



Università degli studi di Padova
Statistiche

Facoltà di Scienze

**IL PASSAGGIO DEL SISTEMA INFORMATIVO SANITARIO
DELL'U.L.SS. N° 4 ALTO VICENTINO
IN AMBIENTE WEB-BASED**

Relatore: Ch.mo Prof. Dulli Susi

Laureando: Idelfo Borgo

ANNO ACCADEMICO 2005-2006

INDICE

INDICE	1
1.1 IL CONTESTO DI RIFERIMENTO	2
2. LA PIATTAFORMA APPLICATIVA	6
2.1 INTRODUZIONE.....	6
2.2 OVERVIEW DELL'ARCHITETTURA : MODELLING	8
2.3 OVERVIEW DELL'ARCHITETTURA : TAILORING	17
2.4 STANDARD	19
2.5 ARCHITETTURA FUNZIONALE	22
2.6 ALERT - AUDIT.....	24
2.7 VALORI AGGIUNTI	30
3. LA PIATTAFORMA TECNOLOGICA	34
3.1 I SISTEMI RDBMS : DAL RDBMS A OODBMS	34
3.2 CACHE'.....	36
3.3 COMPONENTI DI BASE DI CACHE'.....	41
4. REPORT WEB XML	44
4.1 CHE COSA SONO XSLT E XSL-FO?	45
4.2 ARCHITETTURA	45
4.3 SOLUZIONE	46
4.4 ESEMPIO	51
5. CONCLUSIONI	56
5.1 ALTA AFFIDABILITA'.....	56
5.2 ...CONCLUDENDO.....	59
BIBLIOGRAFIA.....	60

1. INTRODUZIONE

1.1 IL CONTESTO DI RIFERIMENTO

L'Azienda U.L.SS. n° 4 – Alto Vicentino, nel corso dell'anno 2001, ha scelto di dotarsi di un sistema informativo integrato per la gestione delle attività cliniche, diagnostiche, amministrative ed autorizzative del complesso delle prestazioni dell'Azienda stessa.

La trasformazione in atto in campo sanitario secondo modelli di tipo aziendale ed in particolare i meccanismi di finanziamento basati sulle prestazioni erogate e la necessità di raffrontarsi con un mercato delle prestazioni sanitarie, richiedono che il sistema consenta un effettivo controllo in tempo reale dei processi produttivi. Questo deve quindi assicurare che l'erogazione delle prestazioni e delle cure mediche venga supportata da una visione complessiva dell'intero processo assistenziale, incentrato sull'utente, in tutte le sue connotazioni.

Per conseguire tali obiettivi occorre che ogni singolo atto che coinvolge il paziente venga registrato alla fonte insieme a tutte le sue caratteristiche (tipologia dell'atto, chi lo ha eseguito, quando, in che struttura, con quali risultati, ecc.). L'insieme ordinato di questi atti (o transazioni) elementari deve costituire, insieme con gli atti "precedenti" relativi allo stesso paziente, il supporto del processo clinico e di quello amministrativo che si devono estrinsecare in un completo rendiconto delle attività sostenute dalla struttura per ogni singolo contatto con l'assistito.

Le esigenze sopra riportate, si sono consolidate nel corso dell'anno 2001, con l'espletamento di una gara, volta ad individuare un software che permettesse di gestire tali attività.

La scelta, è caduta su un sistema informativo integrato, sviluppato in Australia, ma commercializzato (grazie alla sua parametricità) in tutto il mondo, la denominazione commerciale di tale sistema è MEDTRAK che nella sua ultima release (descritta nel capitolo successivo) viene denominato TRAKCARE.

I motivi di questa scelta, sono stati essenzialmente condizionati dai seguenti fattori :

- tra le offerte presentate era l'unico vero sistema informativo integrato;
- era l'unica offerta che presentava una parametricità tale da poter garantire il rispetto del D.p.r. 318;
- l'utilizzo del Database cachè, benché non uno standard, nel panorama informatico delle pubbliche amministrazioni, ha permesso di rimandare l'acquisto di hardware più potente (scelta necessaria

per le altre offerte), nonché il garantire la riduzione dell'attività di Database Management.

MEDTRAK

La piattaforma applicativa MedTrak è un sistema di informatizzazione Windows-based incentrato sul paziente che, grazie alla sua tecnologia informatica, consente una gestione efficiente dei processi clinico-amministrativi della Struttura Sanitaria.

Medtrak è una piattaforma che supera il concetto di integrazione tra moduli e si configura come un sistema E.R.P. per l'ambiente sanitario in grado di assolvere in modo organico alla totalità delle esigenze dell'azienda.

Il sistema, infatti, è in grado di recepire e gestire, in modo integrato e trasparente, i dati riguardanti tutta l'attività di erogazione di prestazioni socio-sanitarie utilizzate e/o utilizzabili da ogni utente facente parte di un'anagrafe di circa 200/250mila assistibili (su una base di 180mila assistiti).

Sostanzialmente ad ogni paziente sono associati tutti gli episodi di contatto con le strutture dell'Azienda sanitaria, siano esse l'ospedale, il distretto, la casa di riposo, ecc., in modo da avere a disposizione l'intera storia "sanitaria" della persona; la perfetta trasparenza delle informazioni coesiste con la selezione e la differenziazione degli accessi degli utilizzatori, abilitabili separatamente per l'alimentazione dei dati, per la consultazione di parte degli stessi o di tutti e per l'elaborazione.

Inoltre, l'interfaccia utente uniforme facilita non poco l'interscambiabilità delle informazioni e gli utilizzatori stessi.

La piattaforma è inoltre munita di un sistema di protezione e di auditing delle singole transazioni, nonché di una tecnologia di archiviazione e di firma elettronica in linea con la normativa italiana.

L'implementazione di tale sistema informativo è stata graduale, in quanto non si trattava di una installazione ex-novo, ma in parte anche di una sostituzione di vecchi sistemi informativi, non sempre integrati.

Allo stato attuale, si stima di aver raggiunto il 50% del piano di installazione presentato nel corso dell'anno 2001.

Tale ritardo nei tempi di implementazione è dovuto principalmente a nuove esigenze di implementazione, che hanno inevitabilmente distolto l'attenzione agli obiettivi iniziali : come ad esempio l'implementazione di un sistema informativo sanitario, centrato non solamente verso l'attività ospedaliera, ma soprattutto verso l'attività territoriale, che negli ultimi anni è stata uno dei principali obiettivi dell'attuale direzione Aziendale.

Dal punto di vista prettamente tecnologico, Medtrak si presenta come un classico applicato a due livelli (Client Server), sviluppato in Visual Basic, lato client, mentre invece lato server viene utilizzato il database Caché; si tratta di un Database Post-Relazionale che risulta molto performante nel trattamento di oggetti e istruzioni SQL e che è in grado di garantire un

livello ottimale di sicurezza sul controllo di voluminose e complesse quantità di dati quali quelli sanitari.

Il dialogo tra client e server, avviene tramite un'apposita ocx messa a disposizione direttamente da Intersystems (il produttore del database). La configurazione lato client, ne risulta notevolmente semplificata, evitando complesse (e spesso invasive) configurazioni del client.

Come già accennato, l'esigenza di implementare l'applicativo in tutto il territorio di competenza dell'Azienda ULSS 4, ha comportato l'introduzione di un'ulteriore livello applicativo, per sopperire la mancanza di linee a banda larga necessarie a garantire un funzionamento dell'applicativo con tempi di risposta adeguati anche per un applicativo client server come Medtrak.

La trasposizione dell'applicativo a due livelli, in un applicativo a tre livelli, è stata realizzata, introducendo nel sistema, la piattaforma Citrix-Metaframe che permette di centralizzare la logica applicativa su degli Application Server, lasciando al client la sola logica di presentazione delle maschere. Questo ha permesso di portare l'applicativo Medtrak anche in zone territoriali disagiate, con la semplice disponibilità di linee ISDN a bassa velocità. In aggiunta a questo, l'architettura Citrix-Metaframe ha portato anche all'innegabile vantaggio di abbattere il TCO del client (total cost of ownership) permettendo l'utilizzo di terminali WBT (Windows Based Terminal).

Spostare la logica applicativa dal client al server ha introdotto alcune problematiche, che sono affrontate nel capitolo conclusivo.

TRAKCARE

Nel corso dell'anno 2005, l'autorizzazione da parte delle regione, alla costruzione dell'unico ospedale (in sostituzione degli attuali presenti a Thiene e a Schio), ha spinto ulteriormente l'implementazione di tale sistema informativo verso il nuovo obiettivo dell'ospedale "Paperless&Filmless".

Tali nuove esigenze hanno portato, alla necessità di migrare l'attuale piattaforma applicativa MEDTRAK, verso la nuova versione TRAKCARE, per permettere una maggior integrazione con ambienti finora scollegati dal sistema informativo (strumentazione elettromedicale e non, sistemi informativi relativi a diagnostica strumentale ecc.) o poco integrati (sistemi informativi verticali costruiti su specifiche esigenze come apparecchiature di monitoraggio di parametri clinici vitali); fino alla facilitazione del deployment della stessa, grazie all'interfaccia WEB, verso MMG o strutture che collaborano con l'Azienda (Case di Riposo e/o Comuni).

Nei capitoli successivi, vengono illustrate alcune caratteristiche peculiari dell'applicativo TRAKCARE, nonché le problematiche (prevalentemente

tecnologiche) che l'Azienda ULSS 4 deve e dovrà affrontare per tale cambio tecnologico.

I due ambienti applicativi Medtrak e Trakcare, sono perfettamente integrati tra di loro e possono funzionare in parallelo; questo, grazie alla peculiarità del database cachè che permette la visione della base dati unica, in formati differenti (relazionale ed ad oggetti) allo stesso tempo. Si prevede perciò che l'introduzione di Trakcare, avverrà per i nuovi moduli¹, mentre i moduli già implementati in Medtrak continueranno a funzionare nel consueto ambiente. Essendo la base di dati unica ed integrata, qualsiasi episodio² inserito in Medtrak sarà visibile ed eventualmente integrabile nell'interfaccia Trakcare e viceversa.

¹ Per modulo si intende, una specifica attività svolta da un servizio (sia sanitario che amministrativo), come ad esempio il modulo "Gestione Sale Operatorie" o il modulo di "Gestione Vaccinazioni"

² Per episodio, si intende qualsiasi attività venga fatta sul paziente : prenotazione, ricovero, vaccinazione ecc.

2. LA PIATTAFORMA APPLICATIVA

2.1 INTRODUZIONE

TrakCare è un Sistema Informativo Ospedaliero integrato, completamente riscritto con moderne tecnologie di progettazione web based.

Oltre a questo aspetto prettamente tecnologico, TrakCare rispetto alla versione Client Server in Visual Basic (MedTrak), introduce un'importante novità: è un sistema basato sui processi sanitari e non più secondo una logica di moduli software. Al WORKFLOW MANAGER è affidato il compito di disegnare ed implementare i processi ospedalieri.

E' un sistema completamente Enterprise Resource Planning: tutte le funzioni dell'architettura si compongono per realizzare i processi aziendali. TrakCare è progettato con una visione olistica³ e integrata, riconciliando le diverse esigenze dei singoli reparti e dell'organizzazione ospedaliera nel suo insieme per un miglioramento globale dell'efficacia e dell'affidabilità.

E' un sistema scalare e quindi può essere implementato completamente o essere integrato ad altri moduli preesistenti in azienda.

TrakCare offre la scelta tra un modello organizzativo accentrato o decentrato, per consentire la migliore cooperazione tra reparti per l'ottimizzazione delle risorse; rende disponibili informazioni clinico-sanitarie della massima precisione a tutto il personale sanitario in ogni luogo ed in qualunque momento. TrakCare aiuta a creare il miglior modello organizzativo per la migliore cura del paziente.

E' una soluzione completa sia per le articolate funzioni applicative oggi disponibili che grazie al potente ambiente di disegno e gestione dei processi integrato nell'architettura.

Il risultato è un sistema completamente configurabile per funzione, reparto o per utente. Questo è il presupposto per creare soluzioni cliniche aderenti a qualunque specificità.

Nella figura 1, sono riportati solo alcuni degli ambienti di lavoro che sono coperti dalle funzionalità. Si parla di ambiente di lavoro in quanto ognuno di questi viene costruito in funzione dell'organizzazione. Allo stato dell'arte non tutti gli ambienti di lavoro sono attivati nell'Azienda ULSS 4 :

- Pronto Soccorso
- Laboratorio di Analisi
- Centro Trasfusionale
- Anatomia Patologica

sono moduli trattati esternamente al Sistema Informativo Medtrak/TrakCare, anche se integrati con esso. Il fatto di essere trattati

³ La visione olistica si propone di interpretare i sistemi complessi dividendoli nelle loro componenti e studiandone separatamente le proprietà

separatamente, è dovuto principalmente al fatto che tali moduli, sono stati attivati molto tempo prima dell'acquisizione dell'infrastruttura Medtrak/Trakcare, la loro riconversione nel nuovo ambiente sarebbe un costo eccessivo (dal punto di vista organizzativo); ciò non toglie che alla fine del processo di implementazione del nuovo Sistema Informativo Aziendale (S.I.A.), non si prenderà in considerazione l'opportunità di convertire anch'essi.

I sottostanti moduli, inoltre, non sono esaustivi per l'intero S.I.A., in realtà, nella figura sono rappresentati i principali moduli per i flussi ospedalieri, mentre nella realtà dell'Azienda ULSS 4 c'è una forte espansione nell'attività territoriale.

Processi Ospedalieri – Ambienti di lavoro							
Esterni	Pronto Soccorso/DEA	Lista d'attesa	Prenotazione	Accettazione	Visite	Day Hospital	Fatturazione Cassa
Ricovero	Lista d'attesa	ADT	Reparti e posti letto	Programmaz. Attività medica	Programmaz. Attività inferm.	Day Hospital	Fatturazione DRG
Diagnosi-Terapie	Lab. Analisi, Anat. Patol.	Radiologia e immagini	Medicina nucleare	Trasfusionale	Terapia Intensiva	Chirurgia, Sale Operatorie	Emodinamica
Area clinica	Visite e procedure	Cartella Clinica	Analisi diagnostiche	Ordini e refertazione	Protocolli di cura	Ricerca scientifica	Discussione casi clinici
Servizi comuni	Magazzino	Farmacia	Pasti e diete	Centro sterile	Archivio cartaceo	Logistica	Ingegneria clinica

Figura 1 – Ambienti di lavoro relativi ai principali processi ospedalieri

Per ogni ambiente di lavoro vengono definiti e programmati i processi (tramite il WORKFLOW MANAGER), segue la modellizzazione dell'interfaccia utente per processo e funzione (LAYOUT MANAGER), sono definite le regole cliniche di compatibilità come allarmi per soglie critiche o per incompatibilità farmacologiche (CLINICAL DSS), i protocolli (CLINICAL PATHWAY), ecc..

L'architettura TrakCare rende la tradizionale gestione a moduli software un concetto obsoleto e completamente superato, sia dal punto di vista organizzativo che informatico.

TrakCare rappresenta un Sistema Informativo Sanitario completo, flessibile ed organizzato. Ad esempio, esaminando i processi ospedalieri, possiamo schematicamente elencare:

- processi per Esterni e Pronto Soccorso.
- processi di Ricovero.
- processi Diagnostico-Terapeutici.
- processi dell'area Clinica.
- processi relativi ai Servizi comuni (magazzino, logistica, farmacie, ecc.).

Le caratteristiche dominanti del sistema sono:

Peculiarità Architetture:

- Costruito completamente su architettura workflow.
- Alta configurabilità.
- Performante e sicuro.
- Modulare e scalabile.
- Piattaforma applicativa completamente integrata.
- User Interface a livelli funzionali.

Interfaccia Utente:

- Interfaccia WEB con visualizzazione in Base Browser.
- Display grafico.
- Pagine e menu definibili a livello di utenti e gruppi di utenti.
- Uso di icone.
- Alerts multilivello.
- Workflows / data entry strutturato.
- Multilingua
- Gestione nativa di computer palmari PDA in wireless.

Informazioni condivise:

- Disegnato come Sistema Globale
- Flessibile e facilmente integrabile con altri sistemi
- Disegnato sugli Industry Standard Sanitari, HL7, DICOM, in modo nativo nell'architettura
- Implementato con il nuovo Standard XML
- Sistemi diagnostici integrati in nativo
- Alert by Paging & SMS
- Integrato con Email and Fax

Peculiarità Tecnologiche:

- Open System
- Indipendente dalla piattaforma
- Architettura 3-tier
- Livelli di Security
- Clustering e Shadowing

2.2 OVERVIEW DELL'ARCHITETTURA : MODELLING

L'architettura di MedTrak nella sua ultima versione "TrakCare" è stata progettata per essere modellata su qualunque HIS (Healthcare

Information System), mettendo a disposizione strumenti di configurazione secondo una logica ERP.

Di seguito vengono riportate alcune componenti fondamentali che conferiscono a TrakCare la qualifica, internazionalmente riconosciuta, di sistema completamente configurabile.

Nella figura 2 vengono rappresentate in color rosa le principali sottocomponenti del sistema che, come si evince dal diagramma, sono trasversali a tutte le aree applicative. Ognuna di esse non solo concorre alla modellazione del sistema ma coopera con le altre. Ad esempio: avendo definito delle regole di compatibilità farmacologia nel CLINICAL DSS, queste interagiscono sia nell'ambiente di ORDER MANAGER (dove vengono richieste le prestazioni per il paziente) sia, ad esempio, nella programmazione di attività che vengono gestite dal CUP, sia nel CLINICAL PATHWAY dove vengono programmati e gestiti i protocolli di cura.



Figura 2 – Sottocomponenti del sistema

Workflow manager

WORKFLOW MANAGER è sicuramente uno degli ambienti più innovativi: consente la configurazione dei processi.

L'ospedale è un'organizzazione complessa fatta da processi definiti ma mutevoli. Qualunque ambito lavorativo deve quindi rispondere ad un principio di organizzazione e controllo e non a logiche statiche di programmazione. La potenza del sistema si fonda quindi sulla capacità di comporre processi aziendali partendo dalla selezione di attività elementari. Ad ogni attività vengono successivamente attribuite le caratteristiche del processo per ogni ambito di lavoro (workbench). Definito quindi il workbench, si passa alla costruzione del processo vero e proprio dove vengono correlate le attività e i criteri di cooperazione del workflow.

WORKFLOW MANAGER consente di definire l'appropriato workflow, per ogni processo principale, attraverso la definizione delle correlazioni fra le componenti software, garantendo il monitoraggio dei flussi di lavoro.

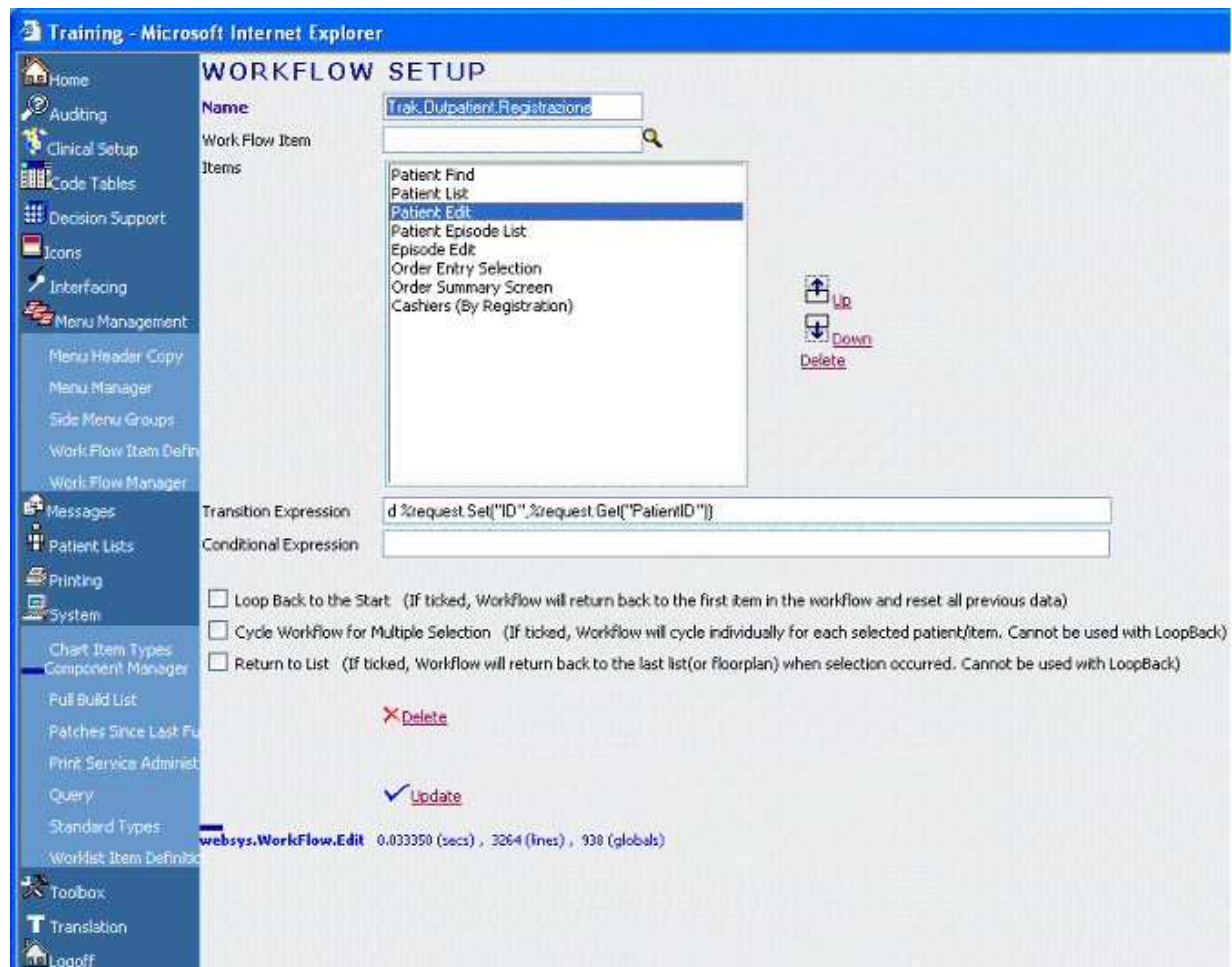


Figura 3 – Workflow setup

Order manager

Principalmente utilizzato in ambito clinico sanitario, ORDER MANAGER consente di definire i criteri di richiesta di prestazioni per tutta l'azienda, per ciascun reparto, per operatore.

E' basato su un criterio universale: le codifiche aziendali centralizzate. Queste rappresentano tutto l'universo delle prestazioni che possono essere richieste ed erogate da qualsiasi operatore o utente dell'ospedale. Essendo l'ambiente TrakCare completamente integrato, è possibile stabilire centralmente le politiche organizzative e cliniche del processo di ordini. E' evidente che la complessità nel mondo operativo ospedaliero risiede proprio nel rapporto "molti a molti" ("n a m") dei criteri e diritti da definire. Per facilitare tale definizione è disponibile un ambiente di lavoro che imposta e controlla queste politiche.

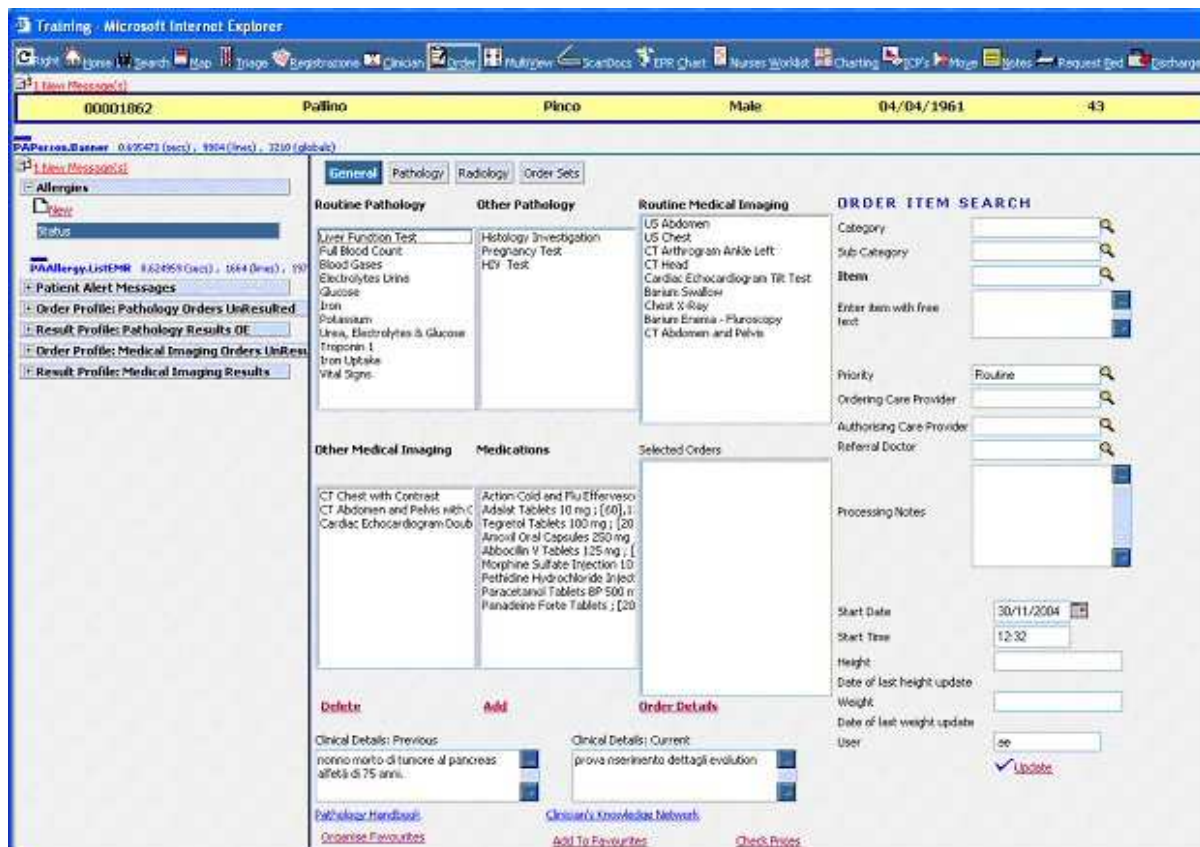


Figura 4 – Order manager

Altro aspetto fondamentale, derivante da questo modo di progettare, è la possibilità di impostare il controllo di gestione. Infatti, ad ogni prestazione è associato uno o più listini. Tra questi è possibile identificarne uno predefinito come costo standard della singola prestazione. Grazie quindi al corretto utilizzo del sistema è possibile arrivare a capire quali prestazioni sono state erogate ma anche quali costi sono stati sostenuti per ogni singolo ricovero. Si arriva pertanto a quella che generalmente viene definita come una contabilità per “Commessa”.

Prestazioni	Reparto Esec.	Costo std.	Peso
Accettazione Medica	Cardiologia	90	0,9
Accettazione Medica	Medicina II - Accettazione medica	70	2
Accettazione Medica		50	1,2
Attività infermieristica reparto	Cardiologia	110	1,2
Attività infermieristica reparto	Diabetologia	40	2
Attività infermieristica reparto	Neurologia	80	1
Emocromo	Laboratorio analisi	5,7	1
Emocromo		3	0
Glicemia	Laboratorio analisi	4,75	1
Glicemia		5,15	0
TAC del cranio		300	3

Figura 5 – Impostazione dei costi standard

Inoltre, prestazioni e relativi costi possono essere raggruppati in base a due criteri: da un lato la patient location, ossia il reparto al quale il paziente è assegnato, dall'altro la receiving location, cioè dove la prestazione viene erogata. In questo modo possono essere controllate sia le prestazioni erogate per reparto richiedente che l'attività svolta, ad esempio, da un servizio o da un reparto per visite di consulenza.

Come prima specificato, ORDER MANAGER interagisce con il sistema delle regole di compatibilità descritte tramite CLINICAL DSS. Grazie a questo metodo clinico/informatico, è realmente possibile diminuire drasticamente gli errori umani nel processo di cura.

Worklist manager

Ulteriore innovativo strumento di modellazione disponibile con TrakCare è WORKLIST MANAGER che consente di creare comode liste di lavoro, in ogni ambito aziendale, aderendo ai processi lavorativi pertinenti ad ogni operatore (ad esempio: tutte le prestazioni di un certo reparto ancora da refertare). Il sistema quindi è un ambiente di programmazione del lavoro e monitoraggio di tutti i processi. La gestione delle worklist consente di bilanciare al meglio l'organizzazione secondo un criterio di assorbimento del carico di lavoro realmente sostenuto. Uno dei valori aggiunti derivanti è la possibilità di identificare sia i colli di bottiglia sia il costo sostenuto per la loro mancata eliminazione e riduzione.

WORKLIST MANAGER si complementa come ambiente di gestione al WORKFLOW MANAGER.

Nel caso di richieste, ad esempio per la radiologia, è possibile vedere in ogni momento quali siano quelle attive, quale "stadio" del processo abbiano raggiunto, ed eventualmente quali ritardi si siano verificati ed in quale fase.

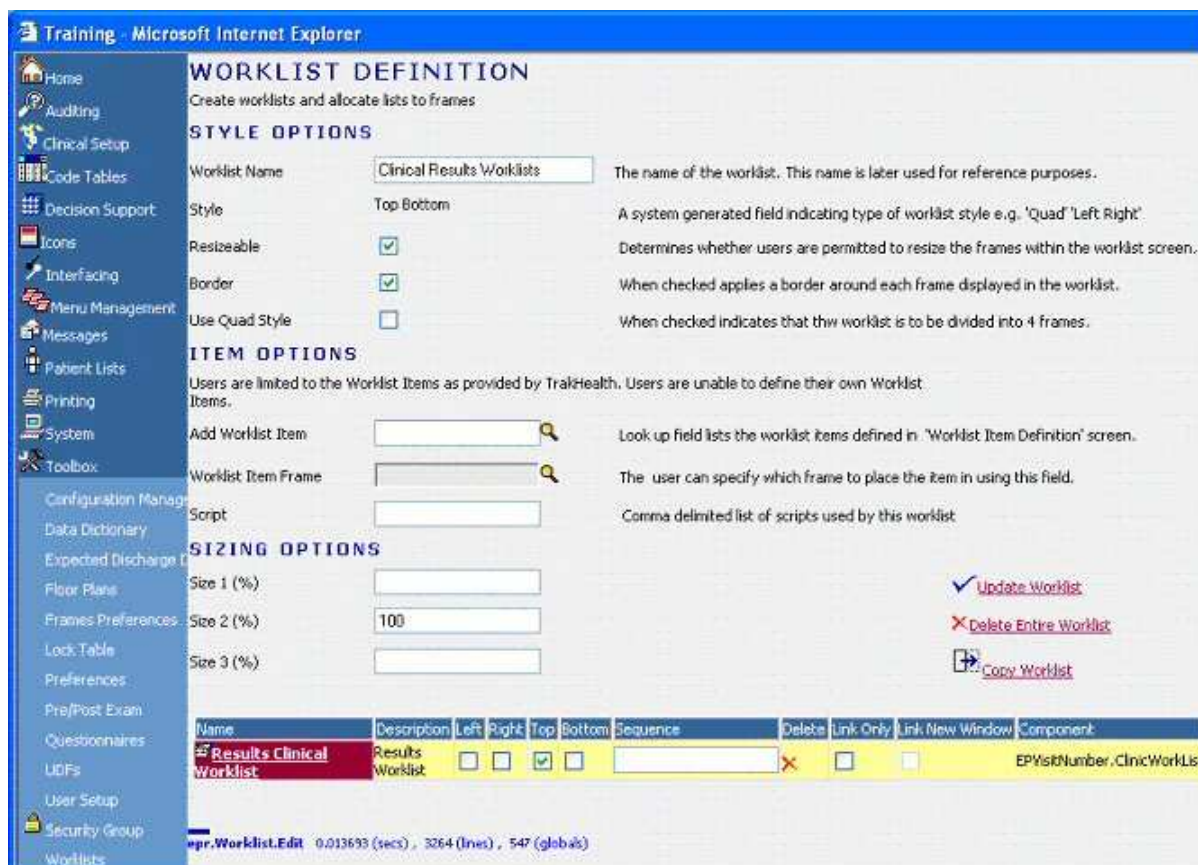


Figura 6 – Definizione Worklist

Questionnaire Manager

TrakCare è un sistema integrato e contemporaneamente aperto. Anche se questa po' sembrare una contraddizione, in realtà è uno dei punti di forza, in quanto significa poter corredare l'applicazione anche di nuove strutture informative che diventano a tutti gli effetti parte integrante del sistema. E' fondamentale quindi che qualunque struttura di informazioni si voglia aggiungere mantenga l'integrità complessiva della base dati.

Per dare alcuni esempi concreti basti pensare a differenti metodi di denominare le stesse prestazioni oppure alla necessità di aggiungere scale di valutazione clinico/scientifiche proprie di una specializzazione o di un particolare progetto sperimentale (ad esempio: il Glasgow coma scale⁴). QUESTIONNAIRE MANAGER viene anche utilizzato per costruire referti normalizzati e personalizzati dai singoli reparti. Questo aspetto risulta determinante in fase di ricerca dei risultati per valutazioni statistiche ed epidemiologiche.

⁴ Il Glasgow Coma Scale (Scala di Glasgow), nota anche in medicina come scala GCS è stata sviluppata dai neurochirurghi Graham Teasdale e Bryan Jennet per tenere traccia dell'evoluzione clinica dello stato del paziente in coma: essa si basa su tre tipi di risposta agli stimoli (oculare, verbale e motoria) e si esprime sinteticamente con un numero.



Figura 7 – Questionnaire manager

Clinical Decision Support System

CLINICAL DSS assolve due importanti requisiti:

- Creare le regole di compatibilità cliniche
- Generare report dinamici definiti dall'utente.

Regole di compatibilità cliniche

CLINICAL DSS è stato pensato per essere un valido aiuto nella complessa attività di indagine e di cura. E' un ambiente di generazione dove vengono definite in modo agevole le propedeuticità e compatibilità delle prestazioni così come dei farmaci. Ha un ruolo solo consultivo quindi, in caso di possibile errore clinico, il sistema suggerisce o avvisa l'utente secondo quanto stabilito e descritto nelle regole del sistema (gestione automatica degli alert). CLINICAL DSS agisce anche in fase di prenotazione della prestazione.

Il sistema non impone alcuna regola: quelle da implementate sono stabilite dai referenti medici della struttura. Tipicamente è un ambiente che si costruisce gradualmente con il tempo e condividendo a livello di azienda l'impostazione scientifica.

L'utente può definire un evento di qualunque natura (clinica o amministrativa) al verificarsi del quale il sistema intraprende alcune possibili attività di segnalazione o allarme (gestione centralizzata degli alert) quali:

- Eseguire funzioni legate al processo (ad esempio impostare stati, fare ordini, ecc.)
- Mandare un messaggio (via SMS, email, etc.) di alert all'utente per, ad esempio, valori di un referto fuori soglia di tolleranza, rifiuto della terapia da parte del paziente, etc.
- Interrompere un certo processo

Report dinamici

TrakCare mette a disposizione un sofisticato sistema di reporting che agisce in ambiente web. In aggiunta, CLINICAL DSS consente a tutta l'organizzazione di estrarre report non standard secondo necessità.



Figura 8 – Creazione query

La necessità di estrarre dati da un sistema informatico non è propria solo delle unità amministrative o sanitarie ma anche (e soprattutto) dell'ambito medico. In generale TrakCare, vista la sua completezza, è un generatore di grandi volumi di dati. Tali dati devono rispondere almeno a due criteri : essere corretti ed accessibili. Per questa ragione CLINICAL DSS mette a disposizione un sistema di reporting guidato (tipo wizard) che permette anche ad utenti non esperti di generare richieste di estrazione dati (anche complesse) in breve tempo (per breve si intende il tempo di creazione della query da parte dell'utente). Le variabili sono quelle della base dati: quindi tutti i dati possono essere estratti con qualunque criterio di aggregazione. L'ambiente di report è incluso nel sistema ed è possibile accedervi da qualunque utente autorizzato. Sempre con CLINICAL DSS è possibile definire indicatori di performance per monitorare le attività svolte.

Clinical Pathway

E' già da anni che le Aziende Sanitarie pubbliche e private cercano di valutare l'impatto introdotto dai DRG sull'andamento aziendale. E' dimostrato che per alcune patologie è possibile definire un percorso di cura standard. La vera complessità è la gestione di un quadro del paziente spesso non corrispondente allo standard clinico/scientifico pre-definito. CLINICAL PATHWAY (Percorso Clinico) consente di definire i protocolli da applicare in ragione di sintomi riscontrati e diagnosi effettuate. CLINICAL PATHWAY consente di disegnare i Protocolli di Cura coordinando il percorso diagnostico e terapeutico di dettaglio ed i relativi attori coinvolti in una scala temporale.

L'assegnazione di un protocollo di cura può, a sua volta, essere il risultato dell'applicazione di un Protocollo Diagnostico, stabilito sulla base di indagini ma anche di questionari definibili ad hoc.

L'applicazione dei protocolli consente non solo di prevedere l'assorbimento delle risorse necessarie per determinate patologie ma di derivarne anche una previsione di costi da sostenere, consentendo quindi il confronto con i ricavi attribuibili per singolo DRG.

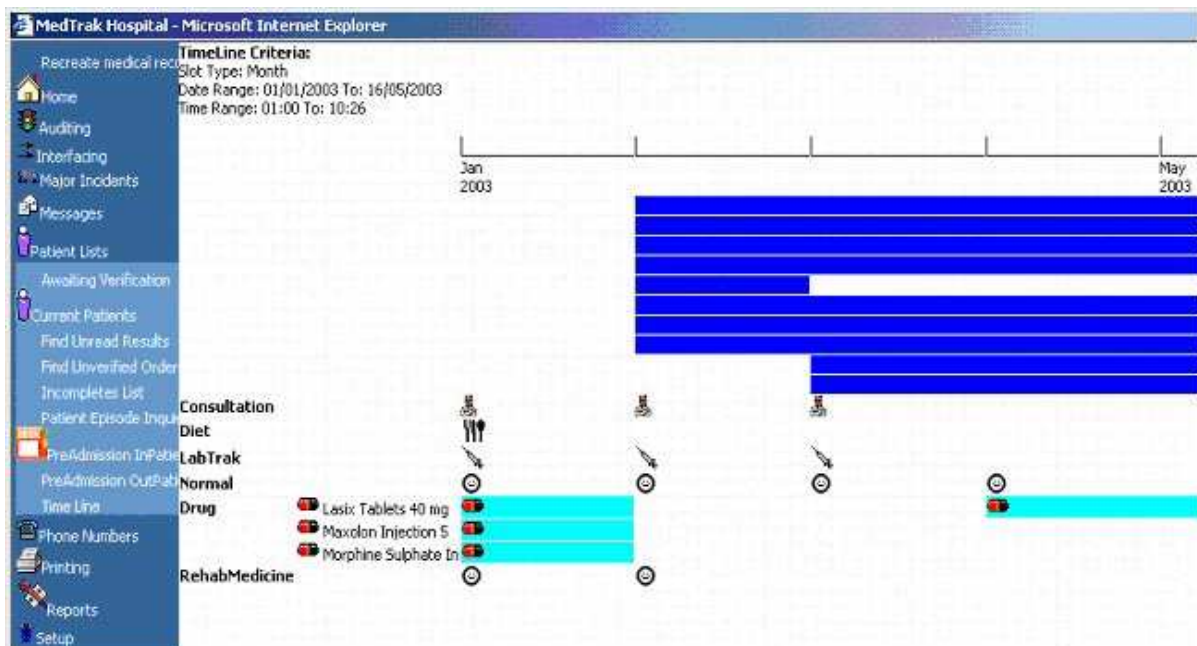


Figura 9 - Clinical Pathway

Le risorse programmate nel protocollo di cura sono quelle configurate nel sistema, quindi anche a livello di controllo di gestione determinano un'assoluta accuratezza nel calcolo dell'assorbimento.

L'ambito scientifico ha da tempo avanzato la richiesta di strumenti a supporto della ricerca applicata, ma solo di recente sono state messe a disposizione tecnologie informatiche in grado di dare una risposta professionale a questo capitolo gestionale molto importante. CLINICAL PATHWAY aiuta anche nell'applicazione del controllo di Qualità per la valutazione dell'appropriatezza delle risorse impegnate in ragione del caso clinico trattato e della deviazione verificatesi rispetto allo standard. Il

traguardo che è possibile raggiungere è la Medicina Basata sull'Evidenza (Evidence-based medicine – EBM), un processo di apprendimento continuo basato su casi specifici e sui risultati conseguiti in termini di validità ed applicabilità clinica.

2.3 OVERVIEW DELL'ARCHITETTURA : TAILORING

A garanzia dell'assoluta flessibilità applicativa, TrakCare mette a disposizione anche importanti strumenti di tailoring, attraverso i quali si può disegnare completamente l'applicazione, adattarla all'organizzazione ma anche a gruppi di utenti. Particolarmente sofisticate sono tutte le funzioni di gestione del sistema per le quali si rimanda agli allegati specifici.

Layout editor

Forse il più potente degli strumenti di tailoring è LAYOUT EDITOR, attraverso il quale vengono disegnate le maschere applicative. Ovvero è possibile disegnare una medesima applicazione, ad esempio una cartella clinica di reparto, in modo diverso per medici e/o infermieri, per reparti differenti ma anche per utenti del medesimo reparto. Il risultato è un vero Clinical Information System.

LAYOUT EDITOR è un configuratore dinamico di applicazioni. Ogni applicazione lavora in un ambito predefinito (workbench) dove valgono regole, parametri e dati.

LAYOUT EDITOR risponde ad ogni necessità clinica organizzando dati e sequenza logica delle applicazioni così come i processi clinico-organizzativi richiedono.

Dalla cardiologia alla medicina generale tutti i reparti utilizzano in modo diverso, con criteri diversi, lo stesso insieme di dati e di informazioni.

Cambia il look and feel ma viene mantenuta l'integrità del Data Base.

Come si vede dalla figura i campi sono scelti direttamente dal DB. Possono cambiare di posizione, dimensione, carattere e qualunque componente grafica si voglia.

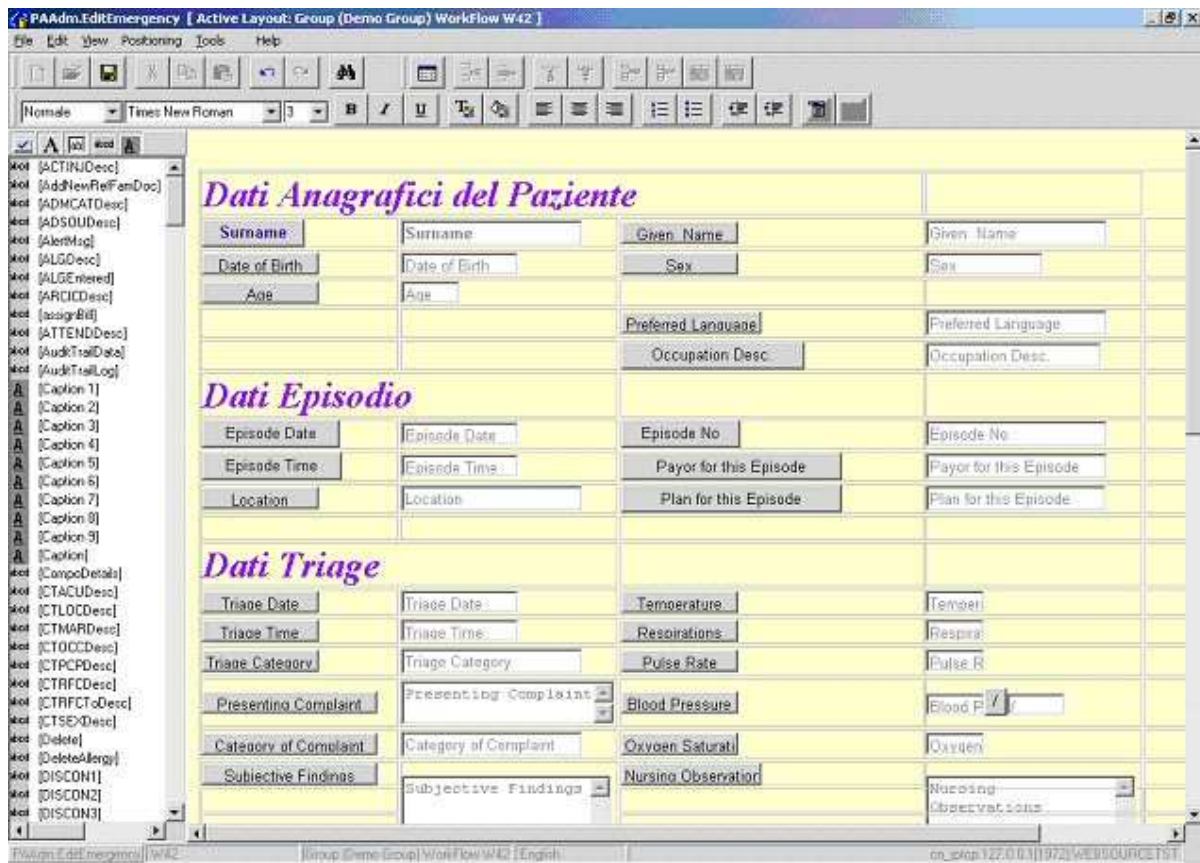


Figura 10 – Layout editor

Per ogni campo è possibile definire le proprietà e gli attributi: ad esempio l'obbligatorietà.

Le applicazioni possono così adattarsi a qualunque esigenza organizzativa ma anche funzionale si voglia.

Floor plan manager

L'organizzazione è anche fisica e ambientale. Per una corretta navigazione nell'ambiente di lavoro è stato introdotto un nuovo strumento che consente di disegnare i reparti in vere e proprie mappe. Per ogni piano / reparto vengono poi definiti i singoli posti letto.

Tutti i sistemi di visualizzazione dei dati del paziente e di informazione vengono quindi attivati su una visuale chiara come un cruscotto di controllo.

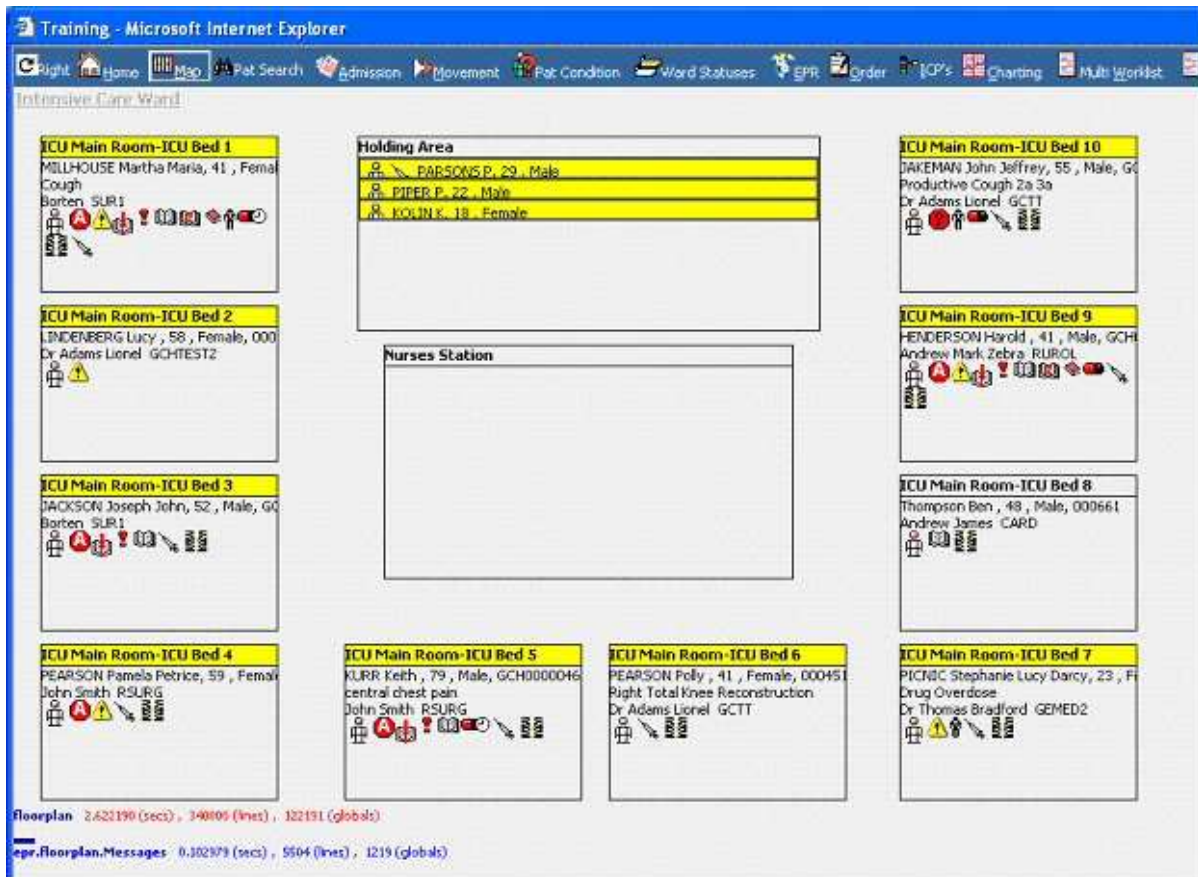
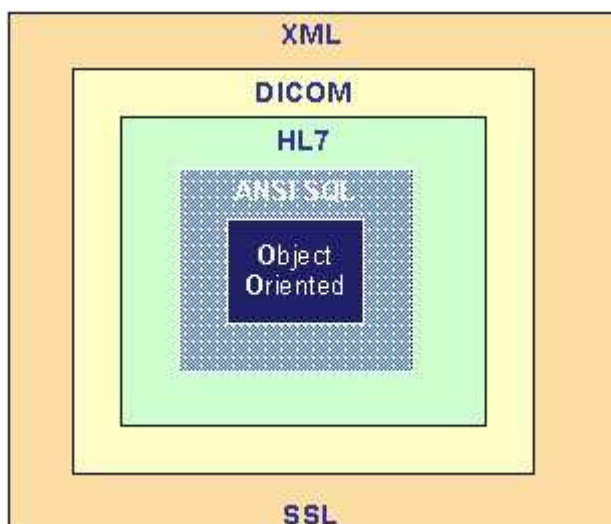


Figura 11 – Floor plan manager

2.4 STANDARD



TrakCare rispetta tutti i principali standard architetturali, gestionali e di colloquio del mondo software in generale e di quello sanitario in particolare. Può pertanto definirsi, anche dal punto di vista tecnologico, come un sistema realmente aperto, integrato ed integrabile.

HL7

I processi di comunicazione di TrakCare sono tutti basati sullo standard internazionale HL7. Questa scelta progettuale è stata definita per rendere l'architettura completamente aperta all'integrazione con qualunque altro sistema senza richiedere l'utilizzo di pesanti interfacce.

Il protocollo HL7 è nativo nell'applicazione.

HL7 è utilizzato sia per la comunicazione con sistemi dipartimentali di terze parti che per l'accesso ad anagrafiche esterne. Questo consente di mantenere le anagrafiche centrali di riferimento e di colloquiare in modo paritetico con le altre secondarie.

XML

Altro standard di riferimento dell'architettura TrakCare è XML, utilizzato per colloquiare con sistemi aperti come strutture regionali e/o sovraCUP. Gli oggetti definiti nella struttura dati, possono fungere da rappresentazione diretta dei documenti XML e viceversa. Le classi del DB Caché possono essere utilizzate automaticamente come documenti XML sotto forma di file o contenuto online. Inoltre, le classi Caché sono in grado di generare automaticamente i file DTD XML corrispondenti. I documenti XML possono essere trasformati automaticamente nell'oggetto Caché equivalente. I contenuti XML possono essere letti da file, flussi di dati o richieste HTTP. Il sistema convalida il contenuto XML da importare in base allo standard del settore DTD XML. Il supporto di XML può essere adattato in base alle particolari esigenze delle applicazioni.

DICOM

La soluzione TrakCare comprende sia il RIS che il PACS. I due ambienti sono cooperativi attraverso l'utilizzo dello standard DICOM supportato dall'architettura.

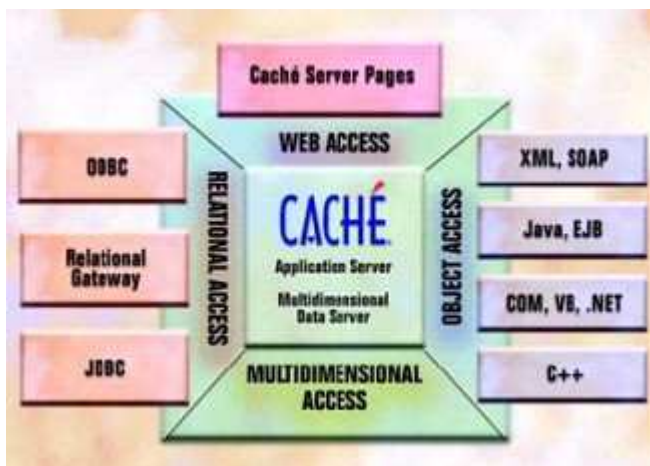
Il sistema prevede:

- Il colloquio con le modalità, tramite MODALITY WORKLIST SERVER, STORAGE SERVICE CLASS PROVIDER e DICOM QUERY PROVIDER.
- L'archiviazione delle immagini, a breve o a lungo termine (DICOM SERVER e ARCHIVE).
- WORKFLOW SERVER per risolvere le eventuali esigenze di pre-fetching dall'archivio a lungo termine.
- Ed infine CLINICAL VIEWER per le funzionalità di visualizzazione e ri-visualizzazione; seleziona automaticamente l'immagine necessaria e la visualizza, anche su un secondo monitor, in fase di refertazione o di successivo esame dei referti.

Ansi SQL

L'accesso al Data Base è completamente standard, certificato ANSI SQL; supporta sia ODBC che JDBC. Questo consente qualunque connettività a data base relazionali nonché l'accesso ai dati diretto con query standard, reporting e analysis tools.

Object oriented



Il data base che supporta la tecnologia TrakCare è completamente Object Oriented. Questo consente: encapsulation, multiple inheritance, poliformismo dei dati, referenzialità delle strutture, ecc. Per questa ragione, tutti i dati possono essere letti e utilizzati da altre applicazioni in modo veloce e sicuro. L'accesso è anche garantito dall'utilizzo di java class che consentono integrazioni bidirezionali efficienti.

Nella figura sotto riportata sono indicati altri standard presenti nel sistema e la loro posizione architetturale.

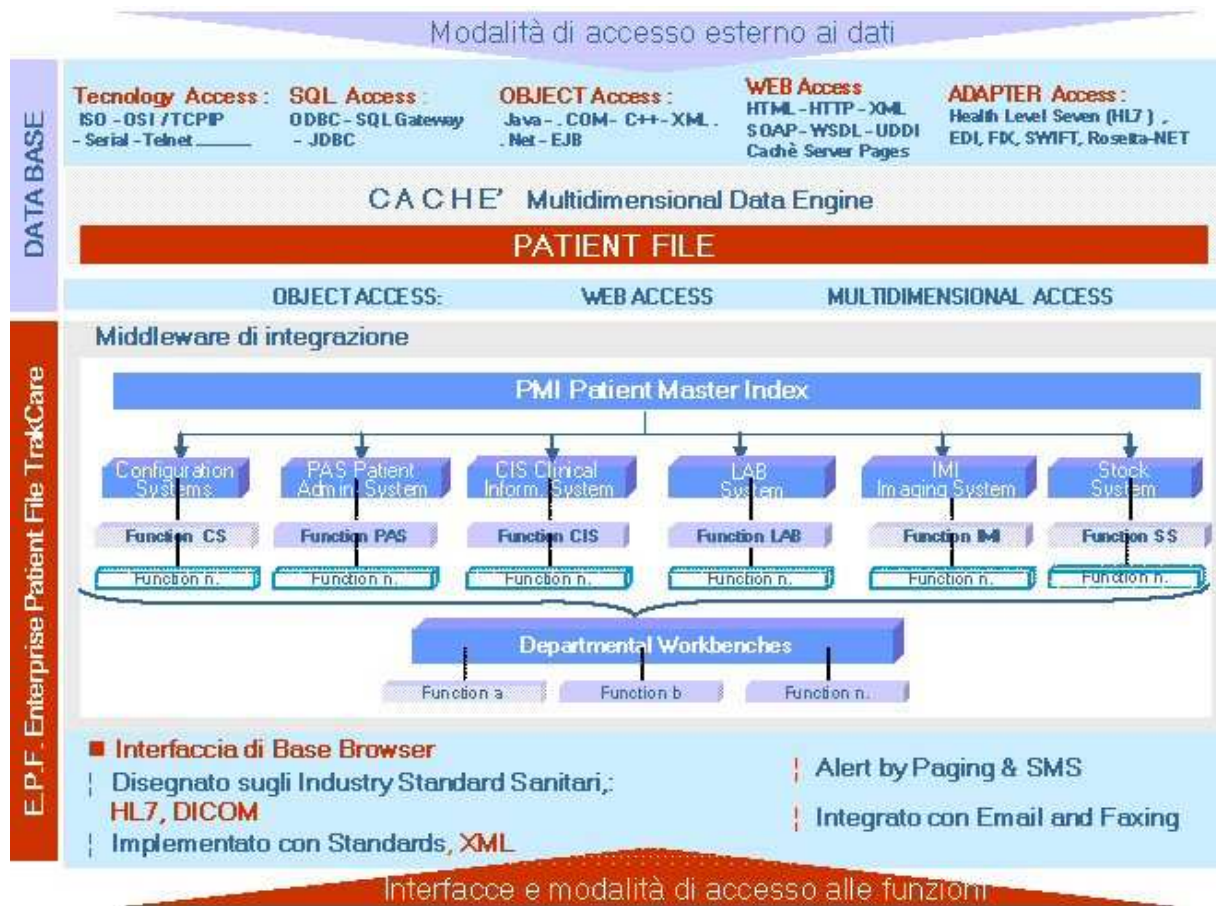


Figura 12 – Standard supportati e modalità di accesso ai dati

2.5 ARCHITETTURA FUNZIONALE

La piattaforma sanitaria TrakCare controlla tutti i processi relativi alla gestione ed erogazione dei servizi sanitari al Paziente. Tali processi, siano essi orizzontali o verticali, sono coperti dai moduli logici riportati nella figura 13.

Enterprise Patient File System (E.P.F.)

La piattaforma applicativa Clinico / Sanitaria TrakCare è definita Enterprise Patient File (E.P.F.) System.

In particolare, il sistema rappresenta la base di riferimento funzionale ed il motore di integrazione dei processi Sanitari tramite i quali l'Azienda può pianificare e realizzare liberamente il proprio percorso di informatizzazione, con la garanzia di raggiungere gli obiettivi prefissati. L'EPF, architeturalmente ed in modo nativo, pone al centro dei suoi processi l'Assistito/Paziente con tutti i suoi dati.

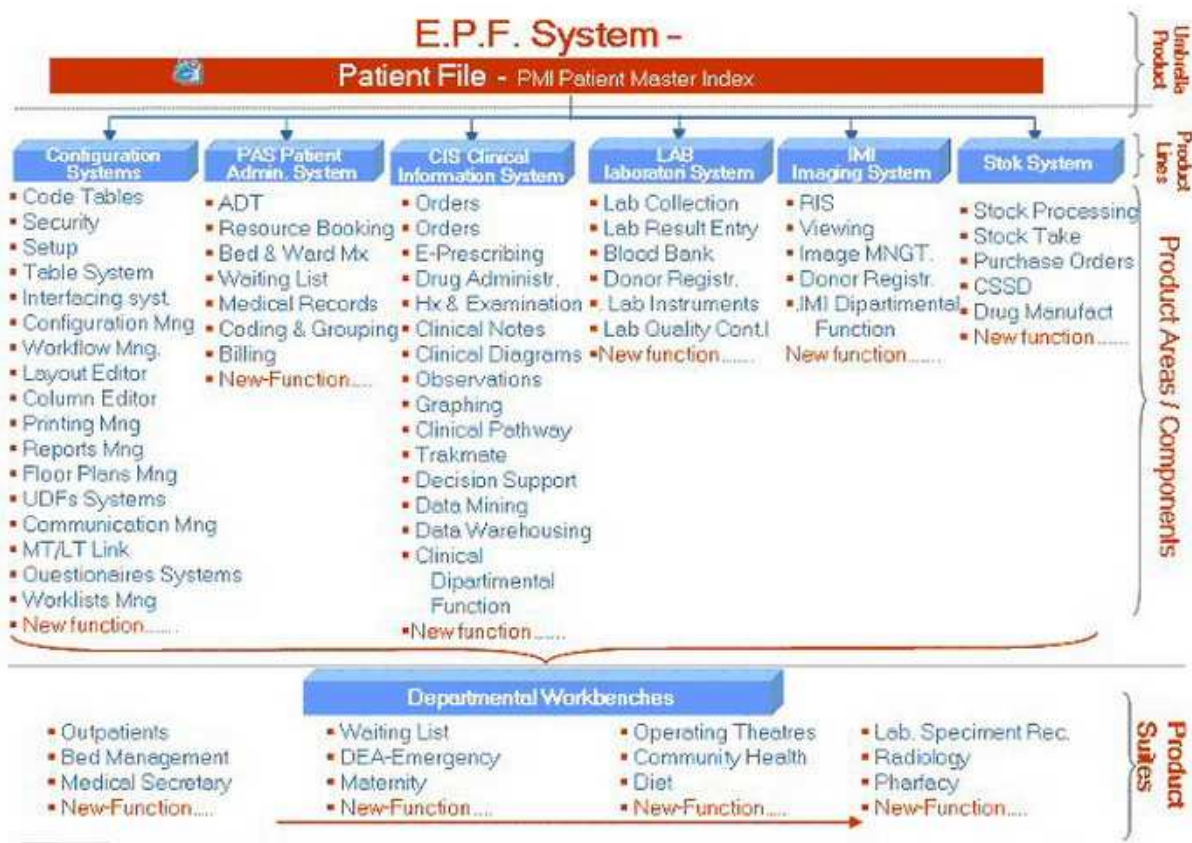


Figura 13 – Architettura funzionale

Questi sono raccolti ed elaborati nel Patient File durante il percorso di assistenza e cura:

- attraverso le proprie funzioni applicative verticali.
- tramite l'integrazione con altri moduli, utilizzando interfacce applicative e tecnologiche standard di mercato.

Il Patient File (altrimenti definibile come dossier, cartella, ecc.) è il contenitore di tutti i dati gestionali, amministrativi e clinici dell'Assistito, strutturati, accessibili e processabili dai diversi moduli applicativi.

Il sistema EPF si caratterizza come sistema Aziendale Clinico/Sanitario proprio perché, potendo gestire in modo nativo tutti i processi sanitari di assistenza e cura, è anche in grado di integrare altri sistemi esterni e gestirne correttamente il colloquio applicativo

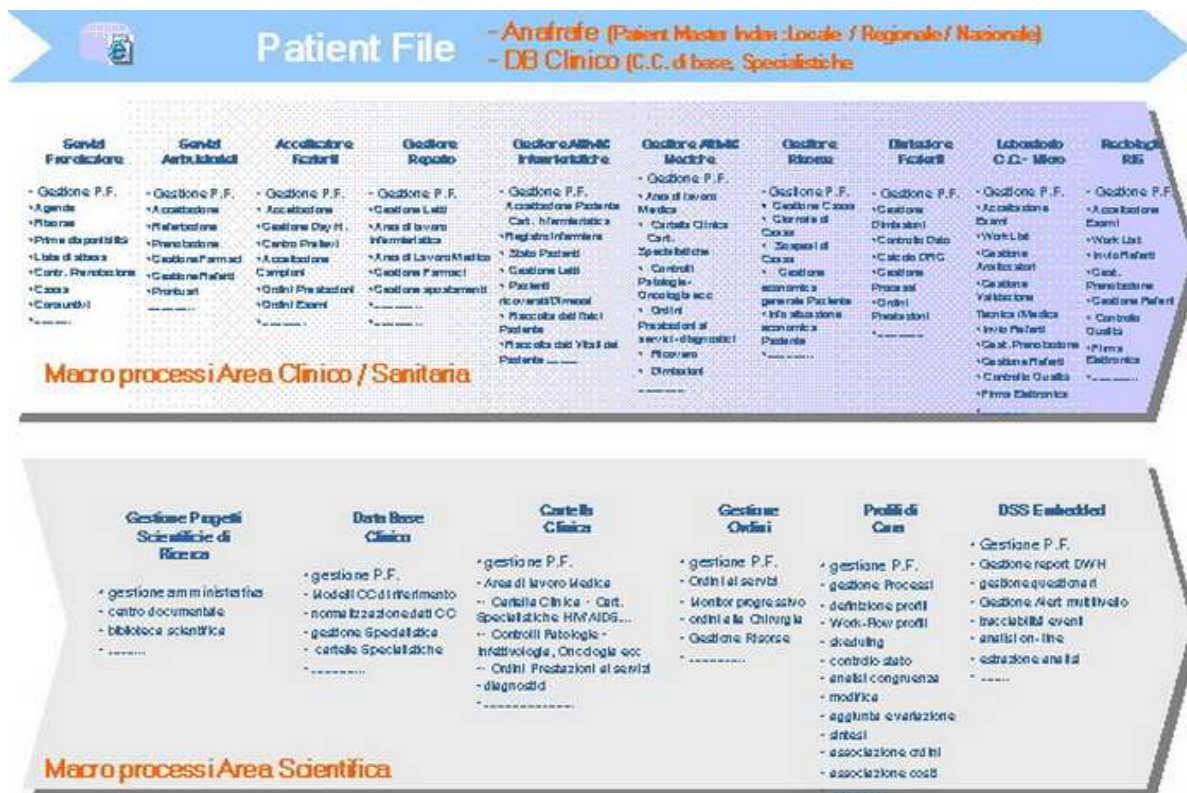


Figura 14 – I Macro Processi delle Aree Clinico-Sanitaria e Scientifica

La figura 14, sintetizza come il sistema gestisce il Percorso di Assistenza e Cura che il Paziente compie all' interno di una struttura Sanitaria.

Il sistema realizza una "reale rete" di servizi trasversale, disponibile per l'organizzazione delle singole strutture aziendali e del territorio, che lavora su dati condivisi e strutturati (amministrativi, gestionali e clinici) del Patient File.

I processi gestiti da applicazioni pre-esistenti che si desidera mantenere sono "visti" dal sistema come applicazioni proprie attraverso interfacciamenti standard.

L'Azienda Sanitaria, comunque, ha fin dall'inizio disponibile tutta la piattaforma nella sua estensione applicativa/funzionale e quindi può scegliere di attivare di volta in volta quelle funzioni che risultano più consone al proprio percorso progettuale o più necessarie alle diverse fasi organizzative.

2.6 ALERT – AUDIT

I Sistemi Informatici Sanitari, trattando dati sensibili, devono garantire opportuni meccanismi di sicurezza e riservatezza delle informazioni, volti a limitare accessi indesiderati come prescritto nel DPR 318.

A fianco di tali sistemi, sono però utili strumenti che consentano il monitoraggio dell'accesso ai dati, sia a scopo di analisi retrospettiva degli accessi effettuati, che per scoraggiare tentativi di accesso non autorizzato. TrakCare possiede una serie molto articolata di funzionalità di auditing, configurabili a seconda delle diverse esigenze e policy aziendali. Le informazioni vengono rappresentate in un formato di facile lettura per il responsabile della sicurezza, che non deve quindi esaminare log spesso criptici creati automaticamente dai gestori di database. Funzionalità particolarmente interessante è quella che traccia i semplici accessi ai dati, e non solo la loro modifica. Di seguito vengono sinteticamente riportate le misure di auditing attivabili.

Monitoraggio accessi

Possono essere tracciati sia gli accessi effettuati in modo corretto al sistema, che tentativi di accesso non andati a buon fine.

Regole di audit

Poter effettuare un audit per qualsiasi informazione venga visionata o modificata è certamente un'opzione importante. Tuttavia, spesso non tutte le informazioni rappresentano dati sensibili, o che comunque si vuole monitorare.

Inoltre, la generazione dei dati di audit potrebbe portare a consumare risorse di sistema in modo non desiderato.

E' per questo possibile modificare la configurazione standard, ove è previsto il tracciamento solo di alcune opzioni, in modo più restrittivo o viceversa più esteso.

Regole audit: componenti

TrakCare è composto da una serie molto dettagliata di componenti. Componenti sono le singole videate, o funzioni, disponibili per l'attivazione; possono essere funzioni standard dell'applicativo o create ad hoc per una specifica esigenza.

Per tutti i componenti è possibile stabilire se sottoporre ad audit l'accesso alla funzione o anche la modifica al componente stesso, tracciando così le operazioni di configurazione del sistema.

Regole audit: questionari

Un'altra caratteristica peculiare di TrakCare sono i questionari, tramite i quali è possibile realizzare delle videate di raccolta dati associate a regole ed eventi.

In modo analogo alla gestione dei componenti, i questionari possono essere sottoposti ad audit, per quanto riguarda il loro inserimento, la modifica o la cancellazione.

Audit: i risultati

Una volta impostate le azioni da sottoporre ad audit, è possibile visionare i risultati secondo percorsi diversi.

Audit: qualsiasi azione

Tramite questa funzione è possibile richiamare, eventualmente selezionando una data ed un utente, tutte le azioni eseguite dal sistema. Sono riportati la data ed ora di esecuzione, gli estremi dell'utente che ha eseguito l'operazione (nome e postazione di lavoro), il tipo di operazione svolta e, ove significativo, anche il paziente o l'episodio che è stato oggetto dell'operazione stessa.

I collegamenti a paziente ed episodio sono attivi: selezionandoli, è possibile risalire alle operazioni di modifica effettuate sui dati oltre che le semplici operazioni di consultazione dei dati.

Audit: modifica dati di un paziente

Tramite questa funzione è possibile selezionare una persona e visualizzare tutte le relative variazioni ai dati. Il termine persona sta ad indicare sia pazienti che accompagnatori od altro.

La selezione del paziente può avvenire secondo una combinazione di campi liberamente definibile tra tutti i dati propri del paziente; tra i più comuni citiamo: cognome e nome, codice aziendale o codice fiscale, numero episodio, data di nascita o età, sesso, ecc. E' possibile includere nella ricerca anche i pazienti con omofonie (cognomi o nomi dalla grafia diversa ma di pronuncia simile) o viceversa restringere il risultato a coloro che hanno cognome e nome scritti esattamente come richiesto, piuttosto che ai pazienti che abbiano già avuto un episodio presso la struttura.

E' possibile selezionare quali operazioni sui dati tracciare tra inserimento, modifica e cancellazione.

Ove sono presenti ulteriori dettagli, accessibili all'utente, sono richiamabili tramite dei link attivi.

Audit: accesso ad un paziente

A fronte della selezione di un paziente, questa ha la funzione di elencare tutti gli accessi ai diversi componenti che sono stati sottoposti ad audit.

Audit: tabelle

La ricerca dei dati modificati può anche avvenire sulla base delle tabelle che li contengono. Questo lascia la massima libertà nell'includere tra gli elementi sottoposti ad audit anche informazioni quali la struttura dei reparti, la definizione degli utenti, i parametri di configurazione, ecc.

E' possibile restringere la ricerca per intervallo di date, nome tabella o utente che ha effettuato la modifica.

Anche qui, appositi link consentono di accedere al dettaglio della modifica operata.

Dati con profondità storica

TrakCare consente la memorizzazione dei dati con profondità storica. Se questo è utile ai fini dell'audit, tipicamente attività di una figura tecnica, lo è anche per alcune tipologie di utenti finali del sistema.

Per questo motivo, in alcune funzioni di gestione sono disponibili i collegamenti alle informazioni storiche.

Dati anagrafici e di episodio

All'interno della videata anagrafica, sono disponibili due funzioni:

- la prima consente di visualizzare le modifiche apportate ai dati anagrafici o ai dati relativi all'episodio;
- la seconda richiama gli accessi effettuati ai dati del paziente.

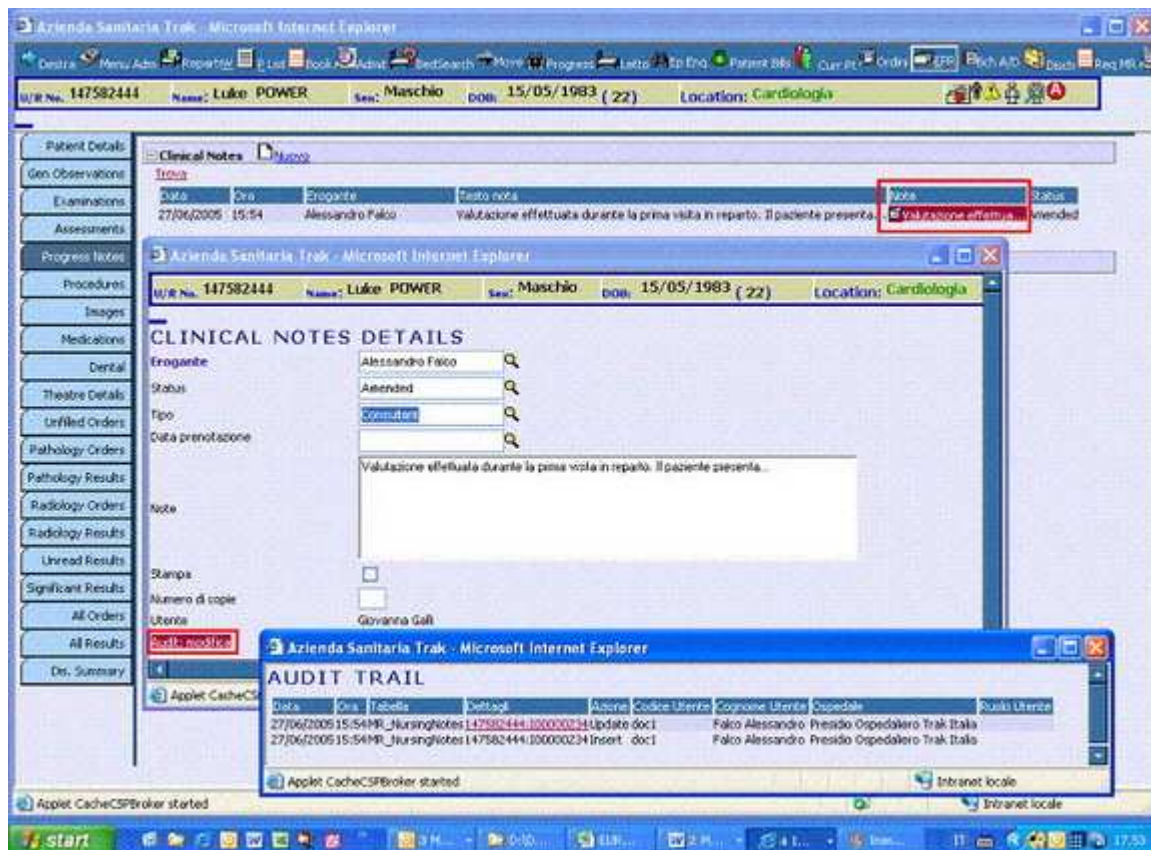
The screenshot displays the 'DETTAGLI CLIENTE/PAZIENTE' (Patient Details) page in a web browser. The main content is a table with the following columns: Data audit, Ora audit, Utente, Nome computer, ClientIP, Tipo, Cod. Paziente, N. Episodio, Class/WorkFlow, and WorkFlowId. The table lists several audit entries from July 2005, all performed by 'Vice- Amministratorelocalhost di sistema'.

Data audit	Ora audit	Utente	Nome computer	ClientIP	Tipo	Cod. Paziente	N. Episodio	Class/WorkFlow	WorkFlowId
25/07/2005	17:10	Vice- Amministratorelocalhost di sistema	127.0.0.1	Open	0000250			DB.Patient Details	PAPerson.Ed (Patient Edit)
25/07/2005	17:08	Vice- Amministratorelocalhost di sistema	127.0.0.1	Open	0000250			DB.Patient Details	PAPerson.Ed (Patient Edit)
25/07/2005	16:59	Vice- Amministratorelocalhost di sistema	127.0.0.1	Open	0000250			DB.Patient Details	PAPerson.Ed (Patient Edit)
25/07/2005	16:57	Vice- Amministratorelocalhost di sistema	127.0.0.1	Open	0000250			DB.Patient Details	PAPerson.Ed (Patient Edit)
08/07/2005	16:53	Vice- Amministratorelocalhost di sistema	127.0.0.1	Open	0000250	00000630			PAPerson.Ed
08/07/2005	16:53	Vice- Amministratorelocalhost di sistema	127.0.0.1	Open	0000250	00000630			PAAdmin.Edit
08/07/2005	16:52	Governia Galli localhost	127.0.0.1	Open	0000250				PAPerson.Ed
08/07/2005	16:51	Governia Galli localhost	127.0.0.1	Open	0000250				PAPerson.Ed
08/07/2005	16:44	Vice- Amministratorelocalhost di sistema	127.0.0.1	Open	0000250				PAPerson.Ed

On the right side of the page, there are sections for 'ALTRI DATI' (Other Data) with links for 'Anamnesi', 'Family Link', and 'Documenti'. Below that is the 'DECESSO' (Death) section with input fields. At the bottom right, the 'AUDIT' section shows the user 'Halqah Binti Kamaruddin' and the last update date '30/06/2005' at '12:42' from 'Fondazione Ospedale Trak Italia'. Two buttons, 'Audit storico' and 'Audit modifica', are highlighted with a red box.

Dati clinici

E' possibile visionare gli inserimenti / modifiche apportati alle note cliniche (Clinical Notes). Ancora una volta, i link presentati sono attivi, e per ognuno di questi possono essere richiamati i dettagli delle modifiche apportate



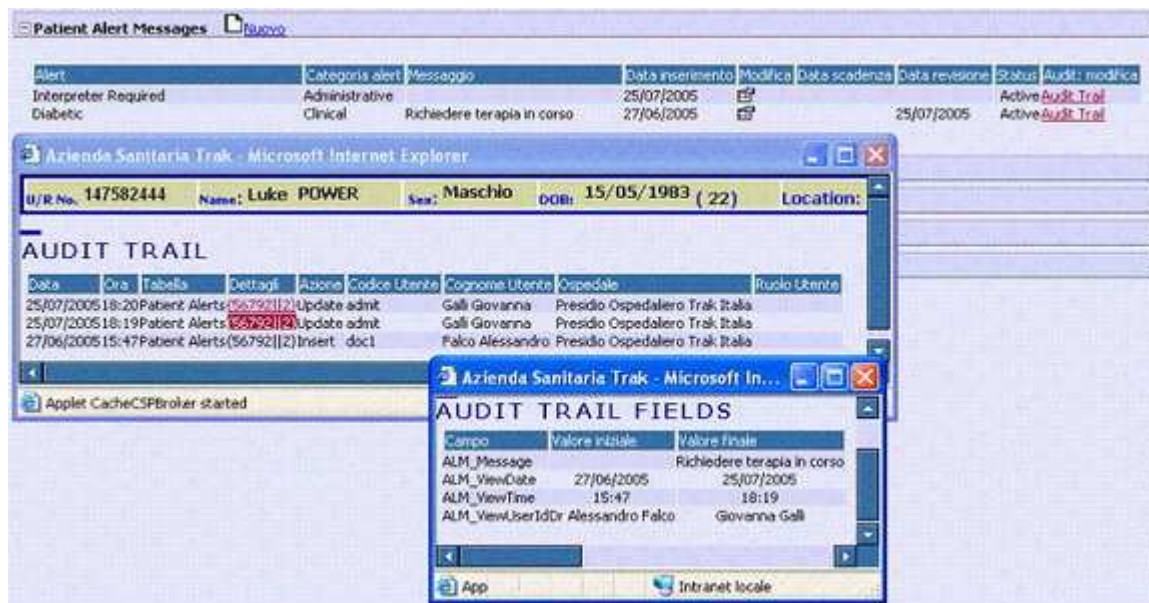
Allergie

Un componente particolarmente significativo che è possibile sottoporre ad audit è quello relativo alle allergie. In questo modo può essere tracciata l'evoluzione della patologia.



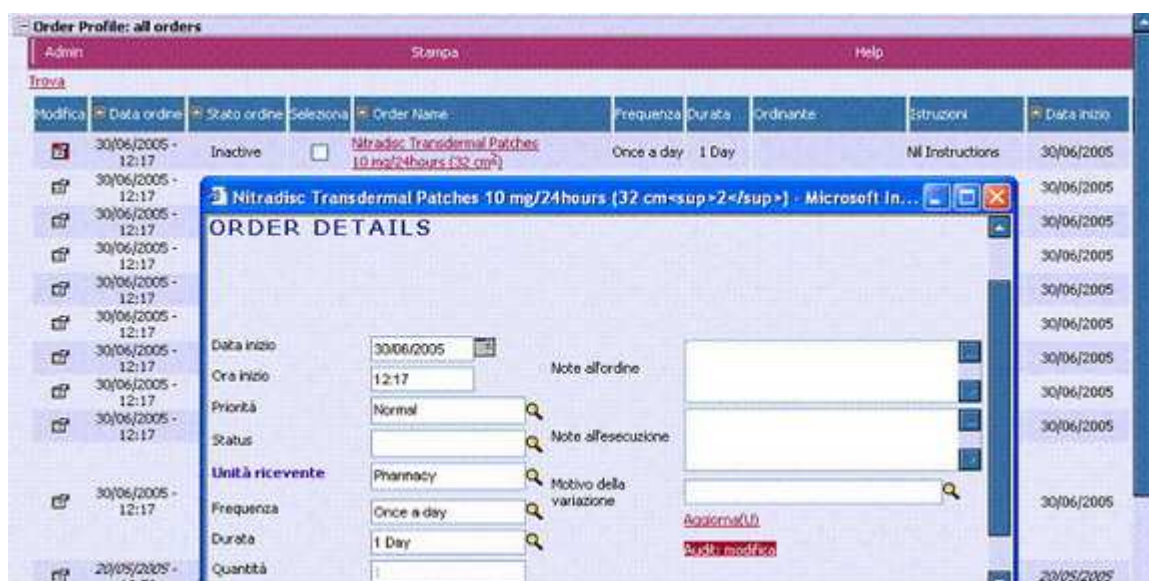
Alert

Anche per gli alert, che possono essere sia clinici che amministrativi, è disponibile il link di audit.



Ordini e referti

Le funzionalità di audit per gli ordini hanno anche lo scopo di tracciare l'evoluzione dell'ordine stesso: dal momento della richiesta, fino al suo completamento o cancellazione.



Analogamente, per i referti inseriti direttamente a sistema sono disponibili le informazioni per il tracking della prestazione: inserimento ordine, completamento, refertazione, ultimo aggiornamento, ma anche lettura e da parte di quale utente.

2.7 VALORI AGGIUNTI

Riassumendo quanto esposto, è possibile indicare i valori aggiunti dei quali i diversi attori sanitari, ma anche il paziente, possono beneficiare a fronte dell'utilizzo di TrakCare.

Vantaggi organizzativi

Dal punto di vista organizzativo i principali vantaggi che il sistema permette di raggiungere sono:

Il Paziente ha sempre disponibili le informazioni relative al suo iter di cura, sia internamente alla struttura dell'Istituto che per collegamenti esterni (Patient File on -line).

Gli Operatori Sanitari (medici, personale paramedico, tecnici, ecc.) accedono ad informazioni relative al Paziente univoche, strutturate e complete per migliorare la capacità di diagnosi, l'accuratezza del processo di cura e quindi la qualità del servizio erogato.

Gli Operatori Amministrativi dispongono di dati omogenei e completi (prestazioni, costi reali, ecc.) per amministrare e governare i diversi processi gestionali allo scopo di rendere un servizio sempre di più alto livello qualitativo e massimizzare le risorse disponibili.

La Direzione possiede dati, informazioni e strumenti per misurare e governare i diversi processi interni attraverso indicatori e modelli di riferimento:

- La Direzione Generale attraverso indicatori di attività ed efficienza operativa.
- Le Direzioni Scientifica e Sanitaria attraverso dati clinici completi e strutturati.

Le Strutture di Governo Esterne (Ministero della Salute, Regione, ASL) dispongono di dati reali e completi riferiti ad attività e prestazioni erogate misurabili e quindi su cui operare scelte di indirizzo e di governo del Sistema Sanitario nella sua globalità.

Le Strutture Sanitarie, Socio Sanitarie e di Assistenza (Associazioni, Enti, Ricercatori Nazionali ed Internazionali) possono essere abilitate ad accedere ad informazioni che l'Istituto è in grado di rendere disponibili per fini o scopi diversi (informativi e scientifici).

Più in dettaglio, il sistema permette di realizzare l'integrazione operativa funzionale (interoperabilità) ai diversi livelli, risultando:

- Multi-area Funzionale/Operativa – tra le diverse aree operative sanitarie e gestionali dell'Azienda (unità amministrative, di ricovero, di cura, ecc.), in grado di garantire funzioni e dati per il controllo globale dei diversi processi.
- Multi-professionale – tra i diversi operatori sanitari, socio assistenziali, di ricerca e di amministrazione (medici ospedalieri, ricercatori interni ed esterni, operatori paramedici, operatori sanitari e socio sanitari esterni, ecc.).

- Multi-disciplinare – tra i diversi servizi resi all'Assistito/Paziente sanitari, socio sanitari e gestionali.
- Multi-struttura – tra le diverse strutture territoriali dell'Azienda o anche esterne all'Azienda stessa (Medico di Base, Aziende Ospedaliere, ASL/Distretti, Ambulatori, RSA, Comune, Università, altri Istituti di ricerca, ecc.) che operano su dati dell'Assistito / Paziente con l'obiettivo di gestire e controllare l'intero processo di assistenza e cura (Assistenza: diretta, territoriale, ospedaliera e domiciliare).
- Multi-dimensionale – nel fornire un "servizio" inteso come progetto di assistenza e cura che presuppone l'intervento pianificato di diversi operatori che operano in diverse strutture (es. piano mirato a patologie specifiche nell'ambito di interventi condivisi con altre strutture che hanno la stessa missione).

Inoltre, l'utilizzo delle funzioni applicative che realizzano la raccolta, la gestione e l'incremento nel tempo dei dati riferiti al Paziente (Patient File) unito alla capacità di elaborazione statistico / analitica degli stessi permette di realizzare:

- Il controllo dei processi di erogazione del Servizio, in relazione alla possibilità di misurare per migliorare sia dal punto di vista clinico sia di controllo del rapporto prestazioni erogate/costi sostenuti.
- La gestione degli indici di produttività, rendicontazione, reportistica per i vari livelli interni ed in rapporto alla Regione o enti associati.

Vantaggi funzionali

Dal punto di vista funzionale e, quindi, delle possibilità di utilizzo del sistema da parte delle diverse aree operative e dei singoli operatori, i vantaggi specifici di TrakCare sono riassumibili in:

- integrazione completa di tutti i processi sanitari compresi i moduli dei dipartimenti di diagnostica (laboratori e radiologie) e quelli relativi alla gestione scientifica come controllo dei profili di cura e analisi on-line sui dati Clinici;
- centralità del trattamento dei dati, univoci, inseriti una sola volta, disponibili all'organizzazione e sicuri;
- minimizzazione delle fasi di introduzione dei dati, anche con uso di apparecchiature specifiche (ad esempio smart card, palm-top, interfacciamento di apparecchiature di diagnosi ambulatoriale ecc.);
- globalità di trattamento dei dati; il sistema accetta ed elabora qualunque tipo di formato di dati (multimediali, multi-valuta ecc.);
- operatività per ambiente internazionale: l'utente può scegliere la lingua con la quale dialogare per tutte le funzioni del sistema (multi-lingua) e quindi operatori che agiscono in un ambiente sempre più aperto e globalizzato (come ad esempio ambienti di ricerca,

universitari ecc.) eseguono le stesse funzioni accedendo al sistema in lingue diverse;

- funzioni applicative distribuite, accessibili da qualunque punto interno o esterno all'Azienda, sono viste dall'utilizzatore come servizio ed in relazione alla propria singola operatività. Ad esempio: qualunque punto del processo di cura può accedere ai dati sanitari (cartella clinica); qualunque tipo di unità operativa può prenotare qualunque tipo di risorsa, ecc.;
- funzioni personalizzabili per i diversi operatori abilitati che permettono di controllare realmente i processi organizzativi di area, per singolo utente o gruppi di utenti (come ad esempio i processi di prenotazione o di attività mediche o infermieristiche per singole unità operative);
- flessibilità delle applicazioni assicurata dall'estrema semplicità nell'effettuare configurazioni o nuove implementazioni in modo da garantire la continua aderenza alla crescente dinamicità e modernizzazione dei processi previsti;
- modularità e scalabilità, quindi estendibilità del sistema attraverso ampliamenti con moduli già presenti nella piattaforma di base o con nuovi sviluppi funzionali sempre nell'ottica dell'integrazione di tutto il sistema informativo;
- funzioni variabili rispetto al flusso operativo che permettono dinamicamente di rendere l'utente finale più libero di usare l'insieme delle funzioni disponibili secondo modalità a lui più consone (come ad esempio: punto del processo in cui eseguire dei controlli di congruità per la presenza dei dati anagrafici, o attivazione/inibizione di singole funzioni in determinati momenti del flusso operativo, ecc.);
- funzioni di alto profilo e contenuto applicativo/funzionale, che coprono interamente il processo e che sono aperte a miglioramenti ed evoluzioni sia specifiche che di normativa (come ad esempio i sistemi di diagnostica strumentale dove l'operatività e la completezza funzionale sono particolarmente importanti).;
- interfacce utente secondo i layout di maggiore diffusione come riferimento visivo e con la possibilità di personalizzazione dei diversi formati e colorazioni, che permette di usare lo strumento inserito in un contesto di interfacce molto comuni e diffuse a tutto vantaggio della capacità e facilità di utilizzo;
- interfacce programmabili dall'utente finale, facilmente aggiornate e disegnate nella forma, contenuti e funzioni ai vari livelli: singolo, di gruppo, di area, ecc.;
- trasparenza funzionale della singola applicazione rispetto al sistema nel suo complesso – Integrazione Funzionale;
- ampia possibilità di parametrizzazione – la struttura è libera nella definizione dei flussi operativi, onde usare il sistema a fini organizzativi e di miglioramento dei processi. La parametrizzazione può arrivare al singolo utente, sia per le funzioni di data entry che per le funzioni di interrogazione (reportistica) e per le stampe;

- piena integrazione dei processi clinici con il sistema amministrativo, considerando i processi trasversali alle due aree quali la consuntivazione a livello contabile delle prestazioni erogate e di quelle relative ai consumi (risorse, farmaci, ecc.);
- piena aderenza alla normativa vigente a livello interno (Istituto locale (ASL / Regione ecc.) e nazionale (direttive ministeriali);
- capacità di fornire in tempo reale report, analisi e statistiche che rappresentino in modo fedele l'andamento della gestione e dell'amministrazione;
- possibilità di elaborazione dei dati anche complesse ai vari livelli: per singolo utente, gruppo di utenti o area organizzativa;
- possibilità di una gestione semplice nella creazione, aggiornamento ed utilizzo dei report tramite tools resi disponibili nel sistema e tali da garantire visualizzazioni e stampe o utilizzando sistemi di produttività individuale;
- capacità di sintesi successiva (drill-down) sui collegamenti e le integrazioni richieste dall'utente su tutti i dati disponibili nel sistema con la possibilità, a partire da un certo dato (ad esempio un Paziente), di poter visualizzare tutti i dettagli ad esso collegati nel tempo (come ad esempio; prescrizioni, referti, farmaci assunti, ecc.);
- possibilità di accesso tramite portale verso applicazioni esterne preesistenti o di nuova implementazione;
- completa autonomia gestionale, grazie a funzionalità che consentano di svolgere le attività proprie di ogni struttura in modo agevole (ad esempio le procedure di parametrizzazione sono user-friendly quanto il resto del sistema e quindi facilmente utilizzabili anche da personale non specialistico);
- sicurezza del sistema in termini sia di dati che di accessi, attraverso procedure standard di sistema e diversificate per la creazione e l'aggiornamento dei profili utente e per la gestione delle loro autorizzazioni nell'accesso al sistema;
- possibilità, all'interno del sistema integrato, di trasmettere e ricevere documenti (posta elettronica, XML, ecc.) da e verso tutte le strutture aziendali o esterne coinvolte nell'utilizzo del nuovo sistema;
- possibilità di collegamento/integrazione del sistema con procedure di gestione documenti (delibere, determinazioni, ecc.) e di firma elettronica.

3. LA PIATTAFORMA TECNOLOGICA

La piattaforma applicativa di TrakCare, è basata sulla piattaforma tecnologica, di cachè definito come un database postrelazionale.

Si parla di piattaforma tecnologica, in quanto il database nell'ambiente TrakCare, (come spiegato di seguito), svolge non solo la funzione store per la parte dati, ma anche di logica-applicativa.

Per meglio comprendere il perché Cachè venga definito database PostRelazionale (termine di derivazione più commerciale che di derivazione tecnologica) si espone brevemente alcune peculiarità e/o problematiche dei database tradizionali.

3.1 I SISTEMI RDBMS : DAL RDBMS A OODBMS

I sistemi RDBMS

Flat Lists

Nel modello relazionale si assume che tutti i dati siano gestibili come tabelle a due dimensioni.

Se sono richieste più dimensioni, le tabelle vengono viste su più livelli e correlate tramite ulteriori chiavi ("foreign key").

Strutture più complesse, come array multidimensionali e tabelle inseriti all'interno di altre tabelle, possono essere rappresentate, ma solo moltiplicando i livelli delle tabelle e le " foreign key ".

In strutture di questo tipo la ricostruzione degli elementi relativi ad un fatto che si vuole analizzare può richiedere una ricerca ricorsiva attraverso una molteplicità di tabelle ed indici, quindi tempo e risorse.

Supporto ai Binary Large Objects (BLOBs)

Il modello relazionale adatto alle tipiche applicazioni gestionali, si è dimostrato inadeguato per gestire il nuovo paradigma ad "oggetti".

Del tutto inadeguato poi se si tratta di oggetti di grandi dimensioni come testi, immagini, audio, video etc.

Il rischio è di procedere verso una strada che porta a risultati modesti perchè si forza il modello relazionale a qualcosa per cui non è stato concettualizzato.

Strutture ricorsive e "nested"

Il modello relazionale ha difficoltà a gestire righe di dati che si riferiscono ad altre righe di dati nella stessa tabella.

Strutture ricorsive come la "distinta base" o concetti come quello che un impiegato può essere manager di qualche altro e riportare ad un terzo non sono direttamente supportati.

Reti di dati intercorrelati

Un ordine viene normalmente scomposto in un insieme di tabelle correlate: ricostruire l'ordine potrebbe richiedere la ricerca dell'ordine, delle righe d'ordine, del cliente, di magazzini etc.. Questo comporta la conoscenza di quali tabelle interrogare e di come vanno combinati i risultati.

Non sono poi trattabili eventi non rappresentabili in forma tabellare: correlare un "problem ticket" con la e-mail di un cliente scontento non è possibile (o comunque difficile).

Dati per oggetti simili ma non uguali

Gli oggetti si possono presentare in gerarchie con attributi leggermente variati (per es: "veicoli a motore" è la generalizzazione di automobili, camion, furgoni).

Nel modello relazionale per ogni attributo è richiesta una tabella, il che comporta selezioni multiple e "nested" anche per una semplice reportistica che riguardi un oggetto.

Object Oriented DBMS

Le basi dati ad oggetti consentono una flessibilità estrema nella definizione dei dati e nella loro memorizzazione. Sono strettamente interconnesse con l'applicativo e non richiedono alcuna interfaccia (API).

Tuttavia i data model del DB e quello dell'applicativo devono essere identici, il che implica che se l'applicativo cambia, cambia anche il DB e che il DB non può essere condiviso da diverse applicazioni.

Database post-relazionali

Prima che si affermasse il modello relazionale esistevano dei DB che consentivano di supportare strutture "nested" e reti di dati interconnesse e che potevano forzare delle regole sui dati basandosi sulla loro semantica.

Questa capacità di associazione semantica è stata persa nel modello relazionale. I DBMS post-relazionali hanno ripreso e sviluppato queste tecnologie dei pre-relazionali ed aggiunto il supporto ad oggetti complessi. Tuttavia rimangono separati dall'applicazione attraverso una API, consentendo uno sviluppo e una manutenzione separata per l'applicazione e per il DB.

3.2 CACHE'

Caratteristiche di Cachè

Cachè grazie ai seguenti fattori risulta posizionarsi nel gruppo dei database post-relazionali:

- Supporto per i dati multidimensionali: Cachè supporta strutture multidimensionali con indici multi-dimensionali, fornendo prestazioni molto superiori rispetto ai sistemi relazionali.
- Approccio object oriented e supporto di SQL: Le definizioni dei dati sono basate sul modello ad oggetti (concetti di gerarchia, polimorfismo, etc.). Gli oggetti sono accessibili via servizi object oriented senza imporre un modello ad oggetti anche per l'applicativo. Cachè supporta appieno anche SQL, senza degrado di prestazioni.
- Cache's object script: Cache object oriented scripting language consente di definire gli oggetti e il loro comportamento e di inserirli in una programmazione ad oggetti. Permette di razionalizzare e automatizzare la gestione dei dati sul DB server, rendendo più semplice la manutenzione delle applicazioni e assicurando l'integrità dei dati e una migliore efficienza operativa.

	Caché	Relational
Data Model	Multidimensional	2 Dimensional
Access	SQL Object Direct	SQL

Potenza e rapidità di sviluppo delle applicazioni

La potenza viene fornita dal "Motore" di gestione dei dati multidimensionali, tutti i dati di Cachè vengono memorizzati in matrici multidimensionali ad accesso rapido, ideali per l'archiviazione di dati complessi e multifaccettati; l'accesso rapido è il motivo che permette alle applicazioni sviluppate su DB Cachè, rispetto a tecnologie tradizionali di supportare migliaia di utenti senza influire sulle prestazioni del sistema.

La produttività dipende dalla capacità di accedere ai dati in modalità diverse, ogni tecnologia eccelle in una particolare applicazione :

- la tecnologia a Oggetti è ottima per rappresentare dati complessi ed è compatibile con i linguaggi di programmazione Web;

- la tecnologia Relazionale è ideale per l'analisi dei dati e la creazione di report;

Gli sviluppatori di applicazioni lavorano meglio se possono utilizzare strumenti che già conoscono ed è per questo che la funzione di accesso multiplo e veloce di Cachè permette agli sviluppatori di usare lo strumento e la tecnologia più adatta in base al tipo di lavoro. Inoltre per le applicazioni che prevedono lo sviluppo di pagine Web, come oramai risulta quasi la norma, Cachè mette a disposizione un ulteriore strumento di incremento della produttività : la tecnologia CSP – Cachè Server Pages.

Prestazioni multidimensionali

Uno degli elementi chiave responsabili delle prestazioni e delle doti di scalabilità di Caché è rappresentato dal suo server dati multidimensionale ottimizzato per le elaborazioni transazionali.

“Multidimensionale” significa che i dati possono essere indicizzati in base al numero di parametri richiesti e non devono per forza essere costretti in tabelle composte di righe e colonne.

Questa maggiore flessibilità permette di creare modelli dati molto più complessi rispetto a quelli creati con la tecnologia relazionale e tali dati possono essere memorizzati e impiegati con modalità più naturali.

I database relazionali sono poco adatti a rappresentare dati complessi perché impongono il vincolo di suddividere e organizzare i dati in tabelle a due sole dimensioni.

Quando si utilizza la tecnologia relazionale per descrivere i dati multifunzionali del mondo reale, è necessario creare tabelle su tabelle, con il risultato che l'assemblaggio di tutte le informazioni necessarie per completare le transazioni richiede un enorme tempo di elaborazione.

Al contrario, i dati multidimensionali non devono essere “riasmblati” perché sin dall'inizio non vengono scomposti.

Il server dati multidimensionali Caché permette di eliminare le operazioni di elaborazione richieste dall'approccio relazionale e di aumentare in modo significativo la velocità di elaborazione dei processi transazionali.

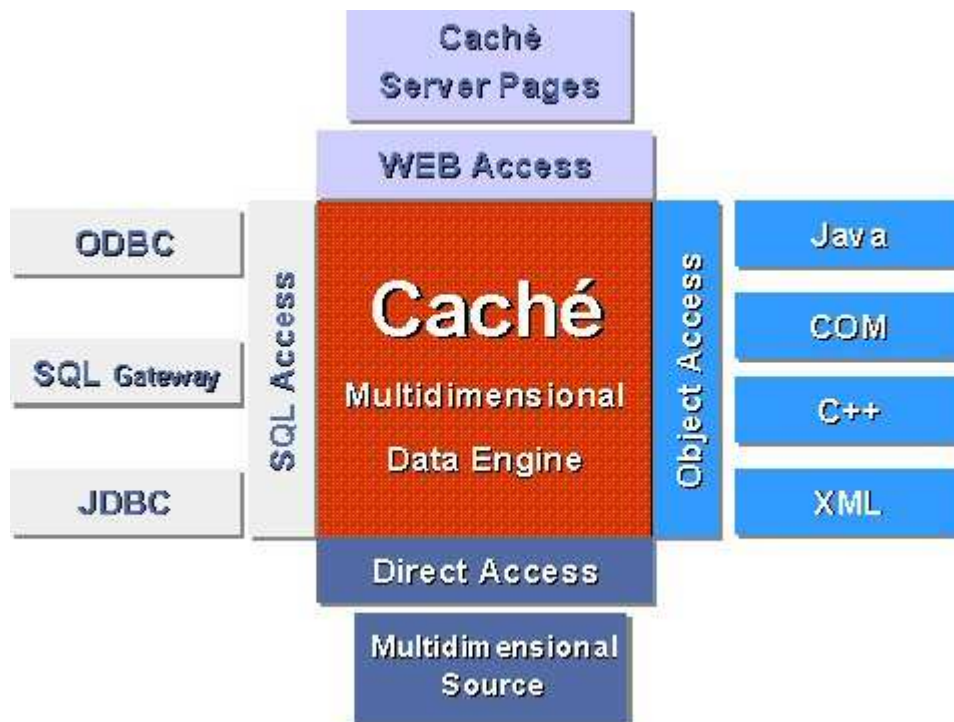
Accesso multiplo ai dati

Gli sviluppatori di applicazioni sono più produttivi quando utilizzano tecnologie e strumenti con i quali hanno dimestichezza.

Per questa ragione, nonostante i dati vengano memorizzati in matrici multidimensionali, l'accesso ai dati può essere effettuato in modalità diverse.

Inoltre sugli stessi dati è possibile impiegare simultaneamente tutte le modalità di accesso.

Grazie a questa particolarità, per il processo di migrazione in ambiente web, non si è dovuto migrare l'intero Sistema Informativo, ma il processo di migrazione è (e sarà) graduale, mantenendo in funzione contemporaneamente tutte e due le versioni di Medtrak e Trakcare.



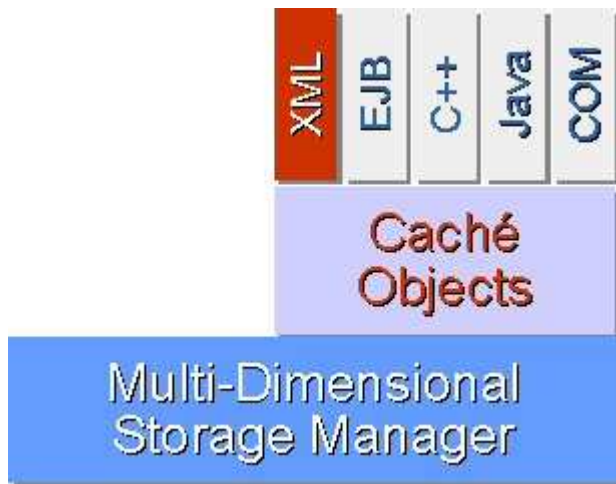
XML

Caché è particolarmente adatto per il formato XML, formato che si sta imponendo nello scambio di dati tra applicazioni.

Gli oggetti Caché possono fungere da rappresentazione diretta dei documenti XML e viceversa.

Caché offre le funzionalità seguenti:

- le classi Caché possono essere utilizzate automaticamente come documenti XML sotto forma di file o contenuto on-line.;
- le classi Caché sono in grado di generare automaticamente i file DTD XML corrispondenti.;
- i documenti XML possono essere trasformati automaticamente nell'oggetto Caché equivalente;
- I contenuti XML possono essere letti da file, flussi di dati o richieste HTTP.



Sviluppo delle applicazioni web

Caché non è solo un motore di database: è anche una tecnologia di sviluppo, specialmente per quanto riguarda le applicazioni Web.

Sia che si tratti di introdurre una nuova applicazione per il Web che di aggiornarne una esistente, il successo è direttamente proporzionale alla rapidità di sviluppo.

Caché è veloce, perché permette di sviluppare applicazioni Web usando gli strumenti preferiti da ciascuno sviluppatore.

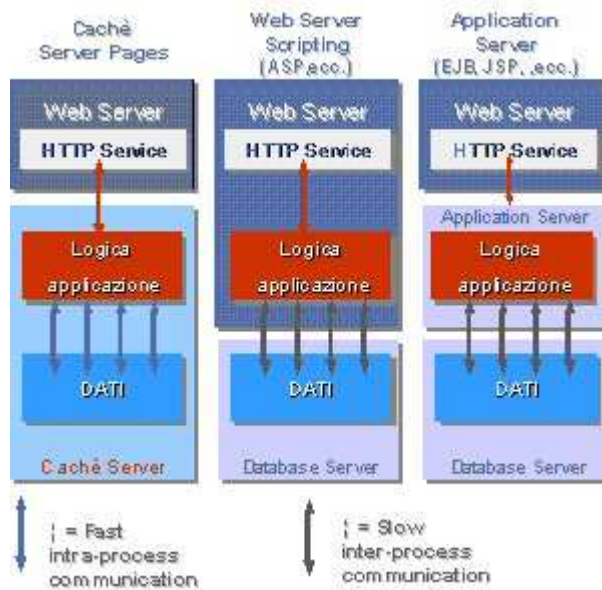
Le pagine server Caché possono essere create usando :

- Caché Studio;
- qualsiasi strumento di creazione di pagine Web di larga diffusione;
- un semplice editor di testi;

Caché è veloce perché le pagine ereditano il codice di gestione delle sessioni degli oggetti di livello sistema. È sufficiente scegliere il livello di protezione della sessione. Caché si occupa di tutto il resto.

È veloce, perché usando i tag applicativi di Caché si possono aggiungere facilmente funzioni alle pagine. È possibile scegliere tra i tag proposti da Caché oppure creare tag personalizzati su misura per le proprie applicazioni Web.

Le prestazioni e la scalabilità sono sempre stati due concetti chiave per gli sviluppatori di applicazioni ad alto contenuto transazionale, ma con l'avvento dell'e-business, questi due fattori sono diventati imprescindibili.



Le Caché Server Pages (CSP) vengono eseguite direttamente sul server dati in modo da velocizzare l'accesso ai dati residenti sul server.

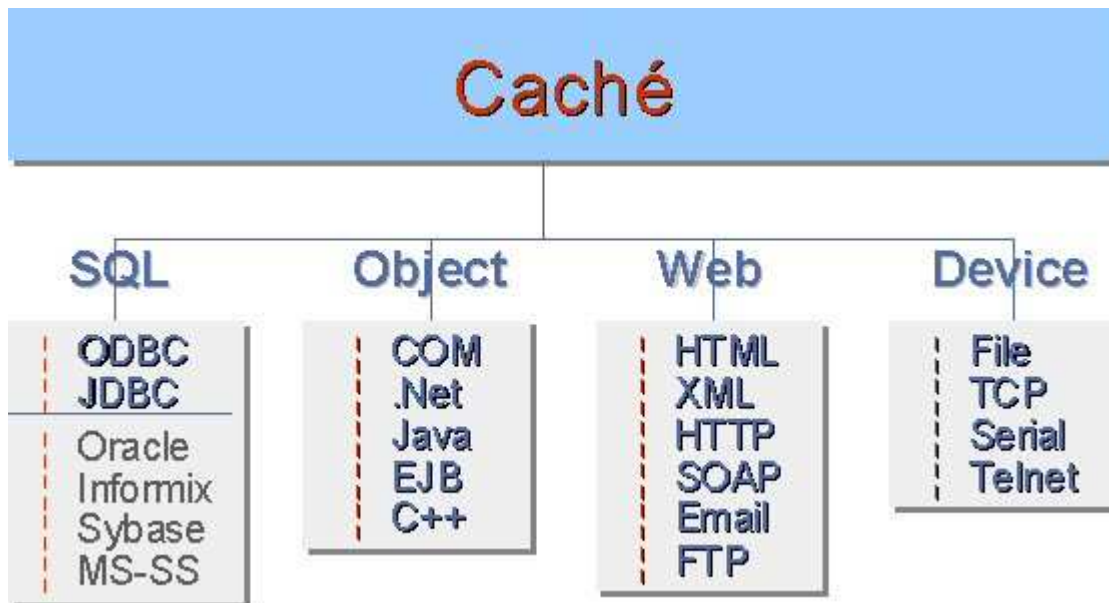
In questo modo la logica applicativa e i dati godono di una relazione molto stretta, il che costituisce un diretto vantaggio per la velocità delle comunicazioni.

Un ulteriore beneficio è costituito dalla maggiore scalabilità.

Il Server Web, non più impegnato nell'elaborazione della logica applicativa, è libero di gestire una maggiore quantità di richieste del browser.

Interoperabilità

L'interoperabilità che Caché permette di realizzare è completa e costantemente aggiornata ed in linea con le evoluzioni standard di riferimento.



3.3 COMPONENTI DI BASE DI CACHE'

Transactional Bit map Indexing

Caché fornisce pieno supporto per il bit-map indexing progettato per poter lavorare in modo estremamente veloce con grandi moli di dati.

Questo permette di sviluppare applicazioni che possono trattare dati complessi che normalmente sono realizzate attraverso sistemi specifici di business intelligence.

La sostanziale differenza è che si possono realizzare queste applicazioni operando direttamente sui dati in linea che stanno lavorando con le applicazioni tradizionali.

In altre parole non occorre fare procedure di estrazione di dati su altri data base (fuori linea) ma la tecnologia permette di operare sui dati correnti senza influenzare le performance del sistema.

Con altri database risulta difficoltoso aggiornare gli indici bit-map se gli accessi sono frequenti.

Infatti questa tecnologia viene usata per dati statici (solo in lettura) attraverso tecniche di data warehouse.

Caché permette di gestire questi indici in modo dinamico e quindi realizzare sistemi di interrogazione molto complessi e veloci.

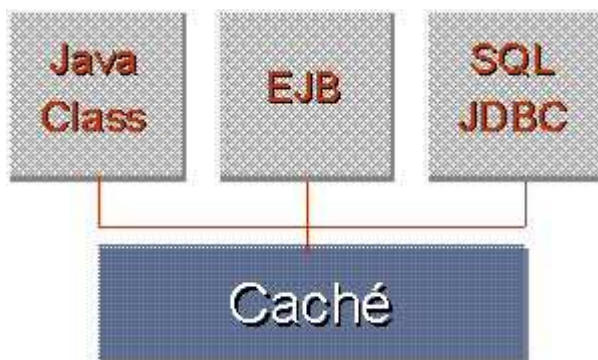




Java

Con tre differenti metodi per connettersi a Java, Caché è il database ad alte prestazioni ideale per applicazioni Java:

- i dati di Caché possono essere prelevati utilizzando il linguaggio SQL su canale JDBC;
- le classi di Caché possono essere proiettate come classi Java;
- infine è pienamente supportata la tecnologia Enterprise JavaBeans per lo sviluppo di applicazioni distribuite e transazionali Java based;



Caché supporta tutti gli approcci sopra elencati per raggiungere l'obiettivo di avere persistenza dei dati all'interno di applicazioni Java. Inoltre l'attività di proiezione è resa automatica in modo che gli sviluppatori Java possano avere persistenza dei dati senza dover generare grandi moli di codice.

Il motore di gestione dati multidimensionale di Caché ha un eccellente tempo di risposta SQL – fino a 20 volte più veloce degli ordinari database relazionali. Infine Caché contiene un driver JDBC di tipo 4, cosicché le applicazioni Java che già utilizzano SQL per connettersi a un deposito di dati, possano essere portate su Caché senza cambiamenti.

Server Pages

Le Caché Server Pages rendono disponibili tutte le funzionalità di Caché all'ambiente Web, ambiente ove ridotti tempi di sviluppo e adattabilità sono tanto importanti quanto performance e scalabilità.

Fra i punti di forza di tale tecnologia si riscontrano :

- Architettura Web snella e veloce poiché la logica applicativa risiede contestualmente ai dati;
- Contenuto dinamico delle pagine tramite utilizzo dei Caché Application Tag o di Hyper Events;
- Disponibilità di oggetti di gestione della sessione per semplificare il lavoro del programmatore nella gestione dello stato delle pagine Web.

Object Technology

La tecnologia a oggetti di Caché promuove una vista naturale dei dati senza limitarne le proprietà a tipi elementari. Le classi possono contenere altre classi o riferimenti ad altre classi; questo rende semplice costruire significativi modelli di dato.

Per via della Unified Data Architecture, tutte le classi di Caché sono automaticamente accessibili come tabelle relazionali via ODBC/JDBC. Sfruttando l'ereditarietà, le classi di Caché possono essere facilmente utilizzate in svariati ambiti, fra i quali :

- Caché Server Pages: una classe designata come Caché Server Page eredita automaticamente tutti i necessari metodi di gestione della sessione Web;
- XML: facendo ereditare proprietà e metodi dalla classe %XML.Adaptor, una classe è abilitata all'import/export XML;
- COM: tramite opportuno comando all'interno di Caché Studio si possono proiettare le classi di Caché come classi COM per l'utilizzo all'interno di strumenti di sviluppo come Visual Basic, Delphi o quant'altro compatibile con l'interfaccia COM;
- C++: analogamente si possono creare proiezioni C++ di classi Caché;
- Java: anche in questo caso si possono proiettare le classi Caché come classi Java;
- Enterprise Java Beans: Caché permette di fruire appieno della velocità di EJB senza dover effettuare noiose codifiche per il mapping fra le classi Java e le tabelle relazionali;

4. REPORT WEB XML

Il punto cruciale, per ogni sistema informativo, è sempre la produzione di reports. E' un elemento normalmente dato per scontato (il dato c'è, si tratta solo di tirarlo fuori), ma alla fine, l'esperienza insegna, che spesso risulta un collo di bottiglia per tutti i flussi informativi.

Per reportistica, non si intendono solamente reports riassunti da utilizzare ai fini del controllo di gestione e/o per la pianificazione direzionale, ma, nel caso di una Pubblica Amministrazione, fanno parte di questa categoria tutti i documenti rilasciati ai pazienti ai fini della trasparenza e/o certificazione dell'attività svolta (Riscontri di prenotazione, referti ecc.).

Benché la previsione futura, sia quella (come già accennato) di strutturare un Sistema Informativo che permetta di dare pratica all'Ospedale Paperless, l'eliminazione totale della carta sarà difficile, se non impossibile in alcuni casi.

Ad esempio, allo stato attuale, nel caso di una prestazione radiologica, al paziente, all'inizio (quando si presenta allo sportello per la prenotazione) viene dato un riscontro di prenotazione, successivamente all'esecuzione dell'esame, al paziente si darà una lastra ed un referto firmato dal medico.

Nell'ospedale paperless, al paziente al posto della lastra e del referto, potrà essere dato un cd-rom contenente la lastra in formato DICOM ed il referto firmato digitalmente, ciò non toglie, che all'atto della prenotazione al paziente si dovrà pur dare qualcosa di scritto, se non per vincolo di legge, almeno per cortesia, ai fini di un pro-memoria dell'appuntamento. Senza contare che quando qualcuno si presenta a pagare una prestazione, per forza di cose si deve rilasciare una ricevuta fiscale che necessariamente dovrà essere cartacea.

Come si vede, l'ospedale paperless, è tale solo nei flussi informativi interni, necessariamente, verso l'esterno ci deve essere un riscontro dell'attività svolta.

La produzione di questa reportistica poi è sempre svolta in ambito sportellistico, il punto di accesso dell'utenza, il collo di bottiglia che normalmente siamo abituati a notare in qualsiasi struttura pubblica: lo spool di stampa è l'unico traffico che non si può centralizzare, ma deve arrivare necessariamente dal server al client, spesso le stampanti sono lente e normalmente rappresentano l'unico elemento meccanico in movimento che facilmente si guastano. E' chiaro che in tale contesto la produzione di un banale report o riscontro, si scontra sempre con la velocità di produzione ma soprattutto l'affidabilità del sistema di produzione reportistica che si andrà a sviluppare.

Ecco perché, in ambiente web, la produzione di reportistica, non viene lasciata al solo strumento "Browser", ma si aggiungono una serie di strumenti di sviluppo e produzione che permettano di raggiungere gli standard qualitativi e di robustezza degli strumenti sviluppati ad hoc.

4.1 CHE COSA SONO XSLT E XSL-FO?

Uno degli ostacoli che si incontra nella produzione di report partendo da XML è il page layout. Esistono due vocabolari raccolti in altrettante W3C Recommendations che riguardano processi di trasformazione e renderizzazione di file XML.

The Extensible Stylesheet Language Transformations (XSLT) è un templating markup language utilizzato per descrivere come un processore debba creare un risultato trasformato a partire da informazioni XML.

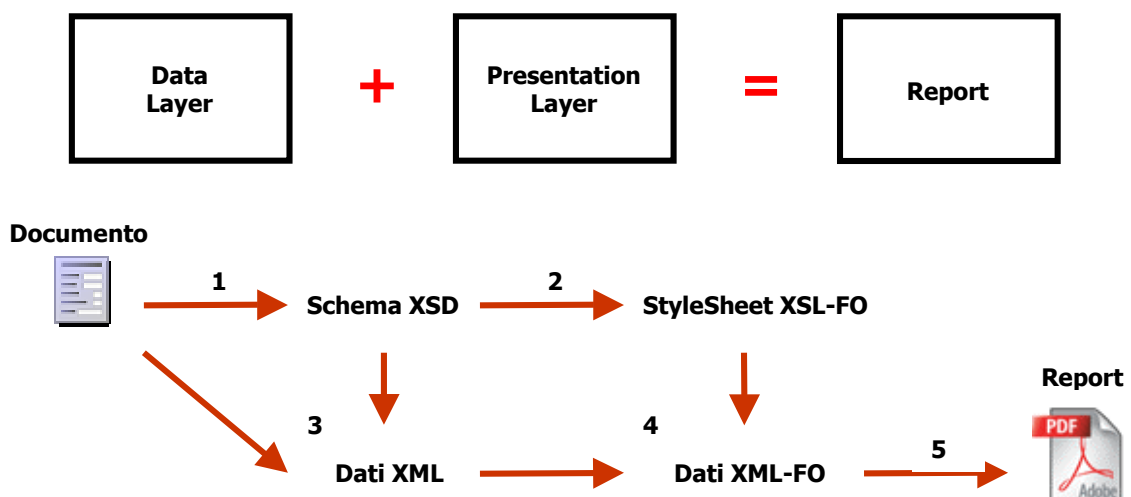
The Extensible Stylesheet Language Formatting Objects (XSL-FO) è un pagination markup language che descrive un vocabolario di renderizzazione contenente la semantica di informazioni formattate.

Che cosa è necessario per creare un documento PDF pronto per essere stampato a partire da un documento XML? Primo, il documento XML deve essere trasferito ad un processore XSLT con un appropriato stylesheet allo scopo di produrre un altro documento XML. Questo processore utilizzerà il namespace XSL-FO. Il secondo passo è quello di passare l'output del primo processore ad un XSL-FO formatter che produrrà il documento finale pronto per la stampa.

Il vantaggio di questo processo è che il documento sorgente XML resta neutrale rispetto ai formati di presentazione e può essere riutilizzato con altri XSLT stylesheets per la produzione di altrettanti media.

4.2 ARCHITETTURA

La soluzione proposta abbraccia l'architettura XSLT-based nella sua accezione FO (Formatting Object):



- Ad un generico documento corrisponde uno Schema XSD che ne definisce la struttura (1).
- Ad un generico Schema XSD corrispondono uno o più fogli di stile XSL che definiscono le trasformazioni XSL-FO da applicare alla struttura del documento (2).
- Ad un generico documento corrispondono dati strutturati in formato XML aderenti al corrispondente Schema XSD (3).
- Ai dati strutturati in formato XML viene applicata la trasformazione XSL-FO definita nel foglio di stile utilizzato (4).
- La trasformazione XSL-FO ottenuta viene renderizzata in pdf attraverso un Formatting Object Processor (5).

Creazione foglio di stile XSL

La generazione di fogli di stile è affidata a strumenti di produttività (Altova StyleVision, XSLFast Jcatalog Software) in grado di modellare graficamente lo stylesheet a partire da uno Schema XSD e/o da un working XML File (dati di test).

Trasformazione XSL-FO

La generazione di trasformazioni XSLT è affidata all'utilizzo del processore open-source Xalan.

Tale processore è integrato in Cachè e risulta accessibile dalla classe wrapper %XML.XSLT.Transformer.

Backporting su Cache 5.0.15 tramite ad-hoc Intersystems 3839.

Renderizzazione PDF

La renderizzazione di una trasformazione XSL-FO è affidata all'utilizzo del processore open-source FOP (Formatting Object Processor). Tale processore verrà integrato in Cache.

4.3 SOLUZIONE

Generazione report in Cache

Il procedimento di creazione report sfrutta, ed estende, le potenzialità built-in Cache relative alla gestione XML applicabile alle classi (%XML.Adapter), ed in particolare:

- Generazione automatica Schema XSD
- Produzione automatica dati XML aderenti allo Schema XSD
- Produzione automatica dati XML di test aderenti allo Schema XSD

a) Struttura del report

Ad ogni report corrisponde una classe che definisce, attraverso le proprie properties, la struttura del documento che descrive.

b) Common functionalities ereditate

La classe che definisce il report estende una nuova superclasse (web.zReport) dalla cui implementazione eredita le funzionalità descritte in precedenza.

Tale procedimento:

- Permette di ragionare in termini di properties evitando di dover costruire manualmente sia lo Schema XSD che il relativo file XML contenente i dati.
- Mette a disposizione in maniera automatica dati di test in formato XML (svincolati dall'estrazione che avverrà a run-time) utilizzati per verificare il report off-line durante la fase di design del foglio di stile.

c) Produzione dati reali a run-time

Il mapping tra la struttura della classe che definisce il report ed i dati da estrarre a run-time avverrà tramite ricopertura di un metodo ereditato dalla superclasse web.zReport che riceverà i parametri passati al report al momento della sua invocazione.

La superclasse web.zReport

La superclasse web.zReport estende %Registered e %XML.Adaptor ed implementa i seguenti metodi:

- XSDOutput()
Scrive su standard output lo Schema XSD della sottoclasse che lo istanzia. Metodo final la cui implementazione non può essere ridefinita nella sottoclasse.
- XSDOutputToFile(ByRef filename As %File)

Come XSDOutput ma con output su file anzichè su standard device. Metodo final la cui implementazione non può essere ridefinita nella sottoclasse.

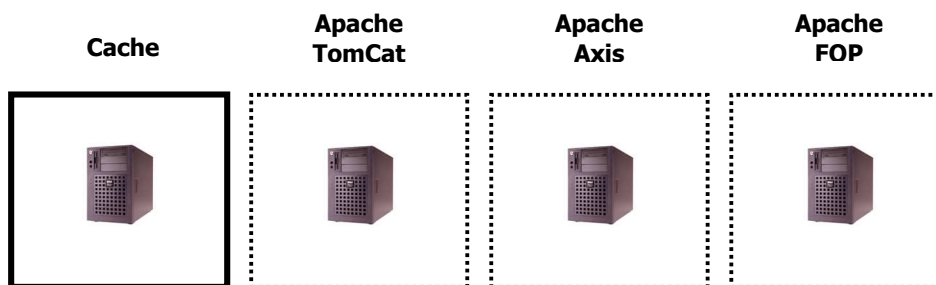
- XMLPopulate(count As %Integer = 10)
Genera in formato XML count istanze di test relative alla sottoclasse che lo istanzia. La tipologia di dati random creati dipende dal valore assegnabile al parametro XMLPOPSPEC di ogni proprietà. Il numero di istanze count generate è relativa alle collection presenti nella definizione della classe. Metodo final la cui implementazione non può essere ridefinita nella sottoclasse.
- XMLPopulateToFile(count As %Integer = 10, ByRef filename As %File)
Come XMLPopulate ma con output su file anzichè su standard device. Metodo final la cui implementazione non può essere ridefinita nella sottoclasse.
- XMLPopulateToStream(count As %Integer = 10, ByRef stream As %Stream)
Come XMLPopulate ma con output su stream anzichè su standard device. Metodo final la cui implementazione non può essere ridefinita nella sottoclasse.
- DataMapping(P0, P1, P2, P3, P4, P5, P6, P7, P8, P9 As %String)
Rappresenta il mapping tra la struttura della classe che descrive il report ed i dati da estrarre a run-time. Riceve i parametri P0-P9 con cui il relativo report viene invocato a run-time. Metodo non final la cui implementazione verrà ridefinita nella sottoclasse che lo implementa. Nella superclasse contiene solo un messaggio di errore.
- XMLOutput()
Genera in formato XML le istanze, create attraverso il metodo DataMapping, della classe Registered che lo implementa. Metodo final la cui implementazione non può essere ridefinita nella sottoclasse.
- XMLOutputToFile(ByRef file As %File)
Come XMLOutput ma con output su file anzichè su standard device.
- XMLOutputToStream(ByRef stream As %Stream)
Come XMLOutput ma con output su stream anzichè su standard device.
- TransformFOFile(xmlFile, xslFile As %File, ByRef filename As %File)
Applica la trasformazione XSL-FO definita nello stylesheet xslFile ai dati xmlFile producendo output su file.

- TransformFOStream(xmlStream, xslStream As %Stream, ByRef stream As %Stream)
Come TransformFile ma con input ed output su stream.
- RenderPDFFile(foFile As %Stream, ByRef filename As %File)
Renderizza in PDF il file foFile contenente la trasformazione XSLT-FO producendo output su file. Utilizza il web service client di integrazione con FOP.
- RenderPDFStream(foStream As %Stream, ByRef pdfStream As %Stream)
Come RenderPDFFile ma con output su stream.

Integrazione Cache - FOP

Il processore open-source FOP (Formatting Object Processor) è sviluppato in tecnologia Java.

L'integrazione con Cache prevede la realizzazione di un web service Java e di un proxy Cache (web service client) di invocazione. L'ambiente tecnologico di riferimento è il seguente:



Integrazione con TrakCare

L'integrazione con TrakCare avviene mediante un sistema di parametrizzazione dei report (nelle apposite code-table) ed un sistema di binding dei medesimi con i menu bars (e/o componenti) di stampa per la relativa esecuzione.

Sistema di parametrizzazione report

E' composto dal nuovo componente : XML Report Manager che permette la definizione di un report.

Proprietà previste:

- Codice
- Descrizione
- Classe di riferimento (che estende la web.zReport) su cui si basa il report
- Note
- Descrizione parametri P0-P9 attesi dal metodo che effettua l'estrazione dati (DataMapping)

Funzionalità previste:

- Genera Schema XSD : genera lo Schema XSD del report selezionato. Output all'interno della virtual directory /custom/web/reports/
- Genera Working File XML : produce un working file XML aderente allo Schema XSD del report selezionato utilizzando dati di test generati casualmente. La tipologia dei dati generati casualmente può essere comandata utilizzando il parametro XMLPOPSPEC. Output all'interno della virtual directory /custom/web/reports/
- Upload StyleSheet XSL : consente l'upload di un foglio di stile per il report selezionato. La memorizzazione avverrà all'interno del database. Le preferenze relative all'utilizzo del foglio di stile verranno salvate secondo i medesimi criteri di layout/columns: Site, Group, User.
- Test report con dati di test : consente, qualora sia stato effettuato upload di un foglio di stile, di lanciare il report in modalità test. La componente dati sarà generata a run-time con le stesse modalità della generazione del working File XML.
- Test report in dati di produzione : consente, qualora sia stato effettuato upload di un foglio di stile, di lanciare il report in modalità live. La componente dati sarà generata a run-time valorizzando mediante interfaccia utente i parametri P0-P9. Per effettuare questo tipo di test la relativa classe su cui si basa il report dovrà aver implementato il metodo DataMapping.
- StyleSheet Preferences : permette la consultazione/eliminazione della parametrizzazione dei fogli di stile di un report. La ricerca avverrà per User, Group, Site e Report.

Sistema di binding report con menu bar items

E' composto dal seguente componente:

XML Report Binding

Permette di associare ad un menu item (child di un menu bar) il relativo Report da lanciare e di parametrizzarne il valore dei parametri P0-P9 con riferimento al particolare contesto di esecuzione. A run-time il report verrà

prodotto utilizzando lo stylesheet opportuno secondo la gerarchia di preferences Site/Group/User impostabile.

4.4 ESEMPIO

Creazione di un report che contiene una lista di appuntamenti, composto dalle seguenti proprietà:

- Data
- Unità operativa
- Risorsa
- Appuntamenti -> Servizio
Ora inizio
Ora fine
Durata

Modellato in classi:

```
/// Appointment List
Class Report.AppointmentList Extends web.zReport
{
    Property Date As %Date;

    Property Location As %String (XMLPOPSPEC = "CTLoc");

    Property Resource As %String (XMLPOPSPEC = "RBResource");

    Property Appointments As Appointment [ Collection = list ];
}

/// Appointment definition
Class Report.Appointment Extends web.zReport
{
    Property Service As %String(XMLPOPSPEC = "RBCServices");

    Property TimeStart As %Time;

    Property TimeEnd As %Time;

    Property Duration As %Integer;
}
```

Il parametro XMLPOPSPEC (XML Populate Specification) consente di comandare la generazione di dati di test (Location utilizzerà dati random da CT_Location, Resource da RB_Resource, Service da RBC_Services,...).

Una volta creata una nuova entry nell'XML Report Manager basato sulla precedente classe, si è in grado di:

- Generare automaticamente il relativo file di Schema XSD:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<s:schema xmlns:s="http://www.w3.org/2001/XMLSchema">
  <s:element name="AppointmentList">
    <s:complexType>
      <s:sequence>
        <s:element name="Date" type="s:date"/>
        <s:element name="Location" type="s:string"/>
        <s:element name="Resource" type="s:string"/>
        <s:element name="Appointments">
          <s:complexType>
            <s:sequence>
              <s:element name="Service" type="s:string"/>
              <s:element name="TimeStart" type="s:time"/>
              <s:element name="TimeEnd" type="s:time"/>
              <s:element name="Duration" type="s:long"/>
            </s:sequence>
          </s:complexType>
        </s:element>
      </s:sequence>
    </s:complexType>
  </s:element>
</s:schema>
```

Generare automaticamente il relativo working XML file contenente dati di test random:

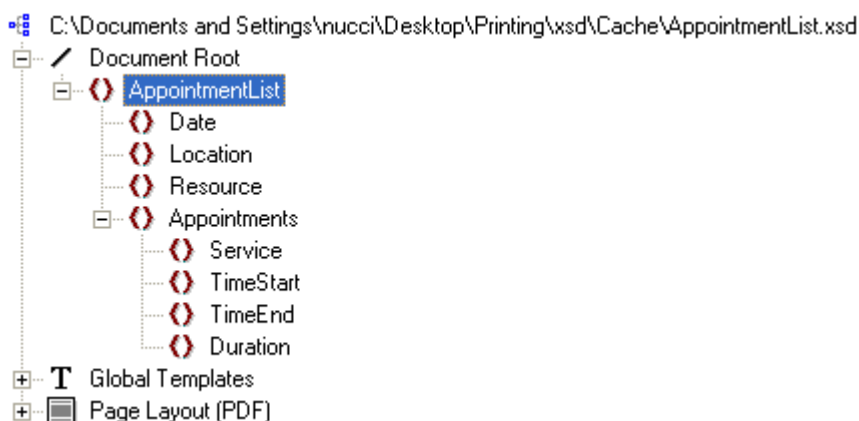
```

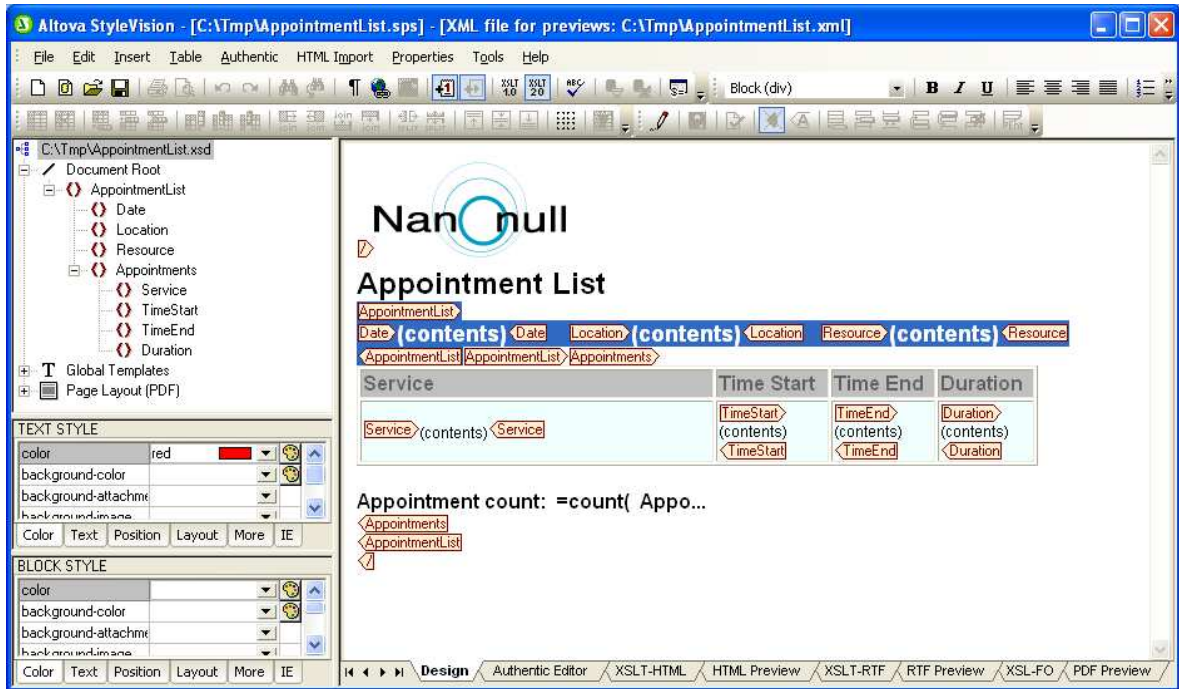
<?xml version="1.0" encoding="ISO-8859-1" ?>
<AppointmentList>
  <Date>10-05-2006</Date>
  <Location>TH-Radiologia</Location>
  <Resource>TH-RX TAC esterni</Resource>
  <Appointments>
    <Service>TAC DEL BACINO</Service>
    <TimeStart>09:00</TimeStart>
    <TimeEnd>09:30</TimeEnd>
    <Duration>30</Duration>
  </Appointments>
  <Appointments>
    <Service>TAC OSSEA</Service>
    <TimeStart>09:30</TimeStart>
    <TimeEnd>10:14</TimeEnd>
    <Duration>45</Duration>
  </Appointments>
  <Appointments>
    <Service>TAC SPINALE</Service>
    <TimeStart>10:15</TimeStart>
    <TimeEnd>11:00</TimeEnd>
    <Duration>45</Duration>
  </Appointments>
  <Appointments>
    <Service>TAC DELLO SPAZIO INTEROSOMATICO</Service>
    <TimeStart>11:00</TimeStart>
    <TimeEnd>11:30</TimeEnd>
    <Duration>30</Duration>
  </Appointments>
</AppointmentList>

```

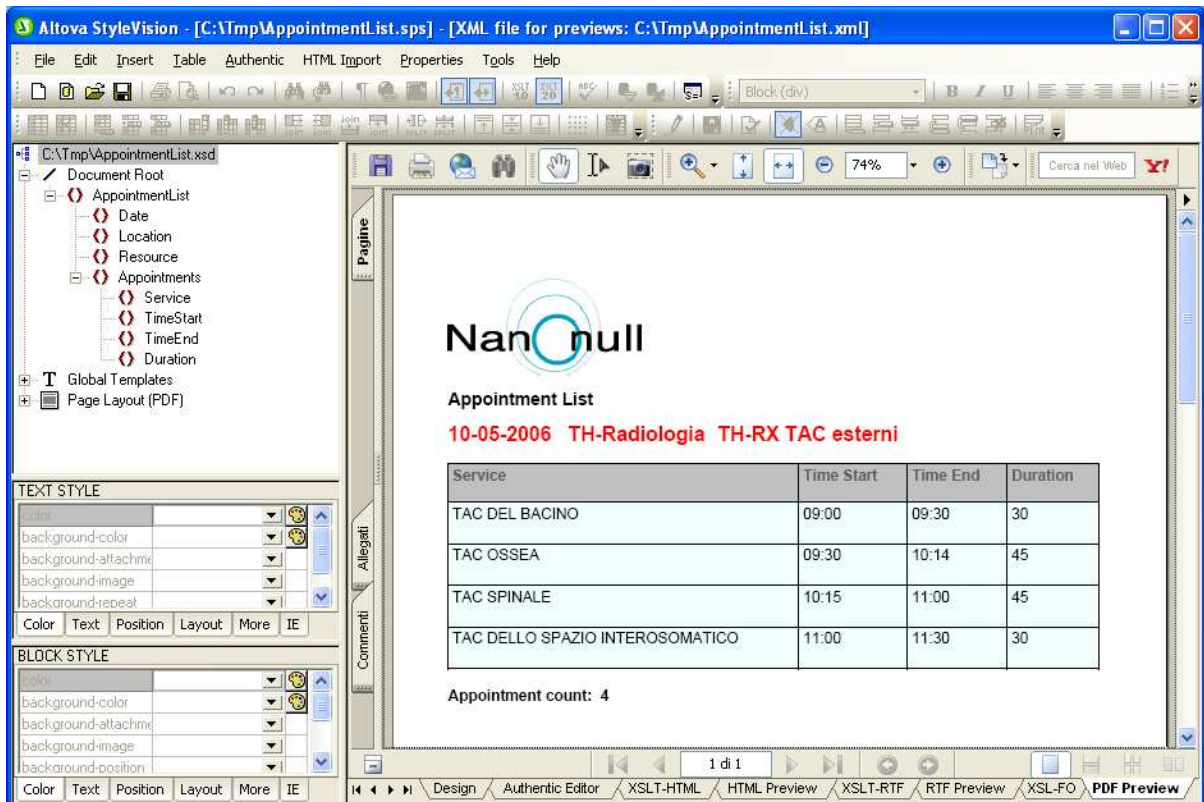
Recuperare i files creati per comporre lo stylesheet con strumenti di produttività:

Altova StyleVision - [C:\Tmp\AppointmentList.sps] - [XML file for previews: C:\Tmp\AppointmentList.xml]

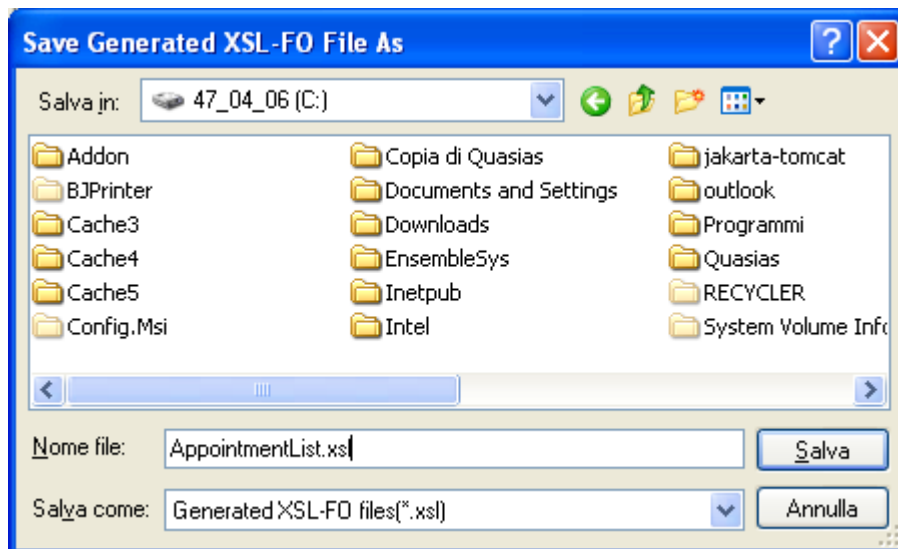




Testare il report con i dati di test contenuti nel working XML file:



Salvare su file lo stylesheet creato:



E farne poi upload sul server per l'utilizzo a run-time.

L'estrazione dei dati reali avviene, invece sviluppando l'implementazione del metodo DataMapping:

```

/*
 * PO = user ID, P1 = location ID, P2 = resource ID, P3 = date internal format
 */
ClassMethod DataMapping(PO, P1, P2, P3, P4, P5, P6, P7, P8, P9 As %String)
{
    set obj=##class(Report.AppointmentList).%New()

    set obj.Date=$zdate(P3,4,,4)
    set obj.Location=$list(##class(web.CTLoc).GetCodeDescriptionFromId(P1),1)
    set obj.Resource=$list(##class(web.RBResource).GetCodeDescriptionFromId(P2),1)

    set queryString=" SELECT APPT_Duration AS Dur, APPT_RBCServ_DR->SER_Desc AS Service, ... " _
        " FROM RE_Appointment " _
        " WHERE APPT_AS_ParRef->AS_RES_ParRef = ? AND APPT_AS_ParRef->AS_Date = ?"

    set result=##class(%ResultSet).%New("%DynamicQuery:SQL")
    do result.Prepare(queryString)
    do result.Execute(P2,P3)

    for quit:'result.Next() Do
    .
    . set apptobj=##class(Report.Appointment).%New()
    . set apptobj.Service=result.Data("Service")
    . set apptobj.TimeStart=result.Data("TimeStart")
    . set apptobj.TimeEnd=result.Data("TimeEnd")
    . set apptobj.Duration=result.Data("Duration")
    .
    . do obj.Appointments.Insert(apptobj)
    . do apptobj.%Close()
    .

    do obj.XMLExport()
    do obj.%Close()
}

```

In questo esempio l'oggetto Report.Appoinment è stato popolato manualmente riempiendo le relative proprietà: l'obiettivo finale è invece quello di consentirne il riutilizzo. Questo target viene raggiunto implementando il metodo DataMapping in ciascuna entità (ad esempio Report.Appoinment) e riutilizzarlo da tutte le altre (ad esempio Report.AppoinmentList).

5. CONCLUSIONI

5.1 ALTA AFFIDABILITA'

Il passaggio del Sistema Informativo Aziendale in ambiente Web Based, che ha rappresentato il tema principale di questa relazione, è solamente uno dei temi che vede coinvolto il Servizio Risorse Informatiche per garantire il salto "culturale" verso il primario obiettivo della Direzione Aziendale che prevede entro sei anni la costituzione dell'ospedale unico "Paperless&Filmless".

Se da un lato l'intera Azienda si trova a dover affrontare, un salto "culturale" per permettere l'informatizzazione di tutti i processi sanitari, dall'altro, il Servizio Risorse Informatiche si vede coinvolto nello sviluppo di un'infrastruttura che riesca a supportare tale salto.

Se precedentemente il Sistema Informativo forniva servizi, prevalentemente definiti di front-office o di back-office ma in orari presidiati dal personale del Servizio Risorse Informatiche, il nuovo Sistema Informativo, prevede di fornire servizi ad operatori che lavorano su turni di 24 ore al giorno su 7 giorni alla settimana (che in campo dei servizi informatici viene definito 7x24). Un sistema di questo tipo, oltre che ad un elevato grado di affidabilità, deve garantire un elevato grado di resilienza⁵.

Nella figura 1, si è schematizzato l'architettura ad Alta Affidabilità che l'Azienda ULSS 4 ha implementato in funzione di disporre di un sistema funzionante 7x24.

L'architettura ad Alta Affidabilità, così come evidenziato nella precedente figura, è composta da quattro elementi principali :

- Sottosistema Storage SAN/NAS (NetApp)
- Cluster di DataBase (nodo1, nodo2)
- Sistema di virtualizzazione VMWARE (vmware1, vmware2)
- Server Farm Application Server

Ciascuna macchina, che compone i vari sottosistemi, è strutturata in modo da non presentare "single point of failure" : dischi, schede, alimentatori e cpu sono ridondati; ogni macchina è poi affiancato da una macchina speculare in modo da garantire che la macchina stessa non rappresenti un possibile punto di "failure"; infine ciascuna macchina è connessa al sistema, con un sistema di connessione ridondato anch'esso.

⁵ Il termine resilienza in biologia è inteso come la "capacità di ripararsi in seguito di un danno", nell'informatica si intende la capacità a fronte di un guasto, non di ripararsi ma di intervenire autonomamente per sopperire all'elemento guasto.

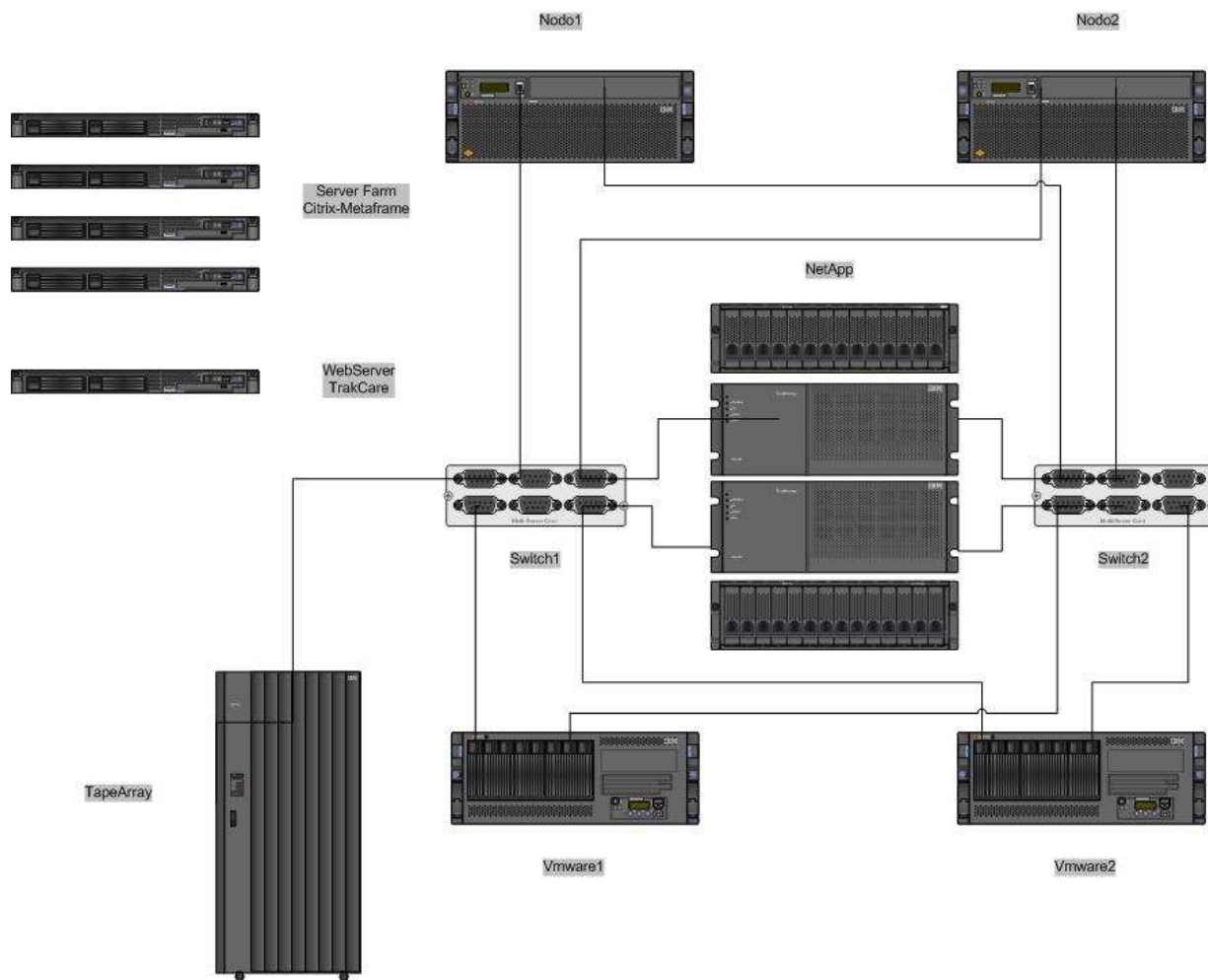


Fig. 1 : Architettura Alta Affidabilità Azienda ULSS 4

Sottosistema Storage SAN/NAS (NetApp)

Il sottosistema storage, IBM NetApp N5200, è un sistema sia SAN⁶ che NAS⁷ anche se nell'architettura ad Alta Affidabilità si è scelto di utilizzare esclusivamente le funzionalità di SAN.

Il sistema è composto da due unità (filer), ciascuna delle quali è dotata di due controller che lavorano in modalità Active-Active sui sottosistemi a dischi.

Le due unità lavorano in modalità cluster Active-Active in modo da presentarsi sulla SAN come un'unica unità.

In questo modo, viene garantita la continuità di servizio a fronte della caduta, sia di un singolo controller, sia di una singola unità.

⁶ SAN : Storage Area Network

⁷ NAS : Network Attached Storage

Il sottosistema, utilizza connessioni basate sullo standard Fibre Channel (FC) sia per connettere le unità, sia (anche se in parte) i singoli dischi. Il sistema è composto da 4 stringhe di dischi, di cui 2 in Fibra e 2 in SATA (Serial ATA) per un totale di 10 Terabyte.

Cluster di DataBase (nodo1, nodo2)

Il Cluster di Database è composto da due nodi (IBM p630) ciascuno dei quali è dotato di 4 processori 64 bit e 8 Gbyte RAM. Ciascun nodo monta il sistema operativo Unix di IBM (AIX 5.2 in aggiunta al software di clusterizzazione HACMP sempre di IBM).

Sistema di virtualizzazione VMWARE (vmware1, vmware2)

Tale sistema è stato introdotto, in quanto ormai da anni, il Servizio Risorse Informatiche ha adottato la politica di consolidamento dei server e degli applicativi.

Se da un lato, nell'ambito dell'area Unix, il consolidamento è stato pressoché totale, dall'altro lato, ossia nel mondo Microsoft, il consolidare ha sempre trovato un grosso ostacolo, legato al fatto che per il mondo Microsoft, vige una silente regola che prevede "un servizio un server"; quest'ultima pur non essendo una regola scritta, è una regola molto praticata, in quanto sullo stesso sistemi operativo Microsoft è difficile installare più servizi, senza che questi non interferiscano tra loro.

Benché la politica dell'Azienda ULSS 4, sia sempre stata quella di basare la propria infrastruttura su sistemi operativi UNIX, si è scontrata, con la difficoltà (se non l'impossibilità a volte) di ritrovare nel mercato, applicativi funzionanti interamente in tale ambiente.

Il dotarsi di una struttura in cluster per il database, si è sempre scontrato con il rischio di dotarsi poi di una struttura per la parte Application Server "non clusterizzata" diventando poi il classico "single point of failure" del sistema. L'alternativa a ciò, significava replicare un'architettura cluster in ambiente microsoft, con l'aumento delle problematiche di gestione connesse, ma soprattutto con l'aggravante della silente regola "un servizio un server" (che nel caso di clusterizzazione diviene "un servizio un cluster").

L'infrastruttura Virtual Center di VMWARE, è un sistema che permette la virtualizzazione de vari server fisici, garantendo, anche a piccoli server applicativi la possibilità di usufruire di un sistema ad alta affidabilità che ne condivide le proprie risorse.

L'infrastruttura è composta da due nodi (vmware1 e vmware2) ciascuno dei quali dotato di 4 processori (tecnologia intel a 32 bit) e di 16 Gbyte di RAM i due nodi sono dotati di elementi ridondati e connessi poi in SAN in modo da poter condividere i dati.

I dati in questo caso rappresentano le macchine virtuali (nella logica di vmware un server è un file), e il fatto di poter essere condivise tra i due nodi, garantisce l'alta affidabilità del sistema.

L'infrastruttura Virtual Center, garantisce la condivisione delle macchine virtuali in modo dinamico, nel senso che il trasferimento di una macchina virtuale da un nodo all'altro, avviene senza interruzione di servizio per la macchina trasferita.

Il trasferimento, può avvenire in modo automatico, per failure di uno dei due nodi o per bilanciamento di carico.

Server Farm Application Server

Il sistema di remotizzazione delle applicazioni Citrix-Metaframe, tramite il servizio di "Load Balancing", oltre a garantire la distribuzione del carico in modo omogeneo sui vari server componenti la Server Farm, garantisce, caso di caduta di uno dei server, il trasferimento dei client nei rimanenti server.

Nel caso dell'Azienda ULSS 4 allo stato attuale i server della Server Farm sono dei server esclusivamente fisici, anche se è prevista di trasferire parte dei server in ambiente virtuale.

Sottosistema di Backup

Rappresentato dalla Tape Library IBM TS3310, unità composta da tre driver di scrittura/lettura e da 80 slot LTO 3.

Assieme alla struttura software Tivoli Storage Manager, garantisce l'infrastruttura di backup del Sistema Informativo, ma soprattutto è un elemento essenziale per l'implementazione di un protocollo Aziendale di Disaster Recovery.

5.2 ...CONCLUDENDO

In conclusione, si vuole sottolineare, come l'implementazione del nuovo Sistema Informativo TrakCare, ha costituito per l'Azienda ULSS, un momento di revisione non solo per il Servizio Risorse Informatiche che si vede coinvolto nel processo come un fornitore di Servizi interno, ma anche per l'intera Azienda che si vede coinvolta in un processo globale di revisione di tutti i flussi dell'attività Sanitaria.

Sono momenti come questi che permettono ad un Azienda Sanitaria, di rivedere in modo critico i propri processi produttivi ed organizzativi per poter restare al passo di una società che richiede sempre di più maggior efficienza ed efficacia per la gestione della propria salute.

BIBLIOGRAFIA

- Ed Tittel, Norbert Mikula, Ramesh Chandak (1998), XML For Dummies, Apogeo, Milano
- Michael Floyd (2000), Costruire siti Web con XML, Tecniche Nuove , Milano
- Peter Wainwright,(2004) Apache Guida Professionale, Apogeo, Milano
- Apache FOP – Formatting Object Processor
<http://xmlgraphics.apache.org/fop/>
- Apache TomCat
<http://tomcat.apache.org/>
- Apache Axis
<http://ws.apache.org/axis/>
- Altova StyleVision
http://www.altova.com/products_xsl.html
- Jcatalog Software XSLFast
<http://www.xslfast.com/>