

PPOA Open Source: l'ultimo passo di una lunga evoluzione

**Giovambattista Benedetti, Roberto Danini,
Alessandro Prandoni, Alessandro Romussi**

CILEA, Segrate

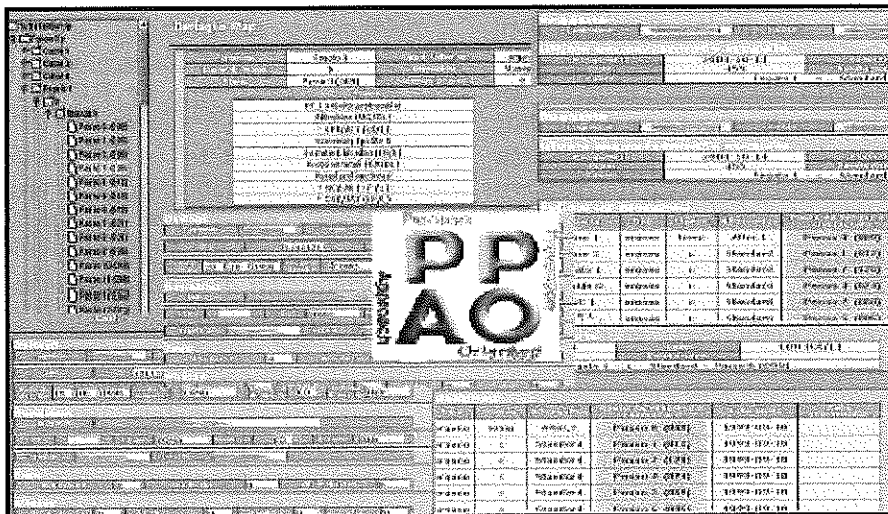
Abstract

E' stata completata una nuova versione del sistema PPOA, destinato da molti anni al monitoraggio dei soggetti guariti da leucemie seguiti dalla Clinica Pediatrica presso l'Ospedale di Monza.

Vengono sottolineati alcuni aspetti significativi del lavoro svolto: sia dal punto di vista scientifico – con lo studio in atto da parte della Clinica Pediatrica e dell'U.O. Emato-Oncologica Pediatrica dell'Istituto Gaslini di Genova – che da quello informatico. Si tratta in particolare dell'inserimento nel sistema di un complesso articolato di regole che automatizzano la scelta e il controllo del monitoraggio appropriato per ogni soggetto.

Viene anche accennato l'utilizzo di metodi di design e tecnologie standard volte alla generazione di codice di qualità.

Keywords: Sanità, oncologia, Object Oriented Analysis and Design, open source, Struts, Hibernate



La nuova versione PPOA

PPOA è un sistema rivolto al monitoraggio dei soggetti guariti da tumore, prima per le sole leucemie e poi per tutti i tumori.

Il nome completo – Person & Prevention Oriented Approach – fa riferimento alla necessità di seguire i soggetti guariti, sia per escludere eventuali riprese della malattia sia per controllare gli effetti tardivi dei trattamenti ricevuti. Ad ogni soggetto viene assegnata una sequenza personalizzata di esami da effettuare a determinate scadenze, chiamata *follow-up*.

Il sistema, entrato in uso molti anni fa [1], è stato più volte rivisto e trasformato [2] [3] [4] [5]. L'aspetto scientifico è curato dalla Clinica Pediatrica di Monza e dall'U.O. Emato-Oncologica Pediatrica dell'Istituto Gaslini di Genova, mentre la progettazione informatica e lo sviluppo software sono sempre stati a cura del CILEA.

Attualmente è stata completata ed offerta all'utilizzatore una versione sviluppata con tecnologie Open Source, accessibile via Web, progettata e modellata tramite diagrammi UML

(realizzati col CASE – Computer Aided Software Engineer – tool Rational Rose [6], mentre come linguaggio di sviluppo è stato adottato Java [5].

Il nuovo sistema è stato completato nell'autunno scorso ed è ora operativo per il test da parte dell'ospedale di Monza.

Nel Bollettino n. 91 [5] si era presentato il quadro generale in cui veniva condotta la "migrazione" dal precedente sistema proprietario a un sistema Open Source orientato ad oggetti. In particolare venivano descritte le caratteristiche tecniche essenziali della fase di analisi e dell'architettura dell'applicazione.

A prodotto ultimato, confermiamo quanto scritto allora, e attiriamo l'attenzione su due aspetti che hanno avuto particolare rilievo nel corso della progettazione e dello sviluppo.

Il primo aspetto riguarda due funzionalità essenziali per l'utilizzatore:

- che il sistema assegni automaticamente al soggetto guarito il/i follow-up appropriati (ovviamente formulando una proposta e chiedendo conferma!);
- che il sistema, mentre monitorizza il soggetto, sia in grado di rilevare se i risultati di determinati esami di controllo innalzano o abbassano la situazione di rischio del soggetto; e quindi suggerisca all'utente le decisioni opportune.

Il secondo aspetto riguarda l'utilizzo del framework java Hibernate, per l'accesso ai dati residenti sul database; e del framework Jakarta STRUTS [8], che agevola l'adozione del pattern MVC [9].

Regole d'assegnazione e regole di salto

All'off-therapy, il medico deve assegnare al soggetto il/i follow-up appropriati, in base alla cartella clinica della degenza e dei trattamenti che il soggetto ha subito (talvolta pluriennali).

Gli elementi fondamentali della cartella – a parte ovviamente quelli anagrafici – sono: i dati rilevanti tratti dalla storia precedente del paziente (dati auxologici, malattie congenite, ecc.), la diagnosi e le eventuali recidive, e infine le informazioni relative alla terapia, comprese le dosi dei farmaci somministrati e gli eventuali interventi chirurgici.

Al sistema vanno forniti i dati della cartella medica, che gli permettono di suggerire il/i follow-up da assegnare.

La diagnosi – ivi comprese l'istologia e l'immunologia, la sede del tumore, le eventuali metastasi e altre informazioni correlate – guida la scelta del *follow-up oncologico* adeguato allo specifico caso, nella gamma dei follow-up

predisposti dai progettisti del sistema per il tipo di tumore curato. Tale scelta è resa possibile dall'inserimento nel sistema di un primo gruppo di **REGOLE d'assegnazione**, che collegano ogni diagnosi a uno specifico follow-up. Il sistema tiene anche conto di eventuali fatti significativi nella storia precedente del paziente, che giustifichino esami di controllo supplementari (ad esempio, se rendono più probabile una recidiva).

Assegnato il follow-up relativo al tipo di tumore da cui il soggetto è guarito, il sistema passa in rassegna un secondo gruppo di regole d'assegnazione. Queste associano particolari trattamenti che il soggetto può aver subito durante la terapia anti-tumorale a possibili danni arrecati a specifici organi. Quando una di queste regole è verificata, viene assegnato al soggetto il relativo *follow-up d'organo*, che prevede una specifica sequenza di esami d'accertamento, volti a monitorare l'apparato coinvolto: cardiologico, nefrologico, fegato, ecc. Da segnalare in particolare i follow-up che gestiscono la situazione immunologica che si crea nei bambini a causa dell'interruzione dei normali cicli di vaccinazioni.

Alcuni esami d'accertamento previsti dal/dai follow-up assegnati al soggetto, possono produrre esiti che impongono ulteriori decisioni. Ciò è formalizzato in **REGOLE di salto**, che individuano un dato esito di un ben preciso esame, sostenuto in una certa tappa di un ben determinato follow-up, e lo associano a una decisione. Tipicamente, gli esiti in questione si riferiscono a valori patologici o comunque non normali. La decisione che il sistema suggerisce può andare dall'interruzione del follow-up e ricovero del soggetto per probabile recidiva, fino a un semplice aggiustamento della sequenza di accertamenti previsti per l'immediato futuro: assegnando al soggetto – nello stesso follow-up – un percorso adatto alla nuova situazione, oppure chiudendo il follow-up in questione ed assegnandone al soggetto uno nuovo.

E' interessante spendere due parole su come è stata progettata e organizzata la porzione di software dedicata alle regole. La struttura che abbiamo descritto potrebbe essere definita, con un pizzico di pomposità, un motore di regole. Esso risulta composto da due macromotori, di off-therapy e di salto, e da una serie di micromotori specifici per ciascun apparato da seguire; oltre che da un insieme di controllori che verificano lo stato del paziente.

I macromotori esternamente presentano un'interfaccia da cui il resto dell'applicativo può

ottenere l'elenco delle successive operazioni attraverso le quali far passare il soggetto; e ciò fornendo all'interfaccia semplicemente l'identificativo e lo stato attuale del follow-up (ovvero l'insieme di esami d'accertamento in corso in quel momento, chiamato step). Internamente, i macromotori semplicemente delegano agli opportuni micromotori la richiesta ricevuta, e ne rielaborano (eventualmente) la risposta: integrandola, traducendola o risolvendo conflitti.

Le regole vere e proprie sono quindi contenute all'interno dello specifico micromotore – in ognuno quelle di uno specifico follow-up. Il micromotore si occupa di valutarle senza preoccuparsi di quanto contenuto negli altri micromotori: cioè le regole relative ad ogni follow-up costituiscono un gruppo autonomo.

Il micromotore delega invece la valutazione delle condizioni del soggetto a dei controllori. Si tratta di porzioni di software che valutano lo stato di alcune caratteristiche del paziente (dati personali, dati sul tumore, dati sui trattamenti subiti in terapia, dati sugli esami precedenti, ecc.). Non sono specifiche di un follow-up, possono anzi essere usate indifferentemente dai singoli micromotori.

In sintesi, si può dire che i controllori valutano i dati reali, i micromotori le regole specifiche ed i macromotori assemblano il tutto.

Utilizzo dei framework Hibernate e Struts

Per quanto concerne l'accesso ai dati residenti sul database, si è deciso di utilizzare Hibernate [7], uno dei più noti framework java per l'Object Relational Mapping.

In sostanza si tratta di uno strumento che automatizza la gestione dei rapporti tra il modello ad oggetti tipico dello strato applicativo ed il modello relazionale della base di dati sottostante. Una volta predisposto opportunamente il file di configurazione che realizza tale corrispondenza, lo sviluppatore viene liberato da quasi tutto il lavoro di gestione della persistenza e dalla scrittura del codice SQL per accedere ai dati (va tenuto presente comunque che il framework fornisce un linguaggio proprietario di interrogazione ad oggetti: HQL – Hibernate Query Language).

Senza entrare in dettagli tecnici che esorbitano dall'ambito di questa presentazione, le potenzialità del framework Hibernate sono apparse notevoli, e si sono rivelate particolarmente utili nello snellire la gestione del complesso archivio sottostante: sia in fase di caricamento dei dati (in particolare i dizionari) che d'aggiornamento (i dati relativi ai pazienti).

Per una corretta separazione tra dati e interfaccia secondo il pattern MVC, si è utilizzato STRUTS che propone un approccio "dichiarativo" alla relazione tra le azioni da svolgere in risposta ad una richiesta web e la visualizzazione.

Oltre all'approccio, collaudato in numerose realtà aziendali, STRUTS offre numerose tag libraries e facilities per aumentare la produttività nello sviluppo.

Conclusioni

Come già sottolineato in passato, il progetto PPOA ha sempre avuto come primo obiettivo quello di offrire agli oncologi un servizio utile e di facile utilizzo. La complessità della materia si è dovuta comporre con le esigenze dell'utilizzo pratico da parte dei servizi di oncologia.

Sono proprio le esigenze pratiche via via emerse, che hanno dettato la strada da seguire: da un lato spingendo i referenti scientifici ad un'analisi sempre più puntuale dello stato dell'arte nel monitoraggio dei soggetti guariti da tumore, dall'altra fornendo ai tecnici informatici l'occasione per applicare metodi di design e tecnologie attuali nel mondo dello sviluppo software.

Bibliografia

- [1] L. Adamoli, D. Frascini, E. Senesi, G.B. Benedetti, M. Camnasio, R. Ferrari, S. Mussi "Gestione di follow-up", Bollettino del CILEA n. 33, Aprile 1992
- [2] R. Ferrari, "P.P.O.A. – Supporto computerizzato per gestire follow-up in ambiente Windows", Bollettino del CILEA n. 52, Aprile 1996
- [3] G.B. Benedetti, "Ampliamento del sistema PPOA per la gestione di follow-up", Bollettino del CILEA n. 75, Dicembre 2000
- [4] G.B. Benedetti, M. Camnasio, F. Fioredda, D. Frascini, R. Haupt, M. Jankovic, S. Mussi, A. Prandoni, "Un sistema per la gestione informatizzata di follow up del bambino guarito per tumore", Bollettino del CILEA n. 86, Febbraio 2003
- [5] G.B. Benedetti, R. Danini, L. Lerra, A. Prandoni, A. Romussi, "PPOA, work in progress", Bollettino del CILEA n. 91, Febbraio 2004
- [6] Rational software, URL: <http://www-306.ibm.com/software/rational>
- [7] Hibernate URL: <http://www.hibernate.org>
- [8] Jakarta STRUTS, URL: <http://jakarta.apache.org/struts>
- [9] Model View Controller, URL: <http://c2.com/cgi/wiki?Model View Controller>