A Centralized Platform for Geo-Distributed PACS Management

Luís A. Bastião Silva · Renato Pinho · Luís S. Ribeiro · Carlos Costa · José Luís Oliveira

Published online: 22 October 2013
© Society for Imaging Informatics in Medicine 2013

Abstract Picture Archive and Communication System (PACS) is a globally adopted concept and plays a fundamental role in patient care flow within healthcare institutions. However, the deployment of medical imaging repositories over multiple sites still brings several practical challenges namely related to operation and management (O&M). This paper describes a Web-based centralized console that provides remote monitoring, testing, and management over multiple geo-distributed PACS. The system allows the PACS administrator to define any kind of service or operation, reducing the need for local technicians and providing a 24/7 monitoring solution.

Keywords Medical Imaging · PACS Management · DICOM Monitoring · PACS Testing · Geo-Distributed PACS

Introduction

Picture Archive and Communication System (PACS) is shifting from an institution-centric to a patient-centric model, reflecting increasing patient mobility between healthcare

institutions during care provision. This endeavor has also boosted the need for inter-site and regional PACS [1].

Healthcare institutions have adopted PACS to store, access, and visualize images, relying on Digital Imaging and Communication in Medicine (DICOM) standards for communications. The availability of PACS in these critical scenarios is indispensable to avoid losing medical exams and to support clinical reporting and decision workflow. However, it is almost impossible to avoid failures in the systems due to network connection problems, software, hard-disk drive, computer hardware, and/or human error [2].

Redundancy in software architecture design allows system recovery after a software failure [3]. Several replication systems and fault-tolerant PACS have been developed in recent years [4, 5]. Nevertheless, redundancy in software and hardware has high costs and some healthcare centers cannot afford this [6]. Furthermore, other operational risks are associated with network issues, human error, or software crashes that might cause problems, even considering redundant architectures. Eventually, human intervention will be necessary to fix unpredicted errors that fault-tolerant mechanisms could not automatically repair. Therefore, detecting system malfunctions as soon as possible is mandatory in order to trigger emergency routines and restore normal PACS workflow without jeopardizing patient safety [7, 8]. Managing a geo-distributed PACS is much more demanding than dealing just with one single PACS, especially when high levels of availability are required [2, 7]. Therefore, preventive measures must be in place, such as continuous monitoring and periodic testing. More importantly is to trigger an action, as soon as possible, when such unexpected events occur. Hence, this paper presents the development of a Web tool that detects system failures before end users realize, notifying administrators, system vendors, or other entities responsible for the PACS well functioning. This O&M console may be configured to control any DICOM compliant

L. A. B. Silva · R. Pinho · L. S. Ribeiro (✉) · C. Costa · J. L. Oliveira
University of Aveiro, DETI/IEETA, Campus Universitário de Santiago, 3810-193 Aveiro, Portugal
e-mail: luisribeiro@ua.pt

L. A. B. Silva
e-mail: bastiao@ua.pt

R. Pinho
e-mail: renato.pinho@ua.pt

C. Costa
e-mail: carlos.costa@ua.pt

J. L. Oliveira
e-mail: jlo@ua.pt

device, monitoring and managing several geo-distributed PACS components. With this solution, the PACS administrator has a centralized view of the geo-distributed PACS and may trigger system recovery scripts remotely (i.e., via Web browser). Moreover, it offers a test framework that supports intensive test routines over the DICOM services, ensuring that a remote node of the distributed PACS is functioning properly over time. PACS administrators or developers may personalize the level of tests, to define custom storage and queries. Finally, it is possible to extract indicators from those programmed tests to support quality control procedures.

The manuscript is organized as follows: **Background** Section introduces several related works pointing out the advantages and disadvantages of each; **Methods** Section describes the methods used to implement the monitoring tool and **Results** Section presents the results and benefits of using the developed tool; **Case Study and Discussion** Section presents the case study and discusses the main contributions of this solution and, finally, **Conclusions** Section presents conclusions.

Background

DICOM Standard

The DICOM standard was established in 1992 and has since become the standard for exchanging medical images in digital format [9, 10]. DICOM facilitates the implementation of PACS and enables interoperability in a multi-vendor environment [11]. This non-proprietary standard is very complete and has many dimensions. However, it is best known for the definition of objects (DICOM objects including images, waveforms, studies, etc.) and for operations applied to these objects such as storing, moving, finding, creating, printing, and many other operations. Operations in DICOM are called Service Classes or, simply, DICOM services, and typically involve two application entities, identified by a unique title (AETitle). One entity assumes the client role, service class user, and the other entity assumes the server role, the service class provider [12]. Each service is a two-step process: negotiation and exchange. The negotiation step (association establishment) is critical to guarantee that the two entities understand and interpret the information sent in the exchange step. After the negotiation has successfully been completed, the exchange step starts. Formally, the negotiation step is known as the DICOM association establishment and the exchange information step occurs within the DICOM association [11]. The exchanged information (Service–Object Pair) is defined as a combination of a command, DICOM Service Element (DIMSE), and an object. Examples of DIMSE are c-store, c-get, c-find, c-move, etc. Examples of

objects are computed tomography (CT) images, magnetic resonance images, waveforms, etc.

The DICOM does not offer mechanisms to monitor deeply a DICOM node but it supplies a verification service (commonly known as DICOM “ping”). It checks if the DICOM services are up or down. Besides verification, the most used DICOM services are storage, query/retrieve and modality work list. All of these are based on the atomic commands of c-store, c-find, and c-move.

PACS Operation and Management

The overwhelming growing of communication networks during the last decades has also lead to an increasing importance of network and systems management solutions. For instance, Simple Network Management Protocol (SNMP) [13] became the standard protocol to manage and monitor network devices, including routers, switches, servers, workstations, printers, and other equipments. It has been used over the years by manufacturers to detect network failures caused by problems such as overload and/or crash servers and network connection.

There are also many software solutions for network monitoring, such as Nagios¹ and Zenoss.² Both provide a Web interface that allows system administrators to monitor availability, inventory/configuration, performance, and events. They are supported by the most common operating systems and also rely on SNMP monitoring. However, these monitoring systems are of a broader nature and do not take into account the specificities of the medical imaging workflow. PACS integrates medical images and patient databases involving many components, such as image modalities, computers, data storage, display workstations, network switches, and computer servers. All components, including network connections, are technically called Single Points of Failures [5, 7, 14]. Thus, it is necessary to develop systems to monitor this kind of system, because they are critical in healthcare institutions and involve patient safety.

Weisser et al. [15] built a system to check the online availability of teleradiology components. They used several open-source tools, such as Nagios, to monitor the availability of a system relying on email protocol to transfer images between different centers in Germany. The authors measured the line speed, e.g., reporting low bandwidth, and they observed daily malfunctions of end users. Nevertheless, this monitoring system relies entirely on email protocol servers and works only on their teleradiology scenario.

In [16] the authors built a system based on Nagios to monitor PACS, RIS, and overall machinery in a radiology department. This system used the features available in Nagios

¹ <http://www.nagios.org/>

² <http://www.zenoss.org/>

to port monitoring and created a script to check availability of a DICOM services via the verification facility. It also supports automated restart of services through pre-configured scripts. The main advantage is the use of well-known protocols and systems, such as ICMP, HTTP, MySQL, and DICOM. The proposed method helps IT professionals to identify port failures, but more specific PACS operational errors are not detected, such as a DICOM service, that cease to work despite the server being able to respond to the DICOM verification.

DVTk [17] is an open-source DICOM integration tool which is very useful during the development of the DICOM services and it may be used to test deployed PACS environments. A drawback of this tool is not being multiplatform and is not run natively in UNIX operation systems. Also, it is hard to be integrated in the test scripts because they rely on a graphical user interface to setup the configurations and run the tests.

The lessons learned from the analysis of several PACS monitoring and test tools, helped us to design and setup our proposal. Besides enhanced monitoring routines, our system integrates a testing suite and management functionalities delivered as a service to its users. Our main motivation was to support the deployment of geo-distributed PACS archives, including the support for administrators to continuously ensure the well functioning of the entire solution.

Methods

Medical imaging equipment has gradually become more powerful and less expensive, resulting in its proliferation from large to small imaging centers. This scenario led to more efficient healthcare processes where the PACS serve multiple sites, fostering inter-institutional data exchange workflows [18]. All those systems use the DICOM standard to supply their services. Therefore, the core of our monitoring system is based on DICOM services. To implement the services, we used the dcm4che toolkit. The dcm4che toolkit is a popular open source and multiplatform implementation of the DICOM Standard developed in Java programming language [19]. Therefore, interoperability is ensured with the DICOM services, as well as portability. The architecture of our system reflects the need to easily access distributed resources, independently from the PACS administrator location. Hence, the distributed environment is managed entirely via a centralized Web browser where all events are aggregated. Figure 1 illustrates the overall architecture, which is composed by two entities: the aggregator and the probe.

The aggregator is the central device of the solution, working as an event repository and as a bridge that forwards the management actions to the various probes of the system. Furthermore, it is also responsible for notifying the administrator whenever PACS status changes, by updating

the information on the Web page frontend and/or by triggering alerts to the administrator (e.g., email, SMS). The aggregator was developed using Python and Django and it is supported by a MySQL database to store the events. In order to support notifications, we used Django-celery, a backend tool to execute background tasks, which sends emails to the subscribers if the target service goes down. The aggregator provides also a RESTful application programming interface (API) for the reporting of events. However, to assure access control, this APIs may only be used if the right credentials are provided.

The probe is a software agent that monitors and enforces the administrator orders inside the geo-distributed domains. The probe interacts with PACS actors through DICOM services. However, the tasks based on DICOM services are not enough to manage the PACS ecosystem. Therefore, we enabled our system to inject and execute specific management scripts in the remote sites. Hence, the probe must be installed in the same machine of the PACS actor to be managed with scripting (e.g., PACS archive in case of Fig. 1).

Monitoring

Within the healthcare institution, the probe periodically confirms the proper functioning of the DICOM services. Besides checking the availability of the DICOM services through C-ECHO command, our solution also tests the services with the actual commands used by DICOM (e.g., C-FIND, C-STORE). Although the administrator may configure the periodicity of the commands, the C-ECHO command by default has much higher frequencies than the remaining commands. It is periodically triggered and when it fails twice consecutively, the probe immediately informs the aggregator about a malfunction with the target DICOM service (i.e., offline). The service has three operational levels: online, offline, and unknown. The unknown state occurs when it is not possible to determine if the DICOM service is online or offline, typically due to a TCP/IP communication fault between the aggregator and the probe. The other commands have a lower frequency and serve two purposes. First, to ensure the correctness of the DICOM service, since the C-ECHO command only checks its availability and not its proper functioning (e.g., if the C-ECHO is answering, it does not mean that the C-STORE is accepting and storing the images). Second, it is used to measure the services response time for statistical purposes. To monitor PACS devices, we developed a Python script that relies on DcmTk³ to invoke DICOM C-ECHO against the services. The administrator only needs to provide the AETitle, the IP, and Port address of the service. After this configuration, the script is autonomously generated and forwarded by the aggregator to the target probe(s), becoming ready to be used within the private network.

³ <http://dicom.offis.de/dcmTk.php.en>

Fig. 1 General architecture of PACS distributed management

On the other hand, the aggregator uses a publisher/subscriber mechanism, i.e., only the subscribers of the specific service will be contacted. The system administrator may subscribe to all services and be notified when the online status of a service changes to offline or unknown. Besides notifications, the system also allows users (e.g., clinicians) to subscribe temporally to the monitoring of a service. For instance, permits to know when a specific service (e.g., DICOM Query/Retrieve) comes online again. Furthermore, there are some periods when PACS is intentionally offline. For example, some imagiology centers only work between 8 a.m. and 7 p.m. In such scenario, to avoid notifications of minor faults out of this period, the administrator may deactivate those alarms. Moreover, it is possible to schedule maintenance periods. Probe events will be ignored during this period and nobody will be notified when the specific service stops answering.

In order to process all these events, we run a Django-celery to execute background tasks. It schedules tasks and periodically checks the last events. If there are no events for a specific service, it generates the unknown event—meaning that the probe is not answering the aggregator system. Thus, subscribers will be notified about the problem on the probe side.

Finally, for security reasons, the probes invoke the Web services with an API key and a secret key that are generated in the aggregator for each institution. Each probe, or set of probes, uses these credentials to send the service status to the aggregator.

Management

Besides detecting and reporting failures in DICOM services, our solution contemplates other issues. For instance, if a service stops (crashes) for some unknown reason, it must be

restarted. Initially, we created script hooks placed on the probe to restart the service remotely. Later, we extended this concept enabling the PACS administrator to push new scripts to the selected probes. The system administrator should write these scripts, uploading them to the aggregator through the frontend. The administrator is responsible for uploading proper scripts, taking into consideration the operating system where the probe and the PACS actor are installed (e.g., bash scripts for UNIX or batch files for Windows). We opted for scripting because it is a language that PACS administrators typically use on a daily basis. Furthermore, the aggregator saves the uploaded scripts to be reused in future managing episodes.

Testing

Embedded in the probe agents, there is a test framework, capable of running a battery of tests against the PACS archive. This facility ensures that the distributed PACS archives are stable enough before entering in production mode. The test facility was developed based on dcm4che2 library.⁴

Our test battery allows the detection of internal problems in the PACS archive. Figure 2 illustrates the message flow of the test suite. Dummy DICOM objects are pushed into the PACS archive through the C-Store command. Then, the probe starts querying the archive for these objects. The C-Find responses are compared with the pushed objects, in order to check if the queries are consistent. Finally, the test results, including quality metrics, are uploaded to the aggregator becoming available in the administration frontend. Nevertheless, the test results are extensive and reading them in their raw format would lead to reading errors. Therefore, we present the administrator with retrieval accuracy metrics such as the Mean Average Precision, commonly used to evaluate the relevance of retrieved information [20, 21].

Results

The developed and presented tool is useful for monitoring and managing multiple devices in healthcare departments. This tool may be used in the deploying and managing of geo-distributed PACS because it is often a demanding process. It supplies a Web-based interface that may be run in any device, such as, computers, tablets and also mobile phones. It is easy to track the status of the overall system and also be notified if some issue happens. By adopting our system, the IT team leverages several advantages and useful features for ensuring the proper functioning of the inter-site PACS:

- Effective and efficient management of the geo-distributed PACS. By combining in the same tool monitoring, management and testing functionalities, it provides the IT team with real-time and complete information about the state of geo-distributed PACS (Fig. 3). This enables preventive fault detection which leads to high availability of the PACS services.
- Problem resolution and extensibility. Besides notifying when failures are detected, it allows the administrator to remotely fix the problem by triggering previously saved routines or by feeding new routines to the malfunctioning node(s). After the problem resolution, the administrator may run predefined tests, in order to ensure full system recovery.
- Convenience on access and setup. All system functionalities are accessed and configured via Web browser (Fig. 3), enabling the interaction with the system, as long there is an Internet connection available—which is useful, due to PACS' 24/7 availability requirements. Furthermore, the offered functionalities are delivered as a service (Software as a Service—SaaS). Therefore, it is not required to install or update supporting software (e.g., databases).
- System integration. Besides the known benefits of the SaaS model, the system is ready to manage, monitor, and test any PACS device, as long it is DICOM compliant.
- Operational dashboard. Through the Web browser control panel, the IT team may consult charts and logs feeding back the historical availability and performance status of the PACS. In fact, it is possible to record several operational logs with customized granularity. Those records are processed and stored in a database that feeds the graphical dashboard with several runtime parameters like, for instance, the impact of every service in the imaging network load.

Overall, the IT team is supported by a synergy of functionalities combined in a centralized manner, which is, from our experience, a valuable aid in the deployment, administration, and quality assurance of the geo-distributed PACS.

Case Study and Discussion

The presented solution was deployed in a geo-distributed PACS, encompassing two distinct institutions (Fig. 4). Before these distributed solution, healthcare professionals did not have a central archive on each site—several workstations were used as archive. With the distributed PACS in place, both sites started to use and to store all DICOM objects produced in a single common archive, located at site A. The central archive relies on Dicoogle [22], an open-source PACS archive compliant with DICOM standard. Furthermore, to connect

⁴ <http://sourceforge.net/projects/dcm4che/>

Fig. 2 Test battery—example of message flow in the PACS archive

the two sites transparently, DICOM Cloud Router [23] was chosen which creates a reliable abstraction layer between the two sites, enabling the DICOM compliant devices to transparently communicate between sites, as if the communication was performed in the same node.

The archive was placed in a machine with eight Intel cores i7-2700 K with 3.50 GHz and 16 GB of RAM. This machine

contains a 6 TB of storage with RAID 1 and one disk with SSD to support reading and writing at high speed. At the other site, the modalities store the exams on the site's Cloud Router. The Cloud Router has a short-term cache that stores the most recent studies to serve the site more efficiently. However, when the study is stored in the short-term cache, the router forwards it in parallel to the central archive. The Cloud Router

Fig. 3 Aggregator's Web frontend: showing the status of two institution units and monitoring setup

Fig. 4 Monitoring use case based on two distinct PACS

was placed in a machine with four atom cores with 1.8 GHz and 2 GB of RAM and a hard-drive with 500 GB. Both machines, archive and Cloud Router, are supported over Linux with Ubuntu Server 12.04 LTS.

The healthcare institutions, in this scenario, provide three acquisition services: computer radiology, ultrasound, and CT. The management system was assessed during 3 months, between November 2012 and January 2013, corresponding to a median of 5,000 studies per month. In each center, there is one computer topography, one mammography, two digital computer radiology scanners, and six echocardiographs. In all the process, there are four medical doctors, six people from clinical staff, two people from IT, and two supporting technical staff. Both sites have the same modalities. In order to reduce the number of hardware resources and their associated costs, it was decided to only store medical images in one datacenter.

Deployment Evaluation

The deployment of the use case was performed in two different stages. The first one was spent setting, configuring and testing all PACS actors, guaranteeing that each new added

actor communicates properly with the remaining systems. The second stage aimed to ensure the continuous availability of clinical workflow. We installed in each site a probe to monitor and execute the management requests. Furthermore, the system continuously tests intra-site and inter-site communications, in order to allow earlier detection and notify possible malfunctions.

At the first stage, we used the management suite to flood the PACS network with heavy tests, mimicking the behavior of a task intensive care flow. In parallel, the monitoring role oversaw flow events reporting several issues, such as:

- Maximum number of associations. Since the central archive serves both sites, it has to handle multiple service requests in parallel. Our system detected that the first archive configuration did not allow enough DICOM association in parallel.
- Voracious cache system. The Cloud Router does not delete automatically previous stored studies. Therefore, the entire available space will be eventually filled, leading to a memory fault. The probe allows notifying when space is becoming scarce. Afterwards, the administrator may remotely activate the execution of a script to free space

in the cache. This event also led to corrective actions, namely the development of cache optimization mechanisms.

- Lack of connectivity. Communication problems between the two sites were detected, reinforcing the need for reliable communication.

Table 1 shows the number of events that happened during both stages of the deployment process. It is possible to verify 16 failures that were solved automatically with the recovery scripts in the deployment and testing stage. Moreover, the system improves detection of failures, faster than previous times. If the system was broken, it took a long time to detect compared to the same situation using the probes.

Management Traffic Overhead

Our system performs active monitoring that could possibly degrade the available bandwidth of the network. Therefore, it is relevant to measure its impact on the network. We defined four different monitoring levels to test what should be an appropriate monitoring level for this specific regional PACS, without significant overhead for the PACS archive and DICOM Cloud Routers. These categories are based on the following commands:

- (1) C-ECHO
- (2) C-ECHO and C-FIND
- (3) C-ECHO, C-FIND and C-MOVE.
- (4) C-ECHO, C-STORE, C-FIND and C-MOVE.

The PACS archive provides C-Store and C-Move in different ports, thus both services were monitored separately. In order to analyze the impact of the monitoring processes on the network resources, we measure the in/out traffic flows. In Fig. 5, one may observe the overhead produced by the monitoring system compared to the imaging traffic during a 24-h average period, using benchmarking level 2. These results confirm that the control flow has no significant impact on the overall traffic. Furthermore, we have also verified that the PACS performance was not affected by the introduction of a management probe.

Table 2 shows a comparison between the different monitoring levels in the configured probes. Level 1 has no great impact on the traffic network. However, it does not

Fig. 5 Average daily distribution traffic: medical imaging and monitoring traffic

perform a deep test of the PACS, for instance, it may answer to C-ECHO, but the remote service may not work properly. On the other hand, level 2 is able to test the search status over the PACS database and it does not add considerable traffic overhead. Nevertheless, a critical issue is also retrieving medical images, i.e., access the exams. For instance, when reviewing from an external workstation the efficient retrieval of studies is critical. Thus, at least level 3 verification is strongly recommended for this scenario. Level 4 tests all services, but also generates a significant amount of traffic. We believe that level 3 is a good approach, because we are able to test the query and retrieve services, which are the most critical services for physician's search and access. Moreover, even within levels, it is possible to tune the quantity of information that we intend and the administrator may choose the exams to retrieve. This will influence the amount of traffic generated daily.

Finally, the PACS administrator may access the aggregator Web frontend to verify the availability of the PACS archive and DICOM Cloud Routes. The users are able to verify the availability of PACS and workstations in the aggregator interface. Thus, for instance, if a PACS archive is down, the vendor may be automatically notified about this issue. Moreover, in this interface, they support new registers of DICOM services to monitor.

Table 1 Number of detected failures in the deployment and testing stage

Description	Events
Number of failures	26
Number of failures reported by the probes	23
Number of failures solved by the probes	16

Table 2 Comparative traffic of different testing levels

Description	Level 1	Level 2	Level 3	Level 4
Number of C-ECHO/h	720	720	240	240
Number of C-FIND/h	0	720	240	240
Number of C-MOVE/h	0	0	60	240
Number of C-STORE/h	0	0	0	240
Traffic KB/h	633	2,472	268,012	2,086,762

Conclusions

Practical reasons drove us to develop this managing system and proved to be a valuable ally while deploying geo-distributed PACS. Although it was designed for the deploying phase, it still endures as a monitoring/managing tool of the deployed PACS, preventing and solving eventual failures. It provides a framework, as a service, that allows combining extensive test batteries to ensure that the managed device is ready for operation or, in runtime, following the quality of service provided.

To setup a new service, third-party PACS administrators just need to create an account in our portal and deploy the probes in the target machines, which is more convenient than other approaches involving acquisition and installation of dedicated infrastructures. Additionally, it is possible to associate reactive actions to specific test results. For instance, if a service stops, it is possible to restart it. Our system does not predict all the malfunction profiles; nevertheless it is extensible, enabling the administrator to quickly execute his/her own recovery routines. Moreover, every action and network event is recorded, supporting the statistics module. Finally, the system provides a Web control panel where several operational indicators may be presented. Those indicators may be linked to the alert mechanism that supports email messages. Our system was tested in a real scenario and this article presents the results of this use case.

Acknowledgments This work was partially funded by FCT – Fundação para a Ciência e a Tecnologia, under the grant agreement SFRH/BD/79389/2011. This work also has received support from the EU/EFPIA Innovative Medicines Initiative Joint Undertaking (EMIF grant no. 115372).

References

1. P. Seifert, "IHE Radiology Technical Framework Supplement XDS-Ib Integration Profile," 2009.
2. H. K. Huang, *PACS and imaging informatics: basic principles and applications* Wiley-Blackwell, 2010.
3. B. J. Liu, H. K. Huang, F. Cao, M. Z. Zhou, J. Zhang, and G. Mogel, "Informatics in Radiology (infoRAD): A Complete Continuous-Availability PACS Archive Server1," *Radiographics*, vol. 24, pp. 1203–1209, July 1, 2004 2004.
4. Erberich SG, Silverstein JC, Chervenak A, Schuler R, Nelson MD, Kesselman C: Globus MEDICUS - federation of DICOM medical imaging devices into healthcare Grids. *Stud Health Technol Inform* 126:269–78, 2007
5. Ribeiro LS, Costa C, Oliveira JL: Clustering of distinct PACS archives using a cooperative peer-to-peer network. *Comput Methods Programs Biomed* 108:1002–1011, 2012
6. L. S. Ribeiro, L. Bastião, C. Costa, and J. L. Oliveira, "Email-P2P Gateway to Distributed Medical Imaging Repositories," in *HealthInf*, 2010, pp. 310–315.
7. F. Cao, B. J. Liu, H. K. Huang, M. Z. Zhou, J. Zhang, X. C. Zhang, and G. T. Mogel, "Fault-tolerant PACS server," pp. 316–325, 2002.
8. O. Dostal, M. Javornik, K. Slavicek, and P. Ventruha, "Integration of Telemedicine Activities in the Czech Republic," in *Innovations in Information Technology, 2007. IIT 07. 4th International Conference on*, 2007, pp. 611–615.
9. T. Treichel, M. Gessat, T. Prietzel, and O. Burgert, "DICOM for Implantations—Overview and Application," *Journal of Digital Imaging*, vol. 25, pp. 352–358, 2012/06/01 2012.
10. H. Oosterwijk and P. Gihring, *DICOM Basic Version 3*: OTech, Incorporated, 2005.
11. O. S. Panykh, *Digital Imaging and Communications in Medicine (DICOM): A Practical Introduction and Survival Guide*: Springer-Verlag, 2008.
12. N. E. M. Association, "Digital Imaging and Communications in Medicine (DICOM) Part 18: Web Access to DICOM Persistent Objects (WADO)" 2004.
13. IETF, "RFC 1157: A simple network management protocol (SNMP)," ed: The Internet Engineering Task Force, 1990, pp. 1–35.
14. B. J. Liu, H. K. Huang, F. Cao, L. Documet, and D. A. Sarti, "Fault-tolerant back-up archive using an ASP model for disaster recovery," pp. 89–95, 2002.
15. G. Weisser, S. Ruggiero, A. Runa, C. Düber, W. Neff, and M. Walz, "Online Availability Check of Teleradiology Components," *Journal of Digital Imaging*, vol. 20, pp. 393–401, 2007/12/01 2007.
16. C. Toland, C. Meenan, M. Warnock, and P. Nagy, "Proactively Monitoring Departmental Clinical IT Systems with an Open Source Availability System," *Journal of Digital Imaging*, vol. 20, pp. 119–124, 2007/11/01 2007.
17. G. Potter, R. Busbridge, M. Toland, and P. Nagy, "Mastering DICOM with DVTk," *Journal of Digital Imaging*, vol. 20, pp. 47–62, 2007/11/01 2007.
18. Benjamin M, Aradi Y, Shreiber R: From shared data to sharing workflow: merging PACS and teleradiology. *Eur J Radiol* 73:3–9, 2010
19. M. Warnock, C. Toland, D. Evans, B. Wallace, and P. Nagy, "Benefits of Using the DCM4CHE DICOM Archive," *Journal of Digital Imaging*, vol. 20, pp. 125–129, 2007/11/01 2007.
20. R. A. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*: Addison-Wesley Longman Publishing Co., Inc., 1999.
21. G. G. Chowdhury, *Introduction to Modern Information Retrieval*: Facet, 2010.
22. Costa C, Ferreira C, Bastião L, Ribeiro L, Silva A, Oliveira JL: Dicoogle - an Open Source Peer-to-Peer PACS. *J Digit Imaging* 1: 1–9, 2010
23. L. A. B. Silva, C. Costa, and J. L. Oliveira, "DICOM relay over the cloud," *International Journal of Computer Assisted Radiology and Surgery*, pp. 1–11, 2012.