evidente a tutti: *“Potremmo mai accettare un dizionario francese che non contiene nessuna rappresentazione delle forme proprie della lingua francese, in cui le parole francesi vanno cercate tramite delle parole italiane eventualmente collegate, in ambienti multimediali, a delle registrazione audio (o audio-video) che presentano le parole francesi associate alle parole italiane, e in cui tuttavia altre informazioni sulle parole francesi (categoria grammaticale, spiegazioni sul significato ecc.) vengono date ancora una volta in italiano?”* (Antinoro Pizzuto et al., 2009).

4 Notazioni proposte

“In maniera diversa tra loro, tutti [...] i sistemi presentano però grandi limiti”

I primi tentativi di rappresentare la lingua dei segni risalgono al 1800. Le soluzioni proposte per rappresentare le lingue dei segni sono state le più varie, in questo paragrafo citerò un articolo (Gianfreda et al., 2009) che richiama tre tipi di notazione.

La descrizione “a parole”. Utilizzata in vari modi per illustrare segni singoli, decontestualizzati (ad es. in dizionari), o sequenze discorsive in lingua dei segni. Nel caso di segni singoli, viene descritto il modo di esecuzione del segno, o l’azione rappresentata dal segno, o entrambe contemporaneamente. Così, riporta Gianfreda et al. il segno ottocentesco della Lingua dei Segni Francese (LSF) che significa “vivere” (Bonnal-Vergès, in stampa) viene descritto, in due diversi dizionari dell’epoca, come *“far risalire la V, in dattilologia, di entrambe le mani, dal basso del petto fino all’altezza delle spalle”* (Lambert, 1865) e come *“rappresentare il principio di calore e di movimento che anima i corpi, che li fa sentire, crescere, muoversi ed agire”* (Sicard, 1808). Nella rappresentazione di sequenze discorsive, le parole tratte dalle lingue vocali/scritte, impropriamente definite “glosse”, sono usate come eti-

chette verbali che in qualche modo “traducono” il significato di base dei segni (in una specifica lingua vocale/scritta), e che “*possono essere accompagnate da particolari annotazioni per descrivere specifici aspetti morfologici o morfosintattici*” (Gianfreda *et al.*, 2009).

Ad esempio, la sequenza

“BAMBINO NONNA ABBRACCIA [+ dir]”

può essere usata per rappresentare un enunciato in LIS che corrisponde all’enunciato italiano “il bambino abbraccia la nonna”. Ciascuna parola in caratteri maiuscoli rappresenta un segno manuale LIS, e la notazione “[+ dir]” segnala la presenza di modifiche morfologiche del segno “ABBRACCIA” nel caso specifico un cambiamento della direzione del movimento del segno per indicare l’agente e il paziente (“chi abbraccia chi”).

La descrizione “a simboli”. Nel secondo tipo di notazione individuato in Gianfreda *et al.* possiamo ascrivere quella elaborata da Bébien (1825) nella prima metà dell’800: un complesso sistema di simboli che permette di descrivere le componenti manuali del segno, parametro per parametro, rendendo così lineari i segni (che lineari non sono). A questo filone appartengono anche sistemi più moderni quali la notazione di Stokoe (1960), l’HamNoSys (Prillwitz *et al.*, 1989), fortemente debitore di Stokoe, ed il Sign Font (Newkirk, 1989).

La descrizione bidimensionale “a disegni”. Nel 1853, l’educatore Frère Louis, della congregazione dei Frères de St Gabriel, scrive, in modo colorito, “*Sarebbe più facile salire in groppa ad un mulo e balzare sopra la cattedrale di Nantes che [...] fare un dizionario [...] dei segni senza usare i disegni*” (Bonnal-Vergès, in stampa). Questa opinione è ancora molto diffusa, tanto che i moderni dizionari dei segni (ad es. Moody *et al.*, 1997; Radutzky, 2001; Romeo, 1991) sono basati su disegni accompagnati da etichette verbali a volte accompagnati da altri sistemi di notazione specialistica (v. ad es. Radutzky, 2001).

In maniera diversa tra loro, tutti questi sistemi presentano però grandi limiti. “*Una lista, non esaustiva, di questi limiti comprende: grande complessità sia nella scrittura che nella lettura, mancata connessione con la forma effettiva del segno, grande difficoltà o impossibilità nel trasporre discorsi in segni, omissione di parti importanti dei segni, quali soprattutto le componenti non manuali, difficoltà di memorizzazione dei criteri utilizzati per trasporre i segni, quasi impossibilità a riutilizzare i dati per scopi diversi da quelli per cui sono stati raccolti, difficoltà a rendere la dinamicità del segno e il peculiare uso dello spazio segnico*” (Gianfreda *et al.*, 2009).

Il SignWriting (SW), a cui è dedicato tutto il capitolo successivo (Cap. 2) sembra, finora, non presentare molti dei limiti appena elencati: la caratteristica che distingue il SW dalle altre notazioni è che riesce a combinare la spazialità di un disegno bidimensionale con un sistema di codifica che utilizza simboli fortemente iconici, quindi intuitivi e facili da ricordare.

Capitolo 2

SignWriting

1 Cos'è il SignWriting

Il SignWriting (SW)¹ è un sistema che usa simboli visuali (i glifi) per rappresentare configurazioni, movimenti ed espressioni facciali delle lingue dei segni. Può essere visto come un alfabeto: una lista di simboli usati per scrivere qualsiasi lingua dei segni del mondo. L'alfabeto del SignWriting può essere paragonato a quello che noi usiamo in Italiano, detto Romano, che può essere utilizzato per scrivere molte lingue vocali differenti. Ciascuna lingua vocale aggiunge o sottrae uno o due simboli, ma gli stessi simboli di base possono essere utilizzati per scrivere in Italiano, Inglese, Francese e così via. Dunque l'alfabeto Romano è internazionale, ma i linguaggi che lo utilizzano non lo sono. Allo stesso modo, i simboli del SignWriting sono internazionali e possono essere usati per la LIS, o per l'ASL, e così via.

Una delle caratteristiche che rende il SignWriting molto promettente rispetto alle altre notazioni, è che riesce, da solo ad esprimere una sequenza segnata, senza il bisogno di essere accompagnato da descrizioni accessorie, magari scritte in un'altra lingua come nel caso delle tanto contestate “glosse”.

I glifi del SignWriting sono tutti raccolti in un alfabeto, l'International SignWriting Alphabet (ISWA) che quindi contiene tutto l'occorrente per esprimere qualsiasi lingua dei segni del mondo. L'ISWA è disponibile, concretamente, sotto forma di archivio contenente decine di migliaia di immagini (.png), ciascuna delle quali rappresenta un glifo del SW. A causa del loro grande numero, i glifi sono stati codificati, associando a ciascuno una stringa della forma visibile in Fig. 2.1 (Ovviamente a ciascun 1 corrisponde una cifra da 0 a 9).

¹Le informazioni sul SignWriting, Valerie Sutton ed altri argomenti strettamente collegati (come il SWML) riportate in questo capitolo sono state prese dai seguenti siti:

<http://www.signwriting.org>

<http://www.valeriesutton.org>

<http://www.signbank.org>

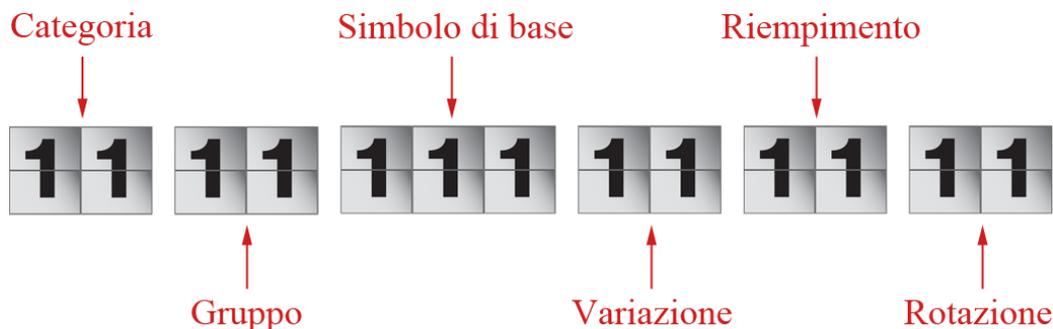


Figura 2.1: Codifica numerica dei glifi del SW.

Nella Fig. 2.1 compaiono, associate ad ogni componente numerica, le seguenti etichette:

- **Categoria:** le prime due cifre della codifica determinano la più importante differenziazione all'interno dell'ISWA; distinguono le aree anatomiche tra loro e da altre categorie come la punteggiatura, i contatti. In Sutton sono presenti 7 categorie (configurazioni, movimenti, testa e viso, corpo, dinamica e ritmo, punteggiatura, annotazione avanzata). Esempio: tutte le configurazioni sono inserite nella categoria 01-*
- **Gruppo:** ogni categoria è suddivisa al massimo in 10 gruppi, che permettano di distinguere diverse aree nelle macro-aree individuate dalle categorie. Spesso i gruppi all'interno di una categoria sono molto eterogenei, basti pensare che i gruppi presenti nella categoria 02-* sono tutti i movimenti (delle mani, dell'avambraccio, del polso e delle dita) ma anche i contatti.
Esempio: le configurazioni sono suddivise in 10 gruppi a seconda delle dita utilizzate, il gruppo 01 (dunque 01-01-*) contiene tutte le configurazioni in cui è dominante l'uso del solo indice
- **Simbolo di base:** individuano uno specifico glifo all'interno di un gruppo.
Esempio: nel primo gruppo delle configurazioni, sono presenti 12 simboli di base, di cui il primo (01-01-01-*) indica l'indice dritto
- **Variazione:** possono essere posti allo stesso livello dei glifi di base, perché servono a differenziare due simboli molto simili ma comunque distinguibili.
Esempio: l'undicesimo simbolo di base delle configurazioni con l'indice (dunque 01-01-011-*) è l'indice piegato all'altezza della nocca. Nelle due variazioni di questo simbolo (01-01-011-01-* e 01-01-011-02-*) la differenza è dovuta all'angolo assunto dalla nocca

- Riempimento:** individuano delle modificazioni di uno stesso simbolo di base.
 Esempio: nelle configurazioni il riempimento serve a distinguere il lato della mano visibile e il piano in cui è eseguito il segno. Così a seconda che si veda il palmo, il taglio o il dorso della mano, e che il la mano sia poggiata sia sul piano verticale od orizzontale, si avranno 6 riempimenti diversi (01-01-011-01-01-*, 01-01-011-01-02-* etc.)
- Rotazione:** come per i riempimenti, individuano delle modificazioni di uno stesso simbolo di base.
 Esempio: nelle configurazioni la rotazione serve a distinguere l'orientamento della mano, ossia come è girata, e la mano usata (destra o sinistra). A seconda della rotazione e della mano usata, si possono vere 16 rotazioni diverse(01-01-011-01-01-01, 01-01-011-01-01-02 etc.)

La tabella seguente mostra una divisione in categorie dei glifi del SW, ed aiuta a capire, concretamente, cos'è un glifo:

Range di codici	Esempio di glifi	Descrizione della famiglia
01-***-***-***-***-***		Configurazioni della mano.
02-01-***-***-***-***		Contatti: indicano dove, come e quante volte una mano entra in contatto con un'altra parte del corpo.
Da 02-02-***-***-***-*** a 02-10-***-***-***-***		Movimenti (delle mani, dei polsi, degli avambracci, ecc.).
03-***-***-***-***		Testa (espressioni facciali, movimenti, ecc.).
04-***-***-***-***		Spalle, braccia e busto.
05-***-***-***-***		Dinamica e coordinazione del movimento.
06-***-***-***-***		Punteggiatura.

Tab. 2.1: Classificazione dei glifi del SW.

L'ISWA si evolve nel tempo: vengono inseriti nuovi glifi di cui si sente il bisogno, si effettuano restyling di alcuni glifi di cui non è immediato il significato, si cambiano codifiche, ecc.

Per un non-addetto ai lavori (ad esempio per un informatico) è molto difficile districarsi tra decine di migliaia di glifi. Individuare le caratteristiche da

utilizzare per classificarli, al fine di sviluppare una funzione di ricerca efficiente ed efficace, è quasi impossibile, perciò mi sono avvalso dell'aiuto di C. S. Bianchini, dottoranda in linguistica presso l'ISTC-CNR, che si è rivelato indispensabile per la progettazione e lo sviluppo del SWift. Il suo progetto di dottorato consiste nello studio delle potenzialità del SW come strumento di scrittura e trascrizione delle Lingue dei Segni, e nell'analisi delle riflessioni metalinguistiche dei sordi dovute al fatto di poter rappresentare in forma scritta la loro lingua (Bianchini, 2010; Gianfreda, 2009). La parte del suo lavoro più rilevante per questa tesi è la sua riclassificazione dell'ISWA, che ha eseguito con lo scopo di migliorarne la coerenza e la funzionalità. Questo ha implicato un aumento notevole dei glifi presenti nell'ISWA ma anche una sua semplificazione in quanto la maggior coerenza ne facilita l'utilizzo e l'apprendimento (Bianchini, 2011). Nell'ambito di questa riclassificazione, Bianchini ha suddiviso l'ISWA in categorie, famiglie e sotto-famiglie, le quali contengono una serie di glifi-prototipi che vengono "declinati" sulla base di regole senza eccezioni: ogni glifo è frutto di una combinazione unica tra un prototipo (appartenente ad una categoria), una famiglia e una sottofamiglia, e una realizzazione di ogni regola che gli si applica.

La Fig. 2.2 dà un'idea fedele di quanto appena detto:

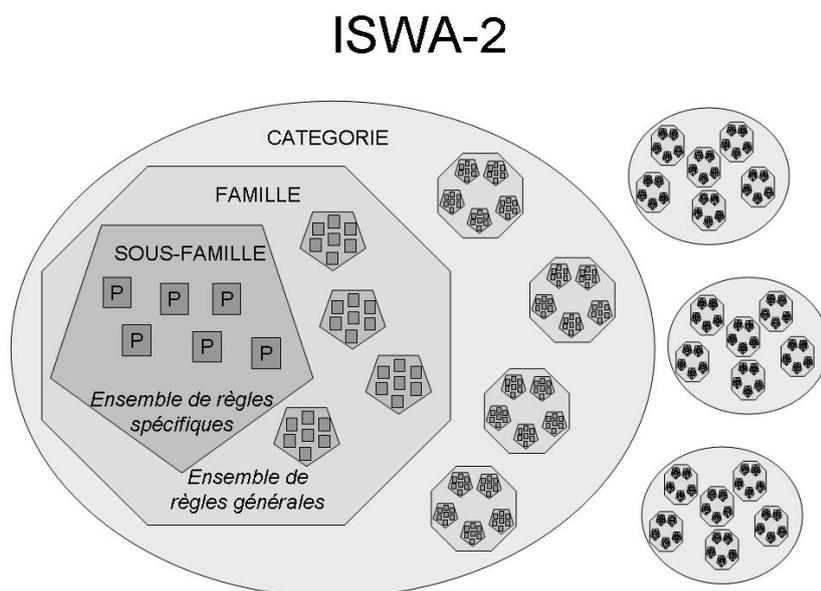


Figura 2.2: Classificazione Bianchini dei glifi del SW (immagine estratta da Bianchini (2011))

Nella Fig. 2.2 compaiono, associate ad ogni componente numerica, le seguenti etichette:

- **Categoria:** suddividono l'ISWA in 12 grandi insiemi, basandosi soprattutto sulle aree anatomiche e i tipi di elementi da codificare: configurazioni, movimenti delle mani, movimenti dell'avambraccio, movimenti dei polsi, movimenti delle dita, dinamica, coordinazione, contatti, espressioni e movimenti del viso, espressioni e movimenti della testa, espressioni e movimenti del corpo, punteggiatura
- **Famiglie:** ritagliano le categorie in sottogruppi, all'interno dei quali vigono regole generali che sono applicabili a tutti i glifi che le compongono. Nessun glifo può appartenere ad una famiglia se non ne rispetta tutte le regole
- **Sottofamiglie:** ritagliano le famiglie in sottogruppi, all'interno dei quali vigono regole sia le regole generali delle famiglie che regole specifiche delle sotto-famiglie. Nessun glifo può appartenere ad una sottofamiglia se non ne rispetta tutte le regole
- **Prototipi:** sono i glifi di base che, se opportunamente "coniugati" sulla base delle regole di famiglie e sottofamiglie, permettono di ottenere tutti i glifi presenti in quelle determinate famiglie e sottofamiglie

Categorie, Famiglie e Sottofamiglie, sono quasi sempre derivate da un ritaglio diverso delle categorie e dei gruppi di Sutton. Le regole sono spesso individuabili in caratteristiche che erano elencate nelle variazioni, nei riempimenti e le rotazioni. I prototipi corrispondono spesso ai simboli di base e alle loro variazioni.

All'interno della mia tesi, ho utilizzato come base questa classificazione molto minuziosa, ma senza sfruttarne tutti i gradi di dettaglio in quanto questo avrebbe comportato un numero troppo elevato di scelte per l'utente, rendendo più lento il procedimento di scelta del glifo.

2 Le origini del SignWriting

Il SignWriting fa parte di un più ampio sistema di scrittura: il Sutton Movement Writing & Shorthand, inventato da Valerie Sutton: ex-coreografa statunitense, che si propone l'ambizioso scopo di fornire notazioni per tenere traccia di ciascun movimento del corpo. Sutton paragona il suo lavoro ad un albero, con il seme profondamente piantato nel ventesimo secolo, ma che darà i suoi frutti nel ventunesimo: effettivamente il sistema che poi avrebbe preso il nome di Sutton Movement Writing & Shorthand nasce nel 1966, quasi per gioco, con il DanceWriting: il primo dei sistemi di notazione inventato da una giovane Valerie Sutton, che all'epoca aveva 15 anni e frequentava un corso di ballo professionale, che aveva come scopo la registrazione delle coreografie di danza.



Figura 2.3: Esempio di notazione Dance Writing.

Negli anni '70 il DanceWriting arriva in Europa insieme alla sua inventrice, che in quel periodo si allenava con gli insegnanti del Royal Danish Ballet. In questo periodo il sistema viene ufficialmente alla luce, prima con la pubblicazione del primo libro di testo sull'argomento, nel 1973, poi con articoli su molti giornali danesi e corsi alla Royal Danish Ballet. Il SignWriting nasce l'anno dopo, nel 1974, quando il prof. Lars von der Lieth e il suo team di ricerca dell'Università di Copenaghen, avendo bisogno di un modo per annotare segni e movimenti, chiesero a Valerie Sutton di annotare i movimenti da un video. Ovviamente il SignWriting non aveva lo stesso aspetto che ha ora: infatti il sistema che abbiamo oggi, testato e usato da molti gruppi di persone, è arricchito dai miglioramenti scaturiti da anni di studio, ma nel 1974, il SignWriting era essenzialmente il DanceWriting dal bacino in su, con poche differenze.

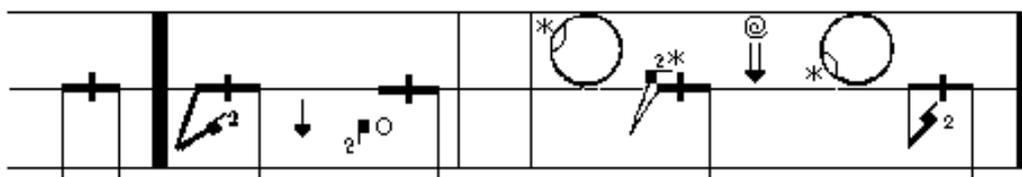


Figura 2.4: Esempio di notazione Sign Writing (1974).

Gli anni dal 1975 al 1980 furono di transizione: l'attenzione si spostò lentamente dal DanceWriting al SignWriting, mentre quest'ultimo si diffondeva dalla Danimarca agli Stati Uniti, e più tardi in altri paesi, ottenendo sempre maggiore attenzione. Una pietra miliare nello sviluppo del SignWriting fu posta nel 1980, anno in cui Sutton presentò l'articolo "A Way To Analyze American Sign Language & Any Other Sign Language Without Translation Into Any Spoken Language" al National Symposium on Sign Language Research and Teaching a Boston, Massachusetts.

3 Studio del SignMaker

Scrivere e trascrivere utilizzando il SignWriting con l'aiuto del computer è la frontiera verso la quale si spinge il mio lavoro, e come ogni viaggio degno di questo nome, è stato necessario informarsi su ciò che era, e che non era, stato fatto in questa direzione. Durante la mia ricerca mi sono imbattuto nel SignMaker: un'applicazione sviluppata dal team di ricerca di Valerie Sutton, che permette la scrittura e il salvataggio dei segni utilizzando semplicemente un browser web.

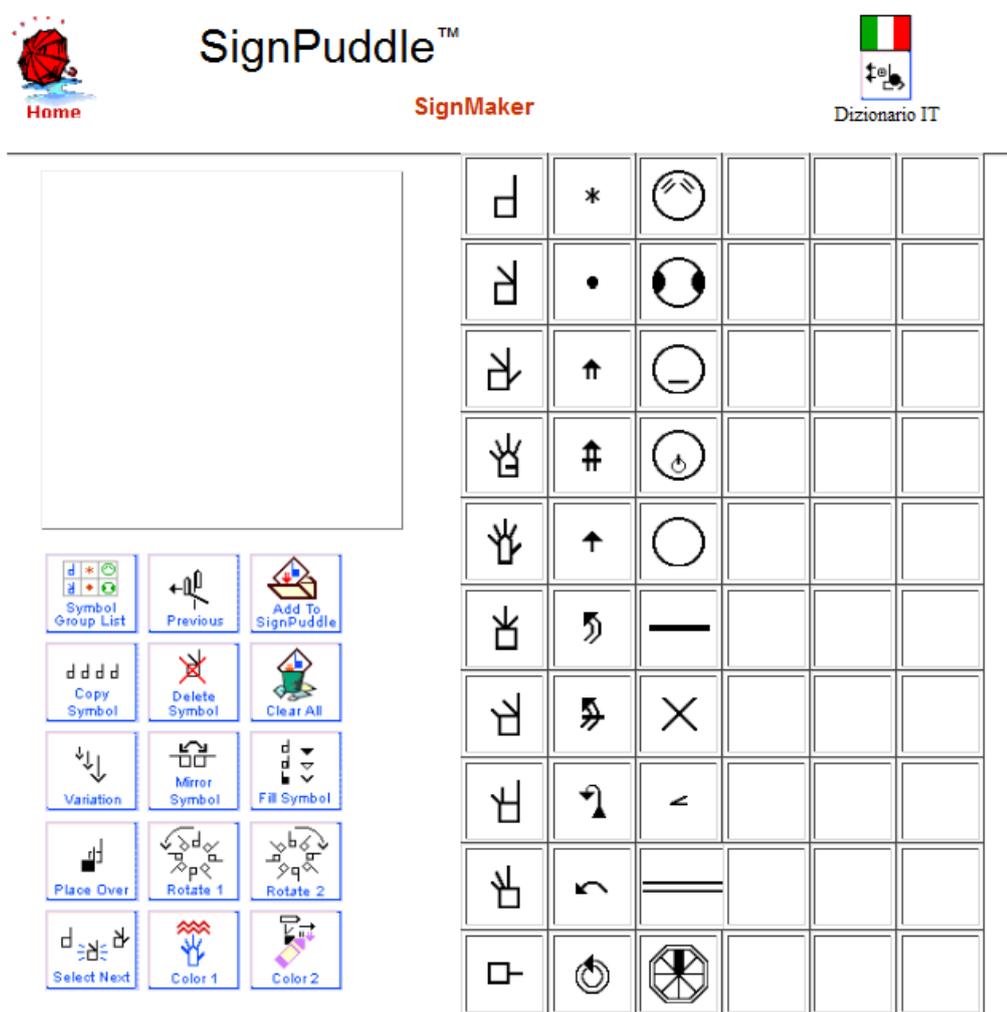


Figura 2.5: Home del SignMaker, attualmente disponibile all'indirizzo <http://www.signbank.org/SignPuddle1.5/signmaker.php?ui=1&sgn=63>.

Analizzando l'applicazione si possono individuare tre aree d'interesse, distinte per aspetto e funzioni svolte, che ho chiamato nei seguenti modi:

- **Display del Segno:** un'area simile ad una lavagna bianca sulla quale verrà formato, glifo dopo glifo, il segno.
- **Menu dei Glifi:** menu che consente, attraverso numerosi sottomenu, di scegliere il glifo da inserire nel Display del Segno.
- **Pannello degli Strumenti:** un' area dell'editor che contiene pulsanti che permettono di eseguire importanti operazioni come: modifica del glifo selezionato, reset del Display del Segno, ecc..

Presento di seguito alcune valutazioni di carattere qualitativo riguardanti il SignMaker: mi sono concentrato principalmente sulle funzioni offerte e sugli aspetti che potevano essere migliorati.

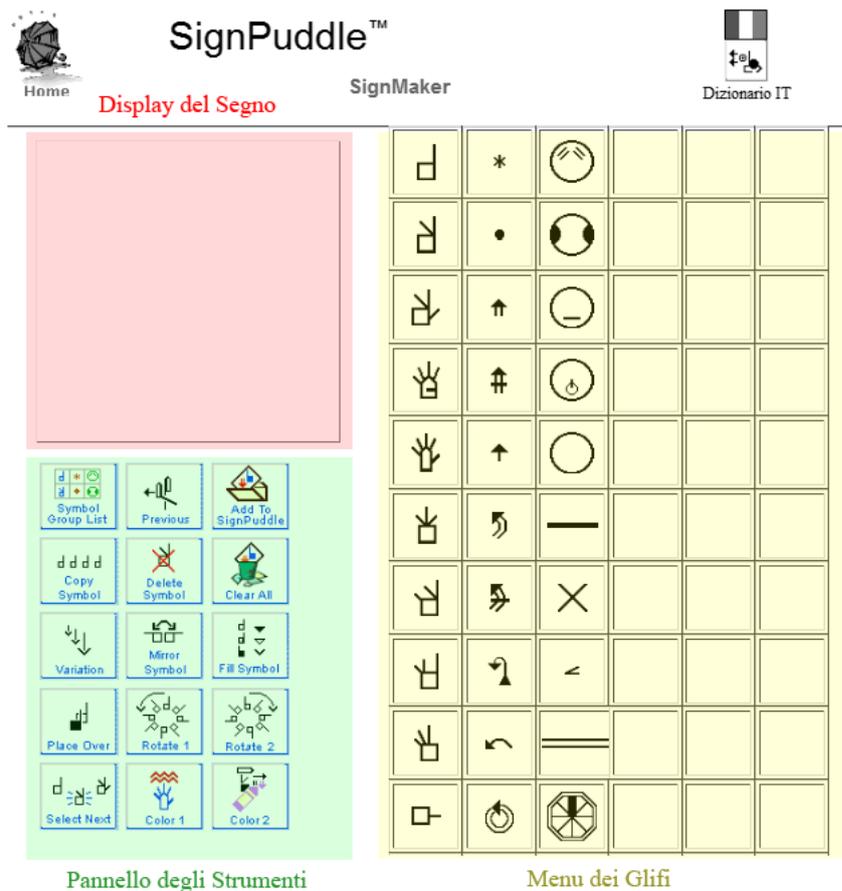


Figura 2.6: Home del SignMaker, divisa in tre aree di interesse.

3.1 Display del Segno

L'utilità del Display del Segno è quella di mostrare all'utente in tempo reale il segno che sta componendo: esso consiste in un riquadro, che possiamo immaginare come una lavagna bianca, sul quale vengono trascinati i glifi presi dal Menu dei Glifi. Una volta spostati all'interno del riquadro, essi rimangono trascinati, sebbene solo all'interno del riquadro stesso e diventano selezionabili (è possibile selezionarne esclusivamente uno per volta); un glifo selezionato può essere modificato utilizzando le funzioni offerte dal Pannello degli Strumenti.

Il Display del Segno svolge in modo sufficiente la sua funzione: la scelta dei glifi trascinati a piacimento (rispetto a glifi posizionabili in aree predefinite) rende più naturale l'interazione, e lascia molta libertà all'utente. Il Display del Segno tuttavia ha alcuni difetti: i glifi, come già notato in precedenza, sono selezionabili esclusivamente uno per volta, rendendo impossibile la modifica di più glifi contemporaneamente e costringendo l'utente ad una lunga e tediosa serie di modifiche glifo per glifo. Un'altra pecca consiste nel fatto che le funzioni per la gestione del Display del Segno si trovano nel Pannello degli Strumenti, ben lontane da dove ci si aspetterebbe di trovarle e mimetizzate tra le altre funzioni.

3.2 Menu dei Glifi

L'utilità del Menu dei glifi è di permettere all'utente di navigare tra le decine di migliaia di glifi del SignWriting alla ricerca di quello desiderato.

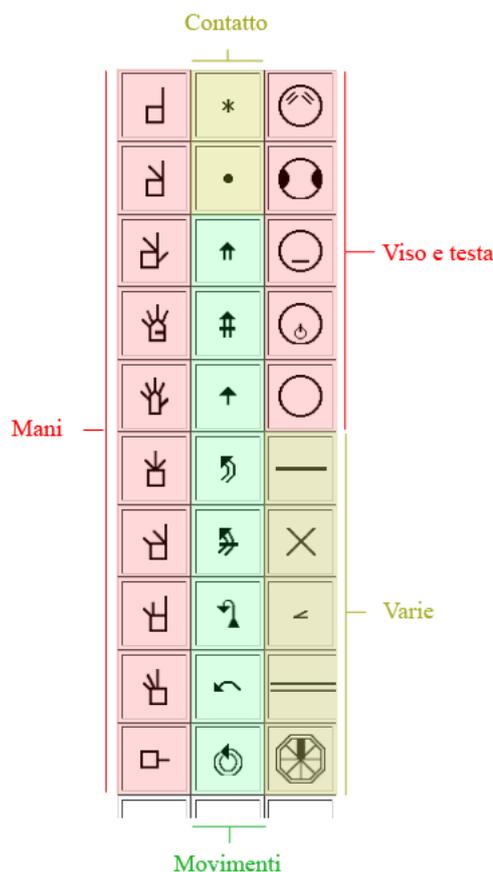


Figura 2.7: Il Menu dei Glifi, diviso secondo il tipo dei glifi.

Il Menu dei Glifi è implementato come un albero di menu: partendo dalla “radice” (la home: l’unico menu a partire dal quale è possibile raggiungere tutti gli altri) l’utente naviga, man mano scegliendo il tipo di glifo che gli interessa, fino ad arrivare ad un menu “foglia”, che contiene un insieme di glifi (l’insieme ha un numero di elementi dell’ordine di grandezza delle decine) tra cui scegliere quello desiderato. Ciascun glifo contenuto in qualsiasi menu (quindi non solo i menu “foglia”) è trascinabile sul Display del Segno. Il SignMaker offre inoltre le funzioni di navigazione di base: un pulsante per tornare al menu precedente ed uno per tornare alla home del Menu dei Glifi. Navigare nel Menu dei Glifi è l’attività alla quale l’utente dedica la maggior parte del suo tempo durante una sessione di utilizzo del SignMaker: scegliere tra decine di migliaia di glifi non è un compito nè facile, nè veloce. Il Menu dei Glifi, così come è stato progettato, non aiuta affatto: come per il Display del

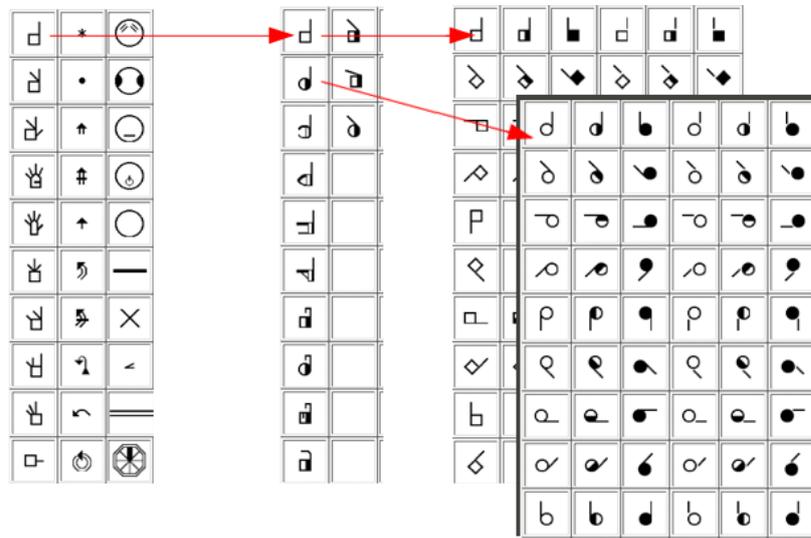


Figura 2.8: Il Menu dei Glifi del SignMaker: la sua ad albero e' chiara in questa rappresentazione.

Segno, le funzioni per la gestione (in questo caso si tratta di funzioni di navigazione) del Menu dei Glifi si trovano nel Pannello degli Strumenti, anche in questo caso ben lontane da dove ci si aspetterebbe di trovarle e mimetizzate tra le altre funzioni. Il problema più grave, tuttavia, responsabile della maggior parte dei ritardi nella composizione di un segno, è l'organizzazione della navigazione tra i menu: i tipi di glifi tra cui scegliere sono presentati l'uno vicino all'altro, senza alcuna distinzione, delegando alla conoscenza del SignWriting dell'utente la scelta (e inevitabilmente la velocità della scelta) del percorso corretto dalla "home" del Menu dei Glifi al glifo desiderato. Mancano completamente i riferimenti grafici, a parte i glifi stessi, per guidare un utente che abbia una conoscenza del SignWriting medio-bassa (e che quindi potrebbe non conoscere alcuni glifi, o alcuni tipi di glifi) nella scelta: niente che distingua, partendo dalla home, i glifi di movimento dalla punteggiatura o dalle configurazioni delle mani, o della testa.

3.3 Pannello degli Strumenti

L'utilità del Pannello degli Strumenti è di fornire all'utente funzioni di gestione dei glifi e del Display del Segno, funzioni di navigazione e funzioni di salvataggio del lavoro svolto.

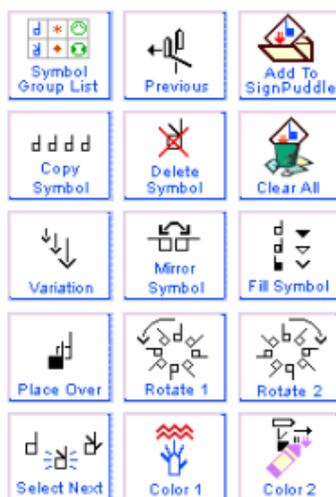
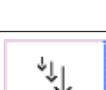
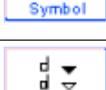
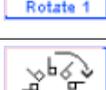
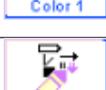


Figura 2.9: Il Pannello degli Strumenti del SignMaker.

La seguente tabella illustra i pulsanti presenti nel Pannello degli Strumenti e ne descrive la funzione associata.

 <p>Symbol Group List</p>	<p>Questo pulsante consente all'utente di tornare alla home del Menu dei Glifi, che può essere visto come un albero di sottomenu. L'utente effettuerà quest'operazione tipicamente quando, una volta scelto un glifo di cui ha bisogno, dovrà cominciare la ricerca del successivo.</p>
 <p>Previous</p>	<p>Questo pulsante consente all'utente di tornare al menu precedente del Menu dei Glifi. L'utente userà questa funzione per navigare tra i numerosi sottomenu alla ricerca del glifo desiderato.</p>
 <p>Add To SignPuddle</p>	<p>Questo pulsante consente all'utente di salvare il segno composto, inserendo alcune informazioni di corredo, sul database dei segni del SignMaker.</p>

	Questo pulsante consente all'utente di eliminare tutto il contenuto del Display del Segno. Non viene chiesta conferma dell'operazione e l'operazione non è reversibile.
	Questo pulsante consente all'utente di eliminare il glifo selezionato nel Display del Segno.
	Questo pulsante consente all'utente di duplicare il glifo selezionato nel Display del Segno. Il glifo duplicato viene posto direttamente nel Display del Segno, le sue coordinate differiscono di qualche pixel da quelle del glifo originale.
	Questo pulsante consente all'utente di applicare alcune variazioni al glifo selezionato nel Display del Segno. Il comportamento di questa funzione varia a seconda del tipo di glifo selezionato.
	Questo pulsante consente all'utente di trasformare il glifo selezionato nel Display del Segno nel suo simmetrico rispetto all'asse verticale .
	Questo pulsante consente all'utente di variare l'“orientamento” del glifo selezionato nel Display del Segno. Il comportamento di questa funzione varia a seconda del tipo di glifo selezionato.
	Questo pulsante consente all'utente di portare in primo piano il glifo selezionato nel Display del Segno.
	Questo pulsante consente all'utente di ruotare in senso antiorario il glifo selezionato nel Display del Segno. Il comportamento di questa funzione varia a seconda del tipo di glifo selezionato.
	Questo pulsante consente all'utente di ruotare in senso orario il glifo selezionato nel Display del Segno. Il comportamento di questa funzione varia a seconda del tipo di glifo selezionato.
	Questo pulsante consente all'utente di selezionare il prossimo glifo presente nel Display del Segno.
	Questo pulsante consente all'utente di applicare una colorazione predefinita ai glifi del SignMaker in modo da identificare cromaticamente i diversi tipi di glifi presenti.
	Questo pulsante consente all'utente di applicare una colorazione personalizzata al glifo selezionato nel Display del Segno.

Tab. 2.2: Analisi dei pulsanti del SignMaker.

Per un'applicazione come il SignMaker è inevitabile prevedere dei pulsanti per fornire all'utente le funzioni di cui ha bisogno: da quelle di navigazione a quelle di gestione del contenuto. Purtroppo, a causa di alcune mancanze nella progettazione dell'interfaccia, il Pannello degli Strumenti è l'area in cui

sono stati riscontrati i maggiori difetti, relativi in particolare alla collocazione dei pulsanti e al loro contenuto. La collocazione dei pulsanti è fuorviante: pulsanti di navigazione nel Menu dei Glifi affiancano e si confondono con quelli di gestione dei glifi selezionati (la maggior parte) e con quelli di gestione del Display del Segno, formando un unico blocco sul quale l'utente nuovo si orienta con molta fatica. Alcuni pulsanti (almeno due) potevano essere evitati: prendiamo ad esempio il pulsante che permette di selezionare il prossimo glifo presente nel Display del Segno: questa funzione diventa inutile, se si prevede un buon meccanismo di selezione nel Display del Segno. Discutibile quindi la scelta di riunire tutti i pulsanti nello stesso posto, allontanandoli dalle aree di interesse a cui le loro funzioni sono applicate. Il problema del collocamento è reso ancor più grave dal contenuto presente sui pulsanti: è una buona regola creare icone che rappresentino da sole, senza bisogno di testo, la funzione offerta. Nel caso del SignMaker questa regola non può essere trascurata, avendo a che fare con utenti che sono abituati ad esprimersi usando una lingua dei segni, non una lingua vocale. Inoltre il testo presente sui pulsanti è in inglese, senza alcuna possibilità di scegliere un'altra lingua; non è raro trovare (nonostante viviamo nel 21esimo secolo, nel cosiddetto Mondo Occidentale, con una predominanza linguistica dell'inglese molto marcata) persone udenti che conoscono solo la loro lingua vocale, dunque è quantomeno azzardato pretendere da un sordo, come requisito base per usare il SignMaker, che conosca, oltre alla sua lingua dei segni e alla lingua vocale del suo paese (con cui inevitabilmente sarà venuto a contatto), anche un'altra lingua vocale. L'espressività delle icone è complessivamente buona, alcune rendono un'idea abbastanza fedele della funzione a cui sono associate, altre meno, ma tutte sono decifrabili esclusivamente da una piccola categoria di persone: segnanti e con conoscenze di SignWriting, restringendo quindi la cerchia dei potenziali utenti del SignMaker.

Con lo sviluppo del SignMaker, e quindi con la possibilità di salvare il segno composto in vari formati tra cui il formato testo, è stata necessaria la definizione del SignWriting Markup Language (SWML). Il SWML, come si legge su <http://www.signwriting.org> è un formato basato sull'XML sviluppato dal team Sutton per la conservazione e l'utilizzo di testi e dizionari in SignWriting, e consente l'interoperabilità delle applicazioni basate su SW e l'accessibilità al Web, per i sordi, nella loro lingua madre.

Capitolo 3

Sordità e Information Technology

1 Situazione attuale

“Membri della comunità dei sordi di solito si trovano a confrontarsi con siti web non scritti nella loro lingua madre, dove possono emergere barriere di accessibilità.”

Per un sordo che voglia avventurarsi nel mondo digitale, le barriere da superare sono altrettanto numerose e difficili da superare quanto quelle che si trova davanti nella vita di tutti i giorni. Stando alle stime della World Health Organization, 278 milioni di persone nel mondo sono sorde o hanno difficoltà uditive (World Health Organization, 2006) e molte di loro usano la lingua dei segni come lingua madre. “Le variazioni regionali della lingua dei segni costituiscono un insieme di minoranze linguistiche relativamente sottorappresentate nel mondo digitale. Pertanto, i membri della comunità dei sordi di solito si trovano a confrontarsi con siti web non scritti nella loro lingua madre, dove possono emergere barriere di accessibilità. Per assicurare l’integrazione sociale della comunità del sordo, le lingue dei segni dovrebbero essere propriamente incorporate all’interno dell’Information Technology (IT)” (Fajardo *et al.*, 2009). L’assenza di linee guida complete, accurate e universalmente riconosciute non contribuisce al superamento di questo gap digitale tra sordi e udenti. Finora, gli unici tentativi degni di nota sono stati fatti dal World Wide Web Consortium (W3C)¹, con il documento Web Content Accessibility Guidelines (WCAG). Il WCAG è un insieme di linee guida che spiegano come rendere i contenuti Web accessibili a persone con disabilità. Le linee guida del WCAG sono destinate a tutti gli sviluppatori di contenuti Web, (autori

¹Le informazioni sul W3C e il WCAG riportate in questo capitolo sono state prese da <http://www.w3.org/>

di pagine e designer di siti) e per sviluppatori di strumenti di authoring. Il loro obiettivo primario è di promuovere l'accessibilità. In ogni caso, rispettarle contribuisce a rendere anche i contenuti Web più accessibili a tutti gli utenti. La prima versione, risalente al 1999, risolve alcuni problemi legati al gap digitale degli utenti sordi. In particolare, come si riporta in Fajardo *et al.* (2009), la direttiva 1.4 raccomanda che l'audio e la sua trascrizione testuale siano sincronizzati, la direttiva 14 indica che gli utenti sordi traggono benefici da un stile di linguaggio semplice e scritto in modo chiaro. È facile notare che le raccomandazioni del W3C sono abbastanza vaghe, ma, cosa ben più importante, le direttive del WCAG1.0 trattano esclusivamente problemi relativi all'etichettatura e alla trascrizione di contenuto audio, tralasciando le caratteristiche visuali-spaziali delle lingue dei segni, causando quindi perdita di informazioni. Nel 2008 il W3C ha prodotto il WCAG2.0, dove i problemi delle lingue dei segni nel mondo digitale sono trattati più accuratamente. In particolare, la direttiva 1.2 è una di quelle la cui osservanza aumenta la percettibilità del contenuto web da parte degli utenti, inclusi i sordi. Ad esempio, il criterio di successo 1.2.6 stabilisce che "L'interpretazione tramite lingua dei segni è fornita per tutti i contenuti audio preregistrati sotto forma di tipi di media sincronizzati". Soddisfare il criterio 1.2.6 è necessario per ottenere il più impegnativo livello di conformità: il criterio di successo AAA. Al criterio sono allegate istruzioni dettagliate sulle tecniche da adottare. In ogni caso, i criteri di successo si concentrano solo sulla trascrizione di contenuto audio. Niente è menzionato su come dev'essere prodotto il contenuto dei link, che per la maggior parte delle volte è testuale e non uditivo, o della trascrizione (Fajardo *et al.*, 2009).

Tenendo conto del problema appena illustrato, e trovandomi a dover scegliere concretamente come progettare l'interfaccia del SWift (nello specifico come costruire pulsanti, icone e hyperlink) ho deciso di documentarmi sui passi compiuti dalla ricerca nell'ambito dell'accessibilità, per i sordi, ai contenuti Web e ho trovato forse più di quello che cercavo. Leggendo un illuminante articolo, già citato, di Fajardo *et al.* (2009) ho trovato alcune importanti informazioni su progetti di ricerca sull'argomento, che riporterò brevemente, e da cui ho tratto alcune euristiche per sapere come procedere con il mio lavoro, che si colloca senza ombra di dubbio in questo contesto, trattandosi di una web-application orientata agli utenti del SW, che è facile immaginare saranno prevalentemente sordi.

2 Information Scent

Prima di procedere è opportuno richiamare un concetto: l'Information Scent (in italiano sunerebbe come "Odore dell'Informazione" o "Traccia dell'Informazione"), termine coniato per la prima volta da Peter Pirolli e da altri

ricercatori della Xerox al Palo Alto Research Centre (PARC) all'inizio degli anni '90. Le loro ricerche rivelano una forte somiglianza tra il modo in cui gli umani cercano informazioni e il modo in cui gli animali cacciano la loro preda (Pirolli, Card, 1999).

L'Information Scent descrive come le persone valutano le opzioni quando stanno cercando informazioni su un sito. Quando gli viene presentata una serie di scelte, l'utente sceglierà quella che gli da l'indicazione più chiara (in altre parole che abbia l'"odore" più forte) per portarlo più vicino possibile all'informazione richiesta. I siti che hanno un forte Information Scent riescono a guidare bene l'utente verso il contenuto richiesto, al contrario, siti con debole Information Scent costringono l'utente a impiegare tempi di valutazione più lunghi, aumentano il rischio che l'utente scelga l'opzione sbagliata e lo costringono a utilizzare molto il bottone back.

Il modo più facile per capire l'Information Scent è all'interno di un esempio: si immagina che le seguenti siano voci di un menu di navigazione di un sito di uno store on-line.



Figura 3.1: Esempio di Information Scent debole.

Se un utente visitasse il sito cercando informazioni su un iPod, quale opzione sceglierebbe?

Tutte le voci di navigazione sono istantaneamente comprensibili, ma quando si considera il task dell'utente ci si accorge che c'è una sovrapposizione tra due sezioni. Un iPod potrebbe facilmente trovarsi sia nella sezione **Audio and TV** che nella sezione **Computing**. Quale opzione si dovrebbe scegliere?

Questo è un esempio perfetto del caso in cui l'utente probabilmente effettua una scelta sbagliata, e frustrato, non trovando l'informazione richiesta, usa il bottone indietro o lascia il sito, pensando che non la contenga. I principi su come creare Information Scent più forti sono abbastanza semplici: fornire all'utente più contenuto rende più facile la scelta dell'opzione migliore. L'esempio seguente illustra bene l'idea:

Nella lista in Fig. 3.2, qual'è la voce che un utente sceglierebbe cercando un iPod? Nonostante l'iPod non sia specificamente menzionato vicino a nessuna delle sezioni, la presenza dell'opzione **Hi-Fi** vicino ad **Audio e TV** e l'assenza di parole simili a iPod nella voce **Computing**, crea un Information Scent più forte intorno alla voce **Audio e TV**.



Audio and TV: Camcorders, DVD and Video, Hi-Fi...
Books: Bestsellers, Factual, Education...
Computing: Computers, Games, Printers
Fashion: Mens, Womens, Kids...
Furniture: Bathrooms, Bedrooms, Kitchen...
Gardening: Seeds, Plants, Pots

Figura 3.2: Esempio di Information Scent più forte.

Concludendo, la forza dell'Information Scent in un sito (o, aggiungo io, in un'applicazione che abbia molte informazioni da presentare) ha un effetto significativo sull'abilità degli utenti di trovare l'informazione che vogliono e quindi di proseguire nell'utilizzo del sito (o dell'applicazione).

3 Hyperlinking per le lingue dei segni

Un volta chiarito cosa si intende per Information Scent, è facile intuire quanto un sordo possa trovarsi in difficoltà nel mondo digitale. Seguendo la “traccia”, l’“odore” dell'informazione, ciascun utente utilizza, ovviamente, tracce semantiche per giudicare quale voce o quale sito scegliere. Il problema, nel caso dei sordi, è che queste tracce semantiche sono generalmente disponibili (tipicamente il testo dei link) in una lingua che non è la loro lingua madre, e nella quale spesso presentano bassi livelli di competenza nella lettura. Di conseguenza, come alcuni studi dimostrano (Fajardo *et al.*, 2009), utenti sordi trovano difficile utilizzare una strategia per seguire l'Information Scent seguendo tracce testuali. Apparentemente, una facile soluzione è l'utilizzo di hyperlink grafici, le icone, poichè facilitano il processo di scelta semantica. Inoltre, in Namatame *et al.* (2007) si osserva che i partecipanti sordi sono più accurati dei partecipanti udenti in task di ricerca e matching utilizzando icone. Nonostante ciò, sia sordi che udenti mostrano lo stesso grado di precisione (Fajardo *et al.*, 2008a) quando vengono utilizzate come hyperlink icone frequenti e familiari.

Alcuni progetti di ricerca hanno sviluppato una serie di tecniche più o meno sofisticate per consentire non solo l'accesso alle informazioni, ma anche la possibilità di passare da un link all'altro per utenti sordi. Partendo da un'idea apparentemente semplice, il progetto Cogniweb (Fajardo *et al.*, 2008b) ha sviluppato due alternative costituite da video incorporati in piccoli frame che contengono una traduzione in lingua dei segni (eseguita da un segnante umano) di ciascun hyperlink testuale nel menu. Questo meccanismo è stato chiamato Sign Language Scent (SLS). Nel primo approccio, quando il cursore passa sopra un link, il video, posizionato sul fondo della pagina, comincia la

sequenza segnata in modo che l'utente possa cogliere il significato del collegamento. Nel secondo approccio, ogni link è legato al suo video che comincia la riproduzione ogni volta che l'utente fa clic su di esso. In due esperimenti (Fajardo et al. 2008B, c), si è osservato che gli utenti sordi si sono dimostrati più efficienti con la seconda versione. Inoltre, gli utenti sordi sono rimasti meno disorientati (utilizzando come misura il numero di pagine visitate per ogni ricerca) e sono stati meno legati alla loro capacità di ragionamento categorico-verbale utilizzando SLS piuttosto che utilizzando solo hyperlinks testuali. Altri approcci più sofisticati sono mirati a creare siti esclusivamente in lingua dei segni, in modo da creare contenuti e possibilità di navigazione in questa lingua. Questi approcci sono basati sulla tecnologia Hypervideo che consiste in link inseriti in un video in modo che l'utente possa recuperare ulteriori informazioni su concetti veicolati in alcune sequenze filmate. I link sono rappresentati attraverso testo o una immagine statica, che purtroppo potrebbe non trasmettere concetti espressi in lingua dei segni in un modo adeguato. Sulla base di questa tecnologia, Fels et al. (2006) hanno sviluppato il sistema Signlinking. Ogni Signlink è un periodo di tempo entro il video clip, definito dall'autore. Quando il video raggiunge un Signlink, un indicatore di collegamento viene visualizzato ad informare l'utente. Poi, l'utente può scegliere se seguire il link o continuare con la riproduzione del contenuto. Un approccio simile è seguito in un sistema progettato da Kaibel *et al.* (2006). In questo sistema, l'utente può ottenere una panoramica degli hyperlink video facendo passare il mouse su ogni snapshot del filmato. Inoltre, vi è una mano o un simbolo di testo in basso a sinistra di ogni collegamento video indicando a quale tipo di documento conduce l'hyperlink. Un video contenente un simbolo di mano conduce ad una pagina in lingua dei segni mentre un video che contiene un testo-simbolo porterà a una tradizionale documento testuale. Non è chiaro, si conclude in Fajardo *et al.* (2009), quale di questi due approcci per la navigazione in lingua dei segni sia più efficace. La questione potrà essere chiarita, come spesso succede in questi casi, esclusivamente con la sperimentazione.

Per quanto riguarda il SWift, ho adottato una soluzione, come si vedrà in seguito (Cap. 5 e 6) molto più simile a SLS che a Hypervideo. Ciascun pulsante dell'interfaccia ha un'icona (mi sono preoccupato di sceglierle tra le icone più familiari possibili), senza testo, ed è collegato ad una sequenza animata (animata, non segnata, come in SLS) che si attiva al passaggio del mouse. Tale sequenza, in pochi secondi, animando l'icona del pulsante, rende l'idea della funzione associata. La scelta di utilizzare un'animazione piuttosto che da un filmato con una sequenza segnata è stata dettata dal fatto che il costo (in termini di spazio e quindi di tempo di caricamento) delle sequenze segnate è piuttosto elevato rispetto al costo di un filmato Flash, inoltre le funzioni associate ai pulsanti sono piuttosto facili da "riassumere" utilizzando solamente

icone e animazioni (basti pensare alle funzioni “salva”, “ruota”, “elimina”).

Capitolo 4

Progettazione del *SWift*

1 Identificazione dei Requisiti

In questo capitolo si espliciteranno i requisiti dell'applicazione: essi non sono altro che elenchi di servizi che il sistema deve offrire, indicano come dovrebbe reagire a particolari input e come dovrebbe comportarsi in particolari situazioni. In alcuni casi possono affermare esplicitamente cosa il sistema non dovrebbe fare. Alcuni tipi di requisiti possono essere vincoli sui servizi o sulle funzioni offerte dal sistema, possono includere vincoli temporanei, sul processo di sviluppo e sugli standard. Solitamente si applicano al sistema completo e non a singole funzioni o servizi. Possono riferirsi a proprietà del sistema come affidabilità, tempi di risposta e occupazione di spazio, ma possono anche definire vincoli come la capacità dei dispositivi di I/O e la rappresentazione dei dati utilizzata nelle interfacce del sistema. Le specifiche dei requisiti fornite in questo capitolo sono complete e coerenti: **complete** indica che tutti i servizi richiesti dall'utente devono essere definiti, **coerenti** significa che i requisiti non devono essere contraddittori.

Definizione Requisiti:

- **RU.F**: Requisiti Utente Funzionali;
- **RU.N**: Requisiti Utente non Funzionali;

Definizione tabella.

ID	TITOLO
Descrizione	Corpo descrizione.
Motivazione	Corpo motivazione.
Priorità	Corpo priorità.
Specifiche	Lista ID di tipo RS.

1.1 Requisiti Utente funzionali

In questa parte saranno elencati in forma tabellare i requisiti utente funzionali.

RU.F.01	COMPOSIZIONE DEL SEGNO
Descrizione	L'applicazione deve permettere all'utente di comporre segni (espressi mediante il Sign Writing) utilizzando uno o più glifi.
Motivazione	La composizione di segni espressi mediante il Sign Writing è lo scopo principale dell'applicazione.
Priorità	MustHave.
Specifiche	<ul style="list-style-type: none"> • RS.F.01.

RU.F.02	INTERFACCIA GRAFICA INTELLIGENTE
Descrizione	L'applicazione deve avere un'interfaccia che velocizzi e snellisca il processo di scelta del glifo, e conseguentemente il processo di composizione del segno.
Motivazione	Rendere veloce e (per quanto possibile) confortevole il processo di composizione del segno.
Priorità	MustHave.
Specifiche	<ul style="list-style-type: none"> • RS.F.02. • RS.F.03. • RS.F.04. • RS.F.05.

RU.F.03	SALVATAGGIO DEL SEGNO
Descrizione	L'applicazione deve permettere all'utente di salvare il segno composto.
Motivazione	Conservare contenuti prodotti con il <i>SWift</i> e riutilizzarli sia all'interno del <i>SWift</i> stesso, sia in altre applicazioni.
Priorità	MustHave.
Specifiche	<ul style="list-style-type: none"> • RS.F.06. • RS.F.07.

RU.F.04	GLIFI A MANO LIBERA
Descrizione	L'applicazione deve permettere all'utente di inserire, qualora non esistesse il glifo desiderato tra quelli disponibili, un glifo disegnato a mano libera.
Motivazione	Offrire all'utente la possibilità di esprimersi senza limitazioni, e di contribuire a rendere più completa l'applicazione, laddove ce ne fosse bisogno.
Priorità	ShouldHave.
Specifiche	<ul style="list-style-type: none"> • RS.F.08.

RU.F.05	COMPLETAMENTO AUTOMATICO
Descrizione	L'applicazione deve offrire all'utente una funzione di completamento automatico del segno.
Motivazione	Rendere più veloce l'inserimento di segni di frequente utilizzo.
Priorità	ShouldHave.
Specifiche	<ul style="list-style-type: none"> • RS.F.09.

1.2 Requisiti Utente non funzionali

In questa parte saranno elencati in forma tabellare i requisiti utente non funzionali.

RU.N.01	PORTABILITA'
Descrizione	L'applicazione deve essere portabile almeno su piattaforme Windows, Linux, Macintosh.
Motivazione	Rendere il software fruibile da diversi sistemi operativi per facilitarne la diffusione.
Priorità	MustHave.
Specifiche	<ul style="list-style-type: none">• RS.N.01.

2 Requisiti di Sistema

Definizione Requisiti:

- **RS.F**: Requisiti di Sistema Funzionali;
- **RS.N**: Requisiti di Sistema non Funzionali

DESCRIZIONE TABELLA	
ID	TITOLO
Descrizione	Corpo descrizione.
Motivazione	Corpo motivazione.
Priorità	Corpo priorità.

2.1 Requisiti di Sistema funzionali

In questa parte saranno elencati in forma tabellare i requisiti di sistema funzionali.

RS.F.01	COMPOSIZIONE DEL SEGNO
Descrizione	<p>L'utente avrà a disposizione quattro risorse che lo aiuteranno a comporre segni espressi mediante il Sign Writing.</p> <ul style="list-style-type: none"> • Il <i>Menu dei Glifi</i>: un pannello di ricerca e navigazione che gli permetterà di orientarsi negli oltre 50000 glifi del Sign Writing e di scegliere i glifi necessari alla composizione del segno. • Il <i>Display del Segno</i>: un'area (la si può immaginare come una rettangolo bianco) sulla quale l'utente potrà operare sui glifi da lui scelti (nel <i>Menu dei Glifi</i>) in modo da formare il segno desiderato. L'utente potrà comporre su quest'area esattamente un segno alla volta. • Il <i>Pannello degli Strumenti</i>: un contenitore di utili funzioni le quali permetteranno all'utente di operare sui glifi contenuti nel <i>Display del Segno</i>. • Il <i>Pannello di Completamento</i> Pannello che, man mano che il segno viene composto, offre all'utente la possibilità di inserire, direttamente nel Display del Segno, glifi compatibili con ciò che sta componendo.
Motivazione	La composizione di segni espressi mediante il Sign Writing è lo scopo principale dell'applicazione.
Priorità	MustHave.

RS.F.02	<i>MENU DEI GLIFI</i>
Descrizione	Per snellire la ricerca nel <i>Menu dei Glifi</i> , l'interfaccia deve consentire all'utente di navigare scegliendo prima la parte anatomica interessata, poi di definire le caratteristiche del glifo, senza priorità alcuna di una proprietà del glifo rispetto all'altra, in modo da non privilegiare alcun modello mentale (più concretamente il ragionamento che ciascun utente segue mentre sceglie un glifo) rispetto agli altri. L'interfaccia deve includere funzioni di navigazione all'interno del <i>Menu dei Glifi</i> (sarà necessario includere, tra le altre, una funzione per tornare alla home del <i>Menu dei Glifi</i>). L'interfaccia deve includere un meccanismo di drag e drop per trasportare agevolmente il glifo desiderato dal <i>Menu dei Glifi</i> al <i>Display del Segno</i> .
Motivazione	Rendere veloce e (per quanto possibile) confortevole il processo di composizione del segno.
Priorità	MustHave.

RS.F.03	<i>DISPLAY DEL SEGNO</i>
Descrizione	Il <i>Display del Segno</i> deve essere il luogo dell'interfaccia in cui l'utente può vedere e gestire il segno da lui composto; i singoli glifi, all'interno di quest'area, devono diventare unità selezionabili (singolarmente o in gruppi) e spostabili a piacimento, con il solo utilizzo del mouse. L'utente potrà comporre su quest'area esattamente un segno alla volta. Il <i>Display del Segno</i> deve comprendere anche le funzioni per il salvataggio del lavoro e per l'eliminazione del contenuto composto dall'utente.
Motivazione	Rendere veloce e (per quanto possibile) confortevole il processo di composizione del segno.
Priorità	MustHave.

RS.F.04	<i>PANNELLO DEGLI STRUMENTI</i>
Descrizione	<p>Il <i>Pannello degli strumenti</i> deve includere numerose funzioni di modifica del glifo/dei glifi selezionato/i, in modo da evitare, dove possibile, dispendiose ricerche nel <i>Menu de Glifi</i> (per quanto l'intento del mio lavoro sia di velocizzare tali ricerche, esse richiederanno comunque più tempo di un semplice click su un bottone del <i>Pannello degli strumenti</i>). Tra le funzioni da sviluppare ci sono le seguenti (quelle con priorità MustHave saranno necessariamente sviluppate):</p> <ul style="list-style-type: none"> • Eliminazione del glifo/dei glifi selezionato/i. - MustHave • Duplicazione del glifo/dei glifi selezionato/i. - MustHave • Rotazione (senso orario) del glifo/dei glifi selezionato/i. - MustHave • Rotazione (senso anti-orario) del glifo/dei glifi selezionato/i. - MustHave • Variazione del riempimento del glifo/dei glifi selezionato/i. - MustHave • Simmetrico rispetto all'asse Y del glifo/dei glifi selezionato/i. - MustHave • Colorazione del glifo/dei glifi selezionato/i. - ShouldHave • Inserimento di un glifo a mano libera. - ShouldHave
Motivazione	Rendere veloce e (per quanto possibile) confortevole il processo di composizione del segno.
Priorità	MustHave.
RS.F.05	<i>PANNELLO DI COMPLETAMENTO</i>
Descrizione	Paragonabile al software T9 per la digitazione di stringhe alfanumeriche per i cellulari, questo pannello dovrà avere il compito di velocizzare l'inserimento dei segni di uso frequente offrendo la possibilità di inserire glifi (scelti in base alla frequenza di utilizzo) compatibili con quello che l'utente sta componendo, direttamente nel suo <i>Display dei Segni</i> .
Motivazione	Rendere veloce e (per quanto possibile) confortevole il processo di composizione del segno.
Priorità	MustHave.

RS.F.06	FORMATI DI SALVATAGGIO
Descrizione	<p>L'applicazione deve fornire all'utente la possibilità di salvare il suo lavoro in diversi formati:</p> <ul style="list-style-type: none"> • Formato immagine (salvataggio locale): l'applicazione deve restituire all'utente uno snapshot del <i>Display del Segno</i> in formato immagine (.png). Tale formato non può essere riutilizzato dal <i>SWift</i>, ma può dare una raffigurazione immediata del segno composto, ed essere incluso (in quanto immagine) in numerose altre applicazioni. • Formato testo (salvataggio locale): l'applicazione deve restituire all'utente un documento XML. Tale formato può essere riutilizzato dal <i>SWift</i>, ma, non essendo un'immagine, ha una minore potenza raffigurativa. • Salvataggio sul DB (salvataggio remoto): l'applicazione deve dare all'utente la possibilità di salvare il segno composto sul DB dei segni del <i>SWift</i>, indicando titolo, note, ed altre informazioni necessarie al suo riutilizzo.
Motivazione	Conservare contenuti prodotti con il <i>SWift</i> e riutilizzarli sia all'interno del <i>SWift</i> stesso, sia in altre applicazioni.
Priorità	MustHave.

RS.F.07	SALVATAGGIO ASSISTITO
Descrizione	L'interfaccia di salvataggio deve prevedere un meccanismo che salvi il segno composto inserendo, nel nome del file, una stringa (decisa dall'utente) e un numero incrementale (Ad esempio, come nel caso della trascrizione della celebre "Pear Story", sarà opportuno che i nomi dei file salvati siano di questa forma: "Pera-0001", "Pera-0002", ecc.).
Motivazione	Velocizzare il lavoro nel caso in cui sia stia trascrivendo/scrivendo un testo composto da numerosi segni.
Priorità	ShouldHave.

RS.F.08	GLIFI A MANO LIBERA
Descrizione	L'applicazione deve prevedere una modalità di inserimento dei glifi a mano libera, qualora il glifo desiderato non fosse presente, o nel caso in cui si sentisse il bisogno di inserire un glifo nuovo, compatibile (per quanto possibile) con il Sign Writing (si tratta comunque di casi molto rari). Il glifo dovrà essere salvato sul DB dei glifi del <i>SWift</i> , prima di essere utilizzato all'interno del <i>Display del Segno</i> .
Motivazione	Offrire all'utente la possibilità di esprimersi senza limitazioni, e di contribuire a rendere più completa l'applicazione, laddove ce ne fosse bisogno.
Priorità	ShouldHave.

RS.F.09	COMPLETAMENTO AUTOMATICO
Descrizione	L'applicazione deve prevedere un meccanismo di completamento automatico differenziato in base alle aree anatomiche. Il completamento automatico deve attirare l'attenzione dell'utente sotto forma di suggerimenti nel <i>Pannello di Completamento</i> subito dopo l'inserimento di un glifo riconducibile ad un'area anatomica (ad esempio la testa) e proporre una serie di glifi (relativi all'area anatomica di cui prima), compatibili con i glifi inseriti fino a quel momento dall'utente. Tali glifi saranno scelti basandosi sulla loro frequenza di utilizzo.
Motivazione	Rendere più veloce l'inserimento di segni di frequente utilizzo.
Priorità	MustHave.

2.2 Requisiti di Sistema non funzionali

In questa parte elencheremo in forma tabellare i requisiti di sistema non funzionali.

RS.N.1	APPLICAZIONE WEB
Descrizione	L'applicazione deve essere utilizzabile da numerosi sistemi operativi, tra cui Windows, Linux e Macintosh. Ciò è possibile in quanto si tratta di un'applicazione web, testata sui seguenti browser: <ul style="list-style-type: none">• Mozilla Firefox.• Internet Explorer.• Opera.• Google Chrome. Qualunque sia il browser, deve avere javascript abilitato ed avere il plug-in Flash Player installato.
Motivazione	Rendere il software fruibile da diversi sistemi operativi per facilitarne la diffusione.
Priorità	MustHave.

Capitolo 5

Storyboard del SWift

Una storyboard, in ambito cinematografico, è una serie di disegni, in genere diverse centinaia, che illustrano, inquadratura per inquadratura, ciò che verrà girato sul set. Il contenuto di questo capitolo è in qualche modo simile ad una storyboard: mi propongo di ripercorrere, dividendola in fasi della durata di qualche mese, la storia della progettazione del SWift. Ad ogni fase della progettazione è seguita una serie di incontri con Claudia S. Bianchini e con i membri del “Laboratorio Lingua dei Segni Scritta”: un raggruppamento informale all’interno del “Sign Language Laboratory” dell’ISTC-CNR, il gruppo è composto in maggioranza da sordi, e da loro ho ottenuto un prezioso aiuto per verificare la validità delle scelte effettuate e per ottenere indicazioni su come proseguire. Per avere un’idea del SWift, nella sua versione definitiva è opportuno, invece, soffermarsi sui Cap. 6 e 7.

1 Prima fase di progettazione (2009.12 2010.03)

1.1 Interfaccia

“Il programma che dovevo sviluppare era relativamente semplice dal punto di vista logico, ma estremamente articolato dal punto di vista dell’interazione con l’utente.”

In tutte le fasi di progettazione del SWift sono stato guidato da una semplice ma illuminante considerazione: il programma che dovevo sviluppare era relativamente semplice dal punto di vista logico, ma estremamente articolato dal punto di vista dell’interazione con l’utente. La considerazione era stata la naturale sintesi di alcune osservazioni: prima tra tutte la particolare categoria di utenti che il SWift è chiamato a soddisfare, i sordi, che come accennato nel Cap. 1 e nel Cap. 3 hanno un diverso modo di comunicare e scambiare infor-

mazioni rispetto agli udenti, cosa che si ripercuote inevitabilmente anche sui criteri di usabilità utilizzati nella progettazione di un'applicazione destinata a loro, rendendo necessario uno studio approfondito dell'interfaccia e test continui con i sordi per aggiustare di volta in volta la rotta. Non meno importante è stata la constatazione del fatto che il tempo di utilizzo di un editor del SW dipende quasi totalmente dal tempo che l'utente impiega nel comporre un segno (ovvero principalmente nel cercare i glifi necessari), quindi l'unico modo di ridurre il tempo di utilizzo è progettare un'interfaccia che lo aiuti in modo significativo in tale ricerca. L'ultima osservazione riguarda la parte logica che anima il SWift; in realtà tutto ciò che dovevo programmare, se si esclude la parte (più impegnativa) del completamento automatico, sarebbe stata qualche query su un database, ben poca cosa rispetto al monumentale lavoro richiesto dall'interfaccia.

Come si può dedurre dalle osservazioni precedenti, mi sono concentrato subito su quest'ultima, producendo un mockup delle principali sezioni del SWift:

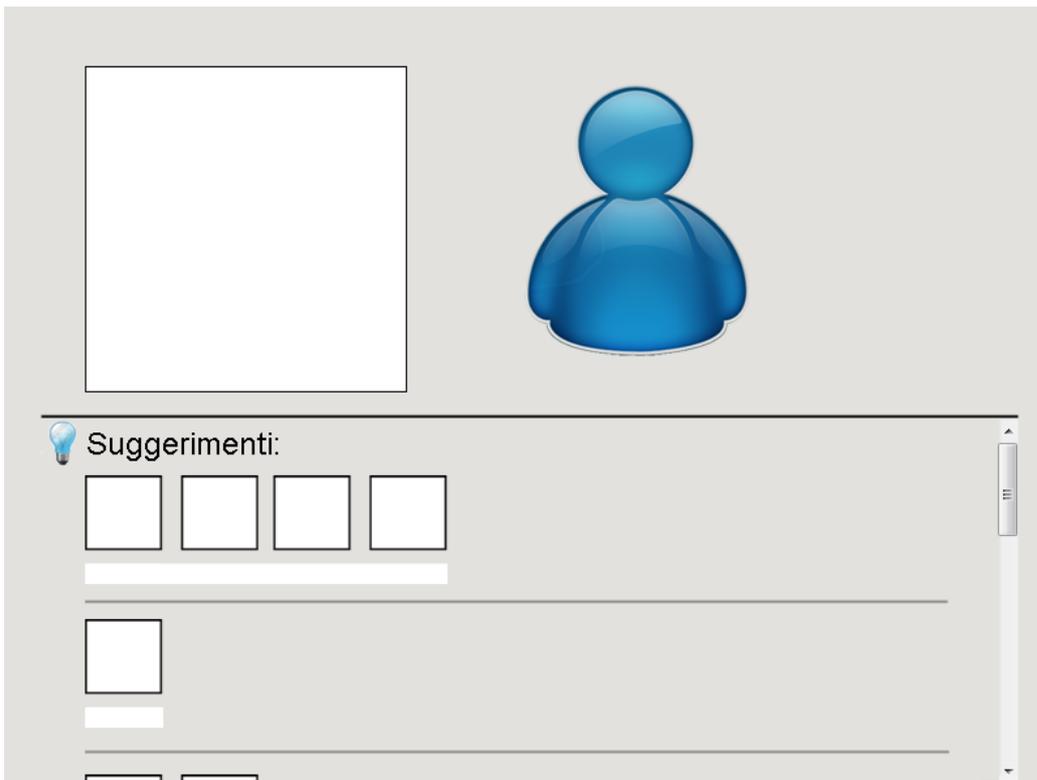


Figura 5.1: Primo mockup dell'interfaccia del SWift (2009.12.26).

Al di là dell'estetica (e del fatto che in questo mockup manca completamente il Pannello degli Strumenti), la Fig. 5.1 ci dà l'occasione di introdurre due importanti idee alla base della progettazione del SWift. La figura umanoide simil-Microsoft, nella sua bluastra paffutaggine, rappresenta una delle più in-

novative caratteristiche del SWift: l'interfaccia anatomica. Essendo la lingua dei segni, e di conseguenza anche il SW, strettamente legata al corpo umano, ho pensato che il modo migliore per aiutare l'utente nella ricerca fosse quello di proporgli una figura umanoide, con aree del corpo cliccabili singolarmente (testa, spalle, braccia, mani, ecc.), ciascuna delle quali collegata ad un sottomenu differente, in modo da effettuare semplicemente una prima importante delimitazione del campo di ricerca. All'interno dei sottomenu l'utente avrebbe potuto navigare e scorrere i glifi fino a trovare quello desiderato, esattamente come accadeva nel SignMaker. Nella parte inferiore dell'immagine si può notare il Pannello di Completamento: [...] Il Pannello di Completamento, così com'era stato concepito all'inizio, doveva fornire, in base ai glifi inseriti, il segno più simile, o un'eventuale sequenza di segni, contenuti nel database del SWift. Nella parte superiore sinistra dell'immagine si può notare il Display del Segno, ovvero l'area sulla quale l'utente compone il suo segno. Il Display del Segno è una delle idee proposte dagli sviluppatori del SignMaker che ho ritenuto valida e quindi ho adottato io stesso. Da notare l'utilizzo di icone contenenti esclusivamente immagini, senza testo, per via delle considerazioni esposte nei Cap. 2 e 3. In questa prima fase di progettazione mi sembrava indispensabile la creazione di due interfacce: una, (quella del mockup) per i principianti del SW, più user-friendly e comprensibile ma meno rapida nell'utilizzo, l'altra, per gli esperti (persone che sanno esattamente dove e come cercare i glifi che gli interessano), con meno aiuti grafici, quindi più veloce, ma meno comprensibile. Solo un mockup (quello dell'interfaccia per principianti) ha visto la luce, la progettazione dell'interfaccia per esperti, ritenuta meno impegnativa, non è stata affrontata in questa fase.

2 Seconda fase di progettazione (2010.04)

2.1 Interfaccia

Il completamento automatico, così com'era stato concepito all'inizio, era di poco aiuto, sarebbe stato molto più efficace avere un completamento automatico differenziato.

Il mockup mostrato nel paragrafo precedente ha rappresentato un punto di partenza nella progettazione dell'interfaccia del SWift, a cui ha fatto seguito, qualche mese dopo, un altro mockup più particolareggiato, a partire dal quale è stata sviluppata l'attuale interfaccia del SWift, coerente con le idee presenti nel primo.

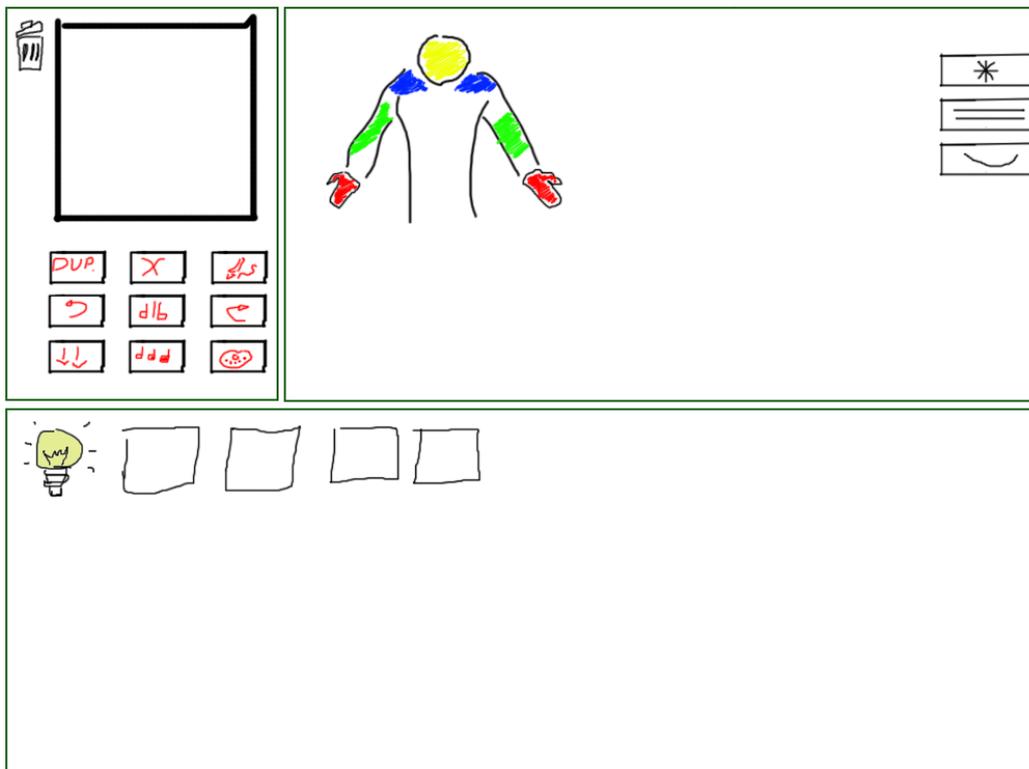


Figura 5.2: Secondo mockup dell'interfaccia del SWift - Home (2010.04.16).

Come si può notare nella parte superiore destra dell'immagine, la paffuta icona dell'utente Microsoft del mockup precedente ha lasciato il posto ad un disegno a mano libera, battezzato "Puppet", (in modo da differenziare totalmente il SWift da forme di appartenenza ad altri enti) sul quale si possono notare alcune aree colorate, che rappresentano le zone cliccabili e portano ad

altrettanti sottomenu che aiutano l'utente a scegliere le caratteristiche del glifo (che variano a seconda della parte anatomica) in modo da circoscrivere ulteriormente il campo di ricerca (si può trovare un esempio dei sottomenu di seguito in questo paragrafo). Analizzando i glifi del SW è emerso che esistono alcune categorie che non sono legate a parti anatomiche, come ad esempio i contatti, la punteggiatura, la dinamica, ecc. quindi ho previsto, nella home, dei pulsanti per accedere ai sottomenu di tali categorie (si trovano all'estrema destra dell'immagine). Nella parte inferiore dell'immagine si può vedere il Pannello di Completamento, il leggero cambio dell'estetica è conseguenza di un radicale cambiamento nella logica. Durante un confronto con il gruppo dell'ISTC-CNR è emerso che il completamento automatico, così com'era stato concepito all'inizio, era di poco aiuto, sarebbe stato molto più efficace avere un completamento automatico differenziato, per fornire suggerimenti su una singola parte anatomica alla volta, piuttosto che su tutte contemporaneamente. Il contenuto del Pannello di Completamento, pertanto, non è più un gruppo di possibili segni interi, bensì un gruppo di possibili "parti di segni" (che più esattamente possiamo definire un insieme di glifi) da inserire nel Display del Segno. La parte superiore sinistra dell'immagine pone l'accento su un altro importante problema: la differenziazione dei pulsanti (che saranno descritti accuratamente in seguito nel prossimo paragrafo): come osservato nel Cap. 2, la scelta di raggruppare pulsanti con funzioni differenti disorienta l'utente ed allunga i tempi di utilizzo di un editor SW, per questo motivo ho deciso di raggruppare i pulsanti che modificano il glifo selezionato nel Display del Segno (eliminazione, rotazione, ecc.) e di collocarli nella parte bassa, separandoli dai pulsanti che intervengono sul segno composto, nella sua interezza, che invece ho posto a sinistra del Display del Segno. Da notare la totale mancanza, in questa schermata dei bottoni di navigazione nel Menu dei Glifi.

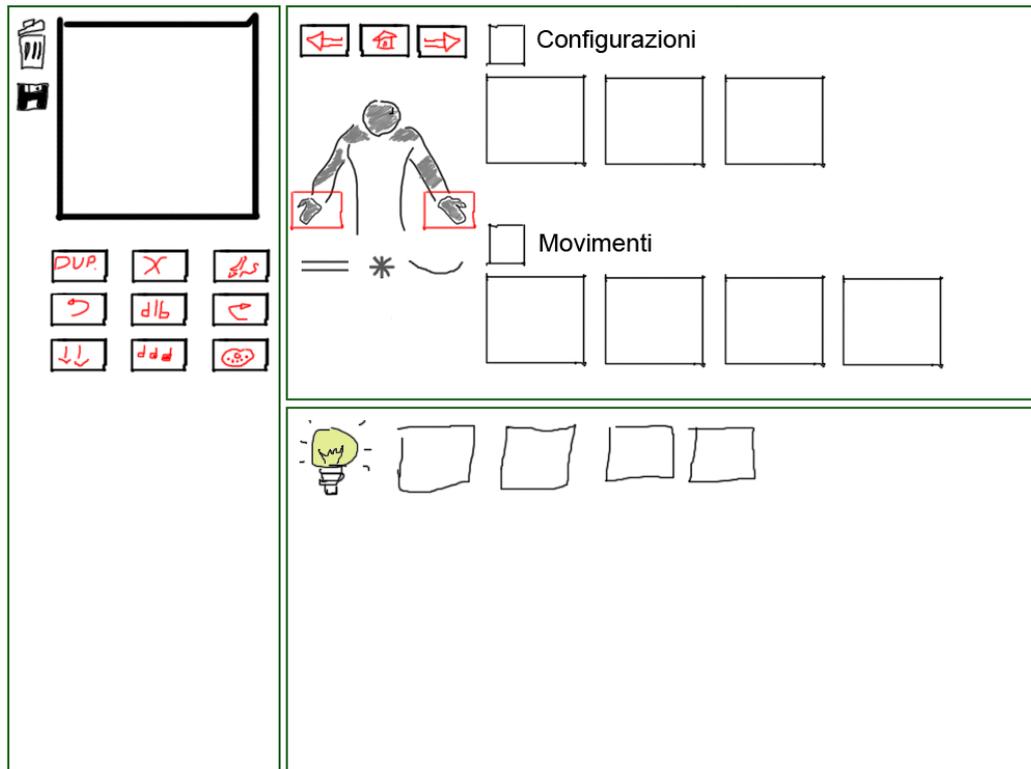


Figura 5.3: Secondo mockup dell'interfaccia del SWift - Sottomenu dell'area delle mani (2010.04.16).

La Fig. 5.3 rappresenta ciò che viene mostrato all'utente quando sceglie una particolare parte anatomica (in questo caso l'area delle mani). Il Pannello di Completamento, il Pannello degli Strumenti e il Display del Segno rimangono pressoché invariati, salvo la comparsa, vicino a quest'ultimo, del pulsante di salvataggio. I cambiamenti più significativi riguardano il Menu dei Glifi: il Puppet passa in secondo piano (in questo caso è rappresentato in grigio) e la parte selezionata viene evidenziata (in questo caso viene inscritta in un rettangolo rosso), tuttavia, le parti anatomiche sono ancora cliccabili, in modo da fungere da menu di navigazione senza dover ripassare necessariamente attraverso la home. Compaiono i pulsanti di navigazione nel Menu dei Glifi, che sono stati opportunamente separati dagli altri pulsanti nelle altre zone dell'interfaccia e inseriti nella loro zona di competenza, laddove ci si aspetta di trovarli. Per ciascuna area anatomica ci sono posizioni (configurazioni nel caso delle mani, espressioni nel caso del viso, ecc.) e movimenti, perciò l'utente è chiamato a scegliere la categoria di glifi che sta cercando. La serie di riquadri vuoti nel Menu dei Glifi corrisponde ad una serie di altrettante caratteristiche fondamentali del glifo che si sta cercando, in modo da restringere progressivamente il campo di ricerca. Tali riquadri sono stati chiamati "Box di Scelta". Si noti che in questa fase della progettazione non era ancora

stata definita esattamente la modalità con la quale l'utente avrebbe potuto cercare il glifo desiderato, anche perché in questa fase ancora non erano chiare, per ogni categoria di glifi, le caratteristiche fondamentali che l'utente avrebbe potuto specificare, perciò quello che si può evincere dalla Fig. 5.3 e 5.4 è che l'utente avrebbe utilizzato una sorta di ricerca- navigazione ancora non meglio definita. L'idea era quella di chiedere all'utente di scegliere, una dopo l'altra, in un ordine predefinito, le caratteristiche del glifo desiderato, in modo da offrire, alla fine, una scelta su massimo 30 elementi.

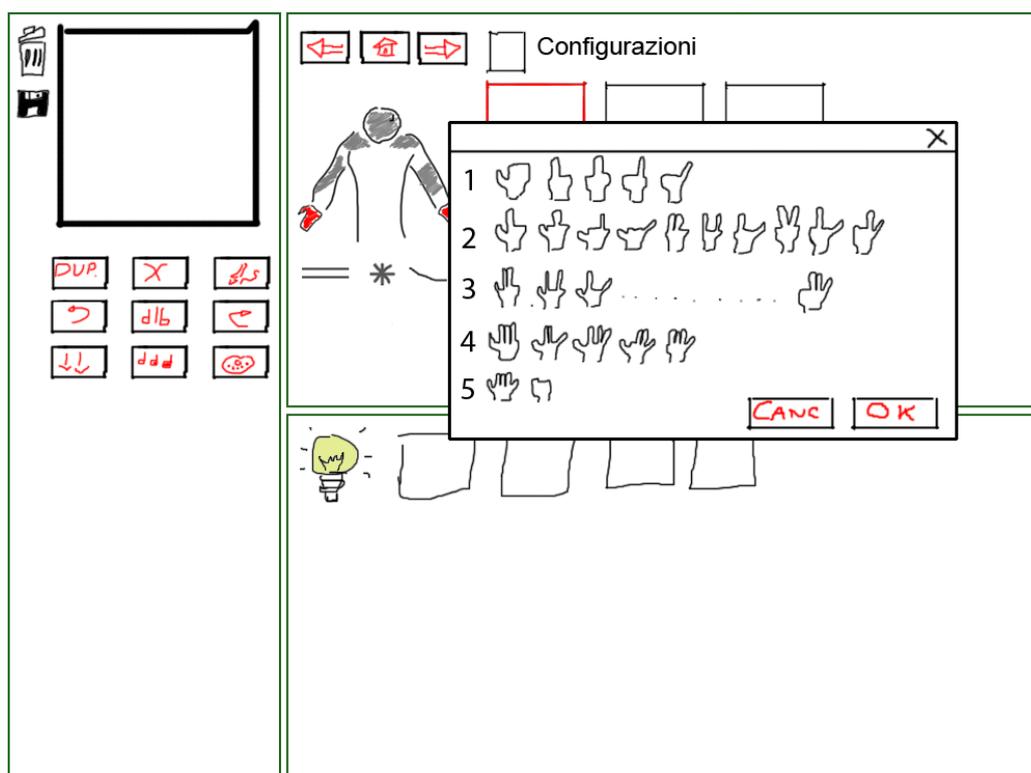


Figura 5.4: Secondo mockup dell'interfaccia del SWift - Sottomenu dell'area delle mani, popup di scelta delle caratteristiche del glifo (2010.04.16).

La Fig. 5.4 descrive il comportamento del SWift dopo che l'utente ha cliccato su un Box di scelta, in questo caso abbiamo ipotizzato che il Box scelto fosse quello relativo al numero delle dita della configurazione di una mano. Un popup offre scelte multiple all'utente, sotto forma di icone disegnate (non ancora di glifi del SW) in questo caso, in pieno stile SignMaker. Vengono proposte all'utente tutte le permutazioni delle configurazioni delle dita di una mano, in modo da identificare le dita coinvolte nel glifo. Nonostante si tratti di un esempio, va comunque notata l'inadeguatezza della scelta proposta all'utente: le immagini sono troppe, perché si stanno proponendo in realtà

due scelte, non una: quante dita e quali dita. L'utente, come si vede, ha la possibilità di approvare una scelta selezionata o annullare l'operazione.

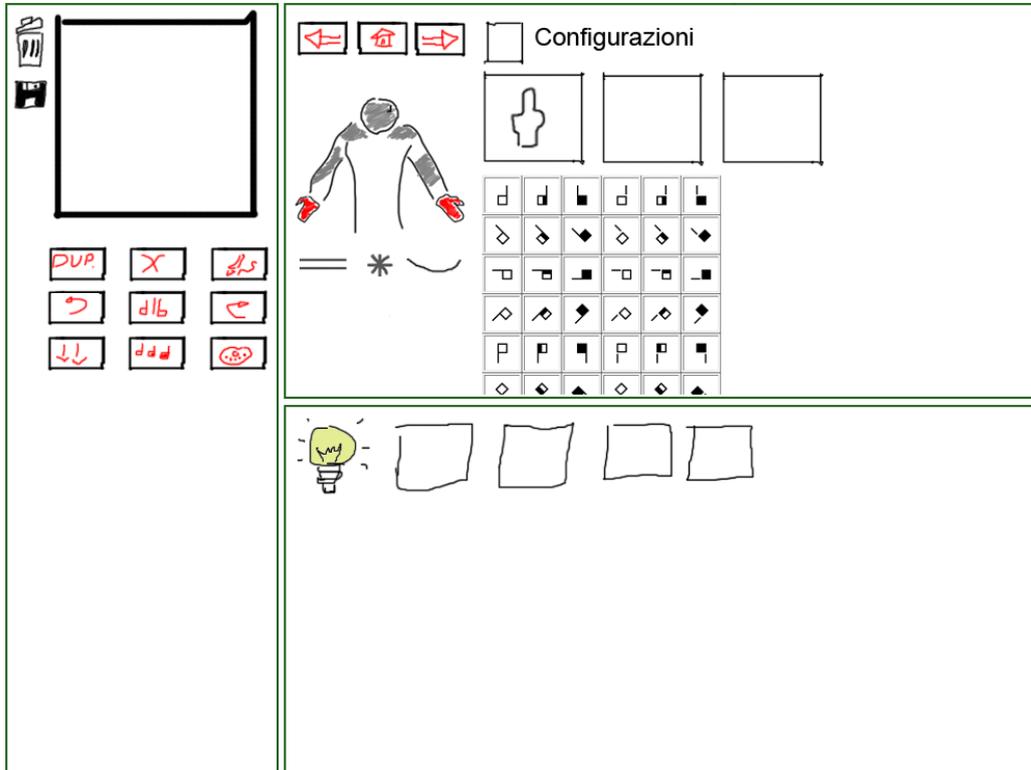


Figura 5.5: Secondo mockup dell'interfaccia del SWift - Sottomenu dell'area della mani, scelta effettuata (2010.04.16).

Una volta approvata la scelta (in caso contrario si torna alla schermata descritta in Fig. 5.3) spariscono alcuni Box dalla schermata (quelli dei movimenti, perché non servono, in quanto l'utente sta scegliendo una configurazione) per far posto ad una serie di glifi tra cui l'utente può navigare, selezionati tenendo conto delle scelte effettuate precedentemente da lui. Una volta trovato il glifo desiderato l'utente può ovviamente trascinarlo nel Display del Segno.

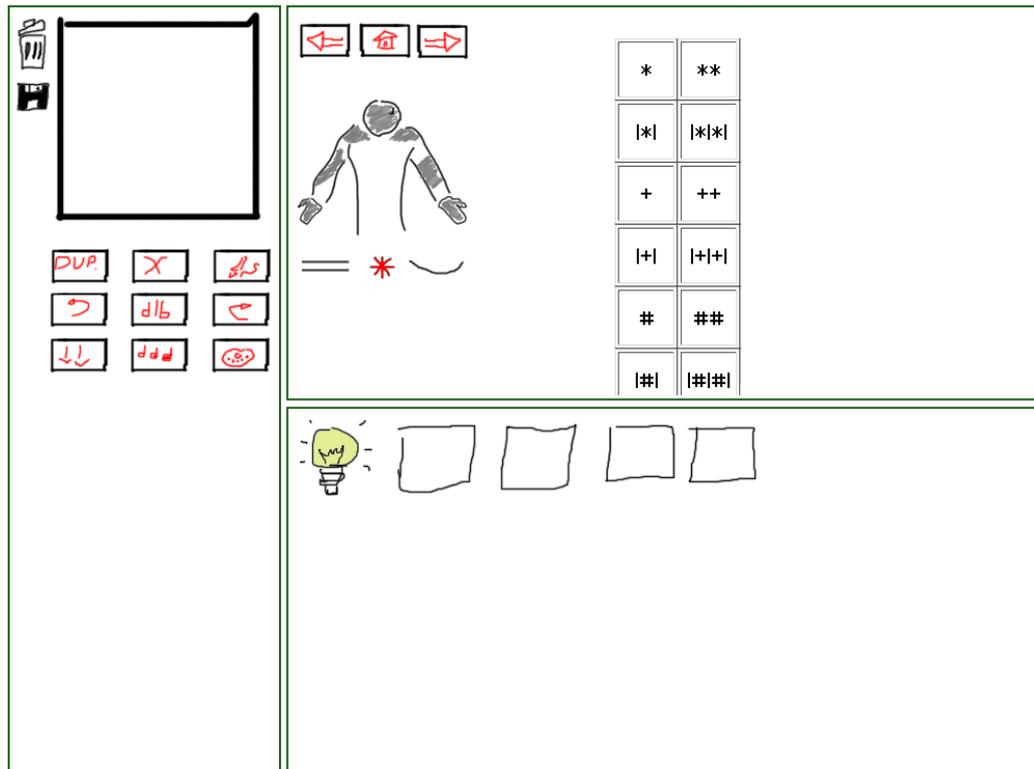


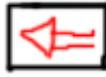
Figura 5.6: Secondo mockup dell'interfaccia del SWift - Sottomenu dell'area dei contatti (2010.04.16).

La Fig. 5.6 mostra semplicemente, per completezza, un menu di scelta dei glifi non legato ad alcuna area anatomica, come ad esempio quello dei contatti.

2.2 Interfaccia - Pulsanti

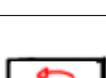
Nelle immagini precedenti abbiamo visto numerosi pulsanti, che ora saranno analizzati. Le funzioni associate ai pulsanti, in questa fase di sviluppo, sono esattamente quelle del SignMaker.

	Questo pulsante consente all'utente di duplicare il glifo selezionato nel Display del Segno. Il glifo duplicato viene posto direttamente nel Display del Segno, le sue coordinate differiscono di qualche pixel da quelle del glifo originale.
	Questo pulsante consente all'utente di eliminare il glifo selezionato nel Display del Segno.
	Questo pulsante consente all'utente di inserire un glifo a mano libera , da utilizzare solo nel caso (rarissimo) in cui il glifo cercato dall'utente non sia presente sul database del SWift.
	Questo pulsante consente all'utente di ruotare in senso antiorario il glifo selezionato nel Display del Segno. Il comportamento di questa funzione varia a seconda del tipo di glifo selezionato.
	Questo pulsante consente all'utente di trasformare il glifo selezionato nel Display del Segno nel suo simmetrico rispetto all'asse verticale .
	Questo pulsante consente all'utente di ruotare in senso orario il glifo selezionato nel Display del Segno. Il comportamento di questa funzione varia a seconda del tipo di glifo selezionato.
	Questo pulsante consente all'utente di applicare alcune variazioni al glifo selezionato nel Display del Segno. Il comportamento di questa funzione varia a seconda del tipo di glifo selezionato.
	Questo pulsante consente all'utente di variare il "riempimento" del glifo selezionato nel Display del Segno. Il comportamento di questa funzione varia a seconda del tipo di glifo selezionato.
	Questo pulsante consente all'utente di applicare una colorazione predefinita ai glifi del SWift in modo da indentificare cromaticamente i diversi tipi di glifi presenti.
	Questo pulsante consente all'utente di eliminare tutto il contenuto del Display del Segno.
	Questo pulsante consente all'utente di salvare il contenuto del Display del Segno .

	Questo pulsante consente all'utente di navigare all'indietro nel Menu dei Glifi .
	Questo pulsante consente all'utente di tornare alla home del Menu dei Glifi .
	Questo pulsante consente all'utente di navigare in avanti nel Menu dei Glifi , questa funzione ha senso, ovviamente, solo e usata in seguito alla funzione di navigazione all'indietro.

Tab. 5.1: Progettazione di pulsanti del del SWift (01).

Al primo mockup dei pulsanti è seguita una fase di creazione delle icone. Il programma utilizzato è stato principalmente Adobe Photoshop, ho preso come punto di riferimento le icone del SignMaker e ho provato a crearne delle altre il cui significato risultasse più intuitivo, sia grazie al colore che alla forma così da essere viste e capite in meno tempo, veicolando solo le informazioni necessarie alla comprensione della funzione associata al pulsante e nient'altro.

		Questo pulsante consente all'utente di eliminare il glifo selezionato nel Display del Segno.
		Questo pulsante consente all'utente di ruotare in senso antiorario il glifo selezionato nel Display del Segno. Il comportamento di questa funzione varia a seconda del tipo di glifo selezionato.
		Questo pulsante consente all'utente di ruotare in senso orario il glifo selezionato nel Display del Segno. Il comportamento di questa funzione varia a seconda del tipo di glifo selezionato.

Tab. 5.2: Progettazione di pulsanti del del SWift (02).

Dopo qualche giorno di lavoro, le icone risultavano ancora insoddisfacenti, si prenda ad esempio l'icona del pulsante per la rotazione antioraria del glifo selezionato nel Display del Segno (Fig. 5.7).

L'idea del movimento in questa icona è data dalle diverse colorazioni dei glifi e dalla freccia, che stabilisce la direzione del movimento. L'icona nel complesso dà un'idea abbastanza fedele della funzione a cui è associata, ma ci sono alcuni difetti che non potevano essere trascurati: i glifi si sovrappongono con la freccia sullo sfondo, creando confusione, la forma della freccia può essere confusa con la freccia della funzione di annullamento dell'ultima



Figura 5.7: Icona del pulsante per la rotazione antioraria del glifo selezionato nel Display del Segno.

azione effettuata (undo), o con quella di navigazione all'indietro. Sorgeva, inoltre, un altro difetto di natura più "grafica": avendo creato le icone con Adobe Photoshop, inevitabilmente utilizzando la grafica raster, esse subivano un fastidioso effetto di alterazione dei pixel (quello che comunemente viene definito "sgranamento") tutte le volte che venivano ridimensionate (sia ridotte, che, ovviamente, ingrandite) perciò ho deciso di passare alla grafica vettoriale per rimediare al problema, continuando il lavoro con Adobe Illustrator. In seguito, leggendo l'articolo di Fajardo *et al.* (2009) ampiamente citato nel Cap. 3 ho pensato che non aveva senso emulare un movimento con un'icona, quando la moderna tecnologia mette a disposizione gli strumenti per rendere animata ogni icona come un piccolo filmato, perciò la scelta è caduta su Adobe Flash, e ho cominciato a creare dei pulsanti che si animano al passaggio del mouse, in modo da far capire all'utente, senza ombra di dubbio, la funzione che si sta per attivare.

Il bottone Flash di rotazione in senso antiorario e quello di eliminazione del glifo selezionato, sono stati le prime componenti animate ad essere sviluppate, il risultato è stato soddisfacente, pertanto l'utilizzo di Flash è diventato uno dei pilastri su cui ho basato la progettazione dell'interfaccia del SWift. Sfortunatamente, lo screenshot converte in grafica raster l'immagine vettoriale dei pulsanti, quindi la qualità è inferiore a quella reale, ma è comunque illuminante per capire il tipo di animazione.

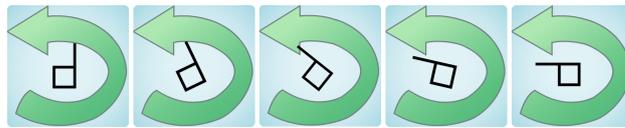


Figura 5.8: Screenshot dell'animazione bottone Flash di rotazione in senso antiorario.

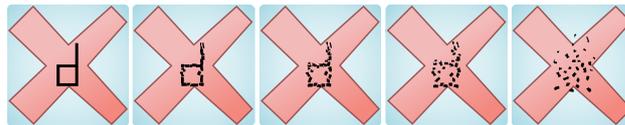


Figura 5.9: Screenshot dell'animazione bottone Flash di eliminazione del glifo selezionato.

3 Terza fase di progettazione (2010.05)

Flash, Flash e ancora Flash.

3.1 Interfaccia

In questa fase della progettazione è venuta alla luce quella che poi sarebbe diventata (seppur con qualche cambiamento) l'interfaccia definitiva del SWift. Da questo momento finisce il periodo dei mockup disegnati pressochè a mano libera, quelli che saranno presentati sono screenshot di un'interfaccia web completa di una componente logica già programmata, che crescerà fino a diventare il SWift definitivo. Sono state trascurate, come si può vedere dalla Fig 5.10, aree come il Pannello di Completamento e una parte del Menu dei Glifi, in quanto gli sforzi si sono concentrati sulle componenti Flash e iconiche dell'interfaccia, vale a dire i pulsanti e il nuovo Puppet.

Tramonta definitivamente l'idea di creare interfacce diverse a seconda della competenza degli utenti, e si passa all'attuale che è stata progettata per aiutare il principiante e non intralciare l'esperto: ad esempio i pulsanti Flash non si animano senza il passaggio del mouse, quindi non creano distrazione a meno che non sia l'utente a volerlo, e il Puppet, così com'è stato concepito, è molto vicino alla realtà, ma effettua comunque la prima importante delimitazione del campo di ricerca.

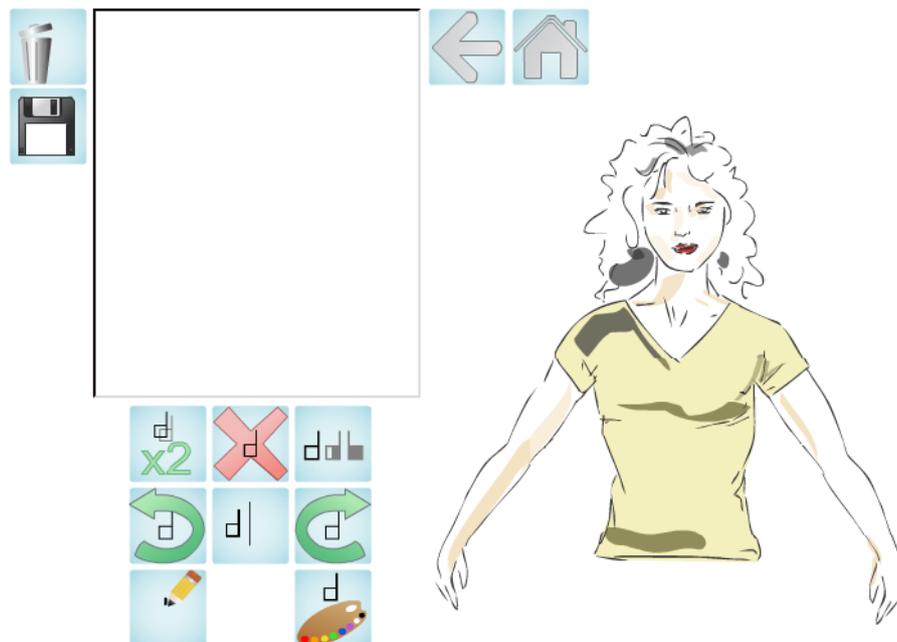


Figura 5.10: Embrione dell'interfaccia del SWift - Home (2010.05.12).

Le componenti che si fanno notare di più sono senza dubbio il nuovo set di pulsanti, che saranno illustrati di seguito, e il nuovo Puppet. Sia i pulsanti che il Puppet non sono altro che componenti Flash, che si comportano in modo opportuno al momento del passaggio del mouse: nel caso dei pulsanti, come già accennato, la loro icona si animerà mostrando all'utente la funzione associata, nel caso del Puppet, la parte su cui il mouse scorre (testa, mani, braccia, spalle) verrà evidenziata per mostrare all'utente la parte anatomica associata. Da notare l'assenza del Pannello di Completamento, a cui è stata data una priorità più bassa, e quindi non sarà visibile fin quando la sua progettazione non sarà completa.

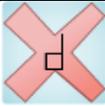
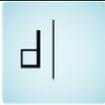


Figura 5.11: Embrione dell'interfaccia del SWift - Sottomenu dell'area della mani (2010.05.12).

Da notare l'assenza della parte del Menu dei Glifi in cui l'utente sceglie il suo glifo, anch'essa ancora non affrontata e quindi non visibile in attesa di una progettazione accurata. Trovano conferma alcune idee già proposte in precedenza, come l'utilizzo di un Puppet di dimensioni ridotte che funga contemporaneamente da menu di navigazione che da "briciola di pane" per ricordare all'utente l'area anatomica sulla quale si sta muovendo.

3.2 Interfaccia- Pulsanti

Un attento lettore avrà già notato la mancanza di due pulsanti: il primo è il bottone di navigazione in avanti nel Menu dei Glifi, assente non per dimenticanza, ma per un motivo più profondo: in questa fase sono sorti numerosi interrogativi sulla reale utilità di avere bottoni di navigazione in avanti e all'indietro poichè l'utente si trova ad effettuare più una ricerca che una navigazione all'interno dei glifi, pertanto la progettazione di questi bottoni è stata sospesa nell'attesa di definire esattamente le modalità di utilizzo del Menu dei Glifi. La seconda importante assenza è quella del pulsante delle variazioni, che assume un comportamento diverso a seconda del glifo al quale è applicato, ho deciso di delegare alcune delle sue funzioni al bottone di variazione del riempimento e di eliminarlo, in quanto fuorviante. La sua mancanza sarà compensata da un più efficace sistema di ricerca che il SWift ha rispetto al SignMaker.

		Questo pulsante consente all'utente di duplicare il glifo selezionato nel Display del Segno. Il glifo duplicato viene posto direttamente nel Display del Segno, le sue coordinate differiscono di qualche pixel da quelle del glifo originale.
		Questo pulsante consente all'utente di eliminare il glifo selezionato nel Display del Segno.
		Questo pulsante consente all'utente di inserire un glifo a mano libera , da utilizzare solo nel caso (rarissimo) in cui il glifo cercato dall'utente non sia presente sul database del SWift.
		Questo pulsante consente all'utente di ruotare in senso antiorario il glifo selezionato nel Display del Segno. Il comportamento di questa funzione varia a seconda del tipo di glifo selezionato.
		Questo pulsante consente all'utente di trasformare il glifo selezionato nel Display del Segno nel suo simmetrico rispetto all'asse verticale .

		Questo pulsante consente all'utente di ruotare in senso orario il glifo selezionato nel Display del Segno. Il comportamento di questa funzione varia a seconda del tipo di glifo selezionato.
	N/A	Questo pulsante consente all'utente di applicare alcune variazioni al glifo selezionato nel Display del Segno. Il comportamento di questa funzione varia a seconda del tipo di glifo selezionato.
		Questo pulsante consente all'utente di variare il "riempimento" del glifo selezionato nel Display del Segno. Il comportamento di questa funzione varia a seconda del tipo di glifo selezionato.
		Questo pulsante consente all'utente di applicare una colorazione predefinita ai glifi del SWift in modo da identificare cromaticamente i diversi tipi di glifi presenti.
		Questo pulsante consente all'utente di eliminare tutto il contenuto del Display del Segno.
		Questo pulsante consente all'utente di salvare il contenuto del Display del Segno .
		Questo pulsante consente all'utente di navigare all'indietro nel Menu dei Glifi .
		Questo pulsante consente all'utente di tornare alla home del Menu dei Glifi .
	N/A	Questo pulsante consente all'utente di navigare in avanti nel Menu dei Glifi , questa funzione ha senso, ovviamente, solo se usata in seguito alla funzione di navigazione all'indietro.

Tab. 5.3: Progettazione di pulsanti del del SWift (03).

I pulsanti presentati nella tabella hanno storie diverse: alcuni hanno avuto un'“infanzia” facile: le loro icone e le loro animazioni sono state subito eloquenti e adeguate al significato della funzione associata (come nel caso del pulsante di eliminazione del glifo selezionato, o del pulsante di salvataggio), per altri, invece, i primi giorni di vita non sono stati facili, e hanno subito

diversi test, ritocchi e a volte completi rifacimenti, grazie anche ai preziosi consigli ricevuti all'ISTC-CNR (uno dei più difficili è stato il bottone che trasforma un glifo nel suo simmetrico rispetto all'asse verticale, rifatto per ben 5 volte). In questa periodo termina la progettazione dei pulsanti: quelli presentati sono definitivi e non saranno più cambiati.

3.3 Database

Il database, nel SWift, contiene esclusivamente due tipi di dati.

- I **glifi**: singoli elementi che l'utente può utilizzare nella composizione del segno che vuole trascrivere. Essi sono circa 45000 e costituiscono una parte importante delle informazioni contenute nel database
- Oltre i glifi è stata prevista la possibilità di memorizzare interi **segni**, che sarà sfruttata nel caso in cui gli utenti decidano di salvare le proprie trascrizioni utilizzando il salvataggio remoto

Comincia in questa fase la progettazione dello schema ER, si può notare, oltre alle entità appena introdotte, una relazione n a m tra **glyphs** e **signs** (tradotti in inglese per coerenza con il resto dell'applicazione) responsabile dell'associazione, per ciascun segno, dei glifi che lo compongono. Tale associazione, essendo n a m, si concretizza nell'entità **components**.

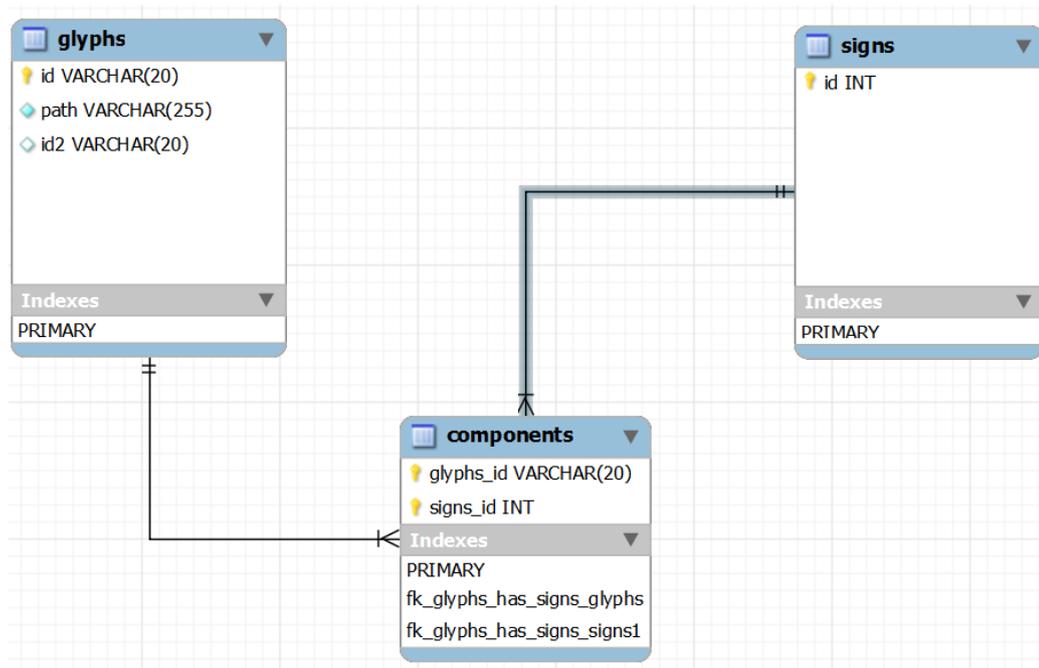


Figura 5.12: Schema ER del database del SWift - Prima versione.

GLYPHS

Tabella che contiene le informazioni sui glifi del SW.

Nome	Tipo	Info	Descrizione
id	VARCHAR(20)	PK	Id secondo l'ISWA 2008, del glifo.
path	VARCHAR(255)	NN	Path dell'immagine png del glifo.
id_iswa	VARCHAR(20)		Id, lasciato per una seconda eventuale codifica ISWA, del glifo.

Tab. 5.4: Definizione della tabella `glyphs` nel database del SWift (01).

SIGNS

Tabella che contiene le informazioni sui segni contenuti nel SWift (tipicamente inseriti con la funzione di salvataggio).

Nome	Tipo	Info	Descrizione
id	INT	PK AI	Id numerica auto-incrementale del segno.

Tab. 5.5: Definizione della tabella `signs` nel database del SWift (01).

COMPONENTS

Tabella che contiene le informazioni sui segni contenuti nel SWift (tipicamente inseriti con la funzione di salvataggio).

Nome	Tipo	Info	Descrizione
glyphs_id	VARCHAR(20)	PK FK	Id del glifo che compone il segno.
signs_id	INT	PK FK	Id del segno.

Tab. 5.6: Definizione della tabella `components` nel database del SWift (01).

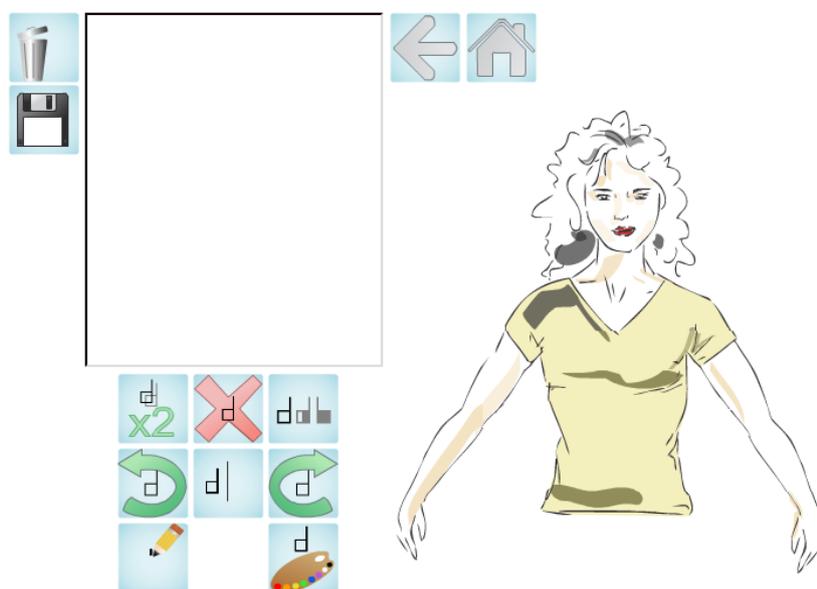
4 Quarta fase di progettazione (2010.06 2010.8)

Costringere tutti gli utenti del SWift a seguire lo stesso ragionamento per cercare un glifo mi sembrava una sorta di violenza intellettuale, perciò ho progettato una modalità diversa di ricerca.

In questo periodo, gran parte del lavoro di progettazione è ormai stato svolto, pertanto la maggior parte del tempo viene dedicata allo sviluppo. Ci sono tuttavia alcuni importanti cambiamenti su cui vale la pena spendere qualche riga.

4.1 Interfaccia

L'interfaccia grafica ha subito grandi cambiamenti, soprattutto per quanto riguarda i sottomenu di scelta nel Menu dei Glifi, come si vedrà nelle figure successive.



 SWift - Sign Writing improved fast transcriber
by [Fabrizio Borgia](#) is licensed under a [Creative Commons](#)
[Attribuzione-Non commerciale-Condividi allo stesso modo 2.5 Italia License](#).

Figura 5.13: Evoluzione dell'interfaccia del SWift - Home (2010.07.29).

L'unica differenza che si può notare nella home è la presenza della licenza Creative Commons, per avere un minimo di protezione legale del lavoro svolto.



Figura 5.14: Evoluzione dell'interfaccia del SWift - Sottomenu dell'area delle mani (2010.07.29).

Passando (Fig. 5.14) al sottomenu dell'area delle mani ci si imbatte finalmente nel meccanismo che permette all'utente di trovare il glifo desiderato in poco tempo, il punto di forza del SWift. Confrontandomi con il gruppo dell'ISTC-CNR, ho capito che imporre all'utente di scegliere, come avevo sempre pensato di fare, le caratteristiche del glifo desiderato in un ordine predefinito e imm modificabile non era la cosa giusta da fare. Ogni persona ha il suo modo di pensare, e, nel nostro caso, anche diversi utenti del SW, quando si trovano a dover rappresentare (o, nel caso del SWift, a cercare) lo stesso glifo, possono seguire ragionamenti diversi: uno può pensare a quante dita ha il glifo che vuole rappresentare, un altro al piano su cui si sta segnando, un altro ancora alla mano, o alla rotazione. Costringere tutti gli utenti del SWift a seguire lo stesso ragionamento per cercare un glifo mi sembrava una sorta di violenza intellettuale, perciò ho progettato una modalità diversa di ricerca. Si osservi la parte superiore destra della Fig. 5.14: l'area della scelta delle configurazioni (l'area dei movimenti è vuota perchè non era stata ancora sviluppata) contiene diversi Box di scelta, ciascuno dei quali identifica e permette di scegliere una caratteristica fondamentale del glifo desiderato, ciascuno dei Box è cliccabile in qualsiasi momento, in modo da lasciare all'utente libertà assoluta di ragionamento. Da notare l'assenza di pulsanti di navigazione nel Menu dei Glifi, completamente eliminati in quanto l'uten-

te effettua semplicemente una ricerca, non una navigazione, all'interno delle decine di migliaia di glifi del SW.



Figura 5.15: Evoluzione dell'interfaccia del SWift - Sottomenu dell'area delle mani, scelta effettuata (2010.07.29).

Nella Fig. 5.15 si può vedere il comportamento dell'applicazione una volta che l'utente effettua una scelta: i Box di scelta inutili spariscono (in questo caso sarebbero dovuti sparire, se fossero stati presenti, i Box di scelta dei movimenti) lasciando il posto al risultato della ricerca sul database. I singoli glifi sono trascinabili nel Display del Segno, come nel SignMaker.

4.2 Database

Giunge al termine la progettazione del database del SWift, lo schema ER e la definizione delle tabelle che presento in questo paragrafo rimarranno tali fino alla versione definitiva del SWift. Tra i significativi cambiamenti si noti l'aggiunta del campo `glyphs.freq`: un contatore numerico che dà informazioni sulla frequenza di utilizzo del glifo e risulterà di vitale importanza per lo sviluppo del completamento automatico. Degno di nota anche l'inserimento, nell'entità `components` di informazioni sulle coordinate spaziali di un glifo nel segno a cui appartiene.

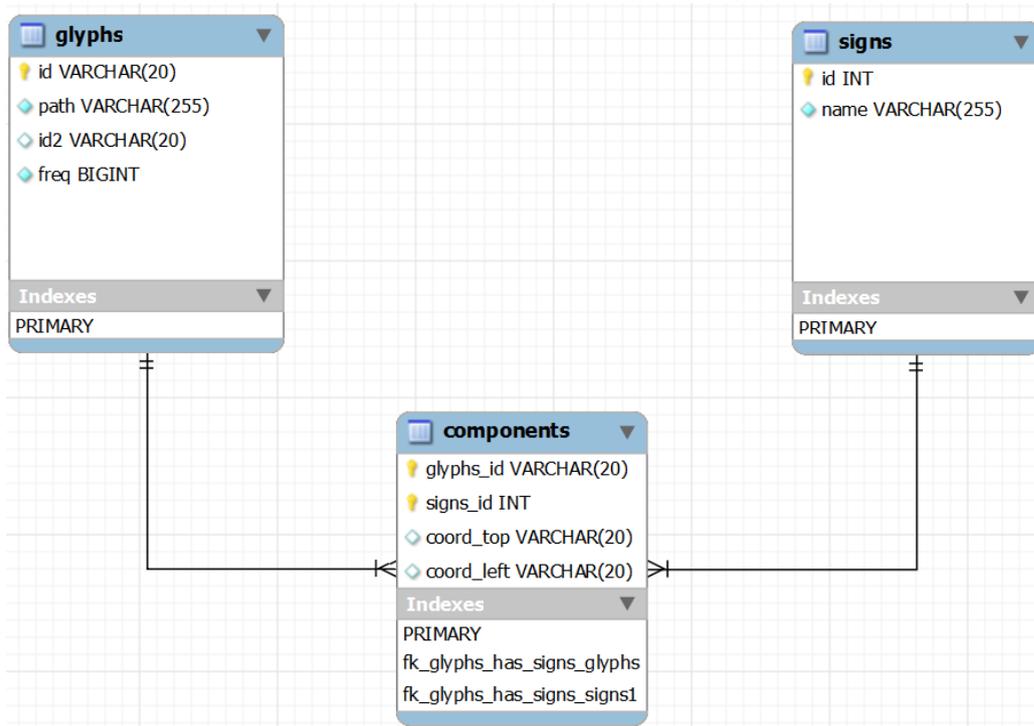


Figura 5.16: Schema ER del database del SWift - Seconda versione (definitiva).

GLYPHS

Tabella che contiene le informazioni sui glifi del SW.

Nome	Tipo	Info	Descrizione
id	VARCHAR(20)	PK	Id secondo l'ISWA 2008, del glifo.
path	VARCHAR(255)	NN	Path dell'immagine png del glifo.
id_2	VARCHAR(20)		Id, lasciato per una seconda eventuale codifica ISWA, del glifo.
freq	BIGINT		Contatore della frequenza di utilizzo, viene incrementato di uno ogniqualvolta il glifo viene utilizzato.

Tab. 5.7: Definizione della tabella `glyphs` nel database del SWift (02).

SIGNS

Tabella che contiene le informazioni sui segni contenuti nel SWift (tipicamente inseriti con la funzione di salvataggio).

Nome	Tipo	Info	Descrizione
id	INT	PK AI	Id numerica auto-incrementale del segno.
name	VARCHAR(255)	NN	Il nome del segno, così come è stato definito dall'utente al momento del salvataggio.

Tab. 5.8: Definizione della tabella `signs` nel database del SWift (02).

COMPONENTS

Tabella che concretizza l'associazione n a m tra `glyphs` e `signs`, associando ad ogni segno i glifi che lo compongono.

Nome	Tipo	Info	Descrizione
glyphs_id	VARCHAR(20)	PK FK	Id del glifo che compone il segno.
signs_id	INT	PK FK	Id del segno.
coord_top	VARCHAR(20)	NN	Posizione sull'asse Y del glifo nel segno.
coord_left	VARCHAR(20)	NN	Posizione sull'asse X del glifo nel segno.

Tab. 5.9: Definizione della tabella `components` nel database del SWift (02).

Con la fase appena descritta termina la storyboard del SWift. Come è facile pensare, sono stati effettuati ancora alcuni cambiamenti durante lo sviluppo, che saranno evidenti dopo la lettura dei Cap. 6 e 7, dedicati alla descrizione del SWift nella sua versione definitiva.

Capitolo 6

Interfaccia del SWift

1 L'abito fa il monaco

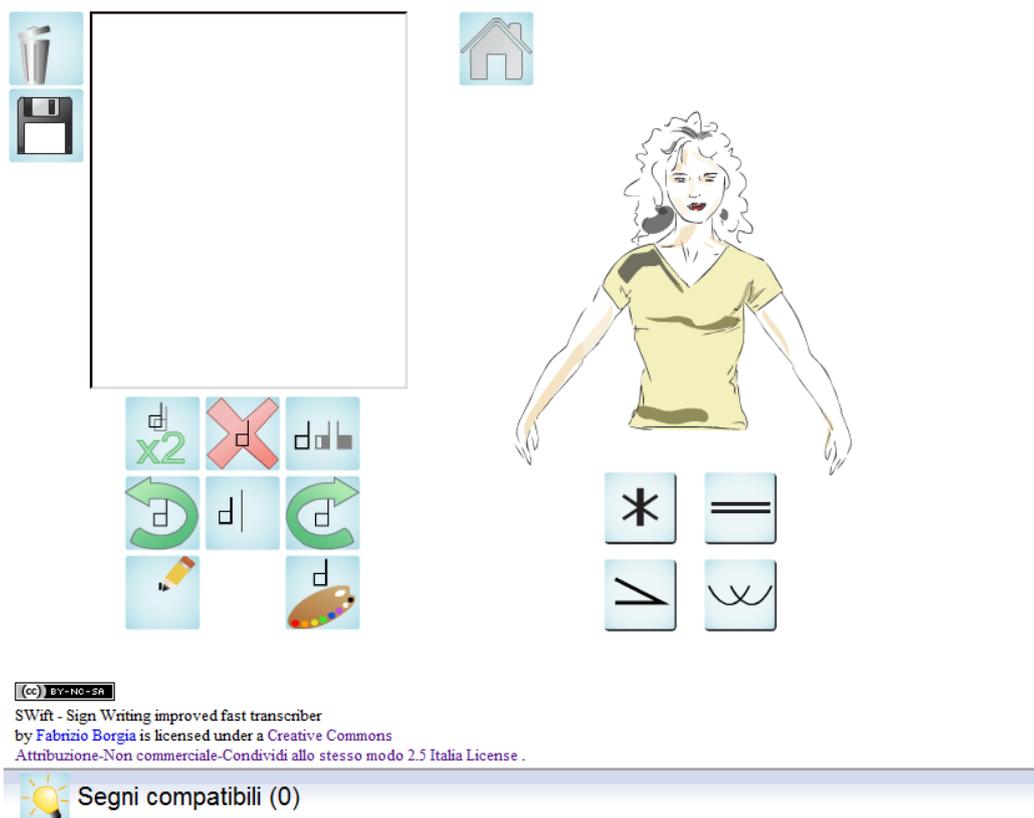


Figura 6.1: Home del SWift.

La Fig. 6.1 mostra l'interfaccia definitiva del SWift, in particolare la home: la prima schermata che l'utente si trova davanti quando decide di utilizzare

il sistema. Come si può notare, rispetto all'interfaccia analizzata nel Cap.2, quella del SWift è semplice, minimizza l'uso di etichette testuali e ha un insieme di icone colorate e il più possibile familiari, l'obiettivo è far sentire l'utente a proprio agio e non disorientarlo subito con una grande mole di informazioni. Tuttavia, un restyling dell'interfaccia, anche se effettuato molto accuratamente, non avrebbe potuto, da solo, migliorare un'applicazione come il SignMaker, infatti, come si può leggere nel Cap.7, alla riprogettazione della grafica è corrisposta anche (e soprattutto) una completa revisione della logica dell'editor.

1.1 Criteri utilizzati

L'interfaccia del SWift è stata progettata tenendo sempre in considerazione alcuni requisiti:

- **Interfaccia intuitiva:** non c'è bisogno che l'utente "impari" ad usare l'interfaccia, quando può semplicemente "capirla", a questo scopo ogni funzione viene presentata in modo intuitivo e familiare
- **Minimalizzazione delle informazioni:** l'interfaccia presenta, di schermata in schermata, solo poche informazioni necessarie, evitando di disorientare l'utente
- **Icone evocative:** le icone sono semplici, familiari e grandi. Nel caso in cui il loro significato rimanga ancora oscuro, è stata prevista, per ognuna, un'animazione che si attiva al passaggio del mouse
- **Minimizzazione dell'uso del testo:** l'uso di etichette testuali è stato limitato il più possibile
- **Validazione dell'interfaccia:** ciascun cambiamento significativo dell'interfaccia è stato concordato con il gruppo dell'ISTC-CNR, sviluppato, e in seguito ricontrollato dal gruppo stesso finché la grafica del SWift non è stata ritenuta del tutto funzionale al suo scopo

Particolare attenzione è stata data ai pulsanti: dopo numerosi tentativi (descritti nel Cap. 5) e test con il gruppo dell'ISTC-CNR ho scelto di creare dei pulsanti con icone vettorializzate (al fine di liberarsi da tutti i problemi legati alla grafica raster) che si animano al passaggio del mouse, dando un'idea della funzione associata al pulsante. I programmi utilizzati per raggiungere l'obiettivo sono Adobe Illustrator per la grafica vettoriale e Adobe Flash per le animazioni.

Per avere un'idea del risultato ottenuto basta guardare le Fig. 6.2 e 6.3: esse contengono gli screenshot delle animazioni di due pulsanti del SWift (il

pulsante di rotazione in senso antiorario e quello di eliminazione del glifo selezionato). Sfortunatamente, lo screenshot converte in grafica raster l'immagine vettoriale dei pulsanti, quindi la qualità è inferiore a quella reale, ma è comunque illuminante per capire il tipo di animazione.

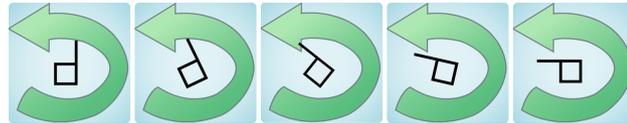


Figura 6.2: Screenshot dell'animazione del pulsante Flash di rotazione in senso antiorario.

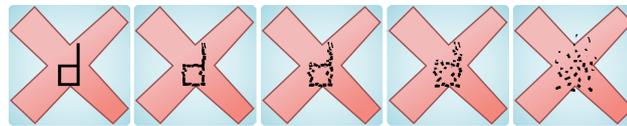


Figura 6.3: Screenshot dell'animazione del pulsante Flash di eliminazione del glifo selezionato.

2 Analisi del SWift

Nell'analisi della parte grafica del SWift utilizzeremo lo stesso metodo adottato per il SignMaker: l'interfaccia sarà divisa in aree di interesse, che saranno prese in considerazione separatamente.

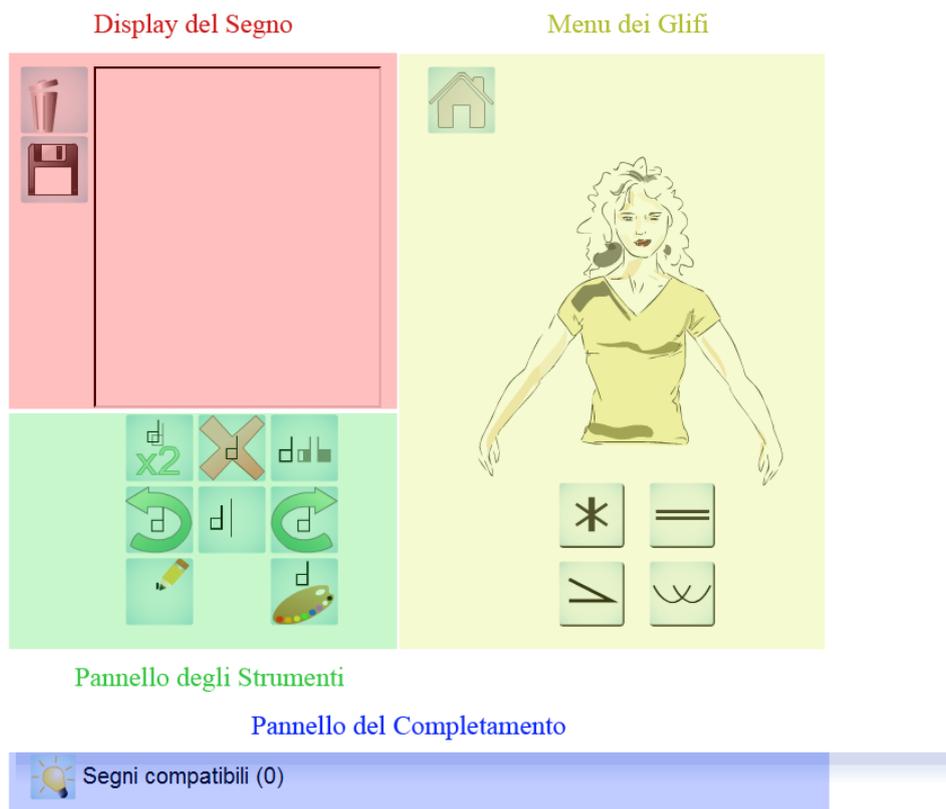


Figura 6.4: Home del SWift, divisa in quattro aree d'interesse.

La Fig. 6.4 distingue l'interfaccia del SWift in quattro aree d'interesse, tre di esse hanno gli stessi nomi e le stesse funzioni di quelle individuate per il SignMaker. Il Pannello del Completamento è una novità introdotta nel SWift, e sarà descritto di seguito.

- **Display del Segno:** area che include un riquadro simile ad una lavagna bianca sulla quale verrà formato, glifo dopo glifo, il segno e alcuni pulsanti per la gestione di tale riquadro
- **Menu dei Glifi:** area che consente, attraverso un sistema di ricerca, di scegliere i glifi da inserire nel Display del Segno

- **Pannello degli Strumenti:** area dell'editor che contiene pulsanti che permettono di eseguire importanti operazioni di gestione dei glifi presenti nel Display del Segno
- **Pannello del Completamento :** Pannello che, man mano che il segno viene composto, offre all'utente la possibilità di inserire, direttamente nel Display del Segno, glifi compatibili con ciò che sta componendo

2.1 Display del Segno

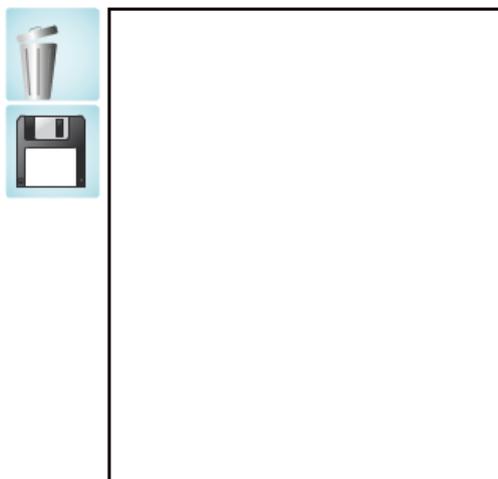


Figura 6.5: Display del Segno.

L'utilità del Display del Segno è quella di mostrare all'utente in tempo reale il segno che sta componendo: esso consiste in un riquadro, che possiamo immaginare come una lavagna bianca, sul quale vengono trascinati i glifi presi dal Menu dei Glifi. Una volta spostati all'interno del riquadro, essi rimangono trascinabili (ovviamente solo all'interno del riquadro stesso) e diventano selezionabili: è stata programmata una funzione che permette all'utente la selezione di più glifi contemporaneamente utilizzando il drag & drop e i pulsanti Shift e Control sulla tastiera, consentendogli di risparmiare tempo nel caso in cui debba effettuare operazioni su più di un glifo. I glifi sono modificabili attraverso le funzioni messe a disposizione dal Pannello degli Strumenti.

Il Display del Segno contiene due pulsanti:

	Questo pulsante consente all'utente di eliminare tutto il contenuto del Display del Segno. Prima dell'eliminazione viene chiesta conferma all'utente.
	Questo pulsante consente all'utente di salvare il contenuto del Display del Segno , aprendo il popup di salvataggio, illustrato in seguito.

Tab. 6.1: Pulsanti presenti nell'area del Display del Segno del SWift.

Il salvataggio, nel SWift, richiede che l'utente inserisca alcuni parametri ed effettui alcune scelte, perciò ho scelto di utilizzare un popup per la raccolta dei dati necessari, come mostrato in Fig. 6.6.



Figura 6.6: Popup di salvataggio del SWift.

All'utente viene data la possibilità di inserire un nome e un numero (facoltativo) per il segno appena composto, che rimane parzialmente visibile, ombreggiato, nel Display del Segno. Una volta inseriti questi dati, si può scegliere di salvare il segno in tre modalità diverse: immagine, testo, e sul database del SWift. Purtroppo non è stato possibile evitare l'uso del testo in questo popup: alcune parole non possono essere associate intuitivamente

ad un'icona, e vanno inserite. Non c'è, ad esempio, un'icona per comunicare all'utente che il file che sta per salvare sarà in formato .png, quindi è stato necessario scriverlo utilizzando un'etichetta testuale.

2.2 Menu dei Glifi



Figura 6.7: Home del Menu dei Glifi.

L'area che consente di scegliere i glifi da inserire nel Display del Segno è detta Menu dei Glifi. Su quest'ultimo si sono concentrati maggiormente gli sforzi di ottimizzazione e progettazione in quanto verosimilmente l'utente passerà la maggior parte del tempo di utilizzo del SWift interagendo con quest'area. Rendere più efficiente il suo utilizzo significa ridurre il tempo di ricerca dei glifi, che a sua volta determina il tempo complessivo di inserimento di un segno.

La figura umana stilizzata, detta "Puppet" e i pulsanti sottostanti, presentano all'utente, in modo naturale, la prima importante scelta: in base all'area anatomica del Puppet scelta, o del pulsante scelto, si potrà accedere al menu di ricerca specifico per quell'area. Ad esempio, facendo clic sulle mani del

Puppet, l'utente sarà portato nel menu di ricerca specifico delle mani, che gli consentirà di scegliere configurazioni e movimenti relativi a quest'area.

Il Puppet è realizzato in Flash, alcune zone (testa, spalle, mani, braccia) si evidenziano al passaggio del mouse, lasciando intuire all'utente che può fare clic per scegliere quella particolare area.

Una volta effettuata una scelta, come già detto, si potrà accedere al menu di ricerca specifico dell'area desiderata, come illustrato in Fig. 6.8.



Figura 6.8: Menu dei Glifi - Menu dell'area delle mani.

Subito dopo il clic dell'utente, il Puppet e i pulsanti sottostanti vengono ridotti e posti a sinistra, e insieme al pulsante per il ritorno alla home del Menu dei Glifi formano il menu di navigazione. Un riquadro rosso intorno all'area in cui ci si trova ricorda all'utente la scelta effettuata, il Puppet e i pulsanti sottostanti rimangono cliccabili, permettendo all'utente di passare da un'area ad un'altra senza dover tornare alla home.

I rettangoli azzurri che occupano la parte centrale del menu sono detti "Box di Scelta", e consentono all'utente di cercare il glifo desiderato. Essi sono raggruppati sotto un'etichetta e un'icona che spiegano all'utente quale tipo di

glifi sono ottenibili utilizzando i Box sottostanti.

Ciascun Box consente all'utente di selezionare le caratteristiche del glifo che sta cercando, nell'ordine che gli è più congeniale, non privilegiando, quindi, un modo di ragionare rispetto ad un altro. Ad esempio un utente potrebbe scegliere prima la rotazione del proprio glifo, mentre un altro potrebbe privilegiare la scelta della mano utilizzata, o del numero delle dita. Per ogni scelta effettuata in qualsiasi box di scelta, vengono aggiornati i glifi mostrati all'utente (si veda la Fig. 6.9). I Box sono progettati in modo che nel caso peggiore, cioè se l'utente, non trovando tra i primi risultati il suo glifo, effettua ogni scelta disponibile, selezionando una casella in ciascun Box, alla fine dovrà scegliere tra circa 30 glifi, scelta che un umano è in grado di operare in tempo accettabile.

In alcune aree, come quella delle mani, in cui c'è un gran numero di glifi tra cui scegliere, può capitare di avere una doppia serie di Box: nella Fig. 6.8, infatti, vediamo che l'utente può scegliere di cercare un glifo che rappresenta un movimento o uno che rappresenta una configurazione. Una volta effettuata la prima scelta, i Box superflui spariscono per far posto ai glifi risultanti della ricerca, ad esempio scegliendo di cercare un glifo che rappresenta una configurazione (magari scegliendo la mano utilizzata) spariscono i Box della ricerca dei glifi di movimento, rimpiazzati dai risultati della ricerca (Fig. 6.9).



Figura 6.9: Menu dei Glifi - Menu dell'area delle mani - Scelta effettuata.

Una volta effettuata la prima scelta in uno qualsiasi dei Box, l'interfaccia mostra all'utente i glifi che soddisfano la caratteristica desiderata (Fig. 6.9), essi vengono collocati in un pannello scrollabile e sono immediatamente trascinabili sul Display del Segno. In casi molto fortunati l'utente riesce a trovare il glifo desiderato effettuando solo una scelta, molto spesso, invece, è necessario definire circa 3 caratteristiche per ottenere il glifo scelto. All'utente viene concessa la possibilità di annullare una scelta effettuata: cliccando su una casella selezionata, essa viene deselezionata e i glifi risultanti si aggiornano di conseguenza.

In quest'area, se si escludono il Puppet e i pulsanti di scelta dell'area, c'è solo un pulsante:

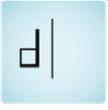
	Questo pulsante consente all'utente di tornare alla home del Menu dei Glifi
---	--

Tab. 6.2: Pulsanti presenti nell'area del Menu dei Glifi del SWift.

2.3 Pannello degli Strumenti

Il Pannello degli Strumenti è una pulsantiera che mette a disposizione dell'utente diverse funzioni per la modifica del glifo o dei glifi che sono attualmente selezionati nel Display del Segno. Grazie alla funzione di selezione multipla offerta dal Display del Segno, l'utente può modificare, utilizzando le funzioni del Pannello degli Strumenti, più glifi contemporaneamente.

I pulsanti sono i seguenti:

	Questo pulsante consente all'utente di duplicare il glifo selezionato nel Display del Segno. Il glifo duplicato viene posto direttamente nel Display del Segno, le sue coordinate differiscono di qualche pixel da quelle del glifo originale.
	Questo pulsante consente all'utente di eliminare il glifo selezionato nel Display del Segno.
	Questo pulsante consente all'utente di inserire un glifo a mano libera , da utilizzare solo nel caso (rarissimo) in cui il glifo cercato dall'utente non sia presente sul database del SWift.
	Questo pulsante consente all'utente di ruotare in senso antiorario il glifo selezionato nel Display del Segno. Il comportamento di questa funzione varia a seconda del tipo di glifo selezionato.
	Questo pulsante consente all'utente di trasformare il glifo selezionato nel Display del Segno nel suo simmetrico rispetto all'asse verticale .
	Questo pulsante consente all'utente di ruotare in senso orario il glifo selezionato nel Display del Segno. Il comportamento di questa funzione varia a seconda del tipo di glifo selezionato.
	Questo pulsante consente all'utente di variare il "riempimento" del glifo selezionato nel Display del Segno. Il comportamento di questa funzione varia a seconda del tipo di glifo selezionato.
	Questo pulsante consente all'utente di applicare una colorazione predefinita ai glifi del SWift in modo da identificare cromaticamente i diversi tipi di glifi presenti.

Tab. 6.3: Pulsanti presenti nell'area del Pannello degli Strumenti del SWift.

2.4 Pannello del Completamento

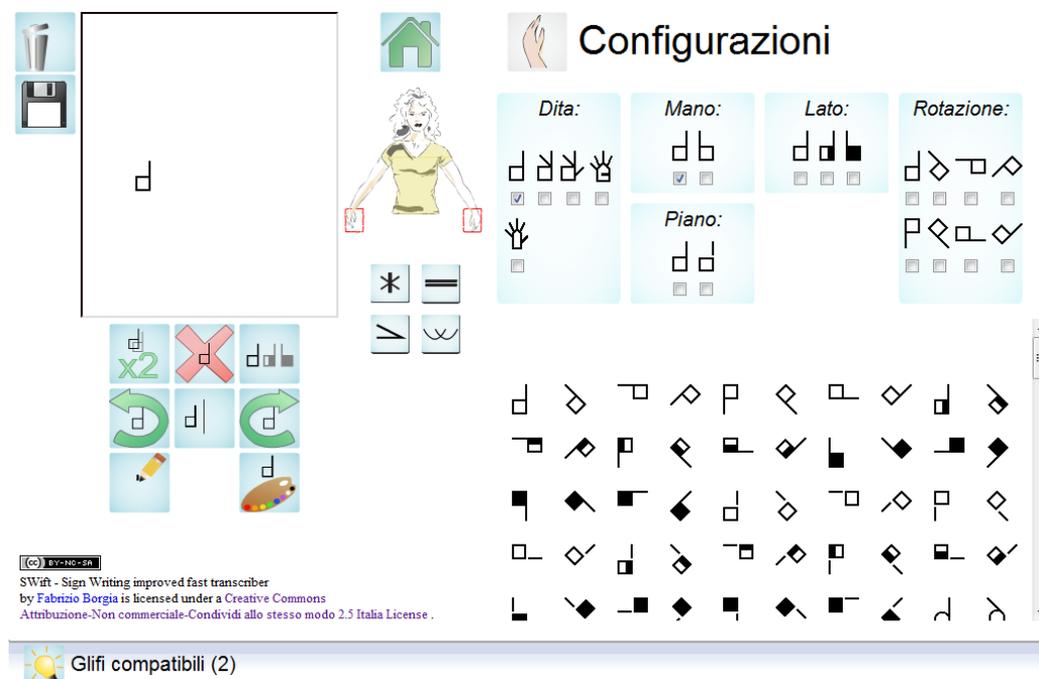


Figura 6.10: Pannello del Completamento minimizzato (in basso).

Il Pannello del Completamento rappresenta una delle più grandi innovazioni del SWift. Ha il compito, subito dopo l'inserimento del glifo, di proporre all'utente una serie di glifi che sono compatibili con quelli presenti nel Display del Segno. Per compatibili si intende che, nei segni salvati sul database del SWift, con una frequenza significativa, i glifi proposti compaiono insieme a quelli inseriti dall'utente nel Display del Segno. Il Pannello del Completamento propone all'utente esclusivamente glifi appartenenti all'area anatomica in cui si trova: se l'utente si trova nell'area anatomica delle mani, i glifi proposti saranno esclusivamente configurazioni e movimenti dell'area delle mani. Nel caso in cui l'utente veda un glifo che vuole inserire nel Display del Segno, utilizzerà il classico sistema di drag & drop previsto dall'interfaccia.

Per motivi di spazio il Pannello del Completamento è implementato come un footer espandibile. Quando è minimizzato (Fig. 6.10), un'etichetta testuale notifica all'utente quanti glifi compatibili sono stati trovati. Quando è massimizzato (Fig. 6.11), l'utente vede e sceglie i glifi proposti. I dati per il completamento automatico sono ancora pochi, in quanto i segni disponibili finora sul database sono stati inseriti solamente per scopi di testing, questo spiega lo scarso numero e l'inadeguatezza dei glifi compatibili forniti.



Figura 6.11: Pannello del Completamento massimizzato.

Capitolo 7

Logica del SWift

1 Linguaggi

Il SWift, come già esplicitato nella definizione dei requisiti (Cap. 4), è stato concepito per essere un'applicazione multiplatforma (cioè funzionante su Windows, Macintosh, e Linux) e per avere una facile diffusione. Di conseguenza, sia per non essere soverchiato dagli innumerevoli dettagli da tenere in considerazione quando si programma un'applicazione desktop multiplatforma, sia per diffondere il SWift più rapidamente possibile (risparmiando all'utente l'onere dell'installazione e della configurazione) ho deciso di sviluppare una web-application. Tenendo conto anche della considerevole mole di dati che il SWift è chiamato a gestire (circa 45.000 glifi in formato immagine), la scelta di sviluppare un'applicazione web da collocare su un server è stata pressochè obbligata. La scelta dei linguaggi da utilizzare è stata determinata dalle mie competenze precedenti, ed è stata la seguente:

Linguaggio	Informazioni	Utilizzo nel SWift
XHTML	L'XHTML (eXtensible HyperText Markup Language) è un linguaggio di marcatura che associa alcune proprietà dell'XML con le caratteristiche dell'HTML.	Essendo una web application, l'XHTML si occupa, coadiuvato dal JavaScript, della grafica e dell'inclusione, nel SWift, delle RISORSE JavaScript necessarie al funzionamento dell'applicazione.

JavaScript + jQuery	JavaScript è un linguaggio di scripting orientato agli oggetti sviluppato da Netscape utilizzato in milioni di pagine web e web application in tutto il mondo per la programmazione web dinamica. jQuery è una libreria di funzioni per le pagine web, codificata in javascript, che si propone come obiettivo quello di astrarre ad un livello più alto la programmazione lato client del comportamento di ogni singola pagina HTML.	Il JavaScript si occupa del funzionamento della maggior parte dell'applicazione (jQuery è stato utilizzato principalmente per le componenti grafiche e per definire i loro comportamenti a seguito dell'input dell'utente) tranne di quelle parti di cui, essendo il JavaScript un linguaggio "lato client", non si può occupare.
PHP	PHP (acronimo ricorsivo di PHP: Hypertext Preprocessor) è un linguaggio di scripting interpretato, originariamente concepito per la realizzazione di pagine web dinamiche. Attualmente è utilizzato principalmente per sviluppare applicazioni web lato server ma può essere usato anche per scrivere script a linea di comando o applicazioni standalone con interfaccia grafica.	Il PHP si occupa delle operazioni lato server del SWift (interazione con il Database, generazione dei file da restituire all'utente a seguito di una richiesta di salvataggio, generazione di immagini, ecc.).
AS	AS (ActionScript) è il linguaggio di scripting di Adobe Flash.	L'AS è stato utilizzato per consentire alle componenti Flash di interagire con il resto dell'applicazione.
SQL	SQL (Structured Query Language) è un linguaggio di interrogazione per database progettato per leggere, modificare e gestire dati memorizzati in un sistema basato sul modello relazionale,	L'SQL è stato utilizzato per interrogare il database.

Tab. 7.1: Linguaggi utilizzati nello sviluppo del SWift.

2 Architettura

2.1 MVC

Il SWift è stato sviluppato utilizzando il pattern architetturale MVC (Model View Controller). Nella programmazione di applicazioni chiamate a supportare un solo tipo di cliente, a volte è utile interlacciare le sezioni dell'applicazione che si occupano dell'accesso ai dati a quelle che si occupano della logica di business (cioè tutta quella logica applicativa che rende operativo un programma) o della logica dell'interfaccia. Tale approccio, tuttavia, è inadeguato per sistemi che devono supportare diversi tipi di clienti. Il SWift rientra in questa categoria, infatti esso è stato concepito per essere un editor per il SW, ma potrebbe essere incorporato, in futuro, in altre applicazioni che richiedano le sue funzionalità, di conseguenza la parte logica dell'applicazione potrebbe trovarsi a dover convivere con una nuova interfaccia, o con un nuovo schema di database. Se il SWift non fosse stato progettato in MVC, per ogni tipo di interfaccia da supportare andrebbe sviluppata una diversa applicazione, ripetendo, in ciascuna di esse, la parte di codice non-specifico per l'interfaccia con conseguente e inutile doppio carico di lavoro di sviluppo (spesso di tipo copia-incolla), così come di prova e di manutenzione. Il lavoro di duplicazione, inoltre è inevitabilmente imperfetto. Lentamente, ma inesorabilmente, applicazioni che dovrebbero fornire le stesse funzionalità fondamentali evolverebbero in sistemi diversi.

Adottando il pattern MVC in un'applicazione (in questo caso una web application), si separano le logiche di business fondamentali dalle logiche di presentazione e di controllo. Tale separazione permette a più viste di condividere lo stesso modello di dati, il che rende il sostegno per più client più semplice da implementare, testare e mantenere.

Il pattern MVC è basato sulla separazione dei compiti fra i componenti software che interpretano tre ruoli principali:

- Il **model** si occupa di fornire metodi per accedere ai dati utili all'applicazione e dell'update di questi ultimi
- La **view** rappresenta graficamente il contenuto di un model: accede ai dati dell'applicazione attraverso il model e determina come i dati vengono rappresentati. E' compito della view di mantenere la consistenza tra i dati rappresentati e quelli presenti nel model, questo può essere effettuato utilizzando una logica "push", dove la view si registra per avere notifiche dei cambiamenti nel model (come nel caso del SWift), o una logica "pull", in cui la view interpella il model per ricevere i cambiamenti recenti dei dati.

- Il **controller** trasforma le interazioni con la view in azioni effettuate dal model. Le azioni effettuate dal controller includono l'attivazione di procedure di business o il cambio dello stato del model. Basandosi sull'interazione dell'utente e sull'esito delle azioni del model, il controller risponde selezionando una vista appropriata.

2.2 SWift e MVC

La Fig. 7.1 ci mostra la directory principale del SWift e il contenuto di alcune importanti sottodirectory.

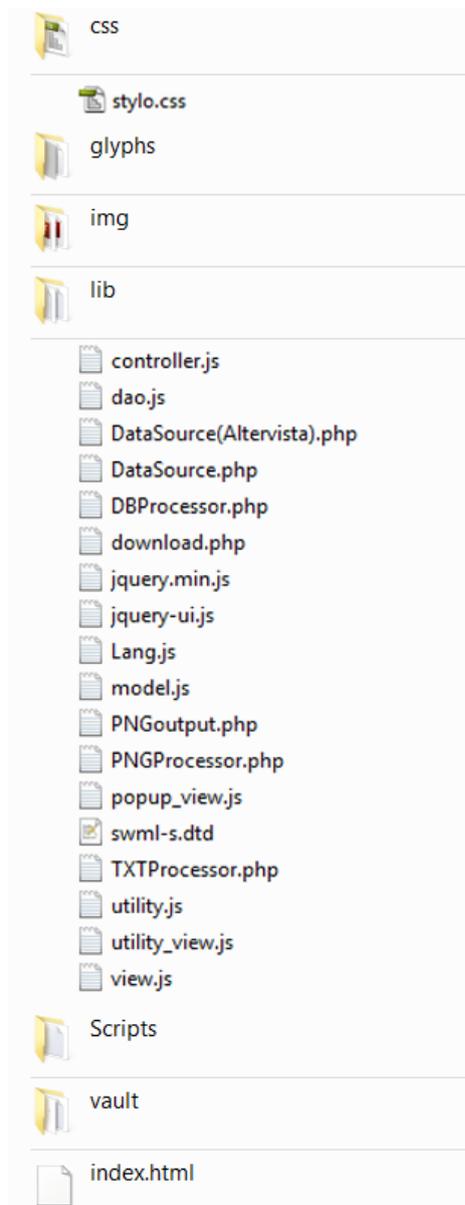


Figura 7.1: Contenuto della directory principale del SWift.

La funzione di ciascun file presente nella cartella `lib`, che costituisce il centro dell'applicazione, sarà esplicitata in seguito. Prima è necessario spendere qualche parola sulla struttura delle directory del SWift. Nella figura 7.1 vediamo le seguenti directory:

- `css`: contiene il file `stylo.css` che definisce lo stile delle componenti grafiche dell'applicazione.
- `glyphs`: contiene le immagini (`.png`) dei glifi del SW (ordine di grandezza delle decine di migliaia).
- `img`: contiene le componenti grafiche dell'applicazione (immagini, componenti Flash, ecc.).
- `lib`: contiene principalmente i file sorgenti del SWift.
- `Scripts`: directory che contiene i file necessari al funzionamento delle componenti Flash.
- `vault`: directory che contiene file temporanei generati in fase di salvataggio e di completamento automatico del segno.

Ora analizzerò i file sorgenti dell'applicazione, seguendo la divisione tracciata naturalmente dal pattern MVC (view, model controller, ecc.) in modo da spiegare in dettaglio la loro funzione.

view



view.js

Contiene la view dell'applicazione: funzioni di inizializzazione, di update e di gestione delle componenti grafiche



utility_view.js

Contiene i dati e le funzioni per l'inizializzazione delle componenti HTML della view.



popup_view.js

Contiene le funzioni per la gestione della grafica del popup di salvataggio.



jquery.min.js



jquery-ui.js

Librerie jQuery e jQueryUI necessarie per il rilevamento e la gestione degli eventi.



PNGoutput.php

File di presentazione dell'output nel caso in cui l'utente scelga il salvataggio in formato immagine.

Figura 7.2: File che si occupano della view del SWift.

La view (Fig. 7.2) si occupa, come già accennato prima, della gestione dell'interfaccia del SWift: dall'inizializzazione e la rappresentazione delle componenti grafiche alla gestione degli eventi dal punto di vista visivo (drag, selezione, clic, ecc) all'aggiornamento dei dati rappresentati in coerenza con quelli contenuti nel model.

I file che si occupano della view sono i seguenti:

- **view.js**: si occupa del funzionamento della view dell'applicazione. Contiene gli array con il codice HTML di ogni singola componente dell'interfaccia grafica. Si occupa dell'inizializzazione dei gestori grafici degli eventi (drag, selezione e clic) cioè di quelle logiche che danno un feedback grafico all'utente a seguito di un suo input (non si occupa propriamente della gestione degli eventi dal punto di vista logico, quello è compito del controller). Ultimo, ma non meno importante, si occupa di chiamare l'inizializzazione (compito delegato ad `utility_view.js`) di tutte le aree dell'interfaccia (escluso il popup di salvataggio), e si occupa in prima persona del loro aggiornamento
- **utility_view.js**: fornisce i dati e le funzioni per l'inizializzazione di tutte le aree e delle singole componenti dell'interfaccia. Visto l'elevato numero dei dati per l'inizializzazione delle componenti grafiche, è stata necessaria, per motivi di ordine, la creazione di un file apposito. La libreria `utility_view.js` non contiene altro che degli array di dati, ciascuna riga di ogni array da informazioni su come inizializzare una componente grafica (ad esempio, nel caso di una componente Flash, la riga conterrà informazioni come: il nome del file Flash, le dimensioni che deve avere, le coordinate spaziali, ecc.) e delle funzioni per trasformare ciascuna riga (a seconda del tipo di componente che descrive) in codice HTML utilizzabile nell'interfaccia grafica
- **popup_view.js**: Contiene le funzioni per la gestione della grafica del popup di salvataggio
- **jquery.min.js** e **jquery-ui.js**: Librerie jQuery e jQueryUI necessarie per il rilevamento e la gestione grafica degli eventi (drag, selezione e clic)
- **PNGoutput.php**: un semplice file php che si occupa della presentazione dell'output nel caso in cui l'utente scelga di salvare il suo lavoro in formato immagine

controller



controller.js

Si occupa della gestione della logica di business dell'applicazione.



utility.js

Contiene funzioni di utilità che coadiuvano il controller in numerosi task.



PNGProcessor.php



TXTProcessor.php

File di generazione dell'output nel caso in cui l'utente scelga il salvataggio in formato immagine o testo.

Figura 7.3: File che si occupano del controller del SWift.

Il controller (Fig. 7.3) si occupa, come già accennato prima, della logica di business dell'applicazione, principalmente trasforma le interazioni con la view in azioni effettuate dal model.

I file che si occupano del controller sono i seguenti:

- **controller.js**: si occupa del funzionamento del controller dell'applicazione. Contiene informazioni sullo stato dell'applicazione (ad esempio conserva, in tempo reale, informazioni sull'area anatomica del Menu dei Glifi consultata dall'utente), si occupa della gestione logica degli eventi e contiene le funzioni per richiedere nuovi dati al model
- **utility.js**: contiene (per motivi di ordine) numerose funzioni di utilità che coadiuvano il controller nello svolgimento dei task più disparati (conversioni tra diversi tipi di dato, interazione con i file php, ecc.)
- **TXTprocessor.php** e **PNGprocessor.php**: librerie php chiamate nel caso in cui l'utente scelga di salvare il suo lavoro in formato immagine o testo, si occupano della creazione vera e propria dei file di output

model

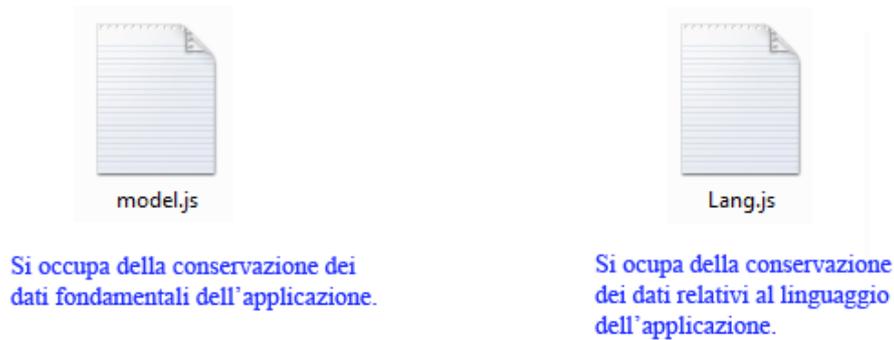


Figura 7.4: File che si occupano del model del SWift.

Il model (Fig. 7.4) si occupa di fornire metodi per accedere ai dati utili all'applicazione e dell'update di questi ultimi.

I file che si occupano del controller sono i seguenti:

- `model.js`: si occupa di conservare i dati fondamentali dell'applicazione: i glifi da rappresentare nel menu dei glifi, le scelte effettuate dall'utente nello scegliere i glifi, ecc. Su richiesta del controller può aggiornare i propri dati richiedendone dei nuovi al database, interagendo con `dao.js`
- `lang.js`: contiene array (uno per ciascuna lingua) con le traduzioni di tutte le stringhe di testo utilizzate nell'applicazione, in modo da supportare più lingue vocali, al momento sono supportate le seguenti lingue: italiano, inglese

DAO

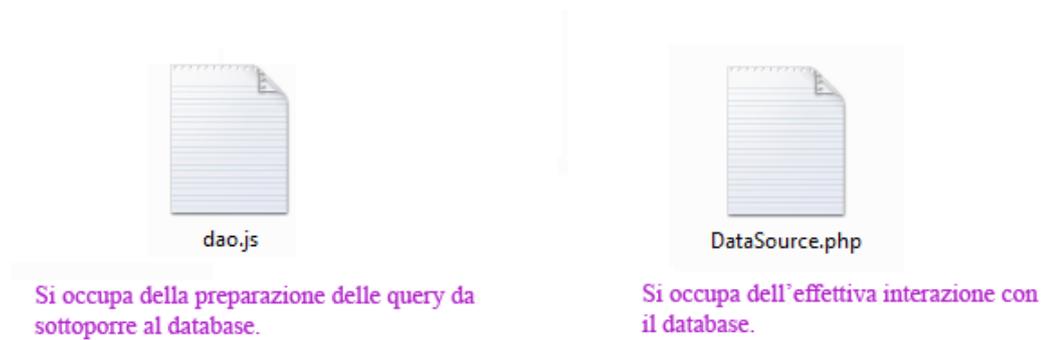


Figura 7.5: File che si occupano del DAO del SWift.

Il DAO (Data Access Object) (Fig. 7.5) si occupa della vera interazione con il database: dalla costruzione delle query all'interrogazione vera e propria.

I file che si occupano del controller sono i seguenti:

- `dao.js`: incorpora dati e funzioni per costruire tutte le query necessarie al SWift. Entra in azione quando le sue funzioni vengono chiamate dal model.
- `DataSource.php`: si occupa dell'interazione vera e propria con il database: effettua la query ricevuta da `dao.js` e gli restituisce i risultati

Ora che è stata esplicitata, per ciascuna libreria, la funzione svolta, è possibile descrivere le loro interazioni. La Fig. 7.6 dà un'idea generale del funzionamento dell'applicazione rappresentando in forma grafica lo scambio di informazioni tra le diverse librerie del SWift.

Quando l'utente carica la home page del SWift sul suo browser (il file `index.html` nella directory principale, visibile in Fig. 7.1) attiva il controller (`controller.js`) che chiama la funzione principale di inizializzazione dell'applicazione. Tale funzione, prepara opportunamente i dati di controllo del comportamento del SWift e tra i suoi compiti, ha quello di chiamare la funzione di inizializzazione dell'interfaccia grafica, contenuta in `view.js`. La funzione di inizializzazione della view, data la grande quantità di componenti grafiche utilizzate nel SWift, si avvale dei dati e delle funzioni di `utility_view.js` per ottenere i blocchi HTML da scrivere nella pagina. Come detto prima, la libreria `utility_view.js` non contiene altro che degli array di dati contenenti informazioni su come inizializzare le componenti grafiche e delle funzioni per trasformare tali informazioni in codice HTML utilizzabile nell'interfaccia grafica. Come si può notare dalla Fig. 7.6, `utility_view.js` si avvale del file `lang.js` (a cui accede attraverso la libreria di utilità del controller) per ottenere le stringhe di testo, tradotte nella lingua opportuna, da inserire nelle componenti al momento dell'inizializzazione.

Una volta ottenuto tutto il codice HTML richiesto, `view.js` scrive le opportune componenti nel sorgente della pagina e chiama le sue funzioni di inizializzazione per controllare il comportamento, a seguito di determinati input, delle componenti stesse. Termina così la fase di inizializzazione dell'applicazione, da questo momento in poi il SWift non farà altro che rispondere agli input dell'utente.

Ogni volta che viene rilevato un evento, il controller interviene e agisce di conseguenza, i tipi di eventi che possono occorrere nel SWift si distinguono in tre famiglie:

- Clic di un pulsante
- Navigazione all'interno del Menu dei Glifi
- Scelta di un glifo e suo trascinamento e gestione nel Display del Segno

Nel primo caso (il clic di un pulsante), `controller.js` chiama l'opportuna routine di gestione (spesso sono funzioni che modificano leggermente le caratteristiche dei glifi selezionati nel Display del Segno. L'unica eccezione è rappresentata dal pulsante di salvataggio, che apre un popup (chiamando `popup_view.js` della view) presentando all'utente le opzioni di salvataggio e agendo di conseguenza: in un caso (salvataggio remoto) il controller chiamerà il model per accedere al DAO e salvare il segno inserito dall'utente

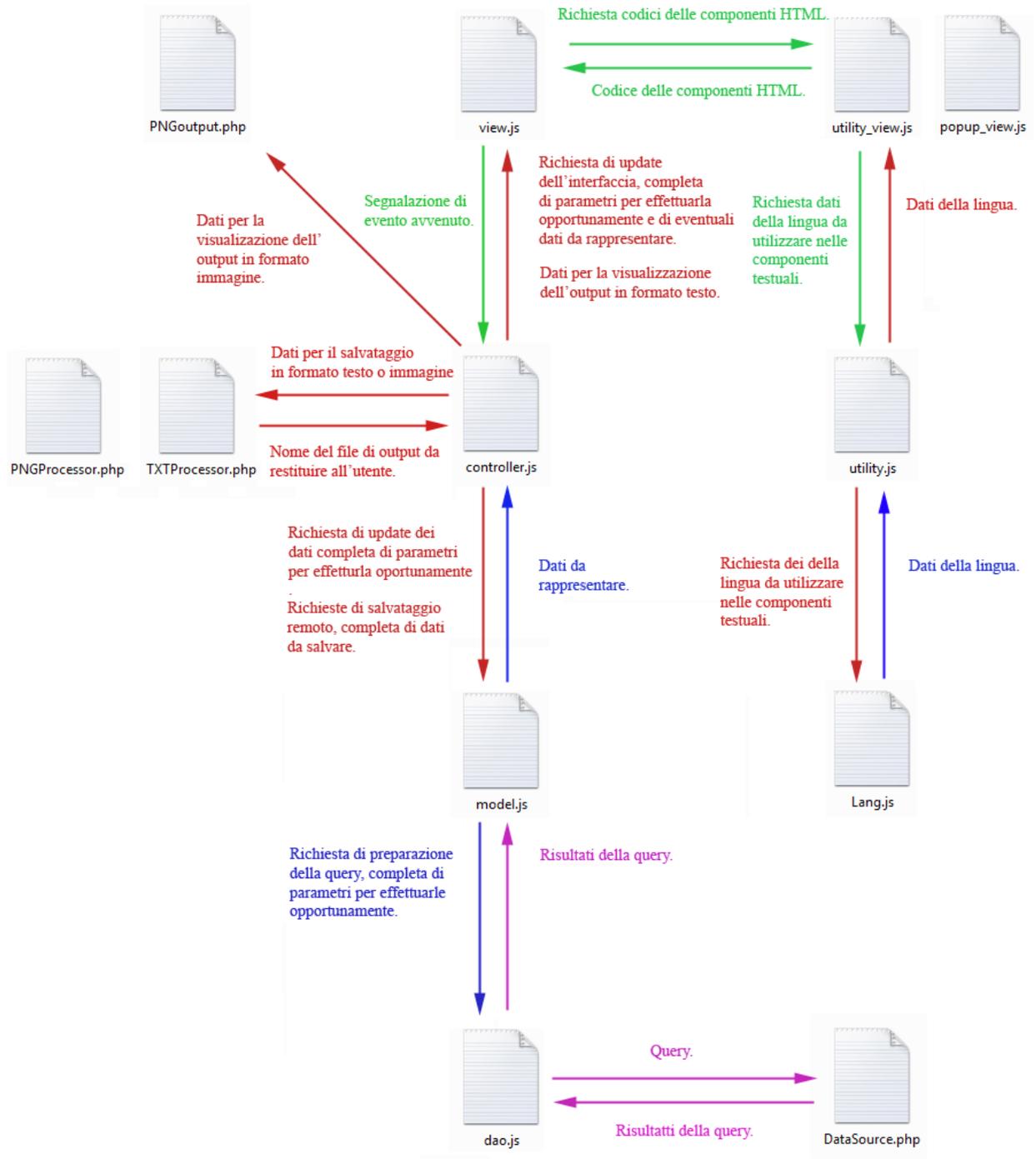


Figura 7.6: Interazione delle librerie del SWift.

sul database, negli altri due (salvataggio in formato immagine e in formato testo) il controller genererà i file di output, utilizzando le librerie `phpTXTprocessor.php` e `PNGprocessor.php`, e chiamerà la funzione della view che si occupa della visualizzazione dell'output richiesto dall'utente.

Nel secondo caso (navigazione all'interno del Menu dei Glifi), ogni volta che l'utente effettua una scelta (selezionando una caratteristica di un glifo piuttosto che un'altra), il controller comunica la scelta a `model.js` (che salva la scelta nella forma opportuna all'interno dei suo array) che a sua volta chiama `DAO.js` per ottenere i dati richiesti. La maggior parte delle query effettuate da `DAO.js` servono ad ottenere i glifi richiesti dall'utente ogni volta che effettua una scelta di una caratteristica nel Menu dei Glifi, pertanto in questa libreria ci sono gli array che contengono i dati per formulare query adeguate ad ogni scelta che l'utente potrebbe effettuare. Una volta composta la query, `DAO.js` la passa a `DataSource.php` che effettua l'interrogazione e restituisce i dati ottenuti dal database. I dati vengono poi restituiti al DAO e da lì al model, che aggiorna i suoi array. Una volta che l'aggiornamento del model è completo il controller chiama l'aggiornamento della view, che visualizza i risultati ottenuti dal database.

Nel terzo caso (scelta di un glifo e suo trascinamento e gestione nel Display del Segno) la view si occupa della gestione del drag del glifo (evidenziazione del Display del Segno quando il glifo si trova su di essa, controllo sulle coordinate sulle quali è rilasciato il glifo, ecc.), il controller interviene solo quando il glifo è trascinato con successo sul Display del Segno, effettuando le operazioni necessarie a mostrare all'utente i glifi compatibili con i glifi inseriti fino a quel momento nel Display del Segno. Quest'operazione comporta di nuovo l'interazione con il model e il DAO per ottenere dati dal database, interazione già illustrata nel caso della navigazione all'interno del Menu dei Glifi.

2.3 Database

Come già detto nel Cap. 5, il database, nel SWift, contiene esclusivamente due tipi di dati.

- I **glifi**: singoli elementi che l'utente può utilizzare nella composizione del segno che vuole trascrivere. Essi sono circa 45000 e costituiscono una parte importante delle informazioni contenute nel database.
- Oltre i glifi è stata prevista la possibilità di memorizzare interi **segni**, che sarà sfruttata nel caso in cui gli utenti decidano di salvare le proprie trascrizioni utilizzando il salvataggio remoto.

La Fig. 7.7 riporta lo schema ER del database del SWift. Si possono notare tre entità: **glyphs** e **signs** contengono informazioni su glifi e segni, l'entità **components**, resa inevitabile dalla relazione n a m tra **glyphs** e **signs** associa ad ogni segno i glifi che lo compongono, tenendo ovviamente traccia anche delle loro coordinate spaziali all'interno del segno.

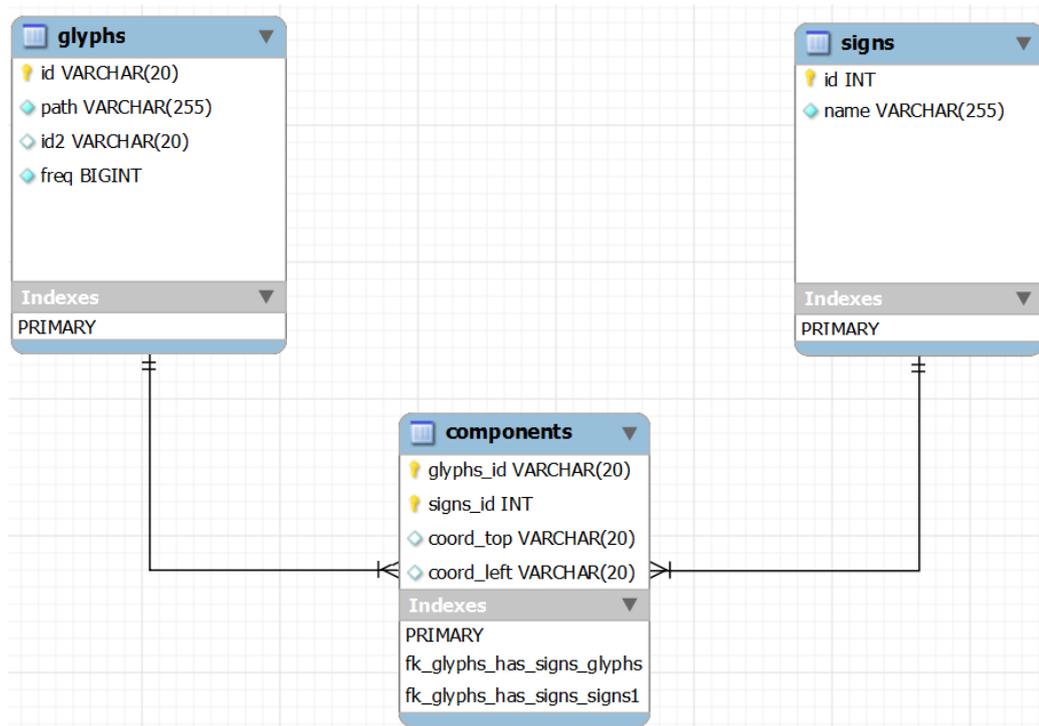


Figura 7.7: Schema ER del database del SWift.

GLYPHS

Tabella che contiene le informazioni sui glifi del SW.

Nome	Tipo	Info	Descrizione
id	VARCHAR(20)	PK	Id secondo l'ISWA 2008, del glifo.
path	VARCHAR(255)	NN	Path dell'immagine png del glifo.
id_2	VARCHAR(20)		Id, lasciato per una seconda eventuale codifica ISWA, del glifo.
freq	BIGINT		Contatore della frequenza di utilizzo, viene incrementato di uno ogniqualvolta il glifo viene utilizzato.

Tab. 7.2: Definizione della tabella `glyphs` nel database del SWift.

SIGNS

Tabella che contiene le informazioni sui segni contenuti nel SWift (tipicamente inseriti con la funzione di salvataggio).

Nome	Tipo	Info	Descrizione
id	INT	PK AI	Id numerica auto-incrementale del segno.
name	VARCHAR(255)	NN	Il nome del segno, così come è stato definito dall'utente al momento del salvataggio.

Tab. 7.3: Definizione della tabella `signs` nel database del SWift.

COMPONENTS

Tabella che concretizza l'associazione n a m tra `glyphs` e `signs`, associando ad ogni segno i glifi che lo compongono.

Nome	Tipo	Info	Descrizione
glyphs_id	VARCHAR(20)	PK FK	Id del glifo che compone il segno.
signs_id	INT	PK FK	Id del segno.
coord_top	VARCHAR(20)	NN	Posizione sull'asse Y del glifo nel segno.
coord_left	VARCHAR(20)	NN	Posizione sull'asse X del glifo nel segno.

Tab. 7.4: Definizione della tabella `components` nel database del SWift.

Capitolo 8

Conclusioni

Il SWift ha tutte le carte in regola per diventare un editor per il SW di larga diffusione. L'applicazione è stata progettata in stretto contatto con il gruppo dell'ISTC-CNR, quindi è stata pensata per rispondere alle esigenze dei sordi, nel modo che loro ritengono essere il migliore. La sua validità andrà ancora testata sul campo, ovviamente, ma i primi risultati sono incoraggianti: l'obiettivo che mi ero posto, cioè velocizzare e rendere più piacevole ed intuitiva la composizione elettronica dei segni è stato raggiunto.

Molto ancora rimane da fare: Il SWift è un editor, ma il suo database è già stato programmato affinché, con le opportune modifiche alla logica di business dell'applicazione, essa possa offrire funzionalità più avanzate: potrebbe gettare le basi per un vero e proprio editor di testi in SW, o per un dizionario, o per un convertitore tra i vari ISWA. In un futuro forse non troppo lontano, si potrebbe sostituire l'intero sistema di ricerca con un sistema di riconoscimento e confronto tra immagini: gli utenti si limiterebbero a disegnare a mano libera il glifo desiderato, e il sistema riconoscerebbe il glifo e si preoccuperebbe del suo inserimento.

Capitolo 9

Glossario

Le tabelle seguenti contengono termini che è necessario conoscere per una corretta comprensione del mio elaborato. E' opportuno leggere attentamente il presente glossario prima di avventurarsi nella lettura degli altri capitoli, per evitare una frustrante esperienza di lettura.

Configurazione	Insieme di parametri che definisce la componente manuale di un segno.
Glifo	Unità grafica minima del SignWriting (può essere paragonata ad una lettera) che può rappresentare una configurazione della mano, un movimento o un'altra componente del segno. Per comporre un segno è necessaria l'unione di più glifi.
Grafica raster	Tipo di grafica in cui le immagini vengono descritte come una griglia di pixel opportunamente colorati.
Logica di business	La logica applicativa che rende operativo un programma.
Segno	Composizione di più glifi, rappresenta un'unità della lingua dei segni.
SignWriting (SW)	Il SignWriting è un sistema che usa simboli visuali (i glifi) per rappresentare configurazioni, movimenti ed espressioni facciali delle lingue dei segni. Può essere visto come un alfabeto: una lista di simboli usati per scrivere qualsiasi lingua dei segni del mondo.

Capitolo 10

Indice degli acronimi

AS	ActionScript
ASL	American Sign Language
DAO	Data Access Object
HTML	HyperText Markup Language
ISWA	International SignWriting Alphabet
IT	Information Technology
LIS	Lingua dei Segni Italiana
LSF	Langue des Signes Française
MVC	Model View Controller
PARC	Palo Alto Research Centre
PHP	PHP: Hypertext Processor
SLS	Sign Language Scent
SQL	Structured Query Language
SW	SignWriting
SWift	SignWriting improved fast transcriptor
SWML	SignWriting Markup Language
XHTML	eXtensible HyperText Markup Language
W3C	World Wide Web Consortium
WCAG	Web Content Accessibility Guidelines

Capitolo 11

Ringraziamenti

Questo lavoro è stato sviluppato nell'ambito del progetto “*Progetto FIRB-VISEL: E-learning, sordità, lingua scritta: un ponte di lettere e segni per la società della conoscenza*”, realizzato nell'ambito dei finanziamenti FIRB (Fondi per la Ricerca di Base) del MIUR.

Per maggiori informazioni: <http://www.visel.cnr.it>.

Ringrazio Maria De Marsico, per avermi seguito e consigliato, non solo sul mio lavoro per il SWift.

Ringrazio Claudia S. Bianchini, per avermi supportato e incoraggiato, e soprattutto per il suo fondamentale lavoro di classificazione. E'anche grazie a lei che il Swift, oggi, è una risorsa che potrebbe semplificare il lavoro e la comunicazione di tutti gli utenti del SW.

Ringrazio Elena Antinoro Pizzuto e il gruppo dell'ISTC-CNR per aver collaborato alla progettazione, validazione e testing delle interfacce e della logica di business dell'applicazione, senza il loro aiuto il SWift sarebbe stata una delle tante applicazioni progettata per i sordi senza tenere conto delle loro reali esigenze.

Ringrazio Clelia Fiori (reperibile a ranocchietta.87@hotmail.it) per aver realizzato il Puppet del SWift, senza il suo lavoro l'applicazione avrebbe avuto un volto molto meno umano.

Bibliografia

ANTINORO PIZZUTO E., CHIARI I., ROSSINI P. (2009): *Strumenti per la tradizione della Lingua dei Segni Italiana: critiche e proposte per una ricerca responsabile*. In stampa in Massariello G., Dal Maso S. *Atti del congresso SLI 2009*. Roma (Italy), Bulzoni, (2009). (Pizzuto et al., 2009).

BEBIAN R. A. A. (1825): *Mimographie ou Essais d'écriture mimique propre à régulariser le langage des sourds-muets*. Paris (France), L. Colas, 1825.

BIANCHINI C. S., GIANFREDA G., DI RENZO A., LUCIOLI T., PETITTA G., PENNACCHI B., LAMANO L., ROSSINI P. (2010): *Écrire une langue sans forme écrite: réflexions sur l'écriture et la transcription de la Langue des Signes Italienne (LIS)*. In *XXIV colloque du CerLiCO: Transcrire, Écrire, Formaliser, Université de Tours, 4-6-2010*.

BIANCHINI C. S. (2011): *Émergence d'un système d'écriture pour les Langues des Signes et réflexions métalinguistiques conséquentes*. Université de Paris8 - Università degli Studi di Perugia. Tesi di dottorato in corso (prevista 2011).

BONNAL-VERGÈS F. *Sémiogénèse de la Langue des Signes Française (LSF)*. Limoges (France), Editions Lambert-Lucas, in stampa.

CORAZZA S., VOLTERRA V. (2004): *Introduzione*. In Volterra V. (a cura di). *La lingua dei segni italiana: la comunicazione visivo-gestuale dei sordi (Nuova Edizione)*. Bologna (Italy), il Mulino, 2004: 9-21.

CUXAC C. (2000): *La langue des signes française (LSF): les voies de l'icônicité*. Paris (France), Ophrys, 2000: 15-16.

DI RENZO A. (2008): *Esperienze e riflessioni sui metodi di trascrizione della LIS*. In Bagnara C., Corazza S., Fontana S., Zuccalà A. (eds.). *I Segni parlano. Prospettive di ricerca sulla Lingua dei Segni Italiana*. Milano (Italy), Franco Angeli, 2008: 159-170.

FAJARDO I., ARFÉ B., BENEDETTI P., ALTOÉ G.M. (2008a): *Hyperlink format, categorization abilities and memory span as contributors to deaf users hypertext access*. In *Journal of Deaf Studies and Deaf Education* 13 (2). (2008): 241–256.

FAJARDO I., PARRA E., CAÑAS J. J., ABASCAL J., LÓPEZ J. M. (2008b): *Web information search in sign language*. In *Symposium Informal Learning on the Web: Individual Differences and Evaluation Processes, XXIX International Congress of Psychology*. Berlin (Germany), 2008: 20-25.

FAJARDO I., VIGO M., SALMERÒN L. (2009): *Technology for supporting web information search and learning in Sign Language*. In *Interacting with Computers*. Elsevier, (2009).

GIANFREDA G., PETITTA G., BIANCHINI C. S., DI RENZO A., ROSSINI P., LUCIOLI T., PENNACCHI B., LAMANO L. (2009): *Dalla modalità faccia-a-faccia ad una lingua scritta emergente: nuove prospettive su trascrizione e scrittura della Lingua dei Segni Italiana (LIS)*. In Furiassi C., Consani C., Sanga G., Sornicola R. (eds.). *Atti del IX Congresso Internazionale di Studi della Associazione Italiana di Linguistica Applicata (AIItLA), "Oralità / Scrittura – In memoria di Giorgio Raimondo Cardona"*, Pescara, 19-20 febbraio 2009. Perugia (Italy), Guerra Edizioni, 2009.

KAIBEL A. GROTE K., KNOERZER K., SIEPRATH H., KRAMER F. (2006): *Hypertext in Sign Language*. In *9th ERCIM "User Interfaces for All" Workshop Adjunct Proceedings*.
<http://www.ui4all.gr/workshop2006/publications/adjunctproceedings.html>.

LAMBERT L. M. ABBÈ (1865): *Le langage de la physionomie et du geste mis à la portée de tous, suivi d'une méthode courte facile et pratique d'enseignement des sourds-muets illettrés qui sont hors des institutions spéciales et des élèves arriérés de ces mêmes écoles*. Paris (France), J. Lecoffre, 1865.

MOODY B., GIROD M., VROUC'H A., GALANT P., GACHE C., MOUSION S.. (1997): *La langue des signes: dictionnaire bilingue LSF / français*. Paris (France), IVT - International Visual Theatre éditions, 1997.

NAMATAME M., NISHIZAKI Y., KITAJIMA M. (2007): *Improving usability of web pages for hard-of-hearing persons: an investigation of the utility of pictograms*. HCI International Posters, (2007): 176–180.

NEWKIRK D. (1989): *SignFont handbook*. Bellevue (Washington, USA), Ed-

mark corp., 1989.

ONG W. J. (1982): *Orality and literacy. The technologizing of the word*. Methuen, London (UK) – New York (USA), 1982.

PIROLI P., CARD S. K. (1999): *Information foraging. Psychological Review*. (1999): 643–675.

PIZZUTO E., ROSSINI P., RUSSO T. (2006): *Representing signed languages in written form: questions that need to be posed*. In *Proceedings of the Second Workshop on the Representation and Processing of Sign Languages: lexicographic matters and didactic scenario, International Conference on Language Resources and Evaluation - LREC 2006 Genoa, May 28, 2006*. ELRA editors, 2006: 1-6.

PRILLWITZ S., LEVEN R., ZIENERT H., HANKE T., HENNING J. (1989): *Hamburg notation system for sign languages. An introductory guide, Ham-NoSys version 2.0*. Hambourg (Germany), Signum Press, 1989.

PULLEYBLANK D. (1987): *Yoruba*. In Comrie B. (ed). *The World's Major Languages*. London (UK), Croom Helm, 1987: 971-990.

RADUTZKY E. (2001): *Dizionario bilingue elementare della Lingua Italiana dei Segni: oltre 2500 significati*. Roma (Italy), Edizioni Kappa, 2001.

ROMEO O. (1991): *Dizionario dei segni: La lingua dei segni in 1400 immagini*. Bologna (Italy), Zanichelli, 1991.

SICARD ABBÈ. (1808): *Dictionnaire des Signes..* Paris (France), 1808.

STOKOE W. C. (1960): *Sign language structure: an outline of the visual communication systems of the American deaf*. In *Studies in Linguistics*. Silver Spring (Maryland, USA), Linstok Press, 1960.

WORLD HEALTH ORGANIZATION. (2006): *Deafness and hearing impairment*. <http://www.who.int/mediacentre/factsheets/fs300/en/index.html>.