

L'USO DI INTERNET/INTRANET COME STRUMENTO PER AGEVOLARE I PROCESSI DECISIONALI IN AMBITO SANITARIO

Vittoria Liguori - Sistemi Territoriali

Massimo Arcà, Giulia Cesaroni, Claudio Grego – Osservatorio Epidemiologico Regione Lazio

1. Abstract

Tutti coloro che operano nella sanità (primari, responsabili amministrativi, medici di base, ecc.) e gli utenti stessi del Servizio Sanitario Nazionale, cioè i cittadini, necessitano sempre più di strumenti semplici tramite cui accedere a dati aggiornati, sia di sintesi sia di dettaglio.

Ovviamente le finalità sono diverse: gli operatori accedono ai dati per avere un supporto nei loro processi decisionali, i cittadini semplicemente per essere informati.

I produttori di tali dati si trovano di fronte alla necessità di prendere in seria considerazione questa richiesta con soluzioni tecnologiche appropriate.

La relazione illustra come sta affrontando questa esigenza l'Osservatorio Epidemiologico della Regione Lazio.

In particolare mostra come un'applicazione per interrogare schede di dimissione ospedaliera (EPICS) costruita in ambiente SAS/AF per utenti in rete locale sia stata trasferita in ambiente INTERNET conservando molte delle originali funzioni, semplicemente modificando l'interfaccia grafica attraverso la scrittura di pagine HTML/Javascript e mantenendo il motore di elaborazione (SAS/SCL) in sostanza intatto.

2. L'ambiente di riferimento

L'Osservatorio Epidemiologico della Regione Lazio ha fra i vari compiti quello di gestire il Sistema Informativo Ospedaliero, il Sistema Informativo dell'Assistenza Specialistica ed il Sistema Informativo dell'Emergenza Sanitaria.

Per adempiere queste finalità deve poter contare su:

- un efficiente Sistema Informativo;
- una infrastruttura di rete che consenta ai propri tecnici (medici, epidemiologi, ecc.) l'accesso alle informazioni;
- un sistema che consenta di effettuare interrogazioni di estremo dettaglio in modo guidato con dispositivi di controllo sugli accessi;
- un sistema di elaborazione analitica (OLAP).

Finora queste funzioni si sono limitate al personale interno connesso ai dati in rete locale.

Il sistema informativo è attualmente gestito da Oracle, le interrogazioni di dettaglio da un applicativo sviluppato in SAS/AF (EPICS), il Data Warehouse è stato costruito con SAS/MDDDB e le elaborazioni analitiche sono gestite da SAS/EIS. Il Server è Unix ed i Client utilizzano l'ambiente Windows. In particolare EPICS lavora in Client/Server utilizzando il Client per la fase di presentazione ed il Server sia come Data Server sia come Compute Server.

Questa organizzazione ha come inevitabile conseguenza qualche aggravio di gestione, nel senso che si rende necessaria una predisposizione dell'ambiente Client per ogni nuovo utente e, in caso di modifiche all'interfaccia, relativamente frequenti in fase di sperimentazione del nuovo sistema, si deve procedere ad una reinstallazione del sistema; per contro, gli utenti di EPICS devono essere connessi in rete locale ed avere SAS installato sul proprio PC. Questo doppio vincolo limita la diffusione al di fuori dell'ambiente dell'Osservatorio verso altri utenti interessati all'applicazione.

Diversa è la situazione della componente OLAP. In tal caso, infatti, è possibile l'utilizzo anche da utenti connessi via Internet, a parità di funzioni rispetto a quelli locali.

Il problema, in un certo senso la sfida, per far utilizzare EPICS in ambiente Internet era di riutilizzare le migliaia di righe di codice scritte in SCL senza dover installare il sistema SAS sulla macchina Client, lasciando il più possibile inalterate le funzionalità non solo applicative ma anche di controllo sugli utenti dell'applicazione!

3. Le alternative tecnologiche

Alcune caratteristiche attuali di Epics relative all'ambiente Client/Server si possono riassumere secondo il seguente schema:

- il Server è sfruttato sia nell'elaborazione dei dati sia come contenitore delle informazioni necessarie all'utente. Scomponiamo le funzioni del Server in:
 1. "Compute Server", nel senso che si incarica dell'esecuzione dell'interrogazione generata a seguito delle scelte operate dall'utente. Tali scelte sono scritte in

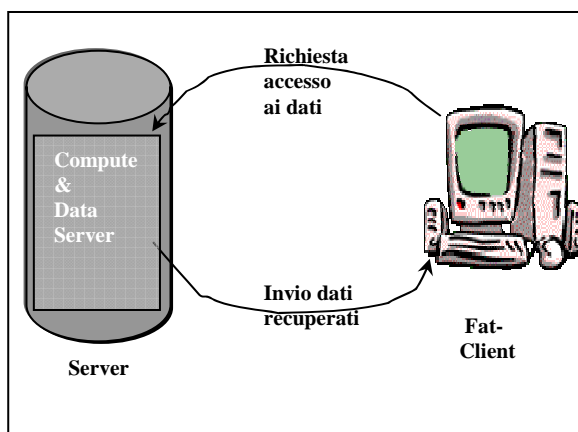
codice SAS, elaborato dinamicamente durante una sessione di lavoro.

2. "Data Server" giacché contiene i dati da elaborare, le interrogazioni salvate in sessioni precedenti ed i dati risultanti da esse.

➤ il Client svolge funzioni di:

1. "Presentazione" utilizzando l'ambiente Frame e il codice SCL che lo gestisce. La grafica fornita è quella di un linguaggio a finestre classico.
2. "Elaborazione" utilizzando codice SCL, ad esempio nella gestione dinamica delle liste di selezione che permettono all'utente di creare la propria interrogazione.
3. "Memorizzazione" di dati di utilità e dello "stato" temporaneo dell'interrogazione, cioè quali scelte ha effettuato l'utente fino a quel momento. L'utente ha inoltre la possibilità di salvare il risultato della propria richiesta in tabelle SAS, richiamabili da una sessione SAS indipendente da quelle in cui si esegue l'applicativo.

Si parla, in questo caso, di Fat Client; ossia una macchina su cui devono necessariamente essere installati tutti i moduli SAS utilizzati dall'applicativo, nonché l'applicativo stesso. I tempi di risposta dipendono dalla configurazione della macchina Server e dalla rete di comunicazione fra Client e Server. In genere le due macchine sono inserite all'interno di una rete non estesa geograficamente.



Date queste premesse, le scelte tecnologiche fra cui scegliere sono state le seguenti:

1. utilizzo di linguaggio HTML/Javascript,
2. utilizzo di AppDev Studio per produrre applet Java,

3. utilizzo di AppDev Studio per produrre un'applicazione Java.

Vediamo di esaminare con maggior dettaglio vantaggi e limiti di ognuna di queste soluzioni, per comprendere meglio la filosofia adottata.

E' importante sottolineare come qualsiasi scelta si fosse preferita, era necessario mantenere la struttura Client/Server come fondamento all'applicazione.

3.1 HTML/Javascript

➤ Il Server è utilizzato come:

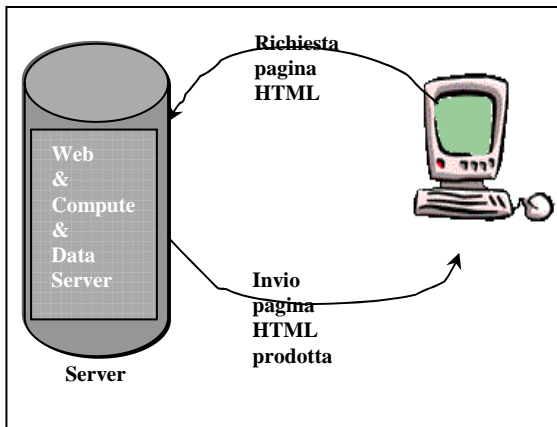
1. "Web Server" per restituire all'utente le pagine HTML richieste. La porzione di codice che implica l'elaborazione di codice SAS è gestita da SAS/INTRNET.
2. "Compute Server" per elaborare lo stesso codice SAS su cui poggia Epics ed il codice SCL di gestione degli oggetti di selezione. Questa fase in Epics era delegata alla macchina Client.
3. "Data Server" secondo le modalità già descritte per Epics.

➤ Il Client svolge funzioni di:

1. "Presentazione" utilizzando un comune Browser (Netscape Navigator, Microsoft Internet Explorer, etc.) per la visualizzazione delle pagine HTML.

Caratteristiche di questa soluzione:

- ❑ Thin-Client, in quanto sulla macchina dell'utente è sufficiente la presenza di un Browser, permettendo quindi una connessione attraverso una rete geografica più estesa eliminando i problemi legati all'installazione di versioni aggiornate dell'applicativo.
- ❑ Riutilizzo di tutto il codice scritto ad esclusione delle schermate grafiche di presentazione, che sono riviste secondo la potenza grafica fornita dall'ambiente HTML, migliorando in taluni casi l'aspetto esteriore, rendendolo più gradevole ed integrato con le scelte grafiche societarie.
- ❑ Performance dipendente dalle condizioni di connessione, in generale peggiori che nel caso di Epics standard.
- ❑ Non può servire un gran numero di utenti perché appesantisce la parte di elaborazione da parte della macchina Server, costretta ad eseguire un'istanza di sessione SAS ad ogni connessione.
- ❑ Non è possibile salvare in locale i data set eventualmente prodotti. Si appoggia totalmente al Server ed alla sua configurazione.



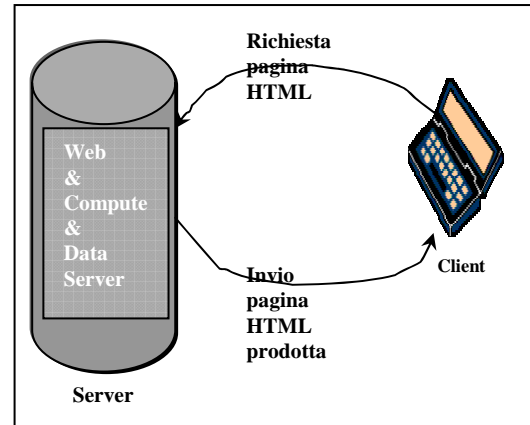
3.2 AppDev Studio con produzione di applet

- il Server è utilizzato come:
 1. "Web Server" per l'invio della pagina HTML richiesta, le parti di comunicazione con l'ambiente SAS sono gestite da AppDev Studio.
 2. "Compute Server" elabora lo stesso codice SAS di Epics, riutilizza una minima parte del codice SCL che in EPICS era eseguito sulla parte Client dell'architettura. La presentazione deve essere riscritta mediante il linguaggio HTML, introducendo il riferimento all'applet costruita.
 4. "Data Server" secondo le modalità già descritte per Epics.
- il Client svolge funzioni di:
 1. "Presentazione" utilizzando un comune Browser abilitato alla gestione di applet Java.
 2. "Elaborazione" utilizzando le funzioni dell'applet, scritta in linguaggio Java.

Caratteristiche:

- ❑ Rappresenta una via di mezzo tra un Fat-Client ed un Thin-Client, sfruttando le caratteristiche di indipendenza dall'ambiente operativo di sviluppo di Java.
- ❑ Una notevole parte del codice SCL non è più riutilizzabile, nonostante la possibilità di inserire classi di oggetti specifiche prodotte dal SAS Institute, oltre alla necessità di dover ridefinire la parte di "Presentazione". Necessita di una buona conoscenza del linguaggio Java.
- ❑ Alti tempi di trasferimento iniziale dell'applet, dato che le sue dimensioni in termini di byte possono raggiungere livelli ragguardevoli.
- ❑ Velocità di utilizzo dell'interfaccia una volta che l'applet è stata scaricata, i tempi di attesa rimangono quelli di restituzione dei risultati.

- ❑ Non è possibile utilizzare la macchina in locale per salvare i data set eventualmente prodotti, a meno di non superare, mediante accorgimenti particolari, i dettami di sicurezza che sono considerati la punta di diamante del linguaggio Java.

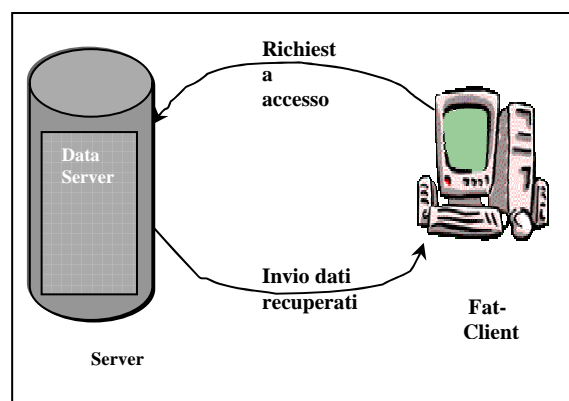


3.3 AppDev Studio con produzione di applicazioni

- il server è utilizzato esclusivamente come:
 1. "Data server" secondo le modalità già descritte per Epics.
- il Client svolge funzioni di:
 1. "Presentazione" utilizzando le modalità grafiche presenti nel linguaggio Java.
 2. "Elaborazione" utilizzando la Virtual Machine necessaria per eseguire un'applicazione scritta in linguaggio Java.

Caratteristiche:

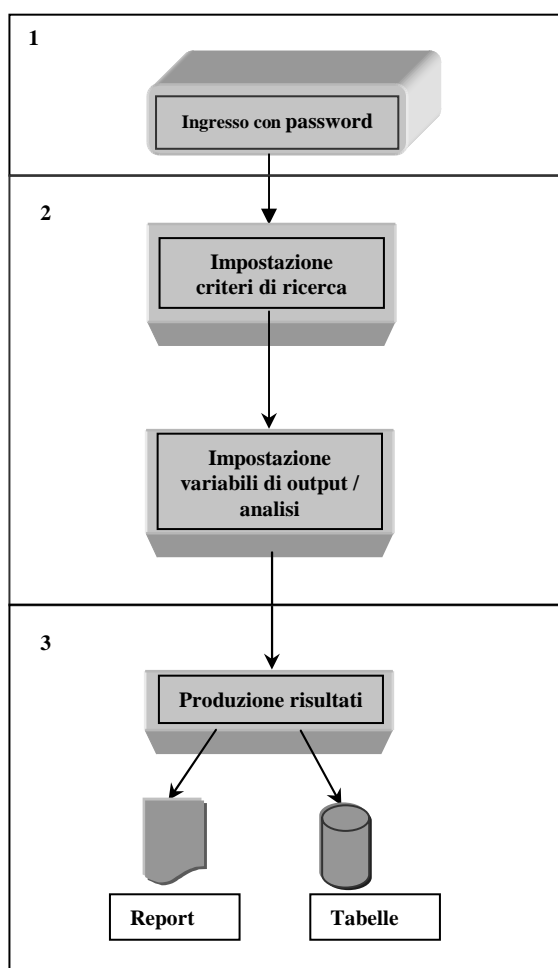
- ❑ Si ritorna ad una configurazione come Fat-Client.
- ❑ Il codice SAS ed SCL scritto per l'applicazione originaria è irrimediabilmente perso, in quanto è necessario riscrivere tutto in linguaggio Java.
- ❑ E' necessario installare presso l'utente la nuova applicazione e alcune componenti fondamentali per Java, quali la Virtual Machine.



4. Architettura della soluzione adottata

Studiate le alternative tecnologiche e l'ambiente di riferimento, è stata preferita la soluzione HTML/Javascript, in quanto era l'unica a permettere il massimo riutilizzo del codice, senza dover provvedere alla manutenzione delle macchine Client eventualmente collegate. Naturalmente questa scelta è stata dettata anche dal basso numero di utenti in accesso contemporaneo alla risorsa Web. Radicalmente diversa sarebbe stata la scelta nel caso in cui ci si fosse dovuti apprestare allo sviluppo di una soluzione Internet completamente nuova. Molto probabilmente, visto il panorama tecnologico ed il sistema informatico a disposizione, ci si sarebbe orientati per la soluzione HTML/Applet Java che permette di gestire la connessione di un maggior numero di utenti con un aggravio del server minimo. La fase di progettazione dell'applicativo nel nuovo ambiente ha avuto come obiettivo la concezione di codice il più possibile indipendente dal sistema operativo, proprio secondo lo spirito tradizionale che ha permesso alla rete di evolversi in brevissimo tempo ed efficacemente.

Il flusso procedurale della nuova applicazione si può scomporre secondo i seguenti moduli:

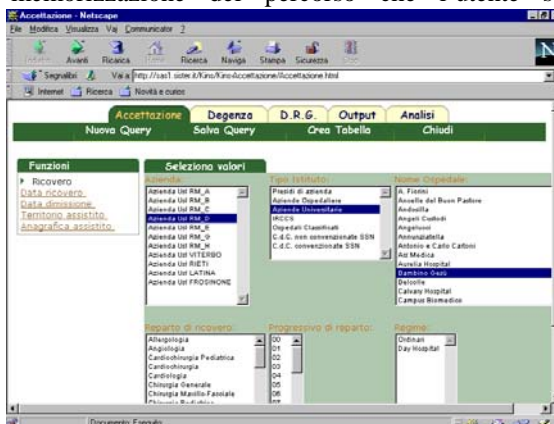


Innanzitutto si è provveduto a separare l'interfaccia dal motore di elaborazione vero e proprio. Le possibilità grafiche messe a disposizione dalla tecnologia Web sono sotto gli occhi di tutti nel momento in cui si decide di navigare sulla rete. Proprio per sottolineare la vicinanza dell'applicazione alle pagine Web classiche, è stato deciso di utilizzare suggerimenti grafici già presenti in ambienti di rete e si è passati al disegno della nuova interfaccia per Epics.

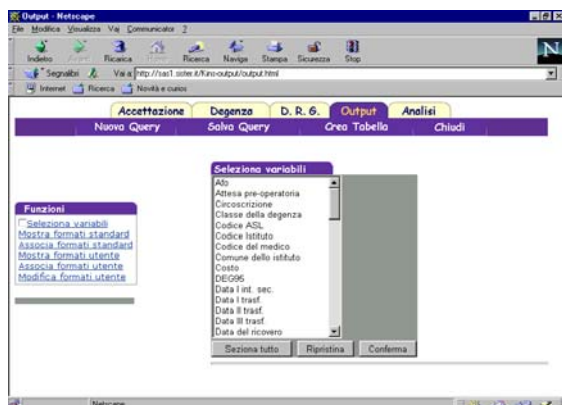
L'implementazione della fase 1 (Accesso con password) ha tenuto conto delle problematiche di sicurezza che da sempre alimentano dibattiti fra i sostenitori ed i detrattori della rete. Nel nostro caso era fondamentale mantenere la possibilità di accesso alle informazioni in dipendenza del livello di accesso dell'utente abilitato. Si è quindi provveduto a riutilizzare il database degli accessi presente nella versione originale, aggiungendo un livello di sicurezza ulteriore che consiste nell'assegnare alla sessione in corso un identificativo unico, non riproducibile o riutilizzabile da utenti cosiddetti malvagi in momenti immediatamente successivi all'abilitazione.

Una volta all'interno del modulo di creazione della query SQL (fase 2), l'utente ha la possibilità di selezionare i valori di interrogazione mediante liste di selezione, oppure di cambiare pagine di selezione mediante i consueti link HTML. La particolarità delle pagine che sono visualizzate è la loro creazione dinamica mediante il codice SCL che gestiva le FRAME SAS in ambiente AF. In altre parole, ogni volta che l'utente segue un link di navigazione HTML, l'applicazione invia un messaggio al server SAS, che controlla l'abilitazione dell'utente e costruisce la pagina di risposta con i nuovi valori. Le liste stesse di selezione sono alimentate dagli stessi data set SAS che già alimentavano gli oggetti ListBox dell'ambiente SAS/AF.

I criteri selezionati via via sono salvati in una directory temporanea creata sul server al momento dell'abilitazione all'accesso ai dati. Questa memorizzazione del percorso che l'utente sta



seguito nella generazione dell'interrogazione, del tutto trasparente, permette di salvare, a richiesta, lo stato dell'interrogazione in qualsiasi punto ci si trovi.



La terza fase è quella che differenzia maggiormente l'applicativo originale da quello realizzato via Web. Infatti, per la logica stessa su cui regge la filosofia Web, tutte le operazioni di calcolo, accesso o salvataggio dati avvengono sulla macchina server. Di conseguenza, la possibilità di salvare le query prodotte dall'utente o di creare dei data set SAS dai risultati ottenuti da una query può essere concessa solo a livello di server e non di macchina locale. E' quindi necessario predisporre sulla macchina server delle directory legate agli utenti abilitati e deve essere permesso loro di poter eventualmente accedere, anche via FTP (File Transfer Protocol), a tali directory per scaricare in locale i dati set costruiti. Rimane invece inalterata la possibilità di scaricare i dati in formato Excel a seguito di una richiesta specifica dell'utente.

Questo fa sì che si ponga molta attenzione nel dimensionamento della macchina Server e pone l'accento sull'importanza dell'amministratore di sistema che deve essere pronto ad abilitare o disabilitare gli utenti secondo necessità. In futuro è prevedibile l'utilizzo di forme di abilitazione on-line.

5. Limiti funzionali della soluzione adottata rispetto ad Epics

Alcune funzioni non sono più disponibili, in quanto sul Client NON è installato SAS. E' questo il caso del salvataggio in locale del data set risultante dall'esecuzione di un'interrogazione, è però sempre possibile esportare tale risultato in tabelle Excel locali. Il codice SQL e lo stato della query possono essere memorizzate solo ed esclusivamente sul server e possono essere poi recuperati dall'applicazione. Come già detto in precedenza, questo richiede che la macchina server sia configurata con ambienti distinti per utente.

Le uscite delle query possono essere memorizzate sul server e scaricate a richiesta con funzioni di

FTP, nel caso in cui comunque l'utente abbia la possibilità di elaborarle in proprio con SAS System. L'accesso con password ed il log degli accessi restano attivi, il livello di sicurezza dipende da fattori sia hardware sia software legati a scelte aziendali (uso di Firewall, certificati per il riconoscimento, crittografia, etc.).

Le performance sono in generale peggiori, in quanto ogni selezione effettuata dall'utente provoca l'invio di una richiesta al Web Server che deve costruire la nuova pagina Web e restituirla al Client che ne ha fatto domanda.

Con la soluzione scelta non possono essere collegati molti utenti contemporaneamente perché debbono partire troppe sessioni SAS. Questo non è in contraddizione con la necessità di aprirsi al mondo Web, in quanto l'applicativo stesso ha come bacino di utenza un numero limitato di persone con conoscenze specifiche. Si vuole cioè dare agli utenti la possibilità di interrogare liberamente un database per estrarre informazioni con le stesse funzionalità ed i controlli di una sessione utente locale. Se lo scopo, invece, è quello di fornire agli utenti report pre-elaborati, si possono collegare all'applicativo gli altri strumenti classici di SAS/Intranet, che non pongono limiti teorici al numero di utenti collegati.

6. Conclusioni

Le novità tecnologiche sono davvero molte e soprattutto in rapida evoluzione, tanto che gli utenti sono invogliati a percorrere una nuova strada, pur avendo poche informazioni sull'effettiva capacità di mantenere un alto livello qualitativo del servizio prodotto. Qualsiasi azienda che desideri sfruttare le potenzialità offerte dal mondo Web deve procedere ad un'attenta analisi delle proprie esigenze, che non possono in alcun modo degradare le prestazioni da fornire all'utente. Per questo motivo abbiamo preferito utilizzare la tecnica dei piccoli passi nel migrare da un'applicazione completamente SAS ad una completamente Java-oriented. Siamo in una fase intermedia che ha permesso un rapido sviluppo della nuova applicazione e che faciliterà l'eventuale volontà di convertire il prodotto in Full Java, riutilizzando sempre parte di codice già scritto. In questo modo si speriamo di riuscire a diminuire la pessima abitudine di buttare via il vecchio per sostituirlo con un nuovo ancora traballante o incerto sulle proprie possibilità. Fortunatamente questo è possibile se si utilizzano strumenti quali SAS System, in quanto già prevedono una portabilità pressoché totale sui più comuni ambienti di sviluppo.

7. E-mail degli autori

Vittoria Liguori: v.liguori@sister.it

Claudio Grego: grego@sirio.regione.lazio

