Secured Hierarchically Dependent Distributed Database Model Applied to Hospitals Information System (HIS)

Sherif M. Badr, PhD.
College of Computer Science,
Modern Academy, Cairo, Egypt

ABSTRACT

The quality of Electronic Patient Record (EPR) became a critical factor in the decision making process, with a corresponding need for error management through automated data quality assurance practices. Although Picture Archiving and Communication System (PACS) have become a mature technology over the past few years and has been widely implemented in several developed hospital, evaluation of PACS success is still a major problem for healthcare organizations especially in developing countries. For that reason it was the aim of this research to introduce design considerations and solutions regarding the implementation and evaluation of a modern (EPR) within an already operating hospital environment. The research proposed a strategy to smoothly transform an already existing hospital to a fully digital one and to integrate all existing medical equipments to a central archiving system with the ability to upgrade and expand as well as to connect with the other hospitals and health care organizations.

With the development of distributed computing, data security becomes more important in a distributed database. This work introduces a solution of this problem. With the analysis of existing system, we get the required and set up a data model for the complete health system. Finally we suggest algorithms for the proposed model.

Keywords-PACS; distributed health system.

I. INTRODUCTION

Advances in digital technologies, particularly in the fields of computing, imaging and communication, have progressed to the point that it is now possible to acquire medical record and images in digital form, archive them on computer systems, and display them in diagnostic quality. The display monitor used to present the all medical records and images can be in an adjacent or distant location to the original point of acquisition. There can be as well multiple monitors at multiple locations (sites) and as soon as the master record and image file has been archived, only a copy of the data is transmitted for display [1]. Hospital information system (HIS) and radiology information management system (RIS) can be considered the main components of electronic patient record (EPR). Therefore, any attempt to develop a universal EPR will require a better integration of both HIS and RIS. The quality of EPR data maintained on clinical support systems became a critical factor in the decision making process, with a corresponding need for error management through automated data quality assurance practices especially in developing countries which lack health care maturity.

2. RELATED WORK

Maglogiannis et al. [2] demonstrated that Picture Archiving and Communication Systems (PACS) have evolved into a healthcare enterprise-wide system that integrates information media in multiple forms, including voice, text, medical records, waveform images, and video recordings. The integration of these various data types requires the technology of multimedia hardware platforms and secured database for privacy, information systems and databases, communication protocols, display technology, and system interfacing. Maass et al. [3] highlighted as well some healthcare related issues with regard to Information Systems (IS) implementations. They demonstrated that the IS installation in health care requires an adequate and sufficient research team. It is a prerequisite, that informatics, medicine, computer sciences and health economy are combined. While many researchers have tried to investigate the applicability of integrated health systems, Xue et al. [4] have investigated the development of PACS showing the different obstacles that face for improving its healthcare integration. They pointed out that many of the obstacles found are similar to other developing countries and can hinder the health care improvement if not resolved. Ebadollahi et al. [5] presented different challenges and directions that play important roles in the healthcare information system's efficiency. They have also presented the idea of concept-based multimedia health records, which aims at organizing the health records at the information level. They explored the opportunities and possibilities that such an organization will provide, what role the field of multimedia content management could play to materialize this type of health record organization, and what challenges will be encountered in the quest to realize the idea. For allowing for more flexible PACS, Bui et al. [6] have proposed an open source PACS that aim to provide a common foundation upon which not only can a basic PACS be readily implemented, but to also support the evolution of new PACS functionality through the development of novel imaging applications and services. Bortis [7] claimed that Integration engines are a crucial piece of the health care information technology puzzle. He proposed a solution to solve the vast amounts of data and the slew of interchange standards and protocols face the healthcare organizations. On the other hand, Pelikan et al. [8] Claimed that the main aspect of performance behavior is the coordinated design between system and network architecture. They suggested the migration to fast Ethernet switching (for modalities and workstations, when available) and to gigabit Ethernet for connecting the servers.

Pare et al. [9] demonstrated that the evaluation of PACS success is a major challenge to healthcare organizations. Their review of previous PACS research suggested a fragmented and focused evaluation approach, thus offering limited discussion of comprehensive views of PACS success or systematic and practical guidance to its evaluations. Based on two prevalent information systems success models, their research proposed and described an integrated framework for evaluating PACS success in hospital settings.

Pare et al. [9] demonstrated that the evaluation of PACS success is a major challenge to healthcare organizations. Their review of previous PACS research suggested a fragmented and focused evaluation approach, thus offering limited discussion of comprehensive views of PACS success or systematic and practical guidance to its evaluations. Based on two prevalent information systems success models, their research proposed and described an integrated framework for evaluating PACS success in hospital settings.

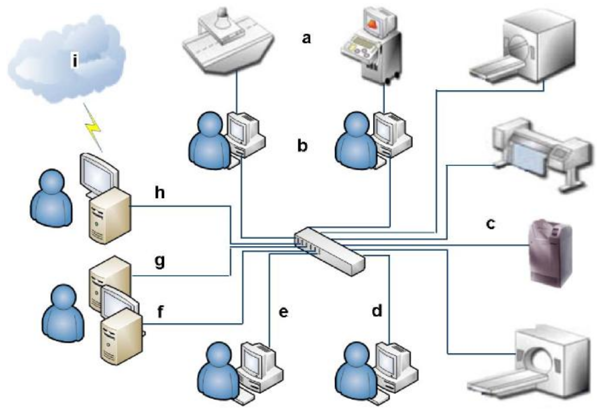

Figure 1. The work flow diagram shows a typical PACS system needs a suitable network topology and network speed to achieve high performance. (a) non-DICOM modalities, (b) gateways, (c) DICOM modalities, (d) reading workstation, (e) review workstation, (f) PACS server, (g) archive server, (h) Web server, and (i) the internet

In a typical radiology department, it is likely that a large number of steps are performed in the sequence of events from the time that the patient is first registered in the department to the time that the clinical report is issued. The required sequence is likely to be a process that has evolved over many years, and it may no longer be the optimal process for a modern radiology department [10]. Installation of a PACS gives the opportunity to re-evaluate the workflow within the radiology department. Rather than merely mimicking an existing, paper-based, system, a carefully planned PACS implementation can encourage an improved workflow, i.e. the more efficient flow of information, images and patients through the department.

The research proposed a strategy to smoothly move the hospital to a fully digital one and to integrate all existing medical equipments to a central archiving system with the ability to upgrade and expand as well as to connect with the other hospitals and health care organizations. The analysis and deployment suggested in this research tackled the critical performance analysis issues required to achieve maximum throughput in the underlying case study.

3. PRPBLEM PRESENTION:

3.1 Distributed Systems:

The distributed system is a collection of independent computers interconnected via point to point communication lines. Each computer, known as a node in the network, has a processing capability, a data storage capability and is capable of operating autonomously in the system. The database is allocated across system nodes in units of distributed relations which allow a general manner of redundancy.

A distributed database is a collection of data which belong logically to the same system but are spread over the sites of a computer network. In other words: a distributed database system is characterized by the distribution of the system components of hardware, control, and data.[11],[12]

The only allocation constraint in that all data must be either locally or globally accessible from any system node. The distributed of data on the network is invisible to the user.

3.2 Location Transparency:

A major objective of distributed database system is to provide what is usually called location transparency, which means that user and user programs should not need to know where any particular item of data is located. Instead, all such location information should be kept by the system as part of its system catalog, and all user requests for data should be interpreted by the system in accordance with that information.

The advantages of such transparency are obvious: it simplifies the logic of application program, and it allows data to be moved from one site to another as usage patterns change without necessitating any reprogramming.

3.3 Replication Transparency:

Data replication means that a given logical data object can have several distinct stored representatives, at several distinct sites.

The Replication transparency: means that all details of locating and maintaining replicas should be handled by the system, not by the user.

The advantages of such an arrangement are that it can provide both improved performance and improved availability: applications can operate on local replicas instead of having to communicate with remote sites, and a given data object remains available for processing so long as at least one replica is available, there are some disadvantages too, updating an object requires the update to be programmed to all copies of that object at all sites, and more storage is required, but the advantages are likely to outweigh the disadvantages.

Location transparency and replication transparency together imply that a distributed system should look like a centralized system to the user.

The user should be able to think purely in terms of logical data objects and should not be concerned with where or how many times those objects are physically stored.

(i.e. the problems of distributed system are internal-level problems not conceptual or external-level problem).[13]

3.4 Problems of Distributed Systems:

The basic problem just outlined gives rise to further problems in several areas, among them the following:

- Query processing.
- Update propagation.
- Concurrency control.
- Commit protocols.
- Catalog management.

This paper will limit the attention to the discussion of the problem of processing queries in a distributed database.[14]

4. THE PROPOSED MODEL:

Many researches have appeared recently on distributed computing systems and distributed database management. Most of the related problems have been defined, thought few experimental corresponding results proposed.

It is clear that the experiments have still to be driven to show how theoretical results can be smoothly merged, for solving under acceptable performance and user-convenience, such problems as: distributed data definition, and moreover, distributed data manipulation, including:

- Concurrent users query analysis.
- Concurrent requests scheduling.

The aim here is to focus on two particular aspects of query management in distributed data sharing systems:

- Query processing over distributed information.
- Update control.

Specific algorithms have been proposed for each, and it will be shown that both types of algorithms contributes in the process of maintaining database consistency and query processing optimization.

4.1 Introduction:

To Maximize the HIS system, a hierarchical distributed hospital system was used.

All the patient information is saved in the patient hospital database and manipulate as work flow between the hospital branches.

For emergency cases the patient can access his private data files from any hospital through the network directly from the patient hospital as read only files, and the new data and status are saved as temporary information till the emergency cases finished, then the treating doctor in the main hospital analysis and diagnoses the patient status and update the patient files.

Accessing data from a remote hospital do not have the permission to delete or edit the stored data.

The main patient hospital called master node and the emergency hospital called slave node.

Figure 2 shows the architecture, which provides dynamically scalable and virtualized resources as a service over the Internet on a utility basis.

4.2 Architecture Modeling Elements:

The word model emphasizes that, the study is not on the object of interest but only on a representation of it. This representation is assumed to have essentially the same behavior as the object itself.

Modeling should refer to the gathering and structuring of data in such a way that the values of the parameters, the initial values of the variables, and their interrelations are formalized. The models may be conceptual, physical, mathematical, or computerized or progressive combination of these.

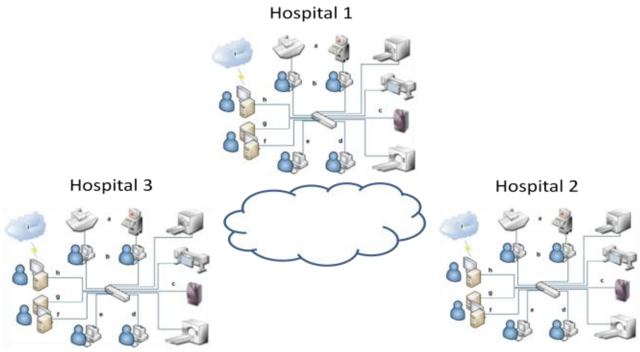

Figure 2 the model architecture

If a variable, a parameter or a relationship is varied from one variant of the model to another, then it is a factor in the sense of the theory of the statistical design of the experiments.

The change of the one or more factors reflects a change in the output. In the experimental design terminology the output is called the response.

4.3 Data Model:

The data model describes the operation of the system in terms of individual events of the individual elements in the system. The interrelationships among the elements are also built into the model. Then the model allows the computing device to capture the effect of the elements actions on each other as a dynamic process. Two methods can be used to set up a distributed database.

In the first method, segments of a file are located at different nodes of the network; a user's request to manipulate a file can be accepted by a local node which has a complete directory about the locations of various segments of the file. Exclusive access to a segment is guaranteed by the owner of that segment, a request will be directed to the node with the segment. If no updating is performed on the element of the segment, the access right is granted to the request, and the element of the segment goes into a locked state. The special requirement of the scheme is that each local node will have a large and complicated directory.

The second method involves multiple copies of the same database located at different nodes of the network. The configuration of a distributed database with multiple identical copies at different nodes requires a control schema to maintain file consistency and multiple updates.

Our architecture coincides with the first method (i.e. the file is segmented and located at different nodes of the network).

The database is distributed through two levels of hierarchical database, which are clusters connected, and these database copies are independent for each site. (i.e. for all database copies at each level, $DB_{ij} \notin DB_{ik}$ where i=1, 2) as shown in figure 3.

In the second: the database for each site is split into two dependent partitions: local and global, the relation between them is as follows: the global partition for site i at second level consists of database elements DE_{gi}, so [X_i:=<DE_{gi}>], and the local partition consists of database elements DE_{1i}, so [Y_i:=<DE_{1i}>].

Then: $X_i = F(Y_i)$

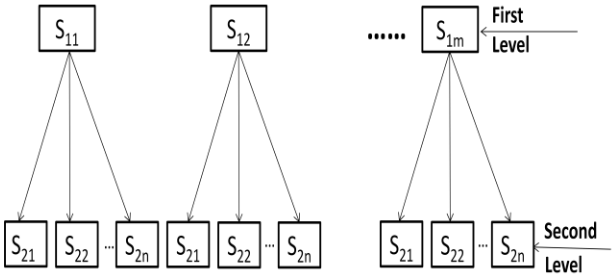

Figure 3 General Data Model

In the first level: the database at each site is a combination of the summary of all global partition of the database in the sites in the second level which are cluster connected with that site.

(i.e. $DB_{1i}= \sum_{j=1}^{n} xj$)

Where n is the number of sites in the second level which are clustery connected with site i in the first level as shown in figure 4.

Note: n is not equal from one site to another.

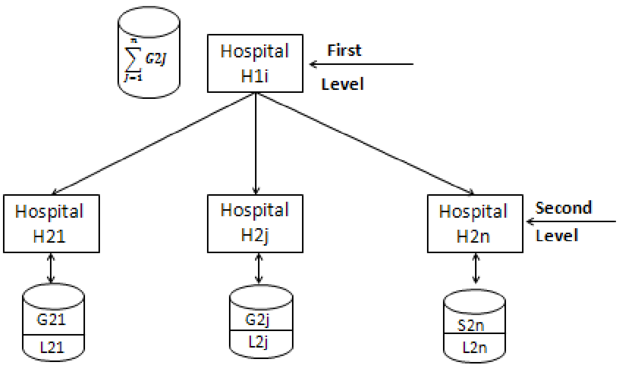

Figure 4 the relation between first and second level

4.4 Query Model:

Query processing in a distributed database corresponds to the translation of requests, formulated in a high level language on one computer of the network, into a sequence of elementary instructions which retrieve data stored in the distributed database.

From the above, the query model of the proposed algorithm will be introduced. Two types of queries are possible: queries applied to the first level of the data model, and others are applied to the second level.

The synchronization is allowed for processing the queries in both levels except if some database elements are locked, which means concurrent queries could be processed simultaneously is not allowed if there is some database elements common and in the locking state. The query can be in one of the following forms:

Arrival	Processed	Data in Consistence
Local	Local	Local
Global	Global	Global
Global	Global	Local
Local	Local	Local

During this work, the first two cases will be taken into consideration.

4.5 Update Transaction Model:

In our architecture the update transaction is applied to the local partitions of the sites at the second level only.

Concurrent transactions are allowed to be processed simultaneously if they are not conflicted, a locking procedure takes place. When a transaction arrives a timestamp is appended to it, and will be executed if all the database elements assigned to it are released. If there is a conflict the transaction waits until its turn comes up. When its turn comes up a locking procedure takes place.

Four types of transactions are allowed to process the database: changing, deletion, insertion or query.

The transaction process in illustrated as follows: it reads values of a set of data elements, performs some computations and writes new values of a subset of these data elements (R, C, and W respectively).

The response time of a transaction is defined as the difference between the finish time and the time when the transaction is launched on its originating site. In other word, it is the sum of the service time and the waiting time in the queues.

4.6 Network Model:

The network is the transport medium between different database copies either within the same traversing different levels.

The network structure or geographical coverage is not on the network itself will be modeled as a delay function. A random variable with certain probability distributed will be used to represent the delay function. The network nature (local, long haul, heavily loaded, weakly loaded, ..) will be mapped through the average value of the random variable representing the delay function.

5. THE PROPOSED ALGORITHM:

As mentioned before, every database copy is segmented and located at different nodes of the network. The new algorithm is presented as two methods (algorithms) for processing the first level query, and their corresponding update transaction methods (algorithms).

The new algorithms allow concurrent access of query or transaction when they are not conflicting, but if there are some conflicts, then they will be serialized, and will be executed in the order of arrival (which is indicated by the appended timestamp) to keep the consistency constraints.

When a query or transaction arrives, a timestamp is appended to it, and a semantic analyzer route them to the corresponding partition, then its algorithm will be executed.

5.1 Processing of first level query:

Two algorithms will be presented in this section. There is no logical judgment to predict which of them works better, the simulation in the next chapter should illustrate the performance criteria for both algorithms and will present the optimum criteria at which every algorithm performs well.

5.1.1 First algorithm "direct transaction query algorithm":

The idea of this algorithm is based on the direct transmission of queries at first level after semantic analysis of the corresponding sites at second level, i.e. each query is semantically analyzed to several sub-queries which are in turn routed to the corresponding sites at second level. When the sites accepting these sub-queries positively acknowledge, the query will be processed, otherwise it will wait.

The Pseudo Code of this algorithm as follows**:**

Step 1: *A query Q arrives at the site i at first level.*

Step2: *Q is semantically analyzed into sub-queries, each sub-queries assigns to a corresponding site at second level.*

Step 3: *The sub-queries directly transmit to the corresponding site at second level.*

Step 4: *A conflict test is performed to find out if there is any locked database element (i.e. in the current process with other transmission) and is required by the sub-queries, if so a negative ACK is transmitted. For the locked database elements, the sub-query will wait until these database elements are released and a positive ACK is transmitted.*

Step 5: *Positive or negative ACK is transmitted to the originating site of first level.*

Step 6: *If all sites at second level were transmitting a positive ACK, then the query is replied.*

If any site at second level transmitted a negative ACK, the query at first level waits for the positive ACK. If there is no reply, either positive or negative from the second level sites then after time T_d the query at sits i at first level will consider it as a positive ACK.

5.1.2 Second algorithm "direct processed query algorithm":

The queries in this algorithm are processed directly if the database elements processed by the query are released.

The Pseudo Code of this algorithm as follows**:**

Step 1: *A query Q has arrived on site I at first level.*

Step 2: *A conflict test is performed, if any database elements are locked and required by Q, then Q waits until these database elements are released. If not it will be replied.*

Step 3: *For the locked database elements at first level, they will be released when the transaction looking these database elements at second level is finished, it will transmit a release order to these database elements at first level.*

5.2 Processing of second level query:

The queries at this level could process the local and /or the global partition. This subject is beyond the scope of this work.

5.3 Processing of updates:

Two updates transaction algorithms are presented to cooperate with the two presented query algorithm.

5.3.1 Update transaction (first algorithm):

This algorithm is coherent with the first query algorithm (direct transmitted algorithm), and its idea is based on the local locking at the database copy at the site of second level only without any locking of the database elements at first level.

The Pseudo Code of this algorithm as follows**:**

Step 1*: An update transaction arrives at site S_{2j} at the second level (for the local partition only).*

Step 2: *A conflict test is performed to detect if this transaction is conflict with any other transaction. If it is conflicting, it will wait until the common database elements are released.*

Step 3: *If it is not conflicting with any current transaction or all database elements required are released, the transaction locks all the database elements used in the second level (in both local and global* partition).

When the transaction finishes, all the locked database elements for that transaction are released.

Note: the query is not processed if it uses some locked database elements (i.e. A negative ACK is replayed for the locked database elements).

5.3.2 Update transaction (second algorithm):

This algorithm is coherent with the second query algorithm (direct process query algorithm), the idea of this algorithm is based on the locking of database elements at both first and second level where the transaction is originated.

The Pseudo Code of this algorithm as follows**:**

Step 1: *An update transaction arrives at site S2j at the second level (for the local partition only).*

Step 2: *A conflicts test is performed, if this transaction conflicts with the current processing transactions, it will wait until its turn comes up, if it is not conflicting or its turn comes up then step 3 takes place.*

Step 3: *A locking is performed for the used database elements at both, first and second level.*

Step 4: *The used database elements in both local and global partition in the second level are locked. And a locking order is transmitted to the corresponding (parent) site of the first level.*

Step 5: *The used database elements at first level are locked.*

Step 6: *The transaction is processed at second level.*

Step 7: *When the transaction finishes the new results are written at both local and global partition at second level, then the used database elements are released. At the same time, these new results are transmitted to the corresponding (parent) site at the first level, with a release order.*

The new results are written at the site of first level, and then the database elements are released

6. CONCLUSION

In this research a systematic scenario for HIS has been discussed. It was important for the team responsible for installing HIS to recognize the importance of using the advanced ways of analysis to carefully plan the system taking into accounts the critical performance issues and the possibility to expand in the future. In addition, a procedure to guarantee system reliability and databases synchronization during maintenance was defined and integrated into the system. The success of the HIS model and the achieved performance could motivate the government to generalize this pilot system to cover the whole country by linking all hospitals together. For that reason, a constant review was carried out, to assess HIS performance, which reflected the benefits of the implemented system. This paper suggests two methods to improve the response time of the query processing in the hierarchically dependent distributed database.

The first method "direct transmission query" which transmits the query directly from one level to another level to check the status of its database elements.

The second method "direct process query" which lock the database elements assigned by the update transaction from the second level. The main future work is to compare between the two suggested methods.

As a further possible continuation of this work is to integrate the data model to have an interconnection between the sites in the first level. Also the sites in the second level may have more than one parent's sits on first level.

7.REFERENCES

[1] G. Kormentzas, I. Maglogiannis, D. Vasis, D. Vergados, and A. Rouskas, "A modeling and simulation framework for compound medical applications in regional healthcare networks," Int. J. Electron. Healthc., vol. 1, April 2005, pp. 427–441.

[2] I. G. Maglogiannis, K. Karpouzis, and M. Wallace, *Image and Signal Processing for Networked E-Health Applications*, Morgan & Claypool, first edition, 2006.

[3] M. C. Maass, and R. Suomi, "Adoption-related Aspects of an Information System in a Health Care Setting," Proceedings of the 37th Annual Hawaii Int. Conf. on System Sciences (HICSS'04) , Jan. 2004, pp. 2-9.

[4] Y. Xue and H. Liang," Understanding PACS Development in Context: The Case of China," IEEE Trans. Inf. Technol. Biomed., vol. 11, Jan 2007, pp. 14–16.

[5] S. Ebadollahi, A. R. Coden, M. A.Tanenblatt, S. Chang, , T. Syeda-Mahmood, and A. Amir, "Concept-based electronic health records: opportunities and challenges," In Proceedings of the 14th Annual ACM international Conference on Multimedia , Santa Barbara, CA, USA, October 23 - 27, 2006.

[6] A. A. Bui, C. Morioka, J. D. Dionisio, D. B. Johnson, U. Sinha, S. Ardekani, R. K. Taira, D. R. Aberle, S. El-Saden, and H. Kangarloo, "openSourcePACS: An extensible infrastructure for medical image management", IEEE Transactions on Information Technology in Biomedicine. Vol. 11, Jan. 2007, pp. 94-109.

[7] G. Bortis, "Experiences with Mirth: an open source health care integration engine," In Proceedings of the 30th international Conference on Software Engineering, Leipzig, Germany, ICSE '08, May 10 - 18, 2008.

[8] E. Pelikan, A. Ganser, E. Kotter, U. Schrader, and U. Timmermann, "Experience with PACS in an ATM/Ethernet switched network environment," IEEE Trans Inf Technol Biomed, vol. 2, March 1998, pp. 26-29.

[9] G. Paré, D. Aubry, L. Lepanto, and C. Sicotte, "Evaluating PACS Success in Hospitals: A Multidimensional Model", 38th Hawaii Int. Conf. on System Sciences (HICSS'05), Jan. 2005, pp. 2-9.

[10] G. Paré, D. Aubry, L. Lepanto, C. Sicotte, "Evaluating PACS Success: A Multidimensional Model," hicss, p. 147c, Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05) - Track 6, 2005.

[11] "Distributed Databases: Principles and Systems", Mcgraw-Hill Computer, Stefano Ceri ,2008

[12] "Distributed systems", sape mullender, second edition, Addison-wesley, 2012

[13] "Distributed Systems: Principles and Paradigms Andrew S. Tanenbaum, Prentice Hall, 2007

[14] "Distributed Systems: Concepts and Design", 5th Edition, George Coulouris,2011