Message Communication Server for Medical Information Systems' Interoperability

Lamprini Kolovou and Dimitrios Lymberopoulos, *Member, IEEE*

***Abstract*— The implementation of a message communication server (MCS) for facing the syntactic interoperability problem in medical communications is the main target of this work. The background of this implementation was a set of recommendations for the types of exchanged messages, general architecture and end-to-end users services, which have been introduced by the authors in [16]. MCS's architecture facilitates different medical information systems to visibly handle real world events and information, without any interference in the basic structure of these systems. The performance of MCS has been tested in three distinct phases employing the application and data store entities of three cooperating medical information systems, a Laboratory Information System (LIS), an Advanced Radiology Information System (RIS) and a Picture Archiving and Communication System (PACS). The evaluation of the developed MCS was also performed and the extracted statistics validated the high degree of MCS's effectiveness and efficiency.**

I. Introduction

Healthcare enterprises are multidisciplinary and heterogeneous domains in nature [7]. They are combined by distinct subunits; each of them is working with different methods of information processing and overlapping procedures [7]. Nowadays, each subunit comprises an individual electronic domain that is constituted by an information system and/or various specialized medical applications; all named Medical Information Systems (MISs). The MISs provide automated workflow and added value services, increasing the quality of healthcare services within the enterprise.

The performance of these as a whole is determined by the effectiveness of coordination and interaction of all MISs, which are implemented independently by diverse vendors [6]. Each such implementation uses different standards and information formats that arise various and difficult interoperability problems [8].

Hitherto, in the context of healthcare enterprises the interoperability is conventionally considered as functional, syntactic and semantic [1]. The functional interoperability is achieved by using the new generation of converged telecommunication networks and technologies. The semantic interoperability is met with terminology standards, while the syntactic with data communication standards.

The syntactic miss-interoperability is due to three factors: a) the lack of a unique interchange format, b) the use of various, overlapping and incompatible data communication standards and c) the different implementations of these standards [4]. Nevertheless, these factors are fundamentals for the implementation of viable MISs within an open healthcare enterprise environment.

The current proposed solutions for syntactic interoperability follow the "specific implementations" approach. These solutions include communication servers that follow the middleware approach, support specific messaging standards and set them as precondition to the connected MISs and include additional translation functions to support the communication of heterogeneous MISs [5], [9], [10].

This paper proposes a message communication server for integrating the different data communication standards and providing a common messaging service to various MISs, without interfering in their own implementations. The presented architecture distributes message handling functions among a central communication server and dedicated interfacing functional entities for each MIS. This architecture is based on a Reference Implementation Model (RImM) that sets the design considerations of Section II. Section III describes the proposed architecture in detail, which's special mechanisms and development are presented in Sections IV and V respectively. The demonstration and evaluation of MCS are analytically described in Section VI.

II. Design Considerations

MCS's design is in accordance with the specifications of a RImM for such communication servers. RImM has been introduced by the authors in [16] and the basic principles of it are the following.

MCS's users are considered as the applications and the data stores of different medical information systems that are intended to exchange information, independently the messaging protocol that they use. The specification of users' profile, the administration of sessions and the application of information security policies are supported by diverse information classes and by their conceptual relations. These classes and their relations comprise the "users' catalogue".

Manuscript received Jule 5, 2008.

L. Kolovou is with the Electrical and Computer Engineering Department, University of Patras, Patras, Greece (phone: +30-2610-996837; fax: +30-2610-996185; e-mail: lamprinik@ wcl.ee.upatras.gr).

D. Lymberopoulos is with the Electrical and Computer Engineering Department, University of Patras, Patras, Greece (e-mail: dlympero@wcl.ee.upatras.gr).

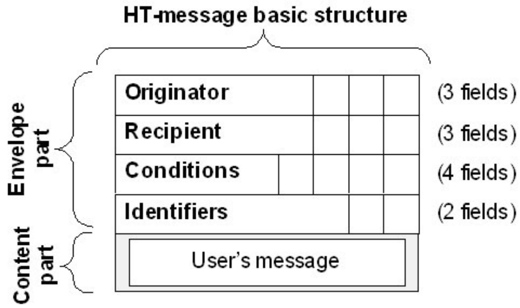

Fig. 1. The structure of HT-message

The general architecture of MCS is also specified by RImM and consists of the Message Transfer System (MTS) and dedicated interfacing entities for the users. The MTS is the core unit of MCS serving messages created by all users and supporting translating functions as well.

The collection of all MCS functional entities and its users, in common with their appropriate operational settings, constitutes a management domain. The administration of this domain constitutes the "*Messages Administration Management Domain (MAMD)*".

During the session, the end-to-end user communication process is organized by a special service, named '*Message Transfer (MT) service*', which is also specified by RImM. MT-service specifies the distinct interaction processes be performed among the entities of the interfaces and the MTS that are involved during all phases of end-users' sessions. At any time instance, each user may be the originator or the recipient of a message. According to the needs of the session, any user may alternate its role.

RImM specifies the common structure of the exchanged messages as well. These messages, thereafter called "*health telematic messages*" (HT-messages) consists of two parts (Fig. 1). The envelope part that constitutes four segments of fields and the content part that includes the initial user's message. The envelope part is used to provide the MT-service consistently and its structure is dependent on the telecommunication protocol that the user applies in the application layer.

III. Architecture of MCS

The MCS has the structure that is depicted in Fig 2. The entities of MCS are of three types: (a) agents that are used for message processing ((1), (10) and (13)) and include proper modules, (b) message stores that are used for message queuing ((2), (3), (6), (7), (8), (11) and (12)) and (c) filters that are used for error checking and message recover functions ((4), (5), (9) and (12)) (§IV.B).

MTS performs storing, transferring and reformatting processes of HT-messages using four entities; the 'Message Transfer Agent (MTA)' (10), the two 'Inbound Message Stores (IMS)' –(6) for application-to-application sessions and (7) for application-to-data store session– and the 'Outbound Message Store (OMS)' (8). Additionally, MTS includes two message filters, (5) for inbound HT-messages and (9) for outbound HT-messages.

MCS's users access the MTS through individual and dedicated interfaces. Application's interfacing is performed by a 'User Agent (UA)' (1), two user's data temporarily storage entities, the 'User Inbound Message Store (UIMS)' (3) and the 'User Outbound Message Store (UOMS)' (2) and a message filter (4). Data store's interfacing is performed by the 'Data Store Agent (DSA)' (13), two message data stores –(11) for inbound HT-messages and (12) for outbound HT-messages- and a special entity called Access Query Library (AQL).

AQL provides extra security mechanism protecting data store's hosted information and controlling the access to this information, with predefined queries that specify all the possible and allowed interactions that the parent data store might perform.

MTS roots the HT-messages "*onward towards to the intended recipients*" accessing and reading the information of the envelope part of HT-messages and also managing the format of the content part of them, conditionally. That means that if the sender and the receiver of the HT-message use the same messaging standard, being implemented with the same way, then MTS transmits the HT-messages only reading the fields of the envelope of HT-message. If the sender and the receiver use different messaging standards or the same with different implementations, then MTS "translates" the initial message and transmit it, based on the fields of the envelope (header) of HT-message (§IV.A: Translation mechanism).

Application's interface constructs the HT-messages, adding the envelope to the initial message and interacts with the MTS, in order to submit or receive HT-messages while communicating with applications and data stores of other MISs.

While communicating with applications of other MISs, data store interacts with MTS in order to decompose its inbound HT-messages and to isolate the content part of them. In the opposite direction, it includes the returned message or notifications of data store, into the content part of new (responding) HT-messages.

IV. Special Mechanisms for HT-Message Handling

A. Translation

Translation is one of the core functions of MCS. The entities of each sender's domain use the content part (payload) of the HT-messages as the carrier of its initially originated messages. If two users communicate applying the same messaging standard, MTA has not any affect on payload of HT-messages, in the transmission process. If users communicate applying incompatible standards, MTA translates properly the payload of HT-messages from the standard format of sender to this of recipient, activating its "Translating" module.

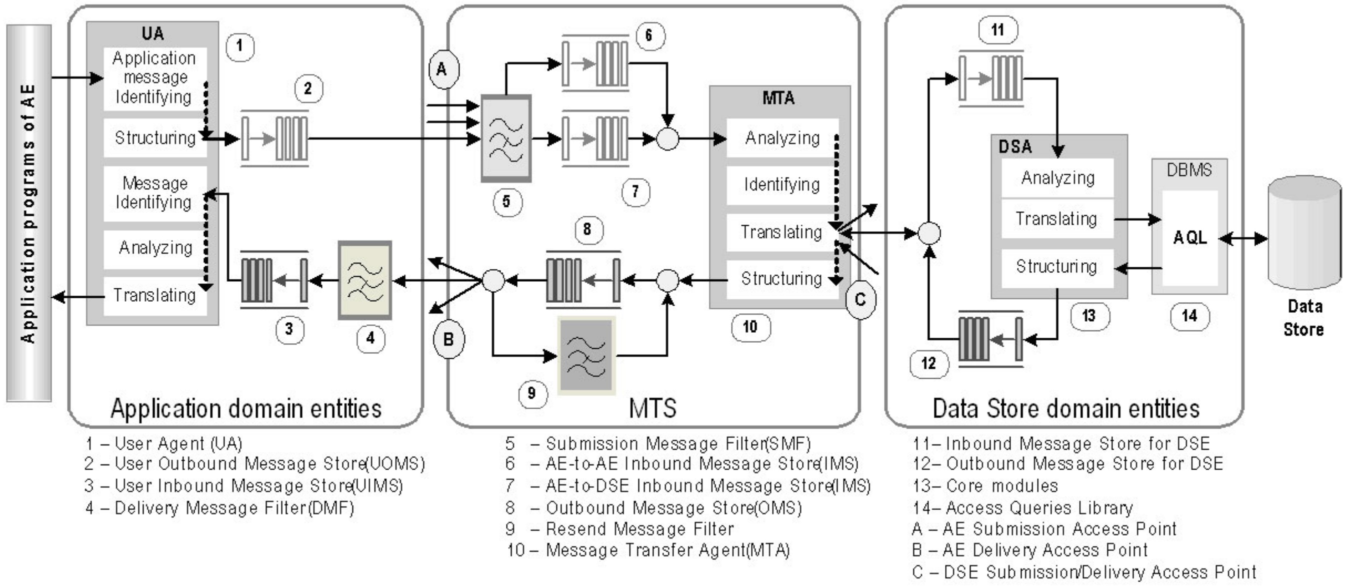

Fig. 2. The entities of MCS's architecture

Translation from this module is performed by three units [15], as depicted in Fig. 3. The first is "*Users' Catalogue*" that hosts -in users' profiles- information about the special conditions that each user applies for the structure of the messages that they send and receive. The second is the "Meta-data store" that hosts information about relations and correspondence for the general rules of message coding and structure that the different messaging standards of users apply. The third is "*Translator*" that re-constructs the content part of HT-messages using the information of the two previous units as following described.

Translator receives the originated messages. According to the information in the envelope of HT-messages, it 'decides' for the translation or not of the payload of them. In the case of translation, it retrieves the special conditions that the users of the session apply from the Users' Catalogue and the general rules for message structures of both messaging standards, from Meta-data store. This information determines the 'final' structure of the payload of the transferred HT-message. The 'new' payload is then forwarded to the next module, in order to construct the HT-message and transfer it to its recipient.

When a new user is added to MCS further configuration of Users' Catalogue and Meta-data store is necessary in order to perform translation properly.

B. *Error handling*

Further to error handling that the utilized telecommunication protocols enforce, MCS provides extra mechanism for transmission failures detection and HT-messages recovery. This mechanism operates at the application level of the used telecommunication infrastructure and includes: (a) supervision of the sessions' performance with the use of special fields of the HT-messages' envelope, that include session and messages identification; (b) check of system resources before starting every session from the interfacing entities of users; (c) administration of queues in message stores with priority in that messages that are signaled with special flags in their envelope; (d) use of the filters for: correct routing of the messages, feedback in the case of failed forwarding from MTA to users and temporal storage of HT-messages until the sessions regular termination. In special cases of system's failures, MCS recovers its last state. The HT-messages that were "in-transfer" and not queued are lost and the corresponding sessions are stopped.

The HT-messages that arrive simultaneously in the message stores are retrieved with priority. First are these that "belong" to sessions in progress and then the initially originated HT-messages. If the HT-messages are of the same case, these are retrieved with a random order. In any case of failure, MCS sends special notifications to the originators.

C. *Auditing*

For auditing the whole operation of MCS, a Watch-Point mechanism had been implemented and embedded in every developed component of MCS.This mechanism is in function continuously and it produces log files including tracing information of the sequential processes that are recorded. For each process the exact time details (timestamps) are marked so the extraction of the willing results is possible. In the form of Fig. 4 the Watch-Point information is presented in a special window for the surveillance from the user-administrator of MTS.

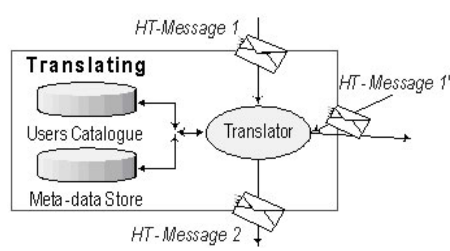

Fig. 3. Translating module's functioning

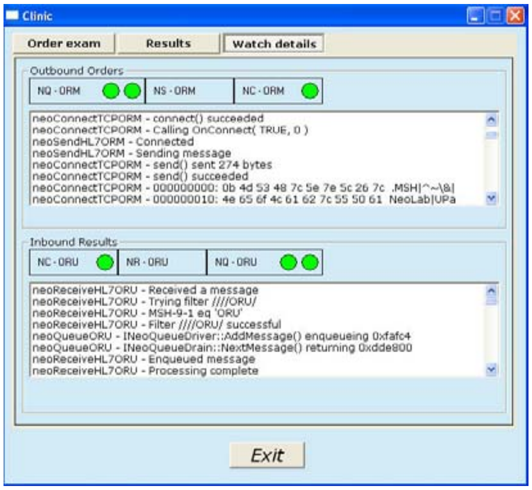

Fig. 4. Graphical user interface for Watch-Point mechanism

The auditing information that is recorded to the log files is divided in three main categories. The first includes the details of the connections between the different entities of MCS and between the various computational systems, where these entities are hosted. In the second category the details of sending an HT-message are transcribed. The fields of this section of the log file describe the start and end of HT-messages' transfer, the successful or failure notifications, the re-sending procedure's details, filtering results and the identification data of HT-message/session. The third category audits the receive process of HT-messages including start and termination time points, filtering results, types of notifications and identification data of HT-message/session.

V. MCS Development

MCS is developed in form of executable modules. These of MTS are installed in an autonomous computational system. These of users' interfacing are embedded to the application programs of them and are installed in the users' terminals.

MTS is developed via three modules: *Send_Receive*, *Resul* and *Translate*. The two first provide a graphical user interface as to watch the MT-service processes.

Send_Receive_Module: (a) includes all the necessary programming that is required to maintain the connections on which the sessions rely on, (b) it performs the functions of MTA and message stores of MTS and (c) performs the audit mechanism for the MTS. The components that have been used for the development of this module are "dynamic" items and their graphical representations define their status (active or not).

Result_Module performs (a) the administration of queues in message stores, using the First-In-First-Out (FIFO) mechanism and (b) the error checking functions for MTS.

Translate_Module performs all the functioning of "*Translating*" mechanism as it is described in §IV.A. This module is activated occasionally, while Result_Module and Send_Receive_Module are permanently active.

The domain entities of users have been developed as these of MTS. For each user, a separated module has been created and installed within the computational system, where application or data store is hosted. AQL of each data store has been implemented utilizing the Database Management System of data store.

The Users' Catalogue and the Meta-data Store have been implemented as independent relational databases that are hosted to the same database server.

For the development of users' interfacing and MTS's modules Java language and Visual Basic 6.0 tools were utilized. HL7 messaging procedures were developed by means of the NeoTool Library, which provides components importable to Java classes and Visual Basic functions. Databases were hosted by means of Oracle9i RDBMS and for access to the AQLs the ADO technology and SQL DML language were applied. DICOM services and message definitions were implemented using the CTN open source software (Washington University of Saint Louis) [11].

VI. Demonstration & Evaluation of MCS

A. Demonstration

The operational behavior of the implemented MCS was studied in three successive steps, presented in [12], [13], [14] and [15] and are following described.

The first application version was created, in order to testify the functionality of MCS. For this reason, there was not used different MISs, but a LIS that comprises several similar applications and a data store. LIS's applications supported practitioners of all hospital's clinics to retrieve results of patients' laboratory exams from the data store of LIS by sending a simple request. This request was based on the patient's identifications data and on the features of the exam. Thus, MCS was employed to support only application-to-data store sessions. The application messages follow the specifications of standard HL7 version 2.3 and their types were: ORU, ORM, classic acknowledgements and specially structured notifications [2].

In the second application version, MCS served the communication among the above LIS entities, as well as the communication among the applications, the data store of a RIS and the data store of a HIS. MCS was employed to support the practitioners of hospital's clinics and radiology department to execute: a) the patients' visits administration (*ordering function*) and b) the programming of requested exams (*scheduling function*). The application messages follow the specifications of standard HL7 version 2.3 and are application requests and response messages (trigger events: S01-S04, S05-S08, S12-S15, S17-S19 and S22-S26, EQQ, SPQ, EDR and TBR) and enhanced mode queries [2].

In the third version the full capabilities of the implemented MCS have been tested. Additional to the entities of the previous versions, MCS served the communication among RIS entities with expanded functionality and PACS entities. RIS expanded functionality includes: the medical reports production (*transcription function*) and the distribution of diagnostic files to corresponding placers of referring documents (*allocation function*). For transcription, the co-ordination of RIS and PACS is required, because the “real data” of examinations are hosted in PACS.

The format of the exchanged application messages differs, according to the MIS they are originated by. The applications and the data store of RIS use standard HL7 v2.3; the applications and the data store of PACS use standard DICOM. HL7 messages are types of document management messages (T01-T04, T07, T08, T10 and T11 trigger events [13]), while the DICOM services that are used are *query* (FIND, GET), *patient management*, *study-management* and *results reporting*.

B. Evaluation

Each application version was evaluated using the Audit mechanism (§IV.C) and for the time period, which the next version lasted. Thus, version 1 (MCS of phase 1) has been evaluated for two months, version 2 (MCS of phase 2) for one month and version 3 (MCS of phase 3) for two months. For the evaluation of versions 1 and 2, a sufficient number of applications and the developed data stores were installed in different computational systems and servers and we created such conditions that approached these of a hospital with low-to-medium traffic and we became the evaluators of these versions. In version 3, an already developed PACS, which was tested in real-world conditions in Patras University Hospital in Greece and it is presented in [17], was used.

The information that was recorded to the log files enabled the estimation of a number of evaluation parameters that are shown in Table I and are of three categories: (a) characteristic times and delays in MCS; these parameters provide the queuing time periods of HT-messages and the delays that are caused by transferring failures and message processing, (b) number of successful and failed transfers of HT-messages and complete and incomplete sessions between participating users and (c) statistical about the sessions and transfers of HT-messages; statistical show a significant improvement from version 1 to versions 2 and 3 at a level of 4-5% about. These parameters were the base for the valuation of quality factors for the developed MCS, according to the ISO 9126 model’s characteristics [3] as it is following described.

1) Quality factors

The quality factors use the characteristics of the ISO 9126 evaluation model. From these characteristics *functionality*, *reliability*, *efficiency*, *maintainability* and *portability* have been estimated for MCS.

Functionality, reliability and efficiency were estimated using the evaluation parameters of Table I (Sets 3, 1 and 2 respectively) and they are depicted in the diagram of Fig. 5.

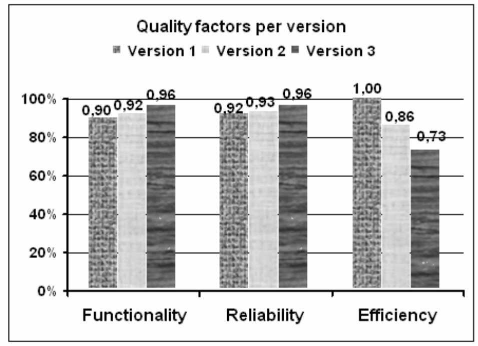

Fig. 5. Graphic representation of quality factors

TABLE I
STATISTICAL FOR EACH MCS’S APPLICATION VERSION

Evaluation parameter	v.1	v.2	v.3
Set 1: MT service parameters (HT-messages)			
Sent	175	70	616
Delivered and acknowledged	162	65	592
Delivered but not acknowledged	4	3	11
Not delivered	9	2	13
Rejected by filters	4	0	7
Resent and delivered	46	18	124
Resent but not delivered	2	3	11
UAp-Uap attempted sessions	0	16	125
Uap-DSU attempted sessions	82	22	220
Uap-Uap complete sessions	0	13	113
Uap-DSU complete sessions	74	18	212
Set 2: Times and delays (sec)			
Memory queuing time	2	2	3
Resending delay	2,5	2,3	2,7
Processing delay in AMIs	4	4	4
MTA processing delay (appl.-data store)	8	9	12
MTA processing delay (appl.-appl.)	3	4	6
Total transferring time	14	16	24
Total completing session time	30	35	48
Set 3: Statistical (%)			
Complete appl.-appl. sessions		81,25	90,40
Complete appl.-data store sessions	90,24	81,82	96,36
Complete transfers of messages	92,57	92,86	96,10

Efficiency seems to be the cost for the high percentages of functionality and reliability, although the real measurements of the delays stand to tolerable levels (Table I). For this factor, to express the level through percentages, version 1 is used as the meter of comparison (100%). In total, although the complexity of the MCS, the number of concurrent session and the traffic of HT-messages are increased from version 1 to 3, MCS seems to respond at a higher level.

Maintainability and portability regard the difficulty in adjusting the system under new operational requirements. These two factors are answered by the methods and standards that were used to develop and evaluate the MCS. For maintainability, the evaluation mechanism enables the surveillance of the processes that occur and any fault can be easily diagnosed. The application programs that execute these processes are totally independent and any change of them can be handily done and do not influence the function of the whole system. For portability, the tools and the standards that were used to implement the MCS supply the easy installation of MTS to different environments and compatibility with already used software solutions. The only issue that is put forward concerns the technical features of the used hardware as HT-messages' processing requires high computational power.

The delays in MCS could be even lower than the measured ones, if an alternative designing solution be applied for handling the traffic of HT-messages. In this system, more message stores could be added or multiple MTAs that would work in parallel and serve more concurrent sessions.

VII. Discussion

For the proof-of-concept the MCS demonstration and evaluation were performed with the use of a small number of MISs (LIS, RIS, PACS, HIS). While designing the presented architecture, the special conditions that a large-scaled healthcare organization set were not taken into account. In that case, the traffic of the messages, which are multiple than that were used in demonstration, the number of the related MISs and the telecommunication and computing infrastructure of the healthcare organization should be counted in. These conditions would led to an enhanced architecture, regarding the number of message stores, agents and filters, as to optimize the distribution of messages processing load, following the same design considerations and the basic architecture of MCS.

All the results and conclusions described in the previous section will be used as feedback for a new and more effective MCS and for the improvement of RImM. More precisely, as future work the followings are scheduled: implementation and testing of MCS in real world conditions with more MISs and different messaging standards in large-scaled hospitals; incorporation of 'already developed' EHR systems and other MISs; development of MCS with new and advanced technologies (XML, web services, etc.).

VIII. Conclusions

Nowadays, the healthcare information domain is oriented to distributed systems manipulating, with a uniform and unbounded manner, information been created by medical applications with different specialties and capabilities. The interoperability within this domain is the main issue. The MCS, demonstrated in this paper, is an effort to overcome this problem, focusing on the syntactic interoperability. While most solutions face interoperability from the technical point of view, MCS contributes to the standardization approach, applying the specifications of a reference model, called RImM, introduced by the authors in [16]. This approach can lead to more viable solutions, while no constraint regarding the used messaging standards or technologies is set.

The presented architecture distributes the message handling processes to a stand-alone middleware system, MTS, and to dedicated interfacing sub-systems that are allocated to users' sides; better administration, maintenance and surveillance of the provided MT-service is achieved.

In comparison with other communication servers that have been proposed or used for the same purposes, MCS provides high degree of extensibility and special mechanisms for failure detecting, error handling and messages recovery, supporting the performed services in a reliable, operational and effective manner.

IX. References

[1] CEN/ISSS e-Health Standardization Focus Group, "Current and future standardization issues in the e-Health domain: Achieving interoperability", CEN 2005

[2] M. Shafarman, M.Tucker, Health Level 7, version 2.3, 1997, http://www.hl7.org

[3] ISO/IEC 9126, International Standard, Information Technology, 1991, http://www.usabilitynet.org/tools/r_international.htm

[4] H. Lu, Y. Chu, R. Sheu, W. Lo, "A generic application sharing architecture based on message-oriented middleware platform", Computer Standards and Interfaces, 30 (2008), pp. 191-199

[5] V. Bicer, G. B. Laleci, A. Dogac, Y. Kabak, 'Artemis message exchange framework: Semantic interoperability of exchanged message in the healthcare domain', SIGMOD Record, Vol. 34, September 2005, http://www.srdc.metu.edu.tr/

[6] A. Berler, S. Pavlopoulos, D. Koutsouris, "Design of an interoperability framework in a regional healthcare system", 26th Annual International Conference of the ISSS, EMBS, pp. 3093-3096, San Francisco, CA, USA, September, 2004

[7] Vargas, B.; Pradeep Ray, "Interoperability of hospital information systems: a case study", Proceedings of 5th International Workshop on Enterprise Networking and Computing in Healthcare Industry, 6-7 June 2003 Page(s): 79 - 85

[8] Kitsiou, S., Manthou, V., Vlachopoulou, M. "A Framework for the Evaluation of Integration Technology Approaches in Healthcare", Proceedings of the IEEE International Special Topic Conference on Information Technology in Biomedicine, October 26-28, 2006, Ioannina, Greece

[9] "Interface engines for healthcare: 21st century trends", McLEOD, March, 2006

[10] S. H. Hsieh, S. L. Hsieh, etc, "Middleware based inpatient healthcare information system", Bioinformatics and Bioengineering, 2007. BIBE 2007. Proceedings of the 7th IEEE International Conference on, 14-17 Oct. 2007 Page(s):1230 – 1234

[11] National Electrical Manufacturers Association, Digital Imaging and Communication, PS 3.4-2003, 2003, http://medical.nema.org

[12] Lamprini Kolovou, Maria Vatousi, Dimitrios Lymperopoulos, Mixalis Koukias, "Advanced Radiology Information System", 27th Annual conference of the IEEE Engineering in Medicine and Biology Society (EMBS), Shangai, China, September 1-4, 2005

[13] L. Kolovou, G. Koutelakis, G. Mandellos, A. Darras, D. Lymperopoulos, 'Innovative cooperative scheme of novel RIS and PACS', The IASTED International Conference on BIOMEDICAL ENGINEERING-BioMED 2005, Innsbruck, Austria, February, 2005

[14] L. Kolovou, A. Darras, D. Lymperopoulos, "HL7 Message Handling for Intra-hospital Information Systems", 2nd International Conference on Computational Intelligence in Medicine and Healthcare, 29 June – 1, Costa da Caparica, Lisbon, Portugal, July, 2005

[15] L. Kolovou, P. Galanopoulos, D. Lymberopoulos, 'MRIMS: A new Middleware Connective Level for RIS and PACS', WSEAS TRANS. on INFORMATION SCIENCE & APPLICATIONS, Issue 1, Vol 1, July 2004, pp 575-580

[16] L. Kolovou, E. Karavatselou, D. Lymberopoulos, 'Reference Implementation Model for Medical Information Systems' Interoperability', 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, August 20-24, 2008

[17] G. Koutelakis, G. Mandellos, G. Triantafyllou, "A new PACS Viewer Eliminates an Archiving Model Problem of the DICOM Standard", Fifth International Symposium on Communication Systems, Networks And Digital Signal Processing, Patras, Greece, July 19-21