

POLITECNICO DI MILANO
FACOLTÀ DI INGEGNERIA

CORSO DI LAUREA IN INGEGNERIA INFORMATICA



MultiAble: nuovi criteri di accessibilità
multimodale a informazioni in rete

Relatore: Prof. Licia SBATTELLA
Correlatore: Prof. Thimoty BARBIERI

Tesi di laurea di:
Ferdinando Carella
Matr. 624260

ANNO ACCADEMICO 2003–2004

A mio padre e a mia madre.

Indice

Sintesi del contenuto della tesi	1
Introduzione	3
1 Il problema dell'accessibilità nell'IT	7
1.1 Accessibilità rispetto alla disabilità	8
1.2 Accessibilità rispetto al dispositivo	10
1.3 Accessibilità rispetto al contesto	11
1.4 ICF (International Classification of Functioning)	12
1.5 Prodotti e strumenti di ausilio ai disabili	13
1.5.1 Strumenti integrati nei sistemi operativi	13
Windows	14
Linux	14
Mac OS X	15
1.5.2 Tastiere, stampanti e display braille	16
1.5.3 Sintesi e riconoscimento vocale	16
1.5.4 Browser vocali	18
1.5.5 Screen reader: Jaws, Hal, Windows-Eye...	18
1.5.6 Browser testuali: Lynx, Links, w3m	19
1.6 Le direttive sull'accessibilità del W3C: WAI	20
1.7 Direttive del governo USA: Section 508	22
1.8 Validatori di accessibilità	24
1.9 Uso e comprensione del linguaggio	25

Formula di Flesch	26
Indice GulpEASE	26
Analisi semantica latente	27
1.10 Accessibilità e multimodalità	27
InkML	29
EMMA	29
DPF	29
XForms	30
Linguaggi per interfacce vocali	30
1.11 Lavori correlati	32
1.11.1 La comunicazione multimodale	35
1.11.2 Semantica e adattabilità	35
2 Linee guida per l'estensione delle direttive sull'accessibilità	37
2.1 Usabilità e accessibilità	38
2.2 Guida pratica all'accessibilità: estensione delle WCAG	39
2.2.1 Separazione della struttura dai contenuti	40
La chiave: XML	41
2.2.2 Layout di pagina e accesso con screen reader	42
Non usare le tabelle	43
2.2.3 Topologia della pagina e semantica	45
Blocchi principali: menu e contenuto	46
2.2.4 Uso del colore e del carattere	47
2.2.5 Uso delle immagini	48
2.2.6 Tabelle HTML e problema della serializzazione	50
Esempio	52
2.2.7 Form e interattività	55
Esempio	56
Accesso multimodale alle form	59
2.2.8 Arricchire la pagina con metainformazioni	60
Esempio	60

2.2.9	Navigazione senza mouse	61
2.2.10	Usare un linguaggio accessibile	63
2.3	Web ontology e web semantico	65
2.3.1	Esempio di documenti “index” e “contents”	68
2.4	Usare i validatori automatici	72
2.5	Accessibilità multimodale in breve	73
3	Un caso di studio: il progetto MultiAbile	75
3.1	Le finalità e gli obiettivi	75
3.2	Struttura del sito	77
3.3	Struttura delle pagine	78
3.4	Profilazione dell’utente	80
3.5	L’interfaccia vocale	81
3.6	La piattaforma di e-learning	82
3.6.1	Lo standard SCORM	85
	Content Aggregation Model	87
	Runtime environment	88
	Il datamodel	90
3.6.2	Fruizione dei corsi	91
	L’interfaccia grafica	91
	L’interfaccia vocale	95
	Identificazione dell’utente	96
3.7	Accesso alle informazioni con syndication	96
3.8	L’esperienza multimodale	98
3.8.1	Utilizzo dei gesti	98
3.8.2	Feedback tattili: la percezione spaziale	100
	Topologia della pagina	101
	Mappa tattile interattiva	101
	Percorso guidato	103
	Esplorazione con touch screen	103
	Tablet PC e PDA	104

3.9	Validazione e valutazione di accessibilità	104
4	Realizzazione di un trasduttore vocale	106
4.1	Requisiti della VUI (Voice User Interface)	107
4.2	Il problema del transcoding	108
4.3	Transcodifica attraverso l'uso delle annotazioni	109
4.4	Implementazione del proxy multimodale	111
4.4.1	Utilizzo di Cocoon come framework di base	111
4.4.2	L'architettura software	114
4.5	Richiesta dell'utente	116
4.6	Inoltro della richiesta da parte del proxy	117
4.7	Transcodifica	117
4.7.1	“Pulitura” della pagina web attraverso TIDY	117
4.7.2	Estrazione dei file index e contents	118
4.7.3	Correzione delle URL	119
4.7.4	Definizione di un XML intermedio	119
4.7.5	Prima trasformazione	120
4.7.6	Seconda trasformazione	120
	Separazioni in sottodialoghi	121
	Sommaro della pagina	121
	Separazione in frasi	122
	Le liste	124
	I collegamenti	125
4.7.7	Serializzazione delle tabelle	126
4.7.8	Transcodifica delle form	131
4.7.9	Menu di pagina e menu del sito	132
4.8	I comandi di navigazione	133
	Navigazione nella pagina	133
	Browser vocale	134
4.9	Fruizione della pagina web da parte degli utenti	135
4.9.1	L'ambiente di sviluppo: IBM Voice SDK	135

4.9.2	L'ambiente di produzione: Loquendo VoxNauta . . .	136
5	Valutazione del modello del sistema realizzato	139
5.1	Valutazione soggettiva	140
5.1.1	Metodologie di analisi	140
5.2	L'interfaccia grafica	142
5.3	L'interfaccia vocale	143
5.4	L'interfaccia multimodale	145
5.4.1	L'esplorazione tattile	146
5.4.2	Feedback audio	147
5.4.3	Uso dei gesti e delle scorciatoie	148
6	Conclusioni e sviluppi futuri	150
6.1	Considerazioni finali	151
6.2	Possibili implementazioni future	152
6.2.1	Per lo sviluppatore: strumenti di authoring	154
6.2.2	Per l'utente: un browser multimodale	154
6.2.3	Un linguaggio formale per i feedback tattili	155
	Bibliografia	156

Sintesi del contenuto della tesi

In questa tesi affronto il problema dell'accessibilità dei siti web con particolare attenzione nei confronti delle persone con disabilità sensoriali e cognitive.

La tesi fa parte dei progetti di ricerca del laboratorio ArcsLab del Politecnico di Milano.

L'obiettivo di partenza era quello di ricavare una linea guida pratica, un'insieme di tecniche e di metodologie innovative, da applicare allo sviluppo dei siti web e in generale ai sistemi per la divulgazione di informazioni e di contenuti attraverso la rete internet.

Dopo una analisi del problema dell'accessibilità e delle soluzioni teoriche illustrate nelle linee guida del W3C e in studi correlati, ho esaminato i principali sistemi di ausilio cui solitamente ricorrono i disabili per utilizzare il computer, allo scopo di capire meglio come tale problema venga affrontato e per valutare in qualche modo l'efficacia di tali strumenti.

A seguito di tale studio ho concluso che non è sufficiente applicare semplicemente alcune regole teoriche per ottenere un sito accessibile, ma è necessario perseguire obiettivi di usabilità assieme all'accessibilità e per farlo bisogna capire prima di tutto le esigenze dell'utente finale. Ho quindi dedotto che il modo migliore per ottenere ciò è quello di adattare la presentazione dei contenuti alle caratteristiche e alle capacità dell'utente e degli ausili utilizzati.

Ho inoltre ipotizzato che la strada migliore per ottenere un elevato livello di accessibilità e di usabilità fosse quella della multimodalità, cioè

la possibilità da parte dell'utente di interfacciarsi ad un sistema attraverso diversi strumenti e metodologie di input e di output utilizzati anche parallelamente.

Per ottenere questa adattabilità ho definito delle linee guida e un insieme di tecniche pratiche per la realizzazione dei siti e ho elaborato un sistema basato su annotazioni semantiche per aggiungere alle pagine di un sito delle metainformazioni utili alla trasformazione automatica dell'interfaccia web in una interfaccia multimodale. Ho realizzato inoltre un transcoder che mostra appunto come un sito web che segua tali linee guida e che implementi il sistema delle annotazioni possa essere convertito in una interfaccia vocale accessibile e usabile.

L'aspetto innovativo della ricerca è l'enfasi data alla multimodalità come mezzo principale per la soluzione dei problemi legati all'accessibilità: sfruttare il canale vocale e tattile per la navigazione e l'individuazione della struttura delle pagine e del sito, utilizzare mezzi alternativi quali l'uso del telefono per accedere alle informazioni, oppure potenziare quelli esistenti, come ad esempio l'uso dei gesti della mano durante la navigazione con il browser, sono le caratteristiche principali della soluzione proposta.

In seguito ho applicato queste tecniche durante la realizzazione del sito "Multiabile", una piattaforma di e-learning accessibile anche ai disabili. Nel progetto, che aveva tra gli obiettivi il raggiungimento di un'elevata accessibilità e usabilità multimodale, sono state sperimentate anche vari sistemi di interfacciamento (vocale, gestuale, tattile).

Grazie al coinvolgimento di vari utenti anche in situazione di disabilità, ho potuto effettuare una analisi soggettiva del livello qualitativo del sito in termini di accessibilità e di usabilità rispetto alle diverse modalità di interfacciamento ottenendo in primo luogo una conferma della bontà delle linee guida e delle tecniche elaborate. Inoltre ho avuto conferma dell'ipotesi secondo cui la multimodalità e l'adattabilità dei contenuti rispetto agli utenti e ai canali di comunicazione siano in effetti le chiavi di successo per l'ottenimento di un sito veramente accessibile ed usabile.

Introduzione

Internet e in particolare il web è uno strumento che, almeno secondo il principale dei suoi autori Tim Barnes-Lee, deve essere utilizzabile da chiunque, dovunque e con il più ampio spettro di mezzi e tipologie di interazione possibile.

La cosiddetta “guerra dei browser” durante la quale sembra che le software house abbiano fatto di tutto per interpretare a proprio favore il linguaggio HTML e gli standard indicati dal consorzio W3C per rendere incompatibili i siti per i browser degli avversari, ha caratterizzato un lungo periodo dopo la nascita del web e ha purtroppo portato alla diffusione di innumerevoli esempi di mancato rispetto dei principi indicati da Barnes-Lee. Basti pensare ad esempio a tutti quei siti nei quali si viene accolti con frasi che cominciano con “sito ottimizzato per...” e si viene invitati a cambiare browser, risoluzione dello schermo se non addirittura sistema operativo!

Negli ultimi anni, forse grazie anche alle politiche nazionali e comunitarie, sembrerebbe che questa tendenza stia cambiando e che sempre di più si cerchi di recepire e mettere in pratica i suggerimenti e le linee guida del W3C per realizzare siti cosiddetti “accessibili” a chiunque, dovunque e con ogni mezzo. Molto interessati a questo aspetto sono soprattutto le persone disabili, soprattutto non vedenti e ipovedenti ma anche coloro che hanno difficoltà cognitive o handicap fisici e che vogliono accedere alle informazioni della rete e hanno perciò la necessità di utilizzare strumenti di supporto quali sintetizzatori e riconoscitori vocali, barre braille, screen

reader e altri dispositivi di input speciali.

Inoltre l'enorme diffusione di dispositivi quali computer palmari, cellulari WAP, GPRS, UMTS e non ultima la televisione digitale ha messo di fronte allo sviluppatore il problema di dover presentare i contenuti attraverso applicazioni molto diverse tra loro e nuovi strumenti di interfacciamento con l'utente che devono comunque consentire a tutti di poter fruire correttamente dei contenuti che vengono proposti.

Con questa tesi si affronta appunto il problema dell'accesso multimodale alle informazioni presenti in rete, cioè attraverso diverse tipologie di interfaccia utente utilizzate anche in contemporanea.

In particolar modo si affronterà il problema della realizzazione di interfacce vocali in grado di consentire all'utente di ascoltare, invece che di leggere i contenuti e di dare i comandi alla macchina attraverso la propria voce piuttosto che con l'utilizzo di tastiera e mouse.

Il punto di partenza di questo studio saranno le linee guida sull'accessibilità proposte dal consorzio W3C attraverso il WCAG 1.0 e 2.0 e gli standard proposti sempre dal W3C per quanto riguarda l'accessibilità multimodale e quindi in particolare i linguaggi XHTML, VoiceXML, EMMA, XForms.

Nella prima parte verrà dapprima introdotto il problema dell'accessibilità con particolare attenzione ai disabili. Per questa categoria di utenti verranno descritti i principali dispositivi e strumenti di supporto per l'utilizzo del computer e per la fruizione delle informazioni.

In seguito verranno appunto descritte le linee guida principali del W3C sull'accessibilità e sulla multimodalità, gli standard proposti che saranno analizzati criticamente in modo da individuare quelli che meglio possono aiutare a risolvere il problema dell'accessibilità soprattutto per i disabili.

L'obiettivo di questa analisi sarà quello di ricavare una guida pratica, che vada quindi oltre le normali linee guida e i suggerimenti, per la realizzazione di siti internet veramente accessibili in modo multimodale.

Nella terza parte verrà illustrato, come caso di applicazione e di studio,

il progetto Multiabile nel quale si è cercato di realizzare un sistema multimodale adatto soprattutto ad alcune categorie di disabili. Nell'ambito dello stesso progetto si vedrà come è stata affrontata la sperimentazione di nuove modalità di interazione, nel tentativo di contribuire all'abbattimento di alcune barriere all'accesso delle informazioni cui si devono scontrare quotidianamente i disabili che vogliono accedere al web.

Al progetto (tuttora in corso di attuazione) partecipano:

- Eurocons: consorzio europeo e per la formazione, ricerca e sviluppo
- ArcsLab (Politecnico di Milano): Adaptable, Relational and Cognitive Software Environments
- Web Learn s.r.l.: società specializzata in e-learning e sviluppo di tecnologie per la formazione a distanza
- Mito: agenzia formativa

Oltre alle società ed enti menzionati, sono coinvolti nel progetto l'Istituto dei Ciechi di Milano e l'Associazione Italiana Dislessia Onlus le cui consulenze hanno portato e stanno tuttora portando un prezioso apporto alla ricerca, sperimentazione e alla verifica diretta di nuove modalità di interazione con il web oltre che al miglioramento del supporto a quelle tradizionali.

Personalmente faccio parte del gruppo di lavoro di WebLearn che si è occupato interamente dello sviluppo del sito e che è composto oltre che dal sottoscritto, dall'Ing. Marco Ferra e dall'Ing. Timothy Barbieri, quest'ultimo il quale si è occupato prevalentemente del coordinamento del progetto. Per quanto mi riguarda, mi sono occupato dello sviluppo dell'LCMS della piattaforma e-learning, dell'interfaccia vocale e del sistema di transcodifica automatica delle pagine HTML che sarà illustrato nel capitolo 4.

Uno dei componenti fondamentali del progetto Multiabile, rispetto al problema della multimodalità, è il sistema di transcodifica automatica delle pagine HTML in documenti VoiceXML al fine di realizzare una in-

terfaccia vocale efficace per siti web esistenti, usabile anche attraverso il telefono.

La quarta parte della tesi sarà appunto dedicata alla descrizione di questo sistema di transcodifica. Vengono ripresi alcuni risultati ottenuti nel contesto dei progetti di ricerca effettuati all'interno dell'ArcsLab del Politecnico di Milano come ad esempio Violin (Voice On-Line), un progetto di tesi del che ha appunto questa funzionalità e Ceriani[24] che ha affrontato il problema della comunicazione multimodale e della comprensibilità dei testi nell'ambito del problema dell'accessibilità. In particolare si vuole migliorare la cosiddetta "navigazione assistita" che consenta cioè all'utente non vedente (o che semplicemente non può vedere in quel momento perchè stà usando il telefono e non un browser visuale) di percepire la struttura della pagina e se possibile dell'intero sito e di navigare agevolmente e in modo efficiente attraverso comandi vocali o di altro tipo.

Nella parte finale della tesi verrà fatta un'analisi dei risultati ottenuti. Si vuole verificare se un sito internet realizzato secondo i criteri proposti sia effettivamente più semplice da usare anche attraverso dispositivi di interazione non tradizionali. Il sito Multiabile sarà usato come oggetto per tale verifica.

Affinché l'analisi dei risultati sia quanto più possibile obbiettiva, si cercherà di coinvolgere nella verifica funzionale del prototipo diverse tipologie di utente: dagli utenti esperti a quelli senza background tecnologico/informatico, con o senza disabilità visive o di altro genere.

Al termine dell'analisi fatta verranno tratte le conclusioni e proposti alcuni spunti per possibili sviluppi futuri.

Capitolo 1

Il problema dell'accessibilità nell'IT

Molto spesso quando si parla del problema dell'accessibilità dei siti internet ed in generale delle informazioni in rete, si pensa principalmente ai disabili.

Le stesse linee guida sull'accessibilità proposte dal gruppo di lavoro del W3C, le cosiddette WCAG (Web Content Accessibility Guidelines) [71] [15], hanno come obiettivo principale quello di consentire l'accesso da parte dei disabili.

Si tratta senza dubbio della categoria di persone che incontra le maggiori difficoltà nell'accedere ai servizi e alle informazioni normalmente fruibili sul web, tuttavia non va dimenticato che chiunque può andare incontro a problemi di accessibilità indipendentemente dal fatto di avere una qualche disabilità o meno.

Il problema dell'accessibilità può essere visto attraverso diversi punti di vista che sono essenzialmente tre: quello della disabilità, quello legato ai dispositivi di input/output utilizzati, quello legato al contesto in cui si opera.

Nelle prossime sezioni verranno illustrate brevemente le principali situazioni per le quali si deve affrontare questo problema, successivamen-

te si approfondirà il problema della disabilità che può essere preso come punto di riferimento per lo sviluppo di soluzioni adatte anche a chi disabile non è.

1.1 Accessibilità rispetto alla disabilità

Rendere accessibili i contesti alle persone in situazioni di disabilità è una finalità di ampio interesse umanitario e scientifico. Le disabilità maggiormente coinvolte nel problema sono quelle di tipo sensoriale, cognitivo e motorie. Persone non vedenti, ipovedenti, sorde, sordo-cieche e ancora dislessiche o persone con problemi cognitivi più o meno gravi usando il web incontrano ancora molti e necessitano a volte di strumenti di ausilio o richiedono sistemi adattativi per superare almeno in parte gli ostacoli dati dalla loro disabilità. A volte queste disabilità si presentano combinate tra loro e allora non sempre può essere sufficiente affiancare semplicemente le tecnologie usate per ciascuna disabilità indipendente, ma è necessario trovare soluzioni ad-hoc.

Il W3C ha tracciato alcuni scenari in cui persone con disabilità utilizzano il web, indicando quali sono le disabilità che causano o possono causare un problema di accessibilità e come tecnologie assistive e strategie adattative possano essere d'aiuto [33].

Di seguito vengono illustrate brevemente le principali disabilità e le barriere all'accessibilità. Vengono inoltre indicate le tecnologie assistive o adattative generalmente usate.

cecità o ipovisione : i contenuti per il web sono pensati fondamentalmente per un presentazione visuale e multimediale. Chi è affetto da cecità, ipovisione, daltonismo, ne risulta ovviamente svantaggiato e ricorre solitamente a lettori di schermo, display braille, browser testuali ecc... con i quali comunque alcune informazioni rimangono in qualche modo inaccessibili (grafici, immagini, animazioni ecc...) Può essere utile avere a disposizione una interfaccia vocale che ripro-

duca nel modo più fedele possibile quella grafica proposta dal browser e che consenta di impartire i comandi per mezzo della propria voce. Gli elementi puramente visuali come ad esempio le immagini devono avere anche una rappresentazione alternativa per consentirne la percezione attraverso tali strumenti. La presentazione dei contenuti deve essere adattabile alle esigenze dell'utente, ad esempio cambiando i colori, la dimensione dei caratteri, ecc. . .

sordità : come per le immagini, bisogna fornire testi alternativi per i componenti audio e audio/video distribuiti via web. In realtà il problema principale nasce per chi è sordo dalla nascita o che comunque lo è diventato prima di sviluppare completamente il linguaggio e/o la capacità di lettura. La sordità in tal caso può portare con sé anche problemi cognitivi. Può essere difficile comprendere testi anche relativamente semplici e l'uso dei testi alternativi per contributi audio e audio/video possono non essere sufficienti. È importante in questi casi, nella redazione, poter utilizzare un linguaggio semplice e facilmente comprensibile.

Le persone sorde dalla nascita, che spesso hanno difficoltà anche a comunicare con la voce, talvolta prediligono utilizzare il LIS, un linguaggio basato su segni fatti con le mani e con il viso: un linguaggio visuale che potrebbe sposarsi bene con l'ambiente web a patto di avere un sistema in grado di tradurre i testi con opportune animazioni.¹

mutismo o difficoltà di espressione : persone non in grado di parlare correttamente possono avere difficoltà nell'utilizzo di sistemi basati sul riconoscimento vocale. È necessario avere sistemi di input alternativi.

cognitive/neurologiche : persone affette da dislessia, difficoltà di appren-

¹Questo aspetto è uno degli sviluppi futuri auspicabili per il progetto Multiabile che sarà illustrato nel capitolo 3

dimento o patologie differenti, possono avere difficoltà di comprensione del linguaggio o delle immagini. Spesso le persone affette da dislessia utilizzano i sintetizzatori vocali per aiutarsi nella lettura, mentre chi ha problemi di elaborazione dei suoni o delle immagini può essere aiutato leggendo un testo alternativo che spieghi un filmato. È importante utilizzare un linguaggio semplice per i testi, scegliere in modo accurato le immagini e le tecnologie usate per la presentazione. Evitare quindi testi lampeggianti, animazioni veloci, creando una struttura generale delle pagine che sia chiara e facilmente comprensibile.

fisiche : debolezza, limitato controllo muscolare, limitata sensazione tattile, mancanza di arti superiori o scarsa mobilità, possono richiedere l'utilizzo di tastiere speciali o dispositivi alternativi. I siti dovrebbero essere pensati quindi anche per agevolare l'uso di diversi dispositivi di input.

miste : non è raro che alcune disabilità si presentino contemporaneamente, ad esempio come detto chi è sordo può essere anche muto e quindi avere difficoltà cognitive e di espressione. In questi casi sono molte le difficoltà cui va incontro chi vuole affrontare il problema di accedere alle informazioni in rete ed è opportuno studiare strategie per combinare le tecniche adattative e gli ausili per facilitare questo compito.

legate all'età : è un caso di disabilità "acquisita" col tempo e in genere coinvolge in un modo più o meno grave varie *patologie*: ipovisione, sordità, difficoltà di memoria e difficoltà di attenzione.

1.2 Accessibilità rispetto al dispositivo

La diffusione di dispositivi e di software diversi dal PC e dal normale browser visuale, porta con se la necessità di adattare la presentazione dei

contenuti anche a questi nuovi sistemi oltre a dover tenere in considerazione la possibilità da parte degli utenti di utilizzare dispositivi di input e output diversi dai classici tastiera, mouse e schermo del computer.

Il problema dell'accessibilità è quindi legato all'uso di interfacce vocali, di penne ottiche, di schermi piccoli e a bassa risoluzione, di bande di trasmissione limitate ecc. . .

Nasce il problema della multicanalità e della multimodalità di interfacciamento. Problemi che si possono risolvere principalmente attraverso sistemi di delivery delle informazioni in grado di adattare i contenuti in base anche ai dispositivi e alle interfacce utente.

1.3 Accessibilità rispetto al contesto

La diffusione dei dispositivi diversi dal PC, soprattutto quelli mobili, ha portato le persone a collegarsi e quindi a richiedere le informazioni dalla rete in contesti ambientali "difficili".

A questa categoria appartengono gli utenti che navigano in situazioni in cui non possono adoperare le mani (per esempio perché stanno guidando l'automobile o perché stanno svolgendo lavori di precisione che richiedono l'uso delle mani), quelli che si collegano stando in ambienti particolarmente rumorosi e hanno a che fare con una interfaccia vocale, oppure ancora coloro che si trovano in ambienti molto luminosi che possono rendere poco visibile lo schermo.

Per questo tipo di situazioni, la cura dell'accessibilità consiste nel realizzare pagine navigabili indipendentemente dal dispositivo di input utilizzato, dotate di contenuti alternativi o comunque adattabili al canale sensoriale principale a cui si rivolge il contenuto. Oppure ancora richiede una creazione ad-hoc dell'interfaccia e quindi una flessibilità della presentazione dei contenuti che consenta questo adattamento.

1.4 ICF (International Classification of Functioning)

Le disabilità sono molteplici e come si è visto non sono l'unico fattore che influenza l'accessibilità delle informazioni. Appare quindi chiaro che per affrontare il problema è necessario conoscere, classificare e formalizzare in qualche modo quali sono gli ostacoli cui possono andare incontro i potenziali utenti di un sito web per capire quali potrebbero essere le difficoltà ad eccedere ai servizi distribuiti in questo modo.

L'ICF (International Classification of Functioning, Disability and Health) [4] è il nuovo strumento elaborato dall'Organizzazione Mondiale della Sanità (OMS) nel 2002 sulla base dell'ICIDH-2 del 1980 per descrivere e misurare la salute e le disabilità della popolazione. Esso cerca di cogliere e classificare ciò che può verificarsi in associazione a una condizione di salute, cioè le “compromissioni” della persona o, in generale il suo “funzionamento”.

È importante sottolineare che l'ICF non è una classificazione delle persone: è una descrizione delle caratteristiche della salute delle persone all'interno del contesto delle loro situazioni di vita individuale e degli impatti ambientali. Pertanto non si tratta di una classificazione che riguarda soltanto le condizioni di persone affette da particolari anomalie fisiche o mentali, ma è applicabile a qualsiasi persona che si trovi in qualsiasi condizione di salute, dove vi sia la necessità di discuterne lo stato a livello corporeo, personale o sociale.

L'ICF è considerato una vera e propria rivoluzione della percezione delle disabilità, che tiene conto per la prima volta anche di fattori ambientali, può essere preso in considerazione per creare un modello di descrizione dell'utente che consenta di risolvere il problema dell'accessibilità in modo più adeguato rispetto proprio alle caratteristiche della persona.

Tale modello infatti tiene conto e mette in relazione diversi aspetti, tra cui:

- *le funzioni corporee*: mentali, sensoriali, della voce e dell'eloquio ecc. . .
- le strutture corporee: sistema nervoso, occhio, orecchio e strutture collegate, strutture legate alla voce e all'eloquio, strutture collegate al movimento ecc. . .
- *attività e partecipazione*: apprendimento e applicazione della conoscenza, comunicazione, mobilità, compiti e richieste di carattere generale ecc. . .
- *i fattori ambientali*: prodotti e tecnologia, ambiente naturale e cambiamenti apportati dall'uomo all'ambiente, servizi sistemi e politiche ecc. . . Caratteristiche che possono avere impatto sulle prestazioni dell'individuo in un determinato contesto

1.5 Prodotti e strumenti di ausilio ai disabili

Molti sono i prodotti hardware e software di ausilio all'utilizzo del computer e delle applicazioni da parte delle persone con disabilità. Lo studio di tali dispositivi e di come vengono utilizzati può dare indicazioni sul modo con cui gli utenti si interfacciano con la macchina, riuscendo così a sopperire almeno in parte le proprie difficoltà.

Tali indicazioni poi possono dare spunti per migliorare il modo in cui le informazioni presenti in rete vengono redatte, organizzate e infine presentate all'interno di una interfaccia per l'utente.

1.5.1 Strumenti integrati nei sistemi operativi

Il primo ostacolo da superare è di tipo tecnologico/informatico. Nel caso in cui si voglia accedere alle informazioni attraverso un PC il primo passo è quello di imparare ad usarlo il che non sempre è facile soprattutto per un disabile.

Si pensi ad esempio ad un non vedente: tutti i moderni sistemi operativi fanno della grafica l'elemento essenziale dell'interfaccia. Icone, finestre, barre dei menu ecc... e la possibilità di fare quasi tutto con il solo mouse sono le caratteristiche principali di tali interfacce. Inutile dire che per un non vedente un'interfaccia simile oltre ad essere perfettamente inutile è anche inutilizzabile senza un supporto adeguato. Per alcuni non vedenti il sistema operativo DOS risultava migliore di qualsiasi altro sistema attuale.

Le interfacce grafiche di tutti i moderni sistemi operativi sono altamente configurabili e consentono a persone che abbiano semplicemente problemi di ipovisione o daltonismo di cambiare i parametri di visualizzazione, ottenendo subito beneficio.

Windows Alcuni tool integrati che possono essere d'aiuto sono:

- un *ingranditore di schermo* che consente di zoomare una parte di schermo sotto al cursore del mouse
- una *tastiera grafica* da usare con il mouse o altro dispositivo di puntamento adatto a persone con disabilità motoria.
- sistema di *sintesi e riconoscimento vocale*, utilizzabile all'interno delle applicazioni.

Linux Anche chi utilizza il sistema operativo Linux, per il quale è molto più recente l'introduzione dell'interfaccia grafica, ha a disposizione diversi strumenti di ausilio all'utilizzo del PC.

Il progetto GNOME Accessibility[26] per il desktop comprende vari ausili tra i quali si possono citare:

gnopernicus uno screen reader e un ingranditore di schermo che possono essere utilizzati con qualsiasi applicazione GNOME e altre compatibili. Adatto per persone con problemi di ipovisione o anche di cecità.

gok : Gnome Onscreen Keyboard. È molto più di una semplice tastiera per schermo: consente a persone con problemi di mobilità di simulare gli input della tastiera e del mouse utilizzando strumenti alternativi.

Anche il progetto KDE Accessibility[35] ha come obiettivo quello dell'accessibilità del PC e in particolare delle applicazioni scritte per l'ambiente KDE.

Le applicazioni di maggiori interesse per questo ambiente sono:

KTTS : sistema di Text-To-Speech per la sintesi vocale (in futuro si prevede il supporto di vari standard tra cui VoiceXML).

KMouseTool : strumento visuale per l'utilizzo del mouse. Adatto a chi ha problemi di movimento.

KMagnifier : ingranditore di schermo

KMouth : sistema basato su KTTS per la sintesi vocale.

Un altro progetto interessante riguardante l'accessibilità in Linux è EmacSpeak[64], basato sul popolarissimo editor di testi Emacs, che integra un sistema di sintesi e di riconoscimento vocale che consentono di trasformare l'intero desktop in una interfaccia vocale. Il supporto degli Aural CSS consente di presentare le informazioni digitali in formato vocale controllando la voce del sintetizzatore, la velocità di riproduzione, il volume ecc...

Mac OS X VoiceOver è il sistema integrato per l'interfaccia vocale di Mac OS X[2] che consente di controllare l'interfaccia e molte applicazioni attraverso comandi vocali.

Analogamente agli altri sistemi operativi, anche Mac OS X è dotato del classico ingranditore di schermo, di un sintetizzatore vocale, di un sistema per simulare la tastiera e il mouse.

1.5.2 Tastiere, stampanti e display braille

Generalmente per un non vedente non è un grosso problema utilizzare la normale tastiera del computer. È possibile tuttavia che qualcuno preferisca utilizzare la metodologia di scrittura della “dattilobrasile”, una macchina da scrivere dotata di un tasto per ciascun punto della cella braille.

La tastiera braille funziona in modo analogo. Ad ogni punto del carattere braille corrisponde un tasto, per la scrittura di un singolo carattere sarà quindi necessario premere contemporaneamente più tasti.

Alcune tastiere sono dotate di tasti aggiuntivi per poter controllare pienamente il computer e le applicazioni.

Le stampanti braille sono periferiche che consentono di effettuare stampe in rilievo dei caratteri braille. Possono essere collegate al PC e stampare qualsiasi testo e alcuni modelli sono in grado di stampare anche grafici.

I display braille sono dispositivi collegabili al PC e spesso utilizzabili assieme ad uno screen reader. Si tratta di barre che vanno dai 20 agli 80 caratteri che riproducono le linee di testo che compaiono sullo schermo. Per controllare il lettore di schermo e quindi il testo che deve essere mandato alla barra braille per la lettura, si utilizza la tastiera normale.

1.5.3 Sintesi e riconoscimento vocale

Il problema dell'accessibilità dei siti web è molto sentito soprattutto da coloro che hanno problemi di vista. Questo perché ovviamente quello visivo è il canale privilegiato per la presentazione delle informazioni in rete. Gli ipertesti creati con HTML sono infatti costituiti principalmente da testo scritto e immagini; l'interazione, inoltre, è fatta prevalentemente attraverso il mouse.

Per i non vedenti è necessario utilizzare un canale alternativo e anche modalità di interazione diverse dall'uso del mouse.

Il linguaggio braille è senza dubbio molto efficace per trasmettere contenuti testuali. Tuttavia risulta essere piuttosto ostico e richiede mol-

to tempo e molto esercizio prima di poter essere utilizzato in maniera efficiente.

Per un non vedente, il canale vocale/uditivo diventa quello privilegiato per la comunicazione e lo scambio di informazioni nella vita di tutti i giorni. Quando poi si utilizza un PC e le informazioni cui si vuole accedere viaggiano in rete, diventa fondamentale un sistema in grado di trasformare i testi in voce.

Un *sintetizzatore vocale* (TTS: Text-To-Speech) è appunto un sistema software in grado di produrre automaticamente i suoni corrispondenti ad un testo scritto.

Negli ultimi anni si è avuto un notevole sviluppo in questo campo. Esistono fondamentalmente due tipi di sintetizzatori. Il primo, più semplice, genera in modo interamente artificiale le forme d'onda necessarie a riprodurre un suono. Il risultato è una voce metallica, robotica, spesso non molto piacevole. Il secondo, più complesso, riproduce i testi concatenando delle particelle vocali preregistrati ottenendo così una voce molto simile a quella umana e anche gradevole da sentire.

Gli *screen reader* e altri software che fanno uso di questa tecnologia fanno normalmente uso del primo tipo di sintetizzatore in quanto più economico e computazionalmente più semplice da eseguire anche su macchine non molto potenti.

Accanto alla sintesi non può mancare il riconoscimento vocale (ASR: Automatic Speech Recognition) che negli ultimi anni è stato notevolmente migliorato rispetto alle prime implementazioni.

Inizialmente utilizzato soprattutto in applicazioni standalone (ad esempio per la dettatura dei testi), si sta imponendo come reale sistema di input alternativo anche per il web [38].

I sistemi presenti sul mercato sono ormai in grado di riconoscere la voce umana e il linguaggio naturale anche in ambienti rumorosi e di adattarsi in qualche modo alla pronuncia dell'utente.

Il riconoscimento vocale viene effettuato confrontando le forme d'on-

da generate dalla voce umana con dei pattern sonori presenti nelle librerie del software. Generalmente si fornisce al software una grammatica, cioè un insieme di parole o frasi che possono essere pronunciate dell'utente e che il sistema deve riconoscere.

Unire sintesi e riconoscimento vocale consente la creazione di interfacce completamente basate su questo canale. Ovviamente è possibile convogliare solo informazioni di tipo testuale, tuttavia è possibile arricchire l'interfaccia con musica ed effetti sonori di altro tipo che costituiscono l'equivalente di immagini e animazioni delle interfacce grafiche.

Le interfacce vocali possono essere usate utilmente non solo da non vedenti e ipovedenti, ma anche dai dislessici o anche da chi semplicemente desidera accedere alle informazioni in modo alternativo, ad esempio attraverso il telefono.

Come si vedrà in seguito, è inoltre possibile integrare una interfaccia vocale in una grafica creando così un sistema multimodale.

1.5.4 Browser vocali

I cosiddetti Voice Browser sono sistemi di navigazione in grado di essere pilotati attraverso comandi vocali e che usano la sintesi vocale per presentare i contenuti.

Alcuni esempi: ConversaWeb, webHearit, SpeechHTML, TelWeb. L'obiettivo di questi browser è quello di rendere accessibile a non vedenti, ipovedenti o persone con difficoltà di lettura, i contenuti del web. Non richiedono l'adozione di particolari linguaggi per la redazione dei contenuti.

1.5.5 Screen reader: Jaws, Hal, Windows-Eye...

Lo strumento principale adottato dai non vedenti per usare il computer è lo screen reader. Si tratta di un software in grado appunto di "leggere"

quello che compare a schermo, attraverso un sintetizzatore che trasforma i testi in voce.

Esistono molti screen reader il più diffuso dei quali è senza dubbio JAWS. Lo screen reader è interamente comandabile attraverso la tastiera ed è dotato di un elevato numero di comandi per la navigazione. Consente l'utilizzo della maggior parte degli applicativi tra cui, il più importante per gli scopi di questo testo, di Internet Explorer.

JAWS è estendibile, per poter supportare nuovi applicativi non previsti nell'installazione standard, attraverso un linguaggio di scripting che consente di controllare la sintesi vocale e l'interfacciamento con i programmi in esecuzione.

Oltre alla sintesi vocale è possibile pilotare, attraverso JAWS, un display per riprodurre in braille il testo che viene letto.

1.5.6 Browser testuali: Lynx, Links, w3m

Alcune persone non vedenti preferiscono utilizzare browser pensati proprio per la visualizzazione delle sole componenti testuali delle pagine web. Questi browser sono molto diffusi in ambienti Unix/Linux ma ne esistono anche versioni per altri sistemi operativi.

Essendo browser testuali non supportano la visualizzazione di elementi multimediali e non gestiscono CSS e Javascript.

Utilizzando un simile software risulta più facile l'accesso alle informazioni attraverso gli screen reader. Sono da considerarsi pertanto degli ottimi strumenti per valutare l'accessibilità dei propri siti web verificandone l'usabilità anche senza componenti grafici o script lato client.

1.6 Le direttive sull'accessibilità del W3C: WAI

Il W3C² ha istituito un gruppo di lavoro per affrontare il problema dell'accessibilità dei siti web. WAI (Web Accessibility Initiative) è il nome di questa iniziativa. Il risultato è la formulazione di diverse linee guida per lo sviluppo dei siti internet, per la realizzazione degli “user agent” (i browser e i dispositivi per la navigazione) e degli strumenti di authoring.

Per quanto riguarda l'accessibilità dei siti web sono di fondamentale importanza le cosiddette WCAG (Web Content Accessibility Guidelines) di cui esistono due versioni: la versione 1.0[71] del 1999 e la 2.0[15] che al momento in cui scrivo è da considerarsi ancora in via di sviluppo seppure già ben delineata³.

Per avere un'idea di cosa si tratta riassumo di seguito i 14 punti delle WCAG 1.0:

Linea guida 1 Fornire un testo alternativo per tutti gli elementi non testuali: immagini, filmati, applet, ecc. . .

Linea guida 2 Non riferirsi unicamente al colore come veicolo di informazione.

Linea guida 3 Utilizzare il più possibile e nel modo più appropriato i tag HTML per caratterizzare il contenuto e i CSS per presentarlo.

Linea guida 4 Specificare la lingua predefinita in cui sono redatti i contenuti ed esplicitare in ogni momento eventuali variazioni. Utilizzare i tag per acronimi e abbreviazioni per chiarire il significato dei termini.

Linea guida 5 Creare tabelle che possono essere trasformate (linearizzate) in modo semplice e chiaro per l'utente.

Linea guida 6 Nel caso in cui si utilizzino tecnologie innovative accertarsi della retrocompatibilità delle pagine. Ad esempio una pagina deve

²<http://www.w3c.org>

³Di fatto il W3C attualmente sta recependo i commenti e le indicazioni di aziende e della comunità di sviluppatori per apportare eventuali correzioni e modifiche

poter essere letta correttamente anche se i CSS o Javascript sono disabilitati.

Linea guida 7 L'utente deve essere in grado di controllare e fruire correttamente dei contenuti che variano nel tempo: evitare l'uso di testi lampeggianti o scorrevoli, il ricaricamento automatico delle pagine ecc. . .

Linea guida 8 Accessibilità dei contenuti indipendentemente dal browser: l'utente deve poter navigare correttamente indipendentemente dalle funzioni di navigazione fornite dalla propria interfaccia grafica, pertanto devono essere forniti strumenti che ne rendano indipendente la fruizione come ad esempio le access key.

Linea guida 9 Indipendenza dalla piattaforma e dallo strumento usato dall'utente.

Linea guida 10 Utilizzare soluzioni intermedie per fare in modo che vecchi browser o strumenti assistivi interpretino correttamente i contenuti. Queste soluzioni andrebbero adottate finché tali strumenti non evolvono. Ad esempio poiché alcuni screen reader leggono link consecutivi come un link unico, sarebbe opportuno separarli ad esempio utilizzando una lista (eventualmente nascondendo i punti della lista e visualizzandoli sulla stessa riga anziché su righe consecutive)

Linea guida 11 Utilizzare le tecnologie e linee guida definite dal W3C, non usare formati proprietari o elementi considerati obsoleti o deprecati.

Linea guida 12 Fornire all'utente informazioni riguardo la struttura della pagina o di elementi complessi. Ad esempio dare un titolo significativo ai frame, raggruppare le opzioni di una form che sono collegate tra loro.

Linea guida 13 Fornire all'utente un sistema di navigazione e di "orientamento" chiaro all'interno del sito. Ad esempio indicare chiaramente

il target di un link (dove porta), utilizzare i metadata, utilizzare menu, barre di navigazione coerenti in tutto il sito.

Linea guida 14 Utilizzare un linguaggio chiaro e di semplice comprensione, compatibilmente con il target di utenza cui è destinato il sito.

Per molte di queste linee guida, il W3C indica anche alcune tecniche per la loro applicazione [72]. Altre sono lasciate allo sviluppatore come ad esempio l'uso del linguaggio, anche perchè questo dipende fortemente dalla lingua usata.

Nel prossimo capitolo queste linee guida saranno in parte riprese al fine di individuare una buona pratica per lo sviluppo dei siti ad accesso multimodale.

1.7 Direttive del governo USA: Section 508

Gli USA sono stati i primi ad aver introdotto per legge, principi per l'abbattimento delle barriere all'accesso alle informazioni da parte delle persone disabili, cui tutti i siti di interesse pubblico devono conformarsi.

Si tratta essenzialmente di una linea guida organizzata in sedici punti illustrati in breve di seguito⁴:

1. Deve essere fornito un equivalente testuale per qualsiasi elemento non testuale
2. Alternative equivalenti per una presentazione multimediale devono essere sincronizzate con la presentazione.
3. Le pagine web devono essere progettate in modo tale che tutte le informazioni veicolate per mezzo del colore siano rese disponibili anche senza l'uso del colore, per esempio tramite il contesto o il codice di marcatura.

⁴Traduzione tratta da Diodati[3]

4. I documenti devono essere organizzati in modo tale che siano leggibili anche senza il ricorso ad un foglio di stile associato.
5. Collegamenti testuali ridondanti devono essere forniti per ciascuna regione attiva di una mappa immagine lato server.
6. Devono essere fornite mappe immagine lato cliente invece che mappe immagini lato server, con eccezione dei casi in cui le regioni della mappa non possano essere definite tramite una delle forme geometriche disponibili.
7. Per le tabelle di dati devono essere identificate le intestazioni di riga e di colonna.
8. Un appropriato codice di marcatura deve essere usato per associare celle di dati e celle di intestazione in tabelle di dati che abbiano due o più livelli logici di intestazioni di riga o di colonna.
9. I frame devono essere denominati con un testo che faciliti l'identificazione di ciascun frame e la navigazione tra di essi.
10. Le pagine devono essere progettate in modo da evitare sfarfallamenti dello schermo che abbiano una frequenza maggiore di 2 Hz e minore di 55 Hz.
11. Una pagina di solo testo, con informazioni o funzionalità equivalenti, deve essere fornita affinché un sito web sia conforme alle disposizioni di questa sezione, quando la conformità non può essere raggiunta in nessun altro modo. Il contenuto della pagina di solo testo deve essere aggiornato ogni volta che la pagina primaria viene modificata.
12. Quando delle pagine utilizzano linguaggi di script per visualizzare contenuti, o per creare elementi dell'interfaccia, le informazioni fornite per mezzo dello script devono essere identificate tramite del testo funzionale che possa essere letto usando tecnologie assistive.
13. Quando una pagina web richiede che un applet, un programma accessorio integrato, o un'altra applicazione sia presente sul sistema

cliente per interpretare il contenuto della pagina è necessario che tali estensioni siano conformi a tutte le precedenti direttive di accessibilità.

14. Quando dei moduli elettronici sono progettati per essere completati in linea, il modulo deve consentire alle persone che adoperano tecnologie assistive di accedere alle informazioni, ai campi e alle funzionalità richieste per il completamento e l'inoltro del modulo, incluse tutte le guide e i suggerimenti.
15. Deve essere fornito un metodo che permetta agli utenti di saltare collegamenti di navigazione ripetitivi.
16. Quando è richiesta una risposta temporizzata, l'utente ne deve essere avvisato e gli deve essere dato tempo sufficiente per segnalare che ha bisogno di un tempo più lungo.

Le WCAG e il Section 508 hanno molti punti in comune ma ad una attenta analisi si notano alcune differenze. Ci sono aspetti tenuti in considerazione da una e non dall'altra e viceversa e addirittura indicazioni incompatibili tra loro. Risulta pertanto difficile se non a volte impossibile realizzare un sito che sia conforme contemporaneamente ad entrambe le direttive.

1.8 Validatori di accessibilità

La maggior parte delle linee guida sull'accessibilità del W3C è costituita da requisiti tecnici che possono essere verificati in modo automatico da un sistema software.

Alcuni strumenti di authoring, quali ad esempio Dreamweaver consentono una verifica diretta di tali requisiti proprio durante lo sviluppo.

Esistono inoltre diversi servizi utilizzabili direttamente online che ri-

sultano molto comodi per effettuare una prima verifica di accessibilità sui siti internet. Alcuni di questi sono ad esempio: Bobby⁵, Cinthia⁶.

I validatori sono in grado di indicare i livelli di priorità raggiunti secondo le specifiche W3C (A, AA, AAA) e Section 508. In realtà il raggiungimento effettivo di ciascun livello non può essere valutato automaticamente ma richiede una analisi più approfondita delle pagine in quanto l'accessibilità coinvolge vari aspetti legati ad esempio all'uso del linguaggio, all'usabilità, all'adattabilità dei contenuti, la cui verifica può essere ottenuta solo grazie all'intervento umano e richiede competenze professionali diverse: esperti di fattori umani quali psicologi e pedagogisti, esperti di ergonomia e usabilità, ecc. . .

1.9 Uso e comprensione del linguaggio

L'utilizzo di un linguaggio semplice, di una corretta punteggiatura e di una struttura delle frasi e dei testi accurata, sono elementi molto importanti da prendere in considerazione nella redazione dei contenuti per il web soprattutto quando i destinatari sono disabili oppure in generale quando vengono utilizzate tecnologie come la sintesi vocale.

Esistono formule matematiche e criteri oggettivi che consentono di valutare la comprensibilità dei testi. Uno dei criteri più importanti per quanto riguarda la lingua italiana è l'uso del vocabolario di base della lingua italiana di Tullio de Mauro [62], un elenco di lemmi elaborato prevalentemente secondo criteri statistici. Esso rappresenta la porzione della lingua italiana usata e compresa dalla maggior parte di coloro che parlano italiano. Viene adottato ad esempio dai redattori del sito Due Parole⁷.

⁵<http://bobby.watchfire.com>

⁶<http://www.contentquality.com>

⁷<http://www.dueparole.it>

Altri criteri di valutazione della leggibilità sono dati dalle formula di Flesch e GulpEASE⁸.

Formula di Flesch La formula considera solo due variabili linguistiche: lunghezza media delle parole espressa in sillabe per parola, e lunghezza media delle frasi espressa in parole per frase.

Pensata inizialmente per la lingua inglese è stata poi adattata alla lingua italiana da Franchina - Vacca⁹ e la sua espressione è indicata nella seguente formula.

$$I = 206 - 0.65S - W \quad (1.1)$$

Dove S indica il numero di sillabe di 100 parole e W è il numero medio di parole per frase. Più è alto l'indice I corrispondente, maggiore sarà la leggibilità del testo cui si riferisce.

Indice GulpEASE L'indice GulpEASE è stato sviluppato dal GULP - Gruppo universitario linguistico pedagogico, presso l'Istituto di Filosofia dell'Università degli Studi di Roma "La Sapienza" ed è stato costruito direttamente per la lingua italiana.

La formula che definisce l'indice di leggibilità è la seguente

$$F = 80 - LP/10 + FR * 3 \quad (1.2)$$

dove LP indica il rapporto tra il numero di lettere moltiplicato per 100 e il numero totale di parole, FR è il rapporto fra numero frasi moltiplicato per 100 e il numero di parole.

L'indice corrispondente viene confrontato rispetto ad una tabella che indica qual'è la leggibilità del testo in relazione al livello di istruzione del lettore: scuola elementare, media, superiore. I valori di leggibilità indicati

⁸Si può ottenere una descrizione di questi due indici al sito <http://www.eulogos.net>

⁹<http://www.eulogos.net/it/glossario/default.htm>

sono: molto facile, facile, difficile, molto difficile, quasi incomprensibile. La figura 1.1 indica graficamente la scala di valori dell'indice.

Il servizio CENSOR di Éulogos¹⁰ consente di analizzare i propri testi valutando l'indice GulpEASE e confrontando le parole usate con quelle presenti nel vocabolario di base di De Mauro.

Analisi semantica latente L'analisi semantica latente (LSA Latent Semantic Analysis)[16] è un modello cognitivo per la rappresentazione e l'acquisizione del senso dei vocaboli e la comprensione di testi. Sono stati elaborati vari modelli cognitivi a base di LSA: rappresentazione delle conoscenze, acquisizione del vocabolario, comprensione di testi, valutazione automatica di testi.

Modelli di comprensione del linguaggio possono essere utili per la realizzazione di sistemi automatici per l'individuazione della semantica dei contenuti del web. Tali sistemi, integrati nei normali tool di ausilio quali gli screen reader, potrebbero consentire una maggiore usabilità e comprensione da parte dell'utente dei contenuti di una pagina.

1.10 Accessibilità e multimodalità

Un altro gruppo di lavoro del W3C, il Multimodal Interaction Activity[68], si occupa del problema della multimodalità, cioè della possibilità di realizzare sistemi di interfacciamento con l'utente che coinvolgano diverse modalità di interazione, diversi strumenti, utilizzati anche contemporaneamente durante l'accesso alle informazioni in rete [55].

Gli obiettivi dell'iniziativa sono riassumibili nei seguenti punti:

- Estendere il web consentendo diverse modalità di interfacciamento: grafica, voce, gesti, tatto. . .

¹⁰<http://www.eulogos.net>

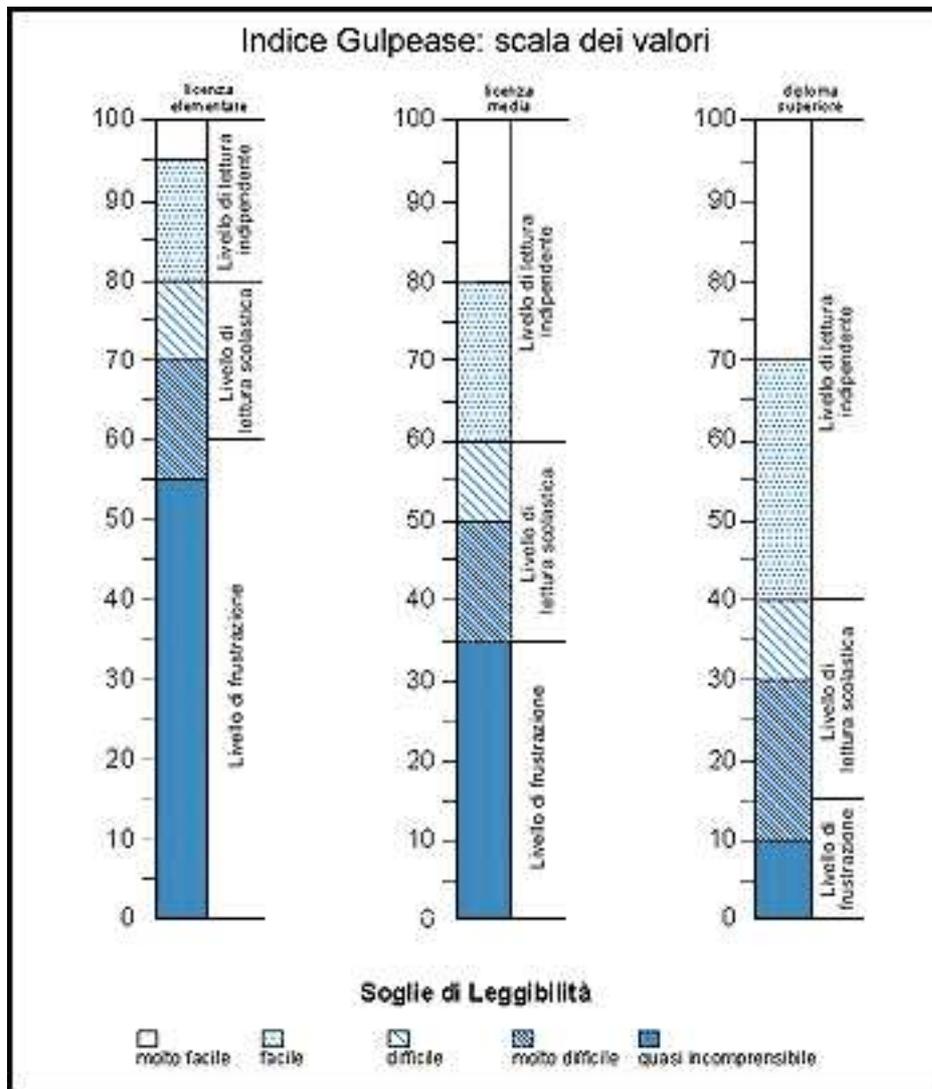


Figura 1.1: Indice GulPEASE: scala dei valori

- Aumentare l'interazione uomo-macchina e tra le persone coinvolgendo diversi dispositivi
- Adattare le interfacce ai dispositivi, al contesto e alle preferenze dell'utente
- Incrementare l'accessibilità delle informazioni.

Il Multimodal Interaction Framework [29] illustra a livello astratto la composizione di una architettura basata sull'interazione multimodale con l'utente e quali sono i linguaggi di markup ritenuti più adatti per ciascun componente di una potenziale applicazione.

Di seguito vengono illustrati alcuni linguaggi nati da questa iniziativa o che possono essere in qualche modo ad essa collegati.

InkML InkML[74] è un linguaggio di markup per la codifica dei segni tracciabili attraverso un dispositivo di puntamento: mouse, penna ottica, touch screen, touch pad, ecc. . .

Può quindi essere utilizzato come linguaggio di interscambio per questa tipologia di input e può essere usato dai sistemi di riconoscimento dei gesti, della scrittura a mano libera ecc. . .

EMMA EMMA, Extensible MultiModal Annotation markup language [73], è un linguaggio pensato appositamente per lo scambio di dati all'interno di un sistema multimodale. L'uso è destinato a sistemi in grado di ricevere ed interpretare input di varia natura quali la voce, dispositivi di puntamento, dispositivi tattili, gesture, DTMF, tastiere ecc. . .

Il linguaggio consente di formalizzare l'interpretazione dei possibili input dando loro un significato ed eventualmente associandoli ad un comando.

DPF Dynamic Property Framework (DPF)[36] è una specifica formale che definisce un'interfaccia per l'accesso a proprietà relative alle capacità

dei dispositivi, alla loro configurazione, alle preferenze dell'utente e alle condizioni ambientali.

Il DPF fornisce un "Interaction Manager" per l'accesso ad una gerarchia di proprietà che rappresentano i parametri di utilizzo corrente di una applicazione. Tali proprietà indicano quali sono le modalità di interazione supportate e quali quelle attive.

La specifica risponde a uno dei requisiti principali delle applicazioni multimodali che è quello di doversi adattare ad un ampio spettro di dispositivi, di contesti di utilizzo, di preferenze utente e alla dinamicità di tutte queste proprietà.

Il gruppo di lavoro del W3C ha definito formalmente l'interfaccia attraverso il linguaggio OMG IDL.

XForms XForms [41] è il linguaggio che auspicabilmente costituirà la nuova generazione dei moduli interattivi per consentire all'utente di inserire dati e inviarli al server di un sito web.

Si tratta di un linguaggio XML che consente di separare le form in tre componenti: il modello della form, l'istanza, l'interfaccia utente.

La netta separazione tra contenuto e presentazione è un punto di forza che verrà ribadito più volte in seguito, in relazione all'accessibilità. I vantaggi di XForms sono la possibilità del riuso dei modelli per diverse istanze e per diversi tipi di interfacce, la forte tipizzazione dei dati e la facilità di integrazione in altri linguaggi basati su XML.

Linguaggi per interfacce vocali Negli ultimi anni sono stati sviluppati nuovi linguaggi per la definizione delle interfacce vocali e per l'integrazione con quelle grafiche per la realizzazione di sistemi multimodali di accesso.

Di seguito vengono illustrati brevemente i principali.

VoiceXML e X+V VoiceXML [53] è un linguaggio per la definizione di dialoghi con le seguenti funzionalità:

- sintesi vocale TTS (Text-To-Speech)
- riproduzione di audio digitale
- riconoscimento vocale ASR (Automatic Speech Recognition)
- riconoscimento codici DTMF¹¹
- registrazione audio
- interfaccia telefonica

Il linguaggio è pensato essenzialmente per il web e il fatto di essere una istanza di XML lo rende facilmente integrabile con altri linguaggi di markup. È stata ad esempio realizzata una specifica per l'integrazione con l'HTML, XHTML+Voice [32] [57], estremamente valida per la realizzazione di interfacce multimodali.

Il linguaggio X+V richiede un browser in grado di interpretarlo e dotato di un sistema di riconoscimento e di sintesi vocale. Al momento esiste una versione particolare del browser Opera realizzato da IBM e Opera Software avente tali caratteristiche ¹².

SSML Speech Synthesis Markup Language (SSML) [20] nasce nell'ambito della Voice Browser Activity del W3C con l'obiettivo di fornire un linguaggio standard per controllare vari aspetti della sintesi vocale quali la pronuncia, il volume, il tono, la velocità ecc. . .

L'obiettivo è quello di migliorare la sintesi vocale dei testi rendendola più precisa, comprensibile e naturale e quindi anche più gradevole. Spesso ad esempio nei testi compaiono vocaboli di diverse lingue che il sintetizzatore non è in grado di pronunciare correttamente in quanto si basa su

¹¹Dual Tone Multi-Frequency, il sistema usato dai telefoni moderni. DTMF assegna una specifica frequenza (consistente in una coppia di toni separati) a ciascun pulsante in modo che possano facilmente essere identificati da un sistema automatico

¹²Attualmente supporta solo la lingua inglese

un vocabolario fisso legato ad una sola lingua. L'utilizzo dei fonemi sviluppati dall'IPA (International Phonetic Association) è una proprietà del linguaggio che risolve proprio questo tipo di problemi.

La versione 2.0 di VoiceXML include proprio le specifiche di SSML per il raffinamento della sintesi vocale.

SALT (Speech Application Language Tags) SALT è un linguaggio i cui obiettivi e le modalità di funzionamento sono molto simili a XHTML+Voice. Si tratta infatti di un linguaggio per l'integrazione di una interfaccia vocale con una grafica realizzata in HTML. Tra gli sviluppatori del linguaggio figurano società private quali Cisco Systems Inc, Comverse Inc., Intel Corporation, Microsoft Corporation, Philips Electronics N.V., SpeechWorks International.

Come X+V richiede un browser compatibile. Attualmente funziona con Internet Explorer a patto di installare un opportuno plug-in.

In realtà SALT consente di realizzare anche una interfaccia solo vocale o telefonica, essendo in grado di gestire il flusso di dialogo tra l'applicazione e l'utente con gli stessi principi di VoiceXML.

1.11 Lavori correlati

Normalmente siamo abituati ad utilizzare interfacce grafiche composte da finestre e icone e nelle quali i principali mezzi di interazione sono la tastiera ed il mouse. Il paradigma di interazione è quello del "punta e clicca".

Con lo sviluppo di nuovi dispositivi e nuove modalità di interazione, primo tra tutti quello vocale, non soltanto legate al mondo della disabilità, sorge la necessità di dover progettare in modo diverso le interfacce, compiendo analisi ad un livello più astratto.

Dan R. Olsen e altri[19] hanno definito uno standard (XWeb) e un protocollo (XTP) per affrontare il problema dell'interazione dell'utente

proveniente da diversi dispositivi e attraverso diverse metodologie di interfacciamento. L'obiettivo è quello di rendere indipendenti i servizi offerti dalle possibili piattaforme sulle quali l'utente si trova ad operare.

Michael Wynblatt, Dan Benson e Arding Hsu [43] hanno affrontato il problema generale di accedere alle informazioni presenti in rete utilizzando ambienti non visuali e che quindi sfruttassero i canali audio e tattile per presentare i contenuti.

Per molti il punto di partenza di ogni studio è sempre la tradizionale interfaccia grafica proposta agli utenti attraverso i siti web e il browser dalla quale si cerca di ottenere una rappresentazione non visuale[47] [51].

Altri come A.W. Huang e N. Sundaresan [10] hanno elaborato modelli concettuali per l'adattamento dei contenuti dei siti per supportare una usabilità universale dei servizi web-based.

Pèrez-Quiñones e altri hanno sviluppato un linguaggio, UIML, basato su XML per la modellizzazione e la specifica di interfacce utente multipiattaforma [46]. Il vantaggio dovrebbe essere quello di descrivere l'interfaccia in modo indipendente dalla presentazione e dai dispositivi di input e di output utilizzati dall'utente. L'interfaccia vera e propria viene poi creata automaticamente mediante una trasformazione (attraverso XSLT) della specifica UIML. In tal modo si può creare una interfaccia grafica per il web fatta in HTML per i browser, in WML per i cellulari, in AWT o Swing per applicazioni standalone o ancora in VoiceXML per l'interazione puramente vocale.

Gli stessi autori hanno poi focalizzato l'attenzione sull'usabilità, ritenuto un requisito fondamentale per la realizzazione di sistemi di interazione efficienti e gradevoli per l'utente [45].

Dunque per costruire una interfaccia che sia adattabile a diverse piattaforme sembra che il modo migliore sia progettarela utilizzando un linguaggio astratto, di alto livello, che consenta poi di realizzare ogni singola istanza attraverso una semplice trasformazione.

Progettare una interfaccia multipiattaforma significa anche pensare a di-

verse modalità di interazione con l'utente. A mio avviso la multimodalità è proprio una delle chiavi di successo per l'accessibilità e l'usabilità estesa a tutti.

Per creare una interfaccia multimodale bisogna seguire alcune importanti linee guida. Reeves, Lai, Larson e altri propongono alcuni principi [37]:

- Progettare l'interfaccia per il più ampio spettro di utenti possibile. Bisogna tener conto delle capacità e delle eventuali disabilità degli utenti. La possibilità di impartire gli stessi comandi e di ricevere gli output in maniere differenti aumenta la possibilità di accesso da parte degli utenti.
- Integrare le diverse modalità di accesso e adattarle alle preferenze dell'utente e al contesto di utilizzo
- Creare interfacce consistenti. Le diverse interfacce devono per quanto possibile essere equivalenti tra loro per quanto riguarda le possibilità di interazione e soprattutto per le informazioni erogate.
- Dare all'utente opportuni feedback relativi alle operazioni compiute. Molto importante ad esempio nelle interfacce basate sul riconoscimento e la sintesi vocale, nelle quali la ricezione da parte del sistema del comando impartito non è sempre scontato.
- Gestire in modo accurato gli errori nel momento in cui si verificano e consentire all'utente, se possibile, di cambiare la modalità di interazione.

Per quanto riguarda più specificamente l'interazione multimodale, T.V. Raman[63] indica alcuni aspetti da tenere in considerazione, che sono riassunti di seguito:

- Sincronizzare le diverse modalità di interazione. L'interazione vocale è legata al tempo mentre quella effettuata con il mouse allo spazio.

- L'interazione deve rimanere efficace anche quando una delle modalità viene a mancare per qualche ragione. Realizzare quindi modalità di accesso alternative e ridondanti in modo da migliorare l'adattamento del sistema ai cambiamenti di contesto o delle capacità dell'utente.
- Diverse modalità di interazione devono condividere lo stesso stato, cosicché l'utente possa passare da una modalità all'altra mantenendo il sistema consistente.
- Le modalità di interazione devono rimanere intuitive, l'effetto dei comandi deve essere facilmente predicibile e si deve fare in modo che l'utente sia sempre in grado di capire cosa fare per raggiungere un determinato obiettivo.

1.11.1 La comunicazione multimodale

Sono molto importanti alcuni spunti di ricerca nati all'interno dell'ArcsLab relativamente alla comunicazione multimodale per le persone con disabilità[14] con cui si enfatizza tra l'altro la necessità della descrizione degli utenti secondo il modello proposto da ICF.

Altri lavori indicano nella multimodalità un mezzo per supportare le persone con disabilità sensoriali, fisiche e cognitive[60][59].

Un ulteriore aspetto già affrontato è quello della valutazione della comprensibilità dei contenuti trasmessi via web[24].

1.11.2 Semantica e adattabilità

Esistono diversi criteri per ricavare la struttura della pagina analizzandone il contenuto [49] [25] [42], ad esempio si può pensare che il contenuto principale della pagina sia quello con la maggior densità di testo e che occupa la maggior parte dello spazio nella finestra una volta visualizzata nel browser. Un blocco costituito da soli link, eventualmente organizzati in una lista o in una tabella, presumibilmente costituisce un menu.

S. Mukherjee, G. Yang e altri hanno realizzato algoritmi di analisi dell'albero DOM relativo ad una pagina HTML per individuare un template che consentisse di identificare i componenti principali e quindi analizzare semanticamente la pagina [52].

Una tecnica interessante che è stata proposta da R. Schaefer[50] e altri per adattare il contenuto di una pagina HTML a dispositivi quali PDA e cellulari è quella che consiste nel definire delle regole fuzzy che consentano di filtrare il contenuto in modo da renderlo più facilmente accessibile attraverso dispositivi limitati.

Si potrebbe pensare di vedere il telefono e in generale un sintetizzatore vocale come un dispositivo "limitato" per la presentazione dei contenuti e di provare ad individuare regole fuzzy per il filtraggio della pagina in modo da semplificarla e creare più facilmente la corrispondente interfaccia vocale.

Partendo da queste considerazioni potrebbe essere possibile realizzare un sistema che supporti lo sviluppatore nella creazione di un sito accessibile facilmente anche in modo vocale senza necessariamente progettare due distinte interfacce.

Un simile sistema potrebbe essere integrato in normale programma di authoring di pagine HTML.

Capitolo 2

Linee guida per l'estensione delle direttive sull'accessibilità

I criteri di accessibilità proposti dal W3C nelle WCAG 1.0 e 2.0 sono un buon punto di partenza e vanno sempre presi in considerazione quando si vuole partire a realizzare un sito accessibile.

Tuttavia a volte risultano essere piuttosto generici e spesso non danno indicazione di come fare per metterli in pratica o ne danno di opinabili e aprono aspre discussioni nell'ambiente degli sviluppatori [3].

Nella versione 2.0 del WCAG si è cercato di costruire delle linee guida applicabili ad un più ampio spettro di tecnologie. Si è tenuta in maggiore considerazione la possibilità della diffusione multicanale delle informazioni attraverso il web, ma senza porre sufficiente accento sul fatto che multicanalità implica molto spesso multimodalità di interfacciamento e quindi la necessità di avere una guida pratica che tenesse conto anche di questo aspetto.

L'iniziativa del W3C per riguardante l'interazione multimodale (Multimodal Interaction Activity) è finora mantenuta separata da quella più strettamente legata all'accessibilità.

In questo capitolo verranno fornite alcune indicazioni pratiche per ottenere l'accessibilità multimodale e si mostrerà come è possibile, grazie a

queste indicazioni, costruire un sito ad elevata accessibilità multimodale e porre le basi per realizzare ad esempio una interfaccia vocale a partire da contenuti pensati per i browser visuali, problema che verrà affrontato nel capitolo 4. Non verranno affrontati tutti gli aspetti messi in luce dalle linee guida sull'accessibilità del W3C ma saranno evidenziate soprattutto quelle tecniche maggiormente legate alla multimodalità che è l'argomento centrale di questa tesi e sul quale ritengo che le WCAG da sole non siano sufficienti.

Nella ricerca di linee guida pratiche per l'accessibilità multimodale, il primo passo sarà quello di verificare l'accessibilità delle pagine utilizzando un browser attraverso l'ausilio di uno screen reader. Verrà preso come riferimento Jaws 1.5.5 che, tra quelli provati sembra raggiungere il maggior rapporto qualità/semplificata d'uso ed è anche quello maggiormente utilizzato da chi ha bisogno di questo tipo di tecnologia assistiva (non vedenti).

Le indicazioni che vengono date dall'utilizzo di un sistema assistivo di questo tipo saranno fondamentali per la realizzazione automatica di una interfaccia vocale e in generale di un sistema multimodale.

2.1 Usabilità e accessibilità

Spesso si ritiene che usabilità e accessibilità dei siti web siano concetti nettamente separati tra di loro. In realtà essi hanno molti punti di contatto. Alcuni degli obiettivi che si prefiggono i criteri di accessibilità sono propri anche dell'usabilità.

Personalmente ritengo inoltre che l'accessibilità sia un fatto fortemente soggettivo e che non sia altro che una estensione del concetto di usabilità: un sito è accessibile ad una data categoria di persone se e solo se è usabile da essa.

In questo contesto, l'usabilità è la proprietà di un sito di presenta-

re i contenuti in modo efficace, efficiente e soddisfacente per l'utente in relazione al dispositivo e ad eventuali ausili utilizzati per la navigazione.

Ritengo che quindi che non si possa parlare di accessibilità senza parlare di usabilità e in entrambe i casi senza avere ben presente a chi ci si rivolge: quali categorie di persone, quali capacità, quali necessità ed eventualmente quali disabilità. Per questo motivo, nella realizzazione di un sito, non credo che si possa raggiungere un livello di accessibilità universale, valido per tutte le tipologie di utente, creando una sola versione delle pagine. Bisogna costruire siti dinamici in grado di adattare i contenuti e la loro presentazione in base alle caratteristiche dell'utente e alle modalità di interazione che si ritengono più consone al suo profilo.

Le linee guida sull'accessibilità nascono soprattutto dalla necessità di consentire agli utenti con qualche tipo di disabilità di fruire delle informazioni in rete. Con l'accessibilità multimodale intendo estendere il campo di attenzione coinvolgendo tutte le tipologie di utenti che sono caratterizzati anche dagli strumenti che utilizzano per interfacciarsi al web oltre che dalle loro disabilità o capacità, le quali restano comunque un punto di riferimento importantissimo.

La profilazione dell'utente è quindi un aspetto cruciale da tenere in considerazione. Inoltre, gli utenti devono essere fortemente coinvolti per la valutazione dell'usabilità/accessibilità in quanto possono dare indicazioni molto più precise e pratiche di una qualunque guida teorica.

2.2 Guida pratica all'accessibilità: estensione delle WCAG

Al momento in cui scrivo le linee guida WCAG 2.0 sono ancora in fase di sviluppo. L'ultima redazione risale al 10 Novembre 2004. Gli obiettivi principali rispetto alla versione 1.0 sono i seguenti:

- organizzare in modo più efficiente le linee guida

- correggere i livelli di priorità di alcuni punti di controllo (checkpoint), soprattutto alcuni che sono stati messi in secondo piano e che invece richiedono maggiore attenzione da parte degli sviluppatori
- aggiungere, rimuovere e modificare alcuni requisiti in relazione alla nascita di nuove tecnologie per il web sviluppatesi dopo la redazione finale della versione 1.0

I miglioramenti che il gruppo di lavoro si è prefisso possono essere riassumibili essenzialmente nei seguenti punti:

- assicurarsi che i requisiti di accessibilità siano applicabili al più ampio spettro di tecnologie esistenti
- chiarire il più possibile i requisiti di conformità.
- rivolgersi ad un'audience più diversificata possibile
- chiarire il più possibile quali categorie di utenti ricevono i maggiori benefici dall'adozione di ciascun criterio di accessibilità illustrato nella guida
- assicurare una compatibilità all'indietro rispetto alla versione 1.0 della guida.

Nelle successive sezioni si affronterà in concreto il problema dell'accessibilità multimodale. Uno degli strumenti usati per la verifica dei criteri e delle tecniche suggerite è JAWS e si farà spesso riferimento ad esso. Un altro metodo pratico di valutazione è quello della trascodifica delle pagine in una interfaccia puramente vocale che viene descritto capitolo 4 e che ha consentito spesso di migliorare i criteri attraverso raffinamenti successivi.

2.2.1 Separazione della struttura dai contenuti

Uno dei requisiti fondamentali per l'accessibilità multimodale è tenere il più possibile separati i contenuti dalla struttura della pagina, dalla logica di presentazione che in HTML è prevalentemente visuale.

Una delle possibili modalità di interazione dell'utente può essere quella vocale, in cui i contenuti sono serializzati e letti da un sintetizzatore invece di essere mostrati da un browser visuale. È quindi fondamentale fare in modo che sia possibile adattare i contenuti anche a questo tipo di presentazione e quindi eliminare completamente dalla pagina web tutte le informazioni relative alla visualizzazione grafica.

La chiave: XML Il linguaggio XML è senza dubbio quello più adatto a questo compito. Tuttavia non va tralasciata la semplicità nel costruire interfacce grafiche ipermediali attraverso l'HTML. La soluzione ottima è quella di utilizzare XHTML per la costruzione delle pagine web. In particolare è auspicabile l'utilizzo del dtd "strict", che meglio interpreta questa necessità di separazione tra struttura delle pagine e presentazione.

Usando XHTML si eliminano tutti i tag e gli attributi di presentazione quali `font`, `center`, `b`, `i`, `align`, `color` ecc...

È necessario poi utilizzare in modo opportuno i tag (X)HTML che caratterizzano in modo logico il contenuto. Ad esempio usare i tag `h1`, `h2`, `h3` ecc... per indicare i titoli, separare i testi in paragrafi usando il tag `p`, usare in modo opportuno le liste ordinate (`ol`), non ordinate (`ul`), di definizione (`dl`), le abbreviazioni (`abbrev`), gli acronimi (`acronym`) e così via. Tutto questo contribuisce ad una maggiore comprensione dei contenuti da parte di chi utilizza vari ausili alla navigazione e consente una più agevole trasformazione della pagina nel caso in cui si voglia creare un'interfaccia vocale o multimodale.

In sostituzione ai tag e attributi di presentazione si utilizzano i CSS che consentono peraltro un effetto grafico spesso molto più gradevole e di realizzare effetti che con il solo HTML non sarebbe possibile ottenere.

Oltretutto i CSS possono essere realizzati in varie versioni in base al tipo di strumento usato dall'utente (il cosiddetto user agent). Ad esempio molto interessanti sono gli aural CSS pensati per quei browser in grado di sintetizzare i contenuti o i CSS per le stampanti braille.

Nella realizzazione di siti accessibili in modo multimodale sarebbe opportuno utilizzare questi diversi tipi di CSS anche se purtroppo a tuttoggi sono pochi i browser in grado di supportarli pienamente e l'effettivo utilizzo deve comunque essere fatto seguire da una verifica degli effetti ottenuti.

Per concludere: utilizzare XML o XHTML per i contenuti, CSS per la presentazione grafica nel caso in cui gli strumenti usati dall'utente lo supportino.

2.2.2 Layout di pagina e accesso con screen reader

Osservando una pagina web, prima ancora di cominciare a leggerla e comprenderla, appare subito chiara (il più delle volte) la struttura logica del contenuto. È cioè facile capire dove è posizionato il menu (solitamente una lista, orizzontale o verticale di link associato ad un testo o a delle icone), il contenuto principale (è quel blocco di testo, immagini ecc... che occupa la maggior parte della pagina) e i vari contenuti accessori quali banner pubblicitari o contenuti secondari che rimandano ad altri siti o ad approfondimenti presenti in altre pagine all'interno dello stesso sito.

Molto spesso si ricorre all'utilizzo di menu gerarchici e quindi è presente un menu principale per l'accesso diretto alle varie sezioni del sito e una serie di menu contestualizzati per la navigazione all'interno di una singola sezione. Anche questa divisione è spesso facilmente individuabile già dal primo impatto con la pagina web.

Tutto questo non è ovviamente possibile per un non vedente che utilizza come supporto per la navigazione un browser vocale o uno screen reader, i quali necessariamente devono "serializzare" il contenuto della pagina e lasciano interamente all'utente il compito di interpretarlo. Sorge quindi il problema di aiutare l'utente a comprendere la struttura, il layout della pagina, per consentirgli di fruire pienamente il sito web accedendo il più rapidamente possibile alle informazioni principali.

Poiché il linguaggio HTML utilizzato per produrre le pagine web non consente una netta separazione tra la logica del contenuto e la sua presentazione, non è possibile per lo screen reader riuscire a riconoscere in una pagina il suo contenuto principale, che presumibilmente l'utente vorrà ascoltare per primo, il menu (che l'utente vuole usare solo per sapere cos'altro propone il sito e per uscire dalla pagina corrente) e tutto il resto (che probabilmente l'utente non vorrebbe nemmeno sentire!). Per fare ciò lo screen reader dovrebbe essere dotato di una certa "intelligenza" o di capacità di analisi semantica del contenuto e dovrebbe poi consentire all'utente che lo utilizza di scegliere quale componente logica della pagina leggere.

Avendo a disposizione un normale screen reader come JAWS possiamo fare affidamento solo a quello che può essere scritto all'interno della pagina e utilizzando dei trucchi, che verranno illustrati in seguito, possiamo comunicare all'utente delle informazioni che non vengono percepite dall'utente che utilizza un browser grafico.

Non usare le tabelle La prima indicazione pratica per la realizzazione dei layout di pagina va in controtendenza rispetto alla pratica comune della stragrande maggioranza dei siti web che è quella di utilizzare le tabelle per l'impaginazione.

L'utilizzo delle tabelle in HTML è peraltro molto comodo a questo scopo, e se ne è abusato a tal punto che ormai è estremamente difficile non trovare siti internet che non le utilizzano per il layout piuttosto che per il reale scopo per le quali sono state definite cioè per contenere dati appunto in forma tabellare.

A rendere le tabelle un potente strumento di impaginazione è bastato avere la possibilità di eliminare i bordi, di dimensionare a piacimento la dimensione delle celle, di unire celle di colonne o righe adiacenti e addirittura di inserire intere tabelle dentro una cella.

Proprio quest'ultima possibilità di creare tabelle innestate fra di loro fa

sì che la lettura di una pagina da parte di uno screen reader risulti estremamente complicata: esso solitamente indica, quando si trova di fronte un tabella, il numero di righe e di colonne prima di cominciare la lettura sequenziale dei dati. Quando però una cella contiene una nuova tabella chi è all'ascolto se non è particolarmente attento perde molto facilmente l'orientamento e non è più in grado di comprendere il testo che gli viene proposto.

Un ulteriore svantaggio derivante da questo uso delle tabelle è la pesantezza in termini di spazio occupato dal codice HTML. Inoltre, maggiore è l'uso delle tabelle, soprattutto se innestate, maggiore è il tempo di elaborazione da parte dei browser.

Il principale problema che si incontra quando si deve leggere una tabella attraverso uno screen reader e che è necessario far capire all'utente che ascolta qual'è la struttura della tabella: quante sono le righe e le colonne, che cosa contiene la tabella, quali sono le intestazione delle colonne e infine devono essere letti in una qualche sequenza i dati in essa contenuti. Lo screen reader deve quindi "serializzare" il contenuto di una struttura dati che è in realtà a due dimensioni (righe e colonne) senza far perdere l'orientamento all'utente che può solo ascoltare.

Ci si rende conto che "raccontare" il contenuto di una tabella non è facile anche nel linguaggio naturale. Solitamente quando si vogliono comunicare i dati di una tabella si indicano solo i dati in celle significativi enunciando ogni volta l'intestazione della colonna.

L'indicazione è quindi quella di non usare mai le tabelle per impaginare ma solo per rappresentare i dati. In realtà non è proibito usare le tabelle per il layout a patto che la linearizzazione non cambi il significato dei contenuti, cosa che si può verificare in caso di più tabelle innestate o in caso di unione di colonne e righe attraverso `colspan` e `rowspan`. L'ideale sarebbe avere un metodo per distinguere una tabella di dati da una usata solo per il layout, basterebbe un attributo nel tag `table` che consenta in questo modo una elaborazione automatica. In seguito si affronterà bre-

vemente come creare tabelle accessibili, assumendo che queste vengano usate solo per i dati.

Il problema della serializzazione delle tabelle verrà affrontato nuovamente nel prossimo capitolo 4, quando si parlerà dell'interfaccia vocale realizzata in VoiceXML.

2.2.3 Topologia della pagina e semantica

Per ottenere lo stesso risultato che si ottiene con le tabelle (o addirittura un risultato migliore) si può ricorrere al box model dei CSS 2[65] e cioè in sostanza alla possibilità di disegnare la pagina definendone i blocchi e posizionandoli a piacimento nella finestra del web (assegnando eventualmente vari attributi quali margini, bordi, spaziatura dal bordo ecc. . .)

Sfruttando questo linguaggio quindi sarebbe possibile eliminare completamente l'uso delle tabelle. Il CSS consente di inserire gli elementi della pagina come se si trattassero di widget di una interfaccia grafica posizionati nel frame principale di una applicazione.

Certamente come già accennato, l'uso delle tabelle rende molto più semplice questo compito e in alcuni casi sembra essere indispensabile per riprodurre in una pagina web quanto previsto da un grafico che si occupa solo dell'aspetto e non della reale implementazione. In effetti c'è da dire che non tutti i browser supportano completamente e allo stesso modo i CSS e quindi spesso senza le tabelle il compito di chi deve impaginare può essere arduo. Oltretutto non è vero che le tabelle usate come layout sono sempre sintomo di inaccessibilità. Tuttavia, in mancanza di un meccanismo semplice per distinguere logicamente tabelle di dati da tabelle per layout ritengo che le tabelle vadano usate solo per i dati.

Non verrà qui approfondita la sintassi del CSS e la tecnica del box model per i quali si rimanda alla documentazione presente in letteratura.

Blocchi principali: menu e contenuto L'utilizzo del box model consente di evidenziare quello che ritengo essere uno dei criteri fondamentali per l'usabilità e l'accessibilità multimodale che è quello di riuscire a separare la pagina in componenti logici ben delimitati ai quali viene attribuito un significato particolare. I più importanti di questi blocchi di pagina sono il menu e il contenuto principale

Pensando all'accesso tramite interfaccia vocale o screen reader, il contenuto principale è quello che il lettore vorrà sicuramente ascoltare quando entra in una pagina mentre il menu risulta essere solo un accessorio. È possibile tuttavia che la prima volta che si entra in un sito il menu assuma un ruolo di primo piano in quanto presumibilmente dà all'utente un'idea di come è strutturato il sito e di quali siano i contenuti del sito stesso e che quindi guidi l'utente alla navigazione e alla ricerca delle informazioni per le quali è arrivato lì.

Pertanto nella home page il menu principale deve essere la prima cosa fatta ascoltare all'utente. Nelle altre pagine invece il menu principale assume un ruolo di secondo piano: serve solo per uscire dalla pagina quindi non ha senso che sia la prima cosa che venga fatta ascoltare all'utente. Deve però essere sempre accessibile per consentire in ogni momento di cambiare pagina senza dover arrivare in fondo (forse l'utente non ha trovato quello che cercava oppure semplicemente ha terminato la lettura e vuole andare da un'altra parte rimanendo nel sito).

Un modo per imporre allo screen reader l'ordine di lettura è quello di sfruttare i CSS e il posizionamento assoluto. I blocchi menu e contenuto possono essere definiti nell'HTML in un ordine qualsiasi, attraverso il CSS si può poi definire l'ingombro e la posizione di ciascun blocco all'interno della pagina in modo da ottenere graficamente quello che si vuole indipendentemente dal codice HTML. JAWS applica solo in parte le informazioni contenute nel CSS, non applica ad esempio il posizionamento ma legge il contenuto della pagina nell'ordine in cui è definito all'interno del codice HTML. In questo modo per definire l'ordine di lettura di un

blocco da parte dello screen reader basta rispettare quell'ordine all'interno del sorgente mentre la visualizzazione che ne darà il browser dipenderà esclusivamente dal CSS.

Per rendere la pagina facilmente navigabile, l'utente deve poter scegliere di saltare la lettura del menu per accedere direttamente al contenuto o viceversa. Si suggerisce pertanto di fornire sempre tramite delle opportune accesskey dei link interni alla pagina che puntino almeno a questi due elementi.

In generale, bisogna definire la topologia della pagina e la semantica di ogni blocco. È necessario poi avere qualche strumento per consentire all'utente di accedere in modo rapido a tali blocchi. Quello delle access key può essere valido per l'interfaccia grafica del browser. In seguito verrà illustrata una tecnica più generica che ha come obiettivo quello di consentire una analisi automatica della pagina in modo da poter definire varie modalità di accesso a tali componenti.

2.2.4 Uso del colore e del carattere

Un uso corretto del colore è di fondamentale importanza per l'accessibilità alle informazioni da parte degli ipovedenti e dei daltonici. Poiché non è un aspetto cruciale per l'accessibilità multimodale che è l'argomento principale di questo testo, si accennerà solamente a quelle che ritengo essere le pratiche migliori.

Il primo aspetto, fondamentale nell'uso del colore, è il contrasto tra il colore usato per il testo e quello dello sfondo. Indipendentemente da quello che può essere il gusto estetico, è importante avere un alto contrasto per fare in modo che i due colori non vengano confusi e il testo non risulti incomprensibile anche se viene ingrandito il carattere.

È possibile utilizzare delle tabelle che danno delle indicazioni pratiche su come utilizzare il colore per andare incontro ai problemi dei daltonici (ne esistono di diversi tipi). Esistono anche software che sono in grado di

simulare come una certa immagine venga vista da una persona con simili problemi di visione, in modo da avere un criterio oggettivo per scegliere al meglio i colori.

Anche un uso corretto del carattere è molto importante per rendere le informazioni più facilmente accessibili a persone con problemi di ipovisione. Anche in questo caso si tratta di accorgimenti che servono solamente a rendere l'interfaccia grafica più usabile per queste categorie di persone. Restano valide le indicazioni del W3C che non hanno in questo caso una applicazione interessante per quel che riguarda la multimodalità.

2.2.5 Uso delle immagini

Le immagini che vengono inserite nelle pagine web sono sostanzialmente di due tipi:

- immagini decorative, loghi, elementi che compongono l'interfaccia grafica
- immagini che convogliano delle informazioni e a cui possono fare i riferimenti i testi contenuti nella pagina

Dal punto di vista dell'accessibilità solo le seconde possono essere interessanti per tutti gli utenti. Pertanto è necessario fornire un testo alternativo che, per quanto possibile, veicoli un'informazione equivalente a quella trasmessa dall'immagine.

Per fare ciò si può cercare di descrivere a parole l'immagine e quindi legarle un testo attraverso l'attributo `alt` o collegarle un documento esterno attraverso `longdesc`.

Sfruttando il canale uditivo si potrebbe legare all'immagine anche un documento audio con una voce o dei suoni che descrivano l'immagine...

Per un non vedente potrebbe essere utile legare l'immagine ad una mappa audio/tattile che attraverso feedback vibrazionali e sonori ne consenta in qualche modo l'esplorazione. Un esempio di questa tecnica viene

illustrata nella sezione 3.8.2 a pagina 100. Sembra che possa essere applicabile soprattutto a immagini che riproducono mappe geografiche, mappe stradali, percorsi, strutture di edifici ecc... Questa tecnica non è stata ancora formalizzata essendo tuttora sperimentale. In seguito potrà essere definito un linguaggio (basato su XML) che possa aumentare l'accessibilità multimodale delle pagine web.

La prima categoria è costituita da immagini che non hanno alcuna utilità pratica ma solo estetica. Se si dovessero seguire alla lettera le linee guida del W3C o i risultati dei validatori automatici, saremmo costretti ad inserire testi alternativi che alla fine risulteranno inutili. Per queste immagini credo che si debba fare un'eccezione e, anche se la cosa sembra contraddittoria, che sia estremamente importante renderle inaccessibili! D'altra parte se non è possibile definire un testo alternativo valido, l'immagine non è altro che un elemento di disturbo per l'utente che non può vederla.

La stessa situazione la si può avere per quelle immagini che non convogliano particolari informazioni ma sono ad esempio usate al posto dei bottoni delle form o in sostituzione di testi per i link. Anche in questo caso infatti non è molto utile avere un testo alternativo che indichi semplicemente la funzione dell'immagine, l'utente potrebbe fare confusione tra ciò che è la spiegazione di una funzione e una semplice descrizione dell'immagine.

Ad esempio, si supponga di avere delle piccole immagini usate per navigare avanti e indietro tra le pagine di un lungo articolo. Presumibilmente le immagini saranno all'interno di un tag `a` per creare il collegamento. In tal modo, la funzione dell'immagine potrebbe essere indicata semplicemente attraverso il titolo del link: "Vai alla pagina successiva(precedente)". In tal caso sarebbe inutile un testo alternativo per l'immagine.

Il modo più semplice per rendere inaccessibile un'immagine è quello di inserire l'attributo `alt` vuoto. Molti screen reader infatti scelgono di

non annunciare la presenza di un'immagine se questa è sprovvista di testo alternativo.

Un metodo più sofisticato è quello di non inserire l'immagine nell'HTML ma definirla come sfondo attraverso un CSS. Di seguito viene mostrato il codice HTML e CSS di esempio di questa tecnica:

```
1 <style type='text/css'>
2 #frecciadx{
3   width: 15em;
4   height: 15em;
5   background: url(img/frecciadx.gif) no-repeat;
6 }
7 </style>
8 ...
9 <a href='...'
10   title='Link per andare alla pagina successiva'>
11 <div id='frecciadx'>
12   Pagina successiva
13 </div>
14 </a>
```

Il codice mostra di nuovo un esempio di separazione tra presentazione (CSS) e contenuto (HTML). Lo screen reader in questo caso non può nemmeno percepire la presenza dell'immagine (se non è in grado di analizzare il CSS). Leggerà semplicemente il link e il testo contenuto all'interno del tag `div`. Questa tecnica risulta utile anche nel caso in cui la pagina venga trascodificata per creare un'interfaccia vocale.

2.2.6 Tabelle HTML e problema della serializzazione

Spesso, quando si realizzano tabelle HTML per accogliere dati, si pensa soprattutto all'effetto visivo e non si attribuisce il giusto significato logico a ciascun elemento della tabella. Attribuire una semantica alle celle è invece molto importante per chi deve utilizzare strumenti assistivi per la navigazione e per consentire ad un sistema automatico di adattare il contenuto a diversi canali. Ancora una volta molto importante risulta essere il canale vocale e quindi il problema della serializzazione delle tabelle di dati.

Probabilmente, per incrementare l'accessibilità multimodale delle tabelle potrebbe essere opportuno studiare un modo per definirle, cercando di separarne la struttura e la semantica dei dati dalla presentazione, analogamente a quanto si è fatto per le form con XForm [41].

In ogni caso il linguaggio HTML è piuttosto ricco anche per la costruzione delle tabelle in quanto consente di definire in modo logico, oltre che visivo, le celle che sono da intendersi come intestazioni di righe o di colonne, distinguendole da quelle di puri dati utilizzando in modo opportuno gli attributi `scope`, `headers` ecc...

Tuttavia esistono vari modi per ottenere lo stesso risultato e spesso c'è un po' di confusione su come usare al meglio tali attributi per renderli veramente utili ad esempio a strumenti di ausilio per non vedenti.

Di seguito vediamo brevemente le indicazioni principali per definire al meglio una tabella di dati:

- usare l'attributo `summary` del tag `table` per descrivere nel modo più dettagliato possibile la tabella. Indicare il suo contenuto ed eventualmente le "coordinate" dei dati più significativi.
- dare un titolo alla tabella usando il tag `caption`
- inserire le celle di intestazione in tag `th` e quelle di dati in tag `td`
- utilizzare, se ritenuto opportuno, l'attributo `abbr` per le celle di intestazione in modo da dare un'abbreviazione della cella da usarsi durante la lettura (il valore viene sfruttato dagli screen reader)
- indicare in modo esplicito a quali celle si applicano le intestazioni definite. Esistono vari metodi:
 - per tabelle con struttura semplice, usare `scope=col` oppure `scope=row` per indicare che la cella `th` corrispondente è la cella di intestazione per tutta la colonna o tutta la riga
 - per tabelle più complesse, con intestazioni multiple e diversi punti di accesso, identificare le celle di intestazione con l'at-

tributo `id` e in ciascuna cella di dati usare l'attributo `headers` per elencare gli `id` delle celle di intestazione corrispondenti

- raggruppare in modo opportuno le righe usando i tag `thead`, `tbody`, `tfoot`

Per una descrizione più dettagliata di questi e altri tag e attributi per la definizione delle tabelle si rimanda ai manuali e specifiche dell'HTML. Mi limito a presentare un esempio di come è possibile creare tabelle accessibili in modo multimodale.

Esempio In questo esempio viene mostrato il codice HTML di una tabella di dati, scritto senza tenere in considerazione i suddetti criteri.

```
1 Indici Trading After Hours
2 <table>
3   <tr>
4     <td><b>Nome</b></td>
5     <td><b>Valore</b></td>
6     <td><b>Var %</b></td>
7   </tr>
8   <tr>
9     <td><b>MIB30-s</b></td>
10    <td>31.353</td>
11    <td>+0,01</td>
12  </tr>
13  <tr>
14    <td><b>MIBTEL-s</b></td>
15    <td>23.830</td>
16    <td>+0,01</td>
17  </tr>
18  <tr>
19    <td><b>NUMTEL-s</b></td>
20    <td>1.270</td>
21    <td>-0,08</td>
22  </tr>
23 </table>
```

Di seguito, la stessa tabella resa accessibile

```
1 <table
2   summary="La tabella mostra i valori degli indici di borsa principali.
3   I valori risultano pressoché invariati rispetto all'ultima rilevazione">
4   <caption>Indici Trading After Hours</caption>
5   <thead>
```

```
6 <tr>
7 <th scope="col"
8     abbr="indice">Nome</th>
9 <th scope="col">Valore</th>
10 <th scope="col"
11     abbr="variazione percentuale">Var %</th>
12 </tr>
13 </thead>
14 <tbody>
15 <tr>
16 <th scope="row">MIB30-s</th>
17 <td>31.353</td>
18 <td>+0,01</td>
19 </tr>
20 <tr>
21 <th scope="row">MIBTEL-s</th>
22 <td>23.830</td>
23 <td>+0,01</td>
24 </tr>
25 <tr>
26 <th scope="row">NUMTEL-s</th>
27 <td>1.270</td>
28 <td>-0,08</td>
29 </tr>
30 </tbody>
31 </table>
```

Graficamente le due tabelle appaiono nel browser nel modo indicato dalla figura 2.1

Indici Trading After Hours			Indici Trading After Hours		
Nome	Valore	Var %	Nome	Valore	Var %
MIB30-s	31.353	+0,01	MIB30-s	31.353	+0,01
MIBTEL-s	23.830	+0,01	MIBTEL-s	23.830	+0,01
NUMTEL-s	1.270	-0,08	NUMTEL-s	1.270	-0,08

Figura 2.1: Due tabelle HTML

Come si vede le tabelle appaiono praticamente identiche, tuttavia la seconda è stata arricchita da attributi che forniscono di logica i contenuti mostrati dal browser.

Utilizzando il transcoder da HTML a VoiceXML descritto nel capitolo 4, ecco come vengono lette le due tabelle dal sintetizzatore vocale (in assenza di comandi da parte dell'utente).

Prima tabella (versione non accessibile):

Indici Trading After Hours

Inizio di una tabella senza titolo

Inizio lettura delle righe

Riga 1. Nome, Valore, Var %

Riga 2. MIB30-s, 31.353, +0,01

Riga 3. MIBTEL-s, 23.830, +0,01

Riga 4. NUMTEL-s, 1.270, -0,08

Fine della tabella.

Seconda tabella (versione accessibile):

Inizio di una tabella il cui titolo è: Indici Trading After Hours

La descrizione della tabella è la seguente: La tabella mostra i valori degli indici di borsa principali. I valori risultano pressoché invariati rispetto all'ultima rilevazione

Inizio lettura delle righe

Riga 1:

MIB30-s, Valore, 31.353

MIB30-s, variazione percentuale, +0,01

Riga 2:

MIBTEL-s, Valore, 23.830

MIBTEL-s, variazione percentuale, +0,01

Riga 3:

NUMTEL-s, Valore, 1.270

NUMTEL-s, variazione percentuale, -0,08

Fine della tabella.

La caratterizzazione delle celle consente di creare una interfaccia vocale sofisticata che possa permettere all'utente di ottenere le informazioni

impartendo comandi del tipo: “Valore dell'indice MIBTEL-s”, “Variazione percentuale dell'indice MIBTEL-s” e così via.

2.2.7 Form e interattività

Le form per il web costituiscono il grado più elevato di interattività con l'utente in quanto non forniscono semplicemente informazioni che devono essere lette ma soprattutto ne richiedono all'utente e lo fanno in modi diversi.

Quando si utilizza uno screen reader e si accede ad una form ci si rende immediatamente conto della difficoltà che può avere l'utente nell'interpretarla correttamente e nell'inserire le informazioni richieste in modo corretto.

Il problema nasce dal modo in cui le form HTML sono realizzate e sono visualizzate nel browser: i campi di input non sono sensati se estratti dal contesto visivo in quanto in genere si utilizza del testo messo “vicino” ad essi per far capire all'utente qual'è l'informazione che viene richiesta. Spesso, per allineare etichette e campi di input, si utilizzano delle tabelle che complicano ulteriormente la navigazione con lo screen reader.

Fortunatamente l'HTML mette a disposizione alcuni tag che aggiungono importanti informazioni ai campi e alla form nel suo complesso che risultano essere molto importanti se si naviga con l'ausilio di uno screen reader.

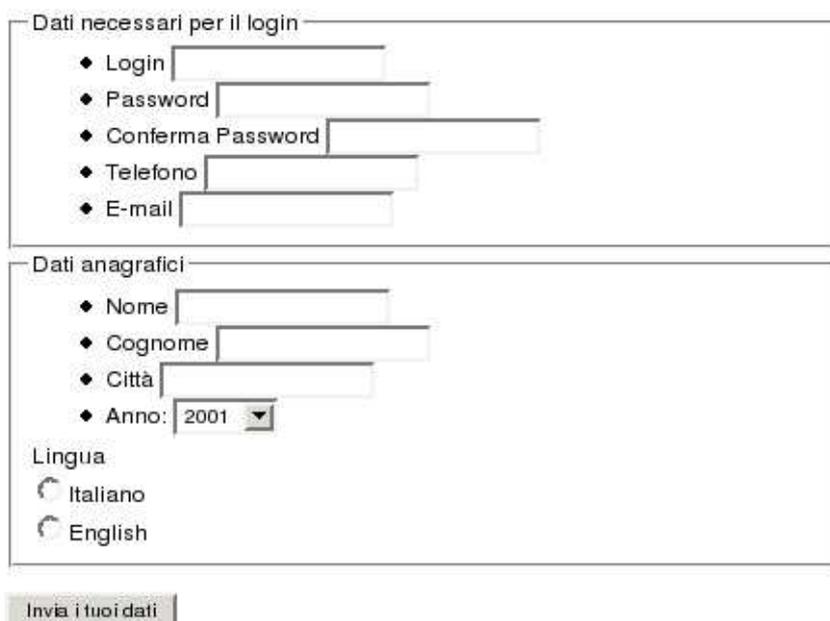
Ad esempio il tag `label` consente di associare logicamente un testo ad un campo, cosicché lo screen reader, quando legge il campo di input, legge anche la sua etichetta.

Inoltre i campi possono essere raggruppati logicamente fra loro utilizzando il tag `fieldset` al quale viene associata un'etichetta. Lo screen reader legge l'etichetta del gruppo di appartenenza di ciascun campo.

Esempio In questo esempio vediamo una form per la registrazione di un utente, nel quale vengono chiesti alcuni dati per l'autenticazione e alcuni campi anagrafici. La prima versione verrà scritta senza tener conto di alcun criterio di accessibilità mentre nella seconda si cercherà di renderla accessibile. Per ciascuna vediamo una trascrizione dell'interazione con la form effettuata con JAWS.

La figura 2.2 mostra una form HTML scritta usando opportunamente i tag suggeriti per l'accessibilità, così come appare in browser.

Form di registrazione



The image shows a registration form with two main sections: "Dati necessari per il login" and "Dati anagrafici". Each section is enclosed in a box with a title bar. The "Dati necessari per il login" section contains five fields: "Login", "Password", "Conferma Password", "Telefono", and "E-mail", each preceded by a diamond-shaped bullet point. The "Dati anagrafici" section contains four fields: "Nome", "Cognome", "Città", and "Anno", each preceded by a diamond-shaped bullet point. The "Anno" field is a dropdown menu showing "2001". Below the "Dati anagrafici" section, there is a "Lingua" section with two radio buttons: "Italiano" and "English". At the bottom of the form is a button labeled "Invia i tuoi dati".

Figura 2.2: Esempio di form accessibile

Di seguito viene trascritta una parte della form. Si vede come i campi per il login siano stati raggruppati in un tag `fieldset` (lo stesso è fatto per i dati anagrafici).

A ciascun campo è stata associata un'etichetta che appare comunque ovvia nel momento in cui la form viene visualizzata ma che con l'uso del tag

label diviene esplicita e usabile dagli screen reader e da altri eventuali ausili.

```
1 ...
2 <fieldset>
3 <legend>Dati necessari per il login</legend>
4 <ul>
5   <li>
6     <label for="logininput">Login</label>
7     <input tabindex="108"
8       id="logininput"
9       type="text"
10      name="login"
11      value="">
12   </li>
13   <li>
14     <label for="passwordinput">Password</label>
15     <input tabindex="109"
16       id="passwordinput"
17       type="password"
18       name="password"
19       value="">
20   </li>
21   <li>
22     <label for="confpasswordinput">Conferma Password</label>
23     <input tabindex="110"
24       id="confpasswordinput"
25       type="password"
26       name="passwordconfirm"
27       value="">
28   </li>
29   <li>
30     <label for="phoneinput">Telefono</label>
31     <input tabindex="111"
32       id="phoneinput"
33       type="text"
34       name="phone"
35       value="">
36   </li>
37   <li>
38     <label for="emailinput">E-mail</label>
39     <input tabindex="112"
40       id="emailinput"
41       type="text"
42       name="email"
43       value="">
44   </li>
```

```
45 </ul>
46 </fieldset>
47 ...
```

Come si nota, i campi della form sono inseriti in un lista. Attraverso un opportuno CSS è possibile eliminare i punti in modo che la lista non appaia come tale. Adottando questa tecnica si fa in modo che lo screen reader indichi il numero di elementi della lista, cosicché l'utente può già conoscere il numero di campi che dovrà riempire.

Di seguito vediamo la trascrizione di un possibile utilizzo di questa form con JAWS, le parti in corsivo sono lette dal sintetizzatore vocale, in maiuscoletto le azioni compiute dall'utente:

Titolo di livello 1 Form di registrazione

FRECCIA GIÙ

Dati necessari per il login

FRECCIA GIÙ

Lista di 5 elementi

FRECCIA GIÙ

Elemento Login FRECCIA GIÙ

Campo di testo

ENTER

Modalità form inserita

Dati necessari per il login, Login, inserisci il testo

INSERIMENTO DEL CAMPO

TAB

Dati necessari per il login, Password, inserisci il testo

INSERIMENTO DEL CAMPO

TAB

Dati necessari per il login, Conferma password, inserisci il testo

INSERIMENTO DEL CAMPO

TAB

...

...

Dati anagrafici, Nome, inserisci il testo

INSERIMENTO DEL CAMPO

TAB

Dati anagrafici, Cognome, inserisci il testo

INSERIMENTO DEL CAMPO

TAB

...

...

Dati anagrafici, Anno, Combo box 2001, 1 di 99

SELEZIONE CON LE FRECCE SU-GIU

TAB

Dati anagrafici Lingua Italiano, radio button non selezionato, per cambiare la selezione premere le frecce su o giù

SPAZIO

TAB

Bottone Invia i tuoi dati

SPAZIO

Accesso multimodale alle form Un corretto uso dell'HTML consente di avere delle form facilmente utilizzabili anche dai non vedenti, attraverso lo screen reader.

Il linguaggio HTML non è però sufficiente a definire form pienamente accessibili in maniera multimodale. Nel capitolo 4, quando si illustrerà una trascodifica delle form HTML in interfaccia vocale emergeranno subito i limiti espressivi del linguaggio. Soprattutto il canale vocale/uditivo che è il più importante dopo quello della vista risente di questa mancanza.

Quello che manca soprattutto è una netta separazione tra la struttura della form e la sua presentazione, manca inoltre una tipizzazione dei campi che specifichi bene il tipo di dato richiesto in modo che si possa creare una interfaccia adatta all'inserimento indipendentemente dalla modalità

di interazione usata dall'utente.

La soluzione proposta dal W3C che cito solamente in questo contesto è quella di adottare XForms [41] che va incontro esattamente a questa esigenza di separazione tra struttura e presentazione. Per la precisione la separazione è tra modello, istanza (lega i dati al modello) e presentazione.

È auspicabile una ampia diffusione di questo standard, che vada a soppiantare le vecchie form dell'HTML, ma prima di tutto sarebbe necessario che i browser più diffusi inizino a supportarlo.

Ritengo che XForms vada preso in considerazione per realizzare siti ad alta accessibilità multimodale e può già essere adottato per i sistemi basati su un transcoder per la presentazione di contenuti in ambienti multicanale. Adottando XForms come linguaggio per la definizione dei moduli risulta poi molto più semplice la trascodifica per la presentazione attraverso interfacce grafiche, vocali o miste.

2.2.8 Arricchire la pagina con metainformazioni

Come detto quando si pensa di realizzare una pagina web pienamente fruibile anche attraverso uno screen reader, spesso può essere necessario fornire all'utente informazioni e strumenti di navigazione che normalmente non sarebbero necessari per utenti che utilizzano un normale browser.

Queste elementi sono ad esempio brevi testi che indichino in quale punto o area della pagina ci si trova (inizio o fine menu, inizio o fine contenuto principale, primo paragrafo ecc...) o link ai blocchi stessi.

Nasce dunque la necessità di inserire testi e link che siano letti dagli screen reader ma che non siano visti (per quanto possibile, a meno di non leggere il sorgente HTML) da chi usa browser visuali.

Esempio L'esempio che segue mostra una porzione di codice HTML che rappresenta un piccolo menu navigazionale.

```
1 <a name="mainmenu"></a>  
2 <span style="width:0px;height:0px;">
```

```
3   Inizio del menu principale
4 </span>
5 <ul>
6 <li><a href="..." title="...">Home page</a></li>
7 <li><a href="..." title="...">Prodotti</a></li>
8 <li><a href="..." title="...">Informazioni</a></li>
9 <li><a href="..." title="...">Persone</a></li>
10 <li><a href="..." title="...">Contatti</a></li>
11 </ul>
12 <span style="width:0px;height:0px;">
13   Fine del menu principale
14 </span>
```

Il browser visuale mostrerà solo l'elenco puntato di link mentre manterrà nascosti i due testi “Inizio del menu principale” e “Fine del menu principale” posti rispettivamente prima e dopo l'elenco.

Infatti i due testi sono stati inseriti in blocchi di dimensione nulla attraverso l'attributo `style` pertanto non saranno visualizzati.

Uno screen reader come JAWS invece leggerà lo stesso le informazioni dando quindi all'utente che non vede la pagina, informazioni ulteriori che sono invece implicite nella visualizzazione grafica.

2.2.9 Navigazione senza mouse

Osservando un non vedente che utilizza lo screen reader si nota immediatamente che egli fa un uso massiccio del tasto di tabulazione per passare da un link all'altro presente nella pagina. Man mano che si porta il focus su un nuovo elemento, lo screen reader legge il link e in tal modo l'utente può farsi una prima di idea di che cosa ci sia nella pagina o meglio di dove si possa arrivare. Il mouse non può essere usato come strumento di navigazione dai non vedenti, se non altro non nella maniera classica, pertanto è importante fornire opportuni strumenti che migliorino l'uso della tastiera in modo da colmare questa mancanza.

Un altro approccio è quello di chiedere allo screen reader di elencare tutti i link presenti nella pagina in modo che l'utente possa scegliere con

le frecce quello che interessa e seguirlo premendo semplicemente il tasto di invio.

Entrambe questi approcci suggeriscono di utilizzare in modo opportuno i tasti di scelta rapida (i cosiddetti *accesskey*) e l'ordine di tabulazione. Mediante l'attributo `accesskey` è possibile assegnare una lettera o un numero ad un link o campo di input di una form. Per accedere all'elemento, l'utente non dovrà far altro che digitare `ALT+<CARATTERE>` per portare il focus sul link o campo di input prescelto.

La scoperta da parte dell'utente di quali siano i tasti di accesso può essere ottenuta solo tramite l'esperienza di navigazione, a meno di avere una pagina di istruzioni che dica all'utente quali sono i tasti tipici usati nel sito. Non va abusato di questa possibilità in quanto non si può pretendere che un utente si ricordi di tutti i tasti definiti dallo sviluppatore. L'importante è mantenere una certa coerenza nell'uso dei tasti.

Una pratica che posso suggerire e che sembra risultare comoda anche per chi non utilizza uno screen reader è quella di assegnare una lettera a ciascun blocco topologico in cui è suddivisa la pagina: "k" per il contenuto principale, "m" per il menu della pagina. All'interno dei menu poi può essere molto comodo numerare le voci e quindi "1" per la prima voce, "2" per la seconda e così via.

Parallelamente a questa pratica è molto importante fare in modo che un utente che esplori la pagina usando il tasto di tabulazione, segua un percorso ben preciso e che si pensa essere il migliore.

Il modo per farlo è utilizzare l'attributo `tabindex` su link e campi di input. Tali elementi vengono ordinati in base al valore numerico dell'attributo. Quando uno di questi acquisisce il fuoco, nel momento in cui l'utente preme ancora TAB, il fuoco passa all'elemento successivo; premendo `SHIFT+TAB` si passa all'elemento precedente. Non è necessario usare numeri consecutivi, anzi può essere comodo usare ad esempio multipli di dieci in modo da avere la possibilità di inserire nuovi link senza dover rinumerare tutti i successivi.

Ad esempio, nel caso del menu, bisogna fare in modo che i link abbiano un valore di `tabindex` che rispettino l'ordine in cui compaiono le voci. Se si usano i numeri come access key per le voci di menu, nel caso in cui le voci siano più di dieci, è importante che l'utente possa continuare a scorrere nell'ordine giusto le voci.

L'ordine di attribuzione del `tabindex` è quindi importante in quanto permette in qualche modo di stabilire le priorità delle informazioni presenti nella pagina e suggerisce quale deve essere l'ordine di lettura.

2.2.10 Usare un linguaggio accessibile

Una delle linee guida del W3C dice di utilizzare un linguaggio accessibile al maggior numero di persone possibile. In realtà il linguaggio dovrebbe essere adeguato al tipo di pubblico cui ci si rivolge. Può essere opportuno scrivere testi alternativi che vengono presentati al momento opportuno in base al profilo dell'utente.

L'utilizzo di un linguaggio semplice, di una sintassi corretta e di una adeguata punteggiatura porta con sé anche un altro importante vantaggio che è quello di rendere le sintesi vocali molto più chiare e intelleggibili. Questo è l'aspetto più importante per quanto riguarda l'accessibilità multimodale delle informazioni. È infatti chiaro che è più facile comprendere un testo scritto piuttosto di un testo ascoltato, pertanto è necessario semplificare il contenuto proprio da un punto di vista cognitivo per renderlo più accessibile attraverso la sintesi vocale.

Per quanto riguarda la lingua italiana si dovrebbero applicare delle semplici regole, indipendenti dal tipo di utenza, che sono state ben indicate da Tullio De Mauro e Raffaele Simone nel 1981. In sintesi, le linee guida principali sono:

- Scrivere frasi brevi: non più di 25 o 30 parole e non più di 2 frasi per ogni periodo.
- Scrivere frasi semplici e lineari: una sola proposizione con un sog-

getto (meglio se esplicito), un verbo e qualche complemento, possibilmente in quest'ordine e non separati da incisi.

- Evitare le espressioni prolisse, le perifrasi, le circonlocuzioni, e preferire invece parole ed espressioni brevi.
- Evitare gli anglicismi (ma anche prestiti e calchi linguistici da altre lingue straniere), se esistono parole italiane equivalenti e più comunemente usate.
- Evitare le forme arcaiche, dotte, letterarie e usare invece parole ed espressioni comuni, della vita di tutti i giorni, e possibilmente brevi.
- Privilegiare la forma attiva dei verbi a quella passiva
- Preferire il più possibile i verbi ai sostantivi e dunque evitare il più possibile la nominalizzazione, cioè la trasformazione dei verbi in sostantivi.
- Specificare sempre il soggetto della frase.
- Preferire la forma affermativa delle frasi a quella negativa, e in particolare evitare la doppia negazione.
- Preferire, quando possibile, modi e tempi verbali semplici: ad esempio l'indicativo è più semplice, nel senso che viene compreso più frequentemente e rapidamente, del congiuntivo e del condizionale, il tempo presente, il passato prossimo e il futuro sono più semplici del passato remoto e futuro anteriore.
- Preferire le parole concrete a quelle astratte. Le parole concrete si riferiscono a oggetti, eventi, esperienze che sono percepibili con uno dei cinque sensi.

Un altro aspetto di più alto livello da tenere in considerazione durante la redazione dei testi, è la separazione in paragrafi. Le frasi che sono legate allo stesso concetto vanno raggruppate in paragrafi brevi, bisogna iniziare un nuovo paragrafo ogni volta che si introduce qualche elemento nuovo,

slegato dal precedente. In questo modo, oltre a rendere più comprensibile il contenuto e più gradevole la lettura attraverso l'interfaccia grafica, consente di costruire un'interfaccia vocale con un'elevata granularità. Permette cioè di presentare il testo separato in blocchi navigabili dall'utente attraverso comandi vocali. Tale separazione dovrà essere esplicitata nel codice HTML inserendo i vari paragrafi in tag `p` separati.

Questo aspetto verrà ripreso nel capitolo 4 quando verrà appunto creata in automatico un'interfaccia vocale a partire da contenuti presi dal web.

Per quanto riguarda il lessico usato, per la lingua italiana è consigliabile fare riferimento al dizionario di base elaborato da De Mauro [62] soprattutto, ma non solo, se gli utenti destinatari hanno difficoltà cognitive, di lettura, di comprensione del linguaggio (dislessici oppure stranieri)

Sempre per quanto riguarda la lingua italiana, un ottimo strumento per valutare l'accessibilità dei testi e quindi i criteri di leggibilità esposti e l'aderenza al vocabolario di base, si consiglia di utilizzare il sistema *Censor* di *Éulogos* (vedi 1.9).

2.3 Web ontology e web semantico

Veniamo ora ad un metodo per incrementare l'accessibilità delle informazioni, basato sull'idea che è necessario rendere usabile l'interfaccia dell'utente indipendentemente dalle modalità di accesso.

Il metodo deve essere sufficientemente flessibile da consentire anche una trasformazione automatica delle pagine per creare l'interfaccia più adatta alle esigenze dell'utente.

Il primo aspetto dell'usabilità multimodale, è la possibilità di individuare i componenti logici in cui è divisa ciascuna pagina, la semantica dei contenuti e la struttura del sito e perciò le relazioni tra le pagine.

Quello del web semantico è uno dei problemi aperti tra i più importanti e affascinanti[67][31].

Il linguaggio HTML non è sufficiente a questo scopo pertanto sono stati

sviluppati diversi linguaggi basati soprattutto su XML per descrivere la semantica della pagine.

Quello più importante è RDF (Resource Description Framework)[21], un XML per la descrizione delle risorse in genere che si può utilmente applicare al web. Per quanto riguarda le ontologie è stato definito dal W3C lo standard OWL (Web Ontology Language)[54] definito a partire dai linguaggi DAML e OIL [18] come estensione del RDF proprio per arricchire i contenuti di una semantica in modo tale da poter essere processati in modo automatico.

Prendendo spunto da queste problematiche e da questi linguaggi sono arrivato alla scelta di RDF come linguaggio per la creazione di documenti da collegare alle pagine HTML per descriverne la struttura e facilitare quindi la creazione automatica di interfacce vocali a partire dalle interfacce grafiche definite nell'HTML (4.3) e in generale per ottenere l'accessibilità multimodale.

Essendo RDF una istanza di XML è per sua natura estendibile, pertanto si possono utilizzare altre specifiche per la definizione formale delle possibili modalità di interazione con i contenuti. Si può utilizzare ad esempio il linguaggio EMMA [73] per definire come e con quali modalità di interazione si accede alle informazioni. Si possono definire ad esempio comandi vocali, gesture, comandi da tastiera ecc. . .

In una prima fase dell'implementazione l'idea è quella di utilizzare documenti semplici, che al limite possano essere creati in automatico utilizzando i normali programmi di web authoring o altri strumenti di progettazione dei siti [8], in modo che l'impatto sui siti esistenti non sia troppo elevato. I comandi vocali vengono definiti in modo implicito dal contenuto dell'RDF mentre le gesture non sono definite.

I documenti RDF da creare sono i seguenti:

index : è l'indice del contenuto della pagina. In realtà serve semplicemente ad individuare il contenuto principale, il menu ed eventuali al-

tri componenti. L'utente che accede tramite interfaccia vocale vorrà ascoltare prima il contenuto e a richiesta accedere al menu.

contents : è l'indice del contenuto del sito. Permette all'utente di navigare all'interno del sito senza perdere l'orientamento.

Chiaramente "index" è tipico di ogni pagina, anche se l'uso di template standard consente di fatto di crearlo una volta per tutte. "contents" essendo la mappa del sito sarà presumibilmente lo stesso per ogni pagina, tuttavia nulla vieta di generarlo dinamicamente anche per permettere di avere un menu che varia a seconda del contesto: area del sito in cui ci si trova, privilegi di accesso dell'utente ecc. . .

Per collegare questi documenti alla pagina si sfrutta il tag `link` che è preposto proprio allo scopo di collegare documenti esterni alla pagina HTML. L'attributo `rel` indica il tipo di relazione del documento collegato con la pagina corrente, ed è qui che si indica "index" oppure "contents". È possibile specificare anche il content-type del documento collegato attraverso l'attributo `type`¹.

Ecco quindi come si presenta ad esempio l'intestazione di una pagina HTML:

```
1 <html>
2 <head>
3   ...
4   <link rel=""index"" type=""application/rdf+xml""
5       href=""index.rdf"">
6
7   <link rel=""contents"" type=""application/rdf+xml""
8       href=""contents.rdf"">
9   ...
10 </head>
11 ...
12 </html>
```

¹Il browser Opera è in grado di leggere e interpretare correttamente i documenti collegati alla pagina in questo modo e consente inoltre di visualizzarli.

2.3.1 Esempio di documenti “index” e “contents”

Di seguito un frammento di un possibile documento “index” di una pagina costituita da un contenuto principale e da un menu. Come si può intuire, non è necessario conoscere il layout del documento per poter costruire una interfaccia efficace che tralasci tutte le informazioni superflue, è sufficiente che la pagina sia stata creata seguendo criteri di accessibilità e che in particolare i blocchi siano inseriti in tag `div` identificati da un ID, in questo caso “content” e “menu”. In tal modo non è necessario scrivere complicate espressioni XPointer per “segnare” i blocchi della pagina HTML[40].

Ritengo che non sia necessario usare particolari estensioni dell’RDF per caratterizzare particolari sezioni della pagina in modo da discriminare i contenuti in base all’importanza o per indicare contenuti alternativi in base al tipo di device utilizzato, alla velocità della linea ecc. . . , come invece fatto per scopi simili da Z. Shao, Pèrez-Quiñones e altri [75], K. Nagao [34], M. Jang e altri [44].

Queste estensioni, pur essendo utili in certi casi, sono sicuramente molto difficili da applicare soprattutto in un sito dinamico nel quale spesso si utilizzano dei template per le pagine mentre i contenuti provengono da fonti dati differenti quali database e documenti XML. Per ciascun documento prodotto bisognerebbe realizzare una annotazione ad-hoc, il che potrebbe risultare improponibile.

```
1 <channel rdf:about="...">
2     <title>...</title>
3 <description></description>
4 <link></link>
5 <items>
6     <rdf:Seq>
7         <rdf:li rdf:resource="#content"/>
8         <rdf:li rdf:resource="#menu"/>
9     </rdf:Seq>
10 </items>
11 </channel>
12
13 <item rdf:about="#content">
```

```
14         <title>Contenuto principale</title>
15         <description></description>
16         <link>#content</link>
17     </item>
18 <item rdf:about="#menu">
19     <title>Menu</title>
20     <description></description>
21     <link>#menu</link>
22 </item>
```

Il primo elemento della lista *deve* essere quello relativo al contenuto principale della pagina. In questo modo si attribuisce implicitamente una certa semantica al contenuto della pagina caratterizzando i blocchi da cui è costituita.

In questo caso abbiamo una struttura molto semplice ma che rispetta di fatto gran parte delle pagine web che si trovano normalmente su internet. Sebbene sia possibile creare strutture gerarchiche ritengo più opportuno realizzare pagine topologicamente semplici, costituite da pochi blocchi non sovrapposti in modo tale che se l'utente che accede al sito lo fa attraverso screen reader o interfaccia vocale non rischi di perdere l'orientamento.

I valori dei tag `link` sono degli indirizzi relativi, sono gli identificativi dei blocchi interni alla pagina definiti attraverso l'attributo `id` di quello che solitamente è un tag `div` che contiene l'intero blocco.

La pagina HTML è quindi strutturata come nel codice che segue:

```
1 <html>
2 <head>
3   <link rel=""index"" type=""application/rdf+xml""
4     href=""index.rdf"">
5
6   <link rel=""contents"" type=""application/rdf+xml""
7     href=""contents.rdf"">
8 </head>
9 <body>
10  <div id=""menu"">
11    ...
12  </div>
13  <div id=""content"">
14    ...
```

```
15 </div>
16 </body>
17 </html>
```

In questo caso il documento “index” definisce topologicamente l'intera pagina. In effetti è possibile eliminare dall'RDF tutto quello che non si ritiene fondamentale rendere accessibile come ad esempio un banner pubblicitario e quindi identificare blocchi di pagina più piccoli che non coprano l'intero contenuto visibile all'interno di un browser.

Il documento “contents” è del tutto simile anche se è prevista la possibilità di creare menu gerarchici che tuttavia risultano più difficilmente comprensibili una volta serializzati in una interfaccia vocale.

```
1 <channel rdf:about="http://itc4d.weblearn.info/">
2   <title>Progetto ITC4D</title>
3 </description>
4 <link>http://itc4d.weblearn.info/</link>
5 <dc:date>2004-11-01T20:12:20Z</dc:date>
6 <items>
7 <rdf:Seq>
8   <rdf:li rdf:resource="http://itc4d.weblearn.info/home.jsp"/>
9   <rdf:li rdf:resource="http://itc4d.weblearn.info/people.jsp"/>
10  <rdf:li rdf:resource="http://itc4d.weblearn.info/infoprog.jsp"/>
11  <rdf:li rdf:resource="http://itc4d.weblearn.info/documents.jsp"/>
12 </rdf:Seq>
13 </items>
14 </channel>
15 <item rdf:about="http://itc4d.weblearn.info/home.jsp">
16   <title>Pagina iniziale</title>
17 </item>
18 <item rdf:about="http://itc4d.weblearn.info/people.jsp">
19   <title>Persone coinvolte</title>
20 </item>
21 <item rdf:about="http://itc4d.weblearn.info/infoprog.jsp">
22   <title>Informazioni progetto</title>
23 </item>
24 <item rdf:about="http://itc4d.weblearn.info/documents.jsp">
25   <title>Documentazione</title>
26 </item>
```

Il documento “contents” è facilmente trasformabile in menu vocale e i titoli degli item, oltre al numero di sequenza associabile ad un tono DTMF,

possono costituire la grammatica di riconoscimento di ciascuna voce. È quindi opportuno fare attenzione alla scelta di questi titoli pensando anche a questa finalità. Ad esempio non utilizzare titoli troppo lunghi in quanto chi usa l'interfaccia vocale spesso tende a ripetere letteralmente la voce che vuole selezionare; evitare inoltre di usare titoli simili tra loro che possono generare ambiguità in fase di riconoscimento. I sistemi di riconoscimento vocale infatti spesso consentono all'utente di pronunciare solo parzialmente una voce di menu (ad esempio solo alcune parole) e sceglieranno poi quella più verosimile. Se però le voci usano spesso le stesse parole è possibile creare ambiguità in caso di riconoscimento parziale.

Per entrambe i documenti, nel caso di utilizzo di EMMA per la definizione dell'interfaccia multimodale, basta aggiungere il nodo `emma` all'interno del tag `item` corrispondente. I comandi definiti saranno interpretati come modalità di accesso al corrispondente elemento o pagina del sito. Di seguito un esempio di definizione di un comando vocale, di una gesture e di un comando da tastiera usati come alternative per l'accesso al menu della pagina:

```
1 <item rdf:about="#menu">
2   <title>Menu</title>
3   <description></description>
4   <link>#menu</link>
5   <emma:emma version="1.0" xmlns:emma="http://www.w3.org/2003/04/emma">
6     <emma:one-of id="r1">
7       <emma:interpretation id="int1"
8         emma:medium="acoustic"
9         emma:mode="speech"
10      >
11       <rawinput>menu</rawinput>
12     </emma:interpretation>
13     <emma:interpretation id="int2"
14       emma:medium="tactile"
15       emma:mode="ink"
16     >
17       <points>
18         227 50 226 64 225 78 227 92 228 106 228 120 229 134
19         230 148 234 162 235 176 238 190 241 204
20       </points>
21     </emma:interpretation>
22     <emma:interpretation id="int3"
23       emma:medium="tactile"
24       emma:mode="keys"
25   >
```

```
26         <rawinput>m</rawinput>
27     </emma:interpretation>
28 </emma:one-of>
29 </emma:emma>
30 </item>
```

2.4 Usare i validatori automatici

Tra i limiti delle WCAG, ritengo che ci sia anche il fatto di non aver sottolineato adeguatamente la differenza tra quelli che sono gli accorgimenti per rendere flessibile e adattabile la struttura della pagina, dal rendere i contenuti accessibile da un punto di vista cognitivo e semantico. Molti sembrano ritenere che l'accessibilità consista semplicemente in un lavoro fatto sul codice HTML per rispettare le linee guida tecniche. Non è difficile infatti imbattersi in siti che si fregiano dei famosi bollini "A" "AA" e "AAA" di conformità, salvo poi rendersi conto che ad esempio i testi alternativi per le immagini o i titoli dei link non sono affatto adeguati, oppure che il linguaggio usato nel testo è a tratti incomprensibile.

Il problema nasce probabilmente dall'uso dei validatori automatici (Bobby, Cinthia, Torquemada ecc...) che riescono a dare solo una valutazione oggettiva di accessibilità, rilevando solo gli errori palesi, ma che ovviamente non possono valutare correttamente l'accessibilità dal punto di vista cognitivo.

La verifica dell'accessibilità deve pertanto essere condotta solo in prima istanza da validatori automatici, il cui intervento deve servire solo ad evitare errori banali. Ritengo indispensabile, per raggiungere la vera accessibilità, una corretta analisi della struttura della pagina, dei suoi contenuti, della navigabilità e usabilità. L'ideale sarebbe poter condurre questa analisi attraverso il coinvolgimento di diverse persone che possibilmente rispecchino il target di utenti cui ci si rivolge in modo da verificare l'effettiva accessibilità. L'usabilità deve essere verificata per tutte le modalità di interfacciamento previste: accesso visuale, vocale, tattile o misto.

2.5 Accessibilità multimodale in breve

Per riassumere, ecco quali ritengo essere i punti cardine per raggiungere il più alto grado di accessibilità multimodale:

- separare la struttura dalla presentazione del documento. Le informazioni devono poter essere presentate in svariati modi, anche molto diversi tra loro, ad esempio tramite sintesi vocale, rappresentazione visuale, tattile. La separazione di logica da contenuto risulta quindi essere fondamentale. In questo il linguaggio XML rappresenta senza dubbio una chiave di successo per raggiungere l'obiettivo.
- fornire, per quanto possibile, "equivalenti" testuali ad elementi multimediali: immagini, suoni ecc... I testi, sono la componente informativa più facilmente riproducibile in diversi modi: possono essere scritti, letti, riprodotti da una barra braille, tradotti in varie lingue e nel linguaggio dei segni ecc... La percezione delle immagini da parte dei non vedenti ² o dei suoni da parte dei non udenti è un'impresa a dir poco ardua. Per questo motivo è fondamentale riuscire ad usare bene dei testi alternativi. Se ciò non è possibile è molto meglio rinunciare a oggetti multimediali per veicolare informazioni.
- in generale, non affidarsi mai esclusivamente ad una sola modalità per veicolare informazioni. Ad esempio non utilizzare esclusivamente bottoni grafici o colori il cui significato è comprensibile solo per chi vede la pagina.
- utilizzare al meglio le informazioni che si hanno a disposizione dell'utente in modo da adattare il contenuto e la presentazione alle sue capacità. La profilazione dell'utente è un aspetto molto importante per migliorare l'usabilità dell'interfaccia che si viene a creare (visuale, uditiva, tattile o mista che sia) e nel contempo, l'accessibilità delle informazioni.

²Un modo è quello di trasformare le immagini in sensazioni tattili 3.8.2

- utilizzare un linguaggio adatto alla tipologia di utente cui ci si rivolge. Più semplice è il linguaggio che si utilizza, più ampio sarà lo spettro di utenti che saranno di comprendere le informazioni. Da un punto di vista più strettamente tecnico, inoltre, si può verificare facilmente che usare una sintassi semplice e una corretta punteggiatura rendere molto più semplice l'intelleggibilità e la stessa comprensione dei testi riprodotti dai sintetizzatori vocali.
- fornire all'utente semplici e chiari meccanismi per orientarsi e per navigare all'interno della pagina e del sito.
- utilizzare il box model del CSS per definire il layout di pagina e per consentire l'individuazione dei componenti logici della pagina.
- per consentire una agevole trasformazione della pagina e per sfruttare al meglio diversi dispositivi di interfacciamento usati dagli utenti, utilizzare dei documenti RDF da collegare alle singole pagine per descriverne la topologia e le relazioni con altre pagine del sito
- un sito per essere accessibile deve essere usabile dagli utenti cui è rivolto e lo deve essere indipendentemente dagli strumenti utilizzati per la navigazione. La forte relazione tra usabilità e accessibilità la ritengo un aspetto fondamentale che nelle WCAG 1.0 non è stata quasi presa in considerazione se non in modo marginale e che solo nella versione 2.0 ha assunto, almeno nei propositi del gruppo di lavoro, un peso più rilevante.
- verificare sempre e l'usabilità dei siti e l'accessibilità delle informazioni da parte degli utenti. Se possibile cercare di coinvolgere un gruppo di utenti che sia significativo per il tipo di sito realizzato e condurre delle valutazioni soggettive attraverso ad esempio prove di utilizzo, questionari, commenti degli utenti. L'uso di strumenti di valutazione automatica dell'accessibilità deve essere solo il primo passo, quello che consente di evitare banali errori tecnici di applicazione delle linee guida.

Capitolo 3

Un caso di studio: il progetto

MultiAbile

In questo capitolo viene presentato il progetto MultiAbile¹, realizzato in conformità con le linee guida descritte nel capitolo precedente e nel quale si è fatta un'ampia sperimentazione d'uso di molteplici dispositivi per verificare in modo diretto l'accessibilità e l'usabilità

Il sito web realizzato costituisce quindi un caso di studio concreto, utile alla valutazione delle linee guida di accessibilità multimodale individuate[61].

3.1 Le finalità e gli obiettivi

Lo scopo del progetto è la realizzazione di un servizio di apprendimento a distanza (e-learning) nonché di un sistema informativo accessibile e usabile destinato prevalentemente a persone disabili.

Le persone cui ci si rivolge sono ipovedenti, ciechi, sordo-ciechi, dislessici.

¹Multiabile fa parte del progetto nazionale Misura D4 “Dispositivo di Azioni di Sistema per il Miglioramento delle Risorse Umane nel Settore della Ricerca e Sviluppo Tecnologico” – FSE Obb. 3 anno 2003, approvato e finanziato il 13/07/2004 n. prot. 1372, il nome del progetto è “ICT Research For Disability” – n. 202618 (durata 14 mesi)

In una prima fase verranno coinvolte persone con competenze informatiche di base sufficienti ad utilizzare in autonomia il sistema.

Grazie anche alla realizzazione di altri canali di distribuzione, primo fra tutti quello telefonico, si pensa in seguito di rivolgere l'attenzione anche verso coloro che non sono interessate o sono impossibilitate ad utilizzare il mezzo informatico. Si pensa ad esempio agli anziani o a disabili senza conoscenze informatiche.

Gli obiettivi primari del progetto possono essere quindi sintetizzati nei seguenti punti:

- la sperimentazione di moduli formativi erogati a distanza tramite rete internet, studiati e realizzati per l'accesso da parte di utenti disabili
- la sperimentazione di tecnologie e interfacce che aiutino il disabile a fruire della formazione a distanza in internet e più in generale delle informazioni dei servizi reperibili in rete. In particolare si vuole sperimentare l'uso di interfacce vocali fruibili attraverso il telefono e, per quanto riguarda il web tradizionale, l'uso dei gesti e di feedback tattili e sonori come supporti alla navigazione.

Per quanto riguarda i contenuti da erogare sono state individuate tre diverse aree di interesse:

- formazione: realizzazione ed erogazione di corsi che vadano dall'alphabetizzazione informatica, alla storia e a corsi più complessi.
- informazione: accesso alle notizie ANSA, notizie di cultura, eventi della propria città ecc. . .
- servizi: accesso ai servizi di pubblica utilità quali spettacoli teatrali, cinema, trasporti, viabilità ecc. . .

Per poter raggiungere il target di utenza indicato è indispensabile l'applicazione e la verifica dei criteri di accessibilità e usabilità multimodale. Per tale verifica, l'obiettivo è quello di utilizzare quanti più strumenti di ausilio sia possibile adottare, eventualmente ideandone di nuovi, e sperimentare personalmente e con l'aiuto di potenziali utenti, il loro uso.

3.2 Struttura del sito

Il sito è diviso in una parte pubblica, accessibile a tutti, e una privata accessibile mediante autenticazione con login e password. Per ottenere l'accesso all'area privata è quindi necessario registrarsi lasciando anche solo le informazioni necessarie all'identificazione.

L'area pubblica è composta dalle seguenti pagine:

- la home page di presentazione del sito
- una pagina dei contatti
- una pagina di registrazione al sito
- una pagina di aiuto per la navigazione via web che illustra le principali modalità di navigazione nel sito e le scorciatoie mappate sulla tastiera.
- alcune pagine che illustrano più dettagliatamente le modalità di interfacciamento che si possono sperimentare sul sito:
 - navigazione attraverso il browser
 - navigazione attraverso il telefono
 - uso dei mouse gesture
 - uso dei feedback tattili e vibrazionali
- una pagina per scegliere le preferenze di visualizzazione e di accesso al sito.
- una pagina per l'accesso alle informazioni prelevate da altri siti attraverso syndication RSS².

L'area privata, accessibile a seguito dell'autenticazione, consiste essenzialmente di:

- una pagina contenente il catalogo corsi che dà accesso a tutti i moduli formativi presenti in piattaforma

²Al momento vengono proposte le notizie ANSA (<http://www.ansa.it>) organizzate per argomento

- una pagina per modificare i propri dati personali e con la quale memorizzare le proprie preferenze di navigazione in modo che vengano ripristinate ad ogni accesso.

Data la semplicità della struttura è stato sufficiente realizzare solo due documenti RDF “contents” secondo le specifiche definite nel capitolo 2, uno associato a tutte le pagine dell’area pubblica e uno per quelle dell’area privata.

3.3 Struttura delle pagine

Tutte le pagine del sito sono realizzate secondo il medesimo template. La figura 3.1 mostra una pagina in cui appare chiaro il layout che viene poi ripreso in tutto il sito.



Figura 3.1: Layout delle pagine del sito

La parte alta è costituita dal banner ed è priva di contenuto informa-

tivo di rilievo. Quest'area viene sfruttata, come si vedrà, nelle pagine di fruizione dei moduli.

Appena sotto è presente una barra orizzontale nella quale viene indicato all'utente dove si trova all'interno del sito.

Il lato sinistro ospita il menu che può variare in base al contesto. Esistono fondamentalmente 3 menu: il menu generale del sito che è presente in tutte le pagine fintantoché l'utente non si identifica; un menu che compare solo quando l'utente si è autenticato, un menu che compare solo durante la fruizione di un modulo formativo.

Il layout della pagina è definito attraverso i CSS (non sono mai state utilizzate tabelle per la disposizione degli elementi).

Nella barra superiore sono presenti dei link nascosti (ma letti dagli screen reader) con delle access key associate, per accedere al menu e al contenuto principale. Tali elementi sono i primi ad essere letti dagli screen reader durante la serializzazione della pagina, in tal modo si permette all'utente di capire come è strutturata e di scegliere subito cosa andare a leggere.

Sono presenti anche alcuni testi nascosti, utili per chi usa l'interfaccia vocale o lo screen reader, che annunciano l'inizio e la fine del menu, l'inizio e la fine del contenuto principale. Per quest'ultimo, il messaggio è personalizzato in base al contesto: ad esempio nelle pagine di fruizione del corso, oltre ad indicare la fine della pagina corrente, si suggerisce all'utente come fare per proseguire, quale combinazione di tasti premere per andare alla pagina o alla sezione del corso successiva e così via.

Essendo il template comune a tutte le pagine, è stato possibile creare un documento RDF "index" unico secondo le specifiche illustrate nel capitolo 2 per l'accesso multimodale.

I documenti "contents" e "index" sono usati dal transcoder per la creazione dell'interfaccia vocale come descritto nel capitolo 4.

3.4 Profilazione dell'utente

Date le diverse tipologie di utente e le diverse modalità di accesso consentite dal sistema si è deciso di consentire la memorizzazione delle preferenze in modo da adattare il più possibile la presentazione e contenuti.

La profilazione degli utenti non è fatta in base alle disabilità, che non abbiamo ritenuto possa essere il criterio principale per poter adattare nel modo migliore il sito.

Abbiamo preferito prendere l'ICF come strumento di riferimento da cui prendere spunto per classificare non tanto le disabilità degli utenti ma le capacità funzionali e i contesti di utilizzo del sistema.

Il risultato è una serie di preferenze generiche che possano andare incontro ad un più ampio spettro di utenti e che tengano conto delle diverse modalità di accesso.

Le preferenze vengono scelte dall'utente nella pagina "Scegli come navigare" e sono:

- Scelta della combinazione di colori. Possibilità di scelta fra 4 diverse combinazioni di colori che sono state studiate tenendo conto dei possibili problemi di ipovisione e daltonismo.
- Scelta della dimensione del carattere. Sono previste tre dimensioni: normale, media e grande. La dimensione dei caratteri è comunque sempre stabilita dai fogli di stile in modo relativo, così da consentire il ridimensionamento anche adottando le funzioni dei comuni browser (Firefox e Opera in particolare).
- Visualizzazione solo testo. Modalità che consente di eliminare tutte le immagini e altri parametri stilistici (eliminazione dei CSS). Utile ad esempio per chi utilizza uno screen reader.
- Uso della barra braille. Indica che l'utente utilizza un display braille per leggere il contenuto delle pagine.
- Stile di scrittura. L'utente può scegliere se preferisce un testo sem-

plificato rispetto a quello standard. La modalità semplificata è utile ad esempio alle persone dislessiche o che comunque possono avere qualche problema di comprensione della lingua. Oppure ancora per chi usa l'interfaccia vocale, nel qual caso la sintesi vocale risulta essere più chiara e comprensibile.

- Uso di un dispositivo tattile. Consente all'utente di ricevere stimoli tattili durante la navigazione delle pagine. Questa modalità è pensata soprattutto per i sordo-ciechi.
- Navigazione gestuale. Consente all'utente di navigare utilizzando semplici movimenti del mouse o di altro dispositivo di puntamento che riproducano dei gesti e a cui si fanno corrispondere dei comandi quali l'apertura delle pagine o l'accesso alle componenti principali (contenuto o menu)

Le preferenze di navigazione possono essere scelte dall'utente occasionale del sito e vengono memorizzate in caso di registrazione alla piattaforma in modo da poter essere ripristinate ad ogni nuovo collegamento.

3.5 L'interfaccia vocale

Per la realizzazione dell'interfaccia vocale sono stati tenuti in considerazione requisiti di usabilità che verranno ampiamente illustrati nel capitolo 4 dove viene descritta dettagliatamente l'implementazione del sistema di trasformazione automatica delle pagine da HTML a VoiceXML e altri formati.

Per alcune parti del sito, come quella relativa alla fruizione dei corsi, l'interfaccia vocale è stata creata ad hoc in modo da poter mantenere la compatibilità con lo standard SCORM che come si vedrà è stato adottato per la piattaforma e-learning. Il resto del sito è navigabile vocalmente mediante l'uso del transcoder.

Analogamente, i contenuti dell'area informativa sono fruibili attraver-

so una interfaccia vocale creata in modo misto: una parte è ottenuta direttamente trasformando i contenuti RSS prelevati dai siti esterni. Le pagine collegate ai singoli elementi vengono poi rese accessibili attraverso la trascodifica.

3.6 La piattaforma di e-learning

La piattaforma realizzata costituisce un cosiddetto Learning Content Management System (d'ora in poi indicato con la sigla LCMS), cioè sostanzialmente un sistema in grado di gestire, amministrare ed erogare corsi di formazione a distanza³. La fruizione dei corsi, solitamente effettuata attraverso il PC utilizzando il browser, può essere fatta, su questa piattaforma, anche attraverso il telefono grazie alla realizzazione di una opportuna interfaccia vocale.

La figura 3.2 illustra schematicamente la struttura dell'LCMS nel quale sono individuate due possibili tipologie di utenti che possono operare sul sistema: gli studenti (utenti ordinari) e gli amministratori.

Runtime environment : l'ambiente di erogazione dei contenuti formativi. Il sistema mette a disposizione del client una API che consente in tempo reale l'interazione con l'LMS e quindi un tracciamento dell'attività dell'utente durante la fruizione dei corsi. Ad esempio, ogni volta che sia accede ad un learning object⁴, il browser (o comunque il sistema usato dall'utente) può comunicare all'LMS l'evento, il quale viene memorizzato dal server assieme al timestamp che indica l'istante in cui l'evento si è verificato. Oppure, il browser può inviare all'LMS la richiesta di passaggio al learning object successi-

³Di seguito si farà riferimento separatamente alle componenti principali con le sigle LMS (Learning Management System) e CMS (Content Management System)

⁴Dal punto di vista dell'LMS, i learning object costituiscono le unità atomiche di cui è composto un modulo formativo. Pertanto può essere costituito da una o più pagine di contenuti relativi ad uno specifico argomento, il quale viene introdotto e si esaurisce nell'ambito dello stesso learning object.

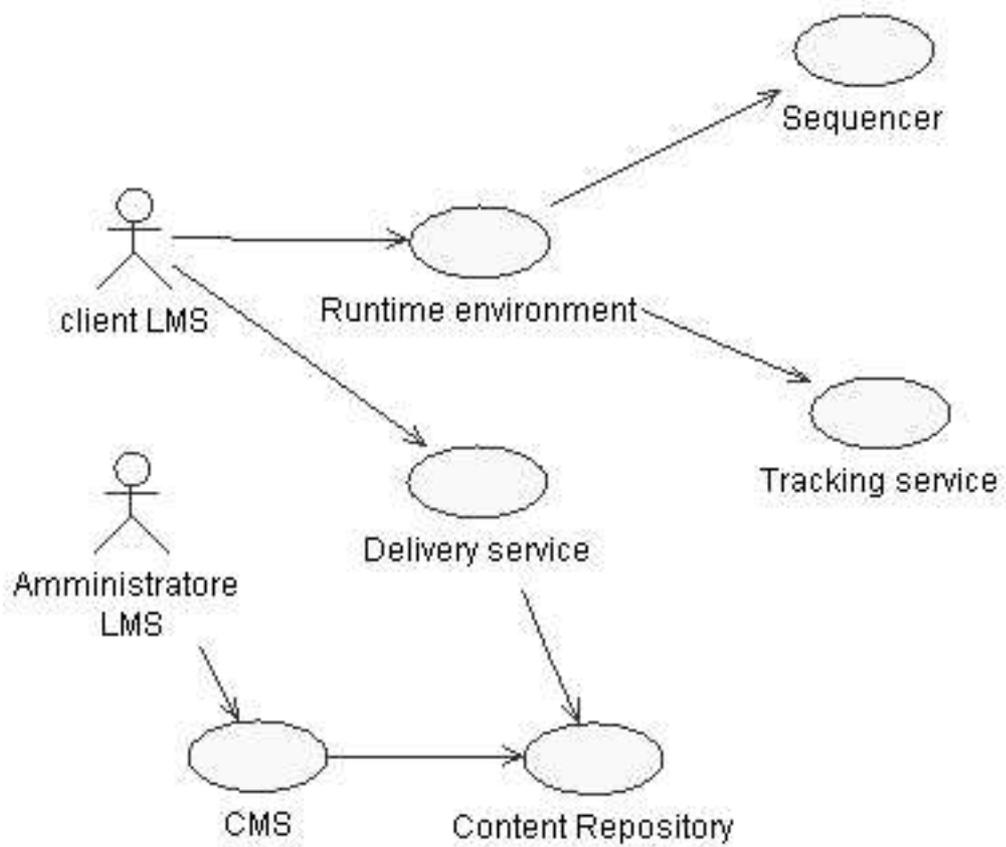


Figura 3.2: Struttura dell'LCMS (Learning Content Management System)

vo. La comunicazione può avvenire in entrambe le direzioni e quindi il client può richiedere all'LMS alcune informazioni che sono state memorizzate per l'utente corrente e che devono essere ripresentate durante la fruizione del corso.

Delivery service : il componente che si occupa dell'erogazione effettiva dei contenuti. I contenuti vengono estratti dal repository gestito dal Content Management System ed inviati attraverso l'application server, dopo essere stati adattati alla modalità di interazione richiesta dall'utente.

Tracking service : il componente che si occupa di gestire la persistenza dei dati comunicati dal client all'LMS. Le informazioni principali che vengono tracciate sono: l'apertura e chiusura dei learning object, il loro lo stato di fruizione da parte dell'utente (iniziato, completato, abbandonato, ecc. . .), i tempi di permanenza dell'utente sul learning object o sul modulo complessivo, ecc. . .

Sequencer : questo componente si occupa di gestire la sequenza di erogazione dei learning object di un modulo. In base alla richiesta dell'utente e allo stato attuale indica quale learning object deve essere erogato. Ad esempio l'utente può chiedere di avanzare al learning object successivo a quello attualmente aperto; il sequencer identifica allora l'LO da erogare verificando nel contempo che l'utente abbia i prerequisiti richiesti: si potrebbe ad esempio desiderare che per erogare il learning object successivo sia necessario aver completato quello precedente.

Content Management System : il gestore dei contenuti veri e propri. Fornisce strumenti amministrativi, l'inserimento dei moduli formativi nel sistema e la creazione dei percorsi di studio. I moduli formativi sono memorizzati all'interno di un repository e resi disponibili al delivery service.

Content repository : il sistema di memorizzazione dei contenuti ammi-

nistrato dal CMS e utilizzato dal delivery service. In questa implementazione il content repository è costituito semplicemente dall'insieme dei moduli formativi costituiti da documenti XML, immagini, animazioni Flash, memorizzati su file system e indicizzati da un database.

3.6.1 Lo standard SCORM

Nella realizzazione dell'LCMS è stato adottato lo standard SCORM (Shareable Content Object Reference Model)[1], un insieme di specifiche pubblicate all'inizio del 2000, che stabiliscono le direttive per lo sviluppo, il confezionamento e la distribuzione di materiali ad alto contenuto formativo e soprattutto indicano i requisiti che deve avere il sistema di management e di erogazione dei contenuti in modo da garantire la possibilità di distribuire i contenuti tra piattaforme diverse.

Lo standard è frutto dell'unione di diverse specifiche proprietarie realizzate negli anni da alcune software house e da organizzazioni internazionali quali IEEE, IMS, AICC e ARIADNE.

Al momento l'ultima versione dello standard è la SCORM 2004⁵, la piattaforma realizzata per Multiabile è invece basata fondamentalmente sulla versione 1.2

Lo standard è costituito da due sezioni principali:

Content Aggregation Model : contiene le specifiche per la definizione di un formato per l'aggregazione di risorse quali documenti, immagini, video ecc. . . in contenuti di e-learning ben strutturati e che seguano principi di riusabilità, accessibilità, interoperabilità, durabilità.

Il documento è composto dalle tre seguenti sottosezioni:

- Meta-data dictionary (da IEEE)
- Content packaging (da IMS ed alcune estensioni)

⁵Si tratta della versione successiva alla 1.2, che dopo le prime stesure indicate con il numero di versione 1.3 è uscita appunto con la denominazione 2004

- Content aggregation components

Questa sezione è il punto di riferimento per la redazione dei contenuti e la costruzione dei moduli formativi.

Run-time environment : contiene specifiche che definiscono un ambiente (solitamente basato su una applicazione web) che permetta l'esecuzione, la comunicazione con l'LMS ed il tracking degli eventi e dei dati. Queste specifiche sono derivate tenendo conto delle funzionalità del runtime environment specificato nelle CMI001 guidelines di AICC. Il documento è composto dalle seguenti sottosezioni:

- Data Model (da AICC)
- Run-time API (da AICC)

Questa sezione è il punto di riferimento per la realizzazione dell'LMS.

I contenuti dei corsi sono stati realizzati interamente in XML, basato su un DTD definito all'interno di Web Learn dal gruppo di sviluppo dei moduli formativi e arricchito in seguito in modo da perseguire due diversi obiettivi:

1. possibilità di veicolare i corsi attraverso un'interfaccia vocale. Ad esempio sono stati inseriti dei tag per specificare la pronuncia delle parole. O ancora, molti tag hanno l'attributo `lang` per consentire di specificare la lingua del testo contenuto in modo che il sintetizzatore vocale possa pronunciare correttamente le parole in lingua straniera.
2. tenere in considerazione alcuni criteri di accessibilità in vista dell'integrazione dei corsi in una interfaccia web che rispetti tali principi. Ad esempio si obbliga il redattore dei testi ad indicare contenuti alternativi per gli oggetti multimediale. Oppure sono stati inseriti dei tag per la definizione di tabelle di dati, che non contengano alcun elemento legato alla presentazione, in modo da poter essere più facilmente adattabili a qualsiasi tipo di interfaccia.

La scelta dell'XML consente di distribuire in modo più efficiente e versatile i contenuti attraverso diversi canali e quindi attraverso interfacce utenti dotate di diverse modalità di interazione.

Nei prossimi paragrafi vengono illustrate brevemente le due sezioni delle specifiche SCORM introdotte precedentemente.

Content Aggregation Model Le specifiche descritte da questo documento permettono di creare dei contenuti ottenuti mediante l'aggregazione di risorse. È necessario definire le entità principali che fanno riferimento alla modalità di aggregazione delle risorse:

Asset : l'elemento elementare nel dominio dell'e-learning. Può essere visto come la rappresentazione elettronica di un testo, un'immagine, un suono, una pagina Web ecc... Ciascun asset deve poter essere considerato come una unità a se stante in modo da poter essere riutilizzato e aggregato assieme ad altri asset per costituire una unità didattica elementare (learning object).

Sharable Content Object (SCO) : è quello che nella letteratura dell'e-learning viene chiamato learning object ed è costituito da un insieme di asset. Questo elemento è fondamentale in quanto costituisce il più basso livello di granularità che una risorsa di e-learning, interagente con l'LMS, può avere. Ciascuno SCO deve essere una unità a se stante, indipendente dal contesto, in modo che possa essere estratta e riutilizzata all'interno di un nuovo corso. Per questo motivo una regola fondamentale di cui si deve tener conto nello sviluppo di moduli formativi secondo questo standard è che un learning object non può fare riferimento (ad esempio attraverso un link interno) ad un altro learning object (SCO). Solo il runtime environment dell'LMS ha la possibilità di erogare uno SCO, su richiesta del client. Lo SCO d'altra parte è il solo responsabile della navigazione attraverso gli asset di cui è costituito: non può essere il runtime dell'LMS a regolare tale navigazione.

Content aggregation : è una collezione di assets e di SCO aggregati tra loro al fine di costituire una risorsa di e-learning di alto livello che può essere erogata dall'LMS. Nella piattaforma realizzata questi oggetti sono i moduli formativi.

L'operazione di aggregazione viene formalizzata attraverso:

- Il content packaging, che consiste nella stesura di un file XML (il file `imsmanifest.xml`) che contiene la struttura e l'organizzazione di SCO e asset e quindi di fatto l'indice del corso.
- I meta-data, che costituiscono un insieme informazioni atte a semplificare le operazioni di archiviazione e ricerca delle risorse che formano i contenuti di un corso (asset e SCO). I meta-data possono essere inseriti in parte all'interno del manifest stesso oppure all'interno di file XML esterni per i quali il manifest contiene semplicemente un riferimento. In ogni caso i meta-data sono informazioni facoltative.

Runtime environment Al fine di poter garantire il riutilizzo delle risorse e l'interoperabilità fra diversi LMS è necessario definire un unico ambiente di esecuzione che fornisca una interfaccia comune per poter permettere l'erogazione delle risorse e la comunicazione di quest'ultime con l'LMS.

Lo standard SCORM prevede che un corso realizzato secondo le specifiche del content aggregation model possa essere erogato su qualsiasi piattaforma che implementi il runtime environment.

Il runtime environment deve quindi fornire una API standard con la quale ogni singolo SCO possa interfacciarsi al sistema LMS che risiede sul server. Per realizzare ciò, nel runtime sono presenti tre tipi di entità:

- un meccanismo di "lancio" di uno SCO
- una interfaccia di comunicazione con lo SCO (la API del runtime)
- un modello dei dati (data model) per l'interscambio di dati tra SCO e LMS.

La figura 3.3 mostra come avviene l'interazione tra SCO ed LMS.

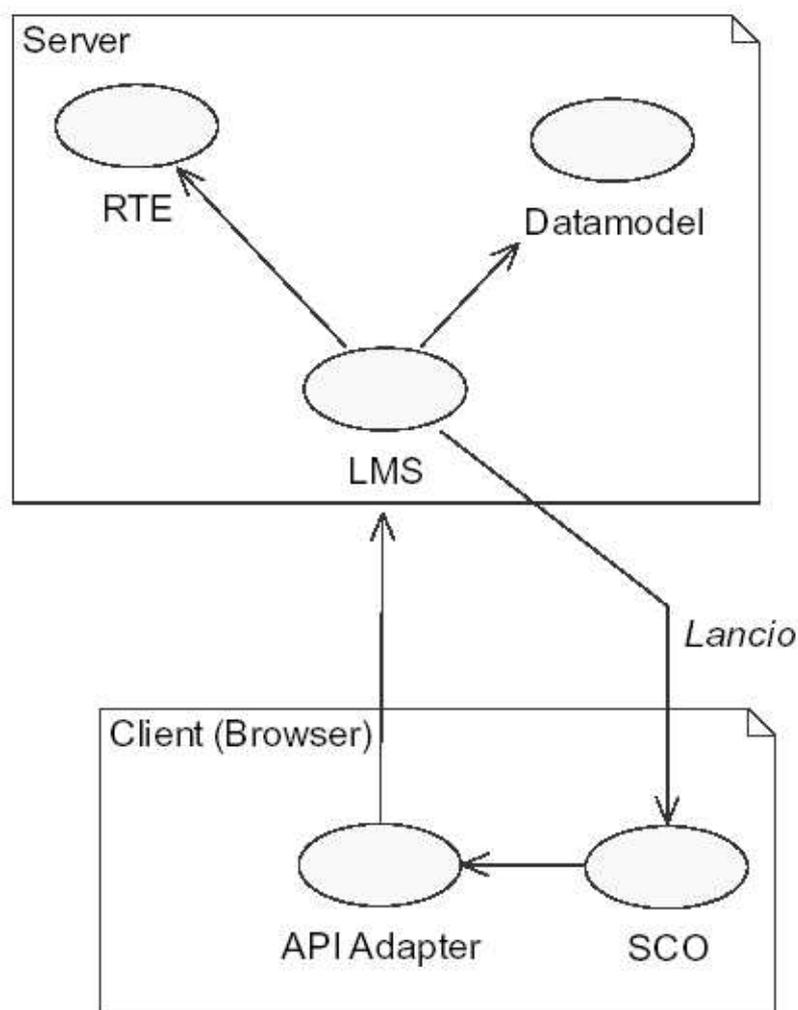


Figura 3.3: Comunicazione tra SCO e LMS secondo lo standard SCORM

Sul lato client viene realizzata una interfaccia per accedere alla API fornita dall'LMS, che può essere utilizzata dallo SCO per effettuare la comunicazione client-server. La API è esportata dal server attraverso un web service che consente chiamate remote attraverso la tecnologia SOAP.

Sul lato client, per quanto riguarda l'interfaccia grafica l'implemen-

tazione fatta è in Javascript mentre per quella vocale viene utilizzata una servlet che fa da proxy per le chiamate al runtime environment.

L'operazione di lancio consiste nell'erogazione da parte dell'LMS e specificatamente dell'application server, delle risorse di cui è costituito uno SCO. Lo SCO può effettuare richieste e ottenere informazioni dall'LMS utilizzando la API messa a disposizione.

La API è molto semplice in quanto fornisce agli SCO un insieme volutamente limitato di funzioni che consentono di ottenere la completa gestione dello stato dello SCO e delle situazioni di errore oltre che la trasmissione di informazioni da SCO a LMS e viceversa.

Il datamodel Lo scopo di avere un modello di dati comune è quello di permettere agli SCO di interagire con diversi LMS. Se, per esempio, fosse necessario registrare le prestazioni di uno studente sarebbe indispensabile un sistema che permetta di informare l'LMS dei risultati ottenuti durante la fruizione dello SCO e la modalità di interazione e i dati passati, devono poter essere interpretati correttamente da qualsiasi LMS che rispetti lo standard.

Quando si realizza un corso basandosi sullo standard SCORM non è necessario conoscere l'implementazione LMS sul quale dovrà essere erogato.

Il data model di SCORM è derivato direttamente da AICC CMI Data Model descritto in AICC CMI Guidelines for Interoperability[9].

Lo standard definisce 49 elementi organizzati in 6 categorie distinte. Non tutti questi elementi sono obbligatori (lo sono solo 14 di essi). Per ciascun elemento viene indicato lo scopo per il quale è stato definito ed il tipo di dato che può essere memorizzato, che può essere una stringa di testo, un numero o un elemento di vocabolario anch'esso specificato nello standard.

A seconda dell'elemento, lo SCO e il runtime dell'LMS possono avere diritti di sola lettura, di sola scrittura o di lettura e scrittura.

3.6.2 Fruizione dei corsi

L'accesso ai corsi è riservato agli utenti registrati. Una volta identificati dal sistema l'utente può seguire qualunque corso presente nel catalogo, secondo la modalità di accesso preferita.

La figura 3.4 illustra i componenti che entrano in gioco in questa fase, indipendentemente dal tipo di interfaccia utilizzata e dalle modalità scelte dall'utente. La prima operazione è quella dell'apertura del corso. L'LMS attiva il "sequencer" che funziona un po' da segnalibro elettronico, in quanto indica qual'è l'ultima pagina visitata dall'utente in modo da consentire di riprendere la lettura da dove era stata interrotta e gestisce l'ordine di apertura dei learning object e la struttura del corso.

Il Runtime Environment (RTE) si occupa dell'accesso al repository dei corsi per prelevare i contenuti adattati al tipo di presentazione richiesta dall'utente, e della memorizzazione degli eventi (apertura/chiusura corso, ultima pagina visitata, tempo di permanenza ecc. . .)

All'interno dei moduli possono essere inseriti dei learning object speciali, i questionari, che sono trattati in modo particolare. Si tratta infatti di una sequenza di domande a risposta chiusa: l'utente deve scegliere quella giusta tra le alternative proposte. I questionari vengono ricostruiti di nuovo ad ogni accesso a partire da un repository di domande: vengono estratte un certo numero di domande le quali vengono riordinate in modo casuale così come le relative risposte (shuffle). Durante l'erogazione del questionario vengono memorizzate le risposte e infine il risultato finale ottenuto.

I questionari sono anch'essi redatti in XML secondo un DTD particolare, diverso da quello degli LO ordinari.

L'interfaccia grafica

L'interfaccia grafica presentata durante la fruizione di un corso è estremamente ricca di elementi di navigazione. Oltre alle normali chiavi di

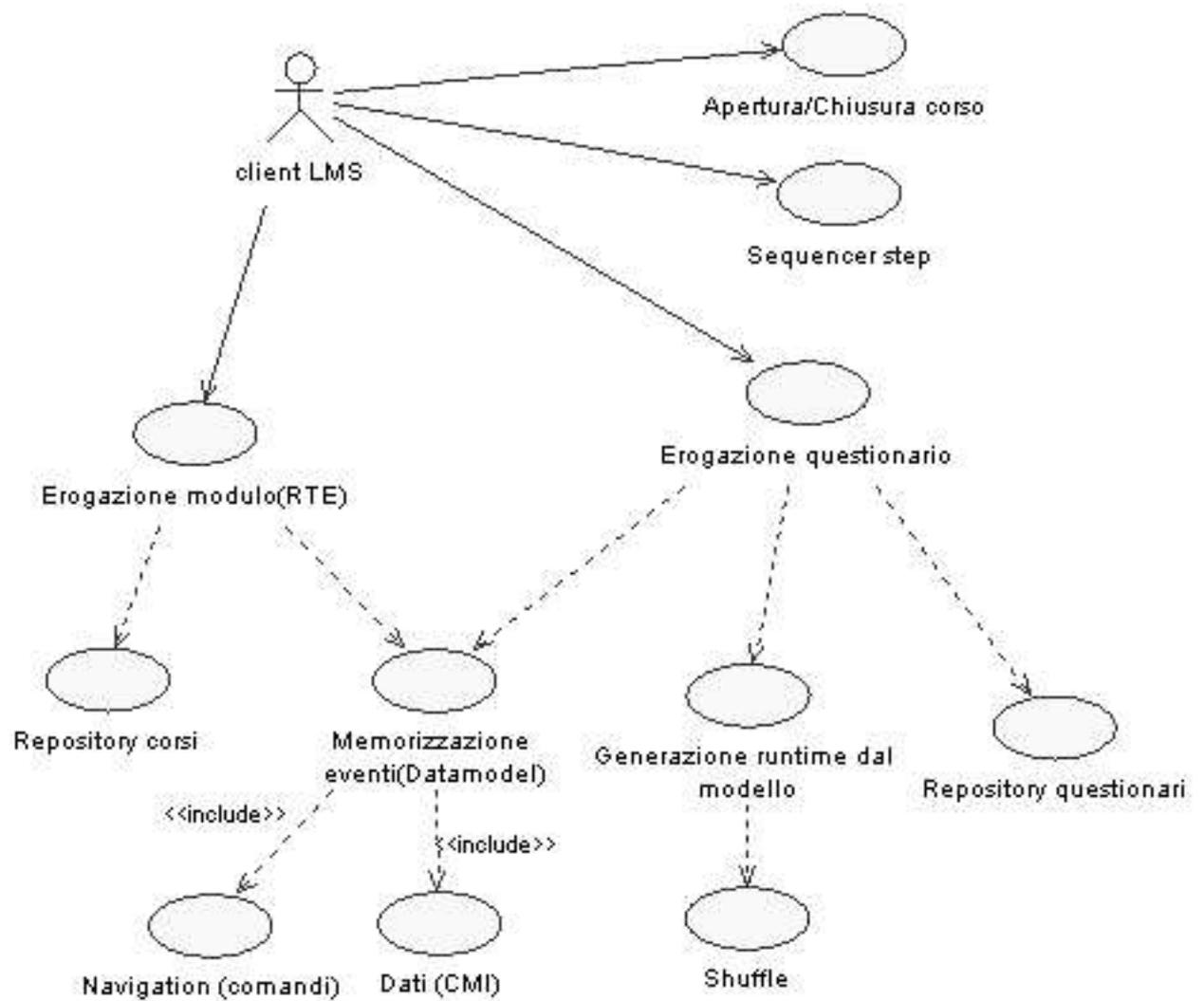


Figura 3.4: Fruizione del corso

accesso per il menu e il contenuto principale della pagina, ne sono state definite una per ciascun “pulsante” di navigazione all’interno del corso.

La navigazione avviene a due livelli: a livello di corso e a livello di learning object. Per la navigazione nel corso sono presenti i seguenti pulsanti:

- apertura dell’indice del corso: consente di visualizzare la struttura del corso e in particolare l’elenco dei learning object. Presenta un menu con il quale è possibile saltare da un LO all’altro.
- learning object precedente: chiede all’LMS l’apertura del LO precedente a quello attuale
- learning object successivo: chiede all’LMS l’apertura del LO successivo a quello attuale
- chiusura del corso: chiude il corso e chiede all’LMS la memorizzazione dello stato corrente

Le informazioni relative alla struttura del corso e alla sequenziazione dei learning object sono prese dal file `imsmanifest.xml` creato in fase di costruzione del modulo formativo.

Per quanto riguarda la navigazione all’interno del singolo learning object, viene messo a disposizione un menu con l’elenco delle pagine di cui è composto e alcuni pulsanti per accedere a: prima e ultima pagina, pagina precedente e pagina successiva.

Viene poi creato poi in automatico un indice di pagina per accedere rapidamente ai paragrafi di cui è composta, molto comodo soprattutto per pagine lunghe.

La figura 3.5 mostra una pagina di un corso. Nella parte alta dello schermo sono presenti i pulsanti di navigazione tra i learning object. Nella parte sinistra compaiono, nell’ordine: la lista dei link alle pagine di cui è composto il learning object corrente, i quattro pulsanti di navigazione tra le pagine del learning object e l’elenco dei link interni alla pagina relativi ai paragrafi.

The screenshot shows a web interface for a course. At the top left is the 'multiAbile' logo. To its right is a navigation bar with five icons: an upward arrow, a left arrow, a right arrow, a question mark, and a person icon. Below this is a yellow banner with the text 'Benvenuto Ferdinando. Ora sei in **Chiusura corso** >> Sezione'. The main content area is divided into two columns. The left column contains a table of contents under the heading 'La scrittura in Braille. [m]', listing 'Elenco delle pagine' with three items: '[1] La scrittura in Braille', 'LA DATTELOBRILLE', and '[3] LE STAMPANTI BRAILLE'. Below the list are four navigation arrows. The right column features the section 'LA DATTELOBRILLE' with the page number 'Pagina 2 di 3'. It includes two images: a vintage Braille typewriter and a modern Braille printer. Below the images is a paragraph of text explaining that a Braille typewriter has only six keys corresponding to the six Braille dots, one for space, one for backspace, and one for carriage return.

Figura 3.5: Struttura della pagina del corso

L'interfaccia vocale

L'architettura illustrata in precedenza è indipendente dall'interfaccia utilizzata dall'utente. Essendo i contenuti dei corsi memorizzati in file XML, è possibile adattarne la presentazione in base alla modalità scelta.

La trasformazione dei documenti XML viene fatta attraverso fogli di trasformazione XSLT al momento della richiesta dell'utente. In tal modo, una modifica fatta ai contenuti si riflette automaticamente su tutte le possibili presentazioni tra cui appunto quella vocale.

L'utente accede all'LMS il quale richiede al CMS le pagine XML direttamente nel formato richiesto per l'interfaccia vocale.

Il sito stesso è interamente navigabile vocalmente attraverso l'uso del telefono, grazie alla trasformazione dei contenuti in VoiceXML 1.0

Requisito fondamentale dell'interfaccia vocale è quello di fornire per quanto possibile le stesse funzionalità date da quella grafica e in particolare:

- la navigabilità all'interno dei corsi
- la segmentazione delle singole pagine in paragrafi più piccoli per consentire una più facile fruizione
- la compatibilità con lo standard SCORM per mantenere il tracciamento degli eventi

L'ultimo punto è stato abbastanza complesso da realizzare dato che il linguaggio ECMAScript usabile all'interno dei documenti VoiceXML, pur essendo molto simile a Javascript non consente di effettuare chiamate SOAP necessarie per l'invocazione remota delle funzioni esportate dall'LMS.

Per ovviare a questo problema ho implementato una servlet con funzione di proxy verso le chiamate all'RTE della piattaforma e-learning. All'interno dei documenti VoiceXML, le chiamate SOAP al server sono sostituite da normali richieste HTTP con le quali vengono inviati tutti i parametri necessari ad effettuare le invocazioni dei metodi esportati. La

servlet si occupa di ritrasmettere le richieste all'RTE e di restituire l'esito al voice browser.

Identificazione dell'utente L'autenticazione è necessaria per consentire il corretto tracciamento dell'attività dell'utente riguardante la fruizione dei moduli formativi e per l'adattamento dell'interfaccia alle preferenze indicate.

Per semplificare il più possibile questa operazione, si è deciso di identificare l'utente in base al numero di telefono utilizzato per la connessione al sistema.

Il linguaggio VoiceXML consente di avere proprio questo tipo di informazione se il documento è fruito attraverso un voice gateway adeguato (come ad esempio Loquendo). In fase di registrazione l'utente sceglie il numero di telefono dal quale intende collegarsi in modo che possa essere automaticamente identificato.

Nel caso in cui il numero non venga indicato oppure il sistema non sia in grado di riconoscerlo (chiamata effettuata da un numero nascosto), viene effettuata una autenticazione classica attraverso l'immissione di username e password. Il sistema impone che questi due valori, scelti durante la registrazione, siano entrambe di tipo numerico in modo da poter essere immessi attraverso il tastierino del telefono e semplificare il più possibile il riconoscimento.

3.7 Accesso alle informazioni con syndication

Una delle applicazioni pratiche relative all'interfaccia vocale, ritenute più interessanti in questo progetto, è quella di rendere disponibili attraverso il telefono, notizie, informazioni di pubblica utilità ecc. . . distribuite nel web attraverso lo standard RSS[6].

RSS è un formato per la distribuzione di news e altri contenuti strutturati in modo simile, costituiti fondamentalmente da: un titolo, un somma-

rio e un link al sito di provenienza e precisamente alla pagina con notizia completa. Si tratta di una applicazione di XML⁶, pertanto risulta essere molto versatile e si presta alla trasformazione per la presentazione in diversi formati.

I documenti RSS sono aggiornati e resi disponibili per il download direttamente attraverso i siti internet di provenienza. Per Multiabile, al momento, sono stati scelti i canali informativi messi a disposizione da ANSA.it che sono i seguenti: Top News, Italia, Mondo, Calcio, Sport, Spettacolo, Economia e Finanza, Cultura e Società, Scienze, Internet e Moda.

Un menu vocale consente all'utente di scegliere la categoria di notizie che desidera ascoltare, dopodiché il sistema scarica localmente una copia del documento RSS dal sito e lo trasforma in un documento VoiceXML.

Solo una parte dell'informazione è contenuta all'interno degli RSS, la motivazione è ovviamente quella che chi distribuisce i contenuti con questa tecnica cerca di far conoscere il proprio sito e di attrarre nuovi visitatori.

Per poter accedere alla notizia completa è quindi necessario accedere al sito andando all'indirizzo specificato nel documento RSS per ciascuna notizia. Per far ciò attraverso l'interfaccia vocale, viene utilizzato il transcoder descritto nel capitolo 4. L'utente, una volta ascoltato il titolo ed il sommario di una notizia cui è interessato, può indicare al sistema il desiderio di seguire il link. La pagina relativa viene caricata attraverso il proxy che trasforma la pagina in VoiceXML e consente pertanto la fruizione della pagina completa.

⁶È di fatto una versione semplificata di RDF

3.8 L'esperienza multimodale

Il tentativo di ottenere una elevata accessibilità delle informazioni da parte degli utenti disabili ha portato alla sperimentazione di nuove modalità di accesso e al tentativo di costruire interfacce che consentano all'utente di impartire comandi e di percepire i contenuti attraverso diversi canali che facilitassero questo compito.

L'uso di diversi dispositivi di input e di output hanno consentito di verificare sul campo la robustezza delle linee guida sull'accessibilità e dei principi di costruzione delle pagine del sito.

La sperimentazione e la ricerca di nuove modalità di interazione e la creazione di interfacce multimodali, costituiscono un elemento di forte innovatività e di interesse per questo progetto.

3.8.1 Utilizzo dei gesti

Oltre alle normali scorciatoie definite attraverso le access key, è stato sperimentato l'uso dei mouse gesture. Attraverso il mouse o qualunque altro dispositivo di puntamento⁷, si possono "tracciare" dei segni che vengono interpretati dal computer e vengono associati a particolari comandi di navigazione.

La sperimentazione è stata fatta utilizzando il software StrokeIt⁸, che va installato sul PC dell'utente e deve essere attivo durante la navigazione. È stato inoltre realizzato un "file di gesti", scaricabile dal sito e utilizzabile da StrokeIt per riconoscere alcuni movimenti.

Per attivare un gesto è necessario compierlo tenendo premuto il tasto destro del mouse e rilasciarlo al termine dell'azione. I movimenti memorizzati in StrokeIt sono associati a dei particolari comandi che solitamente corrispondono alla pressione di una certa combinazione di tasti sulla ta-

⁷Penna elettronica, touch screen, tavoletta grafica ecc. . .

⁸<http://www.tcbmi.com/strokeit/>

stiera. In tal modo, quando un gesto viene riconosciuto, StrokeIt simula il comando corrispondente.

StrokeIt è in grado di apprendere i movimenti fatti dall'utente: nel caso in cui un gesto non venga riconosciuto, appare una finestra di avviso nella quale è possibile indicare al software quale gesto è stato fatto associandolo a uno di quelli memorizzati in modo che in seguito possa essere riconosciuto correttamente. In questo modo StrokeIt adatta il proprio sistema di riconoscimento all'utente.

Attraverso questo meccanismo sono stati creati dei gesti che simulano la pressione delle accesskey definite all'interno della pagina. Ad esempio è stato definito un segno simile ad una "C" associato alla accesskey che porta il cursore del browser al contenuto principale della pagina. I gesti corrispondenti ai numeri da 1 a 9 sono fatti corrispondere alle voci di menu e così via.

L'obiettivo è quello di trovare una serie di gesti comodi che possano poi essere definiti come standard per l'attivazione di determinati comandi. Una volta fatto ciò, tali gesti potranno essere formalizzati attraverso linguaggi quali InkML o EMMA e inseriti all'interno dei documenti di descrizione semantica delle pagine definiti nel capitolo 2.

L'ideale sarebbe quella di poter integrare in tutti i browser questa funzione di riconoscimento dei gesti (come avviene ad esempio nel browser Mozilla) e fare in modo che la definizione delle scorciatoie di accesso siano definite lato server da parte degli sviluppatori.

Nella definizione di questa modalità di accesso nasce il problema di come comunicare all'utente la loro esistenza e come indicare l'associazione tra movimento e azione corrispondente. Ancora una volta, l'uso di un linguaggio formale basato su XML può risolvere il problema per un browser predisposto. In caso contrario, come nel caso del sito MultiAbile, è necessaria una pagina di istruzioni che descriva il funzionamento e i comandi previsti.

3.8.2 Feedback tattili: la percezione spaziale

La navigazione da parte di un sordo-cieco non può essere fatta né attraverso l'uso dell'interfaccia grafica, né attraverso quella vocale. Il tatto risulta quindi essere l'unico mezzo per la comunicazione. Solitamente il mezzo principale che si usa è quello del display braille, che collegato al computer, riproduce il testo presentato dal browser.

Di per sè, tale strumento sarebbe sufficiente alla fruizione di tutte le informazioni testuali una volta che il sito sia realizzato secondo principi di accessibilità. Tuttavia con questo sistema è possibile fruire solo di contenuti serializzabili e non di risorse che per loro natura sono bidimensionali, come le immagini o il layout di una pagina con gli ingombri di ciascun elemento.

Nello sviluppo del progetto abbiamo voluto sperimentare l'utilizzo di altri sistemi che potessero dare sensazioni tattili che avessero qualche utilità per la navigazione o per la percezione delle informazioni.

I principali strumenti utilizzati per la sperimentazione sono stati il mouse Logitech iFeel⁹ e il joystick Microsoft Sidewinder Force Feedback¹⁰. Entrambe questi dispositivi possono essere controllati attraverso opportuni plug-in del browser Internet Explorer¹¹, comandati attraverso istruzioni javascript.

Il mouse può produrre vibrazioni delle quali può essere controllata l'intensità, la durata e la forma (impulsiva, periodica, mista). Il joystick, oltre a questo, può essere anche controllato nei movimenti e quindi può essere utilizzato per "guidare" i movimenti della mano dell'utente che lo utilizza.

Prendendo spunto da queste potenzialità sono state effettuate alcune prove di utilizzo ¹².

⁹<http://www.logitech.com>

¹⁰<http://www.microsoft.com/hardware/sidewinder/default.msp>

¹¹Il plug-in e il software Immersione Desktop sono reperibili all'indirizzo <http://www.immersion.com>

¹²Il metodo di navigazione e gli strumenti usati sono proposti come brevetto [58]

Topologia della pagina Il primo impiego dei feedback attraverso il mouse è stato quello di associare differenti output vibrazionali alle diverse aree topologiche in cui sono divise le pagine del sito.

Una applicazione analoga è stata sperimentata da Offen e Thomlinson per OpenBOOK della Freedom Scientific [22], un software per il riconoscimento e la sintesi vocale di testi ottenuti mediante scansione di una pagina stampata (OCR) in cui i feedback vibrazionali sono stati affiancati a quelli audio.

I blocchi della pagina sono così codificati da particolari vibrazioni avvertibili muovendo il puntatore sull'area occupata dal contenuto principale, dal menu, dal banner superiore e dal footer.

Ulteriori codici vibrazionali sono poi automaticamente forniti dal software del mouse, che vibra con una certa modalità ogni qualvolta si passa sopra un testo o un collegamento.

Ci si è resi subito conto, anche a seguito di riscontri avuti dai non vedenti che hanno sperimentato il sistema, che in effetti l'informazione data dalle vibrazioni può essere molto utile per la percezione spaziale, ma che di fatto difficilmente può dare un supporto alla navigazione, dal momento in cui la lettura delle pagine del sito può essere ottenuta semplicemente attraverso la serializzazione dei contenuti attraverso lo screen reader o la barra braille.

Mappa tattile interattiva In seguito all'esperienza dell'uso delle vibrazioni per la percezione della topologia della pagina abbiamo concluso che un campo più efficace di applicazione potrebbe essere quello delle immagini ed in particolare delle mappe.

Ancora una volta ci si è resi conto che quello tattile vibrazionale non può essere l'unico veicolo di informazione ma può essere un valido sistema di accesso a informazioni da inviare alla barra braille (per i sordociechi) o attraverso audio preregistrati o sintetizzati in tempo reale.

Il primo esperimento è stato fatto con la cartina geografica della Lom-

bardia. Sono stati codificati diversi pattern vibrazionali per i seguenti elementi: pianure, colline, montagne, fiumi, laghi, città e per identificare la parte esterna della mappa. Per le città principali e i laghi sono stati preregistrati i nomi, che vengono letti ogni volta che si passa con il puntatore sull'area da essi occupata.

Le impressioni avute dopo l'utilizzo da parte di una persona cieca (le stesse indicazioni sono state ottenute anche semplicemente provando ad "usare" la mappa ad occhi chiusi) sono state le seguenti:

- l'utilizzo del mouse non consente di rendersi conto degli spostamenti effettuati dal puntatore a seguito dei movimenti della mano. Questo fa sì che, pur conoscendo perfettamente la collocazione dei singoli elementi della mappa, difficilmente si riesce a raggiungere una determinata posizione seguendo il percorso più breve, ma più che altro si va per tentativi. L'accelerazione del puntatore del mouse confonde maggiormente e pertanto andrebbe eliminata.
- Anche eliminando l'accelerazione dal puntatore e migliorando la propria capacità di controllare i movimenti del mouse, è molto facile perdere l'orientamento e diventa impossibile determinare la posizione assoluta del puntatore sullo schermo. È necessario a quel punto compiere ampi movimenti fino a ritrovare un qualche feedback vibrazionale o sonoro per ritrovare la mappa e una zona conosciuta.
- Le aree sensibili della mappa andrebbero ampliate anche oltre il naturale confine in modo da semplificare l'individuazione degli elementi (città, laghi, fiumi. . .)
- L'esplorazione è del tutto casuale e basata sull'eventuale conoscenza pregressa dell'utente rispetto alla mappa. Manca un sistema che consenta di effettuare un percorso guidato.
- Alla fine dell'esplorazione della mappa, l'utente che si è mosso senza un obiettivo ben preciso si chiede se è riuscito a carpire tutte le informazioni disponibili. La serializzazione delle informazioni cui è

abituato un non vedente, seppur scomoda in alcune circostanze, garantisce tuttavia che tutto ciò che può essere letto venga raggiunto. È necessario studiare un sistema per eliminare questa incertezza.

Il mouse non sembra essere quindi il dispositivo più adatto all'esplorazione dello spazio di un'immagine. Il passo successivo è quello di utilizzare una tavoletta grafica e una semplice "cella vibrazionale". L'uso della tavoletta grafica consente di utilizzare il posizionamento assoluto del puntatore. I confini della tavoletta possono poi essere fatti corrispondere al riquadro contenente l'immagine e quindi consentire una migliore percezione dello spazio percorso ad ogni movimento.

Per un risultato migliore, la mappa da esplorare dovrebbe essere posta a tutto schermo, in modo da eliminare il più possibile gli elementi di disturbo.

Abbiamo così utilizzato la tavoletta grafica per spostare il puntatore del mouse e il mouse come semplice "cella vibrazionale". Abilitando il posizionamento assoluto sulla tavoletta ed eliminando l'accelerazione si è notato un netto miglioramento per quanto riguarda l'orientamento sulla mappa e la percezione degli spostamenti.

Percorso guidato La terza fase della sperimentazione è stata fatta nel tentativo di migliorare ulteriormente la percezione dello spazio e in particolare degli spostamenti relativi fatti con il puntatore.

Per questa fase è stato utilizzato il joystick Microsoft Sidewinder Force Feedback. Il vantaggio di questo dispositivo è che può essere utilizzato con lo stesso software e con lo stesso plug-in del mouse Logitech. Come già accennato, oltre ai feedback vibrazionali è possibile controllare via software i movimenti della manopola e quindi sincronizzarli con la mappa per poter guidare l'utente attraverso un determinato percorso.

Esplorazione con touch screen La ricerca nell'ambito dell'esplorazione tattile è continuata cercando di utilizzare device alternativi al mouse.

Attraverso il touchscreen si può risolvere il problema degli spostamenti relativi: di fatto avendo sotto mano il monitor in tutta la sua interezza e spostando il dito su di esso, si può avere la certezza di quali siano gli spostamenti in quanto il puntatore si viene a trovare proprio al di sotto del dito. È necessario tuttavia che il monitor non sia eccessivamente grande in modo da poter avere sempre dei punti di riferimento fissi ed evitare anche qui una perdita di orientamento.

Tablet PC e PDA Per unire i vantaggi della penna elettronica (maggiore precisione nei movimenti e corrispondenza tra spostamento della mano e del puntatore) e del touch screen (corrispondenza tra posizione del dispositivo di puntamento e del puntatore stesso oltre che migliore percezione dello spazio) sono stati utilizzati un tablet PC e un PDA, ottenendo ottimi risultati nell'esplorazione tattile e spaziale della pagina e delle immagini.

3.9 Validazione e valutazione di accessibilità

Come detto, il sito è stato realizzato seguendo le linee guida del WCAG e quelle illustrate nel capitolo 2 per l'accessibilità multimodale.

Tutte le pagine del sito sono state sottoposte ad una serie di validazioni oggettive attraverso strumenti automatici.

Per quanto riguarda il codice, tutte le pagine risultano essere documenti XHTML v1.0 validi secondo il DTD "strict". Il test è stato fatto con il validatore automatico del W3C¹³ e può essere costantemente verificato attraverso il link posto in fondo ad ogni pagina.

In modo analogo sono stati verificati i documenti CSS realizzati¹⁴. Per permettere una maggiore flessibilità, in ogni pagina vengono inclusi tre diversi fogli di stile. Uno fisso che definisce le caratteristiche comuni, cioè non modificabili in base all'impostazione dell'utente. Altri due

¹³<http://validator.w3.org/check>

¹⁴<http://jigsaw.w3.org/css-validator/>

variabili rispetto al tipo di contrasto e alla grandezza del testo. In questo modo si possono ottenere facilmente tutte le possibili combinazioni di visualizzazione.

Il sito ha inoltre raggiunto il massimo grado di conformità definito dal W3C (triple-AAA) e dal “Section 508”, verificato attraverso “Bobby”¹⁵ e Cinthia¹⁶.

I testi sono stati valutati, nella maggior parte dei casi, attraverso l’indice di leggibilità GulpEASE utilizzando il servizio CENSOR di Éulogos¹⁷.

Naturalmente, per la verifica di accessibilità, l’uso dei validatori automatici è stato solo il primo passo della validazione. L’effettiva accessibilità e usabilità del sito è stata valutata attraverso un’analisi soggettiva della struttura delle pagine, della navigabilità del sito, dell’adattabilità e della leggibilità dei contenuti da un punto di vista cognitivo.

Poiché è stata data grande attenzione al problema della disabilità, sono stati utilizzati diversi strumenti, comunemente utilizzati per l’ausilio alla navigazione, primi fra tutti uno screen reader (prevalentemente JAWS, ma anche HAL e Windows Eye) e la barra braille.

Nel capitolo 5 verrà illustrata più approfonditamente la metodologia adottata per la verifica soggettiva di accessibilità sottolineando l’aspetto legato alla multimodalità.

Saranno inoltre riprese e approfondite le osservazioni fatte relativamente alla navigazione attraverso feedback audio, tattili e vibrazionali; osservazioni ottenute grazie all’esperienza diretta e al coinvolgimento di persone esterne al progetto tra cui anche disabili.

¹⁵<http://bobby.watchfire.com/>

¹⁶<http://www.cynthiasays.com/org/cynthiatested.htm>

¹⁷<http://www.eulogos.net>

Capitolo 4

Realizzazione di un trasduttore vocale

In questo capitolo verrà illustrato un sistema di trasformazione delle pagine HTML realizzate tenendo conto dei criteri di accessibilità descritti nel capitolo 2.

La stessa applicazione realizzata è servita a testare la bontà dei criteri individuati e per migliorarli attraverso raffinamenti successivi.

Ritengo che la realizzazione di una interfaccia vocale usabile creata in modo automatico a partire dalla corrispondente interfaccia visuale, sia un buon metro di valutazione oggettivo dell'effettiva accessibilità multimodale di un sito.

L'applicazione non si ferma all'interazione dell'utente tramite comandi vocali, ma è stata realizzata tenendo conto della possibilità di creare diverse modalità di interfacciamento con l'utente attraverso non una trascodifica totale ma mediante una trasformazione che si limitasse ad arricchire il codice HTML della pagina del sito con altri linguaggi quali VoiceXML (secondo le specifiche X+V[32]), SALT[7], e altre modalità di interazione (mouse gesture, feedback tattili).

Il transcoder, pur essendo usabile come strumento a se stante, è un componente fondamentale della piattaforma Multiabile illustrata nel ca-

pitolo 3 in quando consente la navigazione dell'intero sito attraverso una interfaccia vocale e di accedere alle notizie prelevate da altri siti attraverso gli RSS.

4.1 Requisiti della VUI (Voice User Interface)

Il requisito fondamentale dell'interfaccia vocale creata automaticamente, oltre ovviamente a quello di rendere accessibile tutte le informazioni presenti nella pagina e nel sito web di interesse, è quello dell'usabilità.

I principi di usabilità di una interfaccia vocale sono fondamentalmente delle deduzioni soggettive ricavate attraverso il suo utilizzo ¹. In particolare ritengo che debbano essere rispettate le seguenti linee guida:

- Accesso tramite comandi vocali e/o DTMF ai componenti della pagina. In particolare i comandi DTMF sono molto comodi da associare ai comandi vocali in quanto escludono di fatto il problema del riconoscimento.
- Utilizzare comandi vocali semplici, composti da poche parole. Anche per quanto riguarda i menu per i quali le grammatiche di riconoscimento sono generate in automatico, è opportuno utilizzare titoli brevi ²
- Durante la lettura del contenuto, l'utente deve poter:
 - andare avanti e indietro saltando i paragrafi o le frasi;
 - tornare all'inizio o andare alla fine del blocco;
 - fermare momentaneamente la lettura e riprenderla dal punto in cui l'aveva lasciata;

¹Le linee guida sono state ricavate anche a seguito di prove dirette, da interviste con non vedenti e da articoli presenti in letteratura quali quello di Pèrez-Quiñones, N. Dannenberg e R. Capra [39]

²Questa indicazione va presa in considerazione per i siti internet che si vogliono rendere accessibili attraverso un'interfaccia vocale

- per quanto riguarda le tabelle di dati bisogna consentire di avere diversi criteri di lettura: lettura per righe, per colonne oppure scelta dell'utente della cella da leggere.
- I comandi dell'utente devono avere la priorità sulla lettura da parte del sintetizzatore: l'utente deve poter interrompere la lettura in modo che il sistema sia maggiormente interattivo³.
- È importante fornire in ogni momento un supporto all'utente per fargli capire in quale punto della pagina si trova e indicargli quali sono i comandi a disposizione (help contestualizzato).
- Navigabilità nell'intero sito: possibilità di seguire gli hiperlink presenti nella pagina che sta ascoltando o di accedere alla mappa del sito per saltare rapidamente ad un'altra.
- Consentire all'utente di scegliere la velocità di lettura, se il browser vocale lo permette.
- Il sistema deve potersi adattare alle capacità dell'utente: l'utente esperto non vuole sentire continuamente suggerimenti su come effettuare le operazioni e quello inesperto non deve essere abbandonato ma deve ricevere continuamente istruzioni su cosa fare.
- Dare all'utente feedback adeguati su ciò che sta facendo. In particolare, quando viene riconosciuto un comando, indicare con un suono o con un messaggio che il comando è stato recepito e che il sistema sta provvedendo ad eseguirlo.

4.2 Il problema del transcoding

VoiceXML è il linguaggio di riferimento per la creazione di interfacce puramente vocali. La progettazione di tali interfacce è completamente

³Il linguaggio VoiceXML ha in proposito la funzione di barge-in che va utilizzata diffusamente

diversa da quella di una visuale pertanto in genere si preferisce realizzare una analisi ad hoc per la loro implementazione.

Quello che si vuole ottenere qui, tuttavia, è la possibilità di accedere a informazioni già presenti in rete, e strutturate per essere fruite attraverso un normale browser, realizzate quindi in HTML. Si vuole dimostrare che se vengono applicati i criteri di accessibilità multimodale è possibile farlo attraverso un sistema di trasformazione delle pagine.

Il problema non si riduce alla semplice trasformazione di HTML in VoiceXML in quanto quello che si vuole ottenere è una interfaccia usabile in modo da provare il livello di accessibilità raggiunto dalla pagina web.

Nella prossima sezione verrà illustrata la tecnica adottata per la trascodifica.

4.3 Transcodifica attraverso l'uso delle annotazioni

La tecnica di trascodifica mediante l'utilizzo di annotazioni esterne è stata indicata dal W3C già nel 1999 [40].

L'idea era quella di arricchire le pagine web con delle metainformazioni che consentissero di realizzare sistemi automatici per adattare il contenuto alla trasmissione a dispositivi dalle capacità limitate per quel che concerne la visualizzazione, la banda ecc... e quindi inizialmente si parlava di PDA, cellulari o semplicemente sistemi dotati di una banda di trasmissione limitata.

L'idea prevedeva di adottare un sistema per effettuare le annotazioni da affiancare agli strumenti di authoring tradizionali (Figura 4.1).

Attraverso l'uso di documenti RDF[21] e del linguaggio XPointer [56] si cerca di attribuire un livello di importanza agli elementi della pagina, di indicare i requisiti richiesti per la visualizzazione, di indicare alternative in base alla capacità del dispositivo e così via.

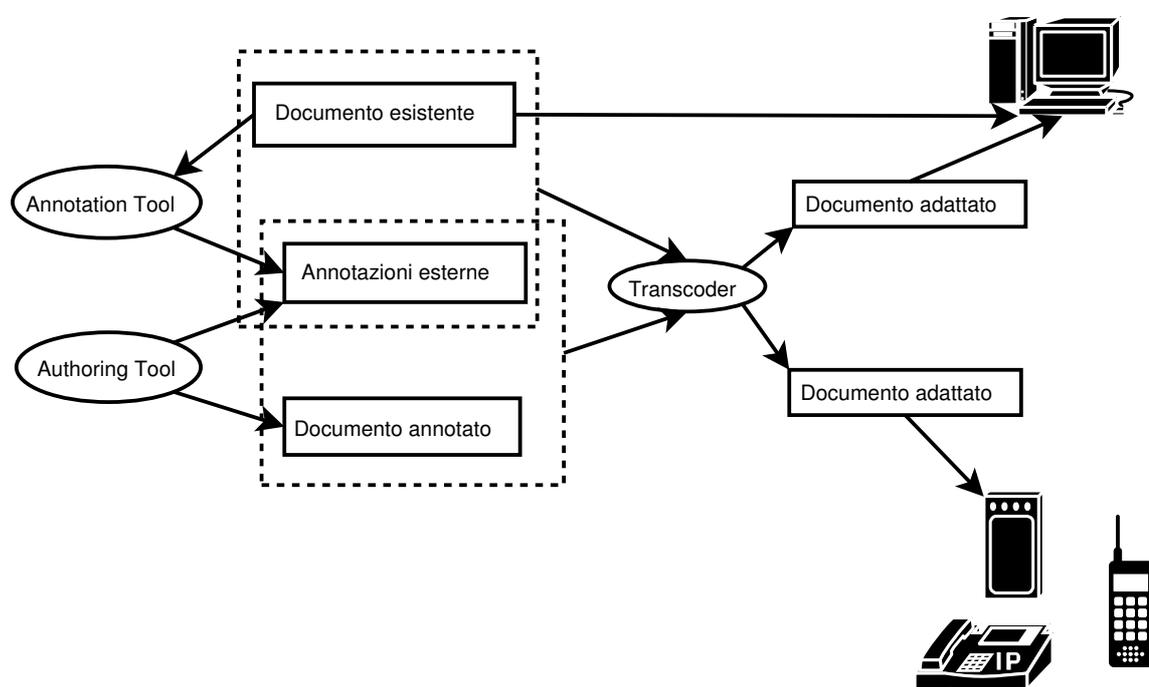


Figura 4.1: Scenario d'uso annotazioni per transcoding

Il lavoro richiesto allo sviluppatore dei siti risulta in questo modo estremamente oneroso, in quanto non solo deve realizzare diverse alternative alla visualizzazione dei contenuti ma deve creare numerosi documenti aggiuntivi di annotazioni che peraltro devono essere costantemente aggiornati assieme ai contenuti stessi.

Esistono alcuni lavori basati su un sistema analogo a quello qui illustrato [75] [34], tuttavia non è mai stata presa in considerazione l'accessibilità dei siti di partenza come prerequisito essenziale per ottenere un transcoding che portasse ad una interfaccia vocale veramente usabile. L'obiettivo principale di questi sistemi è di solito indicato come quello di distribuire i contenuti per diversi dispositivi che accedano al web[17].

4.4 Implementazione del proxy multimodale

Di seguito viene descritta l'implementazione di una web application basata sul framework Cocoon per la trasformazione di una pagina HTML in una VoiceXML che rispetti i requisiti di usabilità esposti.

La trasformazione in VoiceXML è solo una delle possibili uscite del proxy infatti l'uso di XML, che consente una netta separazione fra i contenuti e la logica oltre al semplice sistema di annotazioni descritto nel capitolo 2 e ai criteri di accessibilità, permetterà di arrivare ad un sistema che garantisca accessibilità multimodale a 360 gradi (Figura 4.2).

4.4.1 Utilizzo di Cocoon come framework di base

Tra tutti i framework per l'implementazione di piattaforme multicanale (e quindi anche multimodale), senza dubbio il progetto Cocoon della Apache Group Foundation è uno dei più importanti e avanzati[11].

Si tratta fondamentalmente di un content management system basato su XML[66]. Consente di generare e gestire contenuti scritti in questo linguaggio e di trasformarli per essere adattati al tipo di browser o periferica

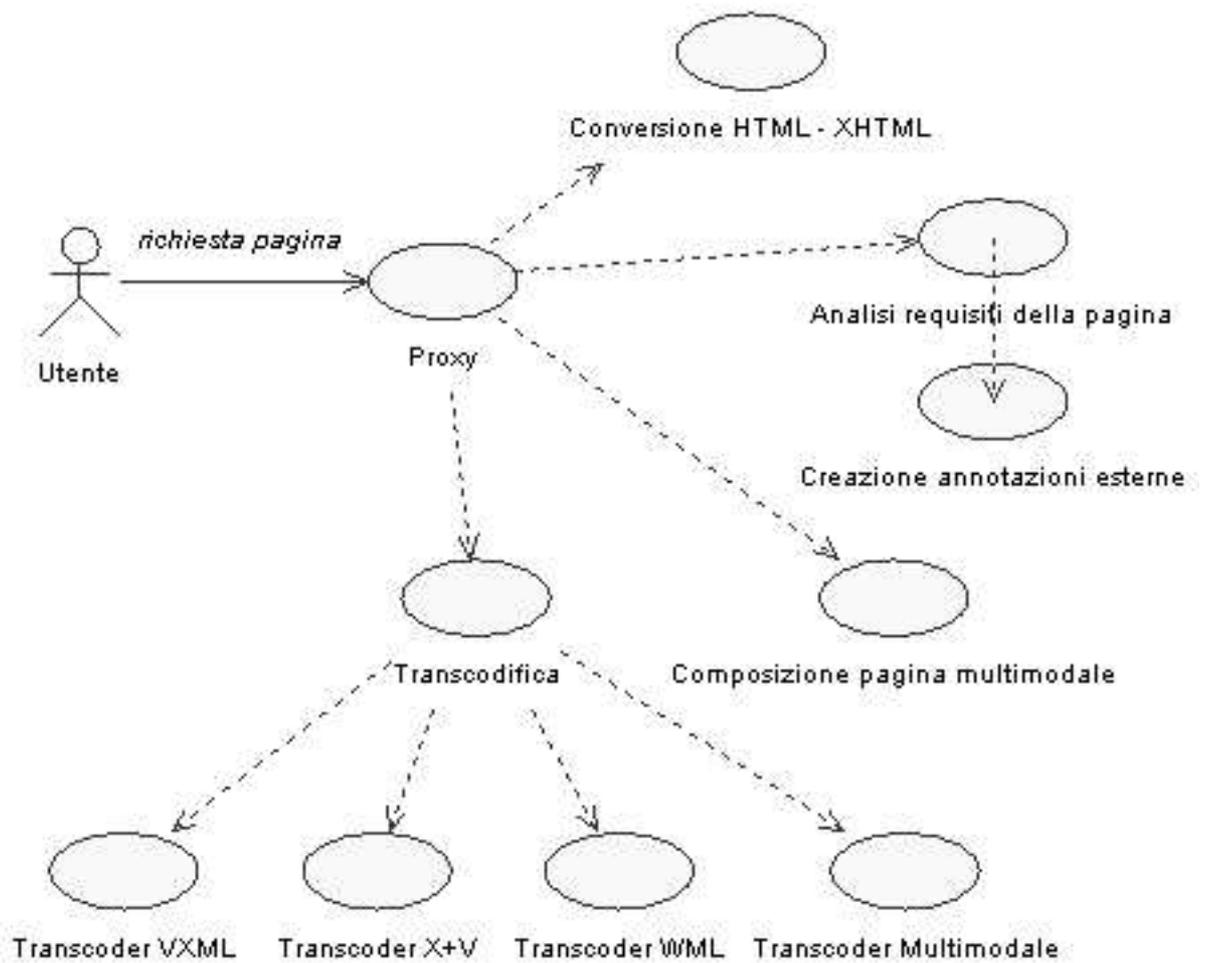


Figura 4.2: Caso d'uso del proxy multimodale

utilizzati dall'utente.

Il metodo principale di trasformazione dei documenti XML è quello dei fogli di stile scritti usando il linguaggio XSLT[70].

Il framework è altamente configurabile e personalizzabile scrivendo opportune parti di codice (Java). Il risultato che si ottiene è quello di una web application che può essere eseguita all'interno di un container come Tomcat[12].

Il trattamento degli XML avviene attraverso la definizione di alcune "pipeline" che descrivono le diverse fasi dalla generazione del contenuto al suo delivery.

Ogni fase della pipeline viene portata avanti da una classe Java che riceve l'output della fase precedente e produce un output per la fase successiva.

Una tipica pipeline è costituita dai seguenti componenti:

- Un *generatore*: produce un documento XML a seguito di una richiesta da parte dell'utente. Esistono molti modi predefiniti per la generazione di un XML, quello più semplice è quello di leggere un file XML dal disco.
- Zero o più *trasformatori*: ricevono in ingresso l'XML prodotto dal generatore della pipeline e lo trasformano in un altro formato (in un altro XML, in HTML o altro). Il documento trasformato può essere passato ad un altro trasformatore oppure al serializzatore per il delivery.
- Un *serializzatore*: riceve il documento finale che deve essere inviato al client e quindi si occupa di gestire la risposta alla richiesta da cui si è partiti all'inizio della pipeline.

Nel trattamento di contenuti in XML con Cocoon, una pipeline deve essere sempre costituita da un generatore e da un serializzatore, mentre

non sempre è necessario avere un trasformatore; è possibile infatti che si desideri distribuire direttamente contenuti in XML nativo.

Ovviamente è possibile gestire anche altre risorse che non siano XML, come ad esempio, immagini, fogli di stile, documenti in formato binario ecc. . .

Per questi tipi di file si costruiscono tipologie di pipeline diverse che non verranno qui descritte.

Questa netta separazione tra la fase di generazione e quella di delivery consente, a partire dalla stessa fonte dati, di presentare il contenuto in un formato adatto al client dell'utente, che sia esso un browser grafico o un browser vocale.

4.4.2 L'architettura software

Affinché l'applicazione sia usabile si vuole fare in modo che l'utente richieda la pagina web che vuole visitare e che questa venga trasformata automaticamente in una pagina vocale.

Per far ciò l'architettura realizzata è quella del proxy illustrato in figura 4.3.

L'utente richiede la pagina al proxy il quale inoltra la richiesta al server remoto e ne riceve la risposta.

La pagina ottenuta viene dapprima manipolata in modo da essere trasformata in XML, vengono corretti tutti i riferimenti ad altre pagine in modo che ogni nuova richiesta del client passi nuovamente attraverso il proxy. Vengono inoltre ricavati i documenti RDF di descrizione del contenuto del sito e della pagina e anch'essi inseriti nel documento XML.

A questo punto il compito del generatore della pipeline è concluso, nella seconda fase si provvede alla trasformazione del documento XML ricavato nel formato richiesto dall'utente. Attraverso un foglio di trasformazione si ottiene ad esempio un documento VoiceXML accessibile attraverso un browser vocale.

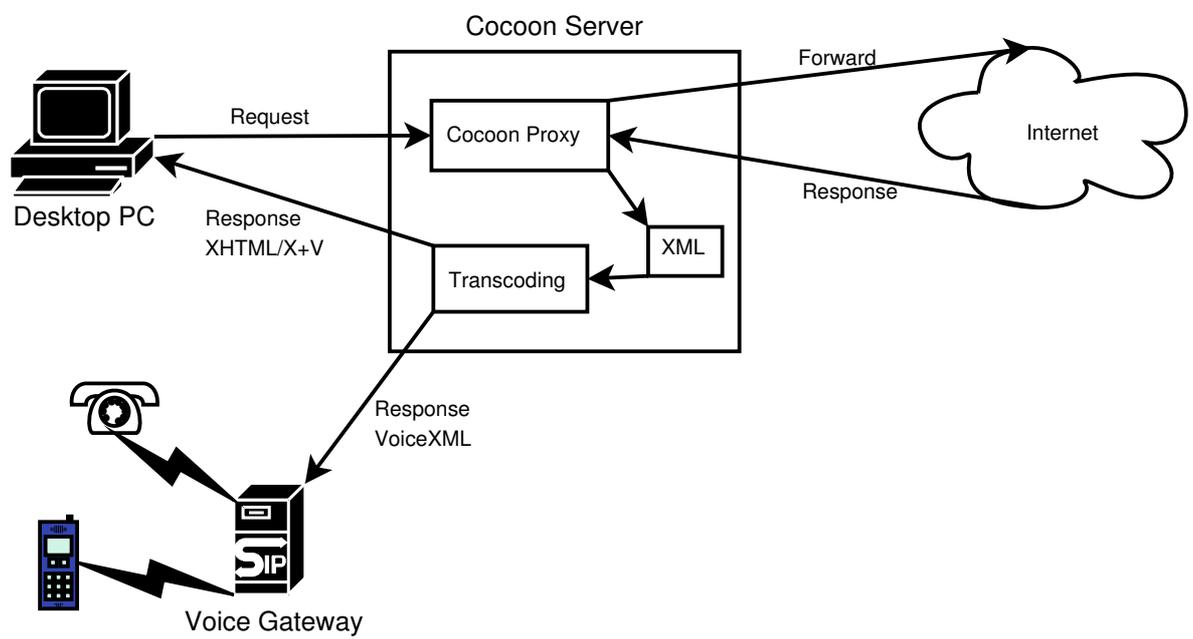


Figura 4.3: Architettura del proxy con Cocoon

Nell'ultima fase il documento trasformato viene rimandato al client che può così accedere ai contenuti attraverso il tipo di interfaccia richiesta.

4.5 Richiesta dell'utente

La richiesta di una URL da parte dell'utente viene fatta ad una servlet della web application. Supponiamo quindi che l'indirizzo della servlet sia `http://localhost:8080/mmp/proxy` e che si voglia ottenere la versione vocale della pagina all'indirizzo `http://www.google.com/`, si dovrà digitare la URL seguente:

```
http://localhost:8080/mmp/proxy?url=http://www.google.com/
```

Il parametro `url` della querystring indica appunto l'indirizzo della pagina che si vuole ottenere.

Esistono due ulteriori parametri che possono essere passati alla servlet:

- `type`: indica il tipo di trasformazione richiesta. Il valore di default è `vxml` e indica appunto che è richiesto la trasformazione in Voice-XML della pagina. Vi sono poi i valori `xhtml` per ottenere appunto la versione XHTML da visualizzare in un browser normale e `xml` usato solo ai fini di debug per visualizzare l'XML generato prima della trasformazione.
- `action`: usato in alternativa a `url` consente di tornare indietro o avanti nell'history delle url richiamate dall'utente. Questo parametro è necessario nel momento in cui l'utente utilizza un browser che non consenta già questa operazione, tipicamente un browser vocale. Il server tiene traccia delle URL chiamate dall'utente durante una sessione di collegamento allo stesso modo in cui lo fa un browser visuale.

4.6 Inoltro della richiesta da parte del proxy

Una volta stabilita quale URL deve essere caricata (in base al parametro `url` o `action`), il proxy apre a sua volta una connessione al server indicato dalla URL stessa⁴ e “scarica” localmente la pagina corrispondente.

Il proxy utilizza la libreria HTTP Client [13] che consente di simulare lo stesso comportamento di un browser e quindi mantenere le sessioni sul server remoto, ottenere e ritrasmettere i cookie ecc. . .

Ai fini della trascodifica è sufficiente ricevere solo la pagina HTML e gli eventuali documenti RDF index e contents mentre gli altri oggetti quali immagini, applet, fogli di stile, script ecc. . . non vengono gestiti dal server ma si lascia che il client li scarichi direttamente dal server remoto⁵.

4.7 Transcodifica

La trascodifica costituisce il cuore dell’applicazione. Essa consiste nella trasformazione della pagina HTML in un nuovo formato, quello sui cui ci si concentrerà è il formato VoiceXML. La trascodifica avviene in passaggi successivi: inizialmente viene creato un documento XML contenente la pagina stessa opportunamente manipolata e, se presenti, i documenti “index” e “contents” associati alla pagina. Successivamente avviene la trascodifica vera e propria attraverso opportuni fogli di trasformazione XSLT.

4.7.1 “Pulitura” della pagina web attraverso TIDY

Per poter effettuare la trascodifica utilizzando dei fogli di trasformazione è necessario che il documento di partenza sia in formato XML.

HTML non rispetta le caratteristiche un normale XML e molto spesso

⁴Al momento è stata implementata solo la connessione HTTP

⁵Solo nel caso in cui venga usato un browser visuale adatto alla presentazione di tali oggetti

le pagine che si trovano su internet non sono nemmeno sintatticamente valide, seppure i browser consentano ugualmente una corretta visualizzazione.

È necessario quindi operare una prima trasformazione di “pulitura” della pagina e trasformazione in XHTML[69]. Per fare ciò viene utilizzato Tidy[23] che opportunamente configurato consente di operare alcune modifiche che rendono la pagina un XML valido e la rendono poi più facilmente trasformabile in seguito, in particolare:

- chiude tutti i tag lasciati aperti
- corregge le sovrapposizioni dei tag
- elimina gli attributi replicati
- inserisce l'attributo `alt` mancante alle immagini
- trasforma i tag `b` e `i` nell'equivalente logico `strong` e `em`
- inserisce i blocchi di testo in tag `p` in modo che non siano isolati
- elimina i tag `p` vuoti
- elimina i tag di visualizzazione `font`

I parametri di trasformazione di Tidy sono memorizzati in un file di configurazione che può quindi essere opportunamente modificato in base alle esigenze.

4.7.2 Estrazione dei file `index` e `contents`

Una volta “ripulita” la pagina e quindi trasformata in XHTML si estraggono le URL dei documenti “`index`” e “`contents`” indicati nell'intestazione (2.3). Per estrarre tali informazioni si usano delle semplici espressioni XPATH[30]. Ovviamente è prevista la possibilità che tali documenti non siano presenti, in tal caso verranno creati due documenti vuoti.

Esiste tuttavia la possibilità, lasciata per il momento a sviluppi futuri, di poter creare tali documenti a partire dalle informazioni presenti nella

pagina: ad esempio si possono individuare i link ad altre pagine del sito e quindi costruire il documento “contents” oppure, con tecniche più sofisticate, individuare la topologia della pagina e la semantica dei blocchi in cui è composta[49][50] per realizzare il documento “index”.

Nel caso in cui i documenti siano presenti, il proxy li ottiene dal server remoto ed effettua il parsing per verificarne la correttezza sintattica e poterli inviare alla fase di transcoding.

4.7.3 Correzione delle URL

Per fare in modo che tutte le url a pagine HTML siano fatte passare dal proxy, vengono ricercati gli attributi `href` della pagina e corretti.

Lo stesso viene fatto per i tag `link` presenti nel documento contents.

Per la trasformazione in XHTML, destinata ai browser visuali vengono corretti anche tutti i riferimenti ai CSS, script, immagini ecc... in modo che vengano richiesti direttamente al server remoto anzichè passare dal proxy.

4.7.4 Definizione di un XML intermedio per l’accesso multicanale

Per ottenere una VUI efficace, oltre alla pagina XHTML si dovranno usare le informazioni presenti dei documenti RDF collegati, pertanto viene creato un XML temporaneo avente la seguente struttura:

```
1 <?xml version='1.0' encoding='ISO-8859-1'>
2 <mmpage>
3   <index>
4     ... qui viene inserito l'index RDF ...
5   </index>
6   <contents>
7     ... qui viene inserito il contents RDF ...
8   </contents>
9   <page>
10    ... qui viene inserita la pagina XHTML ...
```

```
11 </page>
12 </mmpage>
```

4.7.5 Prima trasformazione

Per facilitare la trasformazione in VoiceXML attraverso un foglio di stile è sorta la necessità di semplificare ulteriormente la pagina eliminando alcuni tag e in alcuni casi il loro contenuto.

Questa prima fase coinvolge solo la pagina e quindi solo il nodo `page` dell'XML intermedio che alla fine rimarrà strutturalmente identico.

I nodi che vengono completamente eliminati (tag e contenuto) sono `script`, `noscript` e `style`, si tratta infatti di tag necessari alla visualizzazione grafica e all'esecuzione di codice lato client.

Alcuni tag vengono semplicemente cancellati dal sorgente mentre viene mantenuto il loro contenuto: `strong`, `em` e `center` che sono opzioni di visualizzazione, a il cui attributo `href` comincia con “#” trattandosi di link interni alla pagina che non vengono trattati nell'interfaccia vocale.

4.7.6 Seconda trasformazione

La trasformazione dall'XML temporaneo al VoiceXML avviene attraverso un XSL che effettua fondamentalmente le seguenti operazioni:

- crea un menu del sito e uno della pagina utilizzando le informazioni presenti nelle sezioni “index” e “contents”
- individua il contenuto principale della pagina e gli altri componenti utilizzando le informazioni presenti nell'index e crea dei tag `form` di VoiceXML separati per ciascuno di essi
- il contenuto di ciascun blocco viene separato in “frasi” secondo una politica che verrà illustrata in seguito, in modo da consentire una maggiore semplicità di navigazione da parte dell'utente che potrà

andare avanti e indietro tra le frasi o mettere temporaneamente in pausa la lettura.

- estrae il testo che deve essere letto e le informazioni associate agli elementi non testuali come le immagini, i filmati flash ecc. . .
- i link che portano all'esterno vengono trasformati in modo da poter essere seguiti e richiedere quindi la trascodifica di un'altra pagina.

Separazioni in sottodialoghi Vediamo ora in dettaglio come viene costruita l'interfaccia vocale e quali trasformazioni vengono fatte in base ai tag HTML incontrati nella pagina.

Il documento VoiceXML è costituito da diversi sottodialoghi realizzati con il tag `form` (non è stato utilizzato il tag `menu`). Gli elementi della pagina che aprono una nuova form sono:

- il menu di pagina e il menu del sito, illustrati in seguito (4.7.9)
- ciascuno dei blocchi in cui è divisa la pagina, individuati dal nodo "index"
- le tabelle HTML: tag `table` (4.7.7)
- le form HTML: tag `form` (4.7.8)

Le tabelle e le form sono strutture che richiedono una interfaccia vocale particolarmente complessa per poter essere rese in modo soddisfacente. Più avanti verrà illustrata la soluzione adottata per questi due elementi anche se l'implementazione non è da considerarsi definitiva.

Sommario della pagina Al caricamento della pagina viene letto un messaggio del tipo: "Caricata pagina dal titolo: . . .", in modo da dare all'utente un feedback su ciò che sta avvenendo. Il titolo è quello presente nel tag `title` presente nell'intestazione della pagina.

Esistono altre informazioni preliminari che potrebbe essere utile fornire all'utente al momento del caricamento della pagina. JAWS ad esempio

conta il numero di intestazioni (tag H1, H2, H3, ecc...) e di link e li comunica all'utente.

Per la verità è difficile stabilire quali informazioni siano effettivamente utili all'utente. In questa implementazione ho deciso di non dare molte informazioni se non quelle strettamente necessarie alla comprensione dei contenuti e di lasciare all'utente la scelta di ascoltare o meno queste informazioni accessorie.

È prevista la realizzazione di una modalità "verbose" disabilitata di default e attivabile dall'utente che aggiunge tutta una serie di informazioni aggiuntive rispetto a quelle scritte nella pagina.

Soprattutto nel caso in cui l'accesso viene effettuato per via telefonica, ci si rende conto usando un sistema di questo tipo che per un utente esperto troppe informazioni oltre ad essere considerate inutili possono essere anche fastidiose e fanno perdere tempo.

Separazione in frasi Ciascuna form VoiceXML costruita a partire da un blocco viene suddivisa in frasi per poter consentire un elevato livello di interattività con l'utente. Uno dei requisiti che ci è infatti quello di consentire all'utente di muoversi agevolmente anche all'interno di una pagina complessa, senza costringerlo ad ascoltare l'intero contenuto per trovare l'informazione richiesta o ancora per poter riascoltare parti di testo già passati.

Questo elemento costituisce un importante elemento di innovazione rispetto ad altri sistemi di trascodifica esaminati, tra cui il WebSphere Transcoding Publisher (WTP) di IBM[27], che tendono a presentare un testo unico per l'intera pagina con scarsa se non assoluta possibilità di navigazione o controllo sulla lettura, limitandosi ad individuando i titoli indicati nella pagina HTML con i tag h1-h6 [48].

Ciascuna frase viene resa in VoiceXML con un nuovo tag `field` della form in cui è contenuta. I campi delle form in VoiceXML sono fatti per

ricevere un input da parte dell'utente, in questo caso però vengono sfruttati per operare questa separazione e per poter presentare il testo in tag `prompt` che possono essere interrotti dall'utente con comandi vocali⁶.

Il normale flusso del dialogo è garantito dall'algoritmo FIA di VoiceXML⁷ che consiste fundamentalmente nel presentare in modo automatico all'utente un campo della form una volta che il precedente viene riempito.

In questo caso, poichè i campi non sono effettivamente riempiti dall'utente si provvede a riempirli in automatico per fare in modo che motore di sintesi passi a leggere il campo successivo⁸.

Ecco ad esempio come vengono rese in VoiceXML due frasi consecutive che possiamo supporre siano contenute in due nodi `p` consecutivi nella pagina HTML originale:

```
1  ...
2  <field name="sentence1">
3    <prompt timeout="1ms" bargein="true">
4      Prima frase
5    </prompt>
6    <noinput>
7      <assign name="sentence1" expr="'" />
8    </noinput>
9  </field>
10 ...
11 <field name="sentence2">
12 <prompt timeout="1ms" bargein="true">
13   Seconda frase
14 </prompt>
15 <noinput>
16   <assign name="sentence1" expr="'" />
17 </noinput>
18 </field>
19 ...
```

Il FIA seleziona il prompt corrispondente al campo "sentence1" (riga 2).

Al termine della lettura della frase, l'utente ha a disposizione *1ms* di tempo per immettere un comando vocale o DTMF. Ovviamente non è un tempo

⁶Tutti i tag "prompt" hanno l'attributo "bargein" posto a true

⁷FIA: Form Iteration Algorithm

⁸L'interfaccia vocale è realizzata secondo la tecnica della cosiddetta "mixed initiative": sia l'utente che il sistema possono "compilare" i campi della form e prendere decisioni riguardo la navigazione

sufficiente a compiere alcuna operazione ma l'attributo `bargein` posto a `true` consente di dare il comando anche durante la lettura del prompt. Di fatto si vuole che l'utente dia i comandi solo mentre sta ascoltando la lettura in modo da non avere tempi di attesa lunghi e inutili per il passaggio da una frase all'altra.

Allo scadere del tempo viene generato l'evento `noinput` che viene intercettato (riga 6) in modo da assegnare un valore al field "sentence1" (riga 7). In questo modo l'algoritmo FIA riconosce il campo come riempito e passa al campo successivo e così via.

Sono diversi gli elementi della pagina che costituiscono una nuova frase e che quindi risultano essere gli elementi atomici della sintesi vocale. Tali elementi sono: i paragrafi (tag `p`), le intestazioni (tag da `H1` a `H6`), gli elementi di una lista (tag `li`), i collegamenti (tag `a`), le immagini (tag `img`), i moduli (tag `form`) e le tabelle (tag `table`).

Alcuni di questi come liste, collegamenti, immagini, form e tabelle richiedono un approfondimento su come sono realizzati nell'interfaccia vocale.

Le liste In HTML esistono le liste numerate (`ol`), le liste non numerate (`ul`) e le liste di definizioni (`dl`).

Tali elementi per la verità, pur essendo strutturati non comportano una particolare difficoltà implementativa nell'interfaccia vocale.

Ciascuna lista viene annunciata ad esempio per le liste numerate con una frase del tipo: "Inizio di una lista numerata di . . . elementi". Successivamente ogni elemento della lista viene annunciato con la frase "Elemento numero . . ." seguito dal contenuto che può essere un testo semplice o anche un collegamento o altro ancora.

Per le liste di definizione, ogni riga è costituita da una coppia di elementi: il termine della definizione (`dt`) e il testo vero e proprio della definizione (`dd`).

In tutti i casi, gli elementi della lista vengono trattati come delle frasi semplici (a meno che non contengano elementi più complessi quali link, immagini o altri elementi) pertanto l'utente può "navigare" nella lista con gli stessi comandi che ha a disposizione nel resto del testo.

I collegamenti I collegamenti sono uno dei principali elementi di interattività del web e anche nell'interfaccia vocale devono poter essere usati per rendere navigabile il sito.

La realizzazione in VoiceXML è relativamente semplice: il link viene annunciato dal sintetizzatore vocale con la frase "Collegamento. (il titolo del collegamento è ...)". Il titolo indicato nella frase corrisponde all'attributo `title` del tag `a`.

All'utente vengono dati 3 secondi per decidere se seguire o meno il link. Nel caso in cui l'utente non dica niente o richieda di andare avanti nella lettura il link viene saltato.

Di seguito una porzione di codice VoiceXML che illustra come viene reso il tag ⁹:

```
<a href="/index.html" title="home page del sito">Home</a>

1 <var name="link" expr="" />
2 <link event="follow_link">
3   <grammar>vai | segui | seguilo | ok</grammar>
4   <dtmf>5</dtmf>
5 </link>
6 <catch event="follow_link">
7   <goto expr="link" />
8 </catch>
9 ...
10 <block name="sentence10_assign">
11   ...
12   <assign name="link"
13     expr="/index.html">
14   </assign>
15 </block>
16 <field name="sentence10">
17   <prompt bargein="true" timeout="3s">
```

⁹Per semplicità ho ommesso di indicare l'indirizzo completo, quello che fa in modo che la chiamata della URL venga fatta attraverso il proxy

```
18     Collegamento.  
19     (Il titolo del collegamento home page del sito).  
20 </prompt>  
21 <noinput>  
22     <assign name="sentence10" expr="''"/>  
23 </noinput>  
24 </field>
```

Per prima cosa è stata creata una apposita grammatica di riconoscimento per la navigazione attraverso i collegamenti (righe 2-5) e vi è stato associato un evento (righe 6-8). La grammatica e l'evento sono definiti a livello della form VoiceXML di ciascun blocco pagina.

Nel momento in cui si arriva ad un collegamento, alla variabile `link` definita all'inizio viene assegnato come valore l'indirizzo presente nell'attributo `href`. Quando viene letto il collegamento, nel caso in cui l'utente decide di seguirlo dovrà dire una parola tra “vai”, “seguì” “segui”, “ok” o digitare 5 sul tastierino del telefono. In questo modo si genererà l'evento “follow_link” che indica al browser vocale di caricare il VoiceXML al nuovo indirizzo (riga 7).

4.7.7 Serializzazione delle tabelle

Le tabelle sono elementi della pagina difficili da riprodurre in una interfaccia vocale, questo perchè sono strutture bidimensionali e molto spesso esistono diversi modi per essere lette e quindi diversi modi in cui dovrebbero essere serializzate in una VUI.

La modalità di lettura di una tabella dipende fortemente dai dati che vi sono in essa rappresentati e quindi non è possibile stabilire a priori in modo automatico qual'è la migliore interfaccia vocale da realizzare.

Le tabelle possono essere lette per righe o per colonne, o ancora possono essere a doppia entrata: si sceglie riga e colonna per individuare una cella specifica. La figura 4.4 mostra un esempio di tabella a doppia entrata realizzata in HTML. Il fatto che sia a doppia entrata lo si evince guardando la resa grafica della tabella stessa e dai dati in esso contenuti. Risulta dif-

facilmente deducibile in maniera automatica analizzando il codice se non vengono utilizzati correttamente i tag e gli attributi.

Tabella 1. - Numero di persone disabili di sei anni e più che vivono in famiglia, per sesso e classi d'età - Anno 1999-2000 (dati in migliaia)

Anni	6-14	15-24	25-44	45-64	65-74	75-oltre	Totale
Maschi	40	27	81	153	204	389	894
Femmine	40	32	82	209	323	1.035	1.721
Maschi e Femmine	80	59	163	362	527	1.424	2.615

Figura 4.4: Esempio di una tabella HTML a doppia entrata.

Nella prima implementazione del transcoder si adatterà una soluzione semplice che è quella di leggere le tabelle riga per riga, anticipando, prima della lettura del dato di ogni singola cella, il nome della cella o delle celle di intestazione. Si suppone che le tabelle siano sempre usate per rappresentare dati e non per costruire il layout di pagina, anche se adottando opportunamente i criteri di accessibilità si potrebbe usare una tabella semplice per definire l'intelaiatura esterna della pagina.

Il procedimento seguito per individuare l'intestazione che deve essere letta prima di ciascuna cella di dati¹⁰ è il seguente:

- nel caso in cui sia specificato l'attributo `headers`, vengono ricercate le celle i cui attributi `id` corrispondono a quelli in esso contenuti. Viene quindi letta come intestazione, il valore della cella trovata oppure il corrispondente attributo `abbr`
- nel caso in cui non sia stato specificato l'attributo `headers`, si legge la prima cella `th` della riga, se presente
- ancora nel caso di assenza dell'attributo `headers` viene letta la cel-

¹⁰solo i tag TD, vengono letti mentre i TH verranno letti solo come intestazioni

la t_h nella prima riga della tabella corrispondente alla stessa colonna cui appartiene la cella in esame

La tabella mostrata in figura 4.4 ad esempio viene letta, a partire dalla prima riga di dati, nel seguente modo:

Maschi, Anni, 6-14: 40.

Maschi, Anni, 15-24: 27.

Maschi, Anni, 24-44: 81.

...

Femmine, Anni, 6-14: 40.

Femmine, Anni, 15-24: 32.

Femmine, Anni, 25-44: 82.

...

Ciascuna cella di dati viene inserita in una “frase” diversa pertanto l’utente può agevolmente passare da una cella all’altra della serializzazione con gli stessi comandi usati per il resto del testo.

Ovviamente, più la tabella è complessa, maggiore sarà la difficoltà dell’utente nel farsi un’idea chiara delle informazioni in essa contenute. Maggiormente difficoltosa può risultare poi un’analisi più generale dei dati per chi può solo ascoltarli nell’ordine imposto dalla serializzazione. Leggendo la tabella si potrebbe ad esempio cercare di rispondere alle domande: “Per quali classi di età il numero di femmine disabili supera quello dei maschi?”, “Qual’è la distribuzione di disabili tra maschi e femmine in periodo adolescenziale?” ecc. . . . Le risposte sono immediate per chi ha una visione d’insieme della tabella, mentre può richiedere tempo e più letture per chi può solo ascoltare i dati.

Vediamo ad esempio una semplice tabella di dati. Di seguito il codice HTML.

```
1 <table border="1"
2     summary="Durante tutto il primo semestre
3     il fatturato dell'azienda è cresciuto in modo
4     costante.
5     Si è passati dai 10000 euro di Gennaio ai
```

```
6         60000 euro di Giugno.
7         ">
8 <caption>Fatturato dall'inizio dell'anno</caption>
9 <thead>
10    <tr>
11        <th>Mese</th>
12        <th>Fatturato</th>
13    </tr>
14 </thead>
15 <tfoot>
16    <tr>
17        <td>&nbsp;</td>
18        <td>&nbsp;</td>
19    </tr>
20    <tr>
21        <td><b>Totale</b></td>
22        <td>210.000</td>
23    </tr>
24 </tfoot>
25 <tbody>
26    <tr>
27        <td>Gennaio</td>
28        <td>10.000</td>
29    </tr>
30    <tr>
31        <td>Febbraio</td>
32        <td>20.000</td>
33    </tr>
34    <tr>
35        <td>Marzo</td>
36        <td>30.000</td>
37    </tr>
38    <tr>
39        <td>Aprile</td>
40        <td>40.000</td>
41    </tr>
42    <tr>
43        <td>Maggio</td>
44        <td>50.000</td>
45    </tr>
46    <tr>
47        <td>Giugno</td>
48        <td>60.000</td>
49    </tr>
50 </tbody>
51 </table>
```

La figura 4.5 mostra la tabella visualizzata dal browser.

Mese	Fatturato
Gennaio	10.000
Febbraio	20.000
Marzo	30.000
Aprile	40.000
Maggio	50.000
Giugno	60.000
Totale	210.000

Figura 4.5: Esempio di una tabella HTML semplice.

Il VoiceXML viene creato in modo che la tabella viene annunciata dal sintetizzatore vocale che indica il numero di righe contando il numero di nodi `tr` presenti in `tbody`; il numero di colonne viene ricavato dalla riga di intestazione. Dopodiché viene letto il sommario della tabella e il contenuto del nodo `caption`.

Di seguito la trascrizione di una parte della sintesi vocale:

Inizio di una tabella di 6 righe e 2 colonne.

Il titolo della tabella è "Fatturato dall'inizio dell'anno"

La descrizione della tabella è "Durante tutto il primo semestre il fatturato dell'azienda è cresciuto in modo costante. Si è passati dai 10000 euro di Gennaio ai 60000 euro di Giugno."

Riga 1. Mese: Gennaio. Fatturato: 10.000.

Riga 2. Mese: Febbraio. Fatturato: 20.000.

Riga 3. Mese: Marzo. Fatturato: 30.000.

...

Inizio righe di fine tabella.

Riga 1. Mese: . Fatturato: .

Riga 2. Mese: Totale. Fatturato: 210.000

Fine della tabella.

Il footer viene letto alla fine. In questo caso, trattandosi di una riga riassuntiva sarebbe inopportuna la lettura dell'intestazione di colonna. Anche qui si fa sentire una mancanza di semantica dell'HTML per la descrizione della tabella.

4.7.8 Transcodifica delle form

Le form rappresentano senza dubbio l'ostacolo più elevato alla trascodifica, in quanto costituiscono il più alto livello di interattività di una pagina web con l'utente.

Risulta particolarmente difficoltosa la realizzazione di una interfaccia vocale efficace in quanto richiede la creazione di grammatiche di riconoscimento spesso molto complesse che alla fine possono risultare computazionalmente improponibili.

Il problema principale si ha senza dubbio con i campi di testo liberi (input text o textarea) dove si deve consentire all'utente di inserire un testo arbitrario o parole non contenute nel proprio vocabolario: nomi propri, password, sigle, numeri ecc... La soluzione di questo problema specifico è ancora in fase di studio.

Al momento, per rendere le form in qualche modo accessibili, ho deciso di creare una grammatica per consentire di dettare il testo libero lettera per lettera. L'interfaccia che ne risulta non può essere considerata ancora pienamente usabile ma può essere un primo passo.

Per quanto riguarda i campi a "risposta chiusa" (checkbox, radio button, selection) vengono create delle grammatiche basate sul testo visualizzato per consentire la selezione all'utente.

La form HTML deve comunque essere scritta tenendo conto dei criteri descritti nel capitolo 2 (2.2.7) in modo da ottenere un più alto grado di

accessibilità anche attraverso l'interfaccia vocale.

In particolare, risulta molto utile, in caso di form composta da molti campi, raggruppare le informazioni correlate fra loro attraverso dei `fieldset`. Ogni campo di input inoltre deve avere un'etichetta associata attraverso il tag `label` e un attributo `title` che indichi bene quale sia l'input richiesto all'utente.

4.7.9 Menu di pagina e menu del sito

Per realizzare i menu di pagina e il menu del sito a partire da "index" e "contents" ho deciso di usare due distinti tag `form` anziché il tag `menu` che sarebbe preposto a questa funzione in quanto il primo è più versatile consentendo una maggiore interazione con l'utente e permette di realizzare un feedback ai comandi dell'utente.

I titoli degli `item` presenti nei documenti RDF vengono usati come grammatica di riconoscimento per ciascuna voce assieme ad un codice DTMF: per scegliere la voce di menu, l'utente non deve far altro che ripetere (anche solo in parte) il titolo oppure digitare il numero corrispondente che viene pronunciato dal motore di sintesi prima di ogni voce.

Di seguito è trascritto un esempio di menu del sito generato dal proxy:

```
1 <form id="N40003B">
2   <field name="choice">
3     <prompt>Contenuto del sito. Ripetere la voce di menu o
4     digitare il numero corrispondente.
5     <enumerate>
6       <value expr="_dtmf" />
7       <break size="small" />
8       <value expr="_prompt" />
9       <break size="large" />
10    </enumerate></prompt>
11    <option value="http://localhost:8080/mmp/proxy
12            ?url=http%3A%2F%2Flocalhost%2Fipc4d%2Fpubblico%2Fhome.jsp"
13            dtmf="1">Home page del sito</option>
14    <option value="http://localhost:8080/mmp/proxy
15            ?url=http%3A%2F%2Flocalhost%2Fipc4d%2Fpubblico%2Flinks.jsp"
16            dtmf="2">Collegamenti utili</option>
17    <option value="http://localhost:8080/mmp/proxy
```

```
18         ?url=http%3A%2F%2Flocalhost%2Ftc4d%2Fpubblico%2Finfoprog.jsp"  
19         dtmf="3">Informazioni sul progetto</option>  
20     </field>  
21     <filled>  
22         <audio src="/mmp/audio/beep.wav"></audio>  
23         Caricamento della pagina in corso, attendere...  
24         <goto expr="choice" />  
25     </filled>  
26 </form>
```

In modo del tutto analogo viene creato il menu della pagina ricavato dal documento “index”.

Entrambe i documenti sono sempre accessibili durante la lettura della pagina attraverso dei comandi vocali, pertanto si ottiene un buon “ipertesto vocale” che permette all’utente di navigare agevolmente nel sito e all’interno delle singole pagine.

4.8 I comandi di navigazione

Vengono ora illustrati brevemente i comandi di navigazione messi a disposizione dell’utente per la navigazione vocale. I comandi sono definiti attraverso semplici grammatiche JSGF inserite direttamente dentro i documenti VoiceXML, e sono costituite oltre che da parole o brevi frasi in italiano anche da codici DTMF che possono risultare più comodi per utenti esperti.

Navigazione nella pagina Per quanto riguarda la navigazione all’interno di una singola form VoiceXML (blocco di pagina individuata dal file “index”) i comandi principali sono:

- DTMF 6, *avanti*: salta alla frase o all’elemento della lista successivo
- DTMF 4, *indietro*: salta alla frase o all’elemento della lista precedente
- DTMF 1, *inizio*: ritorna all’inizio della pagina

- DTMF 2, *pausa*: interrompe momentaneamente la lettura della pagina
- DTMF 8, *riprendi*: riprende la lettura della frase interrotta con il comando pausa
- DTMF 0, *termina*: termina la lettura della pagina e interrompe la comunicazione (in caso di browsing attraverso il telefono)
- DTMF 7, *aiuto*: presenta il testo di aiuto che può variare a seconda del contesto.

Browser vocale L'accesso ai menu "index" e "contents" viene fatto attraverso i comandi seguenti:

- DTMF 3, *menu_pagina*: accede al menu della pagina, costruito da "index"
- DTMF 9, *menu_sito*: accede al menu del sito, costruito da "contents"

L'utilizzo del simbolo "_" consente di evitare l'ambiguità tra le due grammatiche: avendo la parola "menu" in comune, non è prevedibile il comportamento del browser vocale nel caso in cui l'utente non dovesse pronunciare la frase intera. In questo modo si obbliga l'utente a pronunciare correttamente la frase completa perchè venga riconosciuta.

Infine, il server mantiene per la sessione utente attiva la cronologia delle pagine visitate. Per potervi accedere sono stati creati due ulteriori comandi di navigazione:

- *pagina_indietro*: richiede al server la pagina web richiamata prima di quella attuale
- *pagina_avanti*: nel caso in cui l'utente sia andato indietro nella navigazione con il comando *pagina_indietro*, indica al server di caricare la pagina successiva nella cronologia.

4.9 Fruizione della pagina web da parte degli utenti

Per poter utilizzare l'interfaccia vocale realizzata in VoiceXML è necessario un sistema che sia in grado di interpretare i documenti vxml e di consentire l'interazione con l'utente. Fondamentalmente deve essere dotato di un sintetizzatore vocale¹¹ e di un riconoscitore vocale¹².

Per l'utilizzo dell'interfaccia vocale realizzata dal transcoder ho utilizzato due diversi strumenti: uno usato ampiamente durante la fase di sviluppo e l'altro che rappresenta una soluzione applicabile in un ambiente di produzione reale.

Nelle prossime sezioni vengono illustrate queste due soluzioni.

4.9.1 L'ambiente di sviluppo: IBM Voice SDK

Per quanto riguarda lo sviluppo ho utilizzato il Voice SDK della IBM[28]. Il software è integrato in WebSphere ma è reperibile separatamente e gratuitamente sul sito IBM ed ha le seguenti caratteristiche principali:

- interprete VoiceXML v1.0
- riconoscimento e sintesi vocale IBM
- logging dell'esecuzione che consente di verificare la correttezza dell'interfaccia
- simulatore della tastiera del telefono per testare l'uso dei toni DTMF
- supporto multilingue tra cui anche l'italiano

Il browser si comporta come un normale client HTTP, pertanto può utilizzare una qualsiasi URL che restituisca un file VoiceXML. Accetta e reinvia al server i cookie consentendo così il mantenimento delle sessioni

¹¹TTS: Text-To-Speech

¹²ASR: Automatic Speech Recognition

o di altre informazioni passate dal server.

La qualità della sintesi vocale non è molto elevata e la voce risulta essere robotica e piuttosto monotona sebbene assolutamente comprensibile¹³.

Anche per questo motivo è opportuno che i testi vengano redatti utilizzando frasi brevi, termini semplici e adottando una corretta punteggiatura che consenta al sintetizzatore di “prendere fiato” in modo che l’ascolto risulti agevolato.

Il motore di riconoscimento vocale è decisamente valido e consente spesso un corretto funzionamento anche in caso di ambienti rumorosi nel quale l’utente opera.

Il problema del riconoscimento comunque dipende molto dall’interfaccia vocale creata e in particolare dalle grammatiche. Il motore infatti confronta i suoni prodotti dall’utente con dei pattern sonori legati alle grammatiche definite dallo sviluppatore per stabilire cosa ha detto e trasformare i suoni ricevuti in testo.

Il browser non supporta a pieno tutti i tag VoiceXML 1.0 e non sempre effettua una corretta validazione dei documenti ricevuti: ad esempio non considera come errore l’uso di attributi non definiti nel DTD pertanto può considerare valido un documento che in realtà non lo è.

4.9.2 L’ambiente di produzione: Loquendo VoxNauta

Per verificare l’applicazione in un ambiente di produzione più realistico ho utilizzato VoxNauta di Loquendo che a oggi rappresenta il miglior prodotto sul mercato, almeno per quel che riguarda la lingua italiana.

Si tratta di una piattaforma professionale per la distribuzione di inter-

¹³Il sintetizzatore vocale standard produce un timbro di voce del tutto simile a quello di Eloquence di default da JAWS

facce vocali fruibili attraverso il telefono. La figura 4.6 illustra l'architettura.

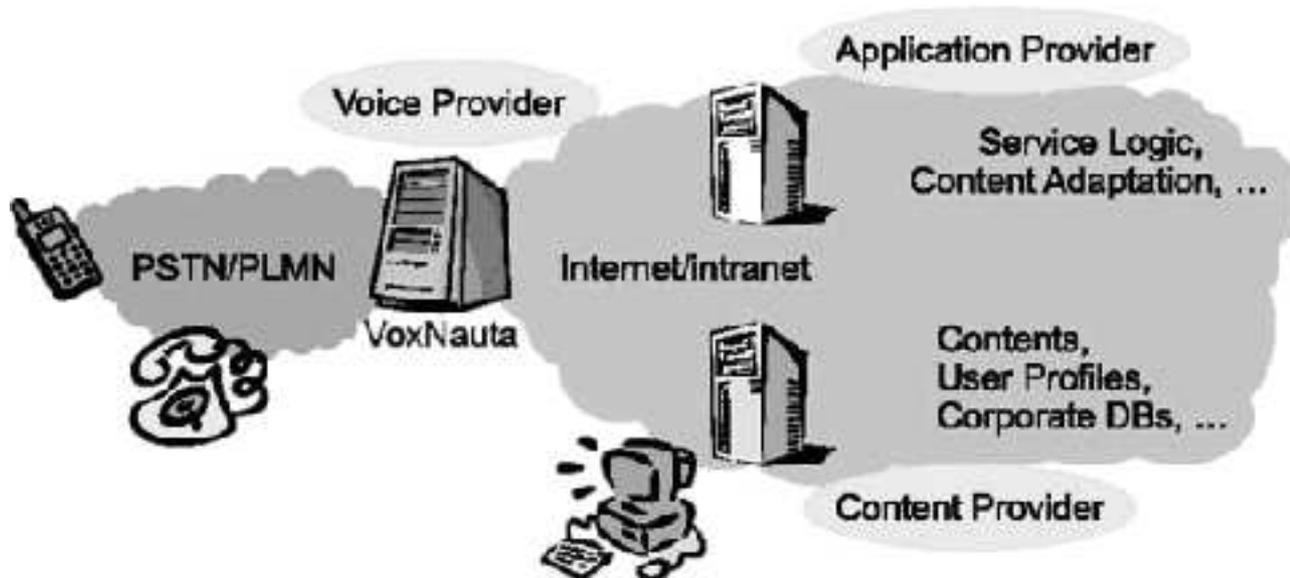


Figura 4.6: VoxNauta: il voice gateway di Loquendo

I contenuti (documenti VoiceXML) vengono prelevati dalla rete dal voice gateway che si occupa dell'interazione con l'utente finale e gestisce la chiamata telefonica.

Le caratteristiche principali sono:

- Supporto VoiceXML versione 2.0 e SSML 1.0[20]
- Sintesi vocale Loquendo Actor. Si tratta del sintetizzatore vocale attualmente più avanzato. I dialoghi prodotti sono realizzati mediante l'opportuna concatenazione di fonemi registrati da attori, il che rende la sintesi estremamente gradevole ed efficace tanto da renderla a volte indistinguibile dalla voce umana. Recentemente sono stati sviluppati delle estensioni del linguaggio che, interpretate dal VoxNauta, consentono di modificare il tono di lettura in modo da esprimere

in modo ancora più efficace gioia, tristezza, rabbia ecc... oltre che migliorare la prosodia delle frasi ¹⁴

- Numerose voci maschili e femminili e supporto multilingua. Una volta selezionata una lingua, il sintetizzatore è in grado di leggere con apprezzabile precisione, testi scritti in altre lingue utilizzando opportune estensioni fornite dalla piattaforma, senza dover cambiare la voce.
- Ambiente di sviluppo e di debug che consente allo sviluppatore di testare i propri documenti VoiceXML prodotti (anche dinamicamente da un application server) e verificare il funzionamento in una situazione reale, quindi con accesso telefonico. Il sito all'indirizzo <http://loquendocafe.com/> consente, previa registrazione, di caricare sul sistema i propri documenti VoiceXML ed effettuare il debug attraverso una applet java[5].

¹⁴Variazione nell'intonazione della voce durante la lettura delle frasi interrogative, imperative e in generale in rispondenza alla punteggiatura usata.

Capitolo 5

Valutazione del modello del sistema realizzato

Un sito è accessibile se risulta essere usabile dalle categorie di utenti per le quali è stato realizzato e se tutte le informazioni utili contenute nelle pagine sono da essi fruibili.

Non ritengo che si debbano separare in modo netto i concetti di usabilità e di accessibilità e che entrambi siano aspetti fondamentalmente soggettivi della qualità di un sito web e in generale di una applicazione.

Per questo motivo per valutare l'accessibilità di un sito è indispensabile coinvolgere in qualche modo gli utenti nell'analisi, o almeno provare a simularne il comportamento e i contesti di utilizzo.

Il risultato che si ottiene è una valutazione qualitativa del grado di accessibilità del sito che consente di correggere eventuali difetti riscontrati o di migliorare il livello raggiunto.

La verifica dell'accessibilità non deve essere un processo messo in moto durante lo sviluppo o al termine della realizzazione del sito, ma deve essere una attività permanente svolta durante tutto l'arco di vita dell'applicazione. È necessaria una verifica continua in relazione all'evoluzione del sito, dei suoi contenuti e soprattutto allo sviluppo delle tecnologie assistive e dei dispositivi potenzialmente utilizzabili dagli utenti.

Per l'analisi dei criteri di accessibilità descritti in questa tesi e in particolare dell'accesso multimodale alle informazioni, è stato utilizzato il progetto Multiabile come caso pratico di studio.

5.1 Valutazione soggettiva

L'utilizzo dei tool di validazione automatica dell'accessibilità (Bobby, Torquemada, Cinthia ecc...) devono essere considerati solo un punto di partenza in quanto verificano solo i requisiti tecnici oggettivi ma non garantiscono in alcun modo il raggiungimento dell'accessibilità multimodale.

Peraltro, molti dei requisiti esposti dalle linee guida richiedono l'intervento umano per poter essere verificati, pertanto a maggior ragione è necessario affiancare a questi strumenti una metodologia di valutazione efficace di tipo soggettivo.

Tralasciando quindi i risultati, peraltro positivi, ottenuti tramite la verifica tecnica automatica, verranno presi in esame esclusivamente quelli della verifica soggettiva.

5.1.1 Metodologie di analisi

La valutazione soggettiva di accessibilità deve servire a verificare oltre alla semplice fruibilità e comprensione dei contenuti, anche l'usabilità del sito, la navigabilità, la consapevolezza dell'utente rispetto a dove si trova, a cosa sta facendo e a come raggiungere determinati obiettivi e reperire le informazioni.

Per fare ciò sono state adottate diverse metodologie di analisi, prendendo spunto dalla letteratura, che vengono qui di seguito brevemente illustrate

Valutazione euristica : si tratta di applicare linee guida di usabilità basate sull'esperienza e sulla buona pratica. Il problema principale è quello che a volte è necessario calarsi nelle condizioni d'uso dell'u-

tente, il che è a volte difficile specie nel caso in cui l'utente tipo è affetto da qualche disabilità. Ad esempio, non basta bendarsi gli occhi per simulare il modo in cui una persona cieca usa il PC; non è possibile simulare la disabilità cognitiva di una persona dislessica, ecc. . .

Cognitive Walkthrough : il metodo consiste nel porre alcune domande relativamente a quello che il sito dovrebbe offrire, in termini di contenuti e funzionalità, e verificare se e come sia possibile rispondere ai quesiti attraverso la navigazione nel sito stesso.

Navigazione assistita : un supervisore assegna dei compiti ad un gruppo di utenti registrandone il successo o l'insuccesso, il tempo impiegato, il metodo usato, le funzionalità delle pagine e dei dispositivi di ausilio sfruttate ecc. . .

Thinking aloud : l'utente descrive quello che sta facendo durante la navigazione per raggiungere l'obiettivo assegnato o commenta le informazioni che trova.

In realtà questo metodo pone l'utente in una situazione anomala in quanto lo porta a fare ciò che non è consono a chi naviga sul web, cioè parlare e descrivere le proprie operazioni. In effetti il metodo usato è stato quello di chiedere agli utenti di cercare di memorizzare le proprie azioni e fare dei commenti solo al termine delle singole prove.

Nella prima fase della verifica sono stati adottati i primi due metodi che sono quelli che coinvolgono principalmente gli esperti, in questo caso gli stessi sviluppatori del sistema.

Successivamente sono stati applicati gli altri due metodi di analisi coinvolgendo anche utenti esterni.

In ogni caso sono state condotte verifiche separate per quanto riguarda l'accesso mediante interfaccia grafica, interfaccia vocale e interfaccia multimodale. In tutti i casi, oltre alla verifica di accessibilità e usabilità

sono state raccolte le impressioni degli utenti che sono di seguito illustrate assieme ad alcune considerazioni di carattere generale.

5.2 L'interfaccia grafica

L'utilizzo del sito da parte degli utenti conferma sostanzialmente i risultati ottenuti mediante la verifica automatica di accessibilità e la bontà delle tecniche adottate. L'applicazione delle linee guida esposte è quindi un requisito essenziale per ottenere una interfaccia web accessibile e usabile.

Il sito è stato provato con diversi browser (Internet Explorer, Mozilla Firefox, Opera ecc...) e diversi dispositivi (PC, tablet, PDA) ottenendo sempre una ottima adattabilità dell'interfaccia grafica e una elevata usabilità.

Per provare diversi aspetti di accessibilità e di usabilità si è deciso di navigare l'intero sito attraverso lo screen reader (JAWS) e un display braille¹.

L'assenza delle tabelle, ed una struttura delle pagine praticamente identica in tutto il sito ha favorito la navigazione da parte degli utenti, anche coloro che non avevano ancora mai avuto modo di provare il sistema. La serializzazione delle pagine risulta essere quindi sempre molto semplice e consente all'utente di fruire dei contenuti senza perdere l'orientamento.

L'uso di testi e link nascosti consente di dare all'utente che usa lo screen reader utili informazioni circa la posizione in cui si trova all'interno della pagina, oltre che strumenti di navigazione interni alla pagina utili se non si può vedere la pagina.

L'utilizzo di testi semplificati, misurati in modo rigoroso attraverso l'indice di complessità GulpEASE, e la cura nella struttura delle frasi, rendono la sintesi vocale di JAWS molto più chiara rispetto alle parti in cui questo lavoro di semplificazione non è stato fatto.

¹Il display braille a disposizione era il "PACMate 40" della Freedom Scientific

5.3 L'interfaccia vocale

Per quanto riguarda l'interfaccia puramente vocale si fa riferimento a quella fruita attraverso il telefono con il quale quindi i soli input sono la voce umana e i segnali DTMF e come output si ha solo una voce sintetizzata e suoni digitali.

Come detto, per la realizzazione di tale interfaccia, l'aspetto principale tenuto in considerazione è stato quello dell'usabilità. La scomposizione delle pagine in primo luogo nei suoi componenti principali e successivamente in componenti atomici più piccoli (paragrafi, elementi delle liste, celle delle tabelle ecc...) è stato senza dubbio un elemento fondamentale per ottenere tale obiettivo.

L'utilizzo delle annotazioni semantiche ha consentito poi di accedere alle informazioni contenute nelle pagine in modo estremamente agevole. Confrontando la trascodifica di un normale sito web, senza annotazioni e senza particolare cura dell'accessibilità si nota immediatamente la differenza. Le pagine del sito normale sono a volte ricche di informazioni superflue e spesso richiedono una lunga attesa da parte dell'utente e una ricerca a tentoni prima di poter arrivare all'ascolto del contenuto principale.

Lo si è verificato ad esempio durante l'accesso alle informazioni ottenute tramite i feed RSS: seguendo il link associato ad un particolare elemento si accede alla pagina della notizia completa opportunamente trascodificata. Il problema è che tale pagina non contiene solo la notizia, ma menu di vari livelli, il banner del sito, un banner pubblicitario ecc..., che vengono letti prima del contenuto principale e non c'è modo di saltarli interamente.

In Multiabile, invece, per ogni nuova pagina si fruisce inizialmente del contenuto principale e a richiesta dell'utente si può accedere agli altri componenti e alla mappa del sito che dà accesso alle altre pagine.

L'interfaccia vocale creata ad-hoc per la piattaforma di e-learning è

stata realizzata secondo gli stessi principi di usabilità e risulta essere ovviamente molto più snella essendo ottenuta direttamente trasformando i contenuti da XML a VoiceXML, pertanto eliminando a priori ogni elemento di “disturbo”.

In entrambe i casi, l’uso dell’interfaccia è risultata semplice sia per chi conosceva i comandi disponibili, sia per chi si avvicinava per la prima volta alla piattaforma.

Infatti in ogni pagina viene spiegato chiaramente all’utente cosa deve fare ed è sempre presente un testo di aiuto che può essere ascoltato in ogni momento. Poichè i comandi sono semplici, intuitivi, coerenti in tutte le pagine e in numero limitato, risulta abbastanza semplice memorizzarli.

Alcuni utenti preferiscono l’uso dei comandi DTMF, che sono più semplici da usare e non creano problemi di riconoscimento da parte del sistema. Per questo motivo sono sempre presenti a fianco di ogni comando vocale.

Come per lo screen reader, anche attraverso il telefono la sintesi vocale risulta essere molto più chiara e semplice per quei testi di cui si è ottenuto un livello di GulpEASE elevato. L’uso di Voxnauta, il voice gateway di Loquendo, ha consentito inoltre di utilizzare un sintetizzatore vocale di elevata qualità e anche gradevole da ascoltare, un requisito importante richiesto da coloro che hanno problemi di comprensione del linguaggio (dislessici) e un po’ meno sentito invece dai non vedenti, abituati alla sintesi degli screen reader², ma che comunque mostrano di apprezzare la qualità raggiunta.

L’unica perplessità che sorge da parte degli utenti sta nel fatto di dover ascoltare lunghi brani attraverso il telefono. In effetti, il sistema di sintesi e riconoscimento vocale attraverso il telefono sembra pensato principalmente per ottenere brevi informazioni, per accedere a servizi in cui

²Per la verità, alcune versioni più recenti di JAWS consentono di utilizzare lo stesso sintetizzatore di Loquendo (Actor), il quale però richiede una potenza di calcolo piuttosto elevata pertanto il più delle volte non viene adottato.

la durata della telefonata sia breve e l'interazione con il sistema sia molto più frequente rispetto a quella che si ha ad esempio durante la fruizione di un corso di formazione.

L'aspetto maggiormente gradito da parte degli utenti è risultato essere quello dell'accesso alle informazioni ottenute attraverso feed RSS, data la brevità dei testi e il continuo aggiornamento delle notizie.

In questo caso l'uso del telefono per l'accesso alle informazioni sembra essere una valida alternativa a quella del PC e dell'interfaccia grafica.

5.4 L'interfaccia multimodale

La valutazione dell'interfaccia multimodale è piuttosto complessa in quanto in essa vengono coinvolte contemporaneamente la grafica, i feedback audio/tattili, i diversi metodi di input.

In generale si è visto che avere tanti modi diversi di dare degli input alla macchina e di ricevere gli output in altrettante modalità è senza dubbio di aiuto a chi in genere incontra difficoltà nell'accesso alle informazioni a causa della propria disabilità e permette di avere interfacce gradevoli, usabili e interessanti anche per coloro che invece non hanno questo tipo di problemi.

Alcune di queste modalità sono pensate specificatamente per consentire a chi è affetto da una disabilità di poter accedere alle informazioni sfruttando canali alternativi.

Nelle prossime sezioni verranno analizzati separatamente gli aspetti principali dell'interfaccia multimodale.

In generale si è verificato che assegnando compiti specifici agli utenti (aprire una determinata pagina di un corso, ottenere l'indirizzo email a cui inviare una richiesta di informazioni ecc. . .), è possibile giungere al risultato utilizzando varie modalità e cambiando la modalità di accesso durante la navigazione. Tuttavia generalmente gli utenti che sono intervenuti nella sperimentazione prediligono modalità tradizionali e anche i non vedenti

abituati ad usare lo screen reader, ne sfruttano tutte le potenzialità senza ritenere necessario cambiare modalità di accesso.

5.4.1 L'esplorazione tattile

L'uso dei feedback vibrazionali è destinato esclusivamente a persone con disabilità visive. Difficile pensare che un utente vedente possa avere qualche vantaggio per la navigazione nell'uso di questo canale di comunicazione.

Le sperimentazioni hanno riguardato sia la navigazione che l'esplorazione di immagini.

Per quanto riguarda la prima, cioè la percezione spaziale di una pagina, gli utenti con difficoltà visive ma con esperienza nella navigazione (assistita da screen reader) reagiscono in modi diversi. Alcuni mostrano una curiosità verso l'uso di questo canale e su come questo tipo di feedback possano aiutare ad immaginare come i contenuti siano organizzati spazialmente, dato che questa informazione non può essere fornita dagli screen reader.

Per molti utenti non vedenti, l'uso del mouse è praticamente sconosciuto, oltre alle difficoltà di collocare inizialmente il mouse, c'è anche il problema di non riuscire a rapportare il movimento della mano con quello del puntatore, data l'accelerazione e la velocità del mouse.

L'utilizzo di altri dispositivi quali il touch screen, la tavoletta grafica, la penna elettronica con il tablet PC, aiutano notevolmente l'esplorazione delle pagine ed in particolare delle immagini.

Per la sperimentazione di questa modalità di accesso, si è verificata la possibilità di riconoscere forme geometriche più o meno complesse attraverso il solo uso di feedback vibrazionali associati alla posizione del puntatore del mouse in corrispondenza del bordo delle immagini.

Per riconoscere forme semplici quali rettangoli, quadrati, rombi ecc. . . , sono necessari mediamente 4-5 minuti, durante i quali gli utenti (esper-

ti e non, disabili e non) esplorano la pagina in diversi modi: andando a zig-zag, tracciando rapidi movimenti in orizzontale o verticale per individuare le dimensioni dell'immagine e i punti per cui passa la figura oppure trovando un punto e cercando di seguirne il contorno.

Si è verificato che il riconoscimento delle figure complesse è pressoché impossibile con un semplice feedback vibrazionale. È necessario raffinare la percezione attraverso feedback audio che diano suggerimenti ad esempio sulle inclinazioni delle linee, sul fatto che ci si trovi su uno spigolo o su un lato ecc... Lo studio di quali siano i riscontri audio più efficaci è ancora tutto da intraprendere e richiede probabilmente ancora molta più sperimentazione da parte degli utenti.

Per quanto concerne l'esplorazione della mappa "audio/tattile", l'aspetto più interessante riguarda il modo in cui diversi utenti si avvicinano alla pagina: l'utente cieco, dopo una prima casuale esplorazione per recuperare i diversi feedback tattili, tende a concentrarsi soprattutto sulle linee di confine delle diverse aree e a seguirle.

Il problema che si ripresenta usando il mouse è l'impossibilità di essere precisi nei movimenti e di non comprendere la distanza tra le parti toccate, a causa della mancanza di coerenza tra il movimento della mano e quello del puntatore.

L'uso di un dispositivo di puntamento più sensibile e preciso risulta quindi indispensabile per sfruttare al meglio questo canale.

5.4.2 Feedback audio

L'uso dei feedback audio è destinato prevalentemente a persone con disabilità visive, ma anche a coloro che hanno difficoltà cognitive quali la dislessia o altre forme di difficoltà di comprensione del linguaggio e delle immagini.

Al momento l'uso di tali segnali è stato affiancato, come detto, all'esplorazione tattile per facilitare il compito di coloro cui viene richiesto

di riconoscere un'immagine oppure di esplorare una figura più complessa come una cartina geografica o la mappa di un edificio ecc. . .

Nell'esplorazione della mappa della Lombardia creata in Multiabile, è stato di fondamentale importanza per convogliare informazioni quali i nomi di città, laghi, fiumi ecc. . . , in modo da evitare che l'esplorazione spaziale rimanga fine a se stessa e di fatto poco utile come canale su cui veicolare contenuti informativi.

La scelta di usare una voce registrata per fornire informazioni è nata proprio dalle osservazioni fatte dagli utenti, che esplorando la mappa non erano in grado di decodificare tutti i feedback vibrazionali inizialmente impostati per indicare zone di pianura, collina, montagna ecc. . . , e che comunque non ricevevano sufficienti informazioni da questo tipo di output.

5.4.3 Uso dei gesti e delle scorciatoie

Il problema principale dell'uso dei gesti e delle scorciatoie da tastiera, è quello di dover comunicare all'utente quali sono quelli esistenti in ciascuna pagina.

In Multiabile si è quindi cercato di realizzare scorciatoie valide un po' in tutto il sito come ad esempio quelle che consentono di accedere alle componenti delle pagine, che come si è visto seguono tutte lo stesso template, oppure quelle per accedere alle varie voci di menu.

Chi conosce già il sito e le scorciatoie messe a disposizione le utilizza spesso quando simula il comportamento dell'utente che usa lo screen reader.

Un non vedente che usa abitualmente JAWS difficilmente sente il bisogno di avere scorciatoie di questo tipo in quanto preferisce di gran lunga l'utilizzo delle funzioni dello screen reader stesso. Tuttavia si è notato che, una volta presa confidenza con quelle che sono le access key principali, a lungo andare gli utenti tendono ad utilizzarle sempre più spesso. Le due

scorciatoie maggiormente sfruttate sono quelle per accedere all'inizio del menu (ALT+M) e quella per accedere al contenuto principale della pagina (ALT+K), che sono comunque individuate da JAWS quando si accede alla funzione che elenca i link della pagina corrente (INS+F7).

L'uso dei gesti non è invece considerato uno strumento idoneo per i non vedenti in quanto introduce una difficoltà in più che è quella di dover utilizzare un dispositivo di puntamento per di più attraverso movimenti abbastanza precisi perchè vengano riconosciuti dal sistema e che pertanto richiede una certa manualità.

Risulta invece uno strumento comodo per chi usa i palmari o i tablet PC nei quali il dispositivo di puntamento è il principale mezzo di input. L'uso della penna elettronica rende inoltre molto più semplice rispetto al mouse il tracciamento dei segni corretti.

Capitolo 6

Conclusioni e sviluppi futuri

Il grado di accessibilità ottenuto con il progetto Multiabile è molto elevato e questo risultato è confermato in primo luogo dagli utenti che hanno avuto modo di utilizzare il sistema e che sono stati coinvolti nella verifica soggettiva.

Per quanto riguarda l'accesso multimodale, il progetto è da considerarsi ancora sperimentale ma ha gettato le basi per una ulteriore analisi e per lo sviluppo di nuove metodologie di interfacciamento e nuovi dispositivi di ausilio alla navigazione sul web.

In riferimento all'esplorazione tattile e all'uso dei feedback vibrazionali in combinata con il canale vocale, ad esempio, si intuisce la potenzialità di questo canale di comunicazione anche se di fatto non si è ancora trovata la strada definitiva per usarlo come veicolo di informazione e come valore aggiunto da dare a coloro che hanno disabilità percettive.

Un discorso analogo può essere fatto per l'uso dei gesti per impartire input senza l'uso della tastiera e quindi adatti a situazioni nelle quali può risultare scomodo l'uso della tastiera (PDA, tablet PC ecc...), mentre si dovrà approfondire l'analisi al fine di trovare un modo di sfruttare questo sistema come ausilio ai disabili.

L'interfaccia vocale è quella maggiormente efficace ed effettivamente utilizzabile da chiunque, anche da coloro che non hanno competenze

informatiche di base per l'uso del computer.

È proprio questo il punto di forza di tale tipo di interfaccia. Si può pensare di spingere ulteriormente su questo canale e anche se in effetti la sperimentazione fatta ha mostrato che gli utenti preferiscono una maggiore interazione con il sistema e quindi dialoghi e testi brevi. È necessario a questo punto un ulteriore coinvolgimento degli utenti per verificare se l'uso del telefono sia un mezzo adatto per la fruizione di un corso di formazione che non può avere questi criteri, oltre a richiedere da parte degli utenti una certa attenzione e concentrazione perchè sia efficace, cosa che attraverso il telefono può essere problematica.

La scelta di utilizzare un metodo rigoroso di valutazione della leggibilità dei testi quale l'indice GulpEASE si è rivelato vincente in quanto ha permesso di constatare che effettivamente un testo leggibile è anche più facilmente "ascoltabile". La sintesi vocale, ottenuta tramite lo screen reader o attraverso il telefono, è risultata essere molto più chiara e comprensibile.

La struttura del transcoder, che consente di trasformare le pagine HTML esistenti in un qualunque altro linguaggio, permetterebbe di realizzare una interfaccia mista (grafica e vocale), ad esempio attraverso XHTML+Voice o SALT, nella quale si possano utilizzare un po' tutti i canali, da quello visivo a quello audio a quello tattile. L'uso delle annotazioni semantiche si rivela indispensabile per poter costruire automaticamente una interfaccia mista efficace e usabile.

6.1 Considerazioni finali

L'idea iniziale della tesi era quella di individuare una linea guida pratica per l'accessibilità che risolvesse i limiti riscontrati in quella definita dal W3C e trovare il modo migliore per aiutare i disabili nel superamento degli ostacoli derivanti dall'accesso alle informazioni tramite il PC.

L'ipotesi fatta era che il modo migliore per raggiungere tale obiettivo

poteva essere l'adattabilità dei contenuti e dell'interfaccia alle caratteristiche e alle capacità degli utenti.

L'ulteriore ipotesi formulata è stata quella secondo cui è possibile raggiungere il più alto livello di accessibilità attraverso la multimodalità, lo sfruttamento di diversi canali di comunicazione e metodologie di input alternative messe a disposizione dell'utente, che può quindi scegliere come accedere alle informazioni.

Le linee guida elaborate sono basate proprio su questi principi e il progetto Multiabile ne è un esempio pratico di applicazione. L'idea alla base del sistema implementato è quella di dare la possibilità all'utente di sfruttare al meglio quello che è in grado di fare e quindi di utilizzare tutte le proprie capacità superando gli eventuali ostacoli dovuti alla propria disabilità.

Per fare ciò è necessario mettere l'utente in condizioni di poter utilizzare canali alternativi di comunicazione e il lavoro descritto in questa tesi illustra appunto come fare.

L'idea di mettere al centro gli utenti nasce dalla considerazione secondo cui l'accessibilità è un concetto fondamentalmente soggettivo ed è legato all'utente, ai contesti, alle condizioni e alle modalità di utilizzo di un sistema. A conclusione del lavoro, anche alla luce dei riscontri diretti avuti personalmente e da parte di alcuni utenti, non posso che confermare tale considerazione.

I risultati ottenuti mostrano infine che la multimodalità può essere effettivamente una chiave per l'accessibilità e per l'usabilità insieme e sono da stimolo per approfondire ulteriormente la ricerca di nuovi sistemi alternativi che vadano in questa stessa direzione.

6.2 Possibili implementazioni future

I criteri di accessibilità esposti sono un requisito fondamentale per ottenere una semplice, per quanto possibile, adattabilità dei contenuti e in partico-

lare per la trasformazione dell'interfaccia tradizionale per il browser, in una interfaccia vocale o multimodale.

Uno degli aspetti cruciali in fase di sviluppo è quello della creazione dei file RDF di descrizione semantica delle pagine e del sito. Tuttavia, nel caso in cui vengano rispettate le linee guida di accessibilità generali esposte, si può pensare di creare in modo automatico tali documenti analizzando semplicemente le pagine.

Nell'implementazione del transcoder proposta si affronta la sfida maggiore che riguarda l'accessibilità multimodale, che è quella dell'accesso vocale. Tuttavia l'applicabilità del sistema non si ferma semplicemente alle interfacce puramente vocali. L'integrazione di altri linguaggi come XForms, EMMA, InkML potrebbe consentire di realizzare sistemi che siano veramente ad accesso multimodale e multicanale e che consentano di utilizzare anche contemporaneamente diversi dispositivi di input e output.

Una delle proposte più interessanti è senza dubbio quella di unire tra loro interfacce grafiche e vocali in un unico linguaggio, come XHTML + VoiceXML[32] proposto al W3C da Opera Software e IBM.

Nel caso dell'applicazione realizzata con Cocoon si tratterebbe in questo caso semplicemente di realizzare uno o più fogli di trasformazione che aggiungano la possibilità di interagire con la pagina dando dei comandi vocali.

Nel caso di utilizzo di XForms per la creazione delle form, sarebbe poi possibile realizzare un perfetto mix di interfacce grafica e vocale.

Un ulteriore miglioramento potrebbe poi essere raggiunto inserendo un meccanismo di gestione degli eventi utilizzando EMMA che consenta di interfacciarsi con qualsiasi tipo di periferica e di interpretare varie tipologie di input dell'utente.

6.2.1 Per lo sviluppatore: strumenti di authoring

Quello dell'accessibilità sarà un requisito sempre più richiesto per lo sviluppo dei siti web, grazie anche alla legislazione italiana ed europea che impongono, anche se non da molto, il rispetto di determinati criteri di formali delle applicazioni da realizzare.

Diventa quindi sempre più importante fare in modo che gli strumenti di authoring e di progettazione dei siti web consentano di assistere gli sviluppatori in modo da semplificare il più possibile la realizzazione di prodotti che tengano conto dell'accessibilità.

L'utilizzo della tecnica delle annotazioni esterne o comunque la necessità di arricchire le pagine con metainformazioni, sono un requisito fondamentale per l'accessibilità multimodale. Può quindi essere molto utile dotare tali strumenti di funzioni che ne semplifichino la realizzazione.

Attraverso lo stesso sistema sarebbe possibile definire le modalità di input, i mouse gesture, i feedback audio/tattili da assegnare ai componenti delle pagine.

6.2.2 Per l'utente: un browser multimodale

Al momento non esistono browser internet che consentano veramente l'accessibilità multimodale delle informazioni in rete e in generale ai siti internet. Pertanto pur rispettando i criteri di accessibilità esposti e sfruttando i nuovi linguaggi basati su XML per l'accesso multimodale non si troverebbero comunque strumenti software idonei a sfruttarli.

Molto lavoro c'è da fare dunque soprattutto dal lato utente per migliorare questi strumenti. Un contributo a questo sviluppo potrebbe essere dato con la realizzazione di appositi plug-in che estendano le funzionalità dei browser inserendo il supporto a diverse modalità di interfacciamento con la macchina.

I documenti "index" e "contents" pensati per l'accessibilità multimodale, estesi con EMMA ed eventuali altri linguaggi per la descrizione di

mappe tattili e vibrazionali potrebbero essere interpretati direttamente dal browser che consenta poi l'interfacciamento con i dispositivi di input e output idonei. In particolare penso a mouse o joystick con force feedback, tavolette grafiche, display braille, tavolette tattili e ancora sistemi di sintesi e riconoscimento vocale, riconoscimento dei gesti definiti lato server, che potrebbero essere integrati direttamente nel browser il quale potrebbe diventare un vero e proprio sistema per l'interfacciamento multimodale.

6.2.3 Un linguaggio formale per i feedback tattili

A conclusione della sperimentazione sull'utilizzo dei feedback vibrazionali e force-feedback iniziata con il progetto Multiabile, nel caso in cui venga identificata una valida modalità di interazione, è auspicabile la formulazione di un linguaggio formale per la specifica degli output che si possono produrre e del modo in cui collegarli agli oggetti di interesse (mappe interattive, immagini, pagine web)

Tale linguaggio costituirebbe una ulteriore estensione dei criteri di accessibilità multimodale e andrebbe usato nella definizione delle metainformazioni da legare alle pagine dei siti web.

Lo studio di questo tipo di interazione è comunque solo all'inizio, è necessario continuare la sperimentazione per individuare in quali situazioni di disabilità possa essere d'aiuto e con quali dispositivi applicarla.

Bibliografia

- [1] ADL. Advanced Distributed Learning.
<http://www.adlnet.org>.
- [2] Apple Accessibility.
<http://www.apple.com/accessibility/>.
- [3] Diodati. <http://www.diodati.org>.
- [4] ICF, International Classification of Functioning, Disability and Health. <http://www3.who.int/icf/icftemplate.cfm>.
- [5] LoquendoCafè. <http://www.loquendocafe.com/>.
- [6] RDF Site Summary (RSS) 1.0. <http://web.resource.org/rss/1.0/>.
- [7] SALT Forum. <http://www.saltforum.org/>.
- [8] WebRatio. <http://www.webratio.com>.
- [9] AICC. AGR 006 - Computer Managed Instruction.
- [10] Anita W. Huang e Neel Sundaresan. Aurora: a conceptual model for Web-content adaptation to support the universal usability of Web-based services.
- [11] Apache Group. The Apache Cocoon Project.
<http://cocoon.apache.org/>.
- [12] Apache Group. Apache Jakarta Tomcat.
<http://jakarta.apache.org/tomcat/>.
- [13] Apache Group. HTTP Client.
<http://jakarta.apache.org/commons/httpclient/>.

- [14] T. Barbieri, A. Bianchi e L. Sbattella. Multimodal communication for vision and hearing impairments. *CVHI04 Granada (Spain)*, lug. 2004.
- [15] T. R. C. Ben Caldwell, W. Wendy Chisholm, T. R. C. Gregg Vanderheiden e U. o. M. Jason White. Web Content Accessibility Guidelines 2.0. <http://www.w3.org/TR/WCAG20/>, 19 nov. 2004.
- [16] Benot LEMAIRE e Philippe DESSUS. Modèles cognitifs issus de l'analyse de la sémantique latente. *Laboratoire des sciences de la education, Bt SHM, Universit Pierre-Mends-France.*
- [17] Chieko Asakawa e Hironobu Takagi. *Annotation-based transcoding for nonvisual web access*. No. 172-179. ACM Press, 2000.
- [18] Dan Connolly, Frank van Harmelen, Ian Horrocks, Deborah L. McGuinness, Peter F. Patel-Schneider e Lynn Andrea Stein. DAML+OIL (March 2001) Reference Description. <http://www.w3.org/TR/2001/NOTE-daml+oil-reference-20011218>.
- [19] Dan R. Olsen Jr., Sean Jefferies, Travis Nielsen, William Moyes e Paul Fredrickson. Cross-modal Interaction using XWeb.
- [20] Daniel C. Burnett, Mark R. Walker e Andrew Hunt. Speech Synthesis Markup Language (SSML) Version 1.0. <http://www.w3.org/TR/2004/REC-speech-synthesis-20040907/>, 7 set. 2004.
- [21] Dave Beckett e Brian McBride. RDF/XML Syntax Specification (Revised). <http://www.w3.org/TR/rdf-syntax-grammar/>.
- [22] B. Dave Offen e F. S. Beth Thomlinson. Good Vibrations: Using a Tactile Mouse to Convey Page Layout Information to Visually Impaired Computer Users. *CSUN 2001*, 2001.
- [23] Dave Raggett. Clean up your Web pages with HTML TIDY. <http://www.w3.org/People/Raggett/tidy/>.

- [24] Davide Ceriani. *Applicazione di tecniche e strumenti di comunicazione multimodale e di valutazione della leggibilità e comprensibilità di contenuti web*. Tesi di laurea, ArcsLab, Politecnico di Milano, 2003.
- [25] Deng Cai, Shipeng Yu, Ji-Rong Wen e Wei-Ying Ma. Extracting Content Structure for Web Pages based on Visual Representation.
- [26] GNOME. The GNOME Accessibility Project. <http://developer.gnome.org/projects/gap/>.
- [27] IBM. IBM Multimodal. <http://www-306.ibm.com/software/pervasive/multimodal/>.
- [28] IBM. IBM VoiceXML SDK. <http://www-3.ibm.com/pvc/products/voice/>.
- [29] James A. Larson, T.V. Raman, Dave Raggett e Altri. W3C Multimodal Interaction Framework. <http://www.w3.org/TR/mmi-framework/>.
- [30] James Clark e Steve DeRose. XML Path Language (XPath) Version 1.0. <http://www.w3.org/TR/1999/REC-xpath-19991116>.
- [31] Jeff Heflin e James Hendler. Semantic Interoperability on the Web.
- [32] O. S. Jonny Axelsson, I. Chris Cross, M. Jim Ferrans, I. Gerald McCobb, I. T. V. Raman e I. Les Wilson. XHTML+Voice Profile 1.2. <http://www.voicexml.org/specs/multimodal/x+v/12/spec.html>, 16 mar. 2004.
- [33] Judy Brewer. How People with Disabilities Use the Web. <http://www.w3.org/WAI/EO/Drafts/PWD-Use-Web/>.
- [34] Katashi Nagao. *Digital content annotation and transcoding*. Artech House, inc.
- [35] KDE. KDE Accessibility Project. <http://accessibility.kde.org/>.

-
- [36] F. T. Keith Waters, I. Rafah A. Hosn, W. Dave Raggett e N. Sailesh Sathish. Dynamic Properties Framework (DPF). <http://www.w3.org/TR/DPF/>.
- [37] Leah M. Reeves, Jennifer Lai, James A. Larson, Sharon Oviatt e Altri. Guidelines for multimodal user interface design. *COMMUNICATIONS OF THE ACM*, gen. 2004.
- [38] Li Deng e Xuedong Huang. Challenges in adopting Speech Recognition. *COMMUNICATIONS OF THE ACM*, gen. 2004.
- [39] Manuel A. Pèrez-Quiñones, Natasha Dannenberg e Robert Capra. Voice Navigation of Structured Web Spaces. *Department of Computer Science, Virginia Tech*.
- [40] Masahiro Hori, Rakesh Mohan, Hiroshi Maruyama e Sandeep Singhal. Annotation of Web Content for Transcoding. <http://www.w3.org/TR/annot/>.
- [41] Micah Dubinko, J. Leigh L. Klotz, Roland Merrick e T. V. Raman. XForms 1.0. <http://www.w3c.org/TR/xforms/>.
- [42] Michael K Brown, Stephen C. Glinski e Brian C. Schmult. Web page analysis for voice browsing. 2001.
- [43] Michael Wynblatt, Dan Benson e Arding Hsu. Browsing the World Wide Web in a Non-Visual Environment.
- [44] Minsu Jang, Jaehong Kim e Joo-Chan Sohn. Web Content Adaptation and Transcoding based on CC/PP and Semantic Templates. 2003.
- [45] Mir Farooq Ali e Manuel A. Pèrez-Quiñones. Using Task Models to Generate Multi-Platform User Interfaces While Ensuring Usability. 2001.
- [46] Mir Farooq Ali, Manuel A. Pèrez-Quiñones, Marc Abrams e Eric Shell. *Building multi-platform user interfaces with UIML*. Harmonia Inc., 2001.

- [47] E. D. Mynatt e G. Weber. Nonvisual presentation of graphical user interfaces: Contrasting two approaches. pp. 166–172.
- [48] Nichelle Hopson (IBM). WebSphere Transcoding Publisher.
- [49] Orkut Buyukkokten, Hector Garcia-Molina e Andreas Paepcke. Accordion summarization for end-game browsing on PDAs and cellular phones. *CHI 2001*, pp. 213–220, 2001.
- [50] Robbie Schaefer, Andreas Dangberg e Wolfgang Mueller. Fuzzy Rules for HTML Transcoding. Rap. tecn., German Ministry for Education and Research (BMBF), 2002.
- [51] S. Rollins e N. Sundaresan. AVoN calling: AXL for voice-enabled Web navigation. *Computer Networks (Amsterdam, Netherlands: 1999)*, vol. 33(1–6):pp. 533–551, 2000.
- [52] Saikat Mukherjee, Guizhen Yang, Wenfang Tan e I.V. Ramakrishnan. Automatic Discovery of Semantic Structures in HTML Documents.
- [53] Scott McGlashan et al. Voice Extensible Markup Language (VoiceXML). <http://www.w3.org/tr/voicexml20/>.
- [54] Sean Bechhofer, Frank van Harmelen, Jim Hendler, Ian Horrocks, Deborah L. McGuinness, Peter F. Patel-Schneider e Lynn Andrea Stein. OWL Web Ontology Language. <http://www.w3.org/TR/owl-ref/>.
- [55] Stéphane H. Maes e Vijay Saraswat. Multimodal Interaction Requirements. <http://www.w3.org/TR/mmi-reqs/>.
- [56] Steve DeRose, Eve Maler e Ron Daniel Jr. XML Pointer Language (XPointer) Version 1.0. <http://www.w3.org/TR/2001/WD-xptr-20010108>.
- [57] T. V. Raman. X+V 1.1 Versatile Multimodal Solutions. *XML Journal*, apr. 2003.

- [58] Thimoty Barbieri. METODO E DISPOSITIVO PER L'ESPLORAZIONE VIA WEB DI IMMAGINI BIDIMENSIONALI PER NON VEDENTI. (domanda di brevetto), 2005.
- [59] Thimoty Barbieri, Antonio Bianchi e Licia Sbattella. Minus Two: Multimedia, Sound Spatialization and 3D Representation for Cognitively Impaired Children. *ICCHP04 Paris (France)*, giu. 2004.
- [60] Thimoty Barbieri e Licia Sbattella. Virtual Sounds and Virtual Spaces to support Persons with Cognitive and Physical disabilities during Information Retrieval, Rehabilitation and Learning. *AAATE 2003*, apr. 2003.
- [61] Thimoty Barbieri e Licia Sbattella. MultiAbile: a multimodal learning environment for the inclusion of impaired e-Learners using tactile feedbacks, voice, gesturing, and text simplification. *AAATE 2005 Lille (France)*, 2005.
- [62] Tullio De Mauro. *Guida all'uso delle parole*. 1997.
- [63] T.V. Raman. User interface principle for multimodal interaction. *CHI 2003*.
- [64] T.V. Raman. Emacspeak Toward The Speech-enabled Semantic WWW. 2 mar. 2001.
- [65] W3C. Cascading Style Sheets (CSS).
<http://www.w3.org/Style/CSS/>.
- [66] W3C. Extensible Markup Language (XML).
<http://www.w3.org/XML/>.
- [67] W3C. Semantic Web Activity.
<http://www.w3.org/2001/sw/>.
- [68] W3C. W3C Multimodal Interaction Activity.
<http://www.w3c.org/>.

-
- [69] W3C. XHTML™1.0 The Extensible HyperText Markup Language (Second Edition).
<http://www.w3.org/TR/xhtml1/>.
- [70] W3C. XSL Transformations (XSLT) Version 1.0.
<http://www.w3.org/TR/xslt>.
- [71] T. R. . D. C. Wendy Chisholm, T. R. . D. C. Gregg Vanderheiden e W. Ian Jacobs. Web Content Accessibility Guidelines 1.0.
<http://www.w3.org/TR/WAI-WEBCONTENT>, 5 mag. 1999.
- [72] W. Wendy Chisholm, T. R. . D. C. Gregg Vanderheiden e W. Ian Jacobs. Techniques for Web Content Accessibility Guidelines 1.0.
<http://www.w3.org/TR/WCAG10-TECHS/>, 6 nov. 2000.
- [73] Wu Chou, Deborah A. Dahl, Michael Johnston, Roberto Pieraccini e Dave Raggett. EMMA: Extensible MultiModal Annotation markup language. <http://www.w3.org/TR/emma/>.
- [74] Yi-Min Chee, Jose-Antonio Magaña, Katrin Franke, Max Froumentin, Gregory Russell, Sriganesh Madhvanath, Giovanni Seni e Larry Yaeger. Ink Markup Language. <http://www.w3.org/TR/2004/WD-InkML-20040928/>.
- [75] Zhiyan Shao, Robert G. Capra e Manuel A. Pèrez-Quiñones. Transcoding HTML to VoiceXML Using Annotation. *Department of Computer Science Virginia Tech Blacksburg, VA*, 2003.