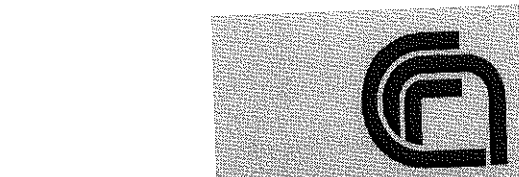


Consiglio Nazionale delle Ricerche

I principi e le tecnologie utili alla realizzazione di una interfaccia web accessibile alle persone disabili

Vittorio Miori, Gabriele Tolomei

B4-23
dic-2002

I principi e le tecnologie utili alla realizzazione di una interfaccia web accessibile alle persone disabili

Vittorio Miori
Gabriele Tolomei

ISTI B4-22
Dicembre 2002

INDICE

PREFAZIONE

Il documento, dopo aver esplorato i principi generali di accessibilità e usabilità nel mondo web, offre una descrizione delle tecnologie utilizzate per la progettazione e realizzazione di un prototipo di interfaccia web, per un sistema informativo distribuito basato sulla rete, rispondente ai criteri di progettazione universale.

La rassegna di tali fondamenti teorici, fa parte del lavoro svolto per lo studio rivolto alla realizzazione di un'interfaccia Web accessibile alle persone disabili, nell'ambito di un progetto congiunto ISTI-CNR e Ministero dell'Ambiente, denominato INFEA.

Si rimanda al documento ISTI B4-23, per le ulteriori informazioni e approfondimenti riguardo le soluzioni tecnologiche specificatamente adottate e la descrizione delle modalità di realizzazione del prototipo.

INTRODUZIONE

Abbattere le barriere.

Si è fatto un gran parlare a suo tempo di riorganizzare l'urbanistica e la burocrazia così da migliorare la qualità di vita dei disabili, garantendo loro l'accesso ad ogni risorsa comune e ad ogni servizio di pubblica utilità. Sono stati varati decreti ed emanate leggi, gli animi si sono placati e l'interesse per questa problematica è andato via via scemando.

E oggi siamo nuovamente a questo punto: abbattere le barriere.

Ogni nuovo mondo che si crea, e tale è il World Wide Web, è costruito intorno all'uomo medio. Internet è a tutti gli effetti una risorsa, un servizio, e come tale dovrebbe essere accessibile a qualunque utente, non solo a chi rientra in un modello basato su numeri e statistiche. Per chi non possiede certi attributi, spuntano allora nuovi e diversi ostacoli.

E ci vuole tempo, impegno e costanza per buttarli giù.

Il lavoro svolto per adattare una parte del sistema INFEA alle norme del *W3C (World Wide Web Consortium)* [b] sui temi di accessibilità e usabilità Internet ha dato luogo ad un prototipo di interfaccia basata su browser web, limitato alla porzione più rilevante dell'intero sistema.

In particolare, con l'impegno di rendere accessibile il sistema agli utenti disabili e quindi superare il problema di *accessibilità*, si è aggiunto, ma già in parte risolto, quello di realizzare l'interfaccia in modo da favorire la fruibilità di informazioni a qualsiasi categoria di utenza (problema di *usabilità*).

Accessibilità ed usabilità, seppur concetti distinti, presentano infatti contiguità ed intersezione; molto spesso, eliminando le limitazioni imposte agli utenti disabili, si finisce per apportare un beneficio alla generalità degli utenti.

---

[b] http://w3.org

1. ACCESSIBILITA' DEL WEB [14] [15]

Questa fase di lavoro racchiude sia lo studio approfondito delle norme e degli standard in tema di accessibilità ed usabilità, sia l'applicazione di questi concetti al caso specifico del sistema INFEA durante la progettazione dell'interfaccia web.

1.1 Il web, Internet e il W3C

Secondo l'opinione di molti, il web è una delle invenzioni più importanti del secolo ormai passato; per tutti, il web è divenuto l'interfaccia standard per l'accesso alla generalità di applicazioni e alla moltitudine di informazioni messe a disposizione dalla rete. Per questo motivo, parlare dei problemi di accesso all'informazione introdotti dalle nuove tecnologie significa, in gran parte, parlare dell'accessibilità al web.

Il ruolo più importante nel complesso sistema governativo per lo sviluppo di nuove tecnologie Internet, e quindi anche per i problemi legati all'accessibilità alla rete, è svolto dal World Wide Web Consortium (W3C).

Per dare una definizione astratta, il W3C è il consorzio che si occupa della pubblicazione di linee guida a favore dell'evoluzione del web; concretamente si è occupato di definire molti standard tra cui, solo per citare il più noto, HTML.

La ben nota facilità di comunicazione resa possibile grazie al web si deve soprattutto all'impiego di un linguaggio standard per la pubblicazione di documenti, come HTML, interpretato allo stesso modo (o quasi) da tutti i browser.

La necessità di adottare standard comuni si è fortemente intensificata in molte altre tecnologie del web e di Intenet in generale, andando a costituire uno dei temi fondamentali del governo di Internet, ciò che oggi si usa chiamare *Open Internet Governance*.

1.2 Il funzionamento del W3C

Il W3C è un'organizzazione non a fini di lucro, finanziata soprattutto dai membri che ne fanno parte (attualmente oltre 500) e in misura minore da organizzazioni esterne coinvolte in progetti ad hoc.

Il W3C ha tre sedi: una presso il M.I.T. a Boston, dove risiede il suo direttore e fondatore Tim Berners-Lee, presso l'Inria, in Francia e presso l'Università di Keio, in Giappone.

La missione ufficiale del Consorzio è quella di portare il web al massimo del suo potenziale, sviluppando dei protocolli comuni in grado di promuovere la sua evoluzione e assicurare la sua interoperabilità; la realizzazione di questa missione è sostenuta dal perseguimento di alcuni obiettivi principali, tra cui la garanzia di un web accessibile a tutti.

Per rispettare questo impegno, sin dal 1997 il W3C ha sviluppato la Web Accessibility Initiative (WAI), guidata da Judy Brewer, formalizzando le proposte di un intervento, mirato esplicitamente al tema dell'accessibilità del web.

1.3 Web Accessibility Initiative (WAI)

Web Accessibility Inititative del W3C si è occupata, e si occupa tuttora, di un ampio spettro di attività che vanno dalla produzione di raccomandazioni (documenti tecnici che hanno raccolto un adeguato consenso e rappresentano la posizione ufficiale del W3C in merito ai temi di riferimento), alle opere di sensibilizzazione e di formazione affinché queste siano adottate, fino alla produzione vera e propria di soluzioni tecniche a titolo esemplificativo.

Un'esigenza primaria per la produzione di documenti tecnici relativi all'accessibilità, ma estendibile ad ogni area di intervento del Consorzio, riguarda il loro coordinamento con l'insieme delle raccomandazioni prodotte; questo compito di armonizzare le attività WAI con lo sviluppo di altre tecnologie collegate è svolto, proprio all'interno di WAI, da *WAI Technical Activity*. Più in generale, WAI Technical Activity si occupa della redazione dei documenti tecnici come le raccomandazioni, definite in questo caso linee guida sull'accessibilità.

Il W3C evidenzia tre linee guida per l'accessibilità [f]:

- Linee guida sull'accessibilità dei contenuti (Web Content Accessibility Guidelines, WCAG 1.0)

Una raccomandazione datata 5 maggio 1999. Queste linee guida spiegano i criteri e le soluzioni tecniche affinché un documento web risulti accessibile agli utenti disabili.

- Linee guida per l'accessibilità degli authoring tool (Authoring Tool Accessibility Guidelines, ATAG 1.0)

Una raccomandazione datata 3 febbraio 2000. Un authoring tool è un programma per la produzione di contenuti web (come ad esempio Macromedia Dreamweaver); le ATAG 1.0 hanno come obiettivo sia l'accessibilità dello stesso authoring tool, sia l'accessibilità del contenuto web che esso permette di creare.

- Linee guida per l'accessibilità di user agent (User Agent Accessibility Guidelines, UAAG 1.0)

Una raccomandazione candidata, e per questo non ancora ufficiale, datata 12 settembre 2001. Uno user agent è un programma che permette di accedere ai contenuti del web come ad esempio un browser (Microsoft Internet Explorer o Netscape Navigator), un lettore multimediale, ma anche un programma che, grazie ad un sintetizzatore vocale, legge una pagina web. L'obiettivo delle UAAG 1.0 è quello di agevolare la realizzazione di user agent in grado di ridurre le barriere di accessibilità ai contenuti del web.

A ciascuna raccomandazione è associato un documento di check list, una sorta di appendice alle linee guida, che indica le priorità suggerite per raggiungere gli obiettivi dichiarati; per le tre linee guida sopraelencate sono stati individuati tre diversi livelli di priorità.

---

[f] http://www.w3.org/TR/WCAG/

1.4 Gli sviluppi attuali delle attività WAI

Nel corso del 2000 è iniziato un lavoro di revisione delle linee guida sui contenuti del web (WCAG 1.0), il cui obiettivo finale è la pubblicazione di un insieme di linee guida aggiornate.

Sino ad ora il lavoro ha prodotto una bozza (working draft) che rappresenta solo il primo passo dell'iter seguito per la pubblicazione di una raccomandazione del W3C. Malgrado manchi di qualsiasi carattere di ufficialità e di qualsiasi forma di consenso da parte dei membri del Consorzio, la bozza di lavoro rappresenta un'utile indicazione di come evolveranno le attuali WCAG 1.0. L'obiettivo resta il medesimo – indicare come produrre contenuti web accessibili – e le soluzioni sono in gran parte analoghe; sostanzialmente però è diverso il modo in cui queste sono presentate, ponendo maggior enfasi sui checkpoint, cioè su quei controlli tesi ad evitare che un documento risulti poco accessibile. Inoltre, si considera in modo più esplicito il rapporto tra accessibilità e usabilità dei siti web; per usabilità dei siti web si intende in generale la facilità con cui un utente può 'usarli' (ad esesmpio, trovare agevolmente l'informazione che cerca, orientarsi nella navigazione, apprendere facilmente la struttura del sito). Si tratta quindi di un aspetto particolare del problema più ampio della realizzazione di interfacce uomo-macchina.

Negli ultimi anni si è capito che vi sono rapporti abbastanza stretti, anche se sfumati e imprecisabili, tra l'accessibilità e l'usabilità di un sito; in WCAG 1.0 si accenna al fatto che contenuti più accessibili producono anche una facilitazione d'uso da parte di ogni categoria di utenza, non soltanto da quella costituita da persone disabili.

In WCAG 1.0 si trattano temi legati all'usabilità, soprattutto le linee guida 12, 13, 14. La bozza, invece sembra essersi trasformata in un documento sostanzialmente a due livelli: un livello superiore che tratta le questioni in termini generali, dove si trovano le linee guida che da 14 scendono a 4, e un livello inferiore in cui sono fornite informazioni puntuali sull'uso delle tecnologie rilevanti.

Quanto indicato rappresenta solamente la direzione intrapresa dal gruppo WAI del W3C; per il risultato finale, si dovrà attendere il completamento del processo di costruzione del consenso attorno a

questa bozza, che avverrà, verosimilmente, non prima della fine dell'anno.

1.5 Le linee guida sull'accessibilità (WCAG 1.0) [16]

Le WCAG 1.0 sono organizzate in 14 linee guida:

1) Fornire alternative equivalenti al contenuto visivo e acustico

Benché alcune persone non possano usare immagini, film, suoni, Applet... possono comunque interagire con pagine che includono un'informazione equivalente al contenuto visivo o acustico; l'informazione equivalente realizzata avrà lo stesso scopo del contenuto visivo o acustico. Questa linea guida rimarca l'importanza di fornire equivalenti testuali al contenuto non testuale, e di fornire equivalenti non testuali di un testo scritto.

Considerando che molti utenti con disabilità visive, soprattutto non esperti, utilizzano browser non grafici (perlopiù Lynx), offrire un'alternativa testuale ad un'immagine rende più comprensibile il contenuto dell'intero documento web; tecnicamente tutto ciò è possibile grazie all'impiego dell'attributo ALT per gli elementi IMG (immagini) fornito dallo standard HTML. Analogamente, l'inserimento di immagini, video o audio pre-registrati al posto di semplice testo può facilitare la comprensione ad utenti illetterati o con disturbi a livello di lettura.

2) Non fare affidamento solo sul colore

È opportuno evitare che un'informazione rilevante sia veicolata da un colore o dal contrasto tra colori, in modo da permettere anche alle persone daltoniche o a coloro che possiedono schermi in bianco e nero di beneficiare della stessa informazione.

3) Utilizzare le annotazioni (markup) di HTML ed i fogli di stile in modo appropriato

L'idea di fondo su cui si basa questa linea guida è la necessità di mantenere distinte le informazioni che sono contenute in una pagina, dal modo in cui esse vengono presentate graficamente all'utente, sia evitando l'utilizzo improprio dei markup di HTML, sia applicando degli opportuni fogli di stile.

4) Chiarire l'uso di linguaggi naturali

Questa linea guida propone l'utilizzo di marcatori che facilitino la pronuncia o l'interpretazione di testi stranieri o abbreviati; infatti, contrassegnando in un documento diversi linguaggi naturali, le sintesi vocali e le periferiche braille possono selezionare automaticamente la nuova lingua, rendendo il documento più accessibile agli utenti multilingue. Il contrassegno del linguaggio naturale, inoltre, consente a tutti la leggibilità del web, compresi gli utenti sordi e con difficoltà di apprendimento.

Un esempio che questa linea guida propone è quello di utilizzare l'attributo LANG associato all'elemento HTML.

5) Creare tabelle che si trasformino in maniera elegante

È necessario assicurarsi che le tabelle create abbiano la marcatura necessaria per essere trasformate dai browser accessibili e da altri interpreti. Innanzitutto però sarebbe preferibile che le tabelle fossero usate per marcare informazioni strettamente tabellari – tabelle di dati –, più che per l'impaginazione di un documento – tabelle di impaginazione.

Le tabelle, per qualsiasi scopo siano impiegate, presentano comunque problemi particolari per gli utenti che utilizzano lettori di schermo ma, realizzando una marcatura corretta, gli interpreti consentono una navigazione completa tra le celle delle tabelle, compreso l'accesso alle intestazioni e ad altre informazioni contenute nelle celle.

Ad esempio, per le tabelle di dati occorre identificare le intestazioni di righe e colonne; in HTML, attraverso l'elemento TD si identificano le celle di dati e con TH le intestazioni. Qualora fosse utilizzata una tabella per l'impaginazione, è opportuno assicurarsi che questa sia comprensibile anche se letta in modo linearizzato; altrimenti si rende necessario fornire un'alternativa equivalente.

6) Assicurarsi che le pagine che danno spazio a nuove tecnologie si trasformino in maniera elegante

Malgrado gli sviluppatori siano incoraggiati ad usare nuove tecnologie che risolvano i problemi creati da quelle già esistenti, è necessario che i documenti prodotti possano essere completi e comprensibili anche se visualizzati con browser non aggiornati o da persone che scelgono di disabilitare alcune caratteristiche.

Ad esempio, i documenti devono essere organizzati in modo da poter essere letti senza i fogli di stile associati e le pagine devono essere utilizzabili quando script, Applet, o altri oggetti di programmazione sono disabilitati oppure non supportati.

7) Assicurarsi che l'utente possa tenere sotto controllo i cambiamenti di contenuto nel corso del tempo

È necessario che gli oggetti in movimento, lampeggianti, scorrevoli o che si autoaggiornano possano essere arrestati temporaneamente o definitivamente. Alcune persone con disabilità cognitive o visive non riescono a leggere testo in movimento con velocità sufficiente, oppure non sono in grado di leggerlo affatto. I lettori di schermo non sono in grado di leggere testo scorrevole e alcune persone con disabilità fisiche potrebbero non essere in grado di muoversi con velocità o precisione sufficienti per interagire con oggetti in movimento.

Ad esempio, fin quando gli interpreti non permetteranno agli utenti di controllare lo sfarfallio, è bene evitare di far sfarfallare lo schermo: persone con epilessia fotosensibile possono avere crisi scatenate da sfarfallio oppure da lampeggiamenti nell'intervallo che va

da 4 a 59 lampi al secondo (Hertz), con un picco intorno ai 20 lampi al secondo. Oppure, fino a quando gli interpreti non permetteranno agli utenti di bloccare il contenuto in movimento, si deve evitare il movimento delle pagine.

Inoltre gli elementi BLINK e MARQUEE non sono definiti in alcuna delle specifiche HTML del W3C e non dovrebbero essere usate.

8) Garantire l'accessibilità diretta delle interfacce utente incorporate

La progettazione delle interfacce utente deve seguire i principi di accessibilità: accesso alle diverse funzionalità indipendente dai dispositivi usati, possibilità di operare da tastiera, comandi vocali...

Quando un oggetto incorporato possiede una propria interfaccia, questa, proprio come l'interfaccia del browser, deve essere accessibile; per informazioni più dettagliate sulle interfacce accessibili, è utile consultare UAAG 1.0 e ATAG 1.0.

9) Progettare per garantire l'indipendenza da dispositivo

L'accesso indipendente dal dispositivo significa che gli utenti possono interagire con l'interprete o con il documento tramite il dispositivo di input o di output preferito (mouse, tastiera, voce, bacchette manovrate con la testa...); fornendo equivalenti testuali per immagini sensibili o per immagini usate come collegamento, si dà la possibilità di interagire con esse senza un dispositivo di puntamento.

10) Usare soluzioni provvisorie

L'impiego di soluzioni provvisorie garantisce che le tecnologie assistive e i browser più vecchi possano operare correttamente.

Questa linea guida e i relativi punti di controllo sono stati etichettati come provvisori, in quanto il gruppo di lavoro sulle WCAG 1.0 li ritiene necessari al momento della pubblicazione del documento WCAG ma eliminabili in un prossimo futuro, quando le tecnologie web avranno incorporato le capacità e le caratteristiche adatte.

11) Usare le tecnologie e le raccomandazioni del W3C

Si incoraggia lo sviluppatore ad usare le tecnologie del W3C, in conformità con le specifiche, e a seguire le raccomandazioni sull'accessibilità. Nei casi in cui non sia possibile usare una tecnologia del W3C, oppure se nell'utilizzarla si ottenesse materiale che non si trasforma in maniera elegante, è utile fornire una versione alternativa del contenuto che sia accessibile.

Molti formati che non sono del W3C (per esempio PDF, Schockwave...) richiedono di essere visti o con plug-in o con applicazioni autonome; spesso questi formati non possono essere visualizzati oppure non è possibile effettuare una navigazione con interpreti standard, comprese le tecnologie assistive. Evitare l'uso di caratteristiche non W3C e non standard (elementi, attributi, proprietà ed estensioni proprietarie) aiuterà a rendere le pagine più accessibili ad un numero maggiore di persone che usino una più ampia varietà di hardware e software. Quando devono essere usate tecnologie non accessibili, proprietarie o meno, devono essere fornite pagine equivalenti accessibili. La conversione di documenti in formato PDF, PostScript, RTF... in linguaggi di marcatura del W3C, come HTML o XML, non sempre crea un documento accessibile. Se, nonostante ogni sforzo, non si può creare una pagina accessibile, è necessario fornire un collegamento a una pagina alternativa che usi le tecnologie W3C, che sia accessibile, che contenga informazioni o funzionalità equivalenti e che sia aggiornata con la stessa frequenza della pagina originale inaccessibile. Proprio per quest'ultimo motivo, vista la difficoltà con la quale le pagine alternative possono mantenersi aggiornate, l'utilizzo di due versioni di uno stesso documento è fortemente scoraggiato. Una pagina non aggiornata può essere frustrante quanto una pagina non accessibile visto che, in entrambi i casi, l'informazione presentata nella pagina originale non è disponibile. Prima di ricorrere ad una pagina alternativa, è bene riesaminare il progetto della pagina originale; è probabile che rendendolo accessibile esso risulti migliore per tutti gli utenti.

12) Fornire informazione per la contestualizzazione e l'orientamento

Relazioni complesse fra parti diverse di una stessa pagina possono essere difficili da interpretare per persone con disabilità visive o cognitive; raggruppare gli elementi e fornire informazioni riguardo alla loro relazione è utile per tutti i generi di utenza.

Ad esempio, si consiglia di dare un titolo ad ogni frame per facilitare l'identificazione del frame e la navigazione (usare l'attributo TITLE in HTML per l'elemento FRAME), nonché di descrivere lo scopo dei frame e il modo in cui essi interagiscono (in HTML usare l'attributo LONGDESC oppure corredare il frame di un collegamento descrittivo).

13) Fornire chiari meccanismi di navigazione

I meccanismi di navigazione quali mappe del sito, barre di navigazione, informazioni per l'orientamento, aumentano la probabilità che una persona trovi ciò che effettivamente sta cercando all'interno di un sito, soprattutto se l'utente presenta invalidità cognitive o visive.

Per questo motivo, è utile identificare con chiarezza l'obiettivo di ogni collegamento: un collegamento testuale dovrebbe essere abbastanza significativo da mantenere un senso anche se letto fuori contesto. In aggiunta ad un chiaro collegamento testuale, è possibile fornire una descrizione più dettagliata del collegamento attraverso l'attributo TITLE che HTML mette a disposizione.

Inoltre è consigliabile fornire informazioni sulla configurazione del sito, ad esempio tramite una mappa, e barre di navigazione per evidenziare e accedere ai vari meccanismi di navigazione.

14) Assicurarsi che i documenti siano chiari e semplici

Una disposizione coerente della pagina, una grafica riconoscibile ed un linguaggio facile da capire giovano a tutti gli utenti, e in particolare aiutano quelle persone con disabilità cognitive o con difficoltà di lettura.

L'uso di un linguaggio chiaro e semplice promuove una comunicazione efficace e risulta importante anche per le persone la cui madrelingua sia diversa da quella con cui è presentato il documento,

compresi gli utenti che comunicano principalmente attraverso la lingua dei segni.

Per verificare il grado di accessibilità di una certa pagina o di un intero sito web, occorre eseguire una fase di validazione, attraverso strumenti automatici e revisione umana. I metodi automatici sono solitamente rapidi e convenienti, ma non riescono ad identificare tutti i problemi di accessibilità; la revisione umana può assicurare la chiarezza di linguaggio e la facilità di navigazione.

Esempi di test di validazione possono essere:

- l'uso uno strumento di accessibilità automatico e uno strumento di validazione browser;
- la validazione della sintassi (es. HTML, XML...);
- la validazione dei fogli di stile (es. CSS);
- l'uso di browser o di emulatori solo testuali (es. Lynx);
- l'uso di differenti browser grafici:
  - con suoni e grafici caricati
  - con grafici non caricati
  - con suoni non caricati
  - senza mouse
  - con frame, script, fogli di stile e Applet non caricati
- l'uso di molteplici browser, vecchi e nuovi;
- l'uso di un browser con la voce incorporata, uno screen reader, un software ingrandente, un piccolo display...;
- l'uso di controlli automatici di spelling e grammatica (una persona che legge una pagina con un sintetizzatore vocale può non esssere in grado di decifrare il miglior tentativo del sintetizzatore per una parola con un errore di spelling): eliminare problemi di grammatica migliora la comprensione;
- la revisione della chiarezza e della semplicità del documento: a tal proposito possono essere utili le statistiche di leggibilità, come quelle generate da alcuni word processor; ancor meglio sarebbe affiancare una revisione del testo da parte di un esperto editor umano;

- la collaborazione di persone con una disabilità nella revisione dei documenti; utenti disabili esperti e principianti forniscono un valido feedback sui problemi riguardanti l'accessibilità e l'usabilità.

1.6 L'adozione delle linee guida WCAG 1.0 in Italia

Almeno fino a non molto tempo fa, l'esistenza stessa del W3C in Italia era nota ad una comunità ridotta di persone direttamente coinvolte nello sviluppo delle tecnologie e non necessariamente a chi, all'interno delle diverse organizzazioni, aveva compiti di tipo operativo per la realizzazione dei siti web.

Si deve aggiungere, inoltre, che la sensibilità verso i temi di accessibilità dei contenuti digitali non ha tratto benefici dall'intensa campagna di sensibilizzazione sui problemi dell'accessibilità tradizionale, quella cioè legata alla riduzione delle innumerevoli barriere architettoniche.

Nella costruzione di documenti web prevale un approccio grafico-estetico piuttosto che funzionale; ciò è dovuto soprattutto alla realizzazione, da parte di molti professionisti del settore grafico, di contenuti web per veicolare informazioni pubblicitarie attraverso un'infrastruttura scalabile ed economica come Internet.

Anche per questi motivi, le linee guida del W3C sono tuttora sconosciute o accolte in modo diverso dal pubblico: HTML 4.01 è ben noto, WCAG 1.0 lo è un po' meno.

Attualmente vi sono 10 paesi, e tra questi l'Italia, che hanno costituito gli Uffici del W3C, il cui obiettivo è rappresentare il punto di contatto locale con il Consorzio. L'Ufficio italiano del W3C, inaugurato nel 1999, è ospitato a Pisa dal CNUCE, un istituto del CNR; nel corso degli ultimi anni, questo Ufficio è stato attivo nel promuovere le tecnologie del W3C in Italia, anche collaborando con i membri italiani del Consorzio, tra i quali la Presidenza del Consiglio dei Ministri, che a

partire dall'inizio del 2000 ha intrapreso delle importanti iniziative per promuovere l'accessibilità dei siti web della Pubblica Amministrazione.

1.7 Attività della Presidenza del Consiglio per l'accessibilità dei siti della Pubblica Amministrazione

Una volta divenuta membro effettivo del W3C, la Presidenza del Consiglio ha diretto la propria attività in tema di accessibilità su due fronti paralleli: da un lato, si è ritenuto opportuno fornire delle indicazioni ufficiali per l'accessibilità dei siti web delle amministrazioni pubbliche; concorrentemente, si è iniziato a studiare concretamente il modo in cui poteva essere reso accessibile il sito della Presidenza, che già allora costituiva un complesso sistema informativo.

La prima attività ebbe inizio nel settembre 2000 che, con un decreto dell'allora Ministro Bassanini, istituì presso il *Dipartimento della Funzione Pubblica (DFP)* un "Comitato di studio interministeriale per il miglioramento dell'accessibilità dei siti web delle pubbliche amministrazioni". Quasi contemporaneamente iniziò l'attività di un gruppo di lavoro presso l'*Aipa (Accessibilità e tecnologie Informatiche nella Pubblica Amministrazione)*. La presenza di alcune persone in entrambe le commissioni, facilitò il coordinamento e la divisione dei lavori tra il Comitato di studio del DFP e l'Aipa.

Il risultato degli sforzi fu una circolare, emessa dal Ministro Bassanini il 13 marzo 2001, che faceva esplicito riferimento alle linee guida del W3C, richiedendone il rispetto da parte delle Amministrazioni Pubbliche.

1.8 Normative italiane sull'accessibilità web

Nell'arco di pochi mesi sono stati pubblicati sulla Gazzetta Ufficiale due importanti documenti sull'accesso ad Internet da parte di utenti disabili: il primo è una circolare del Dipartimento della Funzione Pubblica che detta le linee guida per l'organizzazione, l'usabilità e l'accessibilità dei siti web delle pubbliche amministrazioni; il secondo è una circolare dell'Aipa del 14 settembre 2001 che descrive criteri e strumenti per migliorare l'accessibilità dei siti web e delle applicazioni informatiche a persone disabili.

Presidenza del Consiglio dei Ministri – DFP

Circolare 13 marzo 2001, n.3/2001 (G.U. Serie generale n.065 del 19 marzo 2001)

Il primo documento è indirizzato a chiunque abbia, all'interno delle pubbliche amministrazioni, responsabilità collegate alla progettazione, realizzazione e manutenzione di sistemi informativi basati sulle tecnologie del web. La circolare sottolinea che il web è contemporaneamente uno strumento comunicativo ed una tecnologia organizzativa, in quanto permette di lavorare insieme ad altri e di condividere informazioni tra uffici, di realizzare pratiche di integrazione tra basi di dati e tra procedure, nonché forme di collaborazione con soggetti esterni ad una determinata amministrazione. Perché questo avvenga, occorre che i siti siano usabili e accessibili.

USABILITÀ

L'usabilità di un sito implica che le informazioni siano organizzate e strutturate in maniera da garantire la massima fruibilità. Prerequisito di ogni progettazione di un sito è l'identificazione delle tipologie di pubblico al quale è rivolto; è importante quindi che vi sia uno sforzo per immaginare come i contenuti saranno reperibili ed usabili, tenendo conto della varietà delle caratteristiche personali, sociali e culturali degli utenti.

L'informazione deve essere scritta in modo chiaro e con un linguaggio comune, evitando l'utilizzo di formati commerciali; si raccomanda l'impiego della tecnologia più semplice per risolvere un

determinato problema, e la scelta di tecnologie compatibili e rispettose degli standard Internet.

I siti devono essere coerenti nella loro organizzazione e nella presentazione delle informazioni di cui dispongono, ponendo in essere soluzioni per facilitare la ricerca di argomenti attraverso strumenti di ricerca semantica, e la predisposizione di una mappa del sito corredata di tutti i relativi collegamenti ipertestuali aggiornati, nonché di una guida che chiarisca le principali difficoltà riscontrabili dai navigatori. Le informazioni, inoltre, devono essere inserite in una struttura che evolva senza imporre frequenti spostamenti o cancellazioni, garantendo per quanto possibile l'omogeneità di tale struttura in tutte le parti del sito. È necessario, pertanto, che sia garantito il funzionamento dei link ipertestuali, e quindi la reperibilità delle informazioni, anche a distanza di tempo.

ACCESSIBILITÀ

L'accessibilità di un sito esige che esso sia progettato in modo da garantire la sua consultazione anche da parte di persone con disabilità fisiche o sensoriali, o condizionate dall'uso di strumenti con prestazioni limitate o da situazioni ambientali sfavorevoli.

Autorità per l'informatica nella Pubblica Amministrazione

Circolare 6 settembre 2001, n. Aipa/CR/32 (G.U. Serie generale n. 214 del 14 settembre 2001)

La circolare dell'Aipa si rifà al precedente documento e specifica i criteri da rispettare nella progettazione e manutenzione dei sistemi informatici pubblici, per favorire l'accessibilità ai siti web, che mettono a disposizione di cittadini e imprese informazioni e servizi interattivi mediante tecnologie e protocolli Internet, e per rendere accessibili le applicazioni informatiche utilizzate dal personale della Pubblica Amministrazione, dai cittadini e dalle imprese.

In questo documento si afferma che il grado più elevato di accessibilità si consegue attuando il principio della progettazione universale, secondo il quale ogni attività di progettazione deve tenere conto della varietà di esigenze di tutti i potenziali utenti. Questo principio, applicato ai sistemi informatici, si traduce nella progettazione

di sistemi, prodotti e servizi fruibili da ogni utente, direttamente o in combinazione con tecnologie assistive.

L'elemento architetturale di un sistema informatico che viene maggiormente interessato dal problema dell'accessibilità è l'interfaccia utente: in pratica viene scartata la possibilità che una certa organizzazione realizzi due siti diversi, uno di serie A per le persone cosiddette 'normali', e uno di serie B per gli utenti disabili; il sito deve essere uno solo, ma le interfacce utente possono avere caratteristiche diverse, in funzione del tipo di utente.

In linea generale, il requisito di accessibilità sarà tanto più facilmente soddisfatto quanto più la progettazione sarà basata sulla separazione dei contenuti dalle modalità di presentazione.

La circolare stabilisce che un'applicazione informatica dotata di interfaccia utente viene considerata accessibile se, con l'eventuale ausilio di tecnologie assistive, non presenta difficoltà di:

- lettura del contenuto di tutte le finestre visualizzabili sullo schermo;
- controllo dell'inserimento dei dati;
- controllo dell'interazione con elementi o oggetti dell'interfaccia, quando tali operazioni vengano eseguite da persone sufficientemente esperte nell'uso di una postazione di lavoro, con una configurazione dotata, a seconda dei casi, di strumenti di tecnologia assistiva.

I principali strumenti di tecnologia assistiva attualemente in uso sono: lettori di schermo (con sintesi vocale o display braille), funzioni di ausilio per ipovedenti e disabili motori, applicativo specifico di ingrandimento dello schermo, sistema di input vocale (con dettatura di testo e emulazione di comandi di tastiera e/o mouse).

2. USABILITA` NELLE INTERFACCE WEB [17] [18]

2.1 Gli standard internazionali per la definizione di usabilità

Esistono centinaia di definizioni di usabilità formulate nelle norme *ISO (International Organization for Standardization)*, che rappresentano un riferimento autorevole per scegliere le procedure da seguire nella progettazione centrata sull'utente.

La normativa è orientata da un lato a valutare l'usabilità in termini di prestazioni e di soddisfazione dell'utente e, dall'altro, a specificare i requisiti dell'interfaccia.

ISO/IEC 9126

È lo standard di riferimento per la qualità dei prodotti software; per orientare il processo di pianificazione, questa norma definisce i fattori di qualità di un prodotto software: usabilità, funzionalità, affidabilità, portabilità, mantenibilità ed efficienza.

ISO 9241

Rappresenta lo standard di riferimento per l'usabilità; in questa norma l'usabilità è descritta come il grado entro il quale un prodotto può essere utilizzato da utenti specifici per raggiungere certi obiettivi con efficacia, efficienza e soddisfazione, in un determinato contesto d'uso.

2.2 User Center Design: l'utente al centro della progettazione

Progettare un'interfaccia prevede una varietà di scelte progettuali, che coinvolgono gli utenti e le attività che possono essere condotte tramite il supporto delle tecnologie informatiche.

Dagli anni Ottanta in poi, nello sviluppo di modelli di progettazione basati sull'usabilità, l'utente è stato progressivamente messo al centro del processo, elaborando quello che oggi è definito *User Centered Design*.

Questo metodo individua una direzione di sviluppo invertita rispetto a quella adottata dagli approcci ingegneristici, e coinvolge gli utenti sin dalle prime fasi del processo di elaborazione e realizzazione.

La progettazione orientata all'utente è intesa come un approccio metodologico che permette di giungere alla realizzazione di sistemi facili da apprendere e da usare, ovvero di un prodotto conforme alle abitudini cognitive e ai modelli mentali degli utenti ai quali il prodotto stesso è destinato.

2.3 Usabilità e Contextual Design

Se l'approccio ingegneristico alla questione usabilità presuppone che i problemi possano essere risolti avvalendosi di una serie di linee guida (*euristiche*) stabilite a priori, la realtà della progettazione risente della variabilità dei contesti d'uso, dei bisogni e degli scopi degli utenti.

Mentre si va diffondendo l'idea della centralità dell'utente nella progettazione delle interfacce, l'approccio del design contestuale riconosce l'importanza delle capacità e dei vincoli fisici e cognitivi dei singoli utenti, nonché delle relazioni culturali, sociali e organizzative. Ciò significa che l'utente non è considerato separato dal contesto nel quale opera, e che tale contesto influenza il modo stesso di utilizzare le tecnologie.

2.4 Human-Computer Interaction (HCI)

Tra le metodologie che possono dare un apporto significativo allo studio dell'usabilità, la HCI si occupa di quegli aspetti che affrontano la produzione di interfacce web, in termini di progettazione, valutazione e implementazione di sistemi interattivi per l'uso.

Progettare interfacce interattive è un'attività sperimentale per la quale sono state concepite diverse metodologie, ed è con ogni probabilità la parte di lavoro più difficile nel processo di sviluppo di prodotti multimediali, poiché prevede l'interazione dell'uomo con la tecnologia informatica.

Per quanto concerne i siti web, all'interno del settore di sviluppo delle interfacce uomo-macchina prevalgono gli aspetti del layout grafico, dei contenuti testuali e ipertestuali, e la loro strutturazione in termini di navigazione e accessibilità.

L'usabilità di questi prodotti si pone pertanto come il fattore critico del loro successo e il maggiore indice di qualità.

2.5 Come si realizzano siti usabili: le euristiche di Jakob Nielsen [19]

Al processo di progettazione e di sviluppo dell'usabilità contribuiscono le metodologie e gli approcci definiti dallo *User Centered Design*, dal *Contextual Design*, dalla *Human-Computer Interaction*. Tuttavia il concetto di usabilità lascia ampia scelta per quel che riguarda gli aspetti concreti relativi al processo di sviluppo di un prodotto usabile.

Dal momento che non esistono parametri universalmente riconosciuti per la progettazione di interfacce web, l'usabilità si avvale di una serie di principi, definiti euristiche, che offrono indicazioni generali utili ai realizzatori durante la fase di progettazione. Questo metodo, puramente empirico, si basa su una sintesi dei problemi più frequenti di usabilità: il primo ad averlo formalizzato è stato Jakob Nielsen nei primi anni Novanta.

In particolare, dall'analisi di 249 problemi riscontrati in studi di vario tipo, Nielsen ha ricavato per via statistica dieci regole principali:

1. Visibilità dello stato istantaneo del sistema

L'utente dovrebbe avere sempre la percezione di ciò che sta facendo; affinché questo si verifichi, è necessario fornirgli un adeguato feedback che chiarisca la corrispondenza tra le azioni che può compiere e gli effetti che ne conseguono.

2. Corrispondenza tra il sistema e il mondo reale

Il linguaggio utilizzato deve essere semplice e di immediata comprensione; a questo scopo, è meglio evitare tecnicismi e termini stranieri, strutturando i contenuti in maniera logica, chiara e funzionale.

3. Libertà e possibilità di controllo del sistema da parte dell'utente

L'utente deve essere messo in grado di rimediare ai suoi errori attraverso evidenti 'uscite di emergenza' e funzioni che evitino percorsi complicati e snervanti.

4. Coerenza interna ed esterna e conformità agli standard comunemente accettati dagli sviluppatori web

È opportuno usare le convenzioni generalmente acquisite e mantenerle inalterate per evitare di confondere gli utenti.

5. Cura nella prevenzione degli errori

Oltre a prevedere eventuali messaggi d'errore, è opportuno cercare di limitare il più possibile gli errori dell'utente.

6. Riconoscere piuttosto che ricordare

Per ridurre lo sforzo mnemonico, l'interfaccia dovrebbe fornire le istruzioni per un suo corretto utilizzo e rendere visibili gli strumenti e le opzioni disponibili.

7. Promuovere la flessibilità e l'efficienza

È consigliabile prevedere diversi livelli di utilizzo del sistema, supportando una navigazione di tipo gerarchico per i navigatori meno esperti ma inserendo opportune 'scorciatoie' per gli utenti abituali.

8. Grafica e design minimalisti

È opportuno evitare di inserire informazioni irrilevanti poiché ogni dato superfluo toglie visibilità a ciò che invece è effettivamente importante, distraendo l'utente ed appesantendo l'intero documento.

9. Fornire all'utente i mezzi per riconoscere e riparare gli errori

I messaggi di errore dovrebbero essere espressi in un linguaggio chiaro e non attraverso codici, indicando precisamente in cosa consiste il problema e come fare per risolverlo.

10. Inserire strumenti di aiuto e istruzioni di utilizzo

Anche se il sistema dovrebbe essere usabile senza documentazione, potrebbe essere necessario inserire strumenti di aiuto e istruzioni: in questo caso ogni informazione dovrebbe essere facilmente rintracciabile, focalizzata sul compito dell'utente e strutturata in più passaggi.

2.6 Misurazione del grado di usabilità

Non esiste un solo metodo per valutare il grado di usabilità dei siti web: esistono piuttosto sistemi, tecniche e approcci molto diversi tra loro ed in continua evoluzione.

Alcuni, definiti *metodi analitici* (cognitive walkthrough, simulazioni d'uso, analisi euristiche...), tentano di prevedere il tipo di problemi che incontrerà l'utente, pur senza coinvolgerlo direttamente.

Altri, definiti *metodi empirici* (interviste, questionari, esperimenti...), si basano sull'osservazione del comportamento degli utenti mentre interagiscono con l'interfaccia.

Con entrambi i metodi, l'analisi può essere effettuata a più livelli e con strumenti diversi.

Scegliere il metodo di indagine più appropriato significa in primo luogo confrontarsi con questi metodi di valutazione, che non sono sicuramente in grado di identificare la totalità dei difetti di un sito. Inoltre è indispensabile confrontarsi con la continua evoluzione del concetto e delle implicazioni connesse al termine usabilità, oltre che con la complessità che deriva dalla progettazione di prodotti che hanno scopi e destinatari spesso diversi.

3. TECNOLOGIE IMPIEGATE NEL SISTEMA INFEA

Lo studio del sistema INFEA ha evidenziato l'impiego di tecnologie per l'implementazione delle diverse componenti; prima dell'illustrazione dell'architettura portante e del ruolo specifico che esse rivestono all'interno di tale sistema, si descrivono di seguito sommariamente i loro specifici principi teorici.

In particolare, questo si prendono in considerazione i seguenti supporti tecnologici: HTML; Servlet Java; XML, XSLT, XHTML e XPath.

3.1 HTML [5] [6]

Per comprendere HTML bisogna prima comprendere le necessità alle quali risponde e il mondo nel quale opera.

HTML (HyperText Markup Language) è un linguaggio formato testo che usa semplici tag per supportare uno dei più eccitanti ambienti informativi: il World Wide Web.

Il web nasce nel 1989 per opera di Tim Berners-Lee ed alcuni suoi colleghi del CERN (Conseil Européen pour la Recherche Nucléaire) come sistema per la condivisione e lo scambio di risultati scientifici tra fisici.

Per vedere le pagine web come le conosciamo oggi, si è dovuto attendere fino al 1992/1993, anno in cui venne creato il primo browser – Mosaic – che diede definitivamente inizio al fenomeno secondo il quale adesso web è, erroneamente, sinonimo di Internet.

Il linguaggio HTML è un linguaggio ipertestuale che descrive la struttura relativa al contenuto di un documento web, oltre ad alcune caratteristiche di tipo comportamentale. HTML è stato anch'esso definito mediante un linguaggio, cosiddetto di markup, ancora più complesso, ovvero *SGML (Standard Generalized Markup Language)*; per la precisione, HTML è definito da un particolare tipo di documento, chiamato *Document Type Definition (DTD)* nell'ambito SGML: di conseguenza, qualsiasi documento HTML è anche un documento

SGML che rappresenta uno specifico sottoinsieme delle capacità proprie di SGML.

HTML costituisce un modo per rappresentare il testo e collegarlo ad altre risorse (compreso audio, video e altri tipi di file), permettendo la visualizzazione simuntanea di diversi tipi di dati.

Il codice HTML che viene fornito al client (browser) dal server web non è altro che un semplice file che include sostanzialmente due tipi di testo: il contenuto, che rappresenta il testo o altre informazioni da visualizzare o riprodurre mediante lo schermo, le casse o altri dispositivi del client; i tag, che sono testo o informazioni per controllare la visualizzazione o per riferirsi ad altre risorse anch'esse da visualizzare o riprodurre. I file HTML includono sia le informazioni di controllo (tag) sia il contenuto (testo) della pagina web.

Lo stesso acronimo HTML riflette due concetti importanti del suo funzionamento e quindi del suo successo: il concetto di ipertesto, inteso come metodo per la creazione di documenti multimediali, ma anche come sistema per fornire collegamenti all'interno di uno stesso documento o tra più documenti; il concetto di linguaggio di markup, che rappresenta un sistema per incorporare dei tag speciali, definiti proprio dallo standard HTML, che descrivono la struttura e il comportamento del documento.

La semplicità e allo stesso tempo la potenza dei tag HTML permette a chiunque di realizzare pagine web per uso pubblico o privato; allo stesso modo, la potenza dell'ipertesto, grazie al supporto incorporato per il multimedia e per i collegamenti verso altre risorse remote, ha dato luogo alla scalabilità del web.

La creazione di documenti web è facile e lineare: è sufficiente, come per ogni cosa, giocare secondo le regole.

I caratteri di controllo speciali che separano ogni marcatore HTML dal testo vero e proprio sono le parentesi angolari, < e > rispettivamente; questi caratteri segnalano al parser presente nel software del browser come visualizzare le informazioni che legge. In altre parole, il parser esamina i caratteri presenti all'interno di un file HTML e li suddivide tra marcatori e testo puro; nel primo caso, a

seconda del particolare marcatore trovato, indicherà al browser le azioni da compiere.

Un esempio formale di come appare un tag all'interno di un file HTML è il seguente:

<NOME TAG {ATTRIBUTO1 = ["VALORE"]...ATTRIBUTOk = "VALORE"]}>
Testo Testo ... Testo Testo
{</NOME TAG>}

<NOME TAG> rappresenta il nome del particolare tag HTML, come ad esempio H1 per l'intestazione di primo livello oppure OL per un elenco numerato; tutti i tag definiti da HTML hanno un nome.

Alcuni tag HTML richiedono o consentono la presenza di attributi ad essi associati e questo viene espresso dalla notazione tramite le parentesi graffe, come nel caso {ATTRIBUTO1 = ["VALORE"]...ATTRIBUTOk = "VALORE"]}; inoltre un certo attributo può o meno richiedere uno specifico valore, di qui l'uso delle parentesi quadre.

Il Testo...Testo è il contenuto sul quale opera il tag; se, ad esempio, il tag fosse: <TITLE>Dipartimento di Informatica</TITLE>, il testo visualizzato sulla barra del browser sarebbe proprio "Dipartimento di Informatica".

{</NOME TAG>} rappresenta il tag di chiusura; le parentesi graffe indicano che questo elemento può non essere presente, ma per il fatto che il 70% dei tag HTML richiede un tag di chiusura oltre ad uno di apertura, l'omissione di questo elemento risulta un'eccezione più che una regola.

La struttura tipica di una pagina web scritta secondo il formalismo HTML è così costituita:

<HTML>
<HEAD>
... Intestazione della pagina ...
</HEAD>
<BODY>
... Corpo della pagina ...
</BODY>
</HTML>

L'INTESTAZIONE (l'elemento <HEAD>)

Nell'intestazione vengono inserite delle informazioni riguardo al documento che, a parte il titolo, non vengono visualizzate ma sono comunque utilizzate dai browser e dai motori di ricerca.

Per specificare il titolo della pagina, è sufficiente usare

<TITLE> Titolo della Pagina </TITLE>,

mentre per specificare le parole chiave ed il contenuto del documento web si usano i tag META:

<META NAME= "KEYWORDS" CONTENT= "parole chiave della pagina">

<META NAME= "DESCRIPTION" CONTENT= "parole chiave della pagina">

Il CORPO (<BODY>)

Questa porzione del documento è riservata alla descrizione del layout e alla struttura del documento mediante una vasta gamma di tag per le intestazioni, i paragrafi di testo, gli elenchi, la gestione di elementi grafici quali immagini, suoni, animazioni...

Il corpo di una pagina contiene tutte le parti normalmente visualizzate all'interno della finestra del browser.

La versione attuale di HTML è la 4.01; le raccomandazioni dello standard HTML si possono trovare sul sito web del W3C.

3.2 SERVLET JAVA [7] [8] [9] [10]

La crescente richiesta di pagine web con contenuti dinamici e di nuovi servizi di rete ha guidato lo sviluppo di una particolare tipologia di software, che collabora con il server web per estenderne le funzionalità e poter interagire con i database.

Le tecniche per realizzare questo tipo di software sono molte; le più conosciute sono sicuramente le estensioni *CGI (Common Gateway Interface)*, realizzate principalmente in C o in Perl. Anche Java mette a disposizione una API - le Servlet - che consente di sviluppare applicazioni lato server.

3.2.1 Che cos'è una Servlet

Le *Servlet* sono moduli software scritti in Java che vengono eseguiti in applicazioni lato server per esaudire le richieste dei client. In particolare le Servlet costituiscono per i server web ciò che le Applet costituiscono per i browser; a differenza delle Applet, tuttavia, sono prive di un'interfaccia grafica per interagire con l'utente.

Esse non sono legate ad un particolare protocollo per la comunicazione tra client e server, anche se comunemente si utilizza il protocollo HTTP, parlando così di *HTTP Servlet*.

Per scrivere una Servlet si fa uso delle classi contenute nel package `javax.servlet`, che è il framework di base per la scrittura delle Servlet, e nel package `javax.servlet.http`, che è l'estensione del framework di base per la comunicazione via HTTP. Utilizzando un linguaggio portabile come Java, le Servlet consentono di realizzare soluzioni per estendere le funzionalità dei server web, indipendentemente dal sistema operativo su cui vengono eseguite.

3.2.2 Impiego delle Servlet

Le Servlet vengono principalmente usate per i seguenti scopi:

- elaborare e salvare dati provenienti da form HTML;
- provvedere alla creazione di pagine HTML dinamiche, ad esempio attraverso l'uso di dati contenuti in un database, in seguito alla richiesta di un client;
- gestire informazioni con stato, come ad esempio la gestione di un sistema di commercio elettronico, in cui un certo numero di clienti effettua degli acquisti concorrentemente e dove occorre mantenere in modo corretto le informazioni sui clienti e sui loro acquisti (il classico 'carrello della spesa').

3.2.3 Servlet vs. CGI

Le Servlet presentano alcuni vantaggi rispetto al tradizionale CGI.

Efficienza

Con una tradizionale applicazione CGI, viene generato un processo – istanza del programma o *script CGI* – per ogni richiesta che arriva al server; questo tipo di operazione può risultare eccessivamente dispendioso in termini di risorse impiegate.

Con le Servlet, invece, la *JVM (Java Virtual Machine)* genera, per ogni richiesta da esaudire, un *Thread Java*, molto più leggero rispetto ad un processo generato dal sistema operativo. Tutto ciò si traduce nel fatto che se vi sono ***n*** richieste contemporanee, con il CGI esistono ***n*** immagini in memoria del programma CGI; con le Servlet, invece, ci sono ***n*** Thread ma un'unica copia della classe Servlet in memoria.

C'è da dire inoltre che l'efficienza è ancora superiore dal momento che un'applicazione CGI che si interfaccia con un database si connette ad esso ad ogni richiesta che le arriva e vi si disconnette al termine. Con le Servlet è possibile mantenere aperte una o più connessioni al database, utilizzando ad esempio una Connection Pool, e riutilizzarle da richiesta a richiesta.

Facilità d'uso

Java mette a disposizione una API per la scrittura delle Servlet, che facilita molto le varie operazioni coinvolte nel loro uso; in particolare queste operazioni sono: la manipolazione dei dati provenienti dai form HTML, la lettura e gestione degli headers delle richieste HTTP, la gestione dei cookie e delle sessioni...

Potenza

Le Servlet consentono di realizzare molte funzionalità che con le applicazioni CGI sono difficili o addirittura irrealizzabili. Le Servlet possono dialogare direttamente con il server web migliorando l'uso e la condivisione dei dati (testo, immagini...), facilitando le operazioni come la session tracking, permettendo il mantenimento di informazioni tra una richiesta ed un'altra, ed infine offrendo un meccanismo di caching delle operazioni già eseguite durante una richiesta precedente.

Portabilità

Essendo scritte in Java, le Servlet possono girare su qualsiasi piattaforma; oggi, infatti, le Servlet sono supportate, direttamente o mediante plug-in (Servlet Engines), dalla maggior parte dei server web, sia commerciali che free.

Convenienza

Molti server web che supportano le Servlet sono free o hanno un basso costo; se si dispone di un server web commerciale (ad alto costo!) che non supporta le Servlet, è sufficiente un aggiornamento, normalmente gratuito, tramite appositi plug-in di supporto.

3.2.4 Architettura (di base) e ciclo di vita di una Servlet

Nella sua forma più generale, una Servlet è un'istanza della classe che implementa l'interfaccia `javax.servlet.Servlet`, specificatamente `javax.servlet.GenericServlet`; come già detto, però, quella più utilizzata è quella che si basa sul protocollo HTTP, costruita estendendo la classe `javax.servlet.http.HttpServlet`.

All'avvio, il server carica in memoria la classe Servlet ed eventualmente le ulteriori classi utilizzate; successivamente ne crea un'istanza invocando il costruttore senza argomenti. Il passo immediatamente successivo è quello di chiamare il metodo **`init`**`(ServletConfig config)`, tramite il quale avvengono le inizializzazioni delle variabili globali (procedura eseguita solo una volta durante l'intero ciclo di vita della Servlet) e viene caricato l'oggetto di tipo `ServletConfig` che potrà essere eventualmente recuperato mediante il metodo **`getServletConfig`**`()`. Quest'ultima operazione viene eseguita nel metodo `init` della classe `GenericServlet`, ed è per questo che in ogni classe che la estende, e tale è `HttpServlet`, all'inizio del metodo `init` c'è una chiamata al metodo `init` della superclasse, cioè `super.`**`init`**`(config)`. L'oggetto `ServletConfig` contiene i parametri della Servlet ed il riferimento a `ServletContext` che rappresenta il contesto su cui gira la Servlet.

Una volta inizializzata la Servlet, per ogni richiesta proveniente dal client, viene invocato il seguente metodo: **`service`**`(ServletRequest req, ServletResponse res)`; questo metodo viene chiamato in modo concorrente, ovvero più Thread possono invocarlo contemporaneamente.

In casi particolari, e cioè quando si lavora con risorse non condivisibili, è possibile implementare Servlet non concorrenti.

Quando una Servlet deve essere arrestata o il server necessita di essere riavviato, viene chiamato il metodo **`destroy`**`()` nel quale vengono rilasciate tutte le risorse allocate nel metodo `init`; anche il metodo `destroy()` può essere chiamato una sola volta durante tutto il ciclo di vita della Servlet.

3.3 XML [11] [12] [13]

L'*eXtensible Markup Language (XML)* è un metalinguaggio che permette di creare dei linguaggi personalizzati di markup; nasce dall'esigenza di portare nel web SGML, lo standard internazionale per la descrizione della struttura e del contenuto di documenti elettronici di qualsiasi tipo; ne contiene quindi tutta la potenza, ma non tutte le complesse funzioni raramente utilizzate.

Attualmente XML è il linguaggio più promettente per archiviare e distribuire le informazioni sul web. È stato definito da *XML Working Group* del W3C, che lo identifica così: "EXtensible Markup Language (XML) è un sottoinsieme di SGML... Il suo obiettivo è quello di consentire ad un generico documento SGML di essere trattato, ricevuto e processato sul web nello stesso modo in cui succede attualmente con HTML. XML è stato disegnato per la facilità di implementazione e per l'interoperabilità sia con SGML che con HTML."

Nonostante HTML sia il linguaggio più comune per la creazione di pagine web, presenta dei grossi limiti per quel che riguarda la capacità di memorizzazione delle informazioni. L'utilizzo di XML permette di superare questi limiti legati alla dipendenza da un tipo di documento HTML, singolo e non estensibile.

HTML è nato per permettere agli utenti del web di condividere le informazioni su sistemi differenti; il presupposto era che queste informazioni fossero sostanzialmente testo, con al più alcune immagini e collegamenti ipertestuali. Attualmente però le informazioni sul web sono database di testo, immagini, suoni, video, audio... Quindi HTML è stato chiamato sempre più di frequente a fornire soluzioni a problemi che non aveva lo scopo di risolvere, come dover descrivere tipi differenti e specifici di informazioni, definire relazioni complesse di collegamenti fra documenti, trasmettere informazioni in diversi formati. Per superare questi problemi, sono state create delle estensioni dell'HTML, spesso fra loro incompatibili. D'altro lato, XML ha una sintassi molto flessibile che gli permette di essere impiegato per descrivere virtualmente ogni tipo di informazione, da una semplice ricetta gastronomica ad un complesso database.

Un documento XML, affiancato da un opportuno foglio di stile o da una comune pagina HTML, può essere facilmente visualizzato da un web browser. Dato che un documento XML definisce e struttura effettivamente le informazioni che contiene, il browser può trovare, estrarre, ordinare, filtrare, combinare e modificare tali informazioni in diversi modi.

XML, quindi, propone una soluzione ideale per il trattamento della rapida espansione in termini di quantità e complessità delle informazioni da pubblicare sul web.

3.3.1 La necessità di XML

HTML mette a disposizione un insieme ben preciso e limitato di elementi predefiniti, che possono essere usati per identificare le componenti di un tipico documento web; alcuni esempi di elementi sono intestazioni, paragrafi, liste, tabelle, immagini e collegamenti ipertestuali.

Malgrado l'insieme di elementi predefiniti dallo standard HTML sia notevolmente aumentato dalla prima versione, questo linguaggio è ancora inadatto per la definizione di molti tipi di documenti.

Per i documenti che non consistano di componenti tipiche quali intestazioni, paragrafi, liste, tabelle e così via, HTML non mette a disposizione elementi necessari per contrassegnare uno spartito musicale o un'insieme di equazioni matematiche.

Per quel che riguarda i database, come ad esempio l'inventario di una biblioteca, è possibile usare una pagina HTML per archiviare e mostrare informazioni attraverso un database statico (come una lista che descriva il contenuto di alcuni libri). Tuttavia, se si vuole ordinare, filtrare, ricercare e operare con queste informazioni in un altro modo, ogni singola parte di informazione deve essere etichettata e HTML non dispone degli elementi necessari.

Infine, supponiamo per esempio di voler scrivere un libro e di volerlo contrassegnare in parti, capitoli, sezioni..., secondo una struttura gerarchica ad albero: un programma potrebbe usare questo documento già strutturato per produrre una tabella dei contenuti, uno schema con

vari livelli di dettaglio, o per estrarre sezioni specifiche. Un elemento di intestazione HTML, tuttavia, contrassegna soltanto il testo che rappresenta la stessa intestazione; dato che non è possibile annidare un elemento di intestazione ed il testo che contiene in un altro elemento di intestazione, questi elementi non possono indicare in modo chiaro la struttura gerarchica del documento.

La soluzione a queste limitazioni è rappresentata proprio da XML.

3.3.2 La soluzione XML

Quando viene creato un documento XML, anziché usare un insieme limitato di elementi predefiniti, vengono costruiti elementi ad hoc, ai quali è assegnato un nome qualsiasi; in questo modo è virtualmente possibile descrivere qualsiasi tipo di documento, da uno spartito musicale ad un database. Ad esempio, per descrivere l'inventario di una biblioteca è possibile utilizzare il seguente documento XML:

```
<? xml version="1.0">                                   dichiarazione XML

<!-- File Name: Inventario.xml -->                      commento

    <INVENTARIO>
        <LIBRO>
            <TITOLO>I Malavoglia</TITLE>
            <AUTORE>Giovanni Verga</AUTORE>
            <N_PAGINE>213</N_PAGINE>
        </LIBRO>
        <LIBRO>
            <TITOLO>Il fu Mattia Pascal</TITLE>
            <AUTORE>Luigi Pirandello</AUTORE>
            <N_PAGINE>191</N_PAGINE>
        </LIBRO>
        ...
        ...
    ...
```

```
</INVENTARIO>
```

È importante notare che il nome degli elementi in un documento XML (come INVENTARIO, LIBRO, AUTORE... nell'esempio sopra) non fanno parte della definizione di XML; al contrario, è possibile scegliere arbitrariamente i nomi degli elementi al momento della creazione di ogni singolo documento.

Dall'esempio precedente emerge che un documento XML è strutturato secondo una gerarchia ad albero, con gli elementi completamente annidati all'interno di altri elementi e con un unico elemento che rappresenta il livello più alto della gerarchia (nell'esempio l'elemento INVENTARIO), noto come elemento documento o elemento radice, che contiene tutti gli altri elementi.

3.3.3 Scrivere documenti XML

Dato che XML non prevede elementi predefiniti, sembrerebbe logico ipotizzare che sia uno standard più o meno casuale; nonostante ciò un documento deve seguire delle regole sintattiche ben precise. Ad esempio, a differenza di quanto accade per HTML, ogni elemento XML deve avere sia un tag di inizio che un tag di fine, e qualsiasi elemento annidato deve essere completamente racchiuso dall'elemento che lo contiene.

Infatti, la grande flessibilità con cui è possibile creare i propri elementi esige una sintassi piuttosto rigorosa. Questo perché la natura dei documenti XML richiede l'uso di programmi per trattare e visualizzare le informazioni che tali documenti contengono; la sintassi rigorosa conferisce ai documenti una forma predicibile e facilita la scrittura di questi programmi.

Un documento XML può essere conforme a uno dei due diversi livelli di precisione sintattica a seconda di quale grado dello standard raggiunge: documento ben formato oppure documento valido.

3.3.4 Visualizzazione di documenti XML

Se per un browser è molto facile riconoscere e trattare in modo adeguato elementi HTML poiché predefiniti e standard, come può uno stesso browser o qualsiasi altro programma visualizzare gli elementi che vengono completamente inventati dallo sviluppatore di un documento XML?

Ci sono tre modi fondamentali per indicare ad un browser come gestire e mostrare ogni elemento di un documento XML:

Collegamento con un foglio di stile

Con questa tecnica, viene collegato un foglio di stile al documento XML. Un foglio di stile è un file separato che contiene le istruzioni necessarie per la formattazione di ogni elemento del documento XML; è possibile utilizzare sia un foglio di stile a cascata *(Cascading Style Sheet o CSS)*, che viene impiegato per le pagine HTML, sia un file XSL (eXtensible Stylesheet Language), che è considerevolmente più potente rispetto ad un CSS e disegnato specificatamente per i documenti XML.

Data binding

Questa opzione prevede la creazione di una pagina HTML, il collegamento tra il documento XML e la pagina HTML e il collegamento tra gli elementi standard di HTML presenti nella pagina, come ad esempio SPAN o TABLE, e gli elementi XML.

Gli elementi HTML mostreranno così automaticamente l'informazione degli elementi XML a cui sono collegati.

Scripting

Quest'ultima tecnica richiede la creazione di un documento HTML, il collegamento del file XML a questa pagina ed infine l'accesso e la visualizzazione dei singoli elementi XML tramite codice script (JavaScript o Microsoft Visual Basic Scripting Edition [VBScript]). Il browser mostra il documento XML sotto forma di *Document Object Model (DOM)*, che dispone di un vasto insieme di oggetti, proprietà e

metodi che il codice script può usare per accedere, modificare e visualizzare gli elementi XML.

3.3.5 SGML, HTML e XML a confronto

SGML rappresenta il capostipite di tutti i linguaggi di markup. Sia HTML che XML, anche se con modalità distinte, derivano da SGML; SGML definisce una sintassi di base, ma consente allo sviluppatore di creare i propri elementi. Per poter usare SGML nella descrizione di un particolare documento, è necessario creare la struttura ed un insieme appropriato di elementi; ad esempio, per descrivere un libro, è possibile usare elementi chiamati LIBRO, PARTE, CAPITOLO, INTRODUZIONE, SEZIONE-A, SEZIONE-B, SEZIONE-C...

Un insieme generale di elementi usati per descrivere un particolare tipo di documento viene definito applicazione SGML, e comprende anche le regole che specificano il modo in cui gli elementi possono essere sistemati. È possibile definire la propria applicazione SGML per descrivere una specifica tipologia di documento, ovvero una struttura standard può definire un'applicazione SGML per descrivere un tipo di documento ampiamente usato. L'esempio più famoso di quest'ultimo scopo è rappresentato da HTML, che è un'applicazione SGML sviluppata nel 1991 per descrivere le pagine web.

SGML potrebbe sembrare un linguaggio estensibile adatto a definire i documenti web; tuttavia i membri del W3C hanno ritenuto SGML troppo complicato e pesante per distribuire in modo efficiente le informazioni sul web. La flessibilità e l'eccesso di caratteristiche messe a disposizione da SGML renderebbero difficoltosa la stesura del software necessario al trattamento e alla visualizzazione di un documento SGML su un web browser.

La soluzione ottimale a questo problema è quella di costruire un sottoinsieme più snello di SGML, progettato appositamente per la divulgazione delle informazioni via web; nel 1996, XML Working Group del W3C ha sviluppato questo particolare sottoinsieme che ha preso il nome di eXtensible Markup Language.

XML rppresenta quindi una versione semplificata di SGML, ottimizzata per il web; analogamente a SGML, XML permette di ideare un particolare insieme di elementi per ogni singolo documento creato.

La sintassi di XML offre un minor numero di opzioni rispetto a SGML, rendendo più facile la lettura e la comprensione dei documenti e la produzione di software come browser, script...

3.3.6 XML si appresta a sostituire HTML?

Attualmente la risposta a questa domanda è no: HTML resta tuttora il principale linguaggio per comunicare ai browser come mostrare le informazioni sul web.

Quindi, più che sostituire HTML, XML viene soprattutto usato a fianco di HTML, fornendo così un'estensione notevole alla capacità delle pagine web di: pubblicare (virtualmente) qualsiasi tipo di documento; ordinare, filtrare, sistemare, cercare e modificare le informazioni; presentare informazioni in modo altamente strutturato.

3.3.7 Applicazioni standard XML

Un generico insieme di elementi e la corrispondente struttura che costituisce un documento è detta applicazione XML o *vocabolario XML*.

Un'applicazione XML, normalmente, è corredata di una definizione del tipo di documento che si vuol realizzare – document type definition o DTD –, ovvero da una componente opzionale dell'intero documento XML. Una DTD è simile allo schema di una base di dati: definisce e nomina gli elementi che possono essere utilizzati all'interno del documento, l'ordine con il quale questi devono apparire, gli attributi associati ad ogni elemento e altre caratteristiche del documento.

Per usare una particolare applicazione XML, spesso si include una DTD all'interno del documento XML; in questo modo si restringe notevolmente il numero degli elementi e la struttura da realizzare, forzando l'intero documento ad adeguarsi allo standard dell'applicazione XML.

I vantaggi promossi dall'utilizzo di un'applicazione standard XML per lo sviluppo dei documenti, sono essenzialmente due: la possibilità di condividere il documento con altri utenti dell'applicazione, e la capacità di trattare un documento attraverso l'ausilio di software sviluppato ad hoc per quella certa applicazione.

3.3.8 Applicazioni XML per migliorare i documenti XML

Oltre che per descrivere categorie specifiche di documenti, molte applicazioni XML sono state progettate per essere inserite all'interno di qualsiasi tipo di documento XML; queste applicazioni, oltre a rendere più semplice la creazione, permettono di apportare miglioramenti significativi al documento. Alcuni esempi di applicazioni sono:

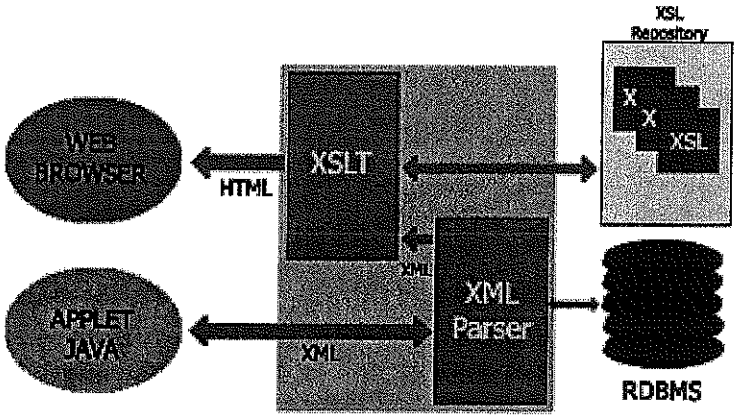
- *EXTENSIBLE STYLESHEET LANGUAGE (XSL)* → permette di realizzare fogli di stile molto potenti, utilizzando la sintassi XML;
- XML SCHEMA → consente di scrivere schemi dettagliati per un documento, utilizzando la sintassi di XML: rappresenta una valida alternativa alla creazione di DTD;
- XML LINKING LANGUAGE (Xlink) → realizza un collegamento tra diversi documenti XML, consente la creazione di link multipli e risulta

considerevolmente più efficace del meccanismo di collegamento proprio dello standard HTML;

- XML POINTER LANGUAGE (XPointer) → permette di definire collegamenti molto flessibili; è possibile utilizzare XPointer insieme a Xlink per costruire collegamenti in qualsiasi punto del documento bersaglio.

3.4 XSL E XSLT [11] [12]

Un requisito essenziale nella realizzazione di WIS consiste nella separazione tra la gestione del contenuto dei documenti web e la loro presentazione, garantendo in questo modo presentazioni diverse delle stesse informazioni a seconda della provenienza della richiesta.



- FIGURA 2.1 -

Per implementare questa specifica (figura 2.1), il sistema INFEA utilizza il formalismo *XSLT (eXtensible Stylesheet Language Transformation)*; sono stati creati dei fogli di stile XSL che vengono associati "on the fly" ai documenti XML per trasformarli e poterne così visualizzare il contenuto.

In particolare, quando la richiesta giunge da un client basato su browser web, un processore XSL genera dei file XHTML che vengono successivamente trasformati in file HTML e inviati al client; se la richiesta arriva da un Applet, i dati vengono restituiti direttamente in formato XML e visualizzati con il rendering opportuno.

La trasformazione di un file XML attraverso un opportuno foglio di stile XSL può avvenire in due diversi modi: tramite il server che si occupa della conversione e restituisce al client un file HTML, come per il sistema INFEA nel caso la richiesta provenga da browser web, oppure direttamente sul lato client.

XSL è suddiviso in due parti: XSLT ovvero le trasformazioni XSL, e *XSL-FO (XSL Formatting Objects)*, ovvero gli oggetti per la formattazione XSL.

Il processo di visualizzazione tramite XSL può essere suddiviso in due parti principali: il primo passo consiste nel costruire un nuovo albero a partire dalla struttura del documento XML cui il foglio di stile si riferisce; il secondo consiste nell'elaborare la nuova struttura ad albero per presentarne il contenuto a video, su carta, a voce (lettura vocale) o su altri media.

La prima parte del processo può a sua volta essere suddivisa in due fasi. In un primo momento si individuano, attraverso dei pattern espressi con il formalismo XPath, gli elementi del documento sorgente che devono comparire nel nuovo albero; in seguito vengono aggiunti gli elementi individuati attraverso l'impiego dei templates, ottenendo così un albero la cui struttura può essere completamente diversa da quella dell'albero sorgente.

La seconda parte del processo rappresenta la fase di formattazione vera e propria ed è implementata attraverso l'uso degli oggetti di formattazione, che rappresentano tutti i costrutti di formattazione che è possibile realizzare con XSL. Infatti la nuova struttura ad albero che il processore XSL genera è in realtà un albero di oggetti di formattazione, le cui foglie rappresentano i caratteri, le immagini o gli altri oggetti che devono essere visualizzati.

Quando XSL viene usato per trasformare un documento XML in un documento HTML, gli oggetti di formattazione utilizzati sono molto simili ai tag propri di HTML.

XSL mette a disposizione due metodologie per realizzare la trasformazione dall'albero sorgente nel nuovo albero che dovrà essere visualizzato: *Template-driven model* e *Data-driven model.*

Il primo modello si applica bene nei casi in cui la struttura del documento sorgente sia ben definita e regolare; in tal caso il foglio di stile XSL contiene un modello della struttura del nuovo albero che verrà completato con i dati del documento sorgente XML.

Il Data-driven, invece, si presta meglio ai casi in cui si devono fomattare documenti XML che hanno una struttura gerarchica irregolare. In questo secondo modello, il foglio di stile è un insieme di regole formate da due parti: la prima individua gli elementi per cui la regola è applicabile, la seconda specifica il tipo di azione da eseguire. Questa azione è tipicamente composta da tre fasi: l'inserimento di un oggetto di

formattazione nel nuovo albero, la scelta delle modalità di visualizzazione attraverso le caratteristiche dell'oggetto, ed infine l'elaborazione di tutti o di parte dei figli dell'elemento selezionato.

I risultati di formattazione migliori si ottengono usando contemporaneamente entrambi i modelli; tuttavia, per semplici applicazioni, è sufficiente l'impiego di un singolo modello di trasformazione.

3.5 XHTML [12]

La porzione HTML presente generalmente nei fogli di stile XSL, per la formattazione di immagini, tabelle ed altri elementi, è stata sostituita nei file XSL impiegati nel sistema INFEA da un altro formalismo: XHTML. XHTML è una raccomandazione ufficiale del W3C e definisce una versione di HTML compatibile con XML o, per meglio dire, ridefinisce HTML sotto forma di applicazione XML piuttosto che sotto forma di applicazione SGML.

Dalla mera osservazione di un documento XHTML risulta difficile accorgersi delle differenze rispetto ad un documento HTML: i tag utilizzati sono gli stessi (<P>, <LI>, <TABLE>...), gli elementi e gli attributi hanno gli stessi nomi familiari di quelli HTML, la sintassi è sostanzialmente conservata. Le differenze, infatti, riguardano non tanto ciò che è permesso, ma ciò che non è permesso: per esempio, XHTML richiede che a tutti i tag iniziali corrisponda un tag finale e che i valori degli attributi siano specificati tra apici. Per questo, un paragrafo che inizi con l'elemento <P> e termini con l'elemento </P> è legale in XHTML, ma un paragrafo che ometta il tag finale </P> non lo è; allo stesso modo l'elemento <TABLE border="0" width="515"> è un tag XHTML legale, mentre <TABLE border=0 width=515> non lo è. XHTML, inoltre, aggiunge alcuni elementi di sintassi a HTML, come la dichiarazione XML e il fatto che i tag di elemento vuoti terminino con la sequenza di caratteri />.

Molti documenti HTML esistenti richiedono modifiche sostanziali per diventare documenti XHTML ben formati e quindi validi; ciononostante, una volta trasformati in documenti XHTML validi, saranno automaticamente documenti XML validi, che potranno esser manipolati con gli stessi editor, parser e altri strumenti per il trattamento di documenti XML.

La maggior parte dei documenti XHTML possono essere visualizzati dai principali browser proprietari, seppur con qualche eccezione rilevante. Per questo motivo, all'interno del sistema INFEA, il client basato su browser web riceve, direttamente dal server, un documento HTML puro e non XHTML, nonostante quest'ultimo sia fortemente utilizzato all'interno dei fogli di stile XSL.

3.6 Xpath [12]

XPath è un linguaggio non-XML utilizzato per identificare particolari porzioni di documenti XML, indicando i nodi in base alla loro posizione assoluta o relativa, al loro tipo, al loro contenuto e ad altri criteri.

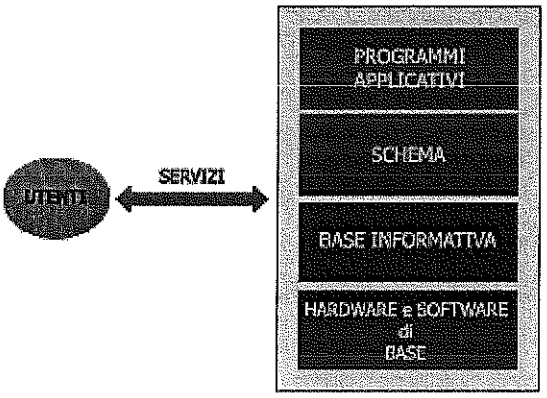
XSLT, come già accennato, utilizza espressioni XPath per cercare corrispondenze con particolari entità del documento XML sorgente, che debbano essere copiate nel documento di output oppure sottoposte ad ulteriori elaborazioni. Le espressioni XPath possono anche rappresentare numeri, stringhe o valori booleani, in modo da permettere ai fogli di stile XSLT di svolgere semplici operazioni aritmetiche per numerare elementi o gestire riferimenti incrociati verso figure, tabelle ed equazioni.

4. IL SISTEMA INFEA

Le tecnologie appena descritte sono state tutte impiegate nella realizzazione del sistema INFEA, che rappresenta un esempio di sistema informativo basato sul Web, del quale adesso andiamo a dare una descrizione specifica, che da per scontati i concetti finora illustrati.

4.1 SISTEMI INFORMATIVI E WEB [1]

Ogni organizzazione per il proprio funzionamento deve disporre di informazioni, che costituiscono una delle principali risorse a disposizione dell'azienda; per questo motivo, è necessario gestire i dati attraverso un particolare sottosistema detto *sistema informativo*. Il sistema informativo è costituito da un insieme di risorse – umane e materiali – e di procedure che permettono di eseguire attività di raccolta, archiviazione, elaborazione e scambio delle informazioni, necessarie per il funzionamento dell'organizzazione stessa.



- FIGURA 1.1 -

Negli ultimi anni, il progresso della tecnologia elettronica e la conseguente riduzione dei costi si sono tradotti nell'impiego degli elaboratori elettronici per agevolare e potenziare il trattamento delle informazioni; l'insieme di strumenti informatici utilizzati per l'elaborazione automatica dell'informazione costituisce una componente essenziale del sistema informativo di un'organizzazione, e prende il nome di *sistema informatico*.

La diffusione crescente del web e la capillare penetrazione della tecnologia ad esso connessa sta causando notevoli cambiamenti rispetto alla gestione ed organizzazione delle informazioni di moltissime aziende, con un notevole impatto sul loro sistema informativo.

In particolare, lo sviluppo di nuove applicazioni basate sul web è da ricercarsi nei vantaggi che questa tecnologia offre, soprattutto in termini di scalabilità e distribuzione.

L'opportunità di utilizzare la piattaforma web per l'implementazione e la gestione di un sistema informativo (Web-based Information System o WIS) rispetto ad un sistema informativo tradizionale, offre notevoli benefici, ma impone comunque il rispetto degli stessi principi.

4.2 WEB-BASED INFORMATION SYSTEM (WIS) [2] [3]

I settori interessati allo sviluppo di sistemi informativi basati sul web comprendono, da un lato, le organizzazioni che hanno una struttura distribuita su un territorio molto vasto o che intendono espandere la propria attività sviluppando applicazioni per il business in rete (il cosiddetto *e-Business*), dall'altro le istituzioni pubbliche. Queste ultime hanno la necessità di raccogliere, archiviare, generare e distribuire una grande quantità di informazioni destinate ad essere condivise sia dai cittadini che dalle imprese.

I sistemi informativi delle istituzioni pubbliche, detti appunto *Public Access WIS (PAWIS)*, sono caratterizzati da un requisito comune, l'*accesso universale*: si cerca di fornire a chiunque, indipendentemente dall'esperienza e dal supporto hardware, la possibilità di utilizzare i servizi o di accedere ai documenti pubblici messi a disposizione dall'istituzione.

La progettazione e la realizzazione di un WIS richiedono, pertanto, nuovi approcci rispetto a quelli utilizzati in un sistema informativo tradizionale e, sebbene si percepiscano chiaramente i vantaggi offerti dalla tecnologia, emergono inevitabilmente nuove problematiche.

Considerati i tre aspetti fondamentali di un sistema informatico – gestione della comunicazione, gestione dell'interfaccia uomo-macchina, gestione dei dati – diversi sono i vantaggi che derivano dall'impiego della tecnologia web.

Gestione della comunicazione

La rete Internet e la suite di protocolli TCP/IP non pongono limiti (geografici o hardware/software) all'espandibilità del sottosistema di gestione della comunicazione. La possibilità di scalare facilmente il sistema da uso locale a uso remoto è uno dei principali vantaggi dei WIS rispetto ai sistemi informativi tradizionali.

Interazione uomo-macchina

Le più importanti caratteristiche del sottosistema di gestione delle interazioni uomo-macchina di un WIS sono i browser grafici e l'impiego di codice (mobile o script-embedded) multipiattaforma per la realizzazione della componente client del sistema.

Gestione dei dati

In un sistema informativo tradizionale, il sottosistema della gestione dei dati è realizzato tramite l'uso di un *DBMS (Data Base Management System)* che implementa lo schema della base di dati; nel caso di un WIS non vi è più una corrispondenza univoca tra lo schema e le proprietà del database a causa della presenza delle URL.

Le URL possono essere viste come puntatori ad informazioni memorizzate in gestori di dati distinti, che possono essere DBMS ma anche file system o programmi (CGI o Servlet) che generano documenti "on the fly".

Il sottosistema di gestione dei dati non è più costituito su un insieme fissato di tipi di dato, ma estensibile con nuovi tipi, e rappresenta un multi-gestore che contempla, tra gli altri, sicuramente anche il tipo DBMS.

4.2.1 Problematiche dei WIS

Per la realizzazione di un WIS efficace ed efficiente, è necessario affrontare problematiche nuove rispetto a quelle riscontrabili per lo sviluppo di un sistema informativo tradizionale.

Meccanismi di ricerca

La realizzazione di un buon motore di ricerca per un WIS non può prescindere dalla definizione e realizzazione di filtri che aiutino l'utente, attraverso interfacce pulite, a compilare interrogazioni prive di ambiguità.

Supporto per il lavoro cooperativo

Un WIS deve supportare sia la collaborazione asincrona (condivisione dei dati tra più soggetti, ognuno impegnato in sessioni di lavoro distinte) che sincrona (condivisione dei dati tra più soggetti impegnati in una sessione di lavoro comune).

Gli strumenti utilizzati vanno dal semplice scambio di posta elettronica, alle più moderne videoconferenze, malgrado l'integrazione tra web e strumenti di lavoro collaborativo siano ancora in una fase embrionale.

Meccanismi di caching locale

Un WIS deve prevedere dei meccanismi sofisticati di accumulazione locale delle informazioni, mirati ad incoraggiare l'utilizzo di applicazioni off-line.

Gestione dei link

Nei WIS i link sono una componente del sistema di gestione dei dati; a causa dei limiti imposti dall'attuale tecnologia web, dovuti soprattutto alla natura integrata dei link all'interno del formalismo HTML, si presentano alcuni problemi: la difficoltà nel realizzare strumenti per la gestione, il controllo e/o l'aggiornamento dei link; la difficoltà nell'interpretazione delle relazioni esistenti tra i link e le parti del sistema.

Per ovviare definitivamente a queste e ad altre problematiche, il web si sta orientando verso l'impiego di un formalismo più strutturato rispetto a HTML, il linguaggio XML.

Sicurezza ed Autenticazione

Come per un sistema informativo tradizionale, anche per un WIS occorre implementare alcuni meccanismi per la sicurezza e la gestione sicura dei dati; in particolare, mentre in un sistema informativo tradizionale si fa riferimento a più tipologie di utenti, ognuna con specifici diritti sui dati, nel caso di un WIS l'impiego di firewall o altri strumenti software è efficace solo nell'impedire tentativi di intrusione nei computer o nelle reti, ma non garantisce la sicurezza delle transazioni sui dati.

Affinché un WIS possa considerarsi sicuro è necessario che garantisca le seguenti condizioni:

- CONFIDENZIALITÀ: la comunicazione tra più parti deve essere ristretta solo alle parti stesse;
- AUTENTICAZIONE: ogni soggetto deve essere sicuro dell'identità del suo interlocutore o dell'origine dell'informazione cui sta accedendo;
- INTEGRITÀ DEI DATI: i dati non devono poter essere modificati durante il trasferimento tra i soggetti di una comunicazione;
- ACCESSO SELETTIVO AI SERVIZI: è preferibile che ogni soggetto percepisca solo i servizi cui può accedere.

Accessibilità

L'accessibilità ai servizi ed alle informazioni rese disponibili dai WIS al maggior numero di utenti possibile è un requisito molto importante, non solo per i WIS pubblici.

È necessario affrontare due aspetti, quello tecnico e quello cognitivo. Per l'aspetto tecnico, l'obiettivo è quello di fornire alternative alla navigazione tramite browser grafici, come ad esempio Lynx; l'aspetto cognitivo impedisce all'utente, attraverso una serie incoerente di link, di perdere il suo obiettivo di partenza.

Per quanto riguarda il primo aspetto, è opportuno distinguere due casi, poiché in un WIS si possono trovare sia interfacce per la presentazione delle informazioni, sia interfacce per l'accesso ai servizi.

Nel primo caso vengono seguite delle linee guida, proposte dal *WAI (Web Accessibility Initiative)* [e] del W3C ed universalmente riconosciute. Per l'accesso ai servizi sono stati effettuati recentemente alcuni passi in avanti, grazie all'integrazione dei client web con strumenti di tecnologie assistive, che consentono alle persone disabili l'utilizzo del computer, e all'implementazione di API ad hoc per i principali linguaggi impiegati nella realizzazione di codice mobile.

4.2.2 Tecnologia per i WIS

La tecnologia Java è ampiamente usata nella realizzazione di WIS, come ad esempio per gli Applet, le Servlet o le API JDBC nell'implementazione della business logic della componente server-side del sistema. Inoltre esiste la possibilità di sfruttare Java e le sue potenzialità per la realizzazione di WIS, grazie ad una sua caratteristica ancora limitatamente utilizzata, cioè la capacità di accedere ad oggetti remoti e di realizzare programmi in cui gli oggetti, distribuiti su diversi host, comunicano tra loro tramite la rete.

L'utilizzo di CORBA (tramite il protocollo IIOP) e RMI (tramite i protocolli JRMP/IIOP) consente di realizzare delle applicazioni in grado di superare parzialmente i limiti di unidirezionalità imposti dal protocollo HTTP.

Java offre particolari contributi all'interno dei tre sistemi componenti un WIS:

Gestione delle comunicazioni

Attraverso l'uso combinato dei protocolli HTTP e IIOP o JRMP, si possono arricchire le caratteristiche di espandibilità e modularità dei WIS con la bidirezionalità e l'invocazione di metodi remoti.

---

[e] http://www.w3.org/WAI

Interazione uomo-macchina

L'impiego della tecnologia RMI o CORBA consente agli Applet Java di integrarsi come parte di un'applicazione distribuita, con i vantaggi che ne derivano.

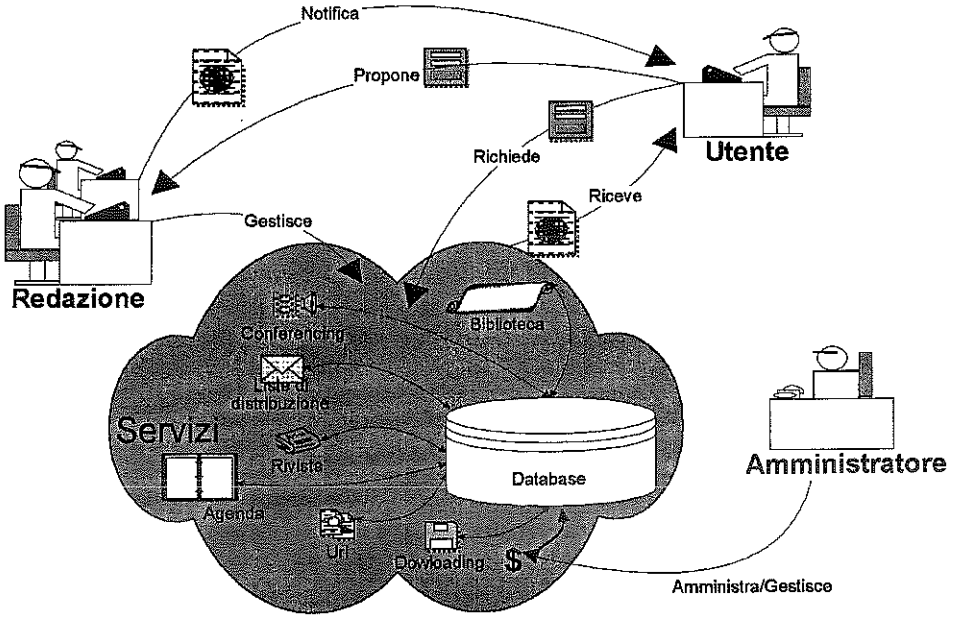
Gestione dei dati

L'utilizzo della comunicazione tra oggetti rende più efficace l'interazione con il gestore dei dati; se quest'ultimo utilizza la tecnologia ad oggetti, si trae un ulteriore vantaggio a causa della corrispondenza tra le classi Java e quelle del DBMS.

4.3 INFEA: CARATTERISTICHE [4]

Il sistema informativo INFEA è un esempio di WIS realizzato per la Pubblica Amministrazione, nel caso specifico per il Ministero dell'Ambiente.

L'obiettivo di INFEA è di fornire, in maniera integrata, sia strumenti di consultazione, sia strumenti di gestione delle varie sorgenti di informazioni, con la possibilità di fruire le offerte informative degli Enti coinvolti nel sistema (come ad esempio l'ISFOL), i centri regionali di Educazione Ambientale, i laboratori territoriali e lo stesso Ministero.



- FIGURA 1.2 -

La figura 1.2 mostra uno schema ad alto livello del sistema INFEA, composto da un insieme di servizi che accedono al database tramite richieste inviate da diverse tipologie di utenti, ognuna avente ruoli e permessi differenti.

Tutti gli oggetti del sistema INFEA sono realizzati mediante l'utilizzo del linguaggio Java, principalmente per garantire uno dei più importanti principi guida nella progettazione di INFEA, ossia l'indipendenza dalle caratteristiche hardware e software della piattaforma.

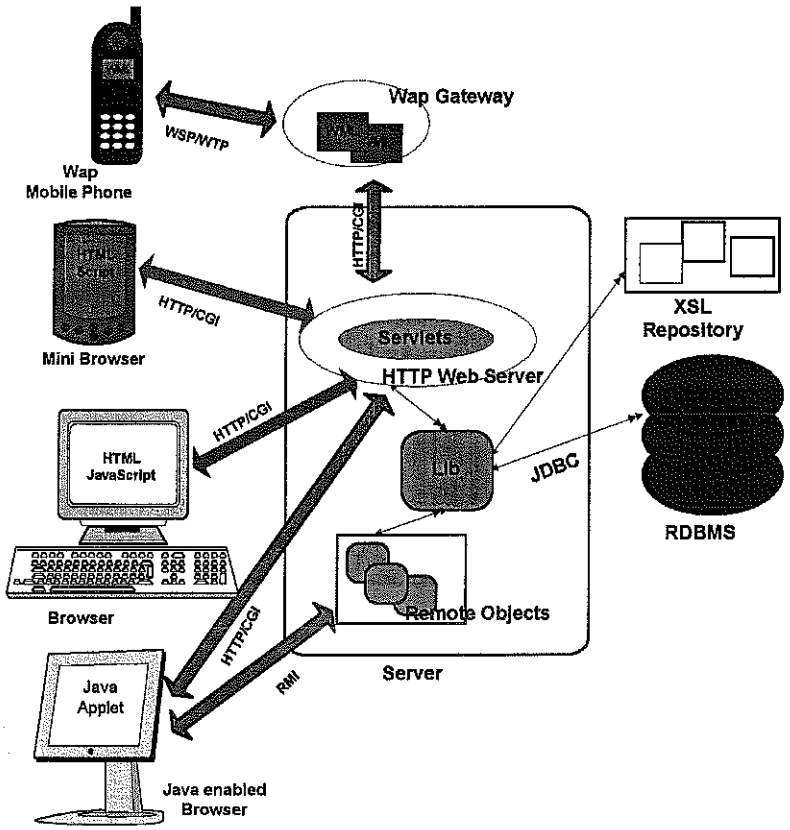
La tecnologia per la programmazione distribuita adottata in INFEA è RMI, che consente di eliminare alcuni limiti imposti dal protocollo HTTP, ad esempio la trasmissione unidirezionale o la mancanza di stato.

Il sistema ha un'architettura client/server a più livelli in cui il primo livello consiste nell'interfaccia utente (*presentation logic*), il secondo comprende la logica dell'applicazione (*business logic*) ed infine il terzo è costituito dal gestore dei dati, un DBMS.

La presentation logic è implementata utilizzando il linguaggio Java, il formalismo HTML ed il linguaggio JavaScript, che ha permesso di implementare anche la parte della business logic sul lato client.

La business logic ed il data access sono implementati tramite applicazioni scritte in Java, mentre i dati sono memorizzati in un DBMS relazionale.

4.3.1 Architettura software di INFEA



- FIGURA 1.3 -

La figura 1.3 mostra l'architettura globale del sistema informativo INFEA, secondo lo schema client/sever multi-tier sopradescritto.

Il server presenta due tipologie di programmi: Servlet ed oggetti remoti. Le Servlet vengono invocate tramite il web server ed eseguono esclusivamente funzioni di interrogazione del DBMS. Gli oggetti remoti vengono riferiti, tramite il protocollo RMI, direttamente dai client; qui sono implementate le funzionalità di gestione e modifica dei dati e di autenticazione dei client.

Le Servlet e gli oggetti remoti condividono librerie comuni per l'interazione con il DBMS.

Prima di essere trasferiti tra client e server, i dati vengono strutturati secondo il formalismo XML ed eventualmente trasformati

attraverso fogli di stile XSL, allo scopo di garantire una maggiore modularità ed un maggiore controllo sulla correttezza delle operazioni.

Sono state definite tre tipologie di client:

Client basato su Applet Java

Questa tipologia di client rappresenta gli utenti amministratori del sistema, ed è utilizzata quando il client implementa una parte della business logic oltre che della presentation logic; questa soluzione è stata adottata per realizzare le interfacce di inserimento, modifica e cancellazione dei dati, operazioni nelle quali, oltre alla verifica della consistenza dei dati forniti, avvengono in tempo reale interazioni con il server per facilitare i compiti dell'utilizzatore.

La comunicazione con il lato server avviene tramite l'invocazione di metodi remoti con il protocollo Java RMI.

Client basato su browser web

Questa tipologia è utilizzata quando la funzione principale del client è quella di presentazione. È stato utilizzato il formalismo HTML, sia per descrivere l'interfaccia per la formulazione della richiesta, sia per la restituzione dei risultati. Anche in questo caso sono implementate delle funzionalità minime di controllo sui dati, per mezzo del linguaggio JavaScript.

La comunicazione con il lato server avviene tramite il protocollo HTTP/CGI.

Client basato su dispositivi mobili

Questa tipologia di client fornisce alcuni servizi di interrogazione al sistema INFEA tramite computer palmari o telefoni cellulari che supportino il protocollo WAP. In entrambi i casi l'implementazione si basa su microbrowser disponibili su questi dispositivi, e utilizza per i palmari un sottoinsieme particolare dell'HTML, e per i cellulari WML (Wireless Markup Language).

L'interfaccia basata su Applet Java è rivolta allo staff editoriale INFEA e al management del servizio Sviluppo Sostenibile. Essa offre

funzionalità di ricerca molto sofisticate e funzioni per la gestione dei dati, quali inserimento, modifica e cancellazione. L'accesso tramite questa interfaccia è consentito solo attraverso un protocollo di autenticazione (user-name e password).

Le specifiche tecniche per l'implementazione si basano sull'utilizzo delle classi Swing di Java, per la realizzazione di un desktop in cui si possono attivare e disattivare finestre, sull'esempio della scrivania dei moderni sistemi operativi.

Sono state disegnate due modalità di comunicazione con il lato server dell'applicazione: la prima invoca direttamente metodi su oggetti remoti tramite il protocollo RMI e viene utilizzata durante il passaggio di valori tra client e server (ad esempio, l'inizializzazione di una lista dell'interfaccia con valori presenti nella base di dati); la seconda invoca una Servlet tramite il protocollo CGI ed è utilizzata in particolari situazioni di visualizzazione di informazioni. Per utilizzare questa interfaccia è necessario installare il plug-in Java se non è già presente nella postazione utilizzata.

L'interfaccia basata su browser web è rivolta alla comunità di utenti occasionali, spesso non esperti nei contenuti trattati dal sistema informativo e/o nell'uso delle tecnologie informatiche.

La realizzazione basata prevalentemente su HTML, con l'ausilio di procedure JavaScript, consente la massima fruibilità, in quanto non richiede nessuna operazione preliminare, come ad esempio l'installazione di un plug-in Java necessaria per il client basato su Applet.

Il meccanismo di comunicazione con il lato server è quello tradizionale fornito dal protocollo CGI: i dati che rappresentano la richiesta effettuata dall'utente vengono inviati ad una Servlet Java, che s'incarica di elaborare e generare dinamicamente la pagina da restituire al client come risultato.

L'interfaccia per dispositivi mobili è stata prevista per garantire ed agevolare eventuali sviluppi futuri di servizi avanzati.

Tramite il protocollo CGI viene invocata una Servlet Java che genera pagine di risultati ottimizzate per le caratteristiche fisiche del dispositivo mobile da cui arriva la richiesta.

Le interfacce progettate sono più limitate per quanto riguarda la potenza di ricerca e sono state disegnate per essere adatte al dispositivo in questione. Nel caso di telefoni cellullari sono stati realizzati due servizi: uno per la ricerca di Enti con condizioni ridotte (i filtri, infatti, riguardano solo la tipologia, il nome e la provincia dell'Ente) ed uno per la ricerca dei referenti (una sorta di rubrica telefonica su cui si possono esprimere condizioni sul nome della persona da ricercare, sulla provincia, sulla tipologia e denominazione dell'Ente cui questa appartiene).

4.3.2 Logica dell'applicazione (Business logic)

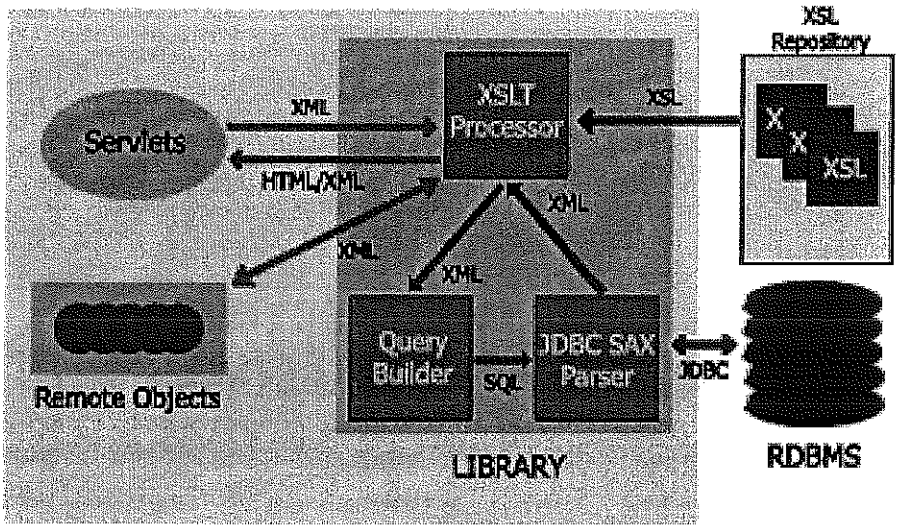
L'intera business logic è stata realizzata mediante il linguaggio Java per garantire l'indipendenza dalle caratteristiche hardware/software della piattaforma.

L'interazione con le varie tipologie di client avviene tramite due componenti: Servlet Java ed oggetti remoti. Le Servlet vengono invocate tramite il web server da un client qualsiasi, mentre gli oggetti remoti sono invocati, attraverso il protocollo RMI, solamente da client basati su Applet Java. L'invocazione di Servlet da parte di dispositivi mobili giunge al web server attraverso il WAP Gateway con cui si è connessi.

Tutte le richieste di informazioni contenenti le varie condizioni di selezione, provenienti da qualsiasi tipologia di client (e quindi sia dalle Servlet che dagli oggetti remoti), sono codificate, attraverso il formalismo XML, in una stringa che viene inviata al modulo *XSLT Processor*. Questo modulo, tramite le informazioni contenute in un foglio di stile collegato alla richiesta, genera una stringa XML che viene inviata al modulo *Query Builder*. Il Query Builder trasforma la richiesta in una query SQL e la invia al modulo *JDBC Sax Parser*, che invoca

direttamente il DBMS tramite le classi JDBC di Java e trasforma il risultato dell'interrogazione in una stringa XML. Questa viene restituita al modulo XSLT Processor, che infine, basandosi su un foglio di stile che descrive le modalità di restituzione, genera il risultato per l'opportuno client.

Sia le Servlet che gli oggetti remoti condividono una libreria di classi per l'accesso al DBMS e per la trasformazione da XML a SQL e viceversa (figura 1.4).



- FIGURA 1.4 -

4.3.3 Il gestore dei dati INFEA

La maggior parte dei dati relativi al sistema informativo INFEA è raccolta in documenti fortemente strutturati, e proprio per questo motivo si è scelto di impiegare un DBMS. Sarebbe stato possibile orientarsi verso un *DBMS con modello dei dati ad oggetti (OODBMS)*, che avrebbe senza dubbio facilitato il mapping tra il modello del documento

con cui interagisce l'utente (interfaccia) ed il modello dei dati del DBMS. Lo stato attuale della tecnologia OODBMS, però, non ancora matura e affidabile, ha imposto la scelta di un *DBMS con modello dei dati relazionale (RDBMS)*, che adotti SQL come linguaggio per la manipolazione dei dati.

L'accesso alla base di dati avviene tramite le classi JDBC di Java. Per la trasformazione dal modello dell'utente al modello del DBMS si è adottato un approccio cosiddetto *model-driven*: ogni documento XML è rappresentato da una struttura ad albero i cui nodi fanno riferimento alle entità della base di dati, mentre le proprietà sono descritte dagli attributi dei nodi dell'albero.

4.3.4 Considerazioni sull'architettura

L'architettura multi-tier comporta notevoli vantaggi sia nella gestione del sistema che nella manutenzione e nelle eventuali estensioni del software realizzato; qualora ce ne fosse la necessità, è possibile distribuire il carico del sistema su più macchine, sia per problemi di sicurezza che di prestazioni (il DBMS può girare su una macchina diversa da quella su cui gira il web server).

E' possibile cambiare o aggiungere modalità di presentazione senza per questo dover intervenire necessariamente sul software che realizza la business logic; oppure cambiare il DBMS senza alcuna modifica a tutto il software realizzato.

Grazie all'impiego dei DBMS sono garantite consistenza, privatezza e sicurezza dei dati.

L'utilizzo di XML assicura la portabilità dei dati tra le differenti componenti del sistema informativo e l'eventuale approccio con altri strumenti; i moduli software sono più facilmente mantenibili ed espandibili poiché ognuno realizza le sole funzionalità richieste dal livello cui si colloca.

Il lato server è stato progettato per gestire applicazioni multi-client con differenti livelli di accessibilità, dalle sofisticate interfacce grafiche a quelle testuali per telefoni mobili, creando un ambiente collaborativo fondato sul paradigma noto come *Web-based Computing Model*. La definizione di differenti categorie di client, oltre a fornire livelli di autorizzazione diversi, consente di utilizzare l'interfaccia più rispondente alle risorse hardware/software di cui un utente dispone. Per ridurre il tempo di attivazione del client basato su Applet Java, l'applicazione è costituita da packages distinti che vengono scaricati solo al momento del bisogno.

BIBLIOGRAFIA

[1] **Basi di Dati Relazionali e ad Oggetti**
di Albano A., Ghelli G., Orsini R.
ed. Zanichelli (2000)

[2] **Web Architectures for Database Access**
(Atti del convegno WebNet "World Conference 1999 on the WWW and Internet", Honolulu – Hawaii, USA)
articolo pgg. 1473-1475
di Aloia N., Concordia C, Miori V.

[3] **Web-based Information Systems**
(Atti del convegno "16th World Computer Congress 2000 Information Technology for Business Management", Pechino – Cina)
articolo pgg. 581-588
di Aloia N., Concordia C, Miori V.

[4] **Architettura software, funzionalità e meccanismi del Sistema INFEA**
di Tardelli S., Versienti L.
rapporto CNUCE-B4-2002-004 (2002)

[5] **HTML: The Complete Reference**
di Powell T.A.
ed. McGraw-Hill (2000)

[6] **HTML for the World Wide Web with XHTML and CSS: Visual QuickStart Guide (5th Edition)**
di Castro E.
ed. Peachpit Pressù (2002)

[7] **Java Servlet Programming Bible**
di Rajagopalan S., Rajamani R., Krishnaswamy R., Vijendran S.
ed. John Wiley & Sons (2002)

[8] **Java & XML (2nd Edition) - Solutions to Real-World Problems**
di McLaughlin B.
(2001)

[9] **Java Servlet Programming (2nd Edition)**
di Hunter J., Crawford W.
(2001)

[10] **http://www.latoserver.it/java/servlets**
articolo di Capodieci G. del *24/11/2001*

[11] **XML Step by Step**
di Young M.
ed. Microsoft (2000)

[12] **XML Guida di Riferimento**
di Harold E.R., Means W.S.
ed. O'Reilly (2000)

[13] **XML Bible (2nd Edition)**
di Harold E.R.
ed. John Wiley & Sons (2001)

[14] **I disabili nella società dell'Informazione - Norme e tecnologie**
di AA.VV.
ed. Franco Angeli (2002)

[15] **Considerazioni per la realizzazione di sistemi informativi basati su web accessibili ai disabili**
(6° convegno "Informatica Didattica e Disabilità")
di Aloia N., Concordia C., Furfari F., Miori V.