

ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA

---

FACOLTÀ DI SCIENZE MATEMATICHE, FISICHE E NATURALI  
Corso di Laurea Magistrale in Informatica

**PROGETTO E REALIZZAZIONE  
DI UN SISTEMA DI  
AUGMENT BROWSING  
PER L'ACCESSIBILITÀ DEL WEB**

Tesi di Laurea in Basi di dati e sistemi informativi

**Relatore:**  
Chiar.ma Dott.ssa  
SILVIA MIRRI

**Presentata da:**  
CATIA PRANDI

**Correlatore:**  
Chiar.ma Prof.ssa  
PAOLA SALOMONI

**Sessione II  
2010/2011**



*A mia nonna Maria  
che non si è mai dimenticata  
di chiedermi com'era andato  
un esame, con quella generosità  
che solo lei sapeva esprimere.*



# Introduzione

In questi ultimi anni si è assistito alla nascita e alla diffusione della filosofia del Web 2.0 in cui l'enfasi è sull'utente e sulla sua partecipazione nella creazione di contenuti in modo interattivo. Ciò può presentare una grande possibilità di inclusione per le persone con disabilità nella società dal momento che il Web viene oggi utilizzato in tutti gli ambiti della vita quotidiana. Per far sì che l'inclusione sia possibile le pagine Web devono essere progettate e sviluppate in modo accessibile.

Le persone con disabilità hanno esigenze specifiche e hanno bisogno di accorgimenti per poter usufruire del contenuto delle pagine Web. Sono varie le disabilità su cui l'accessibilità del Web influisce: motorie, sensoriali (come cecità, sordità e ipovisione) e cognitive (disturbi specifici dell'apprendimento e del linguaggio).

Ogni stato emana le proprie leggi in materia di accessibilità, soprattutto per regolamentare le tecnologie e i servizi informatici utilizzati dalle pubbliche amministrazioni. Un documento molto importante nel quale sono descritte le linee guida da seguire per creare contenuti e applicazioni multimediali accessibili, prodotto dal W3C, è la recommendation WCAG (Web Content Accessibility Guidelines), arrivata alla versione 2.0.

Lo sviluppo del Web 2.0 ha portato anche a nuove sfide per l'accessibilità, sia per l'uso massiccio di componenti multimediali e RIA (Rich Internet Application) che per le tecnologie con cui le pagine Web vengono create, diventando vere e proprie applicazioni Web. Un esempio di queste tecnologie ampiamente utilizzate è AJAX (Asynchronous JavaScript and XML),

basato su JavaScript, che permette di aggiornare porzioni di pagine Web senza richiedere l'intero contenuto e ciò provoca problemi nella navigazione attraverso screen reader. Per questo sono iniziati studi riguardanti la codifica e la transcodifica dei contenuti di applicazioni e pagine Web. Nonostante gli sforzi, questi sistemi presentano limitazioni che li rendono non adatti come strumento per migliorare l'accessibilità: hanno un set limitato di contenuti disponibili che quindi non possono soddisfare tutte le necessità degli utenti; inoltre non permettono di configurare profili in cui gli utenti possano esprimere le loro preferenze.

In questo documento di tesi viene descritta la progettazione e la realizzazione di GAPforAPE, un sistema di augment browsing per migliorare l'accessibilità delle pagine Web adattando i contenuti in base alle necessità degli utenti espresse tramite un profilo configurato direttamente dall'utente stesso. Questo sistema è fornito tramite un'estensione disponibile per il browser Mozilla Firefox. L'idea che ha portato al suo sviluppo è stato "il contenuto migliore per ogni utente", infatti grazie al sistema di transcodifica e adattamento dei contenuti la stessa pagina Web verrà fornita ad ogni utente nel modo che più rispecchia le sue necessità, in base a ciò che lui ha scelto configurando il proprio profilo. Quest'ultimo descrive le necessità ed esigenze dell'utente attraverso la categorizzazione utilizzata nello standard IMS ACCLIP.

L'idea principale del sistema è quella di utilizzare linguaggi di scripting a lato client per modificare e adattare il contenuto e la presentazione delle pagine Web, meccanismo che è alla base dall'estensione per Firefox GreaseMonkey e successivamente del framework AccessMonkey. Questi script permettono di modificare il DOM HTML, le regole CSS, e gli stessi script utilizzati nelle pagine Web.

Una delle importanti differenze tra il nostro sistema e le altre estensioni che permettono di eseguire user script, come GreaseMonkey, è l'utilizzo di un sistema di profilazione che permette di trasformare i bisogni e le necessità degli utenti in adattamenti ai contenuti delle pagine Web. Non è sem-

pre possibile però migliorare tutte le pagine Web utilizzando tecniche non flessibili, in modo particolare quando utilizzano AJAX. Per questo il nostro sistema permette anche di riconoscere le applicazioni Web ed applicare user script specifici, se disponibili, con lo stesso meccanismo utilizzato dagli screen reader più diffusi, come Jaws, nelle applicazioni desktop.

La nostra estensione permette anche di modificare on-the-fly la pagina corrente in modo semplice e intuitivo e di salvarne le modifiche permanentemente così da visualizzare la pagina sempre con il contenuto adattato.

Grazie ai vantaggi descritti, GAPforAPE permette di migliorare l'accessibilità delle pagine Web, in base alle specifiche esigenze e necessità di ciascun utente, adattando i contenuti e la presentazione in accordo al profilo e alle preferenze dell'utente.

Il contributo di questa tesi è stato presentato alla conferenza W4A (Web For All) 2011, "8th International Cross-Disciplinary Conference on Web Accessibility", tenutasi il 28 e 29 marzo scorso in Hyderabad (India); nella quale il paper ha anche ottenuto un importante riconoscimento: "2011 Best Paper Award".

Come sviluppi futuri si potrebbe implementare l'estensione per gli altri browser più diffusi come Internet Explorer e Google Chrome, in modo da rendere il sistema disponibile ad un maggior numero di utenti.

Questo documento di tesi è organizzato nel seguente modo. Nel primo capitolo viene descritto cosa si intende per accessibilità del Web, quali sono le persone che ne beneficiano e quali sono le normative vigenti sull'argomento. Dopo questa prima panoramica, ci si sofferma sul concetto di Web 2.0 e su come esso abbia introdotto nuove sfide per l'accessibilità ma anche nuove opportunità di inclusione per le persone con disabilità. Nel secondo capitolo vengono analizzate una per una le specifiche richieste dal sistema e spiegate le architetture che si è deciso di implementare. L'ultimo capitolo riguarda l'implementazione vera e propria di GAPforAPE ed un caso d'uso relativo ad una persona ipovedente che crea il proprio profilo per rispondere

alle sue necessità ed adattare così i contenuti di tutte le pagine Web. In particolare, vengono mostrate immagini riguardanti l'interfaccia del social network Google+ adattata secondo il profilo dell'utente. Il documento di tesi si conclude poi con le conclusioni e gli sviluppi futuri.



# Indice

<b>Introduzione</b>	<b>i</b>
<b>1 Stato dell'arte</b>	<b>1</b>
1.1 Accessibilità dei siti Web . . . . .	2
1.1.1 Disabilità e tecnologie assistive . . . . .	2
1.1.2 Accessibilità e divario digitale . . . . .	10
1.1.3 Accessibilità ed usabilità . . . . .	11
1.1.4 Norme e linee guida . . . . .	12
1.1.5 WAI . . . . .	18
1.2 Il Web 2.0 . . . . .	21
1.2.1 Tecnologie del Web 2.0 . . . . .	22
1.3 L'accessibilità nel Web 2.0 . . . . .	23
1.3.1 ARIA . . . . .	25
1.4 Adattamento e transcodifica dei contenuti . . . . .	26
1.4.1 Profilazione . . . . .	27
1.5 Augment browsing e Crowdsourcing . . . . .	29
1.5.1 GreaseMonkey . . . . .	31
1.5.2 AccessMonkey . . . . .	32
1.5.3 Web Visum . . . . .	33
1.5.4 Web Adaptation Technology . . . . .	34
1.5.5 Farfalla Project . . . . .	35
<b>2 Progetto</b>	<b>37</b>
2.1 Specifiche e scelte progettuali . . . . .	38

---

2.1.1	Profilazione . . . . .	38
2.1.2	Adattamento e transcodifica dei contenuti . . . . .	39
2.1.3	Augment browsing . . . . .	44
2.1.4	Crowdsourcing . . . . .	49
2.2	Architettura . . . . .	51
2.3	Tecnologie utilizzate . . . . .	53
2.3.1	Add-on SDK . . . . .	54
2.3.2	HTML . . . . .	57
2.3.3	CSS . . . . .	57
2.3.4	JQuery . . . . .	58
<b>3</b>	<b>Implementazione</b>	<b>59</b>
3.1	Tipi di script . . . . .	60
3.1.1	Add-on code . . . . .	60
3.1.2	Content script . . . . .	63
3.1.3	Add-on code vs Content script . . . . .	64
3.2	Struttura . . . . .	64
3.2.1	Scheletro . . . . .	64
3.2.2	Cartella <code>lib</code> . . . . .	67
3.2.3	Cartella <code>data</code> . . . . .	74
3.3	Widget . . . . .	75
3.4	Profilo . . . . .	75
3.4.1	Colori . . . . .	77
3.4.2	Testo . . . . .	80
3.4.3	Visuale . . . . .	80
3.5	Modifiche on-the-fly sulla pagina corrente . . . . .	86
3.6	Caso d'uso . . . . .	88
	<b>Conclusioni</b>	<b>95</b>
	<b>Bibliografia</b>	<b>105</b>

# Elenco delle figure

1.1	Interazione delle varie linee guida del WAI (disponibile da [1])	18
1.2	Architettura AccessMonkey . . . . .	34
1.3	Architettura Farfalla Project . . . . .	36
2.1	Adattamento dei contenuti client-side (disponibile in [2]) . . .	41
2.2	Adattamento dei contenuti server-side (disponibile in [2]) . . .	42
2.3	Adattamento dei contenuti proxy-based (disponibile in [2]) . .	43
2.4	Adattamento dei contenuti service-oriented (disponibile in [2])	43
2.5	Diffusione dei vari browser secondo Wikimedia nel settembre 2011 (disponibile in [3]) . . . . .	47
2.6	Architettura di GAPforAPE (disponibile in [4]) . . . . .	52
3.1	Esempio di utilizzo dei moduli CommonJS (disponibile da [5])	61
3.2	Dipendenze tra il package del add-on <code>new-addon</code> e il package <code>add-on kit</code> del SDK (disponibile a [5]) . . . . .	63
3.3	Dipendenze tra il package del nostro sistema <code>gapforape</code> e i package del SDK . . . . .	65
3.4	Dipendenze tra i moduli del package <code>gapforape</code> . . . . .	67
3.5	Dipendenze tra il modulo <code>main</code> e gli altri moduli . . . . .	68
3.6	Dipendenze tra il modulo <code>file-manager</code> e gli altri moduli . .	71
3.7	Dipendenze tra il modulo <code>user-prefs</code> e gli altri moduli . . . .	73
3.8	Dipendenze tra il modulo <code>content-script</code> e gli altri moduli .	73
3.9	Widget con la funzione di modifica pagina disattivata) . . . .	76
3.10	Widget con la funzione di modifica pagina attivata) . . . . .	76

---

3.11	Icona del widget e visualizzazione del menu dell'add-on . . . . .	76
3.12	Interfaccia preferenze sui colori . . . . .	78
3.13	Interfaccia preferenze sui colori con attiva l'opzione <b>Text yellow</b> and <b>background black</b> . . . . .	79
3.14	Scelta dei colori di testo e sfondo tramite l'opzione <b>Customize</b> .	79
3.15	Opzioni presententi nella voce <b>Text</b> . . . . .	81
3.16	Tipi di font tra i quali l'utente può scegliere per adattare il testo delle pagine Web . . . . .	82
3.17	Opzioni presentati nella voce di menu <b>Visual</b> . . . . .	83
3.18	Immagine visualizzata normalmente dal browser . . . . .	84
3.19	Immagine visualizzata con l'opzione <b>Display images description</b> attiva . . . . .	84
3.20	Immagine con l'attributo <b>longdesc</b> . . . . .	84
3.21	Miglioramento nella visualizzazione del puntatore del mouse .	84
3.22	Opzioni <b>Highlight element focused with tab key</b> e <b>with mouse</b> attivate . . . . .	85
3.23	Opzioni voce <b>Audio</b> . . . . .	86
3.24	Selezione dell'immagine e visualizzazione nel menu di contesto della voce <b>GAPforAPE selected image info</b> . . . . .	87
3.25	Pagina Web in cui è possibile modificare e aggiungere gli at- tributi dell'immagine selezionata . . . . .	88
3.26	Voce di menu <b>Color</b> per uno specifico utente ipovedente . . . . .	90
3.27	Voce di menu <b>Text</b> per uno specifico utente ipovedente . . . . .	91
3.28	Voce di menu <b>Visual</b> per uno specifico utente ipovedente . . . . .	91
3.29	Pagina di GooglePlus adattata in base al profilo scelto dall'u- tente . . . . .	93
3.30	Pagina di GooglePlus visualizzata con il profilo di default . . . . .	93

# Capitolo 1

## Stato dell'arte

Il Web è una risorsa sempre più importante e presente nella vita di tutti i giorni ed è utilizzato nei più svariati contesti: nell'istruzione, nel lavoro, nel governo, nella sanità, nel commercio, nel tempo libero. È importante che il Web sia accessibile per offrire pari accesso e pari opportunità alle persone con disabilità e far in modo che possano partecipare attivamente alla società, in modo particolare in quest'era di Web 2.0, in cui l'enfasi è sulla partecipazione dell'utente. Nonostante ciò, esiste una scarsa cultura dell'accessibilità che crea barriere non superabili, rendendo il Web poco fruibile a persone che accedono con dispositivi non standard. Per questo motivo sono nati strumenti che permettono di agire sulle pagine Web localmente attraverso il proprio browser. Un esempio è l'estensione GreaseMonkey, per il browser Firefox, che permette di eseguire user script che possono manipolare e cambiare il comportamento delle pagine Web. Meccanismo che può offrire grandi potenzialità per il miglioramento dell'accessibilità.

## 1.1 Accessibilità dei siti Web

Per introdurre cosa si intende per accessibilità dei siti Web si è deciso di riportare la citazione di Tim Berners-Lee<sup>1</sup> presente nella pagina del W3C<sup>2</sup> dedicata alla voce “accessibilità”:

The power of the Web is in its universality.

Access by everyone regardless of disability is an essential aspect.

che tradotto in italiano significa: “La forza del Web è nella sua universalità. L’accesso per tutti, a prescindere dal tipo di disabilità, è un aspetto essenziale”. Nella seconda frase sono presenti i due concetti chiave che riguardano l’accessibilità, ovvero “accesso da parte di tutti” e “disabilità”.

In un modo più formale, l’articolo 2 della legge Stanca (che verrà trattata in dettaglio in 1.1.4) definisce l’accessibilità come “la capacità dei sistemi informatici, nelle forme e nei limiti consentiti dalle conoscenze tecnologiche, di erogare servizi e fornire informazioni fruibili, senza discriminazioni, anche da parte di coloro che a causa di disabilità necessitano di tecnologie assistive o configurazioni particolari”.

### 1.1.1 Disabilità e tecnologie assistive

#### Definizione di disabilità

L’accessibilità coinvolge una vasta gamma di disabilità. Ma cosa si intende per disabilità? L’Organizzazione Mondiale della Sanità (OMS) nel primo documento del 1980 intitolato “International Classification of Impairments, Disabilities and Handicaps (ICIDH)” fornisce una sequenza di definizioni che portano dalla menomazione all’handicap: “la menomazione è il danno biologico che una persona riporta a seguito di una malattia (congenita o meno) o di un incidente; la disabilità è l’incapacità di svolgere le normali

---

<sup>1</sup>Direttore del W3C ed inventore del World Wide Web

<sup>2</sup>Il World Wide Web Consortium è una comunità internazionale che sviluppa gli standard del Web.

attività della vita quotidiana a seguito della menomazione; l'handicap è lo svantaggio sociale che deriva dall'aver una disabilità" [6].

Nel secondo documento dell'OMS, del 2001, dal titolo International Classification of Functioning, Disability and Health (ICF) il concetto di disabilità viene definito in modo più complesso, prendendo in considerazione anche una serie di fattori e variabili che possono influenzare la definizione di disabilità stessa, superando la visione classica del rapporto causa/effetto presente nel documento precedente. I nuovi fattori presi in considerazione riguardano soprattutto l'analisi del contesto e del soggetto inserito in ambiente. In particolare il nuovo documento sostituisce ai termini menomazione, disabilità e handicap altri termini nella nuova prospettiva che consente di valutare, non l'incapacità in sé, ma quanto l'individuo è in grado di fare, gli ostacoli da rimuovere o gli interventi da effettuare affinché lo stesso possa raggiungere il massimo della propria realizzazione. Questi nuovi termini sono:

- *funzioni corporee*: intese come funzioni fisiologiche dei sistemi corporei, incluse le funzioni psicologiche;
- *strutture corporee*: sono parti anatomiche del corpo come organi, arti e loro componenti;
- *attività*: è l'esecuzione di un compito o di un'azione da parte di un individuo;
- *partecipazione*: è il coinvolgimento di un individuo in una situazione di vita;
- *fattori ambientali*: sono caratteristiche, del mondo fisico, sociale e degli atteggiamenti, che possono avere impatto sulle prestazioni di un individuo in un determinato contesto. In questa definizione rientrano anche i prodotti informatici e le tecnologie.

Questi fattori vengono quindi valutati su tre dimensioni, che coinvolgono le funzioni e le strutture corporee, le attività e le eventuali limitazioni, nonché

la partecipazione del soggetto. In questo modo la definizione di abilità e disabilità possono essere date con maggiore completezza in un inquadramento generale.

Parlando di accessibilità diventano di particolare interesse i fattori ambientali e la partecipazione perché sono quelli su cui maggiormente impatta.

### **Tipi di disabilità e tecnologie assistive**

Le disabilità possono essere raggruppate in tre macro-categorie[7]:

- disabilità sensoriali;
- disabilità motorie;
- disabilità cognitive.

A differenza della categoria esistono tecnologie assistive, ovvero software o hardware che permettono di superare o mitigare queste disabilità. L'articolo 2 della legge Stanca (che sarà trattata in dettaglio in 1.1.4) definisce formalmente le tecnologie assistive come “gli strumenti e le soluzioni tecniche, hardware e software, che permettono alla persona disabile, superando o riducendo le condizioni di svantaggio, di accedere alle informazioni e ai servizi erogati dai sistemi informatici.”

**Disabilità sensoriali** Le disabilità sensoriali sono tutte quelle limitazioni che riguardano uno dei cinque sensi dell'uomo. I sensi più colpiti sono vista e udito.

Le disabilità che riguardano la vista sono le più complesse e variegata e comprendono cecità, daltonismo e patologie che impediscono la corretta interpretazione dei colori e ipovisione.

**Cecità** La cecità è sicuramente una delle disabilità più conosciute e studiate nell'ambito dell'accessibilità dei siti Web. I non vedenti non riescono ad utilizzare lo schermo e nessun output visivo, inoltre, non possono utilizzare il



mouse che richiedere una forte coordinazione mano-occhi. Per questi motivi, per utilizzare il computer, ricorrono a due diverse modalità: attraverso audio o alfabeto Braille.

- *Dispositivi di output tattile.* Esistono display Braille<sup>3</sup> che traducono l'informazione visualizzata sullo schermo in Braille: la barra Braille converte il testo in impulsi che fanno alzare ed abbassare gli aghi riproducendo l'alfabeto Braille. L'utente può così leggere il testo attraverso il tatto. Svantaggi: le barre Braille hanno un costo molto elevato inoltre l'utilizzo dipende molto dalla conoscenza del Braille dell'utente.
- *Sintetizzatori vocali.* Sono costituiti da un generatore di suoni, un processore audio ed un software interno. Quest'ultimo converte i dati ricevuti in fonemi e sillabe che, opportunamente ricostruiti, vengono trasferiti al generatore di suoni che li trasforma in parole. É quindi una tecnica per la riproduzione artificiale della voce umana. I sistemi di sintesi vocale sono noti anche come sistemi Text-To-Speech (TTS).
- *Screen Reader.* Sono software che permettono di convertire il contenuto dello schermo in un formato di output diverso, che può essere sia audio (sintesi vocale) oppure anche in testo Braille. L'applicazione viene avviata all'avvio del computer in modo da avere da subito il controllo del sistema operativo su cui si appoggia ed interagisce. Lo screen reader riceve in input le "chiamate" del sistema operativo come, per esempio, pressioni di tasti, focus sugli elementi, apertura di applicazioni.
- *Browser vocali.* Sono browser che utilizzano la tecnologia Text-To-Speech (TTS) per leggere direttamente i contenuti permettendo la navigazione tra le pagine Web.

**Daltonismo e patologie che impediscono la corretta interpretazione dei colori** I daltonici sono persone che vedono correttamente ma

---

<sup>3</sup>Il Braille è un sistema di lettura tattile sequenziale.

hanno difficoltà a distinguere correttamente la luminosità, l'intensità e la gamma dei colori. Il daltonismo è un'alterazione ereditaria dei pigmenti della retina che ostacola la corretta percezione dei colori. Esistono tre differenti difficoltà percettive:

- protanopia che ostacola la corretta interpretazione del rosso e del verde e non consente una corretta percezione della luminosità;
- deuteranopia che, come nella pratonopia, ostacola la corretta percezione del rosso e del verde, ma permette la distinzione della luminosità;
- trianopia, patologia rara che impedisce una corretta interpretazione del blu e del verde, consentendo però una corretta percezione del contrasto giallo-blu.

Questo introduce un problema molto diffuso nell'accessibilità delle pagine Web ovvero l'uso del solo colore per veicolare le informazioni.

**Ipovisione** L'ipovisione raggruppa una gamma estremamente eterogenea e variegata di situazioni e patologie. L'Unione Italiana Ciechi (UIC) definisce "ipovedente" una persona con disabilità visiva tale da non consentire lo svolgimento delle normali attività della vita quotidiana. Gli ipovedenti necessitano di elevato contrasto e luminosità tra i colori di testo e sfondo, spesso non sopportano gli sfondi bianchi e preferiscono testo brillante (per esempio giallo o blu) su sfondo nero, inoltre, solitamente, hanno bisogno di testi ridimensionati per poter accedere agevolmente alle informazioni. Le persone con ipovisione utilizzano spesso ingranditori (magnifier), software che hanno la funzione di aumentare le dimensioni di ciò che appare sul monitor.

**Sordità** Gli utenti affetti da sordità profonda o da disturbi all'udito possono incontrare problemi soprattutto nella fruizione di informazioni multimediali, filmati audio/video e in quei casi in cui ci sono degli avvisi di errore in una procedura forniti solo in modo acustico. Essi hanno bisogno principalmente di due tipi di ausili:

- sistemi di sottotitolazione dei contenuti audiovideo che possono essere forniti dagli autori e dagli sviluppatori di contenuti multimediali usando le tecnologie esistenti per la sottotitolazione.
- sistemi di traduzione del parlato in forma di testo o anche di linguaggio dei segni, software con complesse funzioni di riconoscimento vocale.

Un'altro problema spesso non preso in considerazione è che le persone sorde possono avere problemi nella comprensione dei testi scritti. Ciò è dovuto alle difficoltà di apprendere le complessità grammaticali di una lingua in modo puramente astratto, senza cioè avere riscontro immediato e diretto dell'ascolto del linguaggio parlato, che è il motivo per cui i bambini già da piccoli interiorizzano le regole della propria lingua madre, pur senza saperle spiegare. È quindi importante cercare di fornire una semplificazione sintattica dei contenuti testuali [8].

**Disabilità motorie** Le disabilità motorie comprendono tutte le patologie, congenite, progressive o dovute ad incidenti, che limitano la mobilità della persona. I problemi di navigazione e di accesso alle informazioni riguardano in modo particolare persone che hanno paralisi, completa o parziale, degli arti superiori, assenza di un'arto superiore, problemi nel controllo degli arti superiori, difficoltà del controllo ed utilizzo dei movimenti residui degli arti superiori o all'immobilità totale della persona. In tutti i casi appena citati il problema è con l'interazione dei dispositivi di input e di output. Per questo sono stati creati numerosi dispositivi hardware in grado di migliorare l'interazione con il computer e ridurre le difficoltà [7].

- *Tastiere alternative.* Sono tastiere speciali in genere più grandi delle normali e con tasti più estesi per permettere una maggiore precisione nella digitazione. Esistono anche tastiere ridotte per venire in contro ad esigenze di utenti con mobilità particolarmente ridotta.

- *Dispositivi di puntamento elettronici.* Permettono di controllare il cursore del computer attraverso raggi infrarossi, ultrasuoni e movimenti oculari.
- *Sensori a pedale.* Sistemi hardware che sfruttano la mobilità degli arti inferiori per interagire e comandare il cursore del computer.
- *Sistemi sip-and-puff.* Permettono di interagire con il cursore attraverso i movimenti della bocca.
- *Sticks e bacchette.* Utilizzate per agevolare la digitazione sulla tastiera. In genere vengono fissate alla testa delle persone, che con il movimento del capo e facendo pressione, può interagire con il computer. Le bacchette sono utilizzate da persone con ridotta mobilità o con problemi nella conduzione di movimenti precisi.
- *Joystick.* dispositivi hardware per controllare il cursore.
- *Trackball.* dispositivi che permettono di controllare il cursore facendo corrispondere il movimento di una sfera a quello del cursore stesso. Utilizzato soprattutto da persone con buona precisione nei piccoli movimenti.

Per quel che riguarda i software, le persone con ridotta mobilità posso trovare un valido aiuto nelle tecnologie Text-To-Speech per interagire col computer attraverso comandi vocali (come fanno i non vedenti), sfruttando anche le scorciatoie da tastiera.

**Disabilità cognitive** Le disabilità cognitive raggruppano un gran numero di patologie che riguardano disturbi del linguaggio e dell'apprendimento, come la dislessia, la discalcolia, disturbi neurologici, problemi connessi alla memoria a breve termine inclusa una particolare patologia, l'epilessia fotosensibile, che viene scatenata da movimenti rapidi e ripetuti e da sfarfallii sul monitor. Le persone che soffrono di questo tipo di disabilità trovano spesso impossibile navigare su siti mal organizzati.

Sono disponibili varie tecnologie assistive sotto forma di software per permettere a coloro che soffrono di questa disabilità di interagire e fruire dell'informazione in modo efficace presentandogliela in formati appositamente studiati per semplificare l'apprendimento [7].

- *Word prediction program.* Sono dei programmi che permettono di facilitare la redazione e la scrittura. Grazie al riconoscimento delle parole, inserite in apposite liste, è possibile impostare il programma perché riconosca il testo che si sta scrivendo (predicting) prima di aver completato la scrittura dello stesso.
- *Reading comprehension program.* Sono software che permettono di svolgere con l'ausilio del computer esercizi specificatamente studiati per sviluppare la capacità di lettura e comprensione dei testi scritti. Agevolano la comprensione grazie all'aiuto di grafici, schemi, immagini e suoni.
- *Reading tools and learning disabilities program.* Sono supporti per aiutare le persone che hanno difficoltà di lettura e comprensione. Il programma offre diverse opzioni: permette di scansionare il testo scritto, navigare il testo o eseguire la lettura tramite un sintetizzatore vocale.
- *Sintetizzatori vocali e programmi di riconoscimento vocale.* La tecnologia Text-To-Speech si rileva importante anche per utenti con problemi cognitivi, oltre che come supporto ad utenti non vedenti e con disabilità motorie. La ridondanza dell'informazione generata, testo sul monitor e testo ascoltato, permette di migliorare la comprensione del testo stesso. Inoltre questa tecnologia aiuta chi soffre di afasia, ovvero la persona ha perso la facoltà di articolare correttamente la frase facendola pronunciare dal sintetizzatore vocale.

### 1.1.2 Accessibilità e divario digitale

L'altra parola chiave nella citazione di Tim Berners-Lee è "accesso da parte di tutti". Negli anni '90 negli Stati Uniti è stato introdotto il concetto di "Digital Divide" (divario digitale) in alcuni studi del "National Telecommunications and Information Administration" degli Stati Uniti che indicavano come il possesso di personal computer aumentasse solo per alcuni gruppi etnici [9]. Il termine è stato poi usato per riferirsi al divario tra gli individui, famiglie, imprese e aree geografiche a diversi livelli socio-economici sia per quanto riguarda le loro opportunità di accedere alle informazioni e alle tecnologie della comunicazione (ICT) sia al loro utilizzo di Internet per le varie attività [10]. I fattori responsabili possono essere divisi in due livelli:

- a livello macro sono importanti il contesto istituzionale, le politiche di regolazione, la struttura del mercato delle telecomunicazioni e la capacità di innovazione tecnologica di un Paese;
- a livello micro si trova ciò che riguarda le possibilità personali come genere, età, istruzione, reddito, etnia, posizione professionale.

Con la diffusione dell'uso delle tecnologie e di Internet, l'attenzione di è spostata sulle disuguaglianze nell'uso, che vengono indicate con l'espressione "digital inequalities" (disparità di accesso), interpretabili alla luce di fattori come la dotazione tecnica, le capacità cognitive, le reti sociali disponibili, la varietà e l'esperienza in Rete, le abilità e le disabilità. Negli ultimi anni la ricerca sulla disparità di accesso e utilizzo dei nuovi media ha rivolto la sua attenzione all'esperienze delle persone disabili. Anche se le persone con disabilità hanno accesso fisico ad Internet, l'hardware e software che forniscono l'accesso a Internet possono non essere configurati per permettere alle persone con disabilità di usarlo. Molte persone con disabilità utilizzano infatti tecnologie assistive per colmare anche questa problematica [11].

Secondo il W3C, una pagina Web per essere "compatibile con l'accessibilità" deve essere compatibile sia con le tecnologie assistive sia con le funzioni di accessibilità dei browser e degli altri programmi utente [12]. Il Web dovrebbe

quindi essere accessibile a tutti, indipendentemente dal tipo di tecnologia utilizzato per collegarsi alla rete. Questo introduce un significato più ampio della parola accessibilità che riguarda tutti i tipi di disabilità compresa quella “tecnologica”. Con quest’ultima si intende la difficoltà o l’impossibilità di accedere al Web a causa di tecnologie obsolete o poco diffuse o di scarse o nulle competenze informatiche. L’accessibilità ha come obiettivo quello di portare l’informazione a tutti gli utenti, indipendentemente dalla loro dotazione tecnologica, dal tipo di connessione, dal media di accesso, dalla situazione contingente e dalle abilità e disabilità della persona.

### 1.1.3 Accessibilità ed usabilità

Spesso si pensa erroneamente che i termini “accessibilità” e “usabilità” abbiano un significato molto simile. Sicuramente è vero che sono entrambi criteri progettuali ed esecutivi fondamentali ma esprimono concetti differenti anche se strettamente congiunti.

La norma ISO 9241 del 1993 definisce l’usabilità come il “grado in cui un prodotto può essere usato da particolari utenti per raggiungere certi obiettivi con efficacia, efficienza e soddisfazione in uno specifico contesto d’uso”. Questa normativa si riferisce ai prodotti informatici in generale. In realtà, il concetto di usabilità nasce negli anni ’60 nell’ambito dell’ergonomia per riferirsi a qualunque interazione uomo-artefatto. In seguito venne soprattutto studiato per i prodotti informatici (soprattutto per software), nel settore dell’ergonomia cognitiva e della scienza definita Human Computer Interaction (HCI). In questo specifico settore dell’ergonomia si studia il modo in cui un utente si costruisce un modello mentale del prodotto che sta usando, e si crea perciò determinate aspettative sul suo funzionamento. Compito degli studi di usabilità è fare in modo che il modello mentale di chi ha progettato il software, da cui deriva il suo reale funzionamento, corrisponda il più possibile al modello mentale del funzionamento del software così come se lo costruisce l’utente finale [13]. L’HCI è una scienza relativamente giovane, se si considera che la Apple Computer realizzò nel 1983 la prima interfaccia

grafica, anche se stese le prime linee guida in materia di interfacce grafiche e usabilità già nel 1978. L'usabilità si applica soprattutto alle interfacce perché è con quelle che l'utente interagisce.

Con l'avvento di Internet e la proliferazione dei siti Web, il problema dell'usabilità ha preso ancora più importanza: se un software viene normalmente usato dopo esser stato acquistato, un sito Web prima viene usato, e solo se l'uso risulta soddisfacente può dar vita ad una transazione ed eventualmente ad un guadagno [13]. Secondo Jakob Nielsen, uno dei massimi esperti di usabilità, "L'usabilità misura l'esperienza (intensità e qualità) che un utente vive interagendo con qualcosa, sia esso un sito Web oppure un software, o un qualsiasi altro dispositivo". Per lui il centro di Internet è l'utente. Se l'utente non trova ciò che cerca il sito non assolve alla propria funzione informativa; se il sito non è fruibile intuitivamente, esso non è usabile. Una pagina Web usabile è una pagina in cui l'informazione è presentata in modo intuitivo, semplice, chiaro, facile. L'enfasi è su come vengono presentate le informazioni della pagina Web e quindi su quanto sia facilmente navigabile. L'usabilità lavora a livello cognitivo e strategico ed è ben evidente all'utente finale. L'accessibilità non è certo visibile all'utente normodotato, ma la sperimenta nella navigazione più facile, nella completezza delle informazioni e nella modalità di accesso alle risorse; si può dire che è il risvolto tecnologico dell'usabilità [7]. Un sito accessibile risulterà sicuramente più usabile per tutti gli altri utenti.

#### 1.1.4 Norme e linee guida

L'accesso all'informazione è un diritto di qualsiasi cittadino per questo, in tempi e modi diversi, tutti i paesi industrializzati hanno creato norme per permettere di garantire questo diritto.

##### Stati Uniti

Il primo documento normativo è la "Rehabilitation Act" del '73, prima legge federale sui diritti civili a tutela dei diritti dei cittadini disabilitati negli



Stati Uniti. Erano gli anni della guerra in Vietnam in cui più di 100.000 uomini rientrarono in patria mutilati. Questa legge federale proibisce qualsiasi discriminazione contro i cittadini disabili da parte di qualsiasi attività o programma che riceva finanziamenti governativi. L'emendamento "section 508", aggiunto nel 1986 alla "Rehabilitation Act", fu reso effettivo il 7 agosto 1998 dal congresso degli Stati Uniti. La "section 508" era volta a favorire l'accesso alla risorse informative digitali ed elettroniche, in particolare conteneva due requisiti fondamentali:

- gli impiegati federali con una disabilità devono poter accedere alle informazioni e poter utilizzare i dati e le informazioni digitali ed elettroniche come qualsiasi altro impiegato federale non disabile;
- i membri del pubblico con disabilità che stanno cercando informazioni o servizi presso un'agenzia federale devono aver accesso e la possibilità di utilizzare i dati e le informazioni come qualsiasi altro utente non disabile.

La "section 508" conteneva indicazioni progettuali ed esecutive riguardanti: tag testuali, colori, presentazioni multimediali, leggibilità, script, link di navigazione, form e molto altro.

Il 26 luglio 1990 il Congresso americano approvò L'ADA (Americans with Disability Act) per "stabilire un chiaro limite alle discriminazioni fatte sulla base delle disabilità", e la legge entrò in vigore nel 1992. L'ADA si applica a tutti i contesti pubblici a differenza del Rehabilitation Act che si applica solo alle agenzie federali e agli uffici governativi. L'ADA è divisa in cinque sezioni: occupazione, attività statali e dei governi locali, accesso agli spazi pubblici, telecomunicazioni, altre questioni.

Un'altro documento importante è la sezione 255 della Telecommunication Act, divenuto legge nel 1996. Questa legge è diretta ai costruttori e produttori di dispositivi tecnologici, richiedendo esplicitamente che tutti i dispositivi vengano progettati e sviluppati per essere facilmente usabili ed accessibili anche alle persone disabili. Il 28 gennaio 2000, L'Access Board,

agenzia incaricata di definire e realizzare gli standard per l'accessibilità, ha reso effettivi le linee guida della section 255.

Nel frattempo, nel 1997 il World Wide Web Consortium (W3C) annunciò la nascita del Web Accessibility Initiative (WAI), progetto che partì ufficialmente nell'ottobre dello stesso anno, fondato e diretto da Judy Brewer. Obiettivo del progetto era ed è promuovere e studiare il Webdesign in modo da rendere i contenuti dei siti Internet accessibili agli utenti disabili. Il 5 maggio 1999 fu rilasciata la prima "Reccomandation" relativa alle Web Content Accessibility Guidelines (WCAG) 1.0. Le WCAG sono linee guida che spiegano come fare per rendere accessibile un sito Web alle persone con disabilità e sono rivolte a tutti gli sviluppatori.

## **Australia**

In Australia ci fu un caso molto importante perché creò precedente legale. Bruce Maguire, cittadino non vedente, non riuscì a fruire totalmente della pagina Web delle Olimpiadi di Sidney utilizzando un visualizzatore Braille ed un lettore vocale. Per questo motivo egli presentò una querela, il 7 giugno 1999, contro il SOCOG (Sidney Organization Committee for the Olympic Games) ritenendo che non rispettavano le direttive sull'accessibilità, richiamandosi all'DDA. DDA (Disability Discrimination Act) approvato nel 1992, è il documento di riferimento per la definizione di tutti gli standard inerenti l'accessibilità del paese. Nel 1999 viene espressamente specificato che, secondo le disposizioni del DDA, l'informazione online deve essere accessibile a tutti i cittadini [7]. L'HREOC (Human Rights and Equal Opportunity Commission), l'agenzia governativa che supervisiona l'applicazione dei decreti legislativi che garantiscono pari opportunità di accesso ai cittadini disabili, gli diede ragione richiedendo al comitato olimpico di adeguare il sito e di pagare una cospicua multa [14].

## Europa

In Europa nacque, nel dicembre 1999, il progetto eEurope 2002. Nel documento presentato in occasione del Consiglio Europeo straordinario di Lisbona del 23 e 24 marzo 2000, si legge: “eEurope è un’iniziativa politica intesa a garantire che l’Unione europea approfitti dei cambiamenti in atto grazie alla società dell’informazione e ne tragga i massimi vantaggi per le generazioni future.” [15]. Gli obiettivi principali di eEurope erano:

- fare in modo che ciascun cittadino, ciascuna abitazione, scuola, impresa e amministrazione entri nell’era digitale e disponga di un collegamento on-line;
- creare in Europa una padronanza degli strumenti dell’era digitale, con il sostegno di una cultura imprenditoriale pronta a finanziare e a sviluppare nuove idee;
- garantire che l’intero processo abbia luogo con la partecipazione di tutti, rafforzi la fiducia dei consumatori e potenzi la coesione sociale.

Il documento conteneva anche una parte intitolata “ePartecipazione per i disabili” i cui obiettivi erano:

- entro a fine del 2000:
  - La Commissione europea e gli Stati membri dovranno riesaminare la legislazione in materia e i programmi di elaborazione di norme connessi alla società dell’informazione, per garantirne la conformità ai principi di accessibilità e accelerare il processo di normalizzazione.
  - La Commissione europea proporrà una raccomandazione agli Stati membri, per tenere conto delle esigenze dei disabili per quanto riguarda l’approvvigionamento di prodotti e servizi di comunicazione e informazione.
- entro la fine del 2001:

- La Commissione europea e gli Stati membri dovranno impegnarsi a rendere accessibili ai disabili la struttura e il contenuto di tutti i siti Web pubblici.

Dopo eEurope 2002, ci fu il progetto eEurope 2005 e successivamente il progetto iEurope 2010.

### **Italia**

Il 23 giugno 2000 nacque ufficialmente il gruppo “IWA Italy”, la prima sezione indipendente di WAI al mondo nata per supportare al meglio tutti i Webmaster italiani, grazie principalmente al lavoro di Roberto Scano che si è reso promotore di questa iniziativa [16].

Nel settembre del 2000 venne istituito il “Comitato di studio interministeriale per il miglioramento dell’accessibilità dei siti Web delle pubbliche amministrazioni” e anche l’AIPA (Autorità per l’Informatica nella Pubblica Amministrazione) istituì il proprio gruppo di lavoro. Nel marzo 2011, l’allora Ministro della Funzione Pubblica Bassanini promulgò una direttiva intitolata “Le Linee Guida per l’organizzazione, l’usabilità e l’accessibilità dei siti Web delle pubbliche amministrazioni” in cui chiedeva il rispetto di alcuni principi che consentissero la piena accessibilità di tutti i siti della Pubblica Amministrazione. Nel settembre dello stesso anno l’AIPA ha rilasciato il documento “Criteri e strumenti per migliorare l’accessibilità dei siti Web e delle applicazioni informatiche a persone disabili” in cui chiedeva il rispetto di alcuni principi che consentissero la piena accessibilità di tutti i siti della Pubblica Amministrazione. Questo documento propone e suggerisce soluzioni concrete a cui tutte le Istituzioni italiane dovevano fare riferimento. Il problema è stato che l’adesione era a carattere volontario e quindi non hanno portato a miglioramenti concreti.

Dopo varie norme, una svolta è avvenuta nel 2004, con l’emanazione della legge “9 gennaio 2004, n. 4” conosciuta come “Legge Stanca” in onore del ministro che l’ha voluta, approvata all’unanimità dal Parlamento italiano. Già il titolo “Disposizioni per favorire l’accesso dei soggetti disabili agli stru-

menti informatici” chiarisce lo scopo della legge, che però viene definito in dettaglio nell’articolo 1 della stessa [17]:

- la Repubblica riconosce e tutela il diritto di ogni persona ad accedere a tutte le fonti di informazione e ai relativi servizi, ivi compresi quelli che si articolano attraverso gli strumenti informatici e telematici;
- è tutelato e garantito, in particolare, il diritto di accesso ai servizi informatici e telematici della pubblica amministrazione e ai servizi di pubblica utilità da parte delle persone disabili, in ottemperanza al principio di uguaglianza ai sensi dell’articolo 3 della Costituzione.

Nell’allegato A del Decreto Ministeriale 8 luglio 2005 intitolato “Verifica tecnica e requisiti tecnici di accessibilità delle applicazioni basate su tecnologie Internet” [18]. I requisiti tecnici di accessibilità erano stati definiti sulla base:

- quanto indicato nelle Recommendation del World Wide Web Consortium (W3C) ed in particolare in quelle del progetto Web Accessibility Initiative (WAI);
- standard definiti nel paragrafo 1194.22 della Sezione 508 del Rehabilitation Act degli USA;
- standard e specifiche tecniche definite in materia di accessibilità dalla International Organization for Standardization (ISO);
- esperienze acquisite nell’ambito della Pubblica Amministrazione ed in particolare, tra quelle già maturate, quelle relative all’attuazione della Circolare AIPA del 6 settembre 2001 recante “Criteri e strumenti per migliorare l’accessibilità dei siti Web e delle applicazioni informatiche a persone disabili”.

Questo allegato è stato aggiornato il 26 aprile 2010, a seguito dell’emanazione delle nuove raccomandazioni internazionali (WCAG 2.0) da parte del W3C (11 dicembre 2008) e dell’invito esteso ai 27 paesi membri da parte della Commissione europea di adottare tali raccomandazioni.

### 1.1.5 WAI

Come già anticipato, il WAI è l'iniziativa del W3C per l'accessibilità del Web. In questi anni ha sviluppato diverse linee guida per diverse finalità. Nella figura 1.1 si può vedere come le varie linee guida del WAI interagiscono tra di loro per poter fornire servizi e tecnologie accessibili alle persone con disabilità [1].

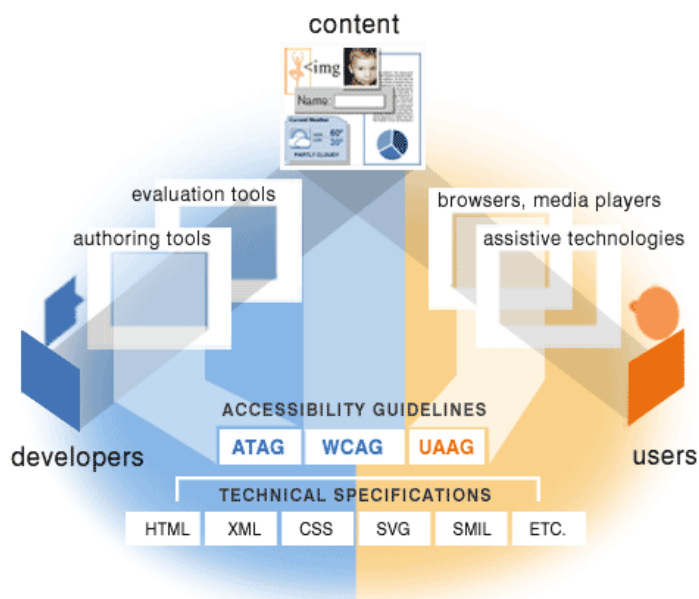


Figura 1.1: Interazione delle varie linee guida del WAI (disponibile da [1])

### WCAG

Web Content Accessibility Guidelines è un documento in cui sono presentate una serie di linee guida volte a supportare gli sviluppatori nel produrre contenuti Web accessibili. Queste linee guida riguardano i contenuti delle pagine Web. La prima versione risale al 1999, mentre la seconda al 2008. Ovviamente la seconda versione è un'evoluzione della prima dalla quale ha preso ispirazione. L'avanzamento di versione era però d'obbligo vis-

to il grande progresso tecnologico che è avvenuto dal 1999 al 2008, in cui si è passati da pagine HTML statiche a vere e proprie applicazioni dinamiche.

**WCAG 2.0** Nella seconda versione delle WCAG [12] innanzitutto vengono definiti i quattro principi che fanno da pilastri all'accessibilità del Web.

**Percepibile** Il WAI definisce il primo principio nel seguente modo: “Le informazioni e i componenti dell'interfaccia utente devono essere presentati agli utenti in modo che possano essere percepiti.”. Rientra in questo gruppo cioè che riguarda:

- *le alternative testuali*: occorre fornire alternative testuali per qualsiasi contenuto non di testo in modo che questo possa essere trasformato in altre forme fruibili secondo le necessità degli utenti come stampa a caratteri ingranditi, Braille, sintesi vocale, simboli o un linguaggio più semplice;
- *i tipi di media temporizzati*: bisogna fornire alternative per i tipi di media temporizzati come i sottotitoli per l'audio e una descrizione audio per i video;
- *l'adattabilità*: bisogna creare contenuti che possano essere rappresentati in modalità differenti (ad esempio, con layout più semplici), senza perdere informazioni o la struttura;
- *distinguibilità*: occorre rendere più semplice agli utenti la visione e l'ascolto dei contenuti, separando i contenuti in primo piano dallo sfondo.

**Utilizzabile** Per utilizzabile il WAI intende che “I componenti e la navigazione dell'interfaccia utente devono essere utilizzabili.”. Per soddisfare questo principio occorre:

- che il contenuto sia accessibile da tastiera: occorre rendere disponibili tutte le funzionalità tramite tastiera;

- fornire un'adeguata disponibilità di tempo: gli utenti devono avere tempo sufficiente per leggere ed utilizzare i contenuti;
- fare in modo che il contenuto non provochi convulsioni: occorre non sviluppare contenuti che possano causare crisi epilettiche;
- il contenuto deve essere navigabile: occorre fornire delle funzionalità di supporto all'utente per navigare, trovare contenuti e determinare la propria posizione.

**Navigabile** Un'altro importante principio dell'accessibilità è la navigabilità delle pagine. La definizione fornita nelle WCAG 2.0 è “Le informazioni e le operazioni dell'interfaccia utente devono essere comprensibili.” Un pagina è navigabile se:

- è leggibile: bisogna rendere il testo leggibile e comprensibile;
- è prevedibile: occorre creare pagine Web che appaiano e che siano prevedibili;
- assistenza nell'inserimento: sarebbe corretto aiutare gli utenti ad evitare gli errori ed agevolarli nella loro correzione.

**Robusto** Il quarto ed ultimo principio è la robustezza, e per soddisfarlo “Il contenuto deve essere abbastanza robusto per essere interpretato in maniera affidabile mediante una vasta gamma di programmi utente, comprese le tecnologie assistive.”. In questo principio rientra anche la compatibilità, infatti occorre garantire la massima compatibilità con i programmi utente attuali e futuri, comprese le tecnologie assistive.

Tutti questi principi corrispondono a precise linee guida in cui viene spiegato tecnicamente come fare per soddisfarli e cosa non fare senza lasciare dubbi interpretativi.



### ATAG

Le “Authoring Tool Accessibility Guidelines (ATAG)” sono linee guida volte a supportare gli sviluppatori di software utilizzati per produrre pagine e contenuti Web a generare contenuti accessibili nonché a produrre applicazioni e documentazione accessibili. Quando si parla di software per produrre pagine e contenuti Web, chiamati anche strumenti di pubblicazione, ci si riferisce in particolare ai Content Management System (CMS) il cui utilizzo si è molto diffuso in questi anni. Per questo il WAI sta lavorando alla seconda versione, il cui ultimo Working Draft risale al 21 luglio scorso [19], per aggiornare le linee guida, visto che la prima versione risale al 2000, quando ancora i CMS non erano molto diffusi.

### UAAG

Le “User Agent Accessibility Guidelines (UAAG)” sono una serie di linee guida volte a supportare i produttori di programmi utente come browser, media player, tecnologie assistive ed altri programmi di navigazione. La prima versione di queste linee guida per l’accessibilità dei programmi utente risale al 2002 ed è attualmente in via di sviluppo la seconda versione, il cui ultimo working draft è stato pubblicato il 19 luglio 2011 [20].

## 1.2 Il Web 2.0

Originariamente il Web è stato concepito come un modo per visualizzare documenti ipertestuali statici, creati utilizzando solamente HTML, un linguaggio di marcatura ipertestuale. L’utente utilizzava il Web in modo passivo, accedeva alle informazioni ma non aveva iterazione con i siti Web. In seguito, con l’utilizzo di linguaggi di programmazione come JavaScript, degli elementi dinamici e dei fogli di stile attraverso il linguaggio di stile Cascading Style Sheets (CSS) i siti sono passati da statici a dinamici. Un ulteriore progresso è avvenuto con l’integrazione con database e all’utilizzo di sistemi di gestione dei contenuti (CMS), con i quali fu possibile creare forum e blog.

In una conferenza nel 2005 Tim O'Reilly<sup>4</sup> ha coniato il termine Web 2.0 per descrivere la seconda generazione del World Wide Web in cui il focus è sulla comunicazione, sull'abilità delle persone di collaborare e condividere informazioni online, sfruttando l'intelligenza collettiva [21]. È cambiato l'approccio dell'utente al Web, che accede alle pagine non solo per consultarle ma anche per contribuire alla creazione dei contenuti. Il termine Web 2.0 non indica quindi un'innovazione tecnologica di per se, quello che fa è unire in modo innovativo componenti e servizi che già esistevano, ma indica un fenomeno sociale, una nuova filosofia di come usare Internet [22]. Esempi lampanti di questa nuova filosofia sono applicazioni Web come Facebook [23], Twitter [24], YouTube [25], Wikipedia [26] ma anche tutti quei blog e siti che promuovono lo scambio di informazioni e la collaborazione degli utenti. Si parla di User Generated Content (UGC) per riferirsi ai contenuti generati dagli utenti dei siti Web; questi contenuti possono assumere diverse forme come video, testo, immagini, foto, audio. I siti possono poi aggregare e condividere queste contenuti.

### 1.2.1 Tecnologie del Web 2.0

Le tecnologie chiave che caratterizzano il Web 2.0 sono Ajax e RIA.

**Ajax** Le tecnologie più utilizzate, oltre HTML, per implementare il Web 2.0 sono linguaggi di scripting e Cascading Style Sheets (CSS). Il più diffuso linguaggio di scripting per browser è JavaScript, che permette di aggiornare gli stili, eseguire calcoli, convalidare i dati inseriti dall'utente e aggiornare le pagine dinamicamente attraverso XMLHttpRequest, il tutto lato client.

XMLHttpRequest (XHR) è un API<sup>5</sup> che può essere usata da JavaScript (o da un'altro linguaggio di scripting) per trasferire dati attraverso il pro-

---

<sup>4</sup>È il fondatore di O'Reilly Media, casa editrice americana che pubblica libri e articoli e tiene conferenze che riguardano l'informatica.

<sup>5</sup>API (application programming interface) è un insieme di regole e specifiche che un programma software può seguire per comunicare con un'altro.

protocollo HTTP<sup>6</sup> per aggiornare porzioni di pagine Web. L'implementazione iniziale prevedeva il trasferimento di dati tramite eXtended Markup Language (XML) ma oggi si sono diffusi altri formati come per esempio JSON<sup>7</sup>. Scripting e trasferimento di dati tramite XMLHttpRequest sono le tecnologie chiavi nel termine Ajax (Asynchronous JavaScript and XML) [27]. Ajax permette di aggiornare dinamicamente solo porzioni di pagine a differenza del Web delle pagine statiche in cui si richiedeva e si caricava l'intera pagina ogni volta [28].

**RIA** Grazie anche alle connessioni sempre più veloci, vengono sempre più spesso inseriti elementi multimediali nelle pagine Web, che contengono video e audio. Per farlo, si è diffuso l'uso di Adobe Flash, una piattaforma multimediale per aggiungere video e animazioni per creare esperienze multimediali [29]. Adobe Flash è la più utilizzata piattaforma per realizzare RIA (Rich Internet Application), applicazioni Web che hanno tutte le caratteristiche delle applicazioni desktop. Sono caratterizzate da una grande interattività e multimedialità e dalla velocità di esecuzione. Infatti la parte dell'applicazione che elabora i dati è trasferita a livello client e fornisce una pronta risposta all'interfaccia utente, mentre la gran parte dei dati e dell'applicazione rimane sul server remoto [30].

Queste evoluzioni hanno fatto nascere nuove sfide per l'accessibilità [31].

## 1.3 L'accessibilità nel Web 2.0

Per introdurre l'argomento si riporta una citazione di Yannis Vardakastanis, presidente dell'European Disability Forum:

---

<sup>6</sup>HTTP (Hypertext Transfer Protocol) è il principale protocollo di rete usato per la trasmissione d'informazioni sul Web.

<sup>7</sup>JSON (JavaScript Object Notation) è un semplice formato per lo scambio di dati che per le persone è facile da leggere e scrivere, mentre per le macchine risulta facile da generare e analizzarne la sintassi

Nell'attuale società dell'informazione, per le persone disabili, il modo in cui i siti Internet vengono realizzati può significare la differenza tra integrazione ed esclusione sociale.

La nuova filosofia di partecipazione dell'utente nel Web 2.0 dovrebbe fornire nuove possibilità alle persone con disabilità di partecipare alla società direttamente da casa, avendo a disposizione i propri ausili tecnologici, favorendo un ulteriore abbassamento delle barriere. Infatti la dinamicità delle applicazioni Web permettono di fornire vari servizi, come shopping, operazioni bancarie, intrattenimento, e-learning in modo sempre più semplice ed avanzato. Lo shopping online permette alle persone con problemi visivi o fisici di acquistare in modo indipendente senza dover raggiungere un posto fisico o chiedere aiuto ad altre persone. E-learning offre formazione per persone di ogni estrazione sociale e abilità.

Questi esempi dimostrano che queste tecnologie entrano a far parte della cultura, portando benefici anche sulle persone con disabilità che possono diventare più indipendenti, avere nuove opportunità di lavoro e interazioni sociali. Comunità virtuali, social network, siti di intrattenimento possono fornire importanti opportunità di interazione e socializzazione. In questi ultimi anni si parla infatti di "Digital inclusion" [32].

Per far sì che queste nuove potenzialità vengano sfruttate anche dalle persone disabili, occorre che le tecnologie assistive possano integrarsi perfettamente con queste applicazioni Web altamente dinamiche.

Il Web tradizionale oggi è sufficientemente supportato dalle tecnologie assistive come screen reader, ingranditori di schermo e dispositivi di input alternativi che però inizialmente hanno avuto difficoltà ad integrarsi con il Web. Con la seconda generazione del Web le problematiche si sono enormemente amplificate a causa della forte dinamicità delle applicazioni Web che usano scripting e altre tecnologie avanzate per rendere le pagine visivamente accattivanti e altamente interattive. La maggior parte di queste applicazioni richiedono un utilizzo massiccio del mouse per interagire con esse. L'incremento degli aggiornamenti delle pagine che vogliono fornire aggiornamenti

in tempo reale non sono sempre accessibili alle persone che utilizzano tecnologie assistive. Queste ultime infatti non riescono sempre ad interpretare il modello di interazione con l'utente o non sono consapevoli dei tanti aggiornamenti che si verificano nella pagina o non sanno come avvisare l'utente dei cambiamenti [28].

### 1.3.1 ARIA

Per risolvere vari problemi delle tecnologie utilizzate nel Web 2.0 [33], in particolare quelli legati a RIA e Ajax, il W3C Web Accessibility Initiative ha sviluppato una suite di documenti tecnici per generare applicazioni e contenuti Web dinamici accessibili alle persone con disabilità. L'obiettivo delle Accessible Rich Internet Applications (ARIA) [34] è di aggiungere dati semantici nel HTML e XHTML per permettere alle tecnologie assistive di rappresentare in modo migliore i componenti dell'interfaccia utente e le interazioni dinamiche con l'utente. La principale funzionalità dello standard ARIA è di permettere alle pagine Web (o porzioni di pagine Web) di dichiarare se stesse come applicazioni piuttosto che documenti statici aggiungendo del ruolo, delle proprietà e delle informazioni di stato di applicazioni Web dinamiche. ARIA deve essere usata dagli sviluppatori di applicazioni Web per aumentare la loro compatibilità con gli screen reader e le altre tecnologie assistive con l'aiuto del supporto per ARIA dei browser. Questo fornisce informazioni addizionali che verranno utilizzate dalle tecnologie assistive per capire il tipo del widget e il suo stato corrente. Aggiungendo il ruolo e le informazioni di stato, un'applicazione Ajax può diventare più accessibile e uno screen reader può ottenere informazioni utili come sapere se il focus si trova su un elemento dell'albero, se l'elemento è espanso o compresso o il livello dell'elemento nella gerarchia dell'albero [35].

## 1.4 Adattamento e transcodifica dei contenuti

Le applicazioni del Web 2.0 utilizzano contenuti sempre più dinamici ed interattivi e anche molti elementi multimediali. Ovviamente anche questo aspetto ha profondamente colpito l'accessibilità del Web, portando la ricerca ad occuparsi di adattamento e transcodifica dei contenuti [4].

Il W3C definisce “adattamento dei contenuti” come “un processo di selezione, generazione, o modifica che produce uno o più unità percepibili in risposta alla richiesta di un Uniform Resource Identifier in un dato contesto di distribuzione.”. Ovvero, attraverso l'adattamento dei contenuti (content adaptation) si trasforma un documento in input in uno o più documenti parziali in base alle informazioni che riguardano il dispositivo client (browser), la rete in uso e le preferenze degli utenti. Proprio questo bisogno di personalizzazione ha portato all'introduzione di servizi di adattamento dei contenuti che adattano le risorse Web alle preferenze degli utenti e alle capacità e vincoli dei loro client. Si cerca di fornire un “accesso universale” ai contenuti multimediali.

La transcodifica dei contenuti viene spesso effettuata per adattare video, immagini, audio e testo della presentazione alle diverse esigenze, effettuando una trasformazione, conversione e riassunto dei contenuti “on the fly” (sul momento).

Da qualche anno si parla di adattamento e transcodifica dei contenuti soprattutto per quel che riguarda i diversi media con cui si può consultare una pagina Web, soprattutto per ciò che riguarda il mobile. Facilmente si può capire la rilevanza di questo meccanismo nel poter garantire accessibilità dei contenuti adattandoli in base alle esigenze ed abilità delle persone e anche in base alle tecnologie assistive utilizzate in modo da inviargli contenuti speciali. Infatti, la crescita e l'efficacia delle attuali tecnologie assistive spinge le persone disabili ad accedere sempre di più al Web e sfruttarne i servizi.

### 1.4.1 Profilazione

Per poter adattare i contenuti in base alle varie preferenze, occorre avere un meccanismo di profilazione delle esigenze degli utenti. La profilazione nel contesto dell'accessibilità è un problema complesso, per le diversità dei bisogni e delle abilità. Inoltre, occorre sia considerare il profilo dell'utente che il profilo del device utilizzato. La profilazione dell'utente in base alle preferenze ed esigenze, così come le caratteristiche del device è una comune ed utile funzionalità dei progetti dedicati all'adattamento dei contenuti. Ci sono molti e diversi livelli che possono essere usati per descrivere questi due insiemi di capacità.

#### Social Network

I principali social network hanno elevate carenze per ciò che riguarda l'accessibilità dei loro contenuti. Per esempio, in Facebook [23] l'accessibilità è affrontata in un modo molto superficiale e parziale, soprattutto sulla versione mobile del sistema. Analogamente in Twitter [24], MySpace [36] e altri social networks ampiamente usati non sono accessibili in conformità alle normative o le linee guida del W3C e ciò crea significative barriere. Solitamente, questi social network permettono poca personalizzazione agli utenti, come la possibilità di cambiare qualche caratteristica del layout, come il colore dello sfondo o la grandezza del testo. Solitamente queste modifiche sono legate solo alla pagina del profilo dell'utente e devo essere specificate manualmente senza alcun supporto di qualche strumento di profiling. Dotare queste piattaforme di sistemi di adattamento in base a profili standard potrebbe migliorare l'accessibilità [37].

#### E-learning

E-learning comprende tutte le forme di supporto elettronico all'insegnamento e all'apprendimento.

I diversi bisogni degli studenti necessitano di sistemi alternativi di accesso e di contenuto presentato in diverse modalità. Infatti, gli studenti con bisogni speciali hanno bisogno di accedere all'e-learning in modo personalizzato potendo accedere alle risorse per l'apprendimento in modo accessibile [38]. Un e-learning accessibile è fondamentale per permettere di svolgere attività educativa a distanza e in modo flessibile, diminuire il rischio di escludere gli individui con particolari capacità di accesso, incrementare l'inclusione sociale degli studenti con bisogni speciali [37]. Una completa informazione, sia sul profilo dell'utente che sul profilo del device, potrebbe essere usata in modo vantaggioso incrementando l'esperienza didattica degli studenti.

### **Protocolli di profilazione**

Esistono vari protocolli di profilazione dei device che descrivono tecnicamente le caratteristiche dei vari dispositivi, in termini di funzionalità software e hardware come: CC/PP (Composite Capabilities/Preferences Profile) [39] del W3C e UAProf (User Agent Profile) [40] del Open Mobile Alliance (OMA) entrambi basati su RDF (Resource Description Framework) e con l'obiettivo di consentire ai dispositivi client di informare i server sulle loro capacità e caratteristiche.

Uno dei principali standard di accessibilità dedicato al profilo degli utenti con vincoli per l'accessibilità è l'IMS ACCessibility for Learner Information Package (ACCLIP) [41], che è una parte delle specifiche IMS Learner Information Package (IMS LIP) [42]. L'intento di questa specifica è quello di definire una serie di pacchetti che possono essere utilizzate per l'importazione (estrazione) di dati in (da) una piattaforma di e-learning compatibile con IMS. ACCLIP descrive gli utenti in termini di bisogni di accessibilità usando una sintassi basata su XML. In sostanza, consente la descrizione delle preferenze dell'utente (visiva, auditiva o dispositivo), che possono essere utilmente sfruttate per adattare contenuti didattici. Esso fornisce quindi un mezzo per descrivere come gli utenti interagiscono con un ambiente di e-learning, con-



centrandosi sui requisiti di accessibilità. Le preferenze di accessibilità possono essere divise in tre gruppi [37]:

- *informazioni sulla visualizzazione* che descrivono come l'utente preferisce visualizzare o presentare le informazioni, per esempio informazioni sul cursore, i font dei caratteri, sulle caratteristiche dei colori. Inoltre, è possibile specificare se si utilizza uno screen reader, specificando le preferenze come il volume, la velocità, o la necessità di allarmi visivi invece di quelli sonori;
- *informazioni di controllo* che definiscono come l'utente preferisce controllare il dispositivo, esprimendo, per esempio, preferenze rispetto all'uso della tastiera. Inoltre, è possibile dichiarare la necessità di utilizzare meccanismi di controllo non tipici, come una tastiera onscreen, una tastiera alternativa, solo l'emulazione del mouse, un meccanismo alternativo di puntamento e qualsiasi riconoscimento vocale;
- *informazioni sul contenuto* che descrivono quali miglioramenti, alternativi o equivalenti contenuti l'utente richiede. Per esempio è possibile definire come presentare testo, contenuti uditi e visuali e se occorre uno style sheet personale.

## 1.5 Augment browsing e Crowdsourcing

### Augment browsing

Il termine “Augment browsing” si riferisce ad un sistema che permette di personalizzare il look e le funzioni dei siti Web “on-the-fly” (ovvero sul momento) e rendere le modifiche persistenti in modo che ogni volta che si torna in quella pagina, il sito viene automaticamente aggiornato in base ai proprio bisogni. L'innovazione di questo sistema è che si basa su scripting a lato client, che vengono eseguiti dal proprio browser, senza quindi necessitare di nessuna modifica da parte degli sviluppatori delle pagine Web. Con questo

sistema non occorre che tutte le parti coinvolte nella creazione di siti si convincano che l'accessibilità è un requisito fondamentale. Infatti, per avere un Web accessibile occorrerebbe che:

- i Web developer capiscano l'importanza dell'accessibilità e progettino pagine Web mettendola in pratica;
- Web authoring tools permettano e facilitino la creazione di contenuti accessibili;
- le tecnologie assistive e i browser supportino gli standard in materia di accessibilità;
- siano promulgate leggi e norme che favoriscano la diffusione della cultura dell'accessibilità e che obblighino il rispetto e l'attuazione dell'accessibilità nei siti Web.

È evidente che lo sviluppo di sistemi di augment browsing per migliorare l'accessibilità possa aprire nuove opportunità per l'accessibilità dei siti, permettendo di coprire molte problematiche direttamente lato client.

Sono due le modalità principalmente utilizzate per fare augment browsing (per un approfondimento delle due tecniche si veda 2.1.3):

- *user scripts*: sono degli script scritti in JavaScript che in modo automatico permettono di effettuare modifiche nella pagina visualizzata nel browser. Per poter essere utilizzati occorre aver installato nel browser delle estensioni che li supportino;
- *bookmarklets*: è una tecnica che permette di iniettare del codice JavaScript nella pagina appena caricata dal browser. Necessita dell'intervento dell'utente che ne attiva il meccanismo su ogni pagina desiderata attraverso un click.

## Crowdsourcing

Wikipedia definisce con “crowdsourcing” “l'atto di esternalizzazione (outsourcing) di attività tradizionalmente effettuate da un singolo individuo ad

un gruppo indefinito di persone o una comunità [43]. In realtà il termine può essere usato per una varietà di approcci che fruttano tutte le potenzialità di una grande folla di persone. Effettuare questa attività sul Web ha la grande potenzialità di poter raggiungere un grande numero di persone. L'idea è *“se una cosa non viene fatta dagli altri falla tu e condividila con la comunità”* [44].

Il crowdsourcing può essere usato per migliorare l'accessibilità permettendo di effettuare una miriade di correzioni e fornire indicazioni utili agli utenti. Si possono intravedere grandi potenzialità [44]:

- risolvere il problema sul momento fornendo un accesso immediato;
- mostrare soluzioni di accessibilità per il proprietario del sito o sviluppatore;
- strumento didattico per i bambini per imparare Web design;
- imparare cos'è l'accessibilità e come potere implementare pagine Web accessibili in modo facile.

Esistono varie applicazioni e strategie che permettono di migliorare l'accessibilità delle pagine sfruttando le tecniche di augment browsing e crowdsourcing. Di seguito ne verranno illustrati gli esempi più significativi e diffusi.

### 1.5.1 GreaseMonkey

Il più popolare sistema di augment browsing per Firefox è GreaseMonkey [45]. GreaseMonkey è un'estensione per il browser Firefox che permette agli utenti di installare script che possono fare modifiche sul momento al contenuto HTML e sull'aspetto delle pagine Web [45], [46]. Già dalla sua nascita, molti sviluppatori hanno capito le potenzialità per l'accessibilità di questa estensione. Infatti, si possono trovare nei numerosi repository vari script che permettono di migliorare l'accessibilità delle pagine Web [47], [48]. Alcune funzionalità fornite sono [4]:

- modificare la pagina per migliorare l'accessibilità aggiungendo le intestazioni oppure rimuovendo i conflitti tra i tasti di scelta rapida;
- adattare le pagine per renderle migliori per utenti con determinati bisogni, per esempio aumentando il contrasto dei colori, la grandezza del testo, lo stile del font. Molti script sono infatti dedicati a persone che navigano il sito attraverso screen reader o persone con ipovisione;
- aggiungere funzionalità che gli sviluppatori non hanno inserito come il supporto per i tasti di scelta rapida o la giusta annotazione delle tabelle;
- superare barriere per l'accessibilità molto significative come, per esempio, provare a risolvere o rimuovere i CAPTCHA<sup>8</sup>.

Questi script sono distribuiti attraverso più repositor che spesso non sono loro stessi accessibili. Inoltre, solitamente ogni script svolge un'unica funzionalità, quindi occorre utilizzarne vari per poter soddisfare i proprio bisogni. Per superare la mancanza di portabilità di questo approccio ed effettuare l'integrazione tra diversi script è stato sviluppato AccessMonkey.

### 1.5.2 AccessMonkey

AccessMonkey [49], [50] è un framework per migliorare l'accessibilità delle pagine Web dopo la loro pubblicazione, ispirato a GreaseMonkey. È stato progettato per offrire agli utenti Web la possibilità di migliorare in modo indipendente l'accessibilità del Web e di rendere tali miglioramenti a disposizione degli utenti Web e sviluppatori Web. AccessMonkey è un plugin per browser che recupera automaticamente gli script utente e permette di applicare le trasformazioni utili alle pagine visitate da parte degli utenti.

---

<sup>8</sup>Con l'acronimo inglese CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) si indica in ambito informatico un test per determinare se l'utente sia umano.

L'interfaccia end user permette a chiunque di creare e condividere i miglioramenti e le nuove interfacce sviluppate permettendo ai creatori di contenuti di modificare e salvare le modifiche che gli script fanno al contenuto Web [49], [50].

Gli obbiettivi di AccessMonkey sono:

- miglioramenti end-user: permette miglioramenti indipendenti al contenuto Web da parte di chi non ha esperienza di programmazione utilizzando un'interfaccia accessibile progettata per gli utenti finali;
- collaborazione: facilitare la condivisione di miglioramenti sia tra gli utenti che tra gli sviluppatori Web. Fornire un forum per gli utenti in cui aiutarsi;
- piattaforma comune: fornisce una piattaforma comune per il miglioramento dell'accessibilità, che contribuisce a raggiungere in modo migliore gli utenti finali.

Nella figura 1.2 si può vedere l'architettura di AccessMonkey.

### 1.5.3 Web Visum

Web Visum è un'estensione per Firefox che permette di migliorare l'accessibilità Web soprattutto per le persone cieche e ipovedenti [51]. Anche esso è un sistema di augment browsing. Offre varie funzionalità interessanti, eccone alcuni esempi.

- Risolve in modo istantaneo e automatico le immagini CAPTCHA.
- Costruisce funzioni che permettono di navigare le pagine con più facilità e meno confusione.
- Effettua varie modifiche al codice HTML per migliorare la navigazione attraverso lo screen reader.

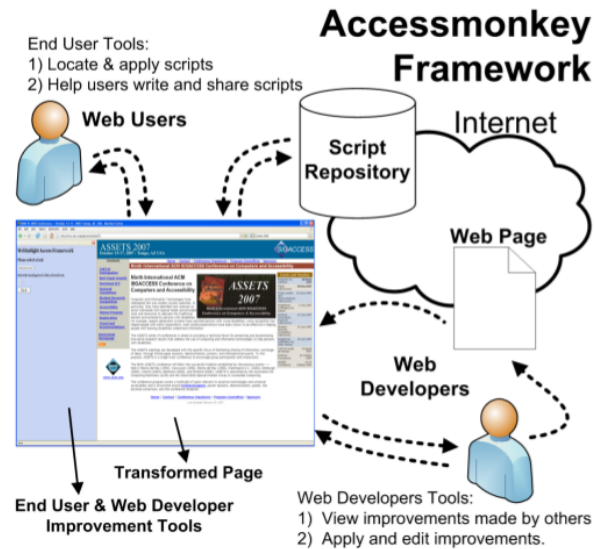


Figura 1.2: Architettura AccessMonkey

- Gli utenti ipovedenti beneficiano di miglioramenti come aumento di contrasto nella pagina, link ed elementi con il focus messi in evidenza.
- Permette di inserire o modificare i nomi dei link non significativi in modo che lo screen reader la volta in cui si tornerà in quella pagina possa leggere il nome giusto. Lo stesso può essere fatto per il titolo della pagina.
- Estrae il testo dalle immagini.
- La comunità può suggerire come migliorare le pagine e come etichettare i vari elementi.

#### 1.5.4 Web Adaptation Technology

Il progetto Web Adaptation Technology (WA) è un progetto di ricerca dell'IBM che fornisce un miglioramento del browser Web Microsoft Internet

Explorer che permette agli utenti di configurare la loro esperienza di navigazione in base alle preferenze personali [52], [53]. Questo progetto consente agli utenti di applicare dinamicamente una o più trasformazioni alla pagina Web. Queste trasformazioni manipolano la presentazione delle pagine Web senza richiedere l'intervento degli autori o dei fornitori di contenuti per riscrivere la pagina. Il sistema che ne risulta consente ad una vasta gamma di utenti con ridotte capacità visive, cognitive e abilità motorie di accedere a un'ampia porzione di pagine Web usando un browser standard.

### 1.5.5 Farfalla Project

Il Farfalla Project [54], [55] è un sistema di augment browsing che però non è implementato attraverso un'estensione del browser, ma si appoggia ad un server che inietta del codice JavaScript nelle pagine caricate attraverso un bookmarklet, permettendo a Farfalla di diventare cross-browser. Il JavaScript iniettato crea nella pagina una toolbar che permette di richiamare diversi servizi che corrispondono ad altrettante modifiche sulla pagina. I servizi disponibili sono:

- *magnifier*: un plug-in che permette di ingrandire la porzione di testo in cui si trova il puntatore, in una parte di schermo dedicata, rendendo il testo nero su sfondo bianco, in modo da renderlo facilmente leggibile.
- *tastiera virtuale*: questa tastiera viene visualizzata solo se l'utente fa click su gli elementi HTML `textarea` o `input` che permettono all'utente di scrivere ed inserire del testo;
- *grandezza del testo*: permette di ingrandire o diminuire la grandezza dei caratteri della pagina;
- *aumento del contrasto*: cambia il colore dello sfondo in nero e il colore del testo in bianco;
- *ingrandimento dell'icona del puntatore*: ingrandisce il puntatore del mouse sulla pagina Web in modo diverso se si tratta di link o di testo;

- *text-to-speech*: il testo selezionato viene tradotto in voce.

Nella figura 1.3 si può vedere l'architettura del progetto Farfalla.

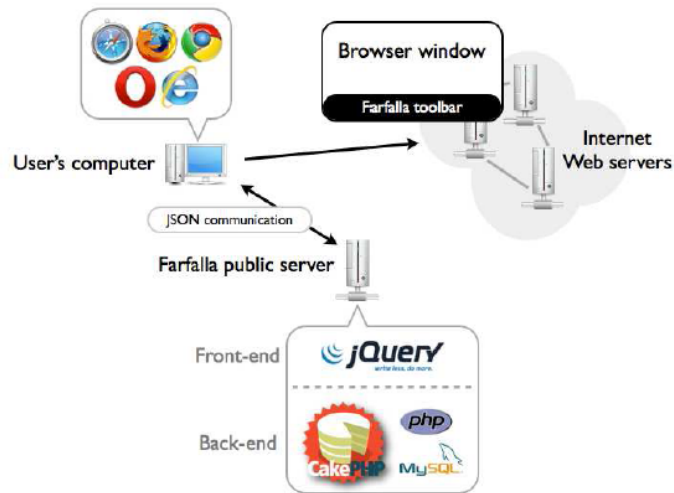


Figura 1.3: Architettura Farfalla Project



# Capitolo 2

## Progetto

Il principale obiettivo di GAPforAPE (GreaseMonkey And Profiling for Accessible Pages Enhancement) è di migliorare l'accessibilità delle pagine Web in modo automatico e dinamico modificandole a lato client. I parametri su come e dove le pagine vengono alterate sono basati sul profilo utente, impostato come un insieme di preferenze ed esigenze attraverso un'adeguata interfaccia fornita dal browser [4].

In questa piccola descrizione sono presenti tutti gli attuali principi chiave che offrono grandi potenzialità per migliorare l'accessibilità. Infatti, le specifiche di questo progetto richiedono la soddisfazione dei seguenti requisiti:

- profilazione;
- adattamento e transcodifica dei contenuti;
- augment browsing;
- crowdsourcing.

Analizzandoli uno ad uno è stata scelta l'architettura che meglio li implementasse.

## 2.1 Specifiche e scelte progettuali

### 2.1.1 Profilazione

Lo sviluppo di GAPforAPE [4] è stato guidato dall'idea che *“un singolo contenuto Web per tutti”* non apporta gli stessi benefici di *“il migliore contenuto Web per ogni uno”*. Sin dalla nascita dei primi principi di accessibilità Web, l'utilizzo di pagine Web parallele, spesso contenenti solo testo, è stato percepito come una discriminazione e segregazione delle persone con disabilità. Occorre anche sottolineare però che un'unica pagina web bensì accessibile non può fornire il migliore contenuto per ogni diversa disabilità. Per esempio, considerando l'insieme delle persone ipovedenti, ci sono all'interno tantissime sfumature e poi non bisogna mai dimenticare che ogni persona è unica e ha caratteristiche proprie anche se si può dare un nome preciso alla sua disabilità. Le tecniche e i criteri per effettuare la transcodifica dei contenuti hanno permesso di fornire una sola pagina, ma con i contenuti adattati ed ottimizzati, incontrando così le preferenze ed esigenze soggettive di ogni individuo. Ovviamente occorre un meccanismo di profilazione tramite il quale gli utenti possano impostare tali preferenze ed esigenze. Un profilo è una descrizione relativa ad un utente contenente le informazioni più rilevanti o interessanti, in base al contesto. Per effettuare la profilazione si è scelto di seguire lo standard IMS ACCLIP (Accessibility for Learner Information Package) [41]. Questo standard è stato sviluppato dal IMS Global Learning Consortium, un consorzio internazionale che sviluppa specifiche tecniche per supportare l'apprendimento attraverso le tecnologie. Come descritto in 1.4.1, IMS ACCLIP è una parte della specifica IMS LIP [42], originariamente sviluppata per descrivere i vincoli di accessibilità degli studenti. Quindi, ACCLIP descrive l'utente in termini di bisogni di accessibilità, senza considerare le caratteristiche del dispositivo. In particolare, ACCLIP permette la descrizione delle preferenze dell'utente (visiva, sonora o dispositivo) che possono essere sfruttate per la personalizzazione dei contenuti (per esempio gli input device preferiti/richiesti o contenuti preferiti alternativi).

Nel nostro sistema di profilazione abbiamo preso in considerazione solo gli attributi appartenenti alle sezioni riguardanti:

- le informazioni di visualizzazione;
- le informazioni di controllo;
- le informazioni di contenuto.

In particolare si è scelto di raggruppare le preferenze e bisogni informativi in:

- colori;
- testo;
- audio;
- visivo.

Tutte queste opzioni, adeguatamente configurate, andranno a formare il profilo dell'utente, tramite il quale potrà quindi esprimere le sue necessità.

Viene anche fornito agli utilizzatori un meccanismo per modificare il proprio profilo aggiungendo funzionalità, ovvero manipolazioni o modifiche che l'utente ha necessità di effettuare su tutte le pagine e che non sono fornite di default nel nostro meccanismo di profilazione (che verrà descritto in modo dettagliato in 2.1.4).

### 2.1.2 Adattamento e transcodifica dei contenuti

Il sistema deve permettere l'adattamento e la transcodifica dei contenuti. L'adattamento dei contenuti è l'attività di trasformazione e manipolazione dei contenuti, come immagini, audio, video, testo e presentazione, per raggiungere gli obiettivi desiderati, definiti dalle capacità del terminale e dalle esigenze delle applicazione [56]. Questi adattamenti includono:

- transcodifica del formato: per esempio da eXtensible Markup Language (XML) a HyperText Markup Language (HTML) oppure da Scalable Vector Graphics (SVG) a Graphics Interchange Format (GIF);
- ridimensionamento di immagini o di video o di stream video;
- conversione di media: per esempio Text-To-Speech;
- ricampionamento;
- comprensione della dimensione dei file;
- frammentazione del documento.

Transcodifica (transcoding) è il processo di conversione di un file multimediale o di un oggetto multimediale da un formato multimediale ad un altro. Questo processo è tipicamente utilizzato per convertire formati video, audio e immagine, ma è anche usato per adattare presentazioni multimediali e pagine Web ai vincoli di dispositivi non standard, per esempio per i dispositivi mobili. La maggior parte delle presentazioni multimediali e delle pagine web esistenti sono create per essere visualizzate su computer desktop e, di solito, i progettisti Web forniscono contenuti complessi e ricchi di dettagli, con esperienze multimediali [56]. La grande eterogeneità dei dispositivi impone un stato intermedio di adattamento dei contenuti per garantire una corretta presentazione su ogni dispositivo di destinazione

Le operazioni di adattamento dei contenuti e di transcodifica su un singolo media possono essere riassunte come segue.

- *Trasformazione*: la conversione dei contenuti dalla sua forma originale ad un'altra. A seconda del tipo di conversione, le trasformazioni possono essere eseguite automaticamente, un esempio è il Text-To-Speech (TTS); oppure hanno bisogno di una dichiarazione predefinita esplicita di contenuti equivalenti (operazione effettuata offline), come nel caso di trasformazioni da immagini a testo. La conversione può essere effettuata anche tra formati di codifica dello stesso tipo, per esempio file audio da formato WAV a formato MP3.

- *Ridimensionamento*: ricodifica e/o compressione di specifici contenuti multimediali. Il ridimensionamento ha effetti in termini di riduzione della grandezza, qualità e data rate dei contenuti. Esempi sono il ridimensionamento di immagini e video; ricodifica e compressione di video.
- *Traduzione*: dal linguaggio originale ad uno differente, in base al profilo utente. Questa operazione può essere fatta solo per contenuti testuali e audio speech.

Dal punto di vista dell'architettura l'adattamento e la transcodifica dei contenuti può essere attuata in quattro modalità [4], [56], [2]:

- *approccio client-side*: il responsabile della transcodifica e dell'adattamento dei contenuti è l'applicazione client.

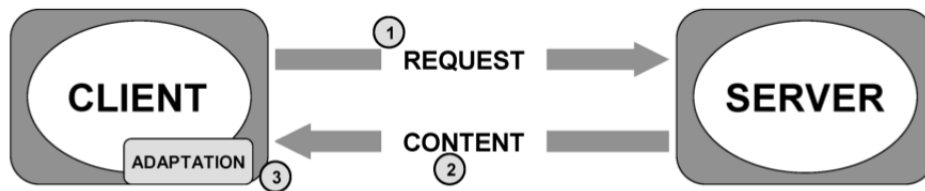


Figura 2.1: Adattamento dei contenuti client-side (disponibile in [2])

Le soluzioni client-side possono essere divise in due categorie in base al comportamento:

- i) il client riceve differenti formati e seleziona il formato più appropriato da utilizzare;
- ii) il client computa una versione ottimizzata partendo dalla standard. Questo approccio suggerisce una soluzione distribuita per la gestione della eterogeneità, supponendo che tutti i client possano decidere localmente e utilizzare l'adattamento più appropriato per loro.

- *approccio server-side*: il server che fornisce i contenuti effettua le funzionalità aggiuntive di adattamento dei contenuti. L'adattamento può essere effettuato:
  - i) offline: la transcodifica del contenuto viene effettuata ogni volta che una nuova risorsa viene creata o una esistente viene aggiornata sul server; solitamente uno sviluppatore effettua le modifiche "a mano" per adattare il contenuto ai differenti profili specifici; i vari formati rimangono memorizzati nel server e selezionati dinamicamente in base alle richieste del client.
  - ii) on-the-fly (sul momento): il contenuto viene adattato dinamicamente prima di essere consegnato al client.

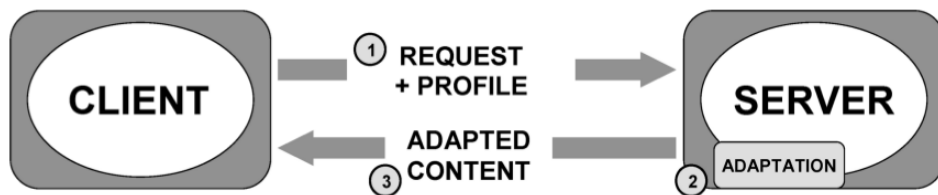


Figura 2.2: Adattamento dei contenuti server-side (disponibile in [2])

- *approccio proxy-based*: il processo di adattamento viene effettuato da un nodo (per esempio il proxy), che è posizionato tra il server e il client. In pratica il proxy intercetta le risposte del server al client, ed effettua tre azioni:
  1. decide se occorre effettuare dei miglioramenti;
  2. effettua l'adattamento dei contenuti;
  3. invia i contenuti adattati al client.

Per poter effettuare questo task, il proxy deve conoscere il dispositivo target, le capacità dell'utente (informazione che può essere fornita dal

client) e la versione completa del contenuto originale (che gli viene fornito dal server). Ciò può portare ad un alto consumo (bandwidth) della rete che collega il server con il proxy.



Figura 2.3: Adattamento dei contenuti proxy-based (disponibile in [2])

- *approccio service-oriented*: la natura dinamica del meccanismo di adattamento unito con l'opportunità offerta dalla tecnologia Web Service, permette un approccio service-oriented per l'adattamento dei contenuti. La filosofia alla base di questo approccio è fondamentalmente differente dagli approcci precedenti dal momento che le attività di transcodifica e adattamento sono organizzate seguendo l'architettura service-oriented. Inoltre, il numero delle tipologie di adattamento dei contenuti così come il numero di formati e dei relativi schemi di conversione. Questo dinamismo è una delle ragioni che rendono difficile sviluppare un unico sistema di adattamento che può accomodare tutti i tipi di adattamento; di conseguenza, servizi di adattamento di terze parti sono importanti.

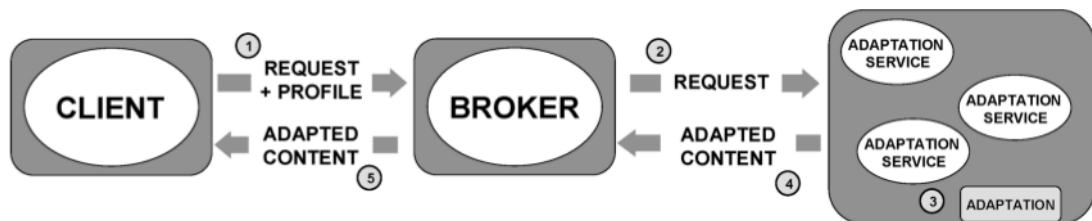


Figura 2.4: Adattamento dei contenuti service-oriented (disponibile in [2])

Alla luce delle architetture appena descritte, si è scelto di adottare il metodo più semplice che presenta più vantaggi: quello client-side. Questo approccio sfrutta un'architettura consolidata e che non deve essere modificata. Ancor più importante è che il server non deve essere a conoscenza dei profili e delle capacità dell'utente, informazioni che rimangono memorizzate localmente nel client. Ciò è fondamentale per il nostro intento, che è quello di adattare il contenuto di tutte le pagine in modo completamente trasparente allo sviluppatore o al fornitore di contenuti, che rimane così completamente estraneo agli adattamenti e alle transcodifiche necessarie. Inoltre, l'approccio client-side di adattamento dei contenuti può essere facilmente implementato nei browser Web attuali, attraverso l'uso di estensioni.

### 2.1.3 Augment browsing

L'augment browsing descrive l'esperienza di utilizzo di un sistema che può automaticamente aumentare o migliorare l'informazione nelle pagine web. Tramite l'augment browsing l'utente finale può scegliere come e se personalizzare il modo in cui visualizzare i documenti finali.

L'augment browsing può essere effettuato principalmente in due modi:

- *user scripts*: sono “pezzi” di codice scritti in JavaScript che aggiungono funzionalità utili alle applicazioni e alle pagine web. Per poterli utilizzare, occorre installare estensioni per il proprio browser che li supportino (ad eccezione che per Google Chrome e Opera che li supportano nativamente). In teoria possono essere utilizzati in ogni browser (tramite le estensioni), in pratica problemi possono esserci a causa dei diversi modi in cui i vari browser implementano JavaScript: questi problemi li rendono quindi “browser-specific” ovvero creati specificatamente per un browser. Gli script possono riguardare funzioni generali (come l'ingrandimento del testo o il cambio del colore di sfondo) oppure essere specifici per certe pagine. Vengono automaticamente invocati nelle pagine web senza il bisogno dell'utente di attivarli e possono quindi modificarne aspetto e comportamento.



- *bookmarklets*: sono piccole “porzioni” di codice JavaScript non intrusivo salvato come un URL<sup>1</sup> di un bookmark<sup>2</sup> (“i segnalibri”) in un browser o come un link di una pagina web. Il termine deriva dall’unione di bookmark e applet. Essi permettono di aggiungere funzionalità alle pagine Web attraverso il click del bookmark o del link; quando si effettua il click il JavaScript viene eseguito sulla pagina corrente e può ispezionarla e modificarla. I browser rendono a disposizione il prefisso “javascript” che per il parser è uguale a qualsiasi URL, così può essere creato il bookmark o salvato in un link; internamente, però, il browser vede che si tratta del protocollo “javascript” e quindi lo esegue quando l’utente fa click.

Un grosso svantaggio di quest’ultima tecnica è che richiede una continua interazione con l’utente, infatti lo script non viene eseguito in modo automatico ma è l’utente ad attivarlo attraverso il click del bookmark o del link. Se lo script è specifico per un certo sito web, o per una certa tipologia di pagine web, l’attivazione non richiede un’interazione costante con l’utente ma è comunque l’utente finale che deve ricordarsi di attivarlo. Inoltre, non “ha memoria” nel senso che se si riapre la stessa pagina dopo averla chiusa precedentemente, l’utente deve nuovamente effettuare il click per poter eseguire di nuovo lo script. La situazione si complica notevolmente se la modifica deve essere effettuata su tutte le pagine Web visitate, come per esempio l’ingrandimento del testo, il cambio di colore dello sfondo o l’eliminazione delle immagini di sfondo; è un problema non trascurabile perché costringe l’utente a dover interagire con ogni pagina. Per un qualsiasi utente può rappresentare solo una scocciatura, ma per persone con disabilità, un meccanismo del genere diventa davvero problematico.

Un altro svantaggio riscontrato è che, dal momento che la nostra idea è quella di permettere il caricamento automatico di profili (standard o modifi-

---

<sup>1</sup>Uniform Resource Locator (URL) è una sequenza di caratteri che identifica univocamente l’indirizzo mnemonico di una risorsa in Internet.

<sup>2</sup>I bookmarks permettono di memorizzare URL di una particolare pagina web o di un file visualizzabile tramite browser, rendono più veloce la consultazione,

cati dall'utente), il bookmarklet per poterlo permettere dovrebbe comunicare con un server che invia le giuste modifiche in base al profilo scelto (ciò che avviene in Farfalla [55]). Ciò viola la scelta di fare l'adattamento di contenuti attraverso l'approccio client-side.

Tutte queste considerazioni ci hanno portato a scegliere di utilizzare user scripts per implementare l'augment browsing a favore dell'accessibilità. Come precedentemente detto, per essere utilizzati hanno bisogno di un'estensione che li supporti, se non sono supportati nativamente.

I browser maggiormente diffusi sono di seguito elencati.

- *Windows Internet Explorer* [57]: web browser proprietario usabile solo su sistemi operativi Microsoft Windows. Per supportare gli user scripts si può installare l'add-on IE7Pro che fornisce diverse funzionalità tra le quali anche la possibilità di eseguire user scripts; può essere installato su IE (Internet Explorer) 6, 7, 8 e 9; ovvero le ultime versioni del browser.
- *Mozilla Firefox* [58]: web browser open source multiplatforma prodotto da Mozilla Foundation<sup>3</sup>. La filosofia progettuale punta alla facilità d'uso, alla stabilità, alla personalizzazione, al rispetto degli standard web, alla sicurezza, alla compattezza e alla velocità. Firefox è stato progettato come prodotto multiplatforma, ed è disponibile per sistemi GNU/Linux, Microsoft Windows, Mac OS X, OS/2 e Solaris. Supporta gli user scripts attraverso l'estensione, molto famosa e diffusa, Greasmonkey [45] (come anticipato in 1.5.1).
- *Google Chrome* [59]: browser sviluppato da Google e può essere installato su Windows, Linux e Mac OS X. L'obiettivo principale che ha portato Google a sviluppare Chrome è la ricerca di una maggiore sicurezza, velocità e stabilità rispetto ai browser esistenti, utilizzando un'interfaccia con molte novità rispetto agli standard precedenti.

---

<sup>3</sup>organizzazione no-profit che promuove l'apertura, l'innovazione e la partecipazione in Internet, di cui lo stesso Firefox

Supporta nativamente gli user scripts quindi basta solo scaricarli e il browser riconosce che sono script e li esegue su ogni pagina.

- *Opera* [60]: è il web browser gratuito creato da Opera Software, disponibile per i sistemi operativi Windows, Mac OS X e Linux. Anche esso supporta nativamente gli user scripts anche se il meccanismo che poterli attivare è un po' più macchinoso rispetto a Chrome.
- *Safari* [61]: è il browser sviluppato da Apple Inc.; è nato per il sistema operativo Mac OS X e successivamente ne è stata creata una versione anche per Windows. Safari non permette di installare scripts nella versione di Windows mentre nella versione per Mac si possono installare pacchetti che permettono di utilizzare Greasemonkey [45].

Nel settembre scorso, Wikimedia<sup>4</sup> ha analizzato i report di traffico relativi a circa 4 miliardi di richieste al server al mese, ottenendo i valori riportati in figura 2.5 [3].

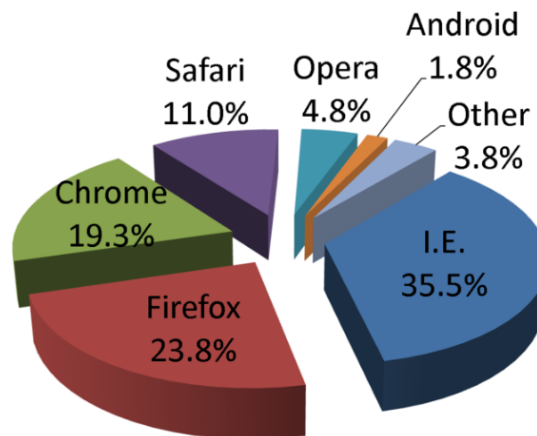


Figura 2.5: Diffusione dei vari browser secondo Wikimedia nel settembre 2011 (disponibile in [3])

<sup>4</sup>Wikimedia è posseduto e gestito dalla Wikimedia Foundation, una fondazione no-profit dedicata a portare contenuti gratuiti per il mondo.

Come si può ben vedere, Internet Explorer [57] è il browser con la percentuale di utilizzo più alta, a causa della grande diffusione del sistema operativo Windows in cui è installato di default. Tuttavia, il browser Firefox [58] ha raggiunto un'alta percentuale in questi anni, diventando il secondo browser più utilizzato al mondo. Questo risultato è molto importante perché gli utenti che scelgono Firefox lo fanno in modo volontario, dichiarando la loro preferenza verso questo browser. Molto importante è anche il fatto che sia un browser multiplatforma e quindi installabile su tutti i sistemi operativi più diffusi senza costringere all'utilizzo di uno in particolare. Inoltre, Firefox è stato il primo browser a permettere la creazione e l'installazione di estensioni, piccoli add-on (programmi supplementari) che permettono di aggiungere funzionalità, anche complesse. Lo scopo principale per il quale sono state pensate è quello di personalizzare la propria versione dell'applicazione. Sul sito ufficiale Mozilla Add-ons [62] è possibile condividere e scaricare un numero elevato di estensioni per una grande varietà di scopi. L'installazione è molto semplice e richiede un solo click da parte dell'utente.

Google Chrome [59], nonostante sia il più recente (la prima versione è stata rilasciata nel 2008) ha già raggiunto una percentuale davvero rilevante. Anche questo browser ha a suo favore il fatto di essere installabile su tutti i sistemi operativi più diffusi. Tuttavia, considerando anche la sua recente nascita, non è noto a molti, in particolare è utilizzato dalle persone con più dimestichezza con l'uso del computer e più attento alle novità. Il tipico utente che utilizza Windows e ha scelto di passare a Firefox, dopo vari anni di utilizzo di IE, magari consigliato da un amico, e con qualche fatica, non rinuncerà alla nuova conoscenza acquisita per cambiare nuovamente, o per lo meno non lo farà nel breve periodo (ciò che in economia viene chiamato effetto *lock-in*<sup>5</sup>). Va anche detto che probabilmente Chrome è ad oggi il browser più valido e con più potenzialità e sicuramente nel giro di pochi anni la sua percentuale di utilizzo aumenterà notevolmente.

---

<sup>5</sup>L'effetto lock-in si verifica quando i costi associati alla transizione da una tecnologia all'altra sono molto rilevanti per l'utente

Safari [61] ha notevolmente aumentato la sua quota di mercato con la grande diffusione che hanno avuto in questi anni i prodotti Apple, in cui, ovviamente, è installato di default. Tuttavia non è multiplatforma e quindi il suo utilizzo è relativo alla “nicchia” dei clienti Apple.

Opera [60], nonostante sia disponibile per tutti i sistemi operativi più diffusi, ha una percentuale di utilizzo ancora troppo bassa per essere preso in considerazione, visto che non è certo di recente nascita (lo sviluppo del progetto è iniziato nel 1994).

Per tutte le considerazioni precedenti, si è optato per la creazione di un'estensione per Firefox che permetta di fare *augment browsing* delle pagine caricate dal browser in modo automatico una volta definito un profilo.

#### 2.1.4 Crowdsourcing

Il crowdsourcing ha aiutato a far diffondere l'utilizzo di GreaseMonkey [45]. Infatti, questa estensione ha permesso di eseguire gli *user scripts* in Firefox, ma la vera potenza del meccanismo sono le persone che hanno creato e creano tutt'oggi gli *script* e li condividono con gli altri utenti. Senza gli *script*, GreaseMonkey non potrebbe di fatto offrire tutte le potenzialità di cui abbiamo precedentemente parlato. Si è quindi pensato di fornire un meccanismo simile a quello di GreaseMonkey in modo da permettere alla comunità di creare i propri *script*, così che gli utenti possano mettere in pratica le loro preferenze. Potranno farlo sia creando *script* “generalisti” con manipolazioni o modifiche potenzialmente applicabili a tutte le pagine visitate; ma sarà data loro anche la possibilità di creare *script* specifici per un particolare sito facendo in modo che vengano eseguiti solo su certe pagine (o addirittura solo su un certo indirizzo).

Non va dimenticato che GAPforAPE [4] ha l'innovazione di permettere agli utenti di configurare inizialmente il proprio profilo fornendogli la possibilità di settare tutte le manipolazioni più comuni per il miglioramento dell'accessibilità, in modo che ogni pagina visualizzata rifletta le particolarità del proprio profilo. In questo modo, l'utente non deve obbligatoriamente

ricercare in rete lo script (che solitamente da solo non copre mai le esigenze dell'utente, quindi ne occorrono vari) ed installarlo; molte funzionalità gli vengono già fornite.

Un'altra innovazione è stata quella di rendersi conto che ogni persona ha le sue caratteristiche e i suoi bisogni, ma anche in senso più tecnico, che certe modifiche su alcune pagine non danno proprio i risultati voluti. In questi casi, l'utente come può migliorare la situazione? Può cercare in giro qualcuno che, avendo avuto lo stesso problema, ha creato ad hoc uno script per quella pagina, e quindi l'ha condiviso. Anche ammesso che lo trovi, deve poter funzionare con il proprio browser e la propria estensioni, e deve verificare che il risultato ottenuto sulla pagina sia quello cercato. Come si può ben capire, è un meccanismo lungo che spesso non si conclude come sperato, inoltre occorre pensare che, se fatto da persone con disabilità, può diventare talmente problematico da fargli scegliere di rinunciare direttamente o, nel caso limite, scrivarsi lo script. Dal momento che, le persone che hanno conoscenze tali da poter scrivere il codice che rifletta le loro necessità sono un numero molto ristretto, si è deciso di permettere a chiunque di creare in un modo molto semplice le proprie modifiche su qualsiasi pagina. Un requisito su cui si è molto lavorato è stato appunto quello di rendere l'utente capace di modificare on-the-fly la pagina visualizzata, nel modo più intuitivo e semplice possibile, e poi salvarsi le modifiche. I cambiamenti potranno essere salvati in modo che vengano eseguiti solo su quella specifica pagina (o sito) oppure che vengano eseguiti su qualsiasi pagina andando così a modificare il profilo dell'utente stesso. Anche questo tipo di script potrà essere condiviso con la comunità ed utilizzato quindi da altri utenti.

Riassumendo, si potrà quindi sviluppare e condividere user script creati sia da utenti esperti di linguaggi di scripting che da qualsiasi altro utente per modificare aspetti generali (ovvero relativi a tutte le pagine, e di conseguenza è una sorta di modifica del profilo utente) o aspetti specifici di alcune pagine.

## 2.2 Architettura

L'idea del sistema è quella di modificare il contenuto Web usando linguaggi di scripting client-side. Si è così deciso di progettare GAPforAPE come un'estensione del browser Firefox che permette una transcodifica dei contenuti a lato client mediante:

- un sistema di profilazione dell'utente in cui esso specifica le sue preferenze e necessità;
- una collezione, liberamente incrementabile e modificabile, di user script, scaricabili dalla rete o creabili dall'utente stesso.

Al fine di migliorare l'accessibilità dei contenuti Web e per fornire il miglior adattamento ad ogni utente, rispondendo alle sue esigenze e preferenze, i nostri script permettono la transcodifica delle pagine Web modificando le regole CSS (Cascading Style Sheets) e DOM HTML. Le modifiche applicate dai nostri script verranno descritte più in dettaglio nel capitolo seguente.

La figura 2.6 mostra in modo molto intuitivo l'architettura di GAPforAPE. Essa è principalmente composta da un'estensione per il browser Firefox che permette di configurare il proprio profilo ed effettuare l'adattamento e la transcodifica dei contenuti.

La principale differenza tra la nostra estensione e le altre estensioni per il browser Web che permettono di eseguire user script, come GreaseMonkey e AccessMonkey, è il sistema di profilazione. Il profilo viene completamente configurato dall'utente e rimane memorizzato localmente, solo lui sa quali sono i suoi bisogni e necessità e queste informazioni verranno utilizzate solo dall'estensione nel pieno rispetto della riservatezza dei dati, sottolineando che non c'è nessuna correlazione tra il profilo e l'utente dal momento che non viene richiesta nessuna informazione sensibile (come nome, cognome, ecc.). Una volta settate le varie opzioni, il sistema provvederà automaticamente ad adattare il contenuto di tutte le pagine visualizzate nel browser trasformando il profilo dell'utente in linguaggio di scripting che verrà eseguito attraverso

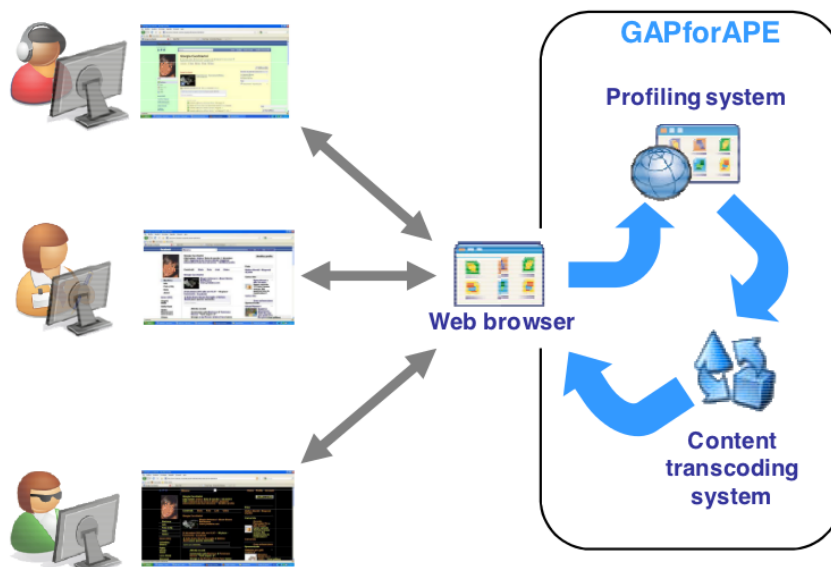


Figura 2.6: Architettura di GAPforAPE (disponibile in [4])

l'estensione sulle pagine Web. Ciò permette al nostro sistema di migliorare l'accessibilità di tutte le pagine visualizzate nel browser e anche di tutte le applicazioni Web che sono fortemente basate su tecnologie Web 2.0, come Facebook e altri famosi e ampiamente utilizzati social network. L'utilizzo di tecnologie Web 2.0 colpisce davvero l'accessibilità di tali applicazioni Web, come descritto in 1.3. Va comunque precisato che non sempre è possibile automaticamente riconoscere e adattare qualsiasi script AJAX in modo efficace ed efficiente. Di conseguenza, abbiamo progettato il nostro sistema in modo da individuare le applicazioni Web e di transcodificarle utilizzando uno specifico insieme di script. Questo meccanismo è stato ispirato dai più diffusi screen reader (come per esempio Jaws) che, in base all'applicazione che devono interpretare, prima controllano se sono presenti regole specifiche per quell'applicazione altrimenti attuano le regole generali. In modo particolare, GAPforAPE sfrutta un sistema a due livelli: se sono presenti specifici script relativi a quell'applicazione Web o pagina Web esegue quelli altrimenti



applica gli script generali ovvero quelli creati in base al profilo dell'utente. Quindi, una pagina web viene inviata nello stesso modo a tutti i browser ed è poi la nostra estensione che effettua automaticamente l'adattamento e la transcodifica dei contenuti a lato client, modificando il codice HTML, le regole CSS e gli script. Questa soluzione su due livelli permette al sistema di transcodifica di identificare e adattare automaticamente gli script AJAX, a differenza degli script generali che non sono in grado di gestire efficacemente tale tecnologia.

## 2.3 Tecnologie utilizzate

Le estensioni per il browser Firefox possono essere create in due differenti modi:

- con il metodo tradizionale;
- attraverso Add-on SDK [63].

Come descritto in [4], il primo prototipo di GAPforAPE lo abbiamo creato utilizzando il metodo tradizionale che utilizza principalmente i seguenti linguaggi:

- *XUL* (XML User Interface Language): un linguaggio basato su XML per descrivere le interfacce grafiche delle applicazioni di Mozilla;
- *JavaScript*: linguaggio di scripting client-base.

Con l'inizio dei lavori di rinnovamento del browser, che hanno portato al rilascio di Firefox 4 (versione che prometteva di rivoluzionare il browser), è stato iniziato anche un progetto chiamato Jetpack [64] il cui obiettivo è rendere più facile la creazione degli add-ons. Add-on è il termine generico con cui si intende un pacchetto software che aggiunge funzionalità alle applicazioni di Mozilla, in cui rientrano ovviamente anche le estensioni per Firefox.

Il progetto Jetpack è composto da due prodotti, che sfruttano le novità e i cambiamenti presenti in Firefox 4:

- *Add-on SDK*: ambiente di sviluppo memorizzato localmente che permette di creare facilmente add-ons per Firefox;
- *Add-on Builder*: ambiente di sviluppo online che permette di creare e memorizzare gli add-ons utilizzando solo il proprio browser.

Con l'uscita di Firefox 4 e della versione 1 dell'Add-on SDK si è scelto di ricreare il prototipo viste le grandi potenzialità che offriva. Infatti, l'SDK permette di creare add-ons usando le tecnologie Web standard:

- JavaScript e altre librerie JavaScript come JQuery [65];
- HTML;
- CSS;

Ciò offre agli sviluppatori un'incredibile libertà e la possibilità di creare vere e proprie applicazioni Web, utilizzando i linguaggi più diffusi e potenti, che, tramite i moduli forniti dell'SDK possono facilmente interagire con il browser e permettere uno scambio di informazioni tra le due componenti.

L'Add-on Builder non è stato preso in considerazione data la complessità e le dimensioni del nostro sistema, per il quale si è preferito utilizzare sistemi di versioning per gestire nel modo ottimale l'avanzamento dello sviluppo.

### 2.3.1 Add-on SDK

L'Add-on SDK include:

- un set di moduli che forniscono API JavaScript che possono essere usate per creare add-ons. Questi moduli semplificano i task come creare un'interfaccia utente e interagire con il Web, e garantiscono all'estensione di continuare a funzionare anche con il cambio di versione di Firefox;

- un insieme di tools per creare, eseguire, testare e fare il “packaging” dell’add-on, in un ambiente sicuro (“sandbox”).

Anche quest’ultimo punto rappresenta una rivoluzione per gli sviluppatori di estensioni. Infatti, utilizzando il metodo tradizionale, l’unico modo per poter provare l’add-on creata era eseguirla su un’istanza del proprio browser. Ciò significa che per possibili errori si andava ad intaccare il proprio browser e, soprattutto, non c’era nessun meccanismo per rilevare qual’era stata la causa del problema. In questo kit invece si può eseguire l’estensione su un’istanza del browser creata appositamente per il testing, è anche presente un buon sistema di controllo e notifica degli errori e la possibilità di crearsi i propri log in maniera davvero semplice. Questi meccanismi vengono forniti attraverso:

- *console*: oggetto che permette di registrare messaggi utili per avere il log delle azioni che l’estensione compie;
- *ctx*: strumento a linea di comando che consente di accedere alla documentazione dell’SDK e permette di eseguire, testare, e costruire il pacchetto che andrà poi eseguito sul browser.

Per ciò che riguarda il primo punto, invece, sono presenti vari moduli che forniscono molti dei servizi solitamente usati nella creazione di estensioni. Questi moduli sono:

- *clipboard*: consente ai “chiamanti” di interagire con il sistema di clipboard, impostando e recuperando il suo contenuto;
- *context-menu*: permette di aggiungere elementi al menu contestuale nella pagina;
- *hotkeys*: permette di creare “keyboard shortcut”, ovvero tasti di scelta rapida, per certe operazioni;
- *notifications*: permette di visualizzare messaggi transitori all’utente;
- *page-mod*: permette agli sviluppatori di eseguire script nel contesto di una specifica pagina, potendola quindi modificare dinamicamente;

- *page-worker*: fornisce il meccanismo per creare una permanente, invisibile pagina e accedere ad essa attraverso il suo DOM;
- *panel*: crea una finestra di dialogo pop-up che appare sopra i contenuti Web ed è persistente fino a che non viene dismessa dagli utenti o dai programmi;
- *passwords*: permette all'add-on di interagire con il "Password Manager" di Firefox per aggiungere, richiamare, rimuovere le credenziali memorizzate;
- *private-browsing*: permette di accedere alla modalità di navigazione privata di Firefox, individuando se è attiva e quando avvengono cambiamenti di stato;
- *request*: consente di effettuare richieste di rete in modo semplice ma potente;
- *selection*: fornisce un meccanismo per ottenere ed impostare il testo e le selezioni HTML nella pagina corrente;
- *self*: fornisce l'accesso ai dati interni all'add-on stesso e fornisce l'accesso al "Program ID", valore unico per ogni componente aggiuntivo;
- *simple-storage*: consente di memorizzare i dati in modo persistente;
- *tabs*: fornisce un facile accesso alle schede e ai loro eventi;
- *timers*: consente di accedere alle funzionalità Web di temporizzazione;
- *widget*: permette di accedere alle funzionalità dell'add-on attraverso una semplice interfaccia utente;
- *windows*: fornisce un facile accesso alle finestre del browser, le loro schede e alle correlate funzioni di apertura e chiusura.

Le alte tecnologie coinvolte nel nostro sistema sono: JQuery e JavaScript, CSS e HTML.

### 2.3.2 HTML

HTML (HyperText Markup Language) è il linguaggio di markup per ipertesti predominante nelle pagine Web. La sua sintassi è stabilita dal W3C. È stato sviluppato alla fine degli '80 da Tim Berners-Lee al CERN (Organizzazione Europea per la Ricerca Nucleare) di Ginevra assieme al noto protocollo HTTP che supporta invece il trasferimento di documenti in tale formato. Verso il 1994 ha avuto una forte diffusione in seguito ai primi utilizzi commerciali del Web. Nel corso degli anni ha subito molte modifiche e miglioramenti raggiungendo oggi la versione 4.01, rilasciata il 24 dicembre 1999 [66]. Dal 2007 il W3C ha ripreso l'attività di specifica con la definizione, ancora in corso, di HTML 5. Caratteristica di HTML è che esso è stato concepito per definire il contenuto logico e non l'aspetto finale del documento. Attualmente i documenti HTML sono in grado di incorporare molte tecnologie, che offrono la possibilità di aggiungere al documento ipertestuale effetti grafici, interazioni dinamiche con l'utente, animazioni interattive e contenuti multimediali [67].

### 2.3.3 CSS

Il CSS (Cascading Style Sheets o Fogli di stile) è un linguaggio usato per definire la formattazione di documenti HTML, XHTML e XML. Le regole per comporre il CSS sono contenute in un insieme di direttive emanate a partire dal 1996 dal W3C. L'introduzione del CSS si è resa necessaria per separare i contenuti dalla formattazione, permettendo così una gestione separata della presentazione dai contenuti veri e proprio, ricordando che, come specificato sopra, HTML era nato solo per descrivere il contenuto logico e non l'aspetto finale del documento. Ciò permette di presentare lo stesso contenuto in differenti modi, permettendo di scegliere un diverso stile in base ai bisogni specifici e ai dispositivi utilizzati dagli utenti. L'ultima revisione relativa alla versione 2.1 è stata effettuata nel giugno scorso (la direttiva 2.0 risale invece al 1999). La versione 3 è in via di sviluppo.

### 2.3.4 JQuery

JQuery è una veloce, potente, gratuita, open source e cross-browser libreria JavaScript, sviluppata per semplificare lo scripting lato client di HTML[65]. È stata rilasciata nel gennaio 2006 al BarCamp<sup>6</sup> NYC da John Resig. Oggi è la libreria JavaScript più utilizzata. Permette di navigare facilmente il DOM HTML, selezionare elementi, creare animazioni, gestire eventi e sviluppare applicazioni Ajax.

Nel nostro sistema è stato usato JQuery perché permette una gestione completa, veloce e potente del DOM HTML. Il DOM (Document Object Model) HTML è uno standard W3C per accedere ai documenti HTML, definendo oggetti e proprietà per tutti gli elementi HTML del documento e i metodi per accedervi. In altre parole, è lo standard per ottenere, cambiare, aggiungere o cancellare elementi HTML. GAPforAPE è incentrato sulla transcodifica e adattamento dei contenuti della pagine Web e questo meccanismo può essere effettuato proprio attraverso una complessa manipolazione del DOM dei documenti HTML che il browser riceve.

JQuery permette anche di manipolare, in modo semplice e veloce, il CSS associato agli elementi HTML, aggiungendo, modificando, rimuovendole le regole, funzionalità molto utilizzata nel nostro sistema di adattamento e transcodifica dei contenuti.

---

<sup>6</sup>BarCamp è una rete internazionale di conferenze generate dagli utenti.

## Capitolo 3

# Implementazione

Come anticipato nel capitolo precedente, GAPforAPE è stato implementato come un'estensione per il browser Web Firefox. Il primo prototipo era compatibile con Firefox 3.6. Con la release di Firefox 4 (avvenuta a fine marzo) si è iniziato a sviluppare l'add-on utilizzando Jetpack e in particolare la versione 1.0b5 dell'SDK. Ad oggi, il browser ha subito rapidi (e non ordinari) avanzamenti di versione arrivando alla versione 7.0.1 (rilasciata a fine settembre) e, questa spinta di miglioramento ancora non si placa, infatti si attende per l'8 novembre la versione 8. L'add-on è stato comunque reso compatibile per tutte le versioni, dalla 4 a quella attuale. Contemporaneamente, l'SDK è arrivata alla versione 1.2.1 (rilasciata lo scorso 19 ottobre). A differenza della compatibilità con le varie versioni del browser che possono compromettere il funzionamento del componente aggiuntivo, l'SDK non crea problemi di questo tipo dal momento che i moduli della stessa vengono inseriti nell'add-on nel momento dell'“impacchettamento”. Considerando però, che problemi potrebbero comunque essere dovuti a cambiamenti del motore di Firefox (Gecko) che comunica direttamente con i moduli dell'SDK, si è scelto di rendere il sistema compatibile anche con questa nuova versione.

Utilizzando Jetpack non esiste più un supporto semi-automatico per la localizzazione (presente nel metodo tradizionale) si è quindi optato per creare il prototipo con testo in lingua inglese.

## 3.1 Tipi di script

Per creare un add-on si fa uso di due differenti tipi di script:

- *add-on code*;
- *content script*;

entrambi utilizzano il linguaggio JavaScript ma si differenziano per:

- lo scopo per il quale sono scritti;
- le cartelle in cui vanno inseriti;
- gli insiemi di API alle quali possono accedere.

### 3.1.1 Add-on code

*Add-on code* identifica il codice tramite il quale creare e sviluppare la logica principale dell'estensione. Un add-on è implementato tramite una collezione di uno o più moduli CommonJS. Ogni modulo viene fornito tramite uno script memorizzato nella cartella `lib`. È in questa cartella che si trova infatti il “cuore” del componente aggiuntivo, il file `main.js`. Da questo file si possono poi richiamare gli altri moduli tramite la funzione `require()`.

*CommonJS* è l'infrastruttura sottostante sia dei moduli dell'Add-on SDK che dell'estensione stessa. Definisce le specifiche per:

- moduli;
- package (insiemi di moduli).

### Moduli CommonJS

Un modulo CommonJS è un pezzo di codice JavaScript riusabile, infatti permette di esportare alcuni oggetti che possono così essere utilizzati da codice esterno. Per farlo definisce:



- un oggetto chiamato `exports` che contiene tutti gli oggetti che un modulo CommonJs vuole rendere disponibili ad altri moduli;
- una funzione chiamata `require` attraverso la quale un modulo può importare gli oggetti `exports` degli altri moduli.

La figura 3.1 rende molto più chiara interazione tra gli oggetti `exports` e la funzione `require`.

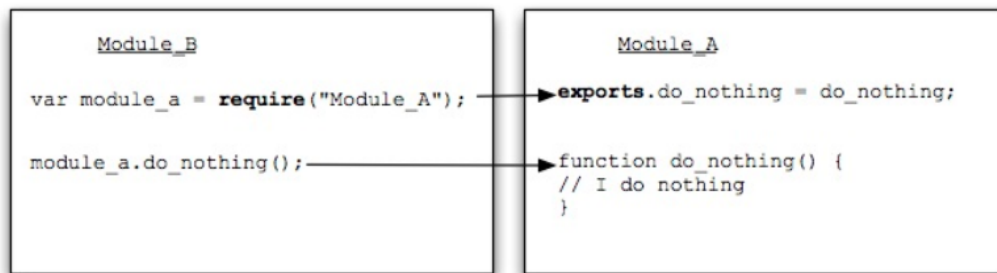


Figura 3.1: Esempio di utilizzo dei moduli CommonJS (disponibile da [5])

L'SDK congela gli oggetti `exports` restituiti dall'invocazione della funzione `require` in modo tale da non permettere di modificare le proprietà di tali oggetti.

### Package CommonJS

Un package CommonJS è una struttura che racchiude una collezione di moduli e permette di distribuire, installare e gestire in modo più semplice i moduli.

I moduli di un add-on costituiscono un package che prende il nome del componente stesso.

L'Add-on SDK è formato da due package:

- **add-on kit**: fornisce API di alto livello che possono essere utilizzate dagli sviluppatori per creare in modo semplificato le loro esten-

sioni; sono contenuti in esso tutti i moduli che permettono l'interazione diretta con il browser: `clipboard`, `context-menu`, `hotkeys`, `notifications`, `page-mod`, `page-worker`, `panel`, `passwords`, `request`, `private-browsing`, `selection`, `self`, `simple-storage`, `tabs`, `timers`, `widget`, `windows` (dei quali è stata fornita una breve descrizione in 2.3.1);

- `api-utils`: fornisce un'infrastruttura CommonJS di base per sviluppare add-on e applicazioni XULRunner; è la base per l'Add-on SDK.

I moduli dei vari package possono comunicare tra di loro sfruttando il meccanismo descritto prima, degli oggetti `exports` e delle funzioni `require`, come nell'esempio di figura 3.2 dove il package relativo ad un certo add-on (chiamato `new-addon`) comunica con il package `addon kit`.

Un package deve sempre contenere un file che lo descrive chiamato: `package.json` in cui inserire l'autore, la descrizione, le dipendenze, ecc. JSON è l'acronimo di JavaScript Object Notation, un formato facilmente comprensibile per gli umani e creabile/analizzabile per le macchine. Inizia e finisce con le parentesi graffe, ed è formato da coppie "nome" : "valore" separate dalla virgola. Per esempio il contenuto del file `package.json` di un add-on può essere quello in codice 3.1.

```
{
  "name": "new-addon",
  "fullName": "New Add-on",
  "description": "This is an example of addon description.",
  "author": "Name Surname",
  "license": "MPL",
  "version": "0.1"
}
```

Codice 3.1: Esempio di codice contenuto del file `package.json`

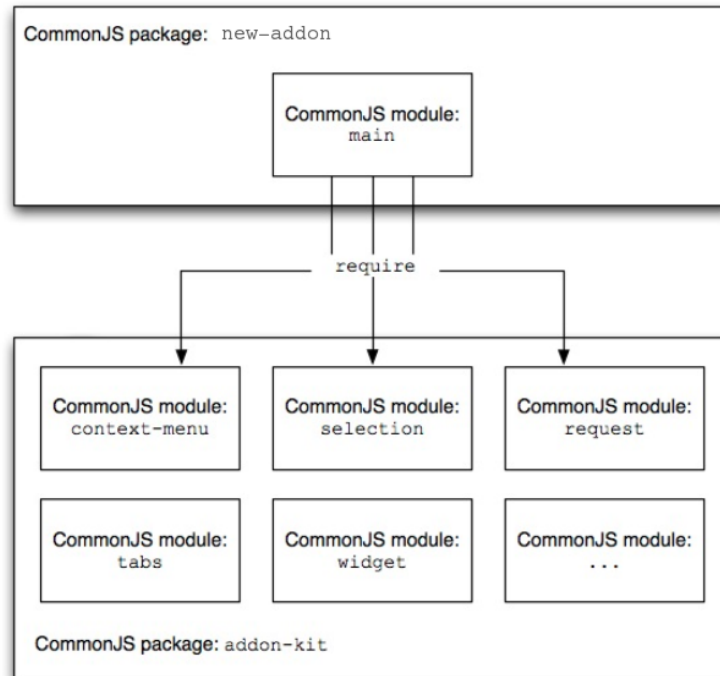


Figura 3.2: Dipendenze tra il package del add-on `new-addon` e il package `addon kit` del SDK (disponibile a [5])

Come si può vedere, le informazioni presentate sono facilmente interpretabili ed autoesplicative.

### 3.1.2 Content script

I content script sono file JavaScript che permettono di interagire con le pagine Web, manipolandole e modificandole (meccanismo simile a quello utilizzato da GreaseMonkey con gli user script). Questi script vengono iniettati nelle pagine Web usando API definite in alcuni moduli della SDK come `page-mod`, `panel` e `widget`. In realtà questi script possono essere inseriti in due modi differenti:

- tramite stringhe letterali da inserire dove occorre nel file `main.js`;

- mantenuti in file separati e referenziati attraverso il loro nome e posizione nella cartella `data`.

Se il content script è memorizzato in un file separato, si può anche utilizzare qualsiasi altra libreria JavaScript, come JQuery, che andrà inserita nella cartella `data`.

I file JavaScript del content script possono interagire solo con il modulo `main` col quale possono scambiarsi informazioni attraverso specifici eventi.

Un add-on deve sempre avere il file `main.js` nella cartella `lib`, mentre non è detto che abbia bisogno di content script (dipendente alla finalità dell'estensione).

### 3.1.3 Add-on code vs Content script

All'inizio di questa sezione, abbiamo accennato al fatto che i due tipi di script si differenziano anche per le API a cui possono accedere, oltre che per le finalità e le cartelle in cui vengono inseriti.

La tabella 3.1 riassume quali API possono essere disponibili nei differenti script [68].

## 3.2 Struttura

Come introdotto nella sezione precedente, tutti i moduli dell'add-on forniti nella cartella `lib` creano un package, che prende il nome dell'estensione stessa. Questo package comunica con i package dell'SDK come si può visualizzare in figura 3.3.

L'estensione ha al suo interno una struttura complessa che verrà spiegata in seguito.

### 3.2.1 Scheletro

Il progetto Add-on SDK mette a disposizione vari strumenti (come visto nel dettaglio in 2.3.1). Attraverso `cfx` si può inizializzare la struttura del-

API	Addon-code	Content script
Oggetti globali definiti nel cuore del linguaggio JavaScript, come <code>Math</code> , <code>Array</code> e <code>JSON</code> .	✓	✓
La funzione globale <code>require()</code> e gli oggetti globali <code>exports</code> definiti nei moduli CommonJS. Da ciò dipende anche l'utilizzo degli oggetti <code>exports</code> forniti dai moduli della SDK.	✓	✗
La globale <code>console</code> fornita dall'SDK	✓	✓
Le globali definite da HTML 5, come <code>windows</code> , <code>document</code> e <code>localStorage</code> .	✗	✓
La globale <code>self</code> , usata per comunicare tra content script e add-on code.	✗	✓

Tabella 3.1: Differenze nella disponibilità delle API tra content script e add-on code

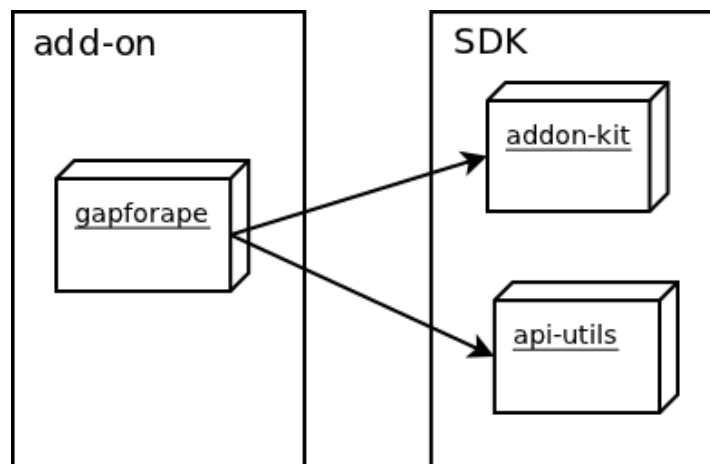


Figura 3.3: Dipendenze tra il package del nostro sistema `gapforape` e i package del SDK

l'estensione; per farlo basta eseguire da linea di comando `cfx init`. Questo comando va eseguito dopo aver attivato `cfx` (sempre da linea di comando) e aver creato la cartella che andrà a contenere tutta l'estensione. Lo scheletro creato da tale comando ha la struttura di file in codice 3.2:

```
README.md
package.json
/data
/lib
  main.js
/tests
  test-main.js
/docs
  main.md
```

Codice 3.2: Scheletro dell'estensione

Ogni cartella ha uno specifico scopo:

- **data**: nella quale vanno inserite varie risorse come icone, file HTML e ogni content script, ovvero script che permettono di interagire con il contenuto delle pagine Web o con l'interfaccia del browser;
- **doc**: permette di includere tutta la documentazione relativa all'add-on sviluppata;
- **lib**: in cui mettere i moduli JavaScript CommonJS, ovvero gli script di tipo add-on code, che permettono di implementare le funzionalità dell'estensione; è in questa cartella che andrà inserito il file `main.js`, che è il cuore di tutta la struttura, ed eventuali altri moduli;
- **test**: in cui inserire il codice per effettuare gli "unit test". Unit test è un un metodo tramite il quale pezzi di codice sono testati individualmente per determinare se si comportano come progettato.

Il file `package.json` permette di inserire informazioni sull'add-on, come autore, descrizione, codice univoco (ID), dipendenze, ecc.

Nel file `README.md` si possono inserire informazioni sul contenuto di ogni cartella interna all'add-on, sull'uso, i requisiti, ecc.

### 3.2.2 Cartella `lib`

In questa cartella sono inseriti tutti i file relativi all'addon code, ovvero i moduli CommoJS che possono utilizzare, tramite la funzione `require()` le funzionalità rese disponibili dai moduli dell'SDK. Nella cartella `lib` sono presenti quattro file corrispondenti ad altrettanti moduli CommonJS: `main.js`, `file-manager.js`, `user-prefs.js`, `content-script.js`. In figura 3.4 si possono vedere le dipendenze tra i vari moduli indicati tramite le frecce. Questi moduli hanno anche dipendenze con i package dell'SDK, come verrà descritto in seguito.

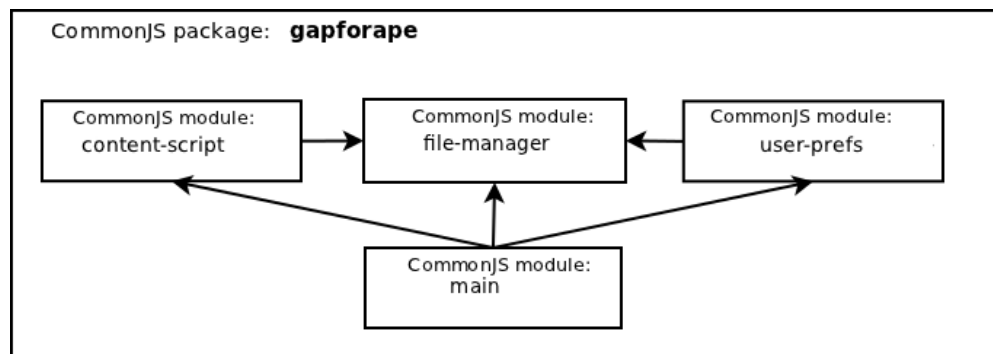


Figura 3.4: Dipendenze tra i moduli del package `gapforape`

#### Modulo `main`

Questo è il file cuore dell'estensione; gestisce e controlla tutti i componenti dell'add-on. In questo file vengono gestite le interazioni tra la parte di *addon code* e quella di *script content*, permettendo lo scambio di dati tra le

due. Esso importa molti oggetti forniti dai moduli dell'SDK. Il diagramma di figura 3.5 mostra le dipendenze del file `main.js` con gli altri moduli, sia dell'SDK che di `gapforape`.

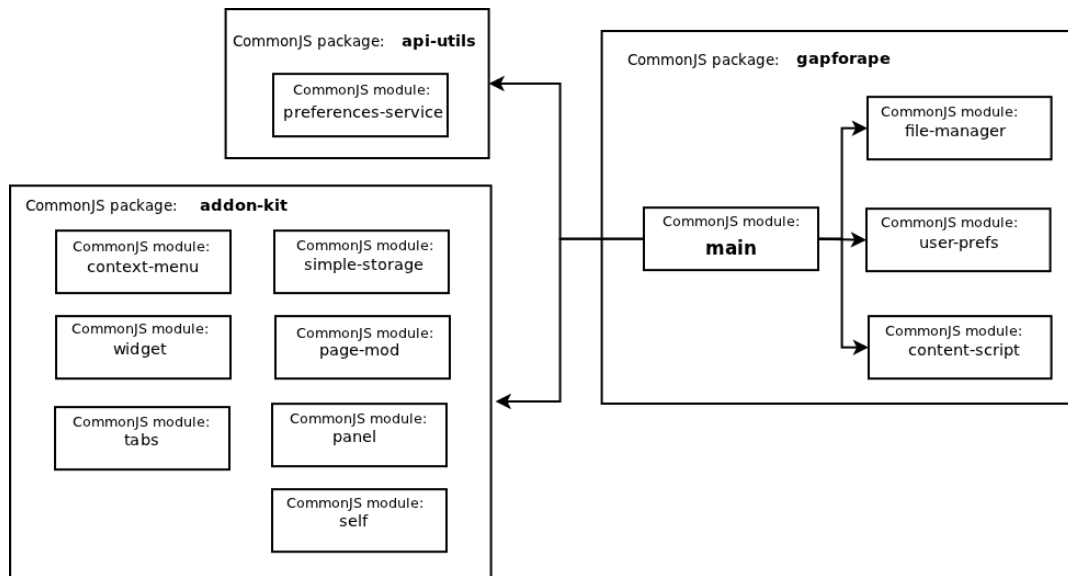


Figura 3.5: Dipendenze tra il modulo `main` e gli altri moduli

In questo modulo sono state inserite tutte le funzionalità principali dell'estensione; di seguito verranno elencate in modo molto sintetico con lo scopo di dare un'idea della centralità di questo modulo (nel seguito del capitolo verranno trattate in modo approfondito una ad una):

- creare il `widget` che permette all'utente di gestire e configurare l'addon;
- associare al `widget` un `panel`, la cui interfaccia può essere modificata attraverso codice HTML e JavaScript. Tramite questo `panel` si fornisce all'utente il menu principale dell'estensione;
- creare un `page-mod` che permette di creare un'interfaccia complessa tramite la quale gli utenti possono configurare il loro profilo;



- ottenere, dal `page-mod` associato alla pagina HTML del profilo, un oggetto JSON che verrà trasformato in un file JavaScript fruttando le funzionalità offerte dal modulo `user-prefs`;
- modificare il contenuto di tutte le pagine caricare, iniettando il codice JavaScript;
- attivare la possibilità di modificare la pagina “on-the-fly”;
- riconoscere di aver effettuato il click destro su un elemento di una qualsiasi pagina (con modalità “enable changes to the current page” attiva) e visualizzare quindi una pagina HTML tramite la quale permettere all’utente di effettuare cambiamenti in modo facilitato;
- salvare i cambiamenti effettuati dall’utente “on-the-fly” in un oggetto JSON e trasformarli in codice JavaScript attraverso le funzionalità offerte dal modulo `content-script`.

### Modulo `file-manager`

Il modulo `file-manager` permette di creare all’interno del profilo attivo del browser, una cartella chiamata `GAPforAPE`, e due sottodirectory:

- `profile`: in cui verranno salvati:
  - il file JavaScript risultante dalla trasformazione delle preferenze dell’utente in codice, chiamato `preferences.js`;
  - eventuali altri file JavaScript che l’utente:
    - \* ha creato attraverso la modifica “on-the-fly” su di una pagina scegliendo poi di rendere la modifica applicabile a tutte le pagine visualizzate dal browser;
    - \* ha creato utilizzando JavaScript e/o JQuery;
    - \* ha scaricato da Internet.

che vanno ad ampliare il profilo dell'utente, essendo modifiche che la stessa persona ha scelto di effettuare su qualsiasi pagina visualizzata dal browser. Questi file vengono inseriti in una sottodirectory chiamata **general**;

- **specialSites**: in cui verranno memorizzati i file JavaScript relativi ad uno specifico sito e collegati ad esso attraverso il nome del file stesso, che l'utente:
  - ha creato attraverso la modifica “on-the-fly” su di una specifica pagina visualizzata;
  - ha creato utilizzando JavaScript e/o JQuery;
  - ha scaricato da Internet.

questi cambiamenti permettono di effettuare miglioramenti specifici ad un certo sito Web o ad una particolare pagina Web.

Oltre alla gestione di queste directory, il modulo **file-manager** ha anche funzioni che permettono di creare ed effettuare semplici operazioni sui file, come restituirne il contenuto, controllare se esiste e altrimenti crearlo, restituire la lista di file all'interno di una specifica directory.

Il modulo **file** del package **api-utils** mette a disposizione alcune funzionalità base come: **read** per leggere il contenuto di un file; **join** per effettuare l'unione tra due path in modo compatibile con il sistema operativo utilizzato; **list** per ottenere la lista di file interna ad una directory. Per altre operazioni base, come la creazione sia di cartelle che file occorre interagire con il modulo **chrome** che è il modulo di basso livello più potente.

Il modulo **file-manager** rende disponibile all'esterno solo due funzionalità (in modo da non permettere di effettuare operazioni “maligne”):

- **getListFile(subDir)**: restituisce un array contenente i nomi dei file presenti nella sottocartella **subDir**;

- `getContentFile(nameFile, nameSubDir)`: restituisce una stringa con l'intero contenuto del file `nameFile` che si trova nella sottodirectory `nameSubDir`.

Le dipendenze di questo modulo sono mostrate nel diagramma di figura 3.6.

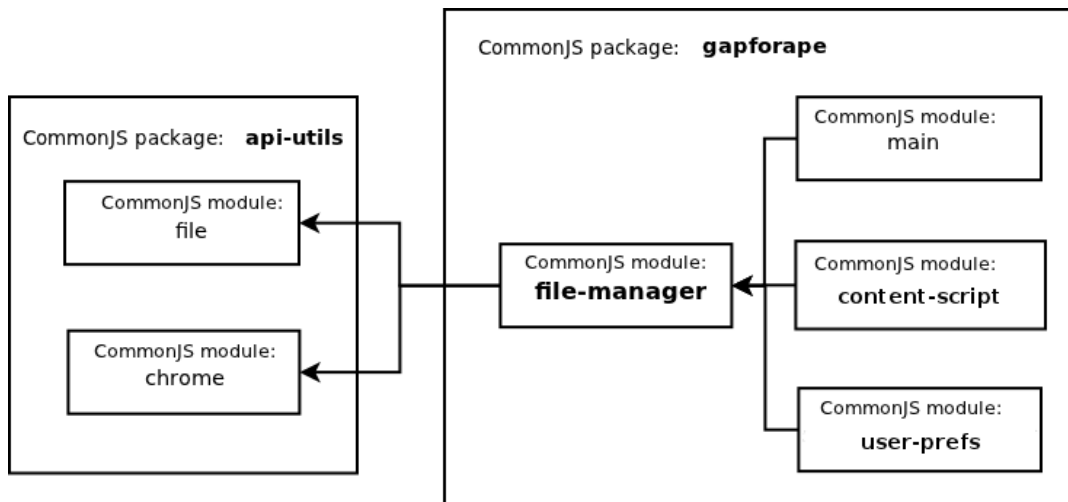


Figura 3.6: Dipendenze tra il modulo `file-manager` e gli altri moduli

### Modulo `user-prefs`

Il modulo CommonJS `user-prefs` permette di gestire la memorizzazione delle preferenze dell'utente. Esso rende disponibile al modulo `main` la funzione `setPrefs(objJSON)` che “trasforma” l'oggetto JSON `objJSON`, contenente le preferenze dell'utente, in codice JavaScript da iniettare in tutte le pagine visualizzate dal browser. Il modulo è a conoscenza della struttura dell'oggetto JSON che il `main` gli passerà come parametro, di consegna, per ogni coppia `nome:valore` dell'oggetto corrisponde un pezzo di codice JavaScript ben definito, che verrà personalizzato in base al `valore`. In realtà, il file creato non sarà in puro JavaScript, ma si farà uso della libreria JQuery, ritenuta più semplice, veloce e diretta.

Un esempio di questa “trasformazione” può essere il seguente. L’utente può scegliere se volere che tutti i link visualizzati abbiano l’aspetto di link, ovvero siano sottolineati. Considerando la coppia `nome:valore` JSON, dove il nome è ‘`underlineLink`’ ed il valore può essere ‘`true`’ oppure ‘`false`’, a seconda che si vuole, appunto, che tutti i link siano sottolineati oppure rimangano come scelto dal creatore del sito Web. Nel momento in cui, nel modulo `user-prefs`, si effettuerà il parser dell’oggetto `obJSON` e si incontrerà la coppia con nome ‘`underlineLink`’ verrà invocata la funzione in codice 3.3:

```
setUndLink(value) {
    var rule = "";
    if(value) {
        rule += "$('a').css('text-decoration', 'underline');\n";
    }
    return true;
}
```

Codice 3.3: Funzione JavaScript per modificare la regola CSS `underline`

Questa funzione permette di creare la regola JQuery che verrà salvata nel file `userPrefs.js` e iniettata in ogni pagina visualizzata. Questo file, verrà creato la prima volta che l’utente configura l’estensione (e ogni qual volta modificherà le sue preferenze).

Nel diagramma in figura 3.7 viene mostrato con che altri moduli interagisce il modulo `user-prefs`.

### Modulo `content-script`

Il modulo `content-script` permette di creare i content script da iniettare nelle pagine visualizzate, oltre a quelli definiti configurando le preferenze

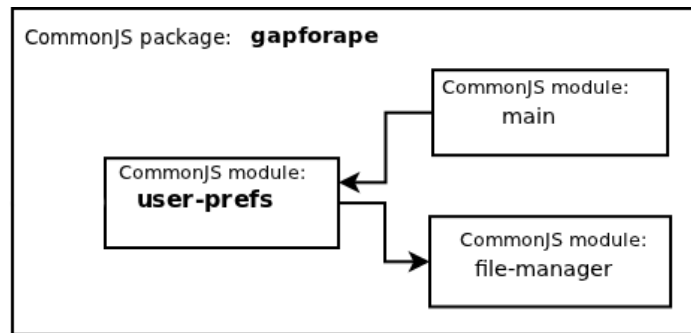


Figura 3.7: Dipendenze tra il modulo `user-prefs` e gli altri moduli

dell'utente. È tramite questo modulo che le modifiche fatte on-the-fly, attraverso il tasto destro sugli elementi selezionati, possono essere salvate come codice JavaScript, più precisamente come codice JQuery (per le motivazioni viste precedentemente). Il meccanismo è simile a quello attuato dal modulo `user-prefs` con la differenza che questa volta il modulo non sa esattamente come sarà l'oggetto JSON che gli viene passato dal `main`. Di questo oggetto sa solo che è composto da coppie `nome:valore` in cui il `nome` è una proprietà `css` o una funzione JQuery e il `valore` è il corrispondente valore.

Nel diagramma in figura 3.8 vengono mostrate le dipendenze.

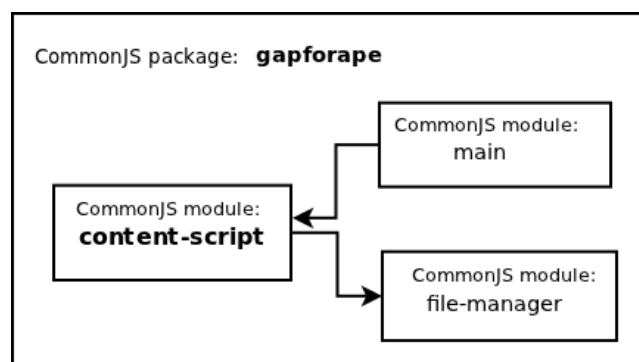


Figura 3.8: Dipendenze tra il modulo `content-script` e gli altri moduli

### 3.2.3 Cartella data

Nella cartella `data` è presente tutta la parte di content script ed è articolata in varie cartelle e file come si può vedere in codice 3.4.

```
\data
  \widget
    logo.png
    logo-ON.png
    widget.js
  \widgetPanel
    panel.html
    panel.css
    panel.js
  \preferences
    prefs.html
    prefs.css
    prefs.js
  \personalizePage
    \personalize.html
    \personalize.css
    \personalize.html
    selector.js
    selectorItem.js
  jquery-min.js
```

Codice 3.4: Struttura di file e sottodirectory della cartella `data`

Il file `jquery-min.js` permette di utilizzare le librerie JQuery all'interno dei file JavaScript. Le altre cartelle verranno spiegate in modo approfondito in seguito.

## 3.3 Widget

Il *widget* è ciò che rende disponibile l'add-on e permette di utilizzare le funzionalità che mette a disposizione. Da Firefox 4 infatti, gli add-on sono presentati solo attraverso l'*add-on bar*, che si trova in basso nell'interfaccia del browser. Il widget è costituito da un'icona a cui si possono aggiungere funzionalità, fino a farlo diventare una complessa pagina Web. Nel nostro sistema, il widget è collegato ad un `panel` associato ad una pagina HTML, un file JavaScript e un foglio di stile CSS (sono il contenuto della cartella `widgetPanel`). Nella figura 3.11 si può vedere il widget (la piccola icona con la "scimmietta") e cosa l'utente visualizza effettuando il click su di essa. Quando il puntatore si trova sopra la voce di menu, il colore di sfondo cambia, così l'utente può facilmente verificare dove si trova il puntatore del mouse. Il colore scelto è nel pieno rispetto delle linee guida WCAG 2.0, garantendo la richiesta differenza di contrasto e di luminosità tra i colori di sfondo e testo. Un altro miglioramento, per permettere anche a chi non riesce a compiere movimenti precisi di selezionare la voce di menu appropriata, è stato quello di rendere l'elemento completamente cliccabile in tutta la zona di colore azzurro.

Il primo elemento del menu, **Preferences**, permette di far aprire una nuova scheda nella finestra del browser in cui viene visualizzata una pagina contenente tutte le preferenze che l'utente può configurare per crearsi il proprio profilo.

La seconda voce, **Enable changes to the current page**, serve per abilitare le modifiche on-the-fly sulla pagina corrente. Quando le modifiche sono attive l'icona cambia lo sfondo e la voce del menu diventa **Disable changes to the current page**, come mostrato nelle due figure 3.10.

## 3.4 Profilo

Il profilo di un'utente è dato:

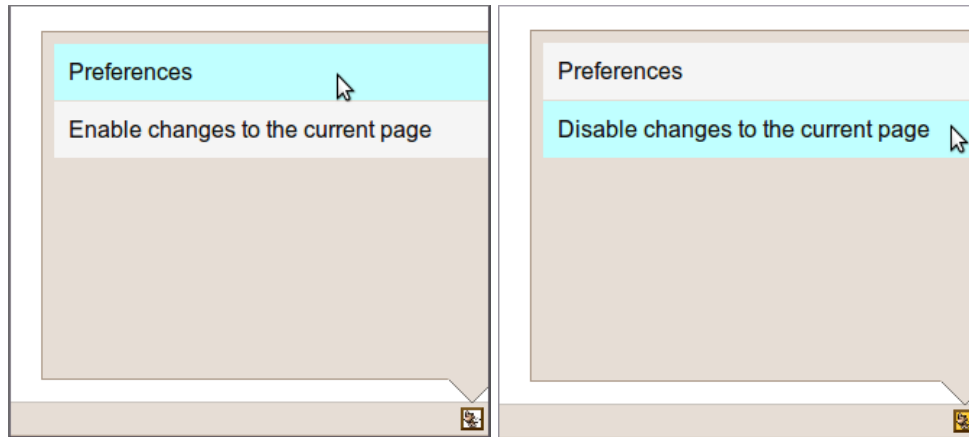


Figura 3.9: Widget con la funzione di modifica pagina disattivata)

Figura 3.10: Widget con la funzione di modifica pagina attivata)

Figura 3.11: Icona del widget e visualizzazione del menu dell'add-on

- dalle preferenze che l'utente stesso configura all'installazione dell'estensione e che potrà modificare ogni qual volta necessario, attraverso la voce di menu **Preferences** raggiungibile effettuando il click sul widget;
- dal contenuto dei file che l'utente ha creato on-the-fly oppure manualmente, e/o scaricato dal Web, presenti nella sottodirectory **general** all'interno della cartella **GAPforAPE** (creata nel profilo attivo del browser).

Appena installata l'estensione, appare una nuova scheda in cui viene presentata la pagina Web visualizzata in figura 3.12 (i file relativi ad essa sono all'interno della cartella **preferences**). Si è scelto di creare l'interfaccia in modo da riprodurre la grafica dell'*Add-on Manager*, così da non spiazzare l'utente e sfruttare la confidenza acquisita dal suo utilizzo. Inoltre, si vuole anche sottolineare come tutte le interfacce fornite dal nostro sistema siano conformi alle linee guida sull'accessibilità WCAG 2.0 e siano state create con tutti gli accorgimenti possibili in materia di usabilità ed accessibilità (come ordine di tabulazione, facilità nelle selezioni delle voci di menu, comandi rapidi).



Il menu ha quattro voci principali:

- color;
- text;
- visual;
- audio.

### 3.4.1 Colori

La sezione **Color** permette all'utente di configurare tutte le preferenze che riguardano i colori, dal momento che, come descritto nel primo capitolo, sono una componente davvero importante nel contesto dell'accessibilità Web. In particolare, l'utente può scegliere:

- il colore di sfondo della pagina;
- il colore del testo;
- il colore dei link visitati e non visitati.

Per i link si può anche scegliere se volere che tutti i link incontrati durante la navigazione abbiano il tipico aspetto di link, ovvero siano visualizzati sempre con la sottolineatura.

Nella figura 3.12 si può vedere l'interfaccia relativa alle preferenze per adattare i colori delle pagine Web.

Quando l'utente sceglie una delle opzioni possibili per il colore di testo e sfondo, la modifica viene immediatamente presentata all'utente, come una specie di anteprima di come sarà l'aspetto delle pagine Web adattate; inoltre ciò permette di disporre immediatamente del miglior contrasto (perché scelto dalla persona stessa) per leggere le informazioni riguardanti le preferenze. In figura 3.13 si può vedere come si presenta la pagina delle preferenze se l'utente sceglie di avere lo sfondo nero e il testo di colore giallo.

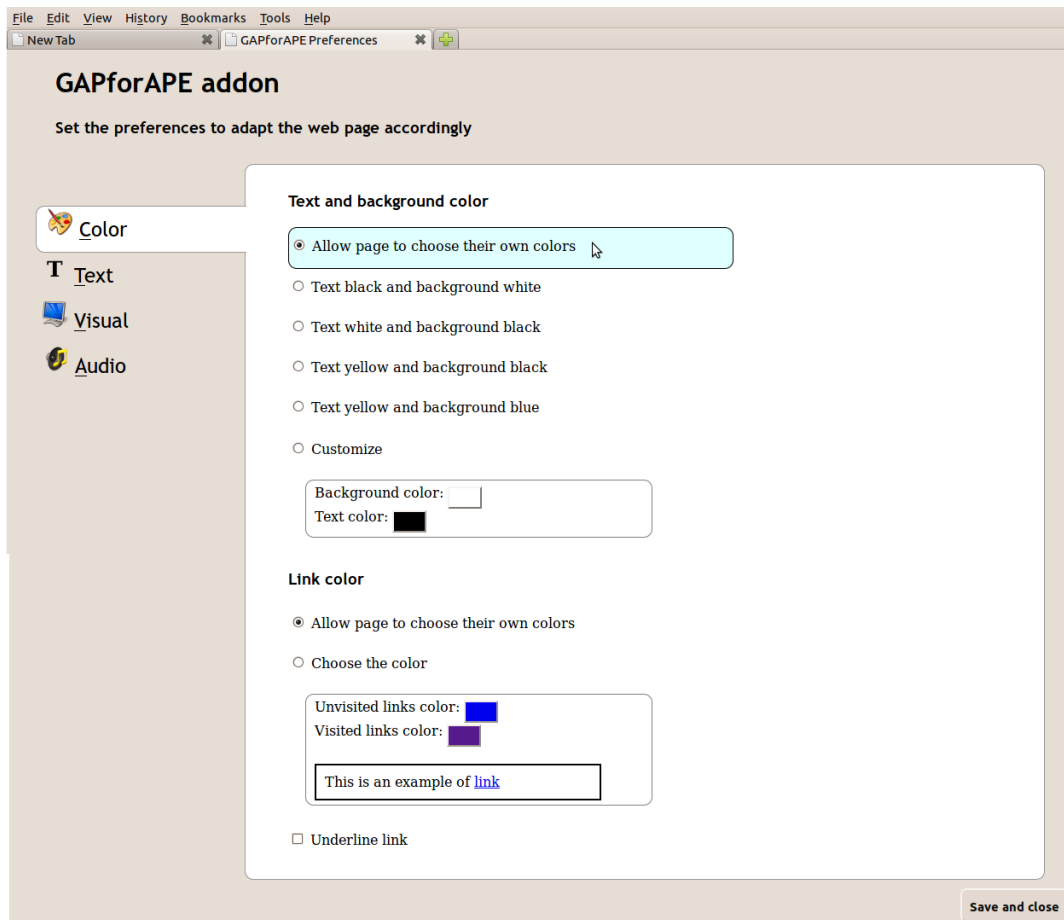


Figura 3.12: Interfaccia preferenze sui colori

Oltre i profili di testo e sfondo già forniti, l'utente può scegliere di personalizzare completamente i colori attraverso l'opzione **Customize** e poi effettuare la scelta del colore tramite il pannello che si apre cliccando sul bottone relativo (come in figura 3.14). Anche in questo caso una volta scelto il colore, la pagina si modifica di conseguenza.

Lo stesso "color picker" viene visualizzato anche nell'opzione **Customize** relativa al colore dei link, visitati e non.

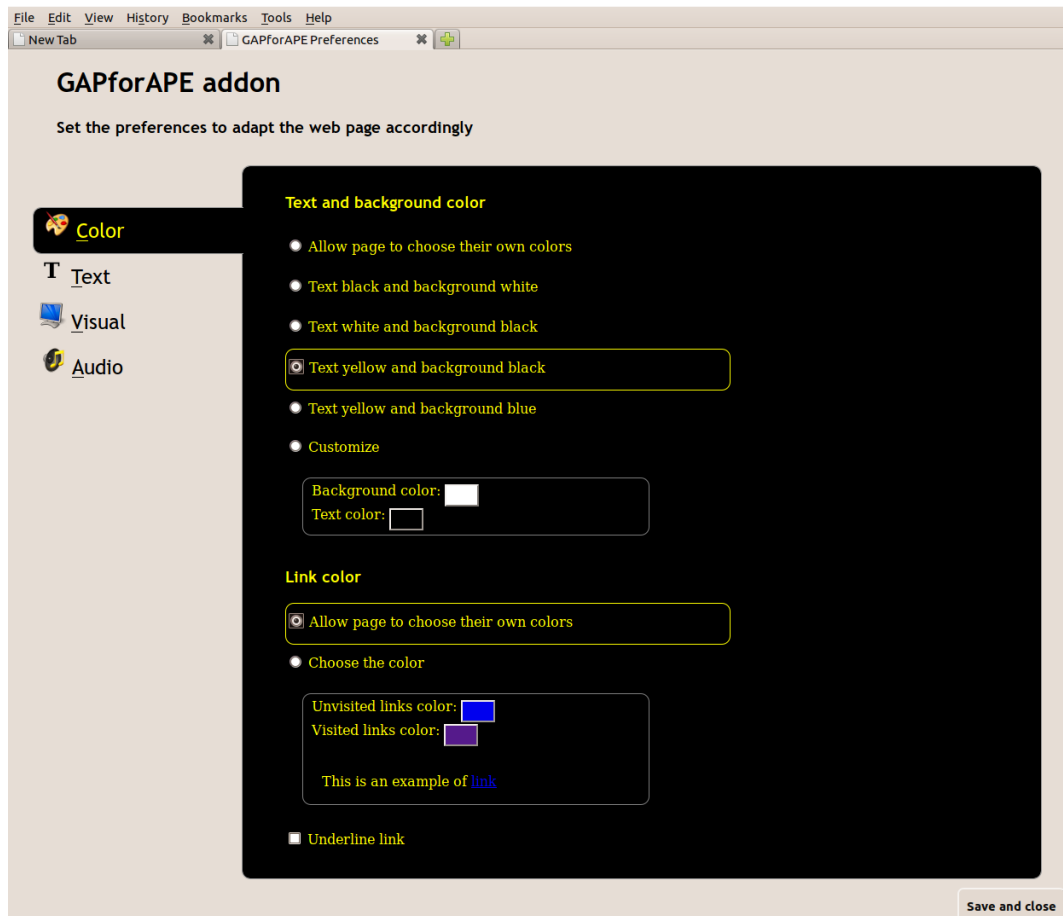


Figura 3.13: Interfaccia preferenze sui colori con attiva l'opzione Text yellow and background black



Figura 3.14: Scelta dei colori di testo e sfondo tramite l'opzione Customize

Per visualizzare se tra i colori scelti per i link e quelli del testo e dello sfondo ci sia abbastanza contrasto è presente un riquadro di “prova”. Un esempio in cui questa anteprima è molto importante è il seguente. Come introdotto nel primo capitolo, molti ipovedenti hanno beneficio utilizzando sfondo blu e testo giallo. Di default il colore dei link non visitati è blu, di conseguenza diventa invisibile su sfondo dello stesso colore. Grazie a questo piccolo accorgimento, la persona può rendersene subito conto e modificare il colore del link di conseguenza.

### 3.4.2 Testo

Ovviamente anche il font del testo può creare problemi nella fruibilità delle informazioni presentate in una pagina Web. Per questo si è fornita la possibilità di modificarlo attraverso la voce **Text**. Le opzioni presenti sono mostrate in figura 3.15. L’utente può scegliere tra i più diffusi tipi di font (come mostrato in figura 3.16). Un altro tipico problema è la dimensione del testo. Oltre ad offrire la possibilità di scegliere una grandezza fissa per tutti i tipi di testo, è presente un’opzione molto interessante. Infatti, l’utente può settare la grandezza minima che il testo deve avere (il testo con grandezza maggiore di quella scelta rimane invariato). Per verificare il tipo di font preferito e la grandezza del testo è presente un riquadro con l’esempio del testo.

### 3.4.3 Visuale

Nella voce di menu **Visual** sono presenti opzioni che permettono di migliorare aspetti strettamente visuali. Le opzioni fornite sono suddivise in quattro gruppi, come si può vedere in figura 3.17.

#### **Immagini**

Nella sezione riguardante le immagini (**Image**) si può scegliere di:

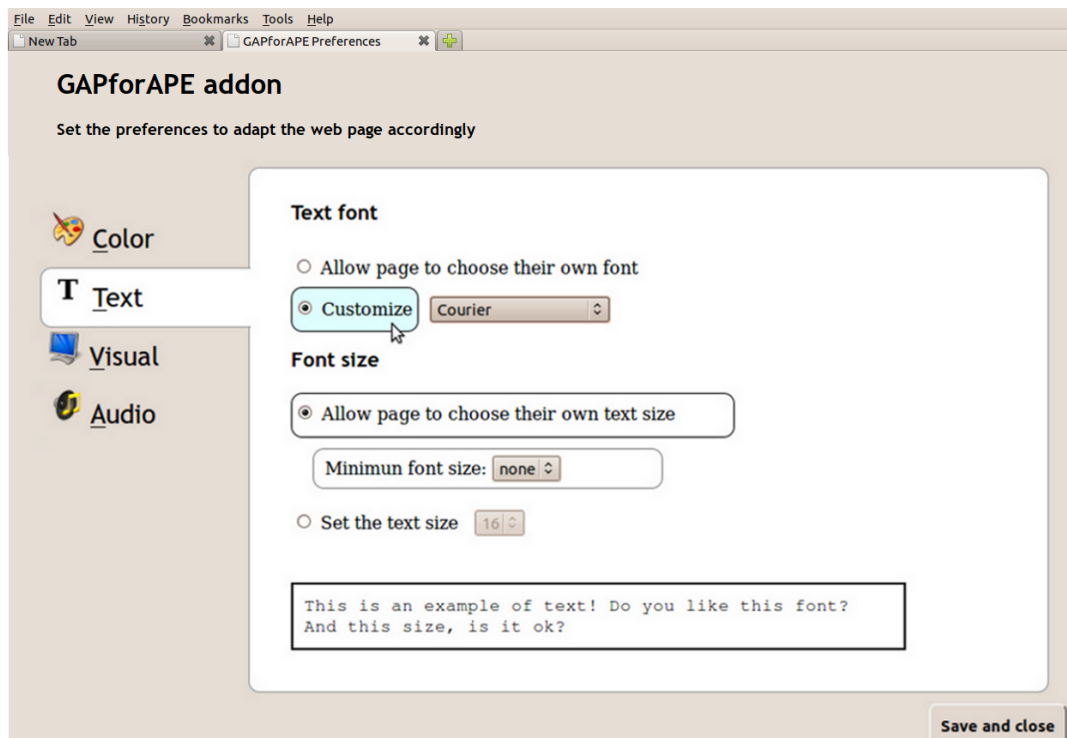


Figura 3.15: Opzioni presententi nella voce Text

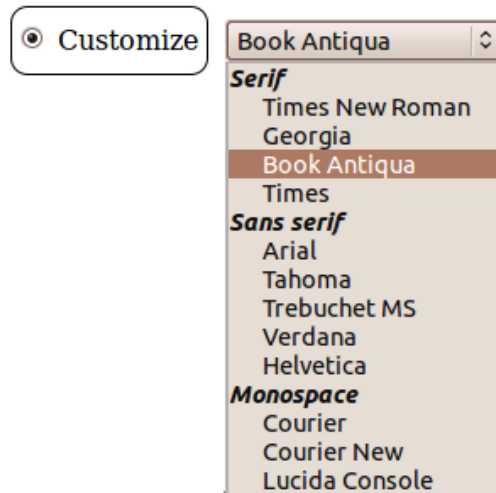


Figura 3.16: Tipi di font tra i quali l'utente può scegliere per adattare il testo delle pagine Web

- *non visualizzare nessuna immagine*: qualsiasi immagine presente in una pagina non verrà mostrata;
- *non mostrare le immagini animate*: questa è un'opzione utile soprattutto per chi soffre di attacchi epilettici ma occorre comunque considerare che i movimenti attirano lo sguardo quindi disturbano e distraggono l'utente. Per ora è presente un semplice controllo sul formato dell'immagine ma ovviamente non è sufficiente e sarà uno dei miglioramenti da effettuare negli sviluppi futuri;
- *visualizzare insieme all'immagine le informazioni presenti su di essa*: ovvero il contenuto dell'attributo `alt` altrimenti, se non presente, il nome dell'immagine senza l'estensione. Un'importante aggiunta per migliorare l'accessibilità è stata l'introduzione dell'attributo `longdesc` per l'elemento html `img` il cui valore è un file (con formato html, txt, ecc.) in cui sono presenti maggiori informazioni per l'immagine. Attualmente non è supportato da nessun browser per questo si

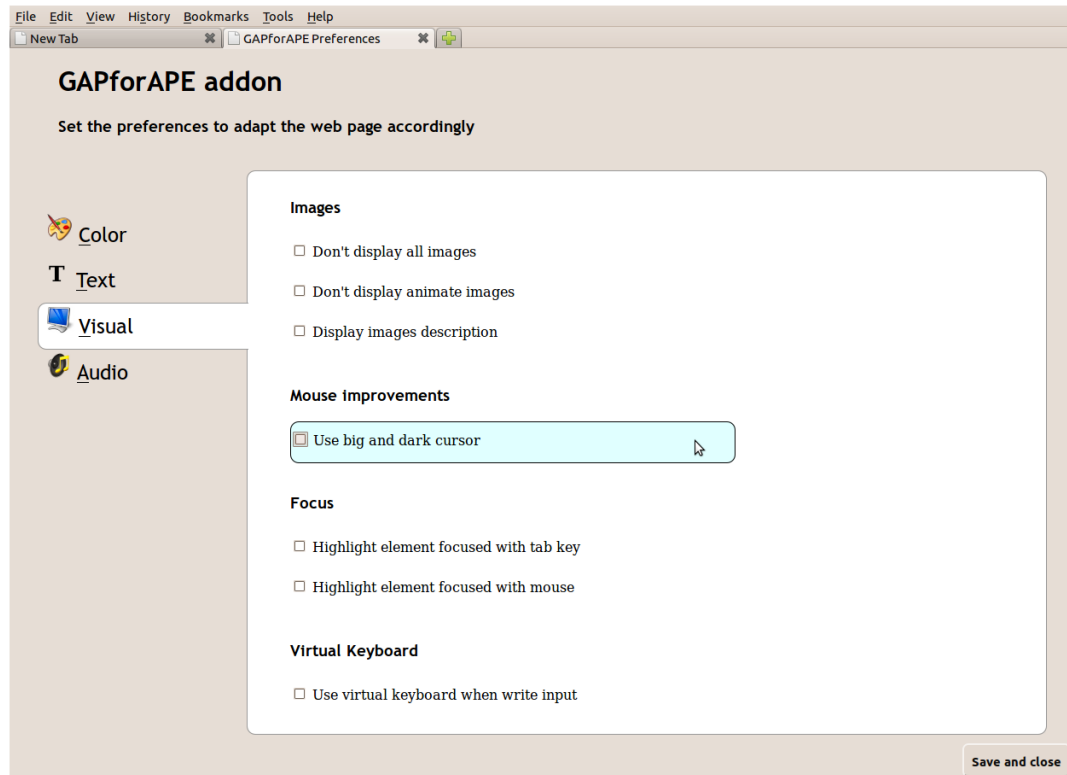


Figura 3.17: Opzioni presentati nella voce di menu Visual

è scelto di visualizzare anche questa informazione (scegliendo questa opzione). Per esempio, dato il pezzo di codice HTML (in codice 3.5), disponibile nella descrizione dell'attributo `longdesc`, fornito dal sito <http://www.w3schools.com/>.

```

```

Codice 3.5: Esempio di un l'elemento `img` con attributo `longdesc`

il risultato ottenuto sarà quello di figura 3.18 nel caso di visualizzazione con un qualsiasi browser mentre quello di figura 3.19 è quello ottenuto con l'opzione Display images description attivata.



Figura 3.18: Immagine visualizzata normalmente dal browser

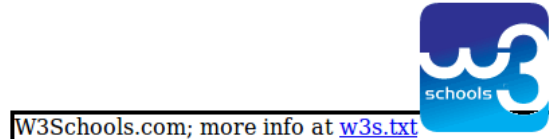


Figura 3.19: Immagine visualizzata con l'opzione Display images description attiva

Figura 3.20: Immagine con l'attributo longdesc

### Miglioramenti del mouse

Nella sua forma tipica, il puntatore del mouse ha una forma piccola e di colore bianco. Si è reso possibile permettere all'utente di scegliere un mouse più grande e di colore nero. Attivando questa opzione il risultato è quello visibile in figura 3.21. Sono presenti due immagini<sup>1</sup>, la freccia da usare di default e la mano per indicare un link.



Figura 3.21: Miglioramento nella visualizzazione del puntatore del mouse

<sup>1</sup>Per le immagini dei due cursori si ringrazia Andrea Mangiatordi del Farfalla Project.



## Focus

Le pagine Web sono sempre più colorate e grafiche e così diventa davvero difficile capire quale sia l'elemento con il focus, in modo particolare se si naviga attraverso il tasto tabulatore. Si è deciso di aiutare l'utente fornendo due opzioni con la possibilità di attivarle entrambe:

- evidenziare l'elemento al quale si dà il focus con il tasto tabulatore con un bordo di colore rosso;
- evidenziare l'elemento sul quale si trova il puntatore con un bordo di colore giallo.

Nell'immagine si può vedere il caso in cui entrambe le opzioni sono attivate.

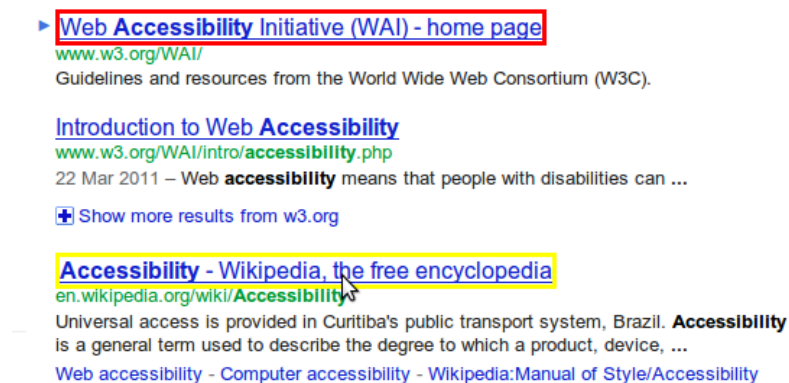


Figura 3.22: Opzioni Highlight element focused with tab key e with mouse attivate

## Audio

Nel nostro sistema è presente anche la possibilità di disabilitare l'audio (figura 3.23):

- nel browser Web Firefox che spesso per selezionare errori o notifiche utilizza il suono;
- nelle pagine Web che fanno uso di musica, suonerie o audio in generale.

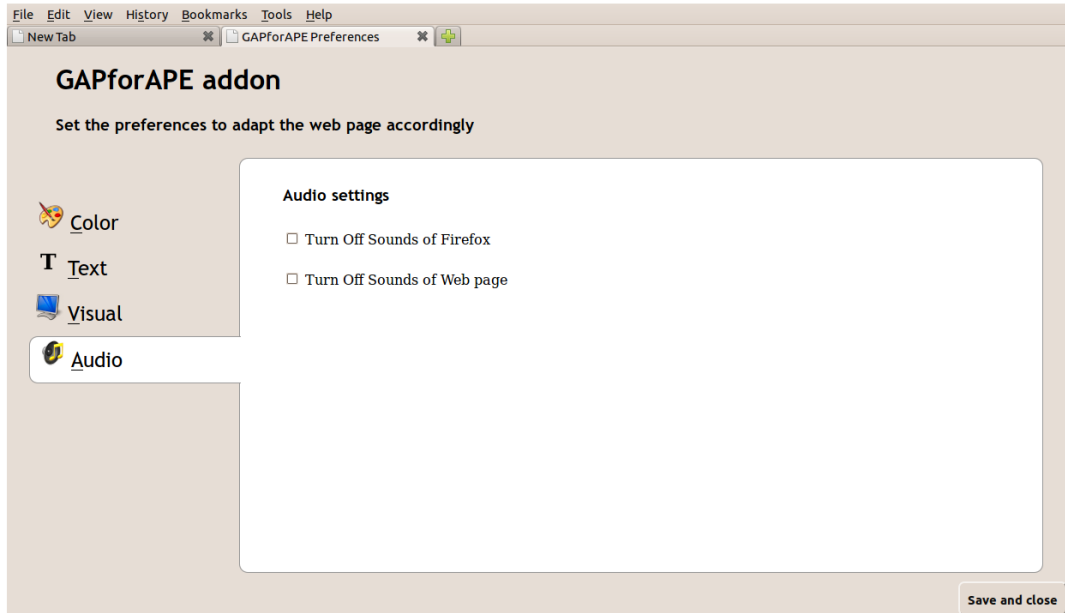


Figura 3.23: Opzioni voce Audio

Sicuramente con l'avanzamento della standardizzazione di HTML 5, si potranno effettuare molti più miglioramenti relativi alle componenti multimediali delle pagine Web.

### 3.5 Modifiche on-the-fly sulla pagina corrente

Rendendosi conto che con le preferenze fornite non è possibile coprire tutte le esigenze di qualsiasi utente si è pensato di rendere possibile inserire i propri script o scaricarli ed inserirli nelle apposite cartelle. Ciò però non è sempre facile, non molti utenti sono intenzionati a crearsi il proprio script o perdere tempo nel cercare lo script voluto che magari nessuno offre. Si è

così pensato di fornire un metodo molto veloce e facile per effettuare quelle piccole modifiche per migliorare ulteriormente la fruibilità di un sito Web.

Per attivare questa opzione, occorre selezionare la voce **Enable changes to the current page** dal menu del widget. A quel punto ogni elemento che si troverà sotto il puntatore verrà evidenziato con un bordo giallo. Se l'utente effettua il click con tasto destro su quell'elemento, il bordo diventa rosso e si apre una nuova scheda con tutte le informazioni e gli attributi sull'elemento che l'utente potrà modificare e ne potrà anche aggiungere di nuovi.



Figura 3.24: Selezione dell'immagine e visualizzazione nel menu di contesto della voce GAPforAPE selected image info

Nella figura 3.24 si può vedere la fase di selezione dell'elemento, nel caso specifico di un'immagine. Quando l'utente fa il click con il tasto destro, il browser visualizza il menu contestuale, dove è stato aggiunto l'elemento GAPforAPE selected image info. Selezionando la voce appare una pagina Web (file che si trovano nella cartella `personalizePage`) in cui sono presenti tutte le informazioni dell'immagine che l'utente può modificare e può aggiungere valori negli attributi non utilizzati. Tutte queste personalizzazioni verranno

poi salvate e la pagina da quel momento in poi sarà sempre adattata in accordo con esse.

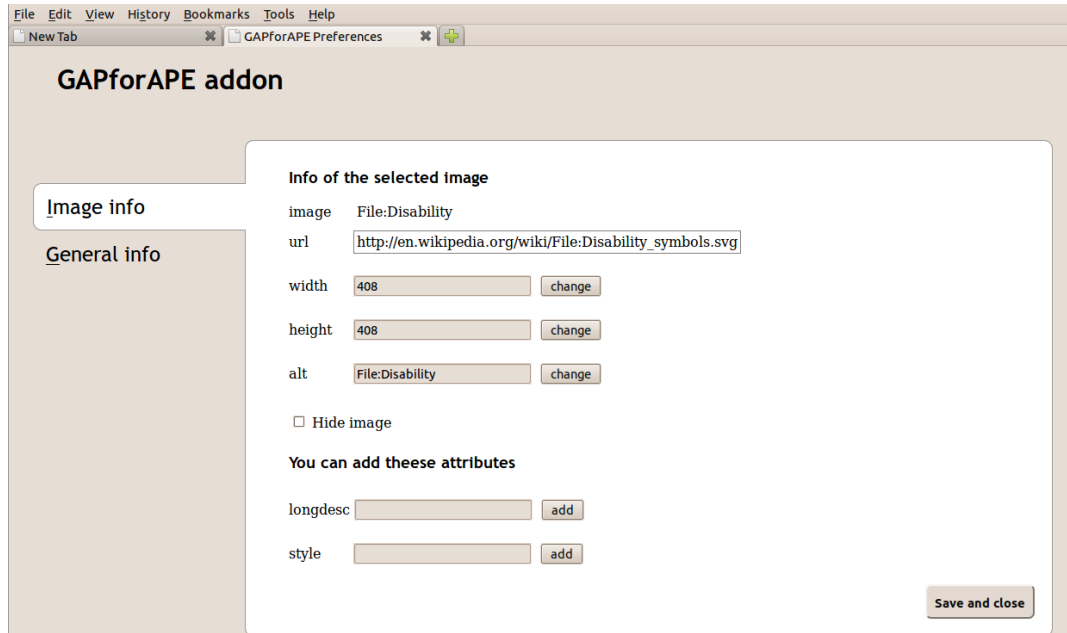


Figura 3.25: Pagina Web in cui è possibile modificare e aggiungere gli attributi dell'immagine selezionata

### 3.6 Caso d'uso

In questa sezione verrà descritto come un utente ipovedente con determinate caratteristiche potrà creare il proprio profilo che gli permetterà di visualizzare le pagine Web secondo le sue necessità e, quindi, nel modo per lui migliore.

Supponiamo che l'utente abbia le seguenti necessità:

- preferisce un forte contrasto tra i colori di sfondo e testo, in particolare necessità di sfondo nero e testo giallo;

- lo sfondo nero però non gli permette di distinguere bene il colore blu, dei link non visitati, e viola, dei link visitati;
- preferisce rendersi subito conto della presenza di un link, riconoscendolo dal suo tipico aspetto (sottolineatura);
- ha difficoltà a leggere il testo troppo piccolo;
- non riesca a distinguere bene i caratteri molto ravvicinati quindi predilige contenuti scritti con un font *monospace*;
- le immagini animate lo distraggono quindi preferisce non visualizzarle;
- spesso usa il tasto tabulatore per navigare la pagina ma trova difficoltà a vedere qual è l'oggetto selezionato.

Attraverso l'interfaccia descritta in 3.4 potrà “informare” GAPforAPE delle sue necessità in modo tale che il sistema, in modo automatico, adatti i contenuti delle pagine Web in accordo con il suo profilo.

Nella voce di menu **Color** potrà scegliere di volere: lo sfondo di colore nero e il testo di colore giallo; i link non visitati verdi mentre quelli visitati rossi e tutti i link sottolineati (come mostrato in figura 3.26). Attraverso il menu **Text** potrà cambiare il font del testo in **Courier New** e decidere la grandezza minima del testo (come visualizzato in figura 3.27). Le configurazioni riguardanti le immagini animate e l'attivazione della modalità che permette di evidenziare l'elemento selezionato tramite il tasto tabulatore potrà invece effettuarle dal menu **Visual** (figura 3.28).

L'oggetto JSON risultante dal profilo configurato, è quello in codice 3.6 (nel quale vengono mostrate solo le opzioni interessanti per questo caso d'uso).

Le pagine Web che verranno visualizzate nel browser avranno tutte l'aspetto che si può vedere in figura 3.29. L'immagine mostra in particolare la pagina principale del social network Google+[69].

La stessa interfaccia visualizzata con il profilo di default si presenta come in figura 3.30.

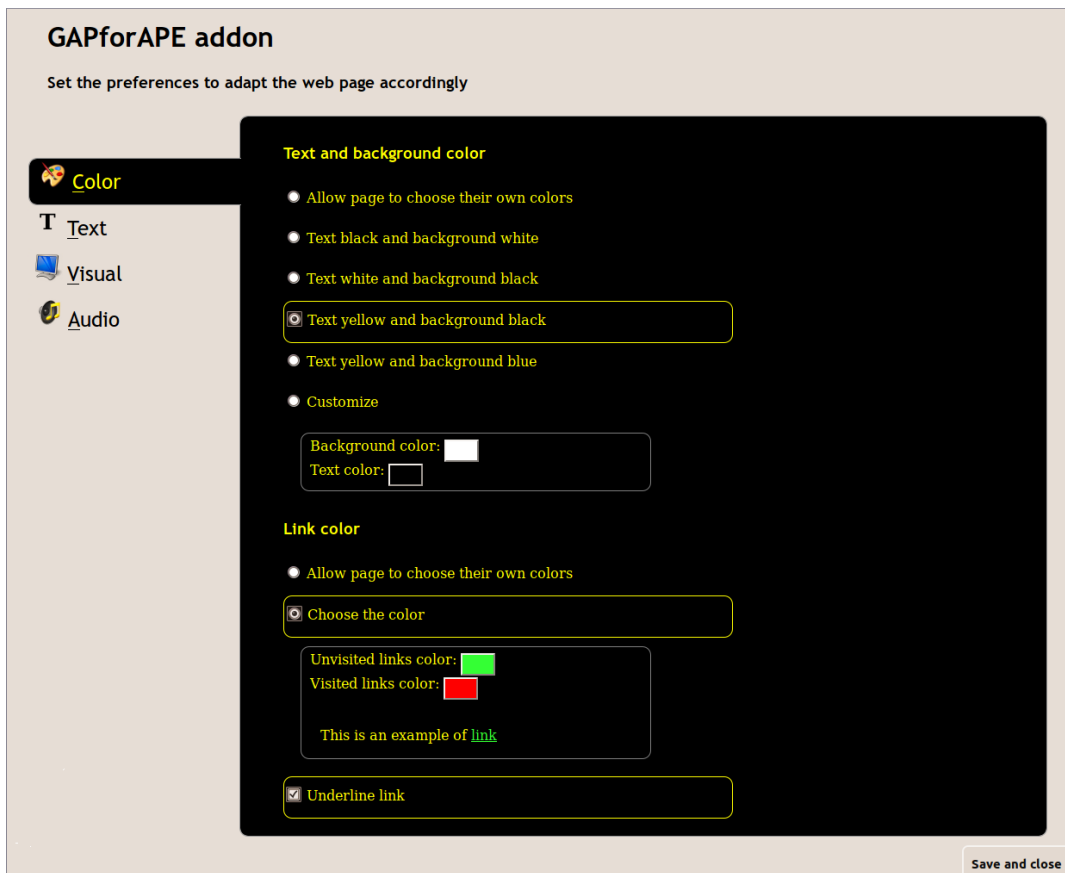


Figura 3.26: Voce di menu Color per uno specifico utente ipovedente

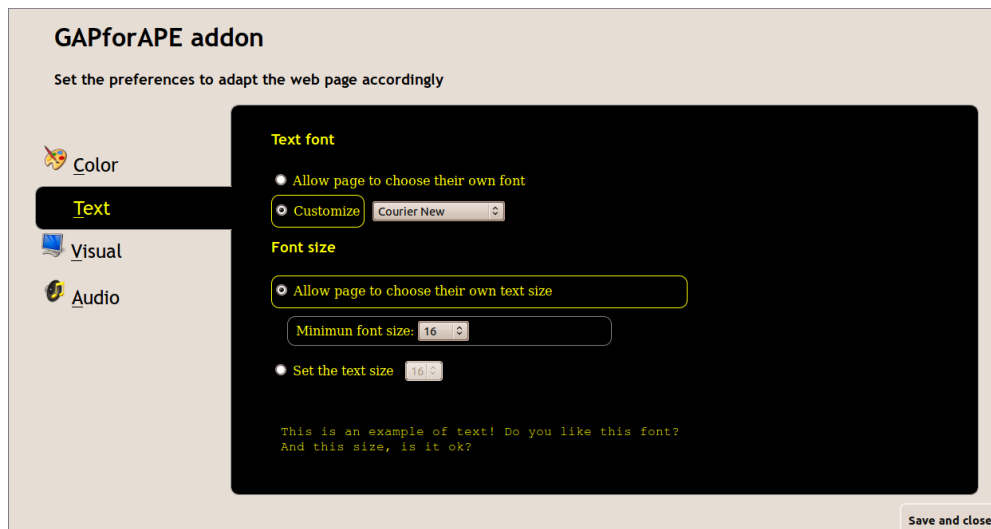


Figura 3.27: Voce di menu Text per uno specifico utente ipovedente

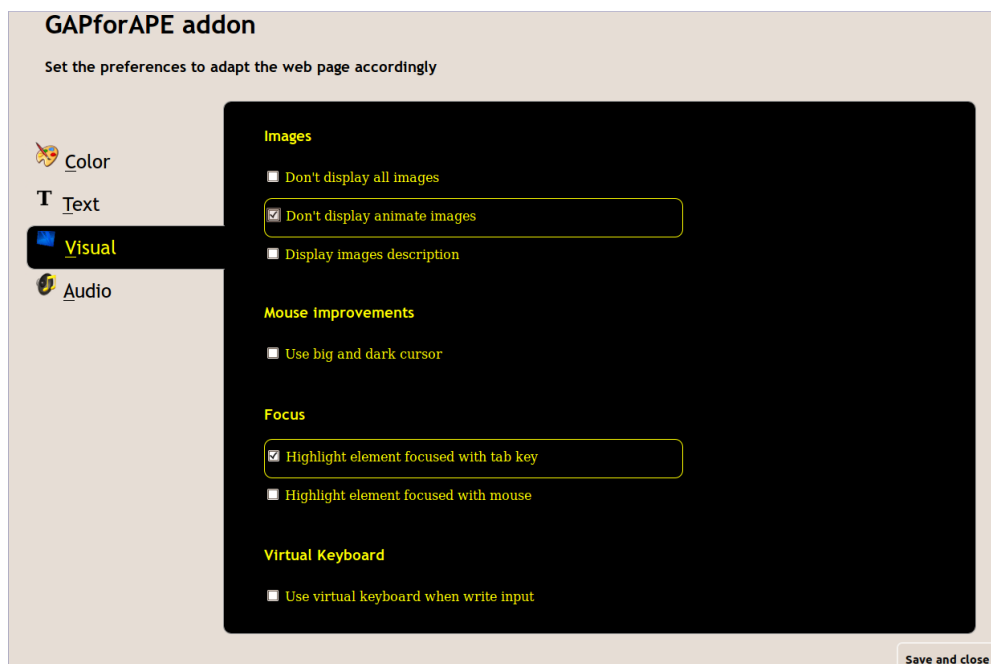


Figura 3.28: Voce di menu Visual per uno specifico utente ipovedente

```
prefs = {
  "color" : {
    "selectedOptColor" : "3",
    "bgColor" : "#000000",
    "textColor" : "#ffff00",
    "selectedOptLink" : "7",
    "unvLinkColor" : "#34ff34",
    "vLinkColor" : "#fe0000",
    "underLink" : "true",
  },
  "text" : {
    "optFont" : "1",
    "font" : "Courier New",
    "optSize" : "2",
    "minSize" : "16"
  },
  "visual" : {
    "image" : "true",
    "animatImage" : "false",
    "bigCursor" : "false",
    "highlightTab" : "true",
    "highlightMouse" : "false",
    "keyboard" : "false"
  }
}
```

Codice 3.6: Oggetto JSON ottenuto creando uno specifico profilo



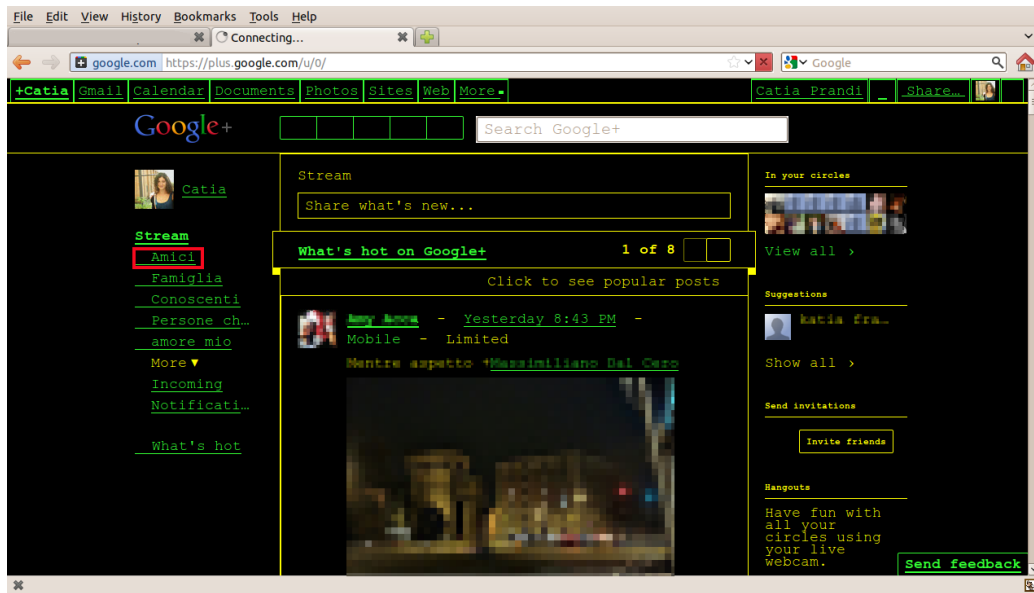


Figura 3.29: Pagina di GooglePlus adattata in base al profilo scelto dall'utente

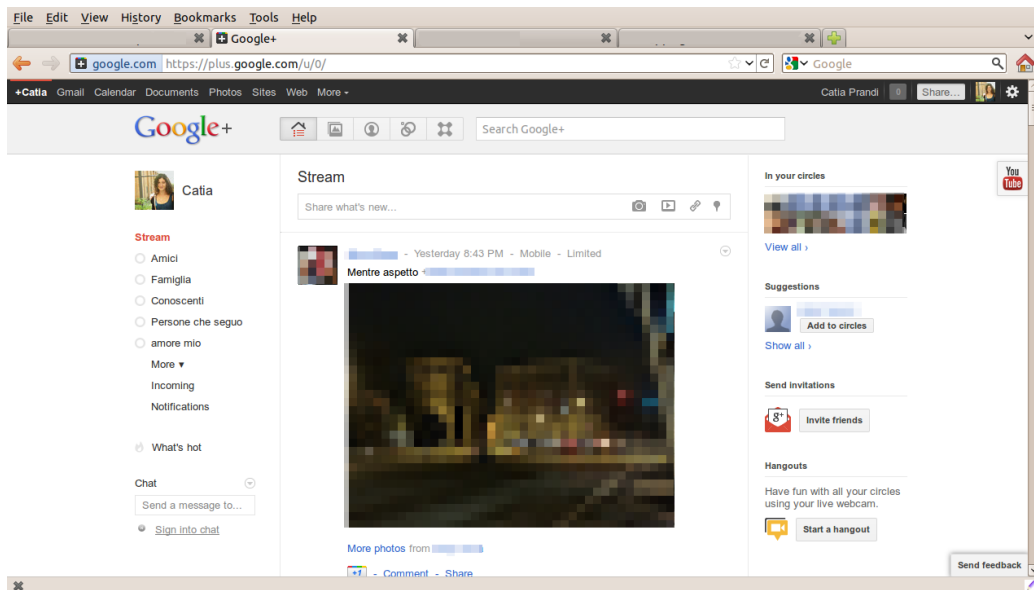


Figura 3.30: Pagina di GooglePlus visualizzata con il profilo di default



# Conclusioni

GAPforAPE è un'estensione per il browser Mozilla Firefox che permette di effettuare la transcodifica e l'adattamento delle pagine Web in base alle necessità e ai bisogni dell'utente, migliorando l'accessibilità del Web.

Il sistema permette di eseguire user script a lato client che modificano il contenuto e la presentazione delle pagine Web, fornendo così "il miglior contenuto a ciascun utente": ad ogni utente viene spedita la stessa pagina che sarà però adattata in modo diverso in base alle necessità dell'utente. Ispirato all'estensione GreaseMonkey e al framework AccessMonkey, fornisce molte funzionalità e vantaggi rispetto ad essi. Per prima cosa, permette all'utente di configurare il proprio profilo permettendo di adattare, in modo completamente automatico, tutti i contenuti Web in accordo con esso. Per descrivere le necessità degli utenti si sono seguite le categorizzazioni utilizzate dallo standard IMS ACCLIP; queste preferenze sono presentate attraverso un'interfaccia usabile ed accessibile, ovvero conforme alle linee guida WCAG 2.0.

Un'altra funzionalità offerta è quella di permettere all'utente di creare o scaricare user script specifici per un certo sito Web. Ciò è importante perché il sistema permette di adattare i contenuti secondo una transcodifica su due livelli. Prima controlla se ci sono user script specifici per quella pagina e quindi adatta la pagina in base a quelli, altrimenti applica solo le modifiche definite dal profilo. Va anche sottolineato che gli adattamenti relativi al proprio profilo vengono applicati anche in questo secondo caso, in modo tale da unire le preferenze specifiche dell'utente a quelle create per quella partico-

lare applicazione Web. Ciò permette di avere effettivamente miglioramenti anche sulle applicazioni Web che sono strettamente basate sulle tecnologie Web 2.0, come AJAX, sulle quali non è possibile agire in un unico modo. Un meccanismo simile è utilizzato dagli screen reader più diffusi, come Jaws.

Un'ulteriore novità introdotta dal nostro sistema, è quella di permettere ad ogni utente di “sistemare” le pagine Web in modo semplice e veloce. “Sistemare” nel senso che non sempre le transcodifiche forniti di default tramite il sistema di profilazione si adattano perfettamente a qualsiasi pagina Web. Per questo si è data la possibilità di modificare on-the-fly le caratteristiche di ogni elemento nella pagina, in modo molto semplice ed intuitivo, e rendere le modifiche permanenti.

Le innovazioni introdotte da GAPforAPE hanno permesso di partecipare e di ricevere il “2011 Best Paper Award” alla conferenza W4A 2011 tenutasi lo scorso marzo, nei giorni 28 e 29, in India (precisamente a Hyderabad).

Gli sviluppi futuri prevedono la realizzazione di un'interfaccia che dia la possibilità di importare e di modificare profili già configurati e complessi (con funzionalità non previste di default dal sistema) in modo da permettere agli utenti di mettere a disposizione il profilo creato, così che, anche altri utenti con le stesse esigenze possano utilizzarlo e beneficiarne. Ciò comporta anche la creazione di un sito Web in cui raccogliere tutti i profili e permetterne il download.

Un altro miglioramento da effettuare è quello di fornire la possibilità di importare facilmente gli user script, senza costringere l'utente ad inserirli manualmente nella cartella giusta all'interno del profilo corrente del browser, in modo da rendere semplice l'utilizzo di script ad hoc per le più diffuse applicazioni Web o per aggiungere una qualche funzionalità ad una pagina specifica.

Altri sviluppi riguarderanno la creazione di nuove funzionalità che permetteranno di rendere più accessibili i contenuti multimediali, grazie alle innovazioni apportate dall'avanzamento delle specifiche di HTML 5.0.

Inoltre, potrebbe essere molto interessante creare GAPforAPE come estensione per gli altri browser, in modo particolare per Google Chrome, per le novità e il potenziale che offre, e per Internet Explorer, per la grande diffusione che continua ad avere, in modo da permettere a sempre più utenti di rendere le pagine Web adatte alle proprie necessità e bisogni senza dover attendere che i siti Web vengano progettati e sviluppati con l'accessibilità in mente.



# Bibliografia

- [1] WAI Guidelines and Techniques. W3C Web Accessibility Initiative. 2011. Disponibile da: <http://www.w3.org/WAI/guid-tech.html>.
- [2] Paola Salomoni, Silvia Mirri, Stefano Ferretti, and Marco Rocchetti. A multimedia broker to support accessible and mobile learning through learning objects adaptation. *ACM Trans. Internet Technol.*, 8:4:1–4:23, February 2008.
- [3] Usage share of web browsers. Wikimedia. 2011. Disponibile da: [http://en.wikipedia.org/wiki/Usage\\_share\\_of\\_web\\_browsers](http://en.wikipedia.org/wiki/Usage_share_of_web_browsers).
- [4] Silvia Mirri, Paola Salomoni, and Catia Prandi. Augment browsing and standard profiling for enhancing web accessibility. In *Proceedings of the International Cross-Disciplinary Conference on Web Accessibility, W4A '11*, pages 5:1–5:10, New York, NY, USA, 2011. ACM.
- [5] CommonJS, Modules, Packages, and the SDK. Mozilla. 2011. Disponibile da: <https://addons.mozilla.org/en-US/developers/docs/sdk/1.2/dev-guide/addon-development/commonjs.html>.
- [6] Scano Roberto. *Libro Bianco: Tecnologie per la disabilità*. Capitolo “La definizione di disabilità”. 2003. Disponibile da: <http://webaccessibile.org/normative/libro-bianco-tecnologie-per-la-disabilita/la-definizione-di-disabilita/>.

- 
- [7] Patrizia Bertini and Marco Trevisan. *Accessibilità e Tecnologie. Dal Web alla telefonia 3G alla domotica: la sfida dell'accesso alla società dell'informazione*. Pearson Education, 2003.
- [8] Michele Diodati. *Accessibilità. Guida completa*. Apogeo, 2007.
- [9] James W. McConnaughey and Wendy Lader. Falling through the net ii: New data on the digital divide. 1998.
- [10] Eszter Hargittai. The digital divide and what to do about it. *First Monday*, pages 1–19, 2003.
- [11] Kerry Dobransky and Eszter Hargittai. The disability divide in internet access and use. *Information Communication Society*, 9(3):313–334, 2006.
- [12] Ben Caldwell, Michael Cooper, Loretta Guarino Reid, and Gregg Vanderheiden. Web Content Accessibility Guidelines (WCAG) 2.0. World Wide Web Consortium Web Accessibility Initiative. 2008. Disponibile da: <http://www.w3.org/TR/WCAG/>.
- [13] Maurizio Boscarol. Che cos'è l'usabilità dei siti web. 2000. Disponibile da: <http://www.usabile.it/012000.htm>.
- [14] Bruce lindsay maguire vs sydney organising. Disponibile da: [http://www.hreoc.gov.au/disability\\_rights/decisions/comdec/Maguire%20v%20SOCOG3.htm](http://www.hreoc.gov.au/disability_rights/decisions/comdec/Maguire%20v%20SOCOG3.htm).
- [15] eEurope 2002. Disponibile da: [http://ec.europa.eu/information\\_society/eeurope/i2010/archive/eeurope/index\\_en.htm#eEurope\\_2002](http://ec.europa.eu/information_society/eeurope/i2010/archive/eeurope/index_en.htm#eEurope_2002).
- [16] Roberto Scano. *IWA ITALY: 10 anni di iniziative*. LULU.COM, 2010.
- [17] Disposizioni per favorire l'accesso dei soggetti disabili agli strumenti informatici. Disponibile da: [http://www.pubbliaccesso.gov.it/normative/legge\\_20040109\\_n4.htm](http://www.pubbliaccesso.gov.it/normative/legge_20040109_n4.htm).



- 
- [18] Verifica tecnica e requisiti tecnici di accessibilità delle applicazioni basate su tecnologie internet. Disponibile da: <http://www.pubbliaccesso.gov.it/normative/DM080705-A.htm>.
- [19] Jan Richards, Jeanne Spellman, and Jutta Treviranus. Authoring tool accessibility guidelines (atag) 2.0. 2011. Disponibile da: <http://www.w3.org/TR/2011/WD-ATAG20-20110721/>.
- [20] James Allan, Kelly Ford, and Jeanne Spellman. User agent accessibility guidelines (uaag) 2.0. 2011. Disponibile da: <http://www.w3.org/TR/2011/WD-UAAG20-20110719/>.
- [21] Tim O'Reilly. What is web 2.0. 2005. Disponibile da: <http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html>.
- [22] Pedro Isaiás, Paula Miranda, and Sara Pífano. Critical success factors for web 2.0 – a reference framework. In *Proceedings of the 3rd International Conference on Online Communities and Social Computing: Held as Part of HCI International 2009*, pages 354–363.
- [23] Facebook. 2011. Disponibile da: <http://www.facebook.com/>.
- [24] Twitter. 2011. Disponibile da: <http://twitter.com/>.
- [25] Youtube is a place to discover, watch, upload and share videos. 2011. Disponibile da: <http://www.youtube.com>.
- [26] Wikipedia, the free encyclopedia that anyone can edit. 2011. Disponibile da: <http://www.wikipedia.org/>.
- [27] Jesse James Garrett. Ajax: A new approach to web applications. 2005. Disponibile da: <http://www.adaptivepath.com/ideas/ajax-new-approach-web-applications>.
- [28] Becky Gibson. Enabling an accessible web 2.0. W4A '07, New York, NY, USA, 2007. ACM.

- [29] Jeremy Allaire. Macromedia flash mx—a next-generation rich client. 2002. Disponibile da: <http://download.macromedia.com/pub/flash/whitepapers/richclient.pdf>.
- [30] Rich internet application. 2011. Disponibile da: [http://it.wikipedia.org/wiki/Rich\\_Internet\\_application](http://it.wikipedia.org/wiki/Rich_Internet_application).
- [31] Mary Zajicek. Web 2.0: hype or happiness? In *Proceedings of the 2007 international cross-disciplinary conference on Web accessibility (W4A)*, W4A '07, pages 35–39, New York, NY, USA, 2007. ACM.
- [32] Cheryl Parsons. Moving from the Digital Divide to Digital Inclusion. *Currents: Scholarship in the Human Services*, 2008.
- [33] Michael Cooper. Accessibility of emerging rich web technologies: Web 2.0 and the semantic web. W4A '07, New York, NY, USA, 2007. ACM.
- [34] James Craig and Michael Cooper. Accessible Rich Internet Applications (WAI-ARIA) 1.0, W3C Candidate Recommendation. 2011. Disponibile da: <http://www.w3.org/TR/wai-aria/>.
- [35] Joshua Hailpern, Loretta Guarino Reid, Richard Boardman, and Srinivas Annam. Web 2.0: Blind to an accessible new world. In *Proceedings of the 18th International Conference on World Wide Web (WWW '09)*, W4A '09, New York, NY, USA, 2009. ACM.
- [36] Myspace. 2011. Disponibile da: <http://www.myspace.com/>.
- [37] Paola Salomoni, Silvia Mirri, Stefano Ferretti, and Marco Roccetti. Profiling learners with special needs for custom e-learning experiences, a closed case? *Higher Education*, 2007.
- [38] Accessibility Research Centre. Profiling for accessibility. Disponibile da: [http://www.slideshare.net/V\\_Gkatzidou/profiling-for-accessibility-presentation](http://www.slideshare.net/V_Gkatzidou/profiling-for-accessibility-presentation).

- [39] World Wide Web Consortium. Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies 1.0 . 2004. Disponibile da: <http://www.w3.org/TR/2004/REC-CCPP-struct-vocab-20040115>.
- [40] Open Mobile Alliance (OMA). User agent profile v. 1.1 approved enabler. 2002. Disponibile da: [http://www.openmobilealliance.org/release\\_program/uap\\_v11.html](http://www.openmobilealliance.org/release_program/uap_v11.html).
- [41] Ims learner information package accessibility for lip. 2002. Disponibile da: <http://www.imsglobal.org/specificationdownload.cfm>.
- [42] IMS Learner Information Profile (LIP). . 2002. Disponibile da: <http://www.imsglobal.org/specificationdownload.cfm>.
- [43] Disponibile da: [http://en.wikipedia.org/wiki/Crowdsourcing#Web-based\\_crowdsourcing](http://en.wikipedia.org/wiki/Crowdsourcing#Web-based_crowdsourcing).
- [44] Terrill Thompson. Crowdsourcing Accessibility: Can Accessibility be fixed for free with Community Help? Disponibile da: <http://staff.washington.edu/tft/talks/ahg10/CrowdSourcing.ppt>.
- [45] Greaspot greasemonkey firefox extension. 2011. Disponibile da: <http://greasemonkey.mozdev.org/>.
- [46] Mark Pilgrim. *Greasemonkey Hacks Tips & Tools for Remixing the Web with Firefox*. O'Reilly Media, 2005.
- [47] Codeidol. Thinking about greasemonkey – accessibility. 2011. Disponibile da: <http://codeidol.com/internet/Greasemonkey/Accessibility>.
- [48] Userscripts.org. Power-ups for your browser: scripts tagged accessibility. 2011. Disponibile da: <http://userscripts.org/tags/accessibility>.
- [49] J P Bigham and R E Ladner. *Accessmonkey: a collaborative scripting framework for web users and developers*, volume 07, page 34. ACM, 2007.

- 
- [50] Jeffrey P Bigham. Accessmonkey: enabling and sharing end user accessibility improvements. *ACM SIGACCESS Accessibility and Computing*, (89):3–6, 2007.
- [51] Webvisum. 2011. Disponibile da: <http://www.webvisum.com/>.
- [52] Alison Lee and Vicki Hanson. *Enhancing web accessibility*, page 456. ACM Press, 2003.
- [53] Beth R. Tibbitts, Susan Crayne, Vicki Hanson, Jonathan Brezin, Cal Swart, and John T. Richards. *HTML Parsing in Java for Accessibility Transformations*. 2002.
- [54] Andrea Mangiatordi and Riccardo Dondi. *Farfalla: a step toward an inclusive web*, pages 87–93. 2010.
- [55] Andrea Mangiatordi and Harpreet Singh Sareen. Farfalla project: browser-based accessibility solutions. In *Proceedings of the International Cross-Disciplinary Conference on Web Accessibility, W4A '11*, pages 21:1–21:2, New York, NY, USA, 2011. ACM.
- [56] Silvia Mirri. *Rich Media Content Adaptation in E-Learning Systems*. PhD thesis, Università di Bologna, 2007.
- [57] Internet explorer. 2011. Disponibile da: <http://windows.microsoft.com/en-US/internet-explorer/products/ie/home>.
- [58] Firefox. 2011. Mozilla. Disponibile da: <http://www.mozilla.org/en-US/firefox/>.
- [59] Chrome. 2011. Disponibile da: [www.google.com/chrome](http://www.google.com/chrome).
- [60] Opera. 2011. Disponibile da: <http://www.opera.com/>.
- [61] Safari. 2011. Apple. Disponibile da: <http://www.apple.com/safari/>.
- [62] Add-ons firefox. 2011. Disponibile da: <https://addons.mozilla.org/en-US/firefox/>.

- 
- [63] Add-on development made easy. 2011. Disponibile da: <https://addons.mozilla.org/en-US/developers/docs/sdk/1.2/>.
- [64] Jetpack. MozillaWIKI, 2011. Disponibile da: <https://wiki.mozilla.org/Labs/Jetpack>.
- [65] JQuery is a new kind of javascript library. 2011. Disponibile da: <http://jquery.org/>.
- [66] Dave Raggett, Arnaud Le Hors, and Ian Jacobs. HTML 4.01 Specification. W3C Recommendation. 1999. Disponibile da: <http://www.w3.org/TR/html401/>.
- [67] HTML. Wikipedia. 2011. Disponibile da: <http://en.wikipedia.org/wiki/HTML>.
- [68] Two Types of Scripts. Mozilla. 2011. Disponibile da: <https://addons.mozilla.org/en-US/developers/docs/sdk/1.2/dev-guide/addon-development/two-types-of-scripts.html>.
- [69] A quick look at google+. 2011. Disponibile da: <http://www.google.com/+/learnmore/>.



# Ringraziamenti

Inizio ringraziando chi mi ha permesso di realizzare questo lavoro di tesi. Ringrazio di cuore la Dott.essa Silvia Mirri che si è sempre dimostrata molto disponibile e che mi ha dato fiducia dal primo giorno in cui mi sono presentata nel suo ufficio. Con lei, ringrazio anche la Prof.essa Paola Salomoni che ha fatto da correlatore a questa tesi, dandoci ottimi consigli e suggerimenti. Ci tengo anche a ringraziare il Prof. Vittorio Ghini per aver accettato di fare da controrelatore decidendo così di interessarsi a questo lavoro.

Ora proseguo con i ringraziamenti alle persone più vicine a me, geneticamente parlando. Ringraziando davvero tanto i miei genitori che mi hanno dato effettivamente i mezzi per poter vivere questi intensi anni di università e che mi hanno sempre sopportata e supportata nelle mie scelte senza mai obbiettare. In modo particolare ringrazio mia madre che, nelle nostre corse, si è sempre dovuta sorbire gli sfoghi dovuti allo “stress da università”. Inoltre ringrazio le mie due nonne che si sono sempre interessate alla mia carriera universitaria nonostante per loro fosse tutto nuovo e strano e qualsiasi cosa gli raccontassi la loro risposta era comunque: “Bene! Brava!”. Ringrazio anche mio fratello perché so che, anche se non mi dice mai nulla, sa sempre tutto ed è molto orgoglioso di me.

Ora è arrivato il momento di ringraziare le mie migliori amiche di sempre: Valentina e Pamela. Le devo ringraziare soprattutto per essermi ancora amiche dal momento che, in questi anni di università, non sono mai andata a festeggiare i loro compleanni, colpa delle date che coincidevano sempre con date di esami. In modo particolare ringrazio Valentina che ha capito che per

me questo ultimo periodo è stato davvero pieno e non si è mai arrabbiata anche se l'ho trascurata in uno dei momenti più belli della sua vita. Inoltre ringrazio Pamela perché, anche se ogni tanto si arrabbia perché esco poco e le dedico poco tempo, le basta vedermi cinque minuti per scordarselo.

Ci tengo a ringraziare anche Giovanni e Cristian, compagni di università e amici sinceri. Cristian lo ringrazio perché, nonostante non si sappia mai dove sia ed è sempre non rintracciabile, è un bravissimo grafico - ha sempre reso le interfacce dei progetti molto accattivanti - e poi mi fa ridere con le sue storie sul perché arriva sempre e costantemente in ritardo agli incontri per i progetti. A parte tutto ciò, è un ottimo amico (part-time). Ringrazio anche Giovanni, per avermi sempre fermata quando avrei voluto uccidere Cristian e perché è un ottimo compagno di progetto - è bravissimo a risolvere i problemi dell'ultimo minuto - ed una persona generosa e sempre disponibile.

Ci tengo anche a ringraziare i miei "amici pelosi", Birba e Buck, che hanno reso meno noiose tutte quelle giornate passate chiuse in casa a programmare, studiare e fare la tesi, regalandomi (senza volerlo) qualche sorriso.

E per ultimo, ma solo per volerlo tenere sulle spine, ringrazio Mattia. Le cose per cui ringraziarlo sono tantissime ma qui mi soffermo solo su quelle che riguardano l'università. Devo ringraziarlo perché mi ha fatto credere in me stessa (non senza difficoltà); mi ha fatto capire che non ci si deve solo limitare a sognare ma che occorre far tutto il possibile per far diventare i desideri realtà, mi ha fatto capire che non si può rimanere immobili ed aspettare che le opportunità ti vengano a cercare ma che occorre andargli incontro e che non si può dire sempre "sì" per non deludere gli altri ma che bisogna anche pensare alla propria felicità ogni tanto. Inoltre è stato proprio lui che, conoscendo i miei interessi, mi ha suggerito di contattare la Dott.essa Silvia Mirri e quindi ha permesso tutto ciò. Grazie di cuore.