An OGSA Middleware for Managing Medical Images using Ontologies.

Ignacio Blanquer Espert
Universidad Politécnica de Valencia – DSIC
Camino de Vera s/n, 46022 Valencia, Spain
iblanque@dsic.upv.es

Vicente Hernández García.
Universidad Politécnica de Valencia – DSIC
Camino de Vera s/n, 46022 Valencia, Spain
vhernand@dsic.upv.es

J. Damià Segrelles Quilis.
Universidad Politécnica de Valencia – DSIC
Camino de Vera s/n, 46022 Valencia, Spain
dquilis@dsic.upv.es
Telephone: +34 96 3847007 (Ext. 88254)

Based on "Creating Virtual Storages and Searching DICOM Medical Images through a GRID Middleware based in OGSA" by Ignacio Blanquer, Vicente Hernández, Damià Segrelles which appeared in Proceedings of the 5th IEEE Internatinal Symposium on Cluster Computing and the Grid, Cardiff, Wales, UK May 9-12, 2005. Sponsored by IEEE and IEEE Computer Society. IEEE Catalogue 05EX1055C. ISBN 0-7803-9075-X.© 2005 IEEE.

Abstract

This article presents a Middleware based on Grid Technologies that addresses the problem of sharing, transferring and processing DICOM medical images in a distributed environment using an ontological schema to create virtual communities and to define common targets. It defines a distributed storage that builds-up virtual repositories integrating different individual image repositories providing global searching, progressive transmission, automatic encryption and pseudo-anonimisation and a link to remote processing services. Users from a Virtual Organisation can share the cases that are relevant for their communities or research areas, epidemiological studies or even deeper analysis of complex individual cases.

Software architecture has been defined for solving the problems that has been exposed before. Briefly, the architecture comprises five layers (from the more physical layer to the more logical layer) based in Grid Thecnologies. The lowest level layers (Core Middleware Layer and Server Services Layer) are composed of Grid Services that implement the global managing of resources. The Middleware Components Layer provides a transparent view of the Grid environment and it has been the main objective of this work. Finally, the upest layer (the Application Layer) comprises the applications, and a simple application has been implemented for testing the components developed in the Components Middleware Layer.

Other side-results of this work are the services developed in the Middleware Components Layer for managing DICOM images, creating virtual DICOM storages, progressive transmission, automatic encryption and pseudo-anonimisation depending on the ontologies. Other results, such as the Grid Services developed in the lowest layers, are also described in this article. Finally a brief performance analysis and several snapshots from the applications developed are shown.

The performance analysis proves that the components developed in this work provide image processing applications with new possibilities for large-scale sharing, management and processing of DICOM images. The results show that the components fulfil the objectives proposed. The extensibility of the system is achieved by the use of open methods and protocols, so new components can be easily added.

Keywords: DICOM, OGSA, Grid Service, Grid Middleware, HealthGrids, Ontologies.

1 Introduction

Medical Imaging departments in modern hospitals are currently working fully in digital. Primary care services are progressively migrated into the digital world, using quite often Computer Radiology devices to digitise, print and temporally store the images. The advantages of digital images respect to analogic format are obvious, the new tecnologies can be used for storing, processing and managing the medical images (thus leading to a less vaste of consumables for storaging, quicker transferring, and more complete analysis processes etc….). Moreover sharing medical images would be a large advantage in training and research, since the amount of digital medical images obtained yearly in Europe is estimated in the order of Petabytes. This vast amount of information is very valuable, mainly considering that training in medical radiology is strongly based on cases and evidence. Medical communities working in specific areas and diseases would benefit of accessing to a larger set of examples, relevant cases and diagnosis.

On the other hand, the use of digital image provides some difficulties for their management. Although the compatiblities in the terms of the digital format is solved with the standard DICOM [6] (the standard digital format more widely used by both the industry and the clinical users), there are several issues that are still unsolved: a) the nature of DICOM [6] images is diverse (MRI, X-Ray, CT, Video, etc…). b) Today most European healthcare centres use the digital format for their databases of images (PACS, Hard Disk Shared Directories, etc…), and these databases remain isolated at their different physical sites. c) The management of those databases is also different in each site (Relational Databases, DICOM Servers, File Directories etc…). d) The vast amount of data requires a carefully organisation of the images to build-up virtual sub-repositories with the information relevant to specific communities or even research studies. e) And the problem of Long-term archiving of medical images.

Thus, the consolidated storage of the information from even a small corporation of few hospitals and centres is a difficult problem. Moreover, individual Medical Imaging studies are large and costly to transmit. The idea of a large, consolidated corporate storage is currently unrealistic. However, current efforts are focused on the federation of databases. Federation consists on the local storage of the data, but providing an upper-layer able to provide a unique view. This procedure requires a highly coupled degree of interoperability, although formats and back-ends can be different. This could be a good solution for networks of hospitals from a same corporation, but very difficult to implement when the centres have a high degree of independency. Sharing medical images is not making different distributed databases seamlessly interoperable. It basically requires to make data accessible, surely through search criteria much different from those of clinical practice, and to ensure the quality and privacy of the data. Thus sharing data should not require homogenising protocols, storage formats or even security policies.

Grid technologies have been designed to deal with sharing distributed heterogeneous resources without compromising local administration. Data grids consist on using distributed storage resources as a single storage resource, more in the idea of a virtually shared file system than a distributed database. Data files are then stored and retrieved transparently and securely to and from any resource on the Grid. Other Data Grid approaches focus on the creation of interfaces and common layers that provide a common view point, without requiring the construction of a new distributed storage from the beginning. Ontology frameworks can be defined on top of the Grid Middlewares to structure the semantics of the vast information stored.

All these problems regarding in the source of images, the location of the storages, the managing systems, the management of ontologies and the long-term archiving of digital imaging problems make the transparent sharing of images and the development of collaborative environments difficult. The development of an infrastructure that could deal with these issues will make new applications of Medical Imaging (Advanced Image-based Diagnosis, Non-Affine Registration/fusion applications, Context-Based searching of DICOM images etc...) more efficient.

Many efforts are being put on the application of Grid Technologies to Medical Imaging storage. Projects such as European DataGrid (EDG [1]) developed a very efficient framework for the distributed management of data. Application of this approach to medical imaging is being performed in projects such as the BIRN [2], MEDIGRID ACI project [3], the IXI [4]or Neurobase [5].

This work describes a Grid MW developed for managing, processing and sharing DICOM digital medical images using different ontologies that the user can define for managing the information that it is interesting for a given medical area or research group. This MW is not neither a specific application for specific medical areas (Neurology, Cardiology etc…), nor an alternative to PACS systems, but a tool for developers and researchers to share medically relevant shared images, abstracting the developers from all features of a distributed infrastructure. The MW has been defined considering non-dedicated TCP/IP networks (such as Internet) for interconnecting resources.

The article firstly defines the architecture of the developed MW, also explaining all the integrated components developed and the alternative technology available. A software library of Grid components based in the OGSA [7] specifications (and using Globus 3 toolkit [8]) has been developed on top of the architecture. Currently, the components are available coded in Java language but in versions of the library in others programming languages (C++, C# etc…) are being implemented.

The next sections cover the following topics: Section 2 (methods) describes the generic architecture of the MW. Section 3 (results) describes the upper-level MW components that have been implemented for downloading, storing and processing DICOM images, and other Grid Services implemented in the lower layers of the architecture.Finally, an analysis of the results obtained in this work is presented, followed by the conclusions and future work.

2 Methods

A Software arquitecture is proposed in this work to provide a solution to the problem of the integration and processing of a network of repositories of DICOM objects. The architecture proposed in this work uses standard components and is fully based in OGSA, fostering the interoperability with future Grid infrastructures to be deployed in the different eScience programmes. A general view of the architecture is showed in Figure 1, which also shows the interactions among the diferents layers. Furthermore, this seccion describes the methods used for building the architecture (protocols, servers, data formats, security components etc…) and the functionality of each layer.

2.1 General Architecture Definition

The software layers defined in the architecture solve specific problems at different levels hiding the implementation details as higher levels are reached. These levels are described starting from the ones more dependending on the physical resources (Core Middleware Layer) and ending on the higher logic layers that abstract the developer of the applications from the Grid features (Component Middleware Layer). The last layer described is the Application Layer, which is not the highest priority objective of this project but it is necessary for testing the components and the MW, in particular has been developed two applications for testing the components developed for manage the virtual storaged DICOM using different ontologies.

2.1.1 Core Middleware Layer

This layer corresponds to the lowest level of the architecture. This layer defines the Grid Services that interact with the physics devices and the information systems. The objective of this layer is to abstract the upper layers from the details that depend on the specific architecture, platform or data format of the Grid resources, offering a view of logic resources. This objective is reached by means of the communication with the upper layers using standard interfaces that define input/output data structures using XML [6] schemes.

Core Middleware Layer communicates directly with the Grid Services Server Layer and the MW Components Layer through protocols provided by the Communication Layer (HTTPS [10] for remote executions and GridFTP [11] for downloading large sets of files).

This layer currently implements a Grid Service for managing the DICOM Storage considering diferents ontologies of digital images, which is explained in more detail in section 3.1.1.

2.1.2 Grid Services Server Layer

This Layer contains the Grid services that perform server tasks in the Grid environment (such as registering of allowed resources, image searching indexation, management of the definition of ontologies etc…). The layer is also composed by logic resources as the Core Middleware Layer, but these are pure logic resources that do not interact with any physic device or informations systems, but with the logic resources provided by the Core Middleware Layer. Subsection 3.1.2 includes a brief description of the Grid Server components developed and defined in this architecture.

The Grid Services Server Layer communicates with the Core Middleware Layer through the interfaces defined in their Grid Services, abstracting from the devices features and data formats of the Core Middleware layer. These layers also communicate with other Server Grid Services in s similar way.

2.1.3 Communication Layer

The Communication Layer includes the components and protocols that implement the communication either between the Core Middleware Layer, Server Grid Services Layer and and the Components MW Layer. The protocols used are HTTPS and GridFTP, which fit perfectly the requirements of the architecture. Compulsory requirements of these protocols are the security of connections, the confidentiality and the integrity of data. The usage of HTTPS and GridFTP protocols guarantees these requirements for the communication stage. The usage of digital certificates [10] signed by the certification servers that contain the Grid Services and GridFTP servers implements the authorisation and authentication. Additional requirements on data encryption are explained with more detail in section 3.3.

The Figure 2 shows the interaction betwen layers of the architecture using the protocols that are provided by the counication layer.

2.1.4 Middleware Components Layer

The Middleware Components Layer implements the services and components common to many medical imaging Grid applications, providing a homogeneous interface to the data stored in different repositories. These components offer to the applications a logic interface to the physic components providing the designer and the researcher with a virtual vision of resources deployed in the Grid. The components of this layer are the main objective of this work. This layer depends on the definition of new Grid Services in the lower layers (DICOM Storage Services, Processing Components etc…). The components that implement the ontological framework are expoded in seccion 3.2.

Middleware Components are virtualisations of several logic components of the Core Middleware Layer, orchestrated by the components of the Grid Services Server Layer. A virtual repository (Mw Component Layer Object), actually could comprise several Storage Objects (Core Layer) which are equal in terms of interface although could relate to different systems. The Server Layer provides the components that locate the individual objects and distribute the queries.

2.1.5 Application Layer

The Application Layer corresponds to the tools and systems that use the Middleware components through its high level interface. A set of sample applications have been developed in this work for testing the Middleware Components Layer. The applications developed make use of the components for creating virtual storages of DICOM images using ontologies, searching images on it, and managing the downloading of the images.

3 Results

The components developed and the performance analyses are presented in this seccion. It comprises the following four subsections: The first subsection is about the Grid Services that has been developed both in the Core Middleware Layer and Grid Services Server Layer. The second subsection describes the componentes developed in the Middleware Components Layer, the third subsection is about the security and finally the last subsection describes the performance analysis using the applications developed.

3.1 Grid Services

This section describes the Grid Services deployed and implemented in the Core Middleware and Services Server Layer, which are used by the Components Middleware Layer for managing the digital images. A graphic general view of the interaction between the diiferents Grid Services is showed in Figure 3.

3.1.1 Core Middleware Grid Services

In this layer comprise several Grid services for managing the devices that store and process the digital images. Mainly, there is a Grid service for managing the storages of digital images and provides the user with diferentes views for sharing, searching, processing, etc, depending on the ontology selected.

DICOM Storages Grid Service. This Grid Service interacts directly with the devices that manage DICOM images (hard disk, DICOM Servers, Relational BD, PACS Systems etc…) and offer to upper layers a view of a logical resource that manage diferents views, one for each ontology that is defined in the Ontology Server Grid Service located in the Services Server Layer. For each ontology definition, the DICOM Storage enables the searching and retrieving of data related to DICOM objects.

The service gets the searching request that is started by the component Middleware C_GRID_DICOM_Storage described in section 3.2.3 and translates the request (encoded in XML) into the language of the device manager (SQL Query, browsing of DICOM headers of a directory, QUERY_RETRIEVE DICOM primitive…), returning the result of this search in a specific XML structure.

The Figure 4 shows the logical vision of the management of diferent ontologies, providing a view for each ontology defined in the system.

On the other hand, it also prepares the transferring, sending a request for downloading and making the DICOM-Storage service to compress the image in lossless JPEG2000 [12][13] format, (reducing the size of the image in a ratio between 2 and 3). The JPEG2000 format allows progressive downloading. The Middleware gets the relative information to download image with the GridFTP protocol.

The DICOM_Storage services interact with the Grid Services Server Layer, registers the results in the IIS Grid Service and sends a notification to the IIS for updating its state. The state includes information on the number of images of each ontology and is updated synchronously.

3D Volume Rendering Grid Service. This Grid service demonstrates the additional computational advantages of the Grid Computing. It implements a Volume Rendering engine that provides 3D projections of medical studies. The Core MW Layer comprises a set of services providing the necessary interaction between the DICOM Storages and the Processing Resources in the MW Layer. The service enables the user to select a study, a projection method and a resource to execute the processing, and automatically receive the results on the client side.

Additional Grid Services. Other Image Processing Grid services are being implemented on the architecture. Image registration has been an important target, since they are computationally intensive and constitute a good combination of storage and computing requirements.

3.1.2 Grid Services Server

The Grid Services Server is located in the architecture between the Middleware Layer and the Core Middleware Layer. The mission of this layer is to manage all services from the Core Middleware and Grid Services Server Layer and to provide information to the MW Components Layer. The services and components developed are described next.

IIS (Index Information Index) Grid Service. The IIS registers all Grid services deployed in the architecture, gets information about the status of services and updates it synchronously. Each Grid Service defines its status using a format defined in an XML scheme. The main mission of this component is to give support to all active and inactive resources of the Grid. This service is integrated into the MDS2 [14] of GT3.

Storage_Broker Grid Service. The Storage_Broker is a service that manages the information of DICOM_Storage components available in the Grid. It does not interact directly with them, but gets the information from the IIS that registers their status and contains all necessary information. This service updates the information asynchronously. The main objective of this component is to optimize the creation and searching index of DICOM_Storage components.

This component is registered in the IIS when deployed in the Grid and interacts directly with the component Middleware C_GRID_DICOM_Storage, described in next section.

Ontologies Server Grid Service. This Grid service manages the ontologies that can be used in the Grid environment. The ontology is defined using a definition language based in XML. Basically each ontology has three diferents type of fields (restrictive, creation nd search) which currently only make reference to data in the DICOM header. The restrictive fields are used to filter images that do not mach the reference values. The creation fields are used for indexing these fields in the creation of virtual storages and the search fields are define the criteria for an specific search in a virtual storage of DICOM images defined by the creation fields.

3.2 Middleware Components

In this section the Middleware Components are described. They are currently split in three packages: the management of DICOM images, the downloading of images and the management of Virtual Storages. All components have been developed in Java and use the GT3.

3.2.1 DICOM Images Package

In this package all Grid components for managing DICOM images are included. The components of this package interact with other components of the download package (see section 3.2.2 for more details).

C_GRID_DICOM. This component virtualises a single DICOM image. It manages all the information related to the DICOM image (header data and image bitmap). The physical location where the image is stored is transparent to the user, who directly interacts with the DICOM_Storage Grid Service containing the DICOM image.

C_GRID_DICOM_Set. This component manages a set of C_GRID_DICOM objects. Each C_GRID_DICOM can be linked to different DICOM_Storage Services.

C_GRID_DICOM_Serie. This component manages a set of C_GRID_DICOM objects in a single series and study UID. Also, images can be distributed in the Grid in different DICOM_Storage Services.

C_GRID_DICOM_ Study. This component manages a set of C_GRID_DICOM objects with the same study UID. Also, images of a study can be distributed in the Grid using different DICOM_Storage Services.

3.2.2 DICOM Images Downloading Package

The main goal of this package is to manage the downloading of either individual or groups of DICOM images. The downloading process provides with different modes and can manage user-selectable priorities that define the behaviour of the process and the order in which images and layers are retrieved.

C_GRID_Download_Order. It manages the downloading of DICOM images that are represented by the C_GRID_DICOM components interacting with the C_GRID_Manage_Image_Set component for multiple downloads. It defines a downloading order by the priorities (from 0 to unlimited) of download. The higher priority is 0, and only when all the images from a priority have been downloaded, images with the next priority are considered. The downloading order can be modified dynamically changing the behaviour on the next step. These modes are further explained.

C_GRID_DICOM_Download_Manager. This is a specific component for managing the downloading of DICOM images. The downloading modes are two: a) Complete image downloading: at each step of the downloading process, the image with the highest priority and lowest sequence number will be completely downloaded; b) Layer by layer downloading: at each step the remaining JPEG2000 layer with the lowest index is downloaded, considering all the images from the highest priority and starting from the lower sequence number. This enables showing the images progressively, from less to more resolution, as the layers are downloaded.

C_GRID_DICOM_Set_Download_Manager. This component manages the downloading of a set of C_GRID_DICOM images. The downloading is performed in a specific order defined by the user in the C_GRID_Download_Order object. This component has different working protocols: a) Image image by image downloading mode, retrieving completely the images attending on the priority; b) Layer by layer, on a single image downloading mode, in which the different layers of an image are downloaded until it is completely retrieved; c) Pure layer by layer downloading mode, in which the lowest remaining layer from the set of images with the highest priority is downloaded (one layer per image on the same priority).

3.2.3 Virtual DICOM Storage Package

C_GRID_DICOM_Storage. It manages the DICOM images that there are in the Grid and which fulfils a determinate set of properties or conditions filter. The images constitute a virtual storage, regardless of their physical location. This component is formed by DICOM Storage Grid Services that provides it with all the information from images.

This component allows showing the source of all the images that composes it, also enabling the definition of basic filters that define the common properties of the images that will constitute the virtual storage (mainly considering the medical interest of a community). It also implements searching of images using higher-level searching criteria.

3.2.4 3D Volume Rendering Package

C_GRID_3D_VolumeRendering. Process set C_GRID_DICOM_Set using 3D projections. It abstracts the user from the location of the resources (DICOM images and resource for processing them) and also from the transference of the images from Stogae DICOM to process Resource.

3.2.5 Ontology Package

C_GRID_Ontology. This component manages the ontologies defining medial areas or research groups. This component inserts, removes and updates the ontologies registered in the Ontology Server Grid Service and modifies the views in each StorageDICOM Grid Service. Once an Ontology is defined and registered, only the data relevant to this framework will be accessible, reducing browsing and searching response time and increasing the knowledge.

3.3 Security

One important concern when dealing with medical information is privacy and security. Medical data is highly private, and confidential pieces of information should only be available to the owner, the medical team in charge of his/her treatment and with some restrictions to a medical research community.

The distributed architecture proposed in this paper has less security constraints than other fully distributed approaches. Typical Datagrid approaches focus on the use of distributed resources to maximise the storage capabilities of a virtual organisation. Since data can be permanently stored on a different organisation than the one that produces it, content must be protected to prevent users, even with administrative privileges, be able to access the content of the files. This approach increases complexity thus requiring secure repositories of encryption keys [15]. However, in the architecture proposed, the information is stored by the centres who own it, being the Grid MW the integrating layer that provides a seamless access. The security diagram is showed in Figure 5.

Data security in the Grid Middleware proposed is based on the GSI (Grid Security Infrastructure), adding some tools and procedures to enhance privacy. Primary motivations behind the GSI are: a) the need for secure communication (authenticated and confidential) between entities of a Grid. b) The need to support security across organizational boundaries, thus preventing a centrally-managed security system. c) The need to support "single sign-on" for users of the Grid, including delegation of credentials for computations involves multiple resources and/or sites. GSI is based on public key encryption, X.509 certificates, and the Secure Sockets Layer (SSL) communication protocol. Extensions to these standards have been added for single sign-on and delegation. The Globus Toolkit's implementation of the GSI adheres to the Generic Security Service API (GSS-API), which is a standard API for security systems promoted by the Internet Engineering Task Force (IETF).

First level of security is the use of certificates to authenticate users entering the Grid. Distinguished Name (DN) contained in the certificate is used for comparison with authorised user lists and Access Control Lists. Transferring of the data is performed through secure protocols (SSL-based) and thus do not present threats. Processing of the images would have involved risks if personal information concerning the image would be transmitted. Since only pseudo-anonymised data is transferred, privacy is not compromised, and thus encryption keys are not necessary (if the security policy of the centre does not require them). Pseudoanonymisation is based on the job id. The pseudoanonymisation is performed at the level of the local repository, preventing private data to be sent outside the borders. Blurring on the sensible data (address, birth date, examination date, etc) is also performed locally.

3.4 Analysis of the functionalities

This section shows the results obtained in the tests of the architecture and components described in this paper. Firstly, it shows the application developed that use the packages of the MW components. Secondly, it presents the performance results in the execution of several of the services and finally discusses the conclusions.

3.4.1 Test Application

The options of the application are a direct interface to the features implemented by the Middleware components. Figure 6 shows the part of the application that enables the creation of DICOM Virtual Storages depending on the input filter that the user defines and the ontology that is choosed. The filter is formed by the Image Modality field. This DICOM Storage object is used to search studies using queries about any field included on the headers of the DICOM images (age, sex, centre, body part etc…) and they are included in the ontology that we manage.

Figure 7 shows the part of the application that corresponds with the management of the downloading of DICOM images. The user first creates DICOM Storages enabling searching on them. Images resulting from the searching process are downloaded specifying an order using the priorities explained in section 3.2.2.

Figure 8 shows the evolution in time and size (the latter in percentage with respect the total size) in the progressive downloading of a CT image 512 x512x12Bits. The vertical line denotes the time-step in which the relative error decreases below 2%. Transferring time is longer using progressive transmission than non-progressive because do operations of decompression for each layer that is downloaded. However, unproductive waiting time is reduced since information is progressively presented as information is available. This approach is favourable for the Grid implementation, since Grid overhead is much less important.

3.4.2 Load Test

A test has been performed to check the performance of the system. Four servers (with a total of 9 processors) have been used and more than 3000 images have been distributed. All servers have GT3 installed and a different configuration of the Grid Services implemented: all servers have the DICOM_Storage Grid service installed and one server (OSIRIS) has the IIS Grid Service and the Storage Broker installed. The features of each server are listed in Table 1.

The tests have been performed using a 100 Mbps TCP/IP Network, and two simultaneous users.

The application described in the previous section has been used for the load test. The first test is the creation of a Virtual DICOM Storage and the execution of a searching operation on top of it. The time spent by the system in the creation of the Storage is shown on the following table:

Table 2 shows that the creation of second storages takes quite shorter time than the first ones. The reason is the overhead due to the connection cost of the Storage Broker, which affects only the first execution.

Table 3 shows the time spent by the components when executing a query based on different criteria. The information regarding the time spent, the virtual storage used, and the number of images involved is included on this table.

The searching time only depends on the database backend, the size of the results and the bandwidth of the network. The resources deployed in the Grid used for load test have the same database manager. Obviously, as the results involve a larger number of images the searching time increases.

4 Conclusions and Future Work

The Middleware layer developed in this work fulfils the objectives of performance, security and interoperability of the problem. The Middleware developed constitutes an efficient framework for transferring, searching and processing DICOM imges and increases the productivity of code developers for building applications for manages DICOM images. Moreover, Grid versions of several components common to a wide range of DICOM Medical Imaging applications have been developed (including 3D Rendering, Segmentation, progressive image transferring…).

The architecture is constructed on top of an ontology framework that provides a way to create virtual sub-repositories comprising only the information that is relevant to an specific community or even to an experiment, reducing the response time of distributed processing.

The architecture is open and has been defined to enable the development of new functionalities and resources (new computational resources for segmentation, data formats, registration, etc.…). The components have been developed in Java but migration to others languages (C++, etc...) is straightforward, since the protocols used in the architecture defined are standard and are supported by the most languages systems.

The future work planned will be focused on three main lines: Increase of the support of different formats of local storages, inclusion of structured radiology reports and migration to new Grid standards.

In relation with the increase of the support of image storages, SQL databases will be supported along with the available DICOM file and Query / Retrieve protocols. In the frame of the support of Structured Reporting, interface to DICOM SR format is being considered. Finally, regarding new Grid Standards, migration from GT3 to WSRF is being performed as well as the creation of interfaces to other languages (C++, C# etc…).

References

[1] "European DataGrid Project". www.eu-datagrid.org

[2] "Biomedical Informatics Research". http://www.nbirn.net/

[3] "ACI project MEDIGRID: medical data storage and processing on the GRID".http://www.creatis.insa-lyon.fr/MEDIGRID/

[4] "Information eXtraction from Images (IXI) Grid Services for Medical Imaging". Working Notes of the Workshop on Distributed Databases and processing in Medical Image Computing (DIDAMIC'04). Pag 65.

[5] "NeuroBase: Management of Distributed and Heterogeneous Information Sources in Neuroimaging". Working Notes of the Workshop on Distributed Databases and processing in Medical Image Computing (DIDAMIC'04). Pag 85.

[6] Digital Imaging and Communications in Medicine (DICOM) Part 10: Media Storage and File Format for Media Interchange. National Electrical Manufacturers Association, 1300 N. 17th Street, Rosslyn, Virginia 22209 USA.

[7] "Open Grid Services Architecture (OGSA)", http://www.globus.org/ogsa.

[8] Globus alliance Home Page. "Relevant documents", http://www.globus.org

[9] Allen Wyke R., Watt A., "XML Schema Essentials". Wiley Computer Pub. ISBN 0-471-412597

[10] Web security and commerce / Simson Garfinkel. - Cambridge: O'Reilly, 1997. - 483 p.; 23 cm. ISBN 1565922697

[11] "The GridFTP Protocol and Software"**.** http://www-fp.globus.org/datagrid/gridftp.html.

[12] JPEG2000: Image compression fundamentals, standards and practice / David S. Taubman, Michael W. Marcellin. - Boston [etc.] : Kluwer Academic, cop. 2002. - XIX, 773 p. ; 24 cm. + 1 CD-Rom - (The Kluwer international series in engineering and computer science) ISBN 079237519X

[13] Bradley J. Erickson, M.D. "Irreversible Compression of Medical Images", Dpt. Radiology, Mayo F., Rochester, MN, Jo. of D. Imaging, DOI: 10.1007/s10278-002-0001-z, 02.

[14] Monitoring & Discovery System (MDS)" http://www-unix.globus.org/toolkit/mds/

[15] "Key management for encrypted data storage in distributed systems". Proceedings of HeathGrid 2004.

Table 1. Resources Deployed.

Name	*Architecture*	*Images*	*Modality*
SEKER	Four processors Intel Xeon 2.0 Ghz 4 GB RAM	2154	OT-> 1820 CT -> 334
AKER	Two procesors Pentium III 1 Ghz 512 MB RAM	933	CT -> 558 MR-> 375
KEFREN	Biprocesador. Intel Xeon 2.0 Ghz. 1 GB RAM	54	CT -> 42 CR -> 12
OSIRIS	AMD 1800 Mhz 512 MB RAM	67	OT -> 67

Table 2. Storage Creation Time.

Name	*Images*	*T. Creation (ms)*	*Description*	*Resources*
ALL1	3208	5367	All images	4
ALL2		380		
CT1	934	5036	Images of CT type	3
CT2		287		
OT1	1887	5128	Images of OT type	2
OT2		226		

Table 3. Searching Times

Search Description	*Storage*	*Images*	*Resp Time (ml)*
All Images	ALL	3208	29729
Images CT	ALL	934	11210
Images OT	ALL	1887	15576
Images MR	ALL	375	2952
Images CR	ALL	12	215
All Images	CT	930	11069
All Images	OT	1887	18383
All Images	MR	375	5895
All Images	CR	12	589
StudyUID= 1.3.12.2.1107.5….	ALL	24	1088
StudyUID= 1.3.12.2.1107.5….	CT	24	1112
StudyUID= 1.2.756.9999….	OT	62	1783
StudyUID= 1.3.46.670589…	MR	24	1127
StudyUID=1.3.46.670589…	CR	2	316

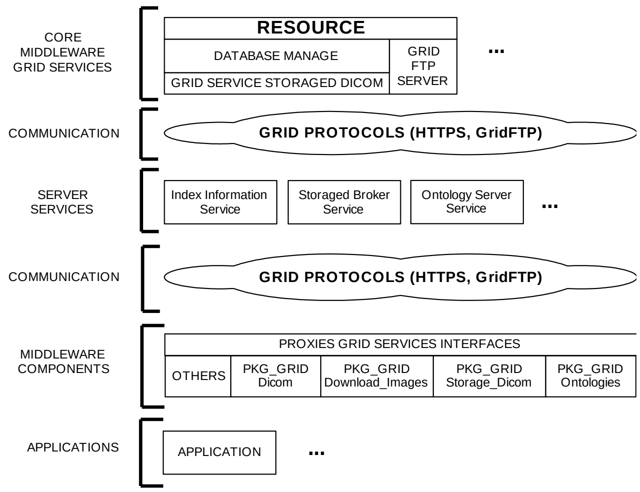

Figure 1. General scheme of the architecture.

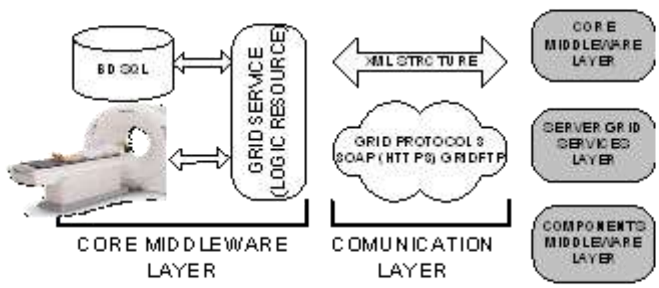

Figure 2. Iteraction betwen Layers of the architecture.

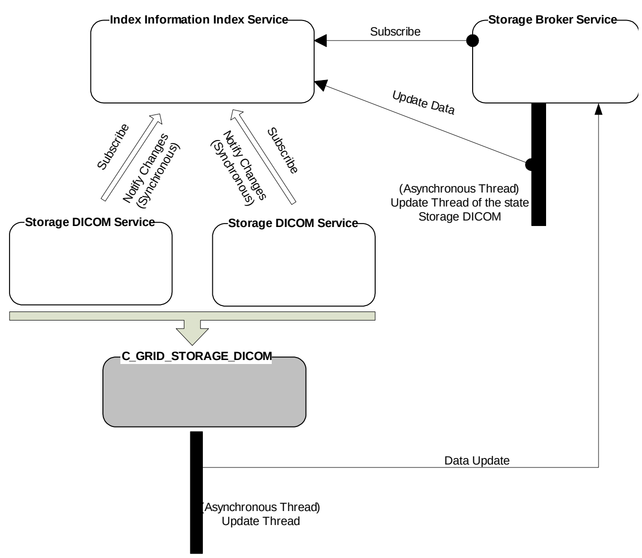

Figure 3. Diagram of connections among Grid Services.

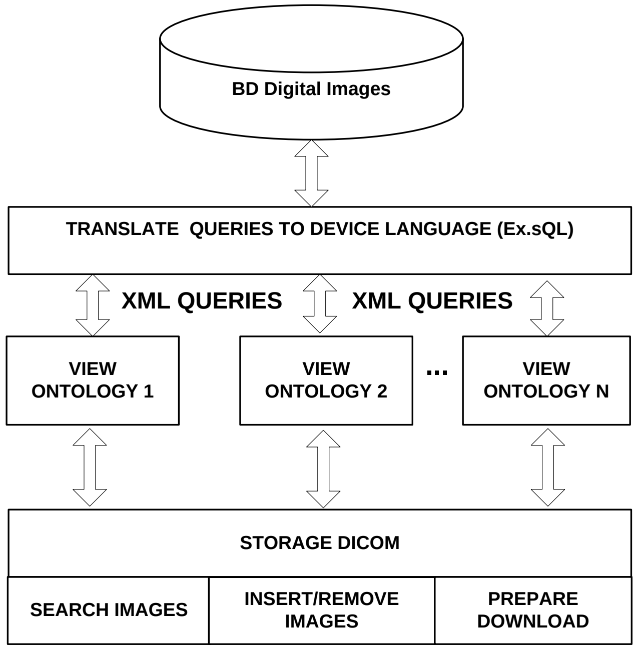

Figure 4. View of ontologies of Storage DICOM Grid Services.

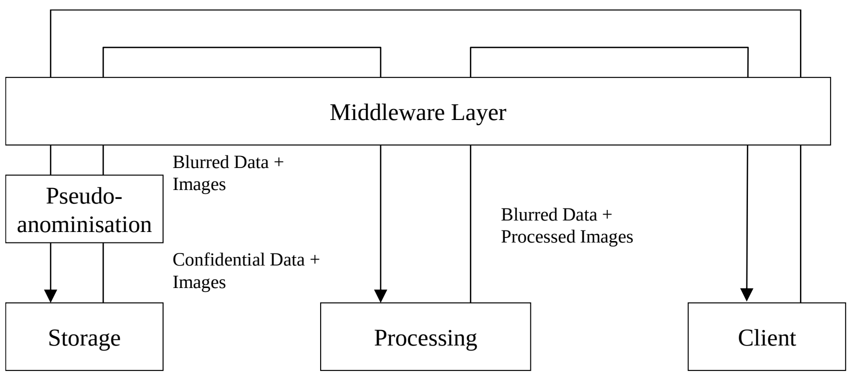

Figure 5. Security Diagram of Middleware.

Figure 6. Application for creating a Virtual Storage and searching DICOM images.

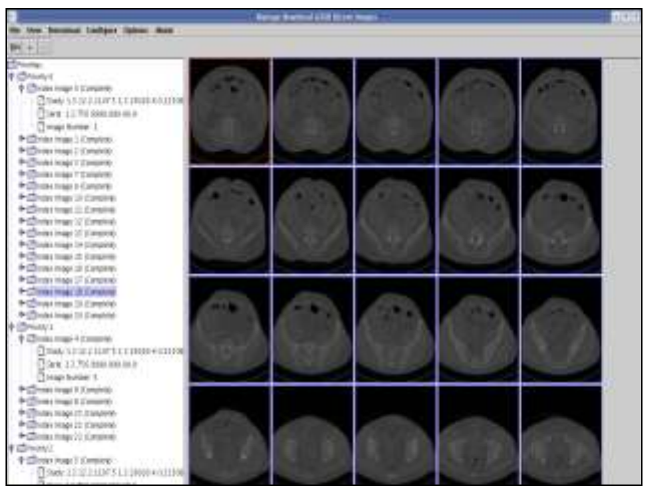

Figure 7. Application for downloading DICOM images enabling the user to define priorities.

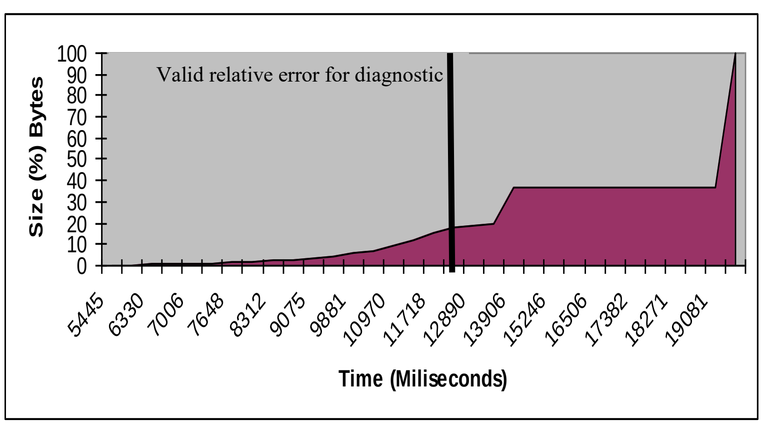

Figure 8. Evolution of the downloading of an image with progressive transmission. The vertical line shows the instant in which relative error goes below 2%.