

Analisi e specifiche per il supporto dell'Audio Browser Semantico

Survey

Nel presente documento si riassumono i contenuti del deliverable **"D4.1_2 Analisi e specifiche per il supporto dell'Audio Browser Semantico"** relativo all'attività *RI 4.1 Analisi sull'estensione per il supporto dello Screen Reader Semantico*, nell'ambito del quarto Obiettivo Realizzativo (OR 4) "Studio di interfacce multimodali avanzate e dispositivi speciali per i disabili". L'obiettivo di SAPI è quello di dare anche ad un'utenza con disabilità visiva la possibilità di usufruire dei servizi messi a disposizione da Poste Italiane. Per tale motivo l'attività di ricerca è stata orientata ad individuare le soluzioni assistive più consone e soprattutto più efficaci per tali utenti. La presenza di una patologia visiva limita notevolmente l'interazione visuale con la piattaforma e addirittura nel caso peggiore (ovvero quello di cecità assoluta) essa è completamente compromessa. La soluzione proposta da SAPI per chi è affetto da patologie visive e vuole navigare per il web consiste nel fornire a tali utenti la possibilità di interagire in maniera bimodale GUI-VUI in modo simultaneo e sincronizzato. Si dà così la possibilità di interagire in modalità vocale senza precludere la possibilità di operare, sfruttando appieno la capacità visiva residua, sulla parte visibile in modalità GUI. Per gli utenti affetti da cecità assoluta una delle soluzioni proposte è quella di un Audio Browser che consenta la navigazione semantica delle pagine Web attraverso dialoghi vocali tenendo conto anche del comportamento dell'utente sulla base della storia delle sue interazioni. È chiaro che mentre in un ambiente grafico l'utente ha fin dall'inizio una visione d'insieme della pagina che intende visitare in un ambiente totalmente vocale tutto ciò non è possibile. L'Audio Browser SAPI dovrà, in linea generale, essere in grado di:

- presentare le informazioni contenute in una pagina;
- effettuare ricerche mirate;
- raggiungere altre risorse tramite link;
- permettere di muoversi da un punto all'altro della pagina;
- fornire tutte le classiche funzionalità utili alla navigazione.

I navigatori attualmente disponibili e diffusi si basano essenzialmente su due tipi classici di navigazione, ovvero navigazione sequenziale e gerarchica (basata sulla struttura del documento). In entrambi i casi, tuttavia, il processamento delle pagine Web avviene in maniera sequenziale, ed il contenuto "interessante" viene o poco o per niente filtrato, con il conseguente ed inevitabile "information overload". L'obiettivo è stato quindi quello di trovare una modalità di navigazione più pratica ed efficiente che:

- si avvicini al modo di operare quotidiano dell'utente;
- sia il più possibile efficiente rispetto agli interessi e alle preferenze dell'utente (information filtering);

- consenta il browsing tematico e per concetti;
- tenga conto della storia delle interazioni dell'utente in modo da poterne predire il comportamento (User Plan Recognition).

In tal modo, anche ad un primo accesso ad una qualsiasi pagina web, un utente sarà in grado di avere una visione panoramica, a volo d'uccello, almeno sui suoi contenuti tematici come rappresentato in Figura 1:

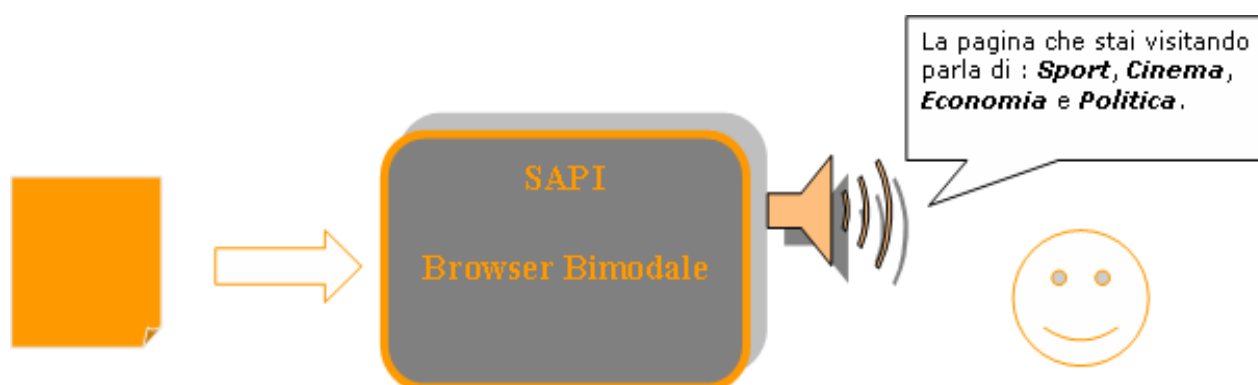


Figura 1: Visione panoramica di una pagina web

L'obiettivo che ci si è prefissi di raggiungere è la progettazione e realizzazione del prototipo di un sistema innovativo che estenda le funzionalità di uno Standard Internet Browser come Internet Explorer in modo tale da farlo apparire all'utente come un browser bimodale (GUI/VUI) durante la fruizione di pagine WEB tradizionali (vale a dire non realizzate ad hoc attraverso l'utilizzo di linguaggi di mark-up multimodali come SALT, EMMA, ... ecc). L'utente usufruendo del sistema proposto potrà usare gli strumenti classici come tastiera e mouse unitamente (in modo sia supplementare che complementare) all'uso di una modalità più naturale rappresentata dall'interazione vocale.

In Figura 2 è rappresentato lo scenario architetturale multimodale Thin Client adottato nella soluzione SAPI. Per "Thin Client" s'intende un dispositivo con poca potenza elaborativa e con limitata capacità di interpretare gli input multimodali. Su un Thin Client sono perciò assenti funzionalità complesse (e costose!) come l'analisi (ASR) e la sintesi (TTS) vocale. Nello scenario architetturale Thin Client le funzionalità ASR e TTS, che richiedono notevole potenza elaborativa, risiedono su un server vocale accessibile tramite la rete IP. Tali funzionalità possono essere richieste dal dispositivo di utente tramite l'utilizzo di protocolli standard o proprietari.

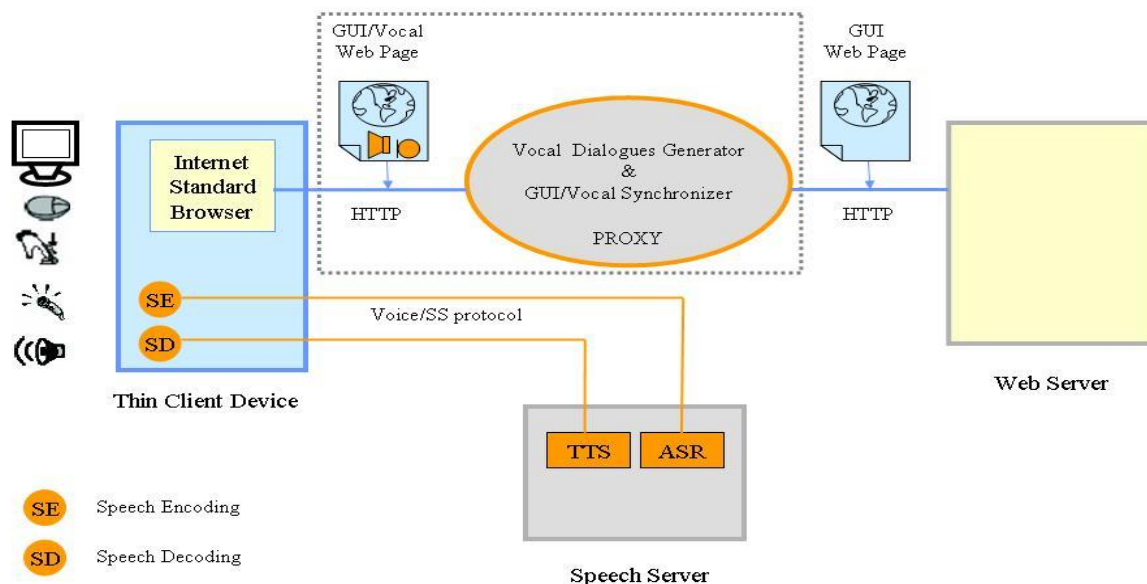


Figura 2: Architettura logica Thin Client – Proxy based adottata in SAPI.

L'area tratteggiata (il PROXY) non fa parte dello scenario canonico Server-based ed è specifica e caratterizzante la soluzione proposta per trasformare on-the-fly le pagine web convenzionali in pagine web bimodali.

Il Proxy ha un ruolo centrale nella realizzazione della bimodalità. Il Proxy è un programma che si frappone tra il Browser e il Web Server inoltrando le richieste e le risposte dall'uno all'altro. In pratica il client (browser) si collega al proxy invece che al server (Web) e gli invia delle richieste. Il Proxy a sua volta si collega al server ed inoltra la richiesta del client, riceve la risposta dal server, e la inoltra indietro al client. Esso interpreta le pagine web richieste dall'utente e le trasforma in modo opportuno per adeguarle all'uso bimodale. Sarà il Browser del client, in maniera del tutto trasparente, ad interagire con il server vocale per la sintesi ed il riconoscimento.

Il vantaggio derivante dall'utilizzo di un Proxy è da ricercarsi nell'assoluta trasparenza di quest'ultimo ovvero l'utente non ne avverte l'esistenza.

In Figura 3 è rappresentata l'architettura funzionale del Proxy SAPI. La Figura riporta sia i principali moduli funzionali costituenti il Proxy sia il workflow del processo di trasformazione della pagina Web convenzionale (*GUI Web Page*) in una pagina Web bimodale (*GUI/VUI*) che il Proxy realizza. Il modulo *Web Page Semantic Analyzer* (presente nel riquadro tratteggiato) che potrebbe essere utilizzato in modo congiunto o in alternativa al modulo *Topic Recognizer* verrà escluso dalla attuale configurazione di SAPI considerato che, rispetto ai fini progettuali, il suo valore aggiunto non giustifica l'effort richiesto dalla sua complessità realizzativa. Questo Analizzatore Semantico non è stato comunque omesso dalla trattazione in quanto si ritiene che esso, così come è stato concepito in SAPI, rappresenti una soluzione di grande interesse per l'analisi morfologica e semantica dei testi ed in particolare per quella orientata al *topic detection*.

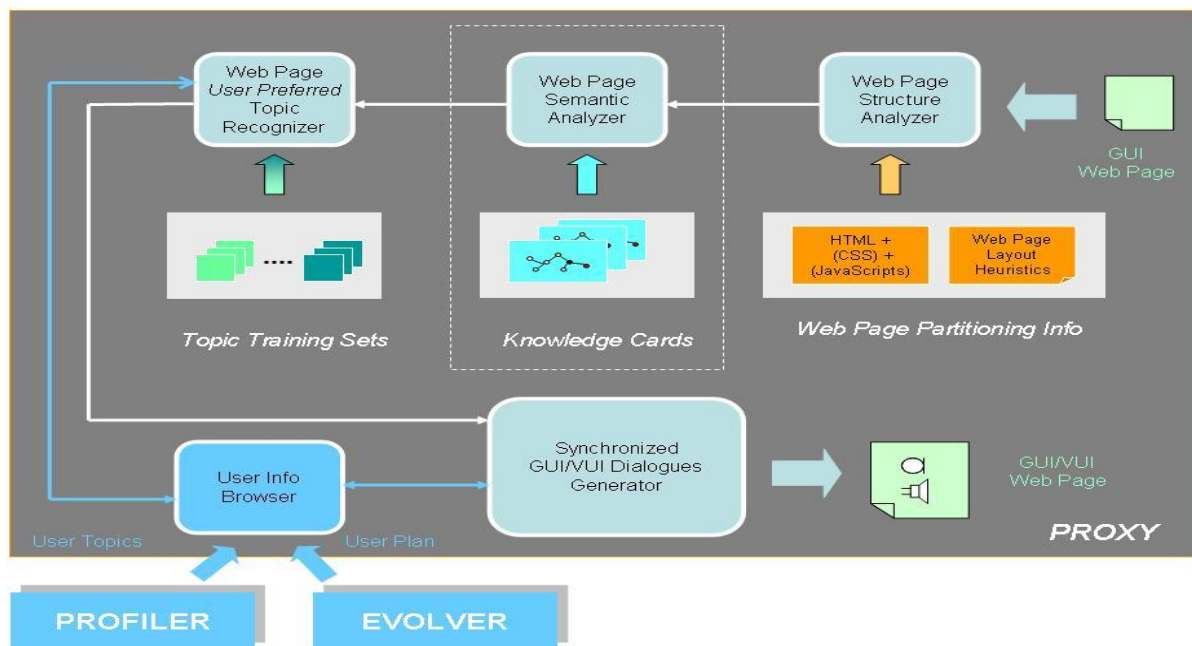


Figura 3: Architettura funzionale del Proxy SAPI

Le principali fasi del processo di trasformazione della pagina Web effettuato dal Proxy sono le seguenti:

- Analisi Strutturale della pagina Web;
- Analisi Semantica della pagina Web;
- Analisi Tematica della pagina Web;
- Generazione dei dialoghi vocali sincronizzati con l'interfaccia GUI (questa fase presuppone che si siano acquisite dal Profiler le preferenze tematiche dell'utente e dall'Evolver le indicazioni predette circa il comportamento dell'utente).

✓ **Analisi Strutturale della pagina Web**

Scopo dell'analisi strutturale di una pagina Web è quello di individuare le diverse sezioni che la compongono sulla base delle regolarità strutturali e di presentazione caratterizzanti le pagine Web convenzionali per poi organizzarle in una struttura gerarchica ad albero. Sono state esplorate diverse tecniche per effettuare l'analisi sintattica di una pagina Web, ciascuna delle quali è riconducibile ad una delle seguenti categorie:

1. Algoritmi basati sui template;
2. Algoritmi basati sull'apprendimento di grammatiche;
3. Algoritmi di segmentazione della pagina.

Gli algoritmi appartenenti alle prime due categorie si basano su assunzioni molto forti e presentano di conseguenza limitazioni significative. In SAPI sono state prese in considerazione le tecniche attualmente esistenti appartenenti alla terza categoria ritenute più

oggettive, più accurate e di più generale applicabilità. In particolar modo sono stati analizzati:

- DOM-based algorithms
- Vision-based algorithms
- Location-based algorithms

Volendo seguire un approccio più generale svincolandosi dalla politica di apertura verso applicazioni terze parti dei singoli produttori di Internet Browser, una valida alternativa sarebbe quella di interpretare direttamente il "sorgente" della pagina Web da visualizzare che nel caso più complesso sarebbe costituito da HTML + CSS (Cascading Style Sheet) + JavaScripts. Ciò equivarrebbe di fatto a realizzare un rendering engine ex-novo il cui risultato di rendering della pagina web, tra l'altro, non è detto che coincida con quello del browser (FireFox, Internet Explorer, Opera, etc.) effettivamente utilizzato sul client. I vari browser infatti, con l'intenzione di offrire ai propri utenti un servizio migliore adottano istruzioni non standard loro proprietarie; inoltre, non tutti i browser, nelle varie versioni più o meno recenti, supportano tutte le istruzioni e non sempre le interpretano nello stesso modo. Di conseguenza non è detto che i vari browser presentino la stessa pagina web nella stessa maniera. Per questo motivo, vale a dire per la mancanza di uniformità comportamentale dei vari motori di rendering attualmente disponibili, aumenta l'effort richiesto per la realizzazione di un interprete del sorgente delle pagine Web convenzionali al fine di effettuare un'analisi strutturale delle pagine Web che tenga conto di come esse verranno effettivamente renderizzate dal particolare Browser presente lato client. L'utilizzo di motori di rendering differenti può inoltre portare a risultati leggermente diversi, dovuti a regole di costruzione dell'albero non uniformi. Per il nostro sistema, però, queste differenze possono risultare trascurabili, non avendo necessità per i nostri scopi di una resa assolutamente precisa della pagina, quanto di una buona approssimazione della struttura bidimensionale dell'ipertesto che ci consenta di generare i dialoghi vocali.

L'approccio proposto in SAPI per l'analisi strutturale delle pagine Web convenzionali si basa perciò su di un interprete DHTML in grado di codificare su richiesta di una Java Application il layout di una pagina Web così come apparirebbe all'utente una volta renderizzata dallo standard internet browser presente sul terminale client. Tale rendering engine "virtuale" presenterà inoltre una Application Programming interface (API) attraverso la quale una Java Application potrà acquisire tutte le informazioni (spaziature orizzontali e verticali, stili di presentazione, ...) relative alle entità costituenti la pagina. In SAPI quest' interfaccia verrà utilizzata nelle fasi di *Page Structure Extraction* e *Page Segmentation* del processo di partizionamento della pagina web per acquisire le informazioni di tipo geometrico e di stile relative agli elementi costituenti la pagina. Utilizzeremo tali informazioni per segmentare il contenuto della pagina ottenendo blocchi e partizioni come mostrato in Figura 4.

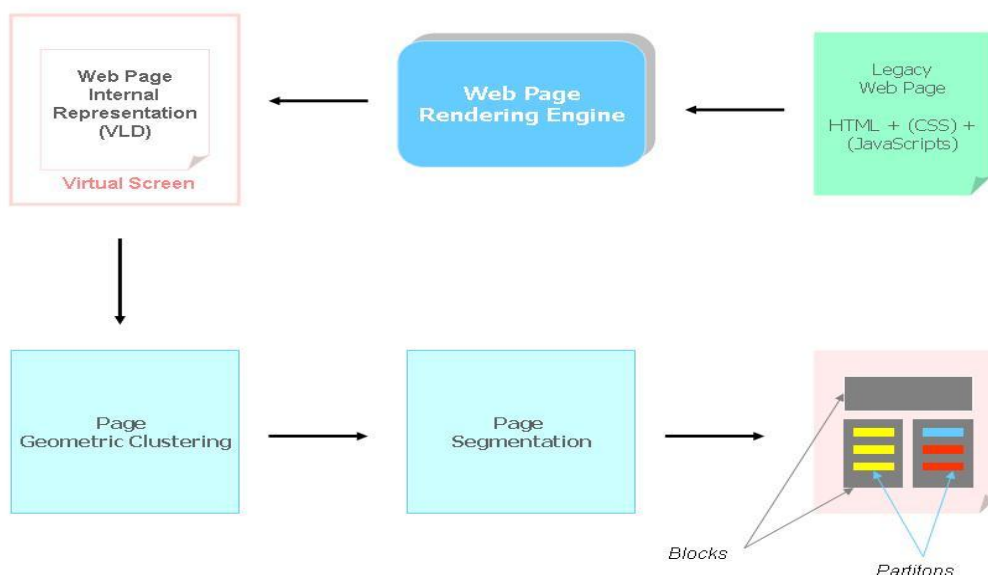


Figura 4: Processo di partizionamento di una pagina web convenzionale.

✓ **Analisi Semantica della pagina Web**

Scopo dell'analisi semantica di una pagina web è quello di partizionare la pagina in sezioni semanticamente distinte pur rimanendo agnostico circa i concetti e gli argomenti trattati nelle singole sezioni. In SAPI il partizionamento semantico di una pagina Web è ottenuto automaticamente a valle dell'analisi strutturale della stessa: le sezioni che si presume siano semanticamente diverse sono rappresentate dai blocchi e dalle eventuali partizioni in essi contenute. Ciò è dovuto sostanzialmente al particolare approccio adottato in SAPI e alla singolare natura semi-strutturata dell'entità da segmentare: la pagina Web.

✓ **Analisi Tematica della pagina Web**

Scopo dell'analisi tematica di una pagina web è quello di partizionare la pagina in sezioni tematicamente distinte in modo da poter associare ad ogni sezione un'etichetta tematica. Tale etichetta può essere usata come voce (vocal tag/voice label) di un menu vocale e conseguentemente come elemento della grammatica che verrà dinamicamente generata per poi essere utilizzata durante la fase di riconoscimento vocale dei comandi di utente. I due approcci presi in considerazione in SAPI per la realizzazione di questa funzionalità sono:

- Approccio misto *NLP – Knowledge based*
- Approccio basato su un Classificatore tematico N-ario

Entrambi gli approcci rappresentano soluzioni scalabili: il primo per *dominio applicativo*, il secondo per *tematica di interesse*. In SAPI sarà utilizzato un categorizzatore testuale n-ario opportunamente addestrato a riconoscere n argomenti comprensivi delle tematiche di interesse dell'utente.

✓ **Generazione automatica dei dialoghi vocali**

A valle del partizionamento tematico della pagina il Proxy effettua le seguenti operazioni:

- associa le varie sezioni tematiche ai corrispondenti link presenti in esse;
- acquisisce dal Profiler le preferenze tematiche dichiarate esplicitamente dall'utente;
- acquisisce dall' Evolver la predizione del comportamento dell'utente (predizione del prossimo tema e/o del prossimo link successivo che verrà selezionato dall'utente).

Per ciascun tema presente nella pagina il Proxy memorizza le seguenti informazioni:

- la corrispondente etichetta vocale;
- il testo da sintetizzare;
- gli eventuali collegamenti ipertestuali relativi.

Per ogni collegamento ipertestuale memorizza:

- il contenuto: il testo o il nome dell'immagine che risulta cliccabile;
- la risorsa: la risorsa web verso cui il collegamento punta (può essere un'altra pagina o anche un punto della pagina stessa);
- una sua eventuale descrizione (il testo da sintetizzare).

Ai blocchi e/o alle partizioni della pagina a cui il classificatore n-ario non è stato in grado di assegnare un' etichetta tematica, verrà assegnata una tag vocale secondo il seguente criterio:

- verrà preso come etichetta vocale il contenuto del link se si tratta di testo descrittivo un collegamento ipertestuale;
- il titolo se è presente;
- le parole scritte in grassetto e/o aventi dimensione del carattere più grande della size del carattere dell' eventuale testo che precedono e a cui sono adiacenti.

Queste informazioni sono necessarie per preparare i dialoghi vocali; rappresentano, infatti, le frasi da sintetizzare e le grammatiche dinamiche per il riconoscimento vocale. Gli array contenenti tali informazioni sono letteralmente "aggiunti" al codice HTML della pagina richiesta, sotto forma di codice javascript. Il codice in questione contiene le istruzioni che permettono un flusso ordinato e sincronizzato del dialogo vocale; tale dialogo è generato a partire dai dati contenuti negli array costruiti in precedenza. A questo punto la pagina è pronta per essere inviata al Web browser (client) da cui è partita la richiesta e l'utente può navigare secondo la modalità preferita tra la GUI e la modalità vocale, o anche sovrapporle. Infatti, proprio perché tutto è gestito dal Proxy, si riesce a realizzare appieno il concetto di bimodalità. L'utente che naviga, può scegliere (sia con la voce che con il click del mouse) un link attraverso cui accedere alla nuova pagina. La sincronizzazione tra le due modalità vocale e visuale, è assicurata intrinsecamente dalla presenza nella stessa pagina dal codice generato dal Proxy che funge da *Interaction Manager*.

In definitiva il Proxy sarà in grado di generare tre tipi di dialoghi tematici:

1. Dialoghi tematici predetti;
2. Dialoghi tematici relativi alle preferenze dichiarate dell'utente;
3. Dialoghi tematici assoluti.