

ATTI DEL  
**DECIMO CONVEGNO NAZIONALE SU  
 SISTEMI EVOLUTI PER BASI DI DATI**

A cura di  
 Paolo Ciaccia  
 Fausto Rabitti  
 Giovanni Soda

19-21 Giugno 2002  
 Portoferraio - Isola d'Elba



# Un sistema informativo basato sul Web

Nicola Aloia, Sabrina Tardelli, Loredana Versienti

CNUCE - Istituto del C.N.R.  
Via Moruzzi 1, 56126 Pisa  
Telefono 050-31562997 - Fax 050-3138091/2  
E-mail: Nicola.Aloia@cnuce.cnr.it

**Abstract.** I sistemi informativi basati sul Web (WIS) suscitano sempre maggiore interesse, sia nella comunità scientifica sia in quella industriale, grazie soprattutto alla loro potenzialità di fruizione da parte di una comunità enormemente più ampia di quella delle tradizionali realizzazioni client/server basati su rete proprietaria. Tale possibilità è enfatizzata dalle caratteristiche intrinseche della tecnologia Web: standard aperti e software molto spesso gratuito e facilmente disponibile. Nel presente lavoro illustriamo il WIS della Direzione per lo Sviluppo Sostenibile del Ministero dell'Ambiente e della Tutela del Territorio, che ha finanziato l'attività.

## 1 Sistemi informativi basati sul Web

L'opportunità di utilizzare il Web, per la realizzazione di sistemi informativi, offre notevoli benefici, sebbene ponga nuove problematiche per la loro realizzazione [3]. Un sistema informativo basato sul Web (WIS) è un sistema distribuito, implementato e gestito con tecnologia Web [5]. In un WIS la stessa componente umana, può essere largamente distribuita, e le informazioni assumono forme sempre più complesse, a causa della continua evoluzione della tecnologia Web. Un'area rilevante per la realizzazione di WIS è costituita dalle istituzioni pubbliche. Queste raccolgono, generano e distribuiscono, una gran quantità d'informazioni, la cui conoscenza può essere molto importante per i cittadini e per le imprese. Questi sistemi, chiamati Public Access WIS (PAWIS) [6] devono fornire servizi accessibili a tutti [7,8,9], a prescindere dall'esperienza e dalle dotazioni hardware dell'utilizzatore [2].

## 2 Il sistema informativo INFEA

In questo capitolo descriviamo, brevemente, il disegno e la realizzazione del PAWIS della Direzione per lo Sviluppo Sostenibile del Ministero dell'Ambiente e della Tutela del Territorio. L'obiettivo di INFEA è di fornire in maniera "integrata", sia strumenti di consultazione, che strumenti di gestione, delle varie sorgenti di informazioni, con la possibilità di fruizione globale delle offerte informative dei vari enti coinvolti nel Sistema, come l'ISFOL, i centri regionali di Educazione Ambientale, i laboratori territoriali e lo stesso Ministero. Per quanto riguarda la componente "risorse umane" del sistema informativo è apparsa necessaria la costituzione di una "redazione",

composta di esperti in materie ambientali, che controlli il contenuto informativo del sistema e che svolga funzioni di filtro e di coordinamento per l'accettazione e la diffusione dell'informazione che affluisce al sistema via Internet. È da notare che la redazione può essere composta di utenti distribuiti sul Web. In figura 1 è schematizzata la struttura del sistema informativo Infea.

La componente informatica è costituita da un insieme di *servizi* e dal *sottosistema d'interazione*. Tutti i servizi (Fig.1) hanno come fondamenta una *base di dati* che integra le varie sorgenti di informazione. All'interno del sistema sono definiti un certo numero di *Ruoli di Utente*. Il ruolo determina l'insieme dei servizi, e per ognuno di questi le funzioni disponibili per l'utente, e di conseguenza anche l'interfaccia uomo-macchina. È stato definito un meccanismo di autorizzazioni per gli oggetti del sistema basato sui ruoli di utente.

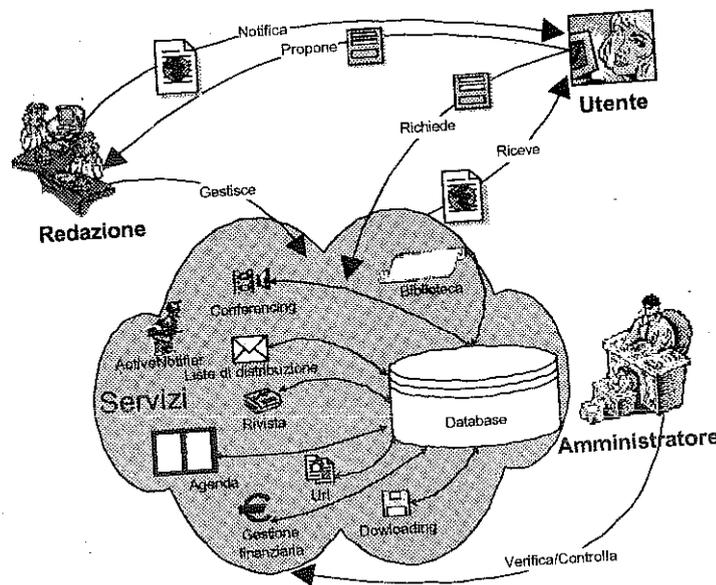


Figura 1 Schema del sistema informativo Infea

### 3 Architettura software di INFEA

La figura 2 mostra l'architettura globale del sistema informativo INFEA, secondo uno schema client/server multi-tier.

#### 3.1 INFEA: Logica di presentazione.

La logica di presentazione è realizzata tramite un'interfaccia *multi-client*. La progettazione di queste interfacce ha prodotto un insieme di diagrammi a stati finiti, per descrivere l'aspetto comportamentale, ed un insieme di descrizioni di layout, fornite ad un grafico professionista, che di concerto con gli addetti del Ministero ne ha definito gli aspetti estetici.

Il server è in grado di riconoscere il dispositivo da cui arriva la richiesta ed attivare l'opportuna interfaccia utente. Con questo meccanismo, che sarà dettagliato nel seguito, è possibile estendere le tipologie di client, per esempio realizzando strumenti di accessibilità, con le caratteristiche accennate in 1.1. Al momento sono state realizzate tre tipologie di client:

- *Client basato sul Web browser.* Questa tipologia è utilizzata quando la funzione preponderante del client è di presentazione. È stato utilizzato il formalismo HTML per la rappresentazione grafica, sono state implementate funzionalità minime di controllo sui dati utilizzando il linguaggio Javascript e la comunicazione col lato server avviene tramite il protocollo HTTP/CGI per l'attivazione di *Servlet Java*. Fig. 2

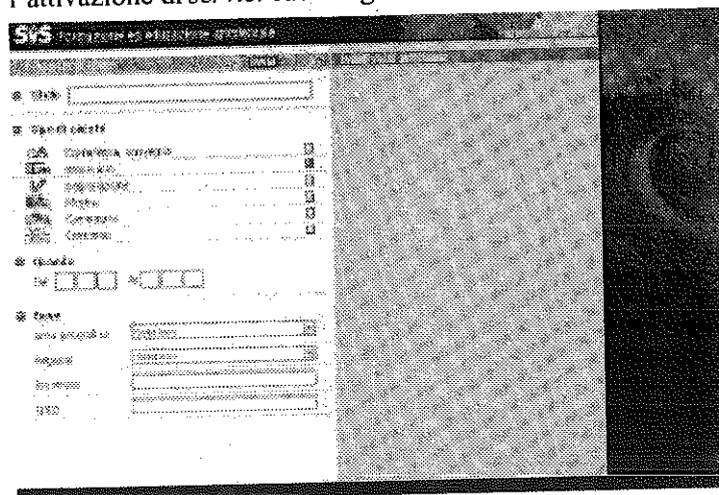


Figura 2 Esempio di interfaccia basata sul Browser

- *Client basato su Applet Java.* È utilizzato nei casi in cui, oltre alla logica di presentazione, il client implementa anche una parte della business logic. Questa soluzione è stata adottata per le interfacce d'inserimento, modifica e cancellazione dei dati, che comunicano col server tramite RMI di Java. In figura 3 è mostrato un esempio di tale interfaccia. L'idea è stata di realizzare la scrivania dell'*Infea-WebWorkstation*, che si autoconfigura secondo il ruolo dell'utente. La scrivania, oltre a finestre per l'interfaccia di immissione/modifica, può contenere altri oggetti, quali ad esempio l'*ActiveNotifier* (oggetto attivo per la segnalazione di determinati eventi), e può essere facilmente estesa per la realizzazione di ulteriori servizi per il lavoro cooperativo (ad es. directories condivise via Web).

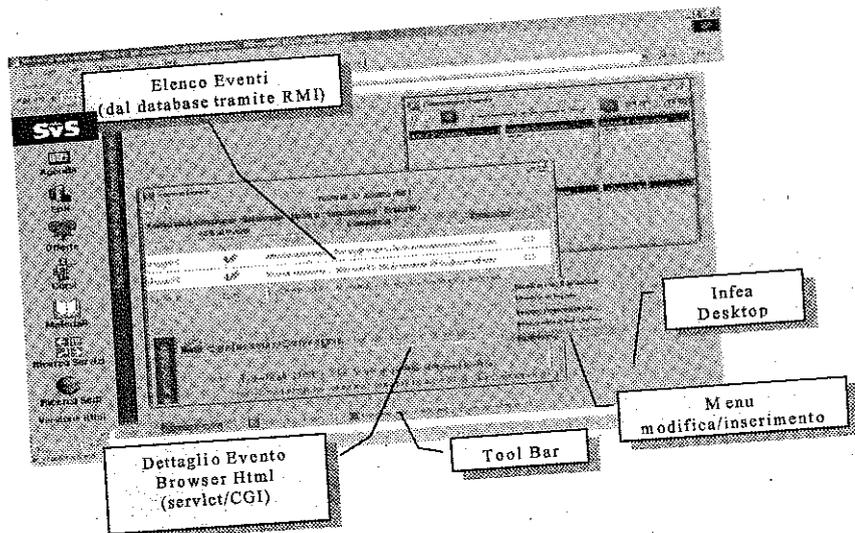


Figura 3 Esempio di interfaccia basata su applet Java

- *Client basati su dispositivi mobili.* Questa tipologia di client, fornisce alcuni servizi di interrogazione tramite computer palmari e telefonini, che supportano il protocollo WAP. In entrambi i casi l'implementazione si basa sui *microbrowsers*, utilizzando un sottoinsieme dell'HTML, nel caso di computer palmari, e del WML (Wireless Markup Language), nel caso dei telefoni mobili. Nelle figure 4 e 5 sono mostrati esempi di tale interfaccia.

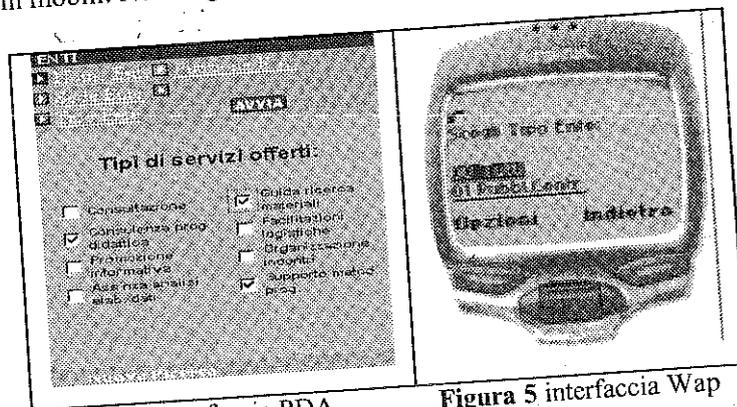


Figura 4 interfaccia PDA

Figura 5 interfaccia Wap

Nella figura 6, le immagini che compaiono sul lato sinistro rappresentano le tipologie di client; le doppie frecce di colore verde, mostrano i protocolli di comunicazione che essi utilizzano.

### 3.2 INFEA: Logica dell'applicazione.

Tutta la logica dell'applicazione è stata realizzata mediante l'utilizzo del linguaggio Java, principalmente per garantire l'indipendenza dalle caratteristiche. L'interazione con le varie tipologie di client descritte nel paragrafo precedente avviene mediante

due componenti: *servlet* ed *oggetti remoti*. I *servlet* sono invocati tramite il Web Server da un qualsiasi client, mentre gli *oggetti remoti* sono invocati, tramite il protocollo RMI, solo da Client basati su Applet Java. I *servlet* e gli *oggetti remoti* condividono una libreria di classi per l'accesso al DBMS e per la trasformazione dei dati dalla logica di presentazione a quella dell'applicazione e viceversa. In figura 7 è illustrato con maggiori dettagli questo processo.

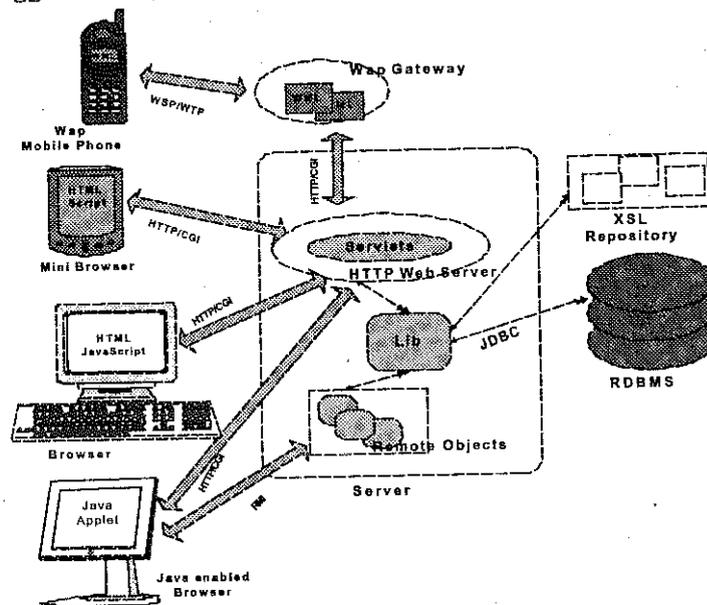


Figura 6 Architettura del sistema Infea

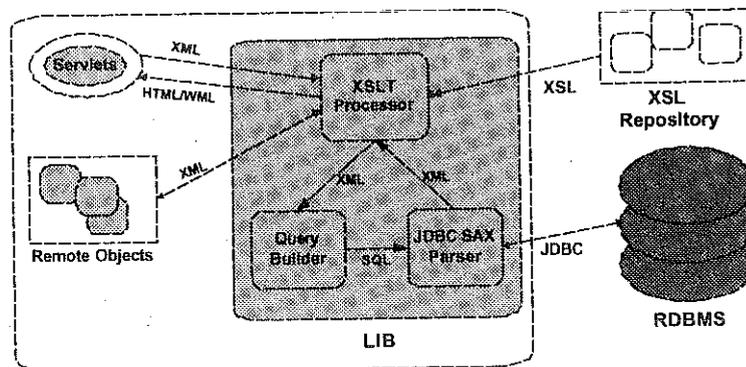


Figura 7 Processo di trasformazione

Sia dai *servlet* che dagli *oggetti remoti*, le richieste di informazioni, contenenti le varie condizioni di selezione, sono codificate in una stringa, utilizzando il linguaggio XML, che è inviata al modulo *XSLT Processor*. Questo modulo, tramite le informazioni contenute in un foglio di stile collegato alla richiesta, genera una stringa XML che è 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 il DBMS, tramite le classi JDBC di Java, e trasforma il risultato dell'interrogazione in una stringa XML.

che è restituita al modulo *XSLT Processor*. Il modulo *XSLT Processor*, basandosi su un foglio di stile che descrive le modalità di restituzione, genera il risultato per l'opportuno client.

### 3.3 INFEA: il gestore dei dati

La maggior parte dei dati relativi al sistema informativo INFEA è raccolta da documenti fortemente strutturati; soprattutto per questo motivo abbiamo deciso di utilizzare un DBMS. Lo stato attuale della tecnologia degli OODBMS, ancora non abbastanza matura ed affidabile, ci ha fatto propendere per l'utilizzo di DBMS con modello dei dati relazionale, perché è una tecnologia affidabile e largamente utilizzata, oltre ad essere l'unica, al momento, ad offrire driver standard per l'utilizzo dal linguaggio Java. L'accesso alla base di dati avviene tramite le classi JDBC di Java, perciò non esiste la dipendenza da specifici RDBMS, purché adottino lo standard SQL [1]. Per la trasformazione dal modello dell'utente (ad oggetti) al modello del DBMS (relazionale) abbiamo adottato un approccio *model-driven*. Ogni documento XML è rappresentato da una struttura ad albero i cui nodi fanno riferimento alle entità della base di dati, le cui proprietà sono descritte dagli attributi dei nodi dell'albero. Il mapping è stato implementato utilizzando le classi SAX e JDBC di Java. Il processo di trasformazione è stato illustrato in figura 7.

### 3.4 Considerazioni sull'architettura.

L'architettura multi-tier adottata comporta notevoli benefici sia nella gestione del sistema sia nella manutenzione ed eventuali estensioni del software realizzato. Per esempio, è possibile distribuire il carico del sistema su più macchine sia per problemi di sicurezza che di prestazioni. È possibile cambiare o aggiungere modalità di presentazione senza la necessità di intervenire sul software che realizza la business logic. È possibile cambiare il DBMS (grazie all'utilizzo delle JDBC) senza intervento sul software realizzato. L'utilizzo di XML garantisce la portabilità dei dati tra le differenti componenti del sistema informativo e l'eventuale utilizzo con altri strumenti.

Il lato server è stato progettato per gestire applicazioni multi-client con differenti livelli d'accessibilità, da quella tramite sofisticate interfacce grafiche a quelle testuali per telefonini mobili, creando un ambiente collaborativo basato sul paradigma noto come *Web based computing model*. L'utilizzo di oggetti distribuiti, permette di superare molti dei limiti delle applicazioni web tradizionali basate su HTML/HTTP, di incrementare notevolmente le prestazioni e di ridurre il carico di rete.

Per ridurre il tempo di attivazione del client basato su applet java, l'applicazione è costituita da packages distinti che sono scaricati solo al momento del bisogno; le dimensioni dei pacchetti sono ridotte effettuando una compressione dei dati prima della comunicazione. La nostra esperienza di utilizzo, ha dimostrato che configurando opportunamente il plug-in Java (cioè abilitando jar caching, jtc, etc.) nell'ambiente del client, l'interfaccia fornisce buone prestazioni.

## 4 La progettazione di INFEA.

La progettazione del sistema informatico ha seguito i principi classici della progettazione del software, offrendo, alcune soluzioni a note problematiche dei WIS [4].

**Gestione dei link.** Tutte le URL, che riferiscono ad oggetti interni ad INFEA (es. link a documenti ipertestuale) o esterni (es. link a pagine Web degli enti), sono memorizzati nel database come proprietà degli oggetti cui si riferiscono. In questo modo è possibile controllare la loro integrità verificando, prima della presentazione del valore, che l'oggetto riferito sia effettivamente raggiungibile. E' inoltre possibile costruiti dinamicamente elenchi di URL significativi, ecc.

**Qualità dei dati.** La qualità dei dati, che affluiscono al sistema via Web, è garantita oltre che dai controlli d'integrità realizzati tramite l'interfaccia d'immissione e modifica, dal vaglio della *Redazione*, che determina ciò che è utile e corretto per la diffusione.

**Gestione della concorrenza.** L'accesso alla base di dati di Infea avviene sempre tramite le interfacce illustrate in fig.3. Poiché il documento dell'utente è complesso, può richiedere intervalli temporali abbastanza lunghi per la modifica. Abbiamo perciò implementato la gestione della concorrenza al di fuori del DBMS, tramite un meccanismo di *long transaction* basato su *check-in check-out* con timeout. Ad ogni oggetto del database è associato l'identificativo dell'utente owner, per poter, tra l'altro, controllare tramite le autorizzazioni sui ruoli l'ammissibilità delle operazioni. I casi di conflitto si possono verificare solo con utenti aventi lo stesso diritto sull'oggetto. La politica del timeout è stata adottata perché i dati, che al momento sono gestiti, non sono critici.

## 5 Conclusioni

Nel presente lavoro sono state analizzate le caratteristiche di un sistema informativo basato sul Web per la pubblica amministrazione. Sono stati presentati gli strumenti tecnologici utilizzati per la realizzazione di INFEA e fornita una descrizione dell'architettura software. Il lavoro è stato finanziato dalla Direzione per lo Sviluppo Sostenibile del Ministero dell'Ambiente e della Tutela del Territorio ed è fruibile al sito <http://www.infea2001.cnuce.cnr.it>. Si ringrazia il gruppo Wis del CNUCE e quanti del Ministero hanno contribuito per la buona riuscita del progetto.

## Bibliografia

- [1] Aloia N., Concordia C. "Accesso a Basi di Dati via Web" Proc. of SEBD '98 "Sistemi evoluti per Basi di Dati", Ancona giugno 1998, (pp. 3-18)
- [2] Aloia N., Concordia C., Furfari F. Miori V. "Considerazioni per la realizzazione di sistemi informativi basati su Web accessibili ai disabili"

Atti del 6<sup>th</sup> Convegno Nazionale Informatica, Didattica e Disabilità – Andria (BA), novembre 1999

- [3] N. Aloia, C. Concordia, F. Furfari, V. Miori: *Caratteristiche, problematiche e tecnologia dei sistemi informativi basati sul Web*, Rivista di Informatica AICA, vol. XXXI n.2 settembre 2001 (pp. 95-108)
- [4] Halasz F. "Reflections on NoteCards: Seven Issues for the Next Generation of Hypermedia Systems", Communication of the ACM, 31 (7) pp 836-852, July 1988.
- [5] Isakowitz T., Bieber M. and F. Vitali: "Web Information Systems", Communication of the ACM, Vol. 41, No. 7, July 1998.
- [6] Kambil A., and Ginsburg M. "Public Access Web Information Systems: Lessons from the Internet EDGAR project", Communication of the ACM 41 (7) pp.91-98, 1998
- [7] Nielsen Jakob "The art of navigating through Hypertext" Communication of the ACM 33(3) 1990, pp 296-310
- [8] W3C Recommendation "Web Content Accessibility Guidelines 1.0", <http://www.w3.org/TR/WCAG10/> Maggio 1999.
- [9] W3C Recommendation <http://www.w3.org/TR/xslt>, Novembre 99