

Document downloaded from:

<http://hdl.handle.net/10251/77357>

This paper must be cited as:

Blanquer Espert, I.; Hernández García, V.; Meseguer Anastasio, JE.; Segrelles Quilis, JD. (2009). Content-Based Organisation of Virtual Repositories of DICOM Objects. *Future Generation Computer Systems*. 25(6):627-637. doi:10.1016/j.future.2008.12.004.



The final publication is available at

<http://dx.doi.org/10.1016/j.future.2008.12.004>

Copyright Elsevier

Additional Information

## ***Title***

Content-Based Organisation of Virtual Repositories of DICOM Objects

## ***Authors***

Ignacio Blanquer Espert, Vicente Hernández García, Fco. Javier Meseguer Anastásio, J. Damià Segrelles Quilis

Members of the Institute for the Applications of Advanced Information and Communication Technologies (ITACA), Polytechnic University of Valencia (UPV).

Camino de Vera S/N, 46022

Valencia (Spain)

Tel: +34 96 387 7007 ext.88254; Fax: +34 96 387 7274

e-mails: [iblanque@dsic.upv.es](mailto:iblanque@dsic.upv.es); [vhernand@dsic.upv.es](mailto:vhernand@dsic.upv.es); [jmeseguer@itaca.upv.es](mailto:jmeseguer@itaca.upv.es); [dquilis@itaca.upv.es](mailto:dquilis@itaca.upv.es)

## ***Abstract***

The integration of multi-centre medical image data to create knowledge repositories for research and training activities has been an aim targeted since long ago. This paper presents an environment to share, to process and to organise medical imaging data according to a structured framework in which the image reports play a key role. This environment has been validated on a clinical environment, facing problems such as firewalls and security restrictions, in the frame of the CVIMO (Valencian Cyberinfrastructure of Medical Imaging in Oncology) project. The environment uses a middleware called TRENCADIS (Towards a Grid Environment for Processing and Sharing DICOM Objects) that provides users with the management of multiple administrative domains, data encryption and decryption on the fly and semantic indexation of images. Data is structured into four levels: Global data available, virtual federated storages of studies shared across a vertical domain, subsets for projects or experiments on the virtual storages and individual searches on these subsets. This structure of levels gives the needed flexibility for organising authorisation, and hides data that are not relevant for a given experiment. The main components and interactions are shown in the document, outlining the workflows and explaining the different approaches considered, including the protocols used and the difficulties met.

## ***Keywords***

Grid Architecture, DICOM Objects, Virtual Storages, Ontologies

## **Main Text**

### **1. Introduction**

The generalisation on the use of digital image format in hospital environments has fostered the evolution of many information technologies (storage, processing and data management). These technologies are changing the way in which radiologists work and cooperate. The use of Digital Imaging and Communications in Medicine [1] (DICOM) as a world-wide standard for the transmission and exchange of medical images in different modalities (CT, MRI, US, X-Ray) has made sharing medical images across a wide set of users and applications. Moreover, DICOM covers other radiology information such as radiology reports. Furthermore, DICOM Structured Reporting [2] (DICOM-SR) is a specification of tags and sections to write radiology reports with seamless references to findings and regions of interests of the associated DICOM studies. Structuring radiology reports is a very important task and constitutes a step beyond coding, since it offers a homogeneous way to structure the reports enhancing the automatic extraction of knowledge by searching and comparing reports.

Nowadays, the information systems managing DICOM objects and related patient data like Picture Archives Computer Systems (PACS), Radiology Information Systems (RIS) or Hospital Information Systems (HIS) work efficiently in medical environments. All these systems, along with standards as High Level Seven (HL7), openEHR or EN13606, as well as initiatives such as Integrating the Healthcare Enterprise (IHE) constitute an important effort in organising and structuring medical data within the departments of medical organisations. Those protocols and systems operate in secure private networks, dealing with a single administrative domain, in which user permissions are well defined.

However, those medical imaging protocols and systems are not efficient when sharing and managing medical data across different networks and administrative domains. This is the case, for example, of a Regional Health Care Networks in Spain. Moreover, those systems organise the data considering different administrative criteria, as they are used for healthcare delivery, rather than for researching and learning. Hospitals hold a huge amount of medical images, radiology reports and other DICOM data (videos, signals) which can be relevant in evidence-based medicine. Besides, RIS cannot be interrogated about cases with similarities to a given one. In this case, semantic organisation could exploit the intrinsic knowledge of such cases.

The work presented in this article describes a solution for interconnecting selected DICOM objects for training and decision support using the TRENCADIS system. It explains the components, relations and protocol stages implemented, which could be of interest for other implementations or for further work on the TRENCADIS system, whose release as open-source is being analysed. The article also describes a real deployment in which the DICOM objects, in this case images and structured reports, are stored in different repositories located in different hospitals, but organised by semantic criteria described by ontologies. The ontologies defined use three levels of information. The first level specifies the target area, where different groups of researchers work. This level is specified through the DICOM fields matching the values that define the target area. These fields are the DICOM header fields and the “concept names” of DICOM-SR objects (e.g. paediatric oncology target group could be defined by restrictions on the fields “age” of DICOM header of the image and the “Neoplasms” ICD codes on the “concept name” of “conclusions” field from the DICOM structured report). The second level is defined on top of the first level, and specifies the fields that can be referenced to create a specific study or experiment (e.g. in paediatric oncology, an experiment on “Neuroblastoma”, could specify restrictions for the topography of primary and secondary lesions to the area of the brain). For each experiment or study, a virtual storage is created, comprising the studies that match the criteria of the first and the second level. This virtual storage implies the creation of index tables at a central indexation resource. Finally, the third level works on top of the second level and defines the searching fields to filter the studies and reports included in the study or experiment. The searching fields could be any of the fields available in images and reports (sex, image modality, TNM staging, relapse status, specific topography, etc.). The users can make complex queries selecting the cases that correspond to a defined search (e.g. male patients above one year with findings of more than 3 mm in the temporal area).

Regarding the security infrastructure, it is important to outline that only the information of fields specified in the ontologies can be managed by the users authorized, even by local users with administrative privileges. Sharing is performed within the VO and images and reports must be structured in a way to enable concept-based searching.

The problem of indexing and organising Medical Imaging data is extensively considered in the literature. PACS and RIS organise data according to patient-centric information since they are oriented to healthcare delivery. They are insufficient for content-based searching and provide flat or user-based access permissions which cannot be extended to multi-domain structures. Data grid technologies such as SRB (Storage Resource Broker [4]) or gLite [5] provide general-purpose tools for storing large amounts of data in distributed environments. DICOM connectivity is provided through the development of DICOM plug-ins [14] [15] to standard protocols (such as the Storage Resource Manager or SRM [6] or Globus Toolkit [31]). Data annotation is performed by centrally storing metadata (such in the case of Medical Data Manager [14] on AMGA databases) or by more complex data representations (such the BIRN Lexicon). Other approaches consider also structured reports to enable context-based retrieval (such as Mammogrid [3] or NeuroBase [10][11]). In those approaches, metadata required for context-based searching is located on central repositories which store the annotation of the medical images. Authorisation is based on Access Control Lists based on login-password or X.509 certificates.

The use of central storage of metadata or annotations for context-based searching could lead to scalability problems. If the representation of knowledge is complex and the content is frequently updated, this service could become the bottleneck. The approach provided in this work proposes organising the relevant data by community interests. Central services will only store the

references to the distributed storages handling data belonging to each one of the community sets (represented by instances of an ontology in the article). Therefore, actual searching on metadata fields is performed distributed in the different repositories. This parallel approach also reduces overhead when new data is stored in the distributed storages, which will only update the central service if data from a new ontology instance is inserted. On the other side, this approach is fairly scalable for authorisation. Authorisation is based on the accessibility to a data ontology instance by a user group (although respecting local policies through exceptions). Then, creating a new user in an existing ontology set does not imply any distributed updating, as in the case of VO-based grid access policies, but enhancing the automatic rearrangement of the permission schema when new ontologies are created, since the relation to data and ontology instance is directly obtained from the context.

The paper is organized as follows. The second section provides a brief introduction of TRENCADIS<sup>1</sup> architecture and the element implemented to share DICOM objects using ontologies, considering the privacy of data in alien administrative domains. The third section describes the CVIMO<sup>2</sup> project, which is a real deployment that use the elements implemented. It also describes real scenarios and some results based on these scenarios. Finally, the conclusions will be presented.

<sup>1</sup> TRENCADIS (Towards a gRid ENvironment to proCess and shAre DIcom objects) is the Valencian Region denomination of a given mosaic type in which the pieces that compose it are irregular. The present work wants the knowledge construction with heterogeneous and different information, but when all these pieces are joined is built an object better.

<sup>2</sup> Project GVEMP06/004 funded by the Valencia Region Ministry of Enterprises, University and Science (*Consellería de Empresa, Universidad y Ciencia*) and coordinated by the Valencia University of Technology (*Universidad Politécnica de Valencia*).

## 2. TRENCADIS Architecture

TRENCADIS is a horizontal architecture that organizes virtual repositories of different types of DICOM objects. TRENCADIS is a Service-Oriented Architecture (SOA) in which the usage of resources is represented as Grid services based on the Open Grid Service Architecture [16] [17][18] (OGSA) specification. TRENCADIS is structured into several layers, which provide the developers with different abstraction levels. Figure 1 shows the infrastructure and application layers, linked through the communication layer.

Figure 1. General Diagram of TRENCADIS Architecture.

### Architecture Layers

An extended explanation of TRENCADIS architecture and the features associated to it can be found in [19]. As described in figure 1, TRENCADIS comprises three layers organized in five sub-layers. These sub-layers are the follows:

- **Core Middleware sub-Layer.** This layer implements a wrap-up for the basic resources of the medical infrastructure (DICOM repositories, PACS, etc.). The objective of this sub-layer, part of infrastructure layer, is to provide upper layers with a unique interface to all resources of the same type. Resources are made available through grid services using Web Services Definition Language (WSDL), protocols such as Simple Object Access Protocol (SOAP) or GridFTP and data formats such as XML Schemes. As an example given in the figure 1, DICOM objects can be structured on a relational database or a file directory, but the core middleware sub-layer will provide the same interface for them.
- **Server Services sub-Layer.** This layer is also part of the infrastructure layer, defines the services that implement server tasks. This layer interacts directly with the services from Core Middleware sub-layer and Server Services sub-layer and with the upper-level components of Components Middleware sub-layer. This sub-layer holds central services, as the code server that provides coding schemes, the VOMS [20] server organizing the VO users into VO groups, the ontology servers that provides ontology structures registered and the relation among VO user groups and ontologies and the Storage Broker that holds the indexation to create virtual storages, storing the location of candidate repositories for each case.
- **Communication Layer.** It defines the protocols that are used by the services implemented in the different sub-layers. GridFTP [21] protocol is used for transferring large amounts of data, and SOAP on top of HTTPS is used for interacting with the services.
- **Components Middleware sub-Layer.** This sub-layer contains the highest components that interact with services of the Core Middleware and the Server Services sub-layer. These components provide the applications with an object-oriented interface for the development of the applications for managing, processing and sharing DICOM objects in general. The components are used to build applications on top of grid services. This sub-layer is part of the user layer.
- **Applications sub-Layer.** This sub-layer comprises the applications for managing, processing and sharing DICOM objects. They make use of middleware components and provide the users with a friendly interface.

### Security Issues

The security is a key issue in the TRENCADIS architecture. 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 patient's treatment and restrictions to the medical research community.

Data security in the TRENCADIS architecture is based on the Grid Security Infrastructure [22] (GSI), plus several tools and procedures to enhance privacy. There are three main issues dealing with security in TRENCADIS:

- **Authentication and Service Access.** The protocols used in the communication Layer (SOAP and GridFTP) uses duly signed X.509 certificates and secure communication protocols based on Secure Socket Layer (SSL). In TRENCADIS, each service has a Gatekeeper that analyses the proxy certificate presented to the service (step 3, figure 2). The Distinguished Name (DN) and group and role attributes contained in the proxy certificate is compared with the lost of permissions for each service (step 3.1, figure 2). The local administrator creates an authorization scheme according to the VO agreements and his own per-site policies. According to the authorization scheme, users are allowed or denied (step 3.2, figure 2) to access the grid services.

Figure 2. Steps to accept a GridFTP transfer by the Gatekeeper component of a Grid Service.

- **Management of VOs.** Grid users are organized into VO groups. In order to access the services of the infrastructure, a Grid user opens a session. Sessions are initiated with the creation of a short-time x509 proxy certificate issued by a Virtual Organization Membership System (VOMS) server (step 1, figure 2), developed in the frame of the European DataGrid [24] and the DataTAG [25]. The VOMS Server validates the user and creates the proxy adding a standard extension x509 with group and role information. The session is closed when the proxy is destroyed or if it expires. The gatekeeper component

(step 3, figure 2) analyses the attributes from the proxy certificate presented, allowing or denying (step 3.2, figure 2) the access to the service according to the authorization policies implemented. If accepted, the gatekeeper manages the authorization to other services. If, for example, the user wants to transfer an image using the GridFTP protocol, the gatekeeper inserts the user's DN in the "gridmapfile" (step 3.3, figure 2) to allow the transfer (step 3.4, figure 2) of the image. Finally, data are transferred (step 4, figure 2).

- **Privacy of data.** Data privacy must be preserved even when data are stored on a remote location. TRENCADIS has been designed to interface with encryption and decryption through the use of distributed keys that ensure that even users at local sites with administrative privileges cannot access the information although if they could access the data files. Only users with the right credentials and VOMS attributes can access the key servers where the distributed keys are stored, and build the decryption keys to decrypt the object. The security model and the infrastructure developed have been presented in [23]. The operation of encryption and decryption is implicit in the middleware components, and thus, it is transparent to the programmers.

### Infrastructure Implemented

This section describes how the grid services implemented share DICOM repositories using ontologies. All grid services have been implemented following the Web Services Resource Framework [26] (WSRF) specification using the GT4.0.0 toolkit [27].

### VOMS Service

Different real domains could be managed as a single administrative domain in TRENCADIS. The architecture supports VO management through VOMS Service. The VOMS [20] Service is implemented in the Server Service sub-layers. There are other systems as CAS [28], PERMIS [29] or AKENTI [30], but VOMS uses standard extensions of x509 certificates compatible with OGSA/WSRF and organizes the VO in groups and roles. The VOMS Service manages user memberships, groups and roles in one or more VOs. The service is made available through two different interfaces: a client interface and an administration interface available only for the service administrator. Both interfaces are implemented with Web Services.

### Ontology Server Grid Service

This service is located in the Server Services sub-layer. As explained above, the ontologies are used to organize the whole data in different subsets (ontologic levels), depending on the interest group and the virtual community or experiment that is being performed.

The ontology of a DICOM object is based on the information considered by that the clinics, researchers and other medical users for their tasks. The ontology contains the attributes, relations and restrictions that are defined among these objects. All this information is extracted from DICOM and DICOM-SR fields.

Parts of this information could be of interest for the classification of the medical images. Moreover, the value of some of these fields can define not only the results of a query, but even the information available, since different experiments defined inside the same medical area or research group will focus on part of the information available to the whole community.

The specification language of the ontology is defined using XML. XML is sufficient for the definition of terms and rules at the level of the data when the complexity of the ontologies is reduced. DICOM and DICOM-SR semantic relations are simple mainly organised by means of tree-like structures.

When the semantics of the fields require semantically interoperability at different levels other approaches based in XML such as RDF, WSDL and OWL could be more convenient.

The ontologies consist on three parts, one by each level (target area, experiment or study and filter fields). All this information is extracted from fields of the DICOM header fields of all DICOM objects including the "concept names" fields of DICOM-SR objects.

Considering those needs, the language that specifies the ontologies in XML contains the following fields:

- **IDOntology:** Unique identifier of the ontology in the VO.
- **Description:** Brief description of the ontology.
- **TypeOntology:** These can be DICOM images or Structured Reports.
- **Restrictive:** DICOM fields that are defined by the DICOM tag that defines the restrictions of the virtual storage. These fields determine the criteria that must match all DICOM objects to be included for a medical area or research group.
- **Creation:** Defined by DICOM tags. These fields define the indexes that will be created in the virtual storages for accessing to the distributed information.
- **Filter:** Fields defined by DICOM tag that will enable the different searching criteria for retrieving the information.

Figure 3. Description of Lung Cancer Ontology in a XML document.

The ontology definitions define the classifications of DICOM information through association of pertinence. All information in the Ontology Server is kept in a SQL database, but the information is presented to the clients with XML documents like figure 3.

DICOM Storages are classified according to the ontologies defined in the Storage Broker Service in the VO. This service uses the “<RESTRICTIVE>” fields from the ontology definitions to identify the storage resources containing DICOM objects matching these restrictions and altogether considers the DICOM fields defined as “<CREATION>” to index these repositories. Finally, “<FILTER>” DICOM fields are used to define specific searching operations performed in parallel by the repositories matching the previous “<RESTRICTIVE>” and “<CREATION>” fields.

DICOM fields in the Structured Reports are coded in the form of a hierarchical structure, by means of “contains” or “specialises” relations. These relations are considered for the matching criteria.

This grid service manages the ontologies available and save the relations among VO groups and authorized ontologies. Only the users belonging to a group which has access to a given ontology, are able to manage the DICOM object related with this ontology.

It also updates the views of the repositories (figure 4) when the ontologies are added, removed, updated or activated in the Ontology Server Grid Service. The more relevant interfaces for this service are the following:

- **xmlCreateOntology.** Creation or deletion of an ontology in the Grid.
- **xmlGetOntology.** Retrieval of an ontology.
- **xmlGetOntologyTypes.** Returns the types of ontologies.
- **xmlGetOntologyData.** Returns data from given ontology.
- **xmlIsGroupAuthorized.** Returns if a group is authorized to manage a given ontology.

#### Storage DICOM Grid Service

The figure 1 shows this service in the Core Middleware sub-layer. This Grid Service offers the services that are needed for sharing DICOM objects using ontologies. Both a plain DICOM directory and a Relational Database storage resource have been implemented and are supported as back-ends. Figure 4 shows the ontology organisation in a Storage DICOM Service, in which a view is created for each ontology.

The fields used in these views are the fields that determine the target area and the filter fields of the experiments and studies. The ontology field values related to the experiments or studies are indexed by the Storage Broker Service. Once a DICOM object is inserted or removed, the indexation must be updated in the Storage Broker Service.

Figure 4. View of the ontologies from Storage DICOM Grid Services implemented on a Relational Database Backend.

Every operation has an ontology identifier associated, and can be executed if the user is authorized to work with the ontology. The more relevant interfaces are:

- **xmlInsert.** It inserts a DICOM object in the repository and updates the Storage Broker Service.
- **xmlSearch.** It searches DICOM objects in the repository and returns the ontology information associated to them.
- **xmlDownloadInit.** It prepares DICOM objects for downloading.

Figure 5. Relation among VO users-VO groups-Ontologies-DICOM objects.

The combination of VO groups, ontologies and DICOM repositories define the authorization to the DICOM data (figure 5). The information about VO groups is kept in a VOMS server, the definition ontologies and the relation among VO groups and ontologies is kept by the ontology server and the access to the information in the repositories depends on the user credentials and ontology presented, since DICOM data is inherently associated to an ontologies through the restrictions on the DICOM fields defined in the ontologies. It is possible that one study to be indexed in several different ontologies. The ontologies, created by the system administrator (nobody without VO administrative privileges could register a new ontology in a VO, since this could open indiscriminate access to the data) are the bases for creation the virtual storages. Users can access virtual storages if they present credentials with VOMS attributes including the reference to the corresponding ontology. Final decision is taken by the local repository gatekeeper, which could deny the access to individual users.

#### Storage Broker Grid Service

This service is located in the Server Services sub-layer, and it saves information related to DICOM objects from the repositories deployed. When a user creates a virtual storage, this service holds the information about which local repositories store studies that match the criteria of the virtual storage. The Storage Broker does not hold information about the individual studies, which is kept at the local repositories. This design increases scalability. The more relevant interfaces are the following:

- **xmlGetDICOMStorage.** It searches the repositories that match with the filters defined by the user to create a virtual storage for a specific an experiment or study.
- **xmlUpdateDICOMStorage.** It updates the information of a DICOM object in the indexing system.

Middleware components implemented

This section describes the middleware components implemented on top of the Grid Services presented in the previous subsection. All of them are located in the TRENCADIS Middleware Components sub-layer and are organized into different packages. Each package has a given functionality which is described as follows.

- **SESSIONS Package.** This package implements the classes to open and close Grid sessions to access the infrastructure. The C\_GRID\_Session objects allow executing the operations and saving the user credentials and the additional information when a session is open in its internal structure.
- **DICOM-I Package.** This package implements the classes that manage the DICOM medical images both individually or in groups. Multi-slice DICOM images are grouped by series (sequences of orthogonal slices obtained in the same scan) and studies (groups of series from the same examination), and can be referenced as single objects. The C\_GRID\_Image\_DICOM object constitutes the basic instance class that virtualizes a DICOM image. C\_GRID\_Set\_Images\_DICOMs objects deal with groups of DICOM images, individually managed through a C\_GRID\_Image\_DICOM object. To group the images with the same Series and Study identifier, a specialisation of the C\_GRID\_Set\_Images\_DICOM object is implemented through the C\_GRID\_Serie\_DICOM object. C\_GRID\_Study\_DICOM relate the images that have the same Study identifier. All these components interact directly with the DICOM Storage Service.
- **DICOM-SR Package.** This package is responsible of managing and creating new Structured Reports in DICOM format. This package defines the C\_GRID\_DICOM\_SR class, whose objects consult and retrieve the data from a DICOM-SR that has been previously downloaded. DICOM-SR objects can be managed in a group through the C\_GRID\_Set\_DICOM\_SR objects. All these objects interact directly with the DICOM Storage Services. DICOM SR objects are linked to studies, so they can be related to several DICOM files.
- **DICOM Virtual Storage Package.** This package deals with the storage and accesses to the DICOM images and DICOM Structured Reports that are distributed in the Grid Environment through DICOM repositories. All these DICOM objects are managed as a single virtual storage, regardless its real location. The DICOM Virtual Storage offers a single access point to all DICOM objects fulfilling the restriction of a given ontology. The Virtual Storage only offers searching criteria for the fields that are included in the ontology matching. The main class is the C\_GRID\_DICOM\_Storage. This class virtually gathers the DICOM Storage objects considering the creation fields of a defined ontology. This component enables searching, filtering, adding and removing DICOM objects in the DICOM Storage. It can be composed by one or several DICOM Storage Grid Services.
- **Upload /Download DICOM Package.** This package provides the classes for uploading DICOM Objects created using the C\_GRID\_SR\_DICOM or the C\_GRID\_Image\_DICOM objects in the client sites. It also provides the classes for downloading DICOM objects from the DICOM storage Grid Services to the user system by using the C\_GRID\_SR\_DICOM or C\_GRID\_Image\_DICOM objects. The class for uploading DICOM objects is C\_GRID\_DICOM\_Upload and for downloading DICOM objects is C\_GRID\_DICOM\_Download. Both use an instance of C\_GRID\_SR\_DICOM or C\_GRID\_Image\_DICOM class. Other class is the C\_GRID\_DICOM\_Set\_Download that manages the downloading of a set of DICOM objects. These classes also interact with the C\_GRID\_Download\_Order objects that define the order in which the C\_GRID\_set\_SR\_DICOM or C\_GRID\_Set\_Images\_DICOMs objects download the DICOM objects. These components manage the encryption and decryption using distributed keys transparently to the user.
- **Ontologies Package.** This package provides the classes for managing ontologies. C\_GRID\_Ontology is the instance class to create the XML documents that define the ontologies and C\_GRID\_Ontologies\_Server is the instance class to save the ontologies created with C\_GRID\_Ontology or to retrieve data related with a given ontology.

### 3. CVIMO Deployment

The CVIMO project is a deployment of TRENCADIS in which five hospitals of the land of Valencia and the Valencia University of Technology (UPV) cooperate to create a Virtual Storage of DICOM objects (medical images and structured reports). The hospitals involved are the University Hospital Dr. Peset, the Hospital of the Ribera, the University Hospital La Fe, Hospital Quiron, and the Valencian Foundation Institute of Oncology (FIVO).

Hospitals involving expert radiologists provide DICOM images and structured reports, which are initially stored in a dedicated resource. Relevant cases are sent to this resource to be shared within a given medical community. There are three virtual communities dedicated to three different areas (lung, liver and central nervous system). An ontology has been created for each area, defining the set of cases through the values of the fields DICOM required. The local repositories provide two interfaces. One interface (interface 1 in figure 6) connects the local repositories with the central grid services of CVIMO (VOMS Server, Ontology Server and Storage Broker), and the other (interface 2 in figure 6) connects it with the workstation of the hospital where the interesting cases are sent.



The central services are installed in the UPV, these are an indexation server to create virtual storages (Storage Broker Service) and an Ontology Server Service where the semantic organization is stored. Another central service is a web container where web applications are hosted. The applications of CVIMO can be accessed through only a web browser with the proper digital certificate.

There are two possible use cases. On the one hand, the expert radiologists provide the relevant cases and reports, and on the other hand radiologists and oncologist from the communities that query the cases.

Figure 6. Basic Scheme of the Deployment of CVIMO.

### Web Application

This section describes the web application developed on top of the Middleware Components developed in Components Middleware sub-layer. This application is located in the TRENCADIS application sub-layer and it is responsible for the transfer and formatting of the information for further processing or display using middleware components.

The web application uses HTTPs protocol and it is implemented and deployed using open-source tools. (Java Server Pages and Apache Web Services using the JAVATM 2 Platform and Apache Jakarta Tomcat 5 as the servlet container server).

The CVIMO web application has the following features:

- The application manages three ontologies about DICOM-SR related to three oncological areas (lung, liver and central nervous system cancer).
- The application integrates a visual interface that manages DICOM-SR as trees (figure 7), improving user friendliness. The user (expert radiologist) can create a DICOM-SR report and save it to the repositories.
- The application separates DICOM data between studies reported and not. A Study is considered reported when it has a DICOM-SR associated.
- The application provides user's communities with access to the reports and their associated images, which can be visualized through a plug-in.
- The user's communities can search for the reports through the search fields (figure 8) defined in the chosen ontology.

Figure 7. Snapshots of the user interface application.

Figure 8. Snapshots of the searching application.

### Real Scenarios

In this section four real use cases are presented. In these scenarios the interactions among grid services of the infrastructure are described. All these interactions are initialized by grid users through the web application. It is important to notice, that a grid session is needed to interact with the grid services.

The searching of DICOM objects and the creation of virtual storage have been tested in a controlled environment. The environment used is a local network with four computers SunFire V20Z (AMD Opteron single core 64 bits with 2GB of RAM). Each computer has installed Scientific Linux 3.0.7 with the toolkit GT4.0.0 to deploy the grid services. The StorageBroker and Ontology Server grid services have been installed in the same computer, and the StorageDICOM grid services are showed in table I, showing the total number of images stored and the number of images distributed by modality (CT-Computed Tomography or MR-Magnetic Resonance).

Table I. Computers where the StorageDICOM Service are hosted

The ontology server saves the ontologies. The general Ontology represents all the available images, so it does not contain any restriction of the first level fields. In the second level the "Modality" and "date Study" are included to create virtual storages and the third level contains specified generic fields as "birth date" or "body part" to do the filter operations.

### Ontology Creation

In the CVIMO deployment, each DICOM-SR ontology is defined through a combination of criteria from the values of "concept name" DICOM-SR fields. These criteria correspond to the first level of the ontology and will be the values of "concept name" fields of all DICOM-SR implied in the repositories that match the ontology (e.g. in the lung cancer ontology, the value of the report type field of DICOM-SRs must match with values of lung cancer).

Figure 9. Interactions when an ontology is created.

When a user creates an ontology, a C\_GRID\_Ontology instance is created first, then a C\_GRID\_Server\_Ontology instance is created to send the C\_GRID\_Ontology data to the Ontology server (step 1, figure 9). This information is kept by the server which sends the fields of second level to the Storage Broker service to initialize the indexes (step 2, 3 of figure 9). If the operation was completed successfully, the ontology server sends all the ontology fields to every DICOM-storage service enabled in parallel, which create a new view of the ontology (step 4, figure 9). Then the Storage DICOM Services send the values of the fields defined in the second level to the Storage Broker service to update the index of the new ontology (step 5, figure 9). Finally, the Storage Broker service notifies the Ontology server that the operation was finalized successfully.

### Virtual Storage Creation

In CVIMO, a Virtual Storage is created for each ontology. The Virtual Storages are related to experiments or studies (second level of ontology). In this case, the Virtual Storage references all DICOM-SRs and images included in the ontology. This implies that when the Virtual Storage is created, the second-level criteria are not considered.

Figure 10. Interactions when a Virtual Storage is created.

To create a Virtual Storage, an instance of C\_GRID\_DICOM\_Storage is created. In order to archive this, it is necessary that the user specifies the ontology and the second-level criteria of the ontology. First, the DICOM component asks the Ontology server if the user group presented in the credentials is authorized to manage the ontology (step 1, 2 of figure 10). If the user is authorized, the second-level criteria and the ontology identifier are sent to the Storage Broker Service (step 3, figure 10), which contact the indexation of repositories storing data that match the criteria. The Storage Broker service returns the URL of the DICOM repositories implied (step 4, figure 10) and the Virtual Storage is created.

In the test performed for this scenario, three virtual storages of the generic ontology described have been created, whose structure is shown in the table II.

Table II. Creation Times of Virtual Storages

The creation time of the virtual storage named “ALL” is shorter than the other storages. The overhead of the Storage Broker is lower since there are no restricts as in the other two cases.

### Searching DICOM objects

A CVIMO user can request more studies that match a potential diagnosis to compare to a, for example, complex case. It is not feasible to perform image-content bases searching like “give me a similar image to this”, as the similarities in concept could have different structures. So, the most effective way to search for similar cases is through the structured report.

Figure 11. Interactions when a DICOM-SR is searched.

When a user wants to search for studies given pathology, the active virtual storages are selected first. Then, the user defines the filter criteria from the fields defined in the third-level of the ontology. After that, the query is executed in parallel way through all Storage DICOM services that constitute the Virtual Storage (step 1, figure 11). The DICOM Storage services retrieve the results and when all Storage DICOM services have returned their results, a C\_GRID\_Set\_DICOM\_SR is created with the DICOM-SRs obtained.

Table III. Search Times of Virtual Storages

Table III shows the time spent by the components when executing a query based on different criteria in the test environment. The information regarding the time spent, the virtual storage used, and the number of images involved is included on the table. The searching time 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 data base manager. Obviously, the results involving a larger number of images require a longer searching time, the number the images are the same, the time is similar, independently of the virtual storage used.

### DICOM-SR Uploading

In CVIMO, the uploading of DICOM-SR happens when a Grid user wants to write the report of a DICOM Study.

Figure 12. Interactions when a DICOM-SR is uploaded.

An instance of C\_GRID\_SR\_DICOM is created, as well as another one for C\_GRID\_DICOM\_Upload associated to it. The C\_GRID\_DICOM\_Upload connects with the GridFTP Server of the DICOM Storage that holds the study searched transfers the data. The DICOM Storage Service updates the view of the ontology (step 1, figure 12) if is necessary. This is the case when a new case uploaded belongs to an ontology which is not related any other study in the server. (step 2, figure 12). Finally the Storage Broker service returns whether the operation has been successfully completed or not (step 3, figure 12).

#### 4. Conclusions

This article describes the architecture, components, organisational models and protocols used in the design and development of TRENCADIS middleware, as well as the results and deployment issues of a real infrastructure namely CVIMO. The article aims at discussing an approach to organise data according to their semantic content, focusing on the use case of medical imaging, and analysing different scenarios. The work covers the whole production cycle, from requirements to production.

The TRENCADIS middleware and the CVIMO deployment propose a way to organise data integrated with a distributed authorisation model which is complementary to other approaches and techniques on the literature. The performance of the system is reasonable for interactive use and the security model does not present reasonable threats. All the work developed is based on standards, such as DICOM, WSRF, XML, X509 and medical informatics terminologies. The structure of the structured report is compliant to DICOM-SR and can be adapted as new templates and versions of the standard are available.

Finally, it is important to outline that deployment is a complicated task considering the technical and administrative limitations of health networks. The involvement of experts and decision makers from the departments responsible of health networks is very important for defining the rightmost protocols and connection models. Many of the standard protocols used in grid technologies, such as GridFTP or VOMS ports are inadequate and need to be reconfigured.

TRENCADIS constitutes an efficient framework for integrating storage distributed resources. Notwithstanding these resources can store the information in any system (relational databases, hard disks etc...) and users can access data through GridFTP protocol. A future works can tackle the standardization of the access to data using other protocols such as SRM [32] or OGSA/DAI [33].

This framework is being extended to deal with processing services. Data grid infrastructures based on TRENCADIS can be extended with computing resources to deal with High-Throughput Computing problems, involving parameter sweep or sets of studies. Other works have been presented in this sense [34], which require extending the life-time of the proxies through MyProxy repositories (e.g. in the BiG project [35]).

Workflow orchestration and Quality of Service techniques are being applied to synchronise processing with data access, to improve the exploitability of the system.

#### 5. Acknowledgment

The authors wish to thanks the financial support received from Valencia Region Ministry of Enterprises, University (*Consellería de Empresa, Universidad y Ciencia*) to develop the project “Ciberinfraestructura Valenciana de Imagen médica Oncológica”, with reference GVEMP06/04.

#### 6. References

- [1] 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.
- [2] D. Clunie. “DICOM Structured Reporting”. ISBN 0-9701369-0-0.
- [3] S R Amendolia, F Estrella, W Hassan, T Hauer, D Manset, R McClatchey, D Rogulin, T Solomonides. “MammoGrid: A Service Oriented Architecture based Medical Grid Application”. Lecture Notes in Computer Science. ISBN 978-3-540-39090-9. Volume 3251, pp. 939-942 (2004).
- [4] Moore, R.W, Wan, M., Rajasekar, A. “Storage resource broker; generic software infrastructure for managing globally distributed data.”. Local to Global Data Interoperability - Challenges and Technologies, 2005. ISBN: 0-7803-9228-0
- [5] gLite. Lightweight Middleware for Grid Computing . <http://glite.web.cern.ch/glite>, visited on January 2008.
- [6] Arie Shoshani, Alex Sim, Junmin Gu. “Storage Resource Managers: Middleware Components for Grid Storage”. MSS 2002.
- [7] “eDiaMoND Project”. <http://www.ediamond.ox.ac.uk/objectives.html>, visited on March 2007.
- [8] Bottigli, U.; Cerello, P.; Cheran, S.; Delogu, P.; Fantacci, M. E.; Fauci, F.; Golosio, B.; Lauria, A.; Lopez Torres, E.; Magro, R.; Masala, G. L.; Oliva, P.; Palmiero, R.; Raso, G.; Retico, A.; Stumbo, S.; Tangaro, S. “GPCALMA: A Tool For Mammography With A GRID-Connected Distributed Database”. MEDICAL PHYSICS: Seventh Mexican Symposium on Medical Physics. AIP Conference Proceedings, Volume 682, pp. 67-72 (2003).
- [9] A. Lauria, R. Massafra, S. Tangaro, R. Bellotti, M.E. Fantacci, P. Delogu, E. Lopez Torres, P.Cerello, F. Fauci, R. Magro, U. Bottigli “GPCALMA: An Italian Mammographic Database of Digitized Images for Research”. Lecture Notes in Computer Science. ISBN 0302-9743 (Print) 1611-3349 (Online). Volume 4046, pp. 384-391 (2006).
- [10] C. Barillot, R. Valabregue, J-P. Matsumot, F. Aubry, H. Benali, Y. Cointepas et al.”NeuroBase : Management of Distributed and Heterogeneous Information Sources in Neuroimaging.”. DiDaMIC-2004, a satellite workshop of Miccai-2004.
- [11] “myGRID proyect.” <http://www.mygrid.org.uk>, visited on March 2007.
- [12] “BIRN. Biomedical Informatics Research Network”. <http://www.nbirn.net>, visited on March 2007.
- [13] “CaBIG. Cancer Biomedical Informatics Grid”. <https://cabig.nci.nih.gov>
- [14] J. Montagnat, Á. Frohner, D. Jouvenot, C. Pera, P. Kunszt, B. Koblitz, N. Santos, C. Loomis, R. Texier, D. Lingrand, P. Guio, R. Brito Da Rocha, A. Sobreira de Almeida and Z. Farkas, “A Secure Grid Medical Data Manager Interfaced to the gLite Middleware”. Journal of Grid Computing (JGC). ISSN: 1570-7873 (Print) 1572-9814 (Online). Volume 6, Number 1 / March de 2008. Pages 45-59. Springer Netherlands, 2007

- [15] Dagmar Krefting, Julian Bart, Kamen Beronov, Olga Dzhimova, Jürgen Falkner, Michael Hartung, Andreas Hoheisel, Tobias A. Knoch, Thomas Lingner, Yassene Mohammed, Kathrin Peter, Erhard Rahm, Ulrich Sax, Dietmar Sommerfeld, Thomas Steinke, Thomas Tolxdorff, Michal Vossberg, Fred Viezens, Anette Weisbecker. "MediGRID: Towards a user friendly secured grid infrastructure". *Future Generation Computer Systems (FGCS)*, Volume 25, Issue 3, March 2009, Pages 326-336
- [16] "Open Grid Services Architecture (OGSA)", <http://www.globus.org/ogsa>. visited on May 2007.
- [17] I. Foster, C. Kesselman, and S. Tuecke. The anatomy of the Grid: Enabling scalable virtual organizations. *International Journal of Supercomputer Applications*, 15(3), 2001.
- [18] I. Foster, C. Kesselman, J. Nick, S. Tuecke. The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration. Open Grid Service Infrastructure WG, Global Grid Forum, June 22, 2002.
- [19] I. Blanquer, V. Hernandez, D. Segrelles. TRENCADIS. A Grid Architecture for Creating Virtual Repositories of DICOM Objects in an OGSA-based Ontological Framework. ISSN: 0302-9743, ISBN-10: 3-540-68063-2, ISBN-13: 978-3-540-68063-5. Issue: Volume 4345, 2006. Pages 183-194.
- [20] R. Alfieri, R. Cecchini, V. Ciaschini, L. dell'Agnello, Á. Frohner, K. Lórentey, F. Spataro. "From gridmap-file to VOMS: managing authorization in a Grid environment". *Future Generation Computer Systems (FGCS)*. Volume 21, Issue 4, April 2005, Pages 549-558.
- [21] "The GridFTP Protocol and Software", <http://www-fp.globus.org/datagrid/gridftp.html>
- [22] R. Oppliger. "Security technologies for the World Wide Web". Second edition. Computer Security Series. Artech House. ISBN: 1580533485
- [23] I. Blanquer, V. Hernandez, D. Segrelles, E. Torres. TRENCADIS – Secure Architecture to Share and Manage DICOM Objects in a Ontological Framework Based on OGSA. ISBN 978-I-58603-783-3. Volume 126, 2007. Pages 115-124.
- [24] M. Draoli, G. Mascari, R. Piccinelli, Project Presentation, DataGrid-11-NOT-0103- 1.
- [25] "The Datatag Project". <http://www.datatag.org>, visited on May 2008.
- [26] "The WS-Resource Framework". [www.globus.org/wsrp](http://www.globus.org/wsrp), visited on May 2008.
- [27] "Globus 4 toolkit". <http://www.globus.org/toolkit>. visited on May 2008.
- [28] L. Pearlman, V. Welch, I. Foster, K. Kesselman and S. Tuecke, A Community Authorization Service for Group Collaboration, IEEE Workshop on Policies for Distributed Systems and Networks (2002).
- [29] V. Breton, K. Dean and T. Solomonides (Eds.), "The Healthgrid White Paper", in: *From Grid to Healthgrid – Proceedings of Healthgrid 2005*, IOS Press, Studies in Health Technology and Informatics, pp 249-321, Vol. 112, 2005.
- [30] M. Thompson, A. Essiari and S. Mudumbai, "Certificate-based Authorization Policy in a PKI Environment", *ACM Transactions on Information and System Security (TISSEC)*, 6(4):566-588, November 2003.
- [31] S. G. Erberich, J.C. Silverstein, A. Chervenak, R. Schuler, M. D. Nelson, C. Kesselman, "Globus MEDICUS – Federation of DICOM Medical Imaging Devices into Healthcare Grids" – *From Genes to Personalized Healthcare: Grid Solutions for the Life Sciences*. pp. 269-278. Ios Press, Studies in Health Technology and Informatics, 126. ISBN 978-I-58603-738-3.
- [32] The Storage resource Manager Interface Specification Version 2.2. "<http://sdm.lbl.gov/srm-wg/doc/SRM.v2.2.html>", visited on December 2008.
- [33] "OGSA/DAI." <http://www.ogsadai.org.uk/>, visited on December 2008.
- [34] I. Blanquer, V. Hernandez, D. Segrelles, E. Torres. "Introducing High-Performance and High-Throughput Processing in the TRENCADIS Data Sharing Architecture". *The Second International Conference on Advanced Engineering Computing and Applications in Sciences (ADVCOMP 2008)*. ISBN: 978-0-7695-3369-8. Valencia (Spain), 2008.
- [35] G. Aparicio, I. Blanquer, V. Hernández, D. Segrelles. "Blast in Grid (BiG): A Grid-Enabled Software Architecture and Implementation of Parallel and Sequential BLAST.". *Spanish Conference On e-Science Grid Computing*. ISBN: 978-84-7834-544-1. Madrid (Spain), 2007.