



*Consiglio Nazionale delle Ricerche
Istituto di Calcolo e Reti ad Alte Prestazioni*

OsiriX: Architettura e Use Cases

I. Marra – M. Ciampi

RT-ICAR-NA-06-08

03-2006



Consiglio Nazionale delle Ricerche, Istituto di Calcolo e Reti ad Alte Prestazioni (ICAR)
– Sede di Napoli, Via P. Castellino 111, I-80131 Napoli, Tel: +39-0816139508, Fax: +39-
0816139531, e-mail: napoli@icar.cnr.it, URL: www.na.icar.cnr.it



*Consiglio Nazionale delle Ricerche
Istituto di Calcolo e Reti ad Alte Prestazioni*

OsiriX: Architettura e Use Cases

I. Marra¹ – M. Ciampi¹

Rapporto Tecnico N.:
RT-ICAR-NA-06-08

Data:
03-2006

¹ Istituto di Calcolo e Reti ad Alte Prestazioni, ICAR-CNR, Sede di Napoli, Via P. Castellino 111, 80131 Napoli

I rapporti tecnici dell'ICAR-CNR sono pubblicati dall'Istituto di Calcolo e Reti ad Alte Prestazioni del Consiglio Nazionale delle Ricerche. Tali rapporti, approntati sotto l'esclusiva responsabilità scientifica degli autori, descrivono attività di ricerca del personale e dei collaboratori dell'ICAR, in alcuni casi in un formato preliminare prima della pubblicazione definitiva in altra sede.

OsiriX: Architettura e Use Cases

Mario Ciampi¹, Ivana Marra¹

¹ICAR-CNR, Via Castellino 111, 80131 Napoli, Italia
{mario.ciampi,ivana.marra}@na.icar.cnr.it}

Abstract. Questo lavoro giunge a valle di un'attenta analisi di OsiriX [1], un prodotto software molto potente per il medical imaging [2], usato ormai da numerosissimi medici per effettuare operazioni di image processing. In particolare, questo lavoro descrive l'architettura di OsiriX ed i componenti che la compongono, oltre alle funzionalità più importanti; è anche indicata una sua limitazione: la dipendenza dalla piattaforma MAC OS.

1. Introduzione

OsiriX è un software open source sviluppato da Antoine Rosset che consente di realizzare image processing su immagini medicali prodotte da particolari apparecchiature per la diagnostica, quali MRI, CT, PET e così via. Dal momento in cui il formato di tali immagini è DICOM (Digital Imaging and COmmunication in Medicine) [3], OsiriX è

totalmente compliant a tale standard, anche se supporta altri formati, quali JPEG, TIFF, PDF, AVI, MPEG e Quicktime.

DICOM è uno standard creato dal National Electrical Manufacturers Association (NEMA), che specifica, tra le altre cose, qual è la struttura di un'immagine medica. In particolare, essa è divisa in due parti: nella prima vi è un header, contenente informazioni relative al paziente, all'esame, all'apparecchiatura, ecc.; nella seconda vi è l'immagine vera e propria, che può essere compressa ad esempio in JPEG lossless o JPEG lossy.

Scopo principale di OsiriX è la possibilità di effettuare processing su immagini multimodality e multidimensionali: infatti, è possibile visualizzare immagini da 2 fino a 5 dimensioni e combinare immagini ottenute da apparecchiature differenti, ad esempio una PET con una CT. In particolare, la visualizzazione 3D comprende varie tecniche di rendering, quali Volume Rendering, Surface Rendering, MPR, MIP.

OsiriX permette di interagire con un PACS (Picture Archiving and Communication System) [4], un sistema che consente a) di memorizzare immagini in formato DICOM, b) di effettuare query relativamente alle informazioni contenute in tali immagini e c) di effettuare download di immagini. Per la precisione, anche il protocollo di comunicazione con un PACS è specificato dallo standard DICOM.

OsiriX è stato implementato in Objective-C, ma purtroppo non è un software multiplatforma, infatti, anche se fa uso di librerie open source e cross-platform, è stato costruito al top di COCOA, una piattaforma progettata per sviluppare applicazioni solo su sistemi MAC OS [5]. Questa limitazione ha indotto molti sviluppatori a tentare di

realizzare un porting di OsiriX verso altri sistemi operativi, quali Windows e Linux, cercando fondamentalmente di eliminare la sua dipendenza dalla piattaforma COCOA, senza però avere particolare successo.

Questo lavoro è organizzato come segue: la seconda sezione mostra l'architettura di OsiriX ed i suoi componenti; la terza sezione descrive tutte le principali funzionalità di OsiriX organizzate in categorie; infine, la quarta sezione conclude il lavoro.

2. Architettura

OsiriX è basato su componenti open source che possono essere suddivisi principalmente in due livelli: le librerie di alto livello e quelle di basso livello. OsiriX interagisce solo con le librerie di alto livello, attraverso le API (Application Programming Interface) di queste ultime, che, a loro volta, comunicano con le librerie di basso livello, responsabili dell'interfacciamento con l'hardware.

In Figura 1 è mostrata l'architettura completa di OsiriX.

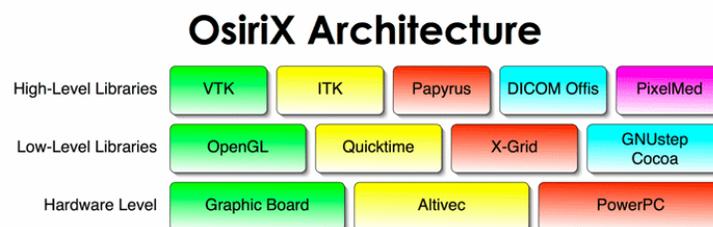


Figura 1. Architettura di OsiriX

Segue una descrizione di tutte le librerie.

Librerie di alto livello:

- *VTK (Visualization ToolKit)* [6]: librerie C++ open source e multiplatforma per la grafica, l'immagine processing e la visualizzazione 3D;
- *ITK (Insight ToolKit)* [7]: librerie C++ open source e multiplatforma che implementano algoritmi di segmentazione e registrazione in due, tre e più dimensioni;
- *Papyrus*: libreria multiplatforma di routines C che facilitano il processo di lettura e scrittura di immagini DICOM;
- *Dicom Offis (DCMTK)*: collezione di librerie e applicazioni che implementano lo standard DICOM in ANSI C e C++;
- *PixelMed*: toolkit DICOM stand-alone che implementa codice Java per leggere e scrivere dati DICOM, protocolli DICOM ed un database di oggetti DICOM.

Librerie di basso livello:

- *OpenGL* [8]: ambiente per lo sviluppo di applicazioni grafiche 2D e 3D. Lo standard prevede la possibilità di utilizzare i linguaggi C, C++, Fortran, Ada, Java;
- *Quicktime*: tecnologia multimediale per sviluppare video, suono, animazione, grafica, testo, interattività e musica;

- *Xgrid*: tecnologia che permette di accomunare virtualmente un gruppo di MAC in un supercomputer, capace in tal modo di risolvere problemi complessi;
- *GNUstep*: piattaforma object-oriented per lo sviluppo di applicazioni desktop in Objective-C basato sulle specifiche del livello applicativo OpenStep della NeXT (ora Apple);
- *COCOA* [9]: ambiente di sviluppo object-oriented progettato per sviluppare applicazioni native MAC OS in Objective-C o in Objective-C++ (esistono anche collegamenti con Java), ma possono essere richiamate anche funzioni Carbon C.

Tutte le librerie elencate sono multipiattaforma, ad esclusione di COCOA e Xgrid, che sono invece realizzate per sistemi MAC OS.

3. Funzionalità

Questa sezione descrive le principali funzionalità di OsiriX, individuate dopo un processo di reverse engineering. Tutte le funzionalità correlate tra di loro sono state raggruppate per classi e, per ognuna di queste, è stato creato uno Use Case Diagram.

3.1 Gestione immagini

Questa classe di funzionalità si occupa della gestione di archivi di immagini, in particolare di immagini DICOM. Le immagini più comuni

infatti possono essere memorizzate in un database interno con lo scopo di evitare di importarle ogni qualvolta si abbia bisogno di esse. Segue una descrizione più dettagliata di tali funzionalità:

- **New Database:** crea un nuovo archivio di immagini DICOM.
- **Open Database:** apre un archivio di immagini DICOM preesistente.
- **Import Files:** importa immagini DICOM consentendo la visualizzazione dell'header.
- **Export Files:** esporta le immagini in vari formati: DICOM, JPEG, TIFF, Quicktime.
- **Delete Exam:** elimina un esame dall'archivio.
- **Anonymize:** rende le immagini anonime cancellando le generalità del paziente dall'header DICOM.

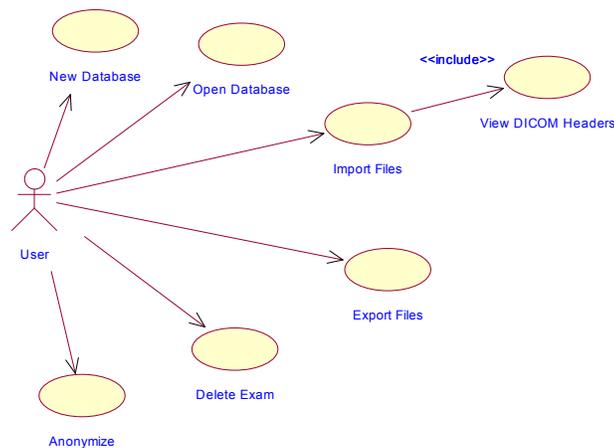


Figura 2. Use Case Diagram della classe di funzionalità *Gestione immagini*

3.2 Interazione con il PACS

Le funzionalità appartenenti a questa classe consentono di interagire con una workstation PACS. In particolare è possibile memorizzare immagini al suo interno, effettuare query o richiedere un download:

- **Send to PACS:** consente di inviare immagini o serie di immagini DICOM al PACS affinché le archivi.
- **Query/Retrieve:** effettua una query al PACS richiedendo il download di immagini o serie di immagini DICOM.

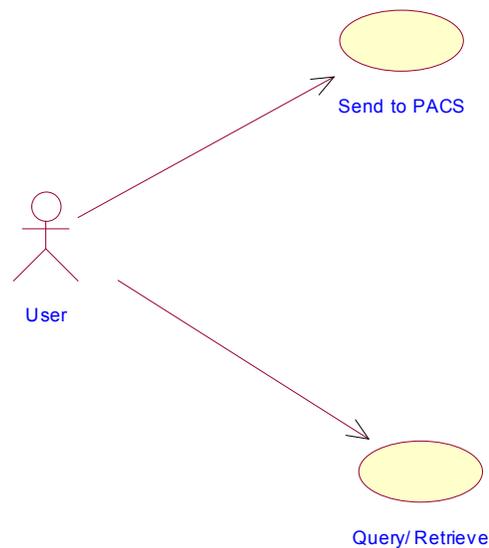


Figura 3. Use Case Diagram della classe di funzionalità *Interazione con il PACS*

3.3 Operazioni su immagini

Le funzionalità qui trattate consentono di modificare l'aspetto e le dimensioni delle immagini, di selezionare regioni di interesse (ROI – Region Of Interest) e di effettuare operazioni su di esse, di sovrapporre immagini composte da più slices e di fondere tra loro immagini. Tali funzionalità sono suddivise nelle sotto-classi seguenti:

Funzionalità base

Queste funzionalità consentono ad un utente di effettuare le operazioni basilari, come ad esempio ruotare un'immagine, traslarla, ecc. Sono mostrate di seguito:

- **Move:** permette di spostare l'immagine in più posizioni.
- **Zoom:** consente di modificare la dimensione dell'immagine.
- **Rotate:** consente di ruotare l'immagine.
- **Animate:** permette di visualizzare in successione una sequenza di immagini.
- **Window Level:** permette di modificare il livello di luminosità dell'immagine.
- **Flip:** consiste nel capovolgere l'immagine verticalmente o orizzontalmente.
- **Image Tiling:** consente di visualizzare in un'unica finestra più immagini affiancate tra loro.
- **Convert to RGB:** consiste nel convertire l'immagine in RGB.

- **Convert to B/N:** consiste nel convertire l'immagine in bianco e nero.

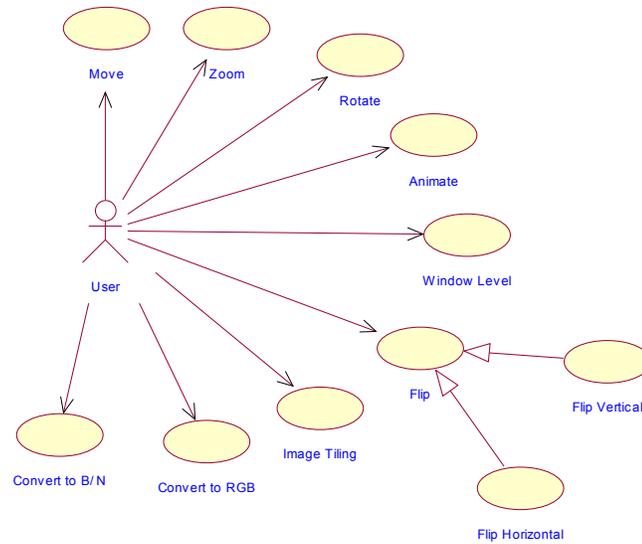


Figura 4. Use Case Diagram della sotto-classe di funzionalità *Funzionalità base*

Segmentazione

La segmentazione è un'elaborazione che consiste nel suddividere un'immagine in regioni disgiunte in base a particolari parametri, come ad esempio la luminanza.

- **Grow Regions Segmentation:** effettua la segmentazione a partire da una regione di interesse selezionata secondo la tecnica growing regions. È possibile scegliere tra le modalità 2D e 3D.



Figura 5. Use Case Diagram della sotto-classe di funzionalità *Segmentazione*

ROI – Operazioni

Le funzionalità appartenenti a questa classe permettono di selezionare regioni di interesse (ROI) nelle immagini e di effettuare operazioni su di esse.

- **Load ROI from File:** carica ROI precedentemente selezionate su un'immagine da un file.
- **Save ROI to File:** salva su file ROI selezionate su un'immagine.
- **Delete All ROI:** elimina tutti le ROI selezionate.
- **Histogram of Selected ROI:** effettua un istogramma delle ROI selezionate.

- **Compute Volume of Selected ROI:** calcola il volume delle ROI selezionate.
- **Show ROI Info:** mostra informazioni associate ad una ROI.

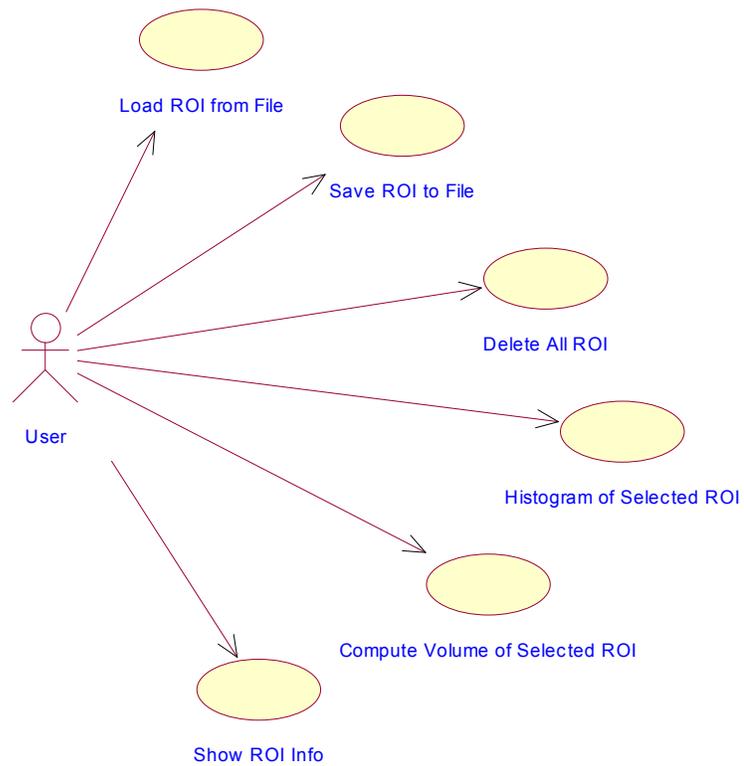


Figura 6. Use Case Diagram della sotto-classe di funzionalità *ROI-Operazioni*

ROI-Tools

Le funzionalità appartenenti a questa classe consentono di scegliere la forma delle ROI.

- **Length:** indica la distanza tra due punti dell'immagine.
- **Angle:** indica l'angolo individuato da tre punti.
- **Rectangle:** individua una regione rettangolare.
- **Oval:** individua una regione ovale.
- **Text:** inserisce righe di testo associate ad una particolare regione dell'immagine.
- **Arrow:** inserisce una freccia che punta ad una particolare regione dell'immagine.
- **Opened Polygon:** individua una regione a forma di un poligono aperto.
- **Closed Polygon:** individua una regione a forma di un poligono chiuso.
- **Pencil:** individua una regione nell'immagine disegnata a mano libera.

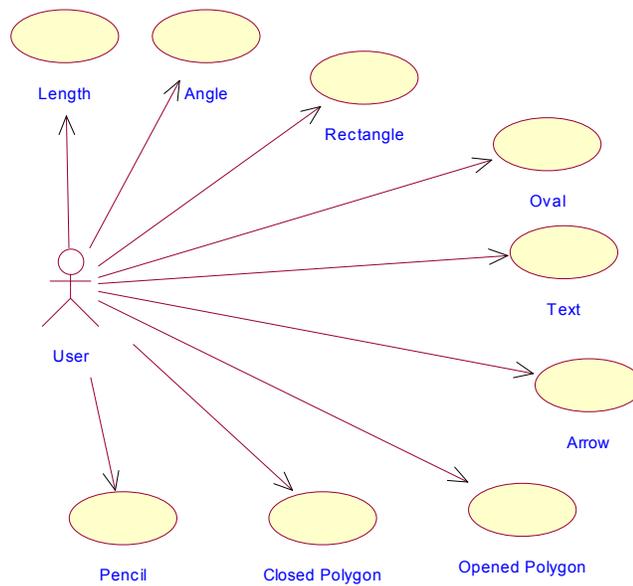


Figura 7. Use Case Diagram della sotto-classe di funzionalità *ROI-Tools*

Thick Slab

Questa funzionalità è particolarmente indicata nel caso di immagini multi-slices. Consente infatti di sovrapporre più slices attraverso diverse tecniche ottenendo un'unica rappresentazione 2D. Per ogni tecnica, è possibile scegliere il numero di slices da considerare.

- **Mean:** visualizza l'immagine ottenuta calcolando la media delle intensità di tutti i pixel delle slices selezionate.
- **Max Intensity Projection:** sono estratti dalle slices selezionate e visualizzati tutti i pixel con luminosità massima.

- **Min Intensity Projection:** sono estratti dalle slices selezionate e visualizzati tutti i pixel con luminosità minima.

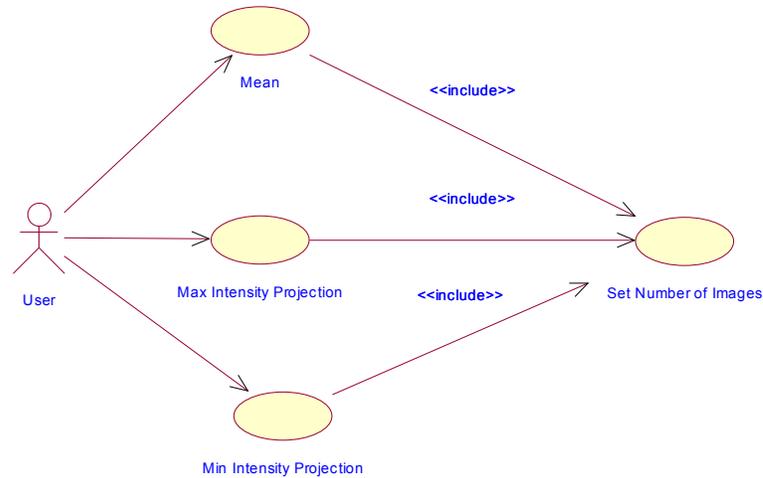


Figura 8. Use Case Diagram della sotto-classe di funzionalità *Thick Slab*

Image Fusion

Le funzionalità appartenenti a questa classe hanno lo scopo di effettuare operazioni di registrazione tra immagini differenti.

- **Fusion:** effettua la fusione tra due serie di immagini che hanno lo stesso numero di slices di uguale formato; è inoltre possibile settare l'intensità della fusione indicandone la percentuale.

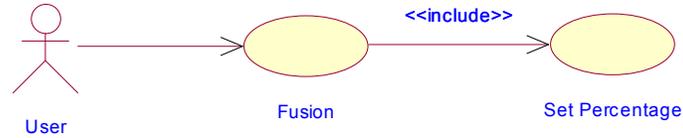


Figura 9. Use Case Diagram della sotto-classe di funzionalità *Image Fusion*

3.4 Rendering 2D

Le funzionalità appartenenti a questa classe si occupano di effettuare il rendering 2D utilizzando differenti tecniche, illustrate di seguito.

- **MPR (Multi-Planar Reformation):** consente, a partire da una serie di immagini 2D multi-planari rappresentanti la vista assiale di un oggetto 3D, di ottenerne le rappresentazioni 2D secondo le viste coronale e sagittale. È possibile localizzare un punto di interesse ed eventualmente effettuare operazioni di rotazione su tali rappresentazioni.
- **Curved-MPR:** è una raffinazione del MPR: invece di ricostruire piani piatti, ricostruisce un piano curvo a partire da una regione individuata da un poligono.

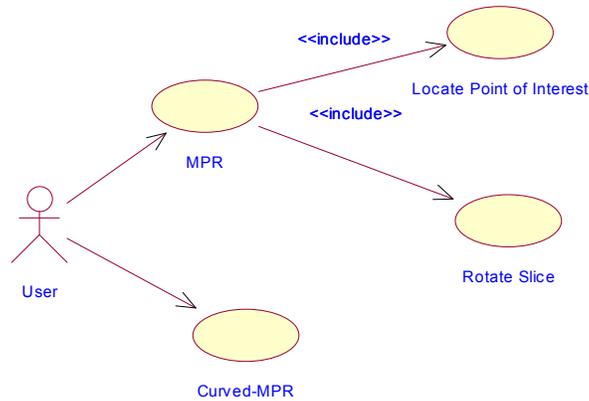


Figura 10. Use Case Diagram della sotto-classe di funzionalità *Rendering 2D*

3.5 Rendering 3D

Le funzionalità appartenenti a questa classe si occupano di effettuare il rendering 3D utilizzando differenti tecniche, illustrate di seguito.

- **MPR:** consente, a partire da una serie di immagini 2D multiplanari rappresentanti la vista assiale di un oggetto, di ottenerne la rappresentazione 3D. È inoltre possibile navigare tra le slices dell'oggetto 3D spostandosi lungo l'asse sagittale, coronale o assiale.
- **Volume Rendering (VTK):** utilizza una tecnica di visualizzazione di volumi dove la rappresentazione tridimensionale avviene attribuendo a ciascun voxel proprietà di

opacità e di colore dipendenti dai tessuti, tenendo conto delle proprietà ottiche dell'intero oggetto (fondamentalmente trasmittanza e riflettanza alle interfacce tra tessuti diversi). È inoltre possibile visualizzare, secondo il piano assiale, sagittale o coronale, l'oggetto 3D rappresentato.

- **Maximum Intensity Projection:** consente di visualizzare i pixel con maggiore intensità lungo un asse prefissato dell'immagine. È inoltre possibile visualizzare, secondo il piano assiale, sagittale o coronale, l'oggetto 3D rappresentato.
- **Surface Rendering:** utilizza una tecnica di visualizzazione per superfici; l'illusione di tridimensionalità viene creata evidenziando il limite tridimensionale della struttura (*contouring*) e inserendo alcune sorgenti luminose fittizie. È inoltre possibile visualizzare, secondo il piano assiale, sagittale o coronale, l'oggetto 3D rappresentato.
- **4D Player:** consente di visualizzare le immagini nelle 3 dimensioni spaziali più quella temporale.
- **Stereo Vision:** consente di visualizzare le immagini in stereoscopia con l'utilizzo di occhiali rosso/blu. Si tratta in pratica di realizzare due diverse immagini da punti di ripresa diversi, ma sul medesimo asse e opportunamente scostati.

4. Conclusioni

In questo lavoro è stato analizzato OsiriX, soffermandosi sulla sua architettura e sulle operazioni che è possibile effettuare con esso,

alcune delle quali molto complesse, come ad esempio quelle che realizzano il rendering 3D o la segmentazione. OsiriX è un software in continua evoluzione: è infatti facilmente espandibile, dal momento che offre un'interfaccia a plug-ins.

Purtroppo, OsiriX è correntemente disponibile solo per piattaforme MAC, però, essendo rilasciato con licenza GPL, un giorno potrà essere reso disponibile per altri sistemi.

Riferimenti

- [1] <http://homepage.mac.com/rossetantoine/osirix/>
- [2] Habib Zaidi, *Medical Imaging: Current status and future perspectives*, <http://dmnu-pet5.hcuge.ch/pdf/Unesco97.pdf>
- [3] <http://medical.nema.org/>
- [4] M. Laßmann, C. Reiners, *A DICOM-Based PACS for Nuclear Medicine*, http://www.healthcare.siemens.com/medroot/en/news/electro/issues/pdf/heft_1_02_e/05_Lassmann.pdf
- [5] Ralf Dreesen, *Open source software for medical images*, <http://www.wcs.uni-paderborn.de/cs/ag-domik/seminare/ss05/Downloads/openSource.pdf>
- [6] <http://www.vtk.org/>
- [7] <http://www.itk.org/>
- [8] <http://www.opengl.org/>
- [9] <http://developer.apple.com/cocoa/>