

A Distributed Database System for Glaucoma Monitoring

Mihai L. Mocanu¹, Mihai Dorobanțu^{1,2}, Carmen Mocanu³, Dumitru Burdescu¹

1. Faculty of Automatics, Computers and Electronics, Software Engineering Department, University of Craiova,
2. Health Information Systems Enterprise Architecture Division, Imaging Science and Information Systems, Georgetown University,
3. Dept. and Clinic of Ophthalmology, University of Medicine and Pharmacy Craiova

Summary: This paper describes, from a practitioner's point of view, the concepts, methods and tools involved in the design of a practical and potentially low cost distributed information system, with web-based capabilities, for monitoring glaucoma.

Our experience with existing Hospital Information Systems (HISs) found them unsuitable in the very important monitoring process of patients with glaucoma. Actual Electronic Patient Record (EPR) schemes are more to do with management and appointment simple aspects than with clinical and decision-making processes. In a closer relationship to the specific of the affection, we found that demographic patient databases, usually known as Patient Administration Systems (PASs), have not been designed for being shared or concurrently exploited by different programs or even several replicas of the same program.

Many of the early deficiencies in the process of following-up glaucoma patients by dozens of different ophthalmologists in many independent offices from different clinics (with heterogeneous information recording, not very well managed by the existing office capabilities) could only be solved by specifying, designing and implementing a new EPR scheme in a mixed distributed environment, based on a distributed database as a demographic core (or PAS) of patients with glaucoma. A specialized health record management system, with core functionality in monitoring glaucoma, and core data organized as a distributed database system, has been designed in a bottom-up manner to meet the immediate needs. Its pilot implementation was intentionally kept flexible, taking in account developing standards, to accommodate any anticipated future requirements. Among many other benefits, the new EPR allowed medical doctors (ophthalmologists) to view and modify patient information and records in a safe, flexible and efficient manner. Improvements in all the managerial and decisional aspects (regarding costs and time delays) could also be remarked rapidly.

Keywords: medical databases, Electronic Patient Record (EPR), glaucoma management

1. Introduction

Today, computers are a key factor in bringing technology to the clinical practice. Since the first attempts to introduce computers during the late 80s, Electronic Patient Record (EPR) schemes and Medical Database Management Systems (MDBMS) have evolved. This means that today, in most European countries, the National Health Services are investing large amounts in Information Technology (IT). This brings a unique opportunity for medical doctors (MDs) to use IT more in their clinical practices.

All over the world, Patient Classification Systems (PCSs) are now used for financing, clinical management, planning, budgeting, evaluation and control purposes in hospitals and in other health care services. Back in the 80s, uncontrolled increases in the U.S. Medicare expenses have determined the adoption of a case-based financing system. Diagnostic Related Groups (DRGs) had been developed by a group of researchers at Yale University in the late 60s as a tool to help clinicians and hospitals monitor quality of care and utilization of services. They proved to be so useful that in 1983 they became the only system used by Medicare in the United States to pay hospitals.

Thus, DRG is used now not only to classify different types of patients, but also to ease the implementation of IT management support systems, to create economical stimuli and to address hospital efficiency. DRGs permit to classify the patients based on the diagnosis, the procedures and other information (the complexity of each case) and to link this type of patients that each hospital treats to the expenses needed [1]. Among the necessary data for the patient classification on the basis of the diagnosis and the procedures in DRG categories are: age, sex, hospitalization period, principal diagnosis, secondary diagnosis, procedures, and health condition when leaving hospital. These data define the DRG classification system.

In a growing number of countries a similar system has been implemented. The Romanian Government has been implementing since 2002 the DRG-based classification system in support to the management of an increasing number of hospitals (276 hospitals, since April 15th, 2005), and it plans to extend this number in the following years.

The Clinical Districtual Hospital in Craiova was among the main units where a successful pilot implementation of this System has been fulfilled. Patient information is gathered using the DRGNational v4.0 specialized software program that is delivered through the district agencies. The electronically registration for a patient, one for the whole period the patient stays in hospital, is in concordance with the

new clinical observation form introduced by the Romanian Health and Family Ministry. Once collected, the data are added to a database that has to be sent monthly to the DRG department from the National Health Institute for Research and Development, Bucharest.

With such complete instruments available for the hospital financial management, an increased general tendency was to address only diseases and patients with increased severity inside hospitals, while simpler procedures are performed outside the hospital. Patients and families may thus have more responsibilities during the treatment, but following up patients may become a difficult and more challenging task for MDs.

In this case, a simple expansion or a mapping of an EPR scheme inside a Hospital Information System (HIS) was not possible. Actually, our experience with HISs showed that EPR schemes are now more about managing simple aspects like appointments than clinical and decision-making processes. The demographic databases for patients, usually known as Patient Administration Systems (PASs), do not have the capability to be shared or concurrently exploited by different programs or several replicas of the same program – a critical feature for the case of an inherently distributed system as the ambulatory setup we are talking about.

Regarding the concrete aspects of the glaucoma disease, this limitation of PASs may explain many of the shortcomings in following-up with glaucoma patients in the area covered by the University Clinic of Ophthalmology in Craiova. This area has five districts and about 2000 monitored glaucoma patients. With dozens of independent ophthalmologic offices in different clinics, tens of different ophthalmologists and heterogeneous information records – this area is a mixed distributed environment where existing office capabilities do not manage very well the flow of patient information.

As shown in the following table and graph, the number of new glaucoma cases grew slowly every year. In our opinion this increase was the result of more careful examinations that resulted in accurately diagnosing more of the existent glaucoma affections and *not because the number of actual affections significantly increased.*

Table 1. Primary Open-Angle Glaucoma (POAG) Cases by Year.

Year	#Hosp	#POAG	#New	% of POAG
1989	2854	393	218	55%
1990	2792	422	225	53%
1991	2250	385	213	55%
1992	2573	413	226	55%
1993	2610	408	229	56%
1994	2553	461	258	56%
1995	2417	432	251	58%
1996	2513	425	261	61%
1997	2479	406	257	63%
1998	1932	367	221	60%
1999	1831	398	236	59%
2000	2379	341	202	59%
2001	2035	317	194	61%
2002	2169	367	228	62%
2003	2128	309	193	62%
TOTAL	35515	5844	3412	58%

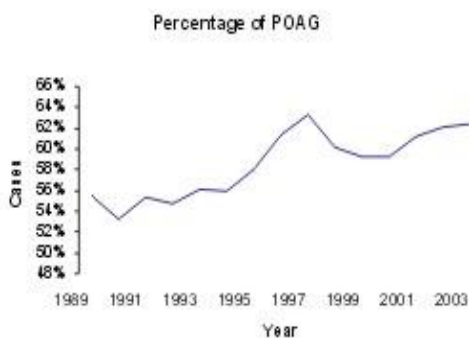


Fig. 1. Percentage of new POAG cases, by year.

The great number of glaucoma cases and complex treatment also called for a system to assist the physicians in the new setup depicted by patient distribution outside hospitals. Our primary goal was to design and rapidly implement such a system, with limited on-site assistance from medical doctors (MDs). For that reason, agile methodologies and extreme programming (XP) were employed, to do the job.

Material and Method

The term “*glaucoma*” in our system specifications refers to a group of diseases that have certain common features, such as increased intraocular pressure leading to the atrophy of the optic nerve head and visual field loss, insidious and slow progress, and absence of pain [2], [3]. Higher intraocular pressure (IOP) compared to the individual tolerance (normal limits), leading to the atrophy of the optic nerve head and visual field loss. Primary open-angle glaucoma (POAG) is insidious in onset, slowly progressive, and painless. Because the visual acuity generally is spared until late in the disease, the visual loss progresses without symptoms [2]. It is estimated that 1-2% of our district population above 40 years has POAG, this representing 12-15% of causes of blindness. This emphasizes the importance of correct monitoring of the disease.

Moreover, a particular form of POAG, the *normal tension glaucoma* - NTG (also called low tension glaucoma), does not show elevated intraocular pressure, although it resembles the primary glaucoma in all other respects. In the absence of any deterioration of visual acuity and of increasing of ocular pressure in NTG, one of the most important investigations for the diagnostic and stadialization of this disease is represented by visual field examination and recording, which has the same importance in supervising medical and surgical treatment [3]. This also calls for cross-examining, sharing and periodical reviewing of information.

Glaucoma is thus a major ocular affection, with maximal incidence in people over 60 years of age. Medical treatment, if correct, may slow down or stop the evolution of the disease. Surgery (classical, laser, or ultrasound) as the ultimate solution must be done within weeks. Statistics show this is performed more and more rarely; this is largely due to correct monitoring and new medication.

As for the other aspects of our *system specification*, the general picture may be put simply like this:

- Ophthalmologic cabinets are provided with at least one computer per office, powerful enough to run database management systems or servers. Terminals are mostly used to display simple screens (for data entries and record examination), with low variability in terms of customization for specialty or particular affection (disease).
- Patients are examined and receive a free prescription, usually once a month from the same medical doctor. Patients may change doctor sometimes – this simple fact leads to a series of complications, which may become critical for an invalidating affection like glaucoma.
- Although previous investigations for a patient may be available locally, there is little room now to make them easily and immediately available for every office. Thus, further correct treatment is heavily dependent on previously issued records carried by patients themselves (and sometimes lost).

All these problems lead us to conclude that designing and implementing a new EPR scheme, based on a distributed database as a demographic core or PAS, is necessary for monitoring glaucoma. This will allow medical doctors (ophthalmologists) to view and modify patient information and records in a safer, more flexible and efficient manner. The very compact and efficient design should allow our system, in a longer-term perspective, to incorporate efficient methods for storing medical images and facilities like content-based medical image retrieval – necessary in the monitoring and comparing visual fields in POAG and NTG, or physician assisting capabilities for which there is an ongoing research [4].

The *structural specifications* derived from the goals of the system were:

- To create a low cost back-office distributed information system.
- To provide web-based capabilities.
- To provide a flexible and efficient manner to manage patients data.
- To allow all registered medical doctors (ophthalmologists) to view and securely modify patient information and records.
- To facilitate future extensions and additions as well as scale-up of the system, without a drastically reduced performance in operation, and exponentially increasing cost.

As foreseen, the correct implementation of an HIS/ EPR scheme in our system was meant to support:

- Availability to more MDs at a time.
- Off-site availability.
- Data backup.
- Audit.

- Decisions.
- Reports on activity, patients etc. (that can be created automatically).

Since the existent information was spread in many different existing databases, managed by different database management systems (DBMS) under the control of several types of operating systems (OS) running on different computers, the envisaged solution was to design the software system (package or application) in a client-server or distributed manner. This also had to account for the possible diversity in inter-networking, and to provide transparency.

The *logical structure* of the system has to retain much of the key characteristics of the initial process, as showed in Figure 2.

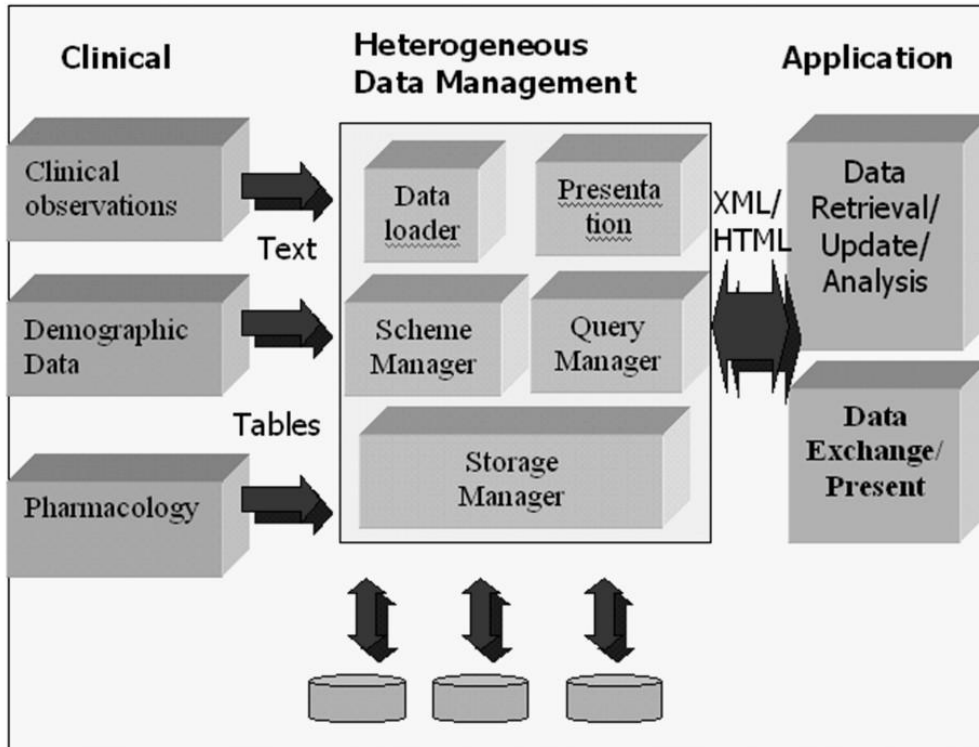


Fig. 2 . Software system architecture.

The core of the system can be seen as a distributed database system, primarily handled by a server program linked seamlessly to several types of clients. This software, making use of the Internet infrastructure and adopting up-to-date technologies, addresses the urgent need to handle medical information flow and management.

As we followed up the above ideas in the implementation, in order to reach all the goals and benefits of the project, we found important to provide firstly a simple and correct network design. The number of levels was kept to a minimum – a central powerful server in the Clinic of Ophthalmology and the existing client computers in the physician offices form two levels. A third level, consisting of dispersed computers accessing data using Internet browsers has been considered but not (yet) implemented.

As for the application, the main functions foreseen were also two:

1. *Simple Network Management (or Back-office)* with respect to the physicians and patients assigned to a node, providing:
 - Local management of information in the node, regarding patient and physician identification, patient disease and therapy history, current therapeutical decisions for a patient.
 - Cumulative statistics, regarding all patients of a given physician, or other activity aspects such as: number of patients, medication costs.
2. *Data-distribution* (obviously this function refers to the data collected from nodes).
 - Data collected for a patient in a local node is uploaded, at least once a day, on the server, thus being available for other nodes as well.

To achieve this functionality, the local database in a node must implement some minimal relationships, i.e. for patients, physicians, current therapy, disease history, investigations. The central database must implement additionally some other relationships, such as the one for medical offices (locations), for events during therapy, a.o.

To implement the functionality of the databases, the DBMS must provide the operations as transactions; a large amount of data and network traffic must be natively supported, especially if we consider keeping the consistency of data for a large number of nodes. A simple method can be similar to the following: if one link (field of a record) in a table points to another table with a missing corresponding record, an update field in the original record must be properly positioned. The management of a large number of concurrent requests and collision avoidance can also be provided by software design, using generalized semaphores specially conceived.

As for the usual communication manager present in such a system, we have considered a dispersed design, i.e. its functions are distributed among several modules (scheme manager, storage manager, query manager). For instance:

- A server part can log all the chat data (as a Java Application).
- A client side can be composed by applets and designed to be platform independent.

The design for data retrieval/ update/ analysis considered only a few main components, of paramount importance being:

- The *back-office management* module: any physician may access personal data about any patient from any host computer where the software is available.
- The *Internet* module: this allows other ophthalmologists on one hand, patients on the other hand, to consult most relevant data managed by back-office management module, like latest prescriptions.

The implementation used a standard 3-tier **MVC** (model-view-controller) architecture. This MVC outline is standard in implementing a formal separation of the three layers: **view** (manages graphical output), **controller** (business logic) and **model** (data and its behavior). We employed this strategy in previous projects [5]. As a remark to facilitate the understanding in our approach: MVC was originally developed to map the traditional input, processing, output roles into the GUI design field, i.e. "Input --> Processing --> Output" is roughly the equivalent of "Controller --> Model --> View".

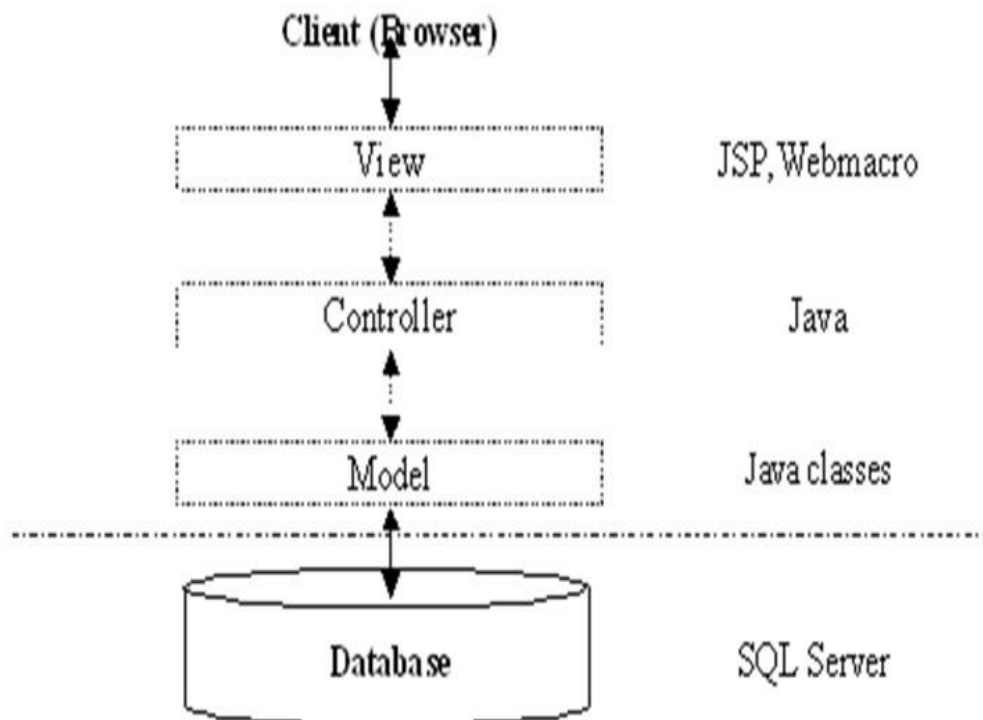


Fig. 3. The standard MVC architecture.

Because one of our implementation goals was to develop a uniform system that can be easily modified, extended and updated in the future we also chose Java technologies (Java Server Pages and Servlets) for the Web part implementation. For Web presentation of information, main features considered were the following:

- Data should be available from any computer connected to the Internet through a secure protocol (HTTP over SSL).
- The only purpose of the Internet module is to present data.

One of our objectives is to smoothly cross over our actual pilot implementation to a fully HL7 – based system (Health Level 7). HL7 is widely used in healthcare systems and our previous working experience recommends it as the best healthcare information standard that we can adopt [6]. Because the existing applications and their back-end databases that have been used in the private cabinets were not using any standard, we tried to make a compromise between using medical information systems standards and our primary goal. The communication in our implementation is done using Extensible Markup Language (XML), which is also used by HL7 v3 messages. The usage of Java for implementation will help us to easily integrate the HL7 standard in our system using HL7 SDK, which is a set of libraries that are adding support for HL7 to Java applications [7].

The back-office software architecture is composed of two modules, as showed in Figure 4:

- Server back-office software.
- Client back-office software.

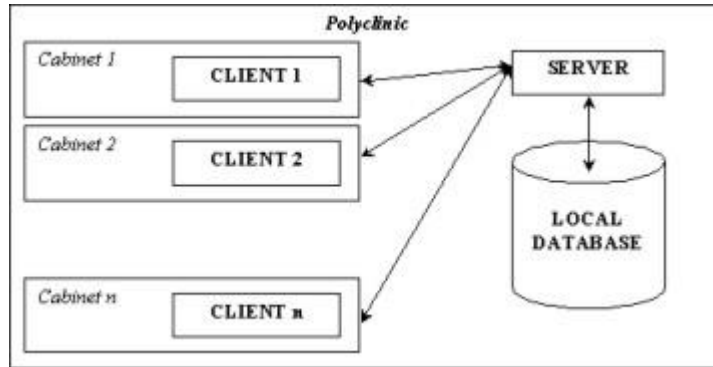


Fig. 4. Back-office system architecture.

Asynchronous communication among physicians and patients, and entity aggregation must be provided: information may reach one patient, a target group of patients or all patients. This includes not only medical information – the patients may be easily informed about community events.

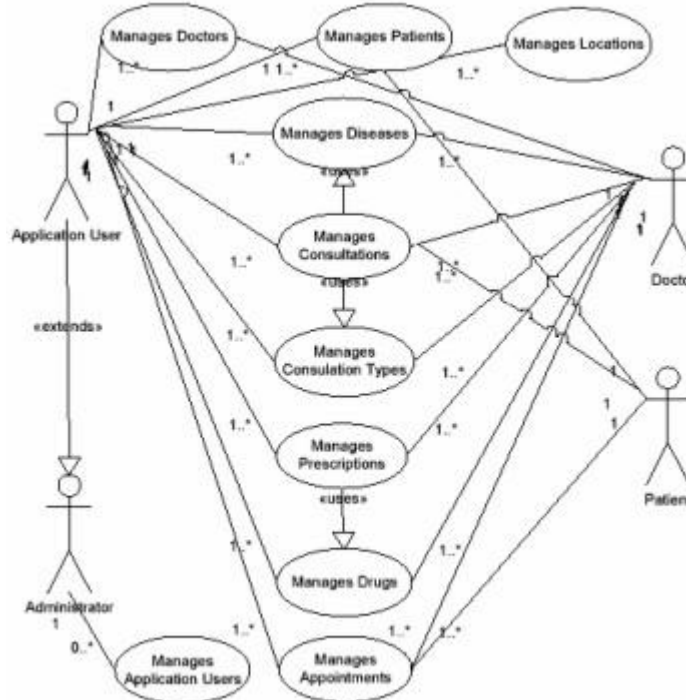


Fig. 5. UML diagram showing how requirements are collected.

Use case diagrams (UML), like the one in Figure 5, have been used to describe this functionality during requirements gathering, across members of the design team and MDs as consultants.

Final aspects considered during design were those regarding possible optimizations, such as: information archiving, garbage collection, and ordering of information using multiple indexing to provide faster response time. Although reviewed in the system documentation, they were not considered in the current implementation.

Results

The technology chosen for the implementation was standard. We considered Microsoft SQL Server 2000 the most suited DBMS and Java capabilities proved sufficient for interfaces and distributed communication among small client programs running locally.

As for the database structure, the back-office management module is also based on a RDBMS (Relational Database Management System). As it can be seen in Figure 4, two different architectures for the databases needed may be used:

- The local database at the server (denoted from now as *central DB*), that is used to gather information from all polyclinics or cabinets enrolled into the system, and maintained for consistency.
- The local databases at clients – installed and maintained for each polyclinic or cabinet.

The main relationships found in local databases are:

- For patients (Patients).
- For physicians (Doctors).
- For current therapy (Prescriptions).
- For disease history (FoundDiseases).
- For investigations (Consultations).
- For doctor's appointments (Appointments).
- For location of:
 - Ophthalmologic cabinet (Location)
 - Patients/doctors (Towns & Counties)

They can be depicted as follows:

- **USER** (id, Firstname, Lastname, sex, Birth date, address, phone1, phone2, email, status)
- **Role** (id, name)
- **USERROLE** (Userid, Roleid)
- **INVESTIGATION** (id, Physicianid, Patientid, date, Prescriptionid, Diagnosisid, observations, status)
- **PRESCRIPTION** (Id, Startdate, Enddate, Medicineids, Observations)
- **DIAGNOSIS** (Id,Name,Observations)

Or, in a pictural view, like in Figure 6:

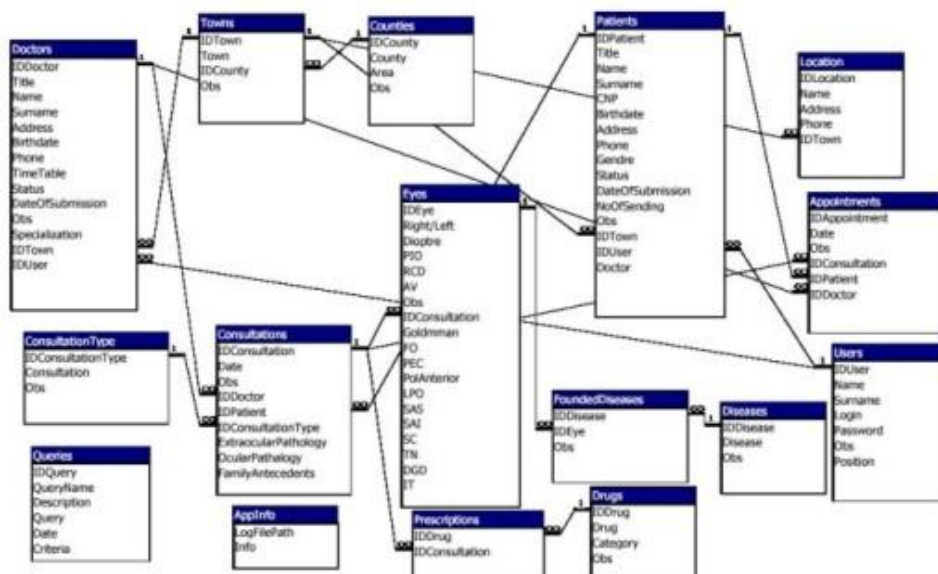


Fig. 6. Database relationships.

The overall system architecture - projected on the distributed database functionality needed – is shown in the next diagram.

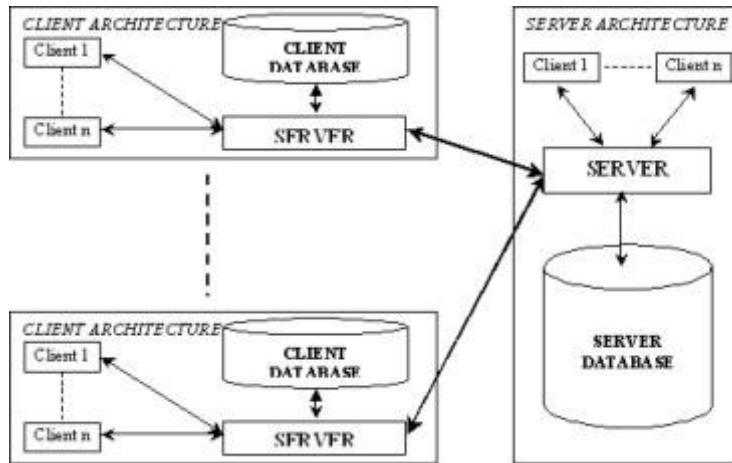


Fig. 7. Distributed database system architecture.

Grouping relationships can be done with a view to obtain:

- Demographic information and statistics (*Demographic Data Block*).
- Clinical information and statistics – number of investigations, treated diseases, evolution of diseases through time (*Clinical Observations Block*).
- Pharmacology information and statistics – drugs prescribed (*Pharmacology Block*).
- a.s.o.

Because a distributed DBMS represents a unitary logical system physically distributed in nodes, compatibility was required between the server and local DB implementations (i.e. the table structure of the local DB is similar to the server database structure). Microsoft SQL Server has been chosen because it is good for large databases with many concurrent accesses, offering support for all databases operations. Local databases may use Microsoft Access, MySQL or Microsoft SQL Server as DBMS, according to the platform, number of patients and the power of target computer machine.

The approach in the implementation can be broadly described as bottom-up, that is, the system grew from small applications to a complex functionality. The requirements and basic features provided by design for the initial local applications implemented were:

- Capability to run as an independent application – i.e. to run with or without a server.
- Local management of information in the node (regarding patients and physicians identification, patients disease and therapy history, current therapeutical decisions for a patient).
- Cumulative statistics (regarding all patients for a given physician, or other activity aspects such as: number of patients, number of investigations, evolution of diseases etc.).
- Possibility of searching patient's data through distributed medium (formed by all computers where the software is available), a.o.

In the local implementations, rapid application development was applied on the basis of user scenarios (UML), and then Java was used, because of:

- Platform independence (use on different platform without changes)
- Its unequalled distributed facilities offered for:
 - o Network applications
 - o Secure communication
 - o Databases support
 - o Graphical statistics
- A lot of classes already implemented providing an attractive Graphical User Interface (GUI)
- Possibilities of extension (Java is a pure Object Oriented Language)

Use case diagrams (UML), similar to the one depicted in Figure 5, have been used to describe the functionality of the local system at level II.

In terms of local application presentation, some of the main features foreseen were:

- Friendly User Interface.
- Good security.
- Options for novice/advanced computer users.
- Statistics at different levels:
 - Local (cabinet level),
 - Global (town/zone level).
- Backup solutions for local database.
- Query facilities for both local/distributed databases.

For instance, the local application record management refers to doctors (physicians), patients, timetables, and administration data - the usual access being provided only to the physician public information (retained by Doctors table in local DB).

Physician public data can be viewed with few restrictions, but private data can be viewed only under administrator password. A supplemental level of security is required if data has to be erased, added, or modified in the DB.

Patient searches can be performed easily both in the local database, and in the entire system, when a proper tag is checked. The explicit local and global searches were preserved for both historical reasons and to speed-up the process in certain situations.

Information can be added to the database only by using the local application, as shown in Figure 8. This is a restriction due to the existing systems and has an important consequence: a local DB is needed even at the location of the server DB. For maintaining consistency, reasonable assumptions have been made (i.e. a patient may be consulted several times a day in the same cabinet or polyclinic and local updates are done each time; he can go to another office that will be happen the sooner after one day, interval in which the central database has to be updated).

The local subsystem actually provides two data security levels, as shown in Figure 9:

- Database level (defined when local database is created).
- Application level (defined by application).

The screenshot shows a 'New Consultation' window for patient James Fraser on 10/7/2004. The consultation type is 'Control'. The form is split into 'Left Eye' and 'Right Eye' columns. Each column contains input fields for 'Dioptre' (values: +1.75 and +2.00), 'Intraocular Pression', 'Cup/Disc Proportion', and 'Visual Acuteness'. Below these are 'Goldman' field sets with letters T, N, I, S, TI, TS, NI, NS. Further down are fields for 'FO', 'PEC', 'LPO', 'SAS', 'SCD', 'TND', 'DG', 'IT', and 'PAn'. There are also 'Diseases' and 'Observation' text areas. At the bottom, there are fields for 'Extraocular Pathology' and 'Ocular Pathology', a checked 'Family Antecedents' box, and a 'Doctor' field with the name 'Raquel Roberts'. Buttons for 'Prescription', 'Add', and 'Cancel' are at the bottom right.

Fig. 8 . Data collection in the local application.



Fig. 9 . Data security.

Transferring sensitive information over Internet can be a risky process; therefore we secured the communication between clients (cabinets) and server, using the Secure Sockets Layer Protocol (SSL). SSL supports a flexible client-server authentication scheme and provides an encrypted client-server communication. SSL runs as a layer between the Transport Control Protocol (TCP) and application layer protocols, such as HTTP and SMTP. It uses public key cryptography to provide authentication and private key cryptography and digital signatures (certificates) to provide privacy and data integrity. To use the power of SSL in our system we also made use of the Java Secure Socket Extension (JSSE) libraries that are a standard part of Java 2 Standard Edition (J2SE) platform (version 1.4).

Conclusions

The specialized health record management system presented here with core functionality in monitoring glaucoma, and core data organized in the form of a distributed database system [8], is designed to match the existing standards for information systems in the medical field. Its bottom-up implementation was kept flexible with the main purpose to accommodate any anticipated future requirements.

By providing a very intuitive interactive environment and a user-friendly interface in use for medical doctors, medical assistants and managers, supplemental envisioned benefits of such an approach are:

- Supervision of treatment and possible change of opinions among physicians.
- Correct planning of investigations (liquid test probe, perimetry - including computer perimetry, ophthalmoscopy, retinography).
- Avoidance of therapeutic "disasters" due to interruption of medication.
- Improvements in all the managerial and decisional aspects (regarding costs and time delays).

References

- [1] M.Tamer Oszu, P.Valduriez, Principles of Distributed Database Systems (2nd ed.), Prentice-Hall, 1999
- [2] Diagnosis Related Groups (DRGs) and the Medicare Program: Implications for Medical Technology – A Technical Memorandum (Washington, D.C.: U.S. Congress, Office of Technology Assessment, OTA-TM-H-17, July 1983)
- [3] J. F. Risse, Exploration de la fonction visuelle: Etude du champ visuel, Ed.Masson, Paris, Milan, Barcelone (1999) 190-245
- [4] J. Caprioli: Automated Perimetry in Glaucoma, Am. J.Ophthalmol., (1991), 111, 235-239.
- [5] C. Mocanu, C. Badica, M. Mocanu, A Knowledge Model for Glaucoma Diagnosis, Craiova Medicala: 133-136
- [6] K. Cleary, A. Patriciu, S. Xu, M. Mocanu, Software Architecture for Robotically-Assisted and Image-Guided Minimally Invasive Interventions, in Proc.of the 16th Int. Congress on Computer Assisted Radiology and Surgery (2002): 218-223
- [7] Systems integration and Globalisation – Global healthcare, <http://www.linkmed.com/Programs/LinkMedical.pdf>, 2002
- [8] Reducing Complexity and Costs in Healthcare IT Development, Sun, http://www.sun.com/br/1004_ezine/hc_hl7java.html