Soluzioni OpenSource per la linearizzazione del problema di integrazione di applicativi nei Sistemi Informativi Ospedalieri.

Riccardo Triunfo CRS4 riccardo.triunfo@crs4.it
Luca Lianas CRS4 luca.lianas@crs4.it
Alessandro Sulis CRS4 alessandro.sulis@crs4.it
Gianluigi Zanetti CRS4 gianluigi.zanetti@crs4.it
Nicoletta Delogu A.O. "G.Brotzu" nicolettadelogu@aob.it

SOMMARIO

L'informatizzazione dei processi e delle attività ospedaliere è un aspetto essenziale per migliorare la gestione delle risorse delle aziende sanitarie e per limitare i fattori di rischio legati al passaggio quotidiano di informazioni tra le diverse unità funzionali.
A tutt'oggi l'ospedale "medio" in Italia ha una struttura frammentata composta da "isole informatiche" in cui l'attività è quasi interamente automatizzata grazie a strumentazioni elettroniche e software applicativi, come ad esempio le Radiologie con i sistemi RIS/PACS o i laboratori con auto-analizzatori e provette associate direttamente ai pazienti grazie ai codici a barre o RFID.
La principale sfida da affrontare è proprio quella di integrare le informazioni e i processi e di offrire agli utenti interni delle strutture, agli operatori della salute e ai pazienti l'accesso a servizi.
Le principali problematiche di integrazione sono legate alla presenza dei diversi applicativi, forniti da vari *Vendors*, con caratteristiche differenti di codifica dei propri dati e la cui integrazione deve garantire invece la condivisione di un medesimo linguaggio.
Fortunatamente anche in Italia si stanno diffondendo linguaggi e protocolli per semplificare e standardizzare lo scambio di dati sanitari sia amministrativi che clinici, come *HL7, DICOM ed IHE*.
Illustreremo brevemente un proficuo esempio di utilizzo di software Open Source nella Pubblica Amministrazione, nel contesto relativo alla gestione integrata dei flussi informativi ospedalieri, che serve da collante e da base per l'integrazione di software commerciali e proprietari.

PAROLE CHIAVE

HL7, Mirth, interoperabilità, software ospedalieri, informatica clinica, Open Source

IL CONTESTO DI RIFERIMENTO

La medicina, come abbiamo accennato precedentemente, ha superato la "*soglia digitale*" imponendosi come uno dei settori trainanti del mercato ICT e dei principali consumatori di tecnologia.
La natura stessa della pratica clinica è incentrata sostanzialmente sulla gestione e l'analisi dell'informazione; la crescita esponenziale della capacità tecnologica di acquisire dati biologici in formato digitale, da cui è possibile estrarre estese informazioni utili a fini diagnostici e terapeutici, sono i due aspetti cardinali di questa lenta ma sostanziale trasformazione.
Due esempi visibili di questo cambiamento sono la maggiore diffusione di PC all'interno degli ospedali e l'enorme quantità di messaggi che viaggiano sulle reti interne delle strutture sanitarie e che vengono utilizzati per scambiare informazioni tra le varie componenti dei sistemi informativi, clinici e non. Il protocollo HL7 cerca di uniformare l'eterogeneità dei dati scambiati; tuttavia, nonostante sia uno standard, lascia ampi spazi di discrezionalità implementativa e di conseguenza comporta diverse difficoltà di integrazione che portano spesso a situazioni complesse dal punto di vista logistico ed economico. Questo si verifica in particolare quando, come usualmente accade nelle strutture sanitarie, i componenti del sistema informativo clinico sono stati comprati in tempi e da fornitori diversi. Nel caso tipico, queste difficoltà scalano con il quadrato del numero dei fornitori (n^2 integrazioni da effettuare, indicando con n il numero dei sottosistemi software). Qui illustreremo come si può linearizzare il problema sfruttando soluzioni software open source ed un moderato investimento in personale esperto, riportando l'esperienza sul campo presso l'Azienda Ospedaliera Brotzu di Cagliari. I componenti coinvolti in questo case study sono i seguenti:

- **Helise:** attuale Sistema Informativo centrale che gestisce ADT, CUP, Ticket, richieste da reparto; basato su database relazionali Oracle 7 e form su terminali carattere. In via di dismissione.
- **DHE:** nuovo sistema informativo, basato su un middleware di servizi (.NET, Java) per l'accesso alla base dati (Oracle 10). In via di adozione.
- **Mirth:** gateway HL7 Open Source. Riceve e inoltra messaggi HL7, si aggancia a qualunque sorgente dati.
- **RIS/PACS:** sistema fornito dalla Ebit, basato su SQL Server e sistemi Web ASP/.NET
- **Sale Operatorie:** software applicativo per la gestione delle sale operatorie basato su Oracle 7.
- **Pronto Soccorso:** software applicativo per la gestione del pronto soccorso. Da questo vengono generati la maggior parte dei flussi (richieste esami, ADT) verso il sistema informativo centrale. Si basa su database Oracle 10 e supporta HL7 nativamente.

E' evidente che tutte le componenti coinvolte nell'integrazione sono estremamente eterogenee e la loro connessione risulta spesso complessa. Lo schema che segue rappresenta l'attuale sistema informativo senza l'utilizzo di un gateway HL7:

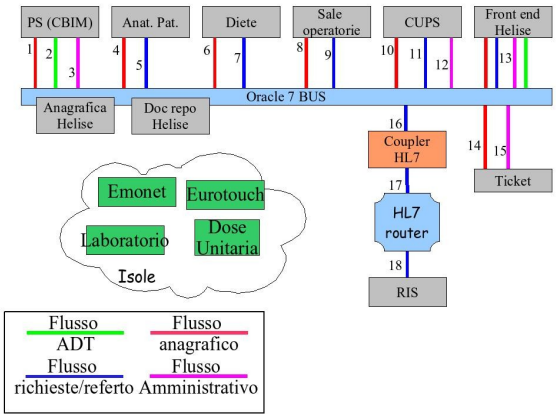
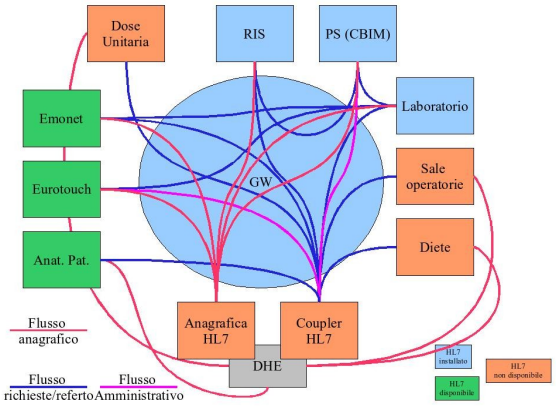


Illustrazione 1: Struttura attuale sistema informativo presso l'Azienda Ospedaliera Brotzu

A regime, tutte le comunicazioni dovrebbero passare attraverso il gateway HL7 Mirth, passando ad uno schema architetturale a stella, che diminuisce notevolmente il numero di interfacce da creare e la complessità globale del sistema:



MIRTH

Mirth (http://www.mirthproject.org) è uno strumento Open Source per la gestione, trasformazione e routing dei messaggi HL7. E' interamente sviluppato in Java e dispone di strumenti di sviluppo facili da usare ed allo stesso tempo molto potenti.
La gestione dei flussi informativi è affidata ai *canali*: un canale può essere inteso fondamentalmente come un componente in grado di prendere in ingresso informazioni di vario tipo (interrogazioni da database, file XML, oltre che naturalmente messaggi HL7), gestirle attraverso l'applicazione di una serie di filtri e trasformazioni, e mandarle in uscita verso i vari sistemi, oppure verso altri canali. Queste operazioni possono essere eseguite in modo semplice grazie all'utilizzo delle HL7 API, librerie Java per la gestione dei messaggi HL7, integrate in Mirth: queste ultime forniscono tutto il necessario per manipolare i messaggi HL7 tramite strumenti comuni agli sviluppatori (file XML, vettori) e di conseguenza poterli modificare facilmente.

I filtri e le trasformazioni vengono eseguiti tramite degli script che utilizzano le suddette librerie Java, e definiscono i passi e le trasformazioni da operare sui messaggi. In uscita le informazioni possono essere salvate in locale oppure instradate verso altri canali.
Mirth è uno strumento molto potente e versatile in quanto dispone degli strumenti necessari a manipolare i messaggi HL7 da un punto di vista puramente sintattico e degli strumenti necessari ad implementare una vera e propria infrastruttura di rete per la gestione delle comunicazioni tra i vari sottosistemi ospedalieri: ciascuno di questi può essere interfacciato tramite vari canali di comunicazione da e verso l'esterno (*gateway* per la gestione dei messaggi *inbound* e *outbound*).
L'architettura di Mirth è composta essenzialmente da due parti principali:

1. Un server che mette a disposizione tutti gli strumenti per la progettazione e l'utilizzo dei vari flussi, sul quale è integrato un database di riferimento per la gestione dei log e delle componenti dell'applicazione (canali, script, listeners e senders, ecc.).
2. Un client che fornisce l'interfaccia grafica per la gestione dell'applicazione e la progettazione dei flussi. Permette la gestione e l'esecuzione dei canali e l'analisi dei risultati, tramite delle opportune schermate.

Di seguito una breve descrizione dei principali componenti di Mirth.

Channels

I Canali (channels) costituiscono la componente principale di Mirth e sono responsabili della comunicazione tra l'applicazione Mirth ed i vari sistemi. Possiamo avere diverse tipologie di canali:

- Broadcast: l'informazione è inviata in broadcast a tutti i sistemi in ascolto.
- Router: permette di eseguire diverse trasformazioni a seconda delle destinazioni e di instradare opportunamente le informazioni.

Ciascun canale è composto da una parte di *source* e una di *destination*; in sostanza, legge le informazioni da un sistema sorgente, opera le relative trasformazioni, e trasmette i messaggi in uscita.
Sia la sorgente che la destinazione hanno vari *readers* e *senders* per leggere e inviare i dati in vari formati (da database, da file, da server, ecc.).

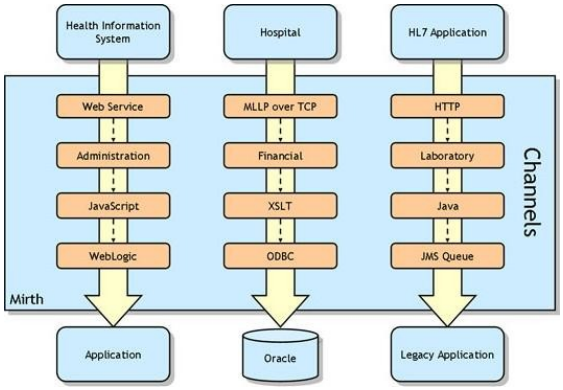


Illustrazione 3: Schema esemplificativo di canali Mirth

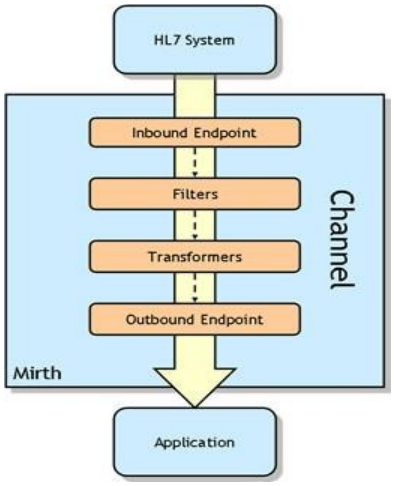


Illustrazione 4: Funzionamento di un canale

Filters

Il filtro è un componente che permette di bloccare le informazioni in ingresso e non renderle più disponibili in uscita. Come vedremo per i transformers, i filtri si basano su script javascript di filtraggio definiti dall'utente. In realtà il linguaggio utilizzato è *Rhino,* un interprete che consente di utilizzare librerie Java all'interno del codice javascript.

Transformers

Le trasformazioni sono degli script Javascript che consentono di operare sui messaggi in ingresso in modo tale da ottenere le informazioni da mandare in uscita. Ciascun *Transformer* è a sua volta suddiviso in vari *steps*.

Progettazione dei flussi

Mirth è nato come gateway HL7 e mette a disposizione tutti gli strumenti necessari per poter modificare facilmente i messaggi. Come vedremo, alcune trasformazioni non presuppongono che in arrivo ci sia per forza un messaggio HL7: questo può essere creato a partire, ad esempio, dall'interrogazione di un database, definendo un template dello stesso messaggio che successivamente si andrà a modificare. La modifica dei campi può essere effettuata facilmente tramite gli oggetti forniti dalle HL7 API.
In particolare, possiamo gestire un messaggio tramite la sua trasformazione in XML, che da un punto di vista del codice è facilmente maneggiabile in termini di campi.

IL PROGETTO DI INTEGRAZIONE DEI FLUSSI INFORMATIVI

Mirth viene ampiamente utilizzato all'ospedale Brotzu di Cagliari come soluzione di riferimento per l'integrazione dei vari reparti/sottosistemi. Il modello scelto permette di avere uno o più canali preposti a gestire le comunicazioni di ciascun reparto verso l'esterno secondo una topologia a stella che diminuisce la complessità del sistema.

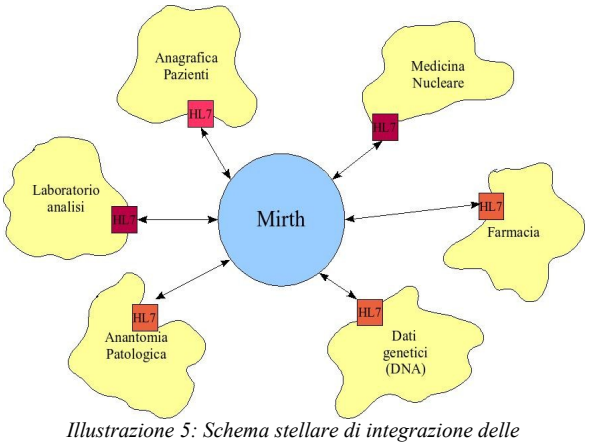


Illustrazione 5: Schema stellare di integrazione delle applicazioni

Il vantaggio principale di questo approccio è di avere a disposizione un sistema che gestisce tutti i diversi flussi in modo indipendente, a prescindere dai sistemi informativi dei reparti interessati: una volta creati i canali Mirth, per ciascuno di questi e verificata la conformità dei messaggi scambiati, saremo in grado di gestirne per intero tutte le comunicazioni. Sarà poi compito del gateway manipolare tali informazioni in modo che siano comprensibili anche dagli altri sistemi e di inoltrarle verso di loro. Una delle maggiori potenzialità di questo metodo è la sua versatilità poiché è utilizzabile non solo tra componenti che parlano "dialetti" HL7, ma anche tra qualunque tipo di sorgente/destinazione, compresi database di diversa natura. Questo consente di ottenere una struttura omogenea capace di integrare tra loro applicativi HL7 compliant e legacy.
Il lavoro portato avanti fino a questo momento riguarda principalmente l'integrazione dei seguenti sottosistemi, fondamentali nell'ambito dell'attività ospedaliera:

- RIS/PACS <-> Helise
- Pronto Soccorso <-> DHE
- Sale Operatorie <-> DHE

Integrazione Ris/Pacs – Helise

Questo lavoro ha coinvolto l'integrazione tra il vecchio sistema di gestione delle prenotazioni radiologiche (Helise) ed il nuovo sistema RIS/PACS (Ebit-AET). Il sistema in questione non è in grado di comunicare direttamente via HL7 verso il RIS. In questo caso Mirth ha il compito di costruire i messaggi di richiesta delle prestazioni, inviarli al RIS ed essere in grado di gestirne le relative risposte.
In primo luogo sono stati eseguiti dei test preliminari per verificare la conformità dei messaggi scambiati e sono stati sviluppati i canali per l'invio dei messaggi relativi alle richieste/cancellazioni di prestazioni radiologiche verso il sistema RIS, in modo da consentire ai due sistemi di comunicare in modo diretto.
Parallelamente a questo sono stati sviluppati degli altri canali di test, al fine di verificare la conformità delle altre tipologie di messaggi (aggiunta di prestazioni lato RIS, cancellazione di prestazioni sia lato anagrafica che lato RIS, trasmissione dei referti). Questo perché il vecchio sistema è al momento in fase di dismissione; sono già state redatte e testate le specifiche per la comunicazione tra il RIS e il nuovo sistema anagrafico. In questo senso Mirth diventerà il punto di riferimento per le comunicazioni su entrambi i lati: in questo modo avremo una soluzione in grado di controllare a basso livello tutti i messaggi scambiati, operando una serie di controlli di conformità e trasformazioni, utili per la gestione/controllo degli errori, che non sarebbero possibili, invece, nel caso di una comunicazione diretta tra i due (il nuovo sistema in fase di attivazione è comunque in grado di supportare lo standard HL7).

Integrazione Sale Operatorie – DHE (non HL7 compliant)

Come descritto precedentemente, Mirth può essere usato per lo scambio di informazioni anche da/verso sorgenti non HL7, attraverso l'uso di canali sviluppati ad-hoc.
In quest'ottica è stato realizzato un gateway Mirth tra il reparto di Sale Operatorie ed il nuovo sistema informativo ospedaliero, il DHE (Distributed Healtcare Environment). L'obiettivo per cui è stato implementato il gateway è stato quello di rendere disponibili nel DHE tutti i dati e le informazioni sugli interventi operatori effettuati nell'ospedale, comprese le eventuali modifiche e cancellazioni. Il sistema è stato implementato tramite due canali Mirth, uno per l'inserimento e l'aggiornamento dei dati su DHE ed uno per la cancellazione. Il canale di inserimento/modifica effettua un *polling* continuo sul database del sistema di sale operatorie e reperisce i nuovi interventi operatori inseriti con il software di reparto, inoltre tramite due *transformer* effettua le dovute transcodifiche e gestisce il flusso di dati proveniente dal database per poi scriverlo su DHE.
All'interno del canale le informazioni sono veicolate attraverso semplici strutture XML codificate opportunamente a run-time secondo le specifiche dei dati letti.

```
<surgical_event>
        <contact_id>2304000000042</contact_id>
        <event_id>0000080099<event_id>
        <unit>30012</unit>
        <op_unit>30017</op_unit>
        <procedure seq="0">
                <code>ICD-01.15</code>
```

```
            <date>2007/90/22</date>
            <time>12:00</time>
        </procedure>
        <procedure seq="1">
            <code>ICD-01.14</code>
            <date>2007/90/22</date>
            <time>12:01</time>
        </procedure>
</Surgical event>
```

Nella struttura relativa ad ogni intervento operatorio è presente una parte statica comune a tutti gli atti che lo compongono, composta da un nodo radice *surgical event* che identifica un singolo atto operatorio; al suo interno sono tracciate le informazioni sul paziente e sull'evento operatorio.
La parte dinamica della struttura è invece generata a seconda del numero di atti che compongono l'intervento, consiste infatti nella creazione di diversi blocchi di procedure per ogni atto. La flessibilità del formato XML si integra alla perfezione con il motore di Mirth attraverso codice Javascript e permette allo sviluppatore di manipolare le informazioni a seconda delle diverse esigenze operative. L'output del canale è rappresentato da file di log creati tramite l'utilizzo della classe logger di Java e da strutture XML. Un ulteriore livello di logging viene messo a disposizione da Mirth: in questo modo è possibile garantire il totale controllo dei dati che transitano all'interno del canale ed allo stesso tempo si ha a disposizione un ottimo strumento per il debugging in fase di test.

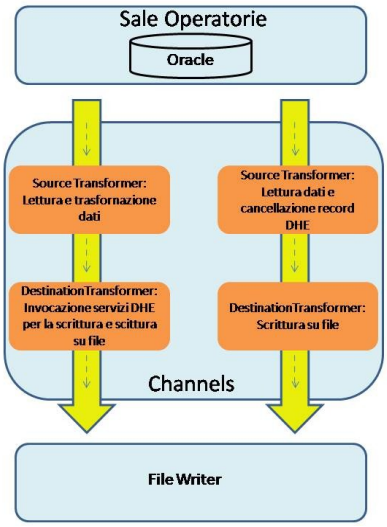


Illustrazione 6: Canali Mirth per Sale Operatorie

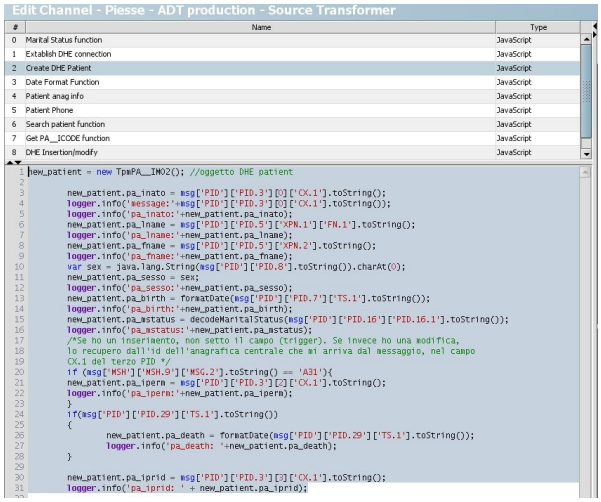
Il secondo canale invece si occupa esclusivamente della cancellazione di record da DHE in caso di errori, causati da mancanza di dati obbligatori o da incongruenze negli stessi, e genera anch'esso in output la struttura XML relativa al record eliminato.
Sebbene lo standard HL7 non sia stato inizialmente utilizzato, esso risulta facilmente integrabile nel gateway grazie alla creazione di un ulteriore canale che funge da listener per il canale di scrittura, oppure sfrutta la funzionalità delle destinazioni multiple offerta da Mirth, che permette la generazione parallela di diverse tipologie di output sul canale.

Integrazione Pronto Soccorso - DHE

L'integrazione dei flussi informativi in arrivo e in partenza dal Pronto Soccorso è una delle problematiche principali da risolvere per un Sistema Informativo Ospedaliero moderno ed efficiente. Ad esempio, la capacità di generare in maniera totalmente automatizzata ed elettronica richieste di esami di laboratorio e di radiologia e allo stesso tempo avere a disposizione l'anagrafica centralizzata dell'ospedale consente non solo di risparmiare tempo e risorse ma di ridurre drasticamente la possibilità di errori, che in condizioni di urgenza o emergenza hanno sempre una buona probabilità di verificarsi.
L'integrazione fra il sistema di Pronto Soccorso PIESSE (Cbim) ed il DHE (Gesi) centrale si basa sulla comunicazione tramite messaggistica HL7 per ADT e richieste di esami di laboratorio e radiologici. L'ingresso del paziente in Pronto Soccorso, la sua successiva dimissione o ricovero in reparto e tutta la sua storia clinica vengono rese note in tempo reale al sistema centrale.
Ad esempio, lo scenario di gestione delle anagrafiche è il seguente:

1. **Consultazione dell'anagrafica centrale dal sistema PIESSE**: avviene in maniera diretta, tramite HL7 ed utilizza un server PDQ sviluppato ad-hoc utilizzando delle librerie HL7 API.
2. **Sincronizzazione delle anagrafiche**: consente al sistema centrale DHE di mantenersi costantemente in sync col sistema PIESSE. Quando un paziente non presente in anagrafica centrale si presenta in Pronto Soccorso, la sua registrazione scatena un trigger e invia un messaggio ADT di inserimento nuova anagrafica verso il gateway centrale, che dialoga con DHE ed inserisce la nuova anagrafica al suo interno tramite invocazione dei relativi servizi java. L'ID centrale dell'utente viene notificato al PIESSE tramite un altro messaggio HL7.

L'utilizzo di Mirth semplifica notevolmente le conversione e l'adattamento dei dati tra sistemi diversi. Lo screenshot seguente mostra come ,all'interno di un canale, si possano eseguire operazioni successive di modifica dei dati e di lettura e scrittura su sorgenti/destinazioni differenti:



L'estratto di codice sopra si collega nativamente ai servizi Java del middleware DHE, converte tutti i campi del messaggio HL7 del PIESSE e li scrive sul sistema centrale.

PERCHE' OPEN SOURCE IN SANITA'

I vantaggi della soluzione descritta si possono riassumere come segue:

- Maggiore adattabilità ai cambiamenti della struttura e dei dati clinici
- Modularità nella interconnessione degli applicativi
- Maggiore ciclo di vita del software
- Totale trasparenza nei flussi dei dati
- Formati aperti di interscambio
- Integra e migliora soluzioni proprietarie e commerciali
- Può essere riusato da altre PP.AA.

Concludendo, l'utilizzo dell'Open Source nei sistemi informativi sanitari porta dei notevoli vantaggi in quanto la versatilità e la riusabilità di questi strumenti assumono una particolare valenza in sistemi complessi e in continua evoluzione come quelli descritti. Inoltre, ancora oggi, gli standard presenti sul mercato non sono sufficientemente stabili e universalmente riconosciuti; l'Open Source consente di avere soluzioni sempre allo stato dell'arte e che possono aumentare notevolmente il ciclo di vita del software.

BIBLIOGRAFIA

[1] Risolution, Ebit AET - http://www.ebit-aet.com/

[2] Microsoft e le tecnologie per la sanità, http://www.microsoft.com/italy/business/mercati/pa/sanita/microsoft_sanita/contributo01.mspx

[3] Mirth Project, http://www.mirthproject.org/