

Internet-based Medical Teleconsultation System

Edgar Pek and Sven Loncaric
Faculty of Electrical Engineering and Computing
University of Zagreb
Unska 3, 10000 Zagreb, Croatia
edgar.pek@fer.hr, sven.loncaric@fer.hr

Anamarija Margan
Internist Practice
Cres, Croatia

Abstract

Rapid development of telecommunication networks has made possible development of telemedicine, and especially on of its segments medical teleconsultation. In this paper we will present system which is designed to provide tools for establishing meaningful communication between general practitioner on dislocated areas and specialists typically located in large medical centers. Technological approach for implementation of the system is based on the open source solutions. System is implemented on Linux platform. Web server is Apache extended with support for PHP, and mod-SSL, which interfaces Apache with Open-SSL library, for data encryption. Data storage is implemented using database management system PostgreSQL. To describe data newly emerging XML (eXtensible Markup Language) standard was used. Transformation of data is achieved using another second-generation Internet technology - XSL (eXtensible Style Language). CSS (Cascading Style Sheet) is technology which is used for visual rendition of transformed documents. PHP is server side scripting language which is used to access database and generate XML documents. It is also used to interface Sablotron XSLT processor. PHP acts as "glue" to all pieces, and produces web page. System is based on newly-emerging technologies, and it represents a base for providing a higher level of health care.

1. Introduction

Croatia has indented coast, characterized by a large number of islands. Specialists practices on islands are not financially self-sustainable, so implementation of efficient health care on the islands and other rural areas has always been a challenging task. In particular, successful implementation of health care on the North-Adriatic islands of Cres, Mali Losinj, and neighboring islands has shown to difficult, especially during high tourist season when the island

population becomes much larger then during the out-of-season period. Patients requiring specialist opinion must travel to distant medical centers which is often not feasible due to patient state or in medical emergencies. Improvement of health care on islands and other rural areas can be achieved using electronic means of communication. Development of telecommunication networks has a key role in development of a telemedicine. Our system represents a possible implementation of teleconsultation, which is one of the many aspects of telemedicine. Very fast growth of Internet, and acceptance of second generation Internet protocols has initiated development of advanced Internet based applications. Following that facts Internet based application for medical teleconsultation is obvious solution. Basic idea is to achieve greater quality of communication between general practitioners(GPs) and specialists using teleconsultation [7], which results in improvement of patient care. Realization of the desired goal is based on two main requisites. First, efficient way to exchange messages between GPs and specialists must be provided. Second, adequate exchange of information concerning patient have to be provided [20]. Furthermore, confidentiality of communication and stored data have to be guaranteed. It can be said that ethical aspect is equally important as a technical aspect of the system [6] [19].

2. System architecture

We have built system that can be described as three-tiered client-server architecture. Client in our architecture is implemented as *thin-client*, so presentation service is just web browser. Process service (application service) is interface between presentation service(web browser) and data(database) and in our system is implemented using PHP. PostgreSQL is database management system and represents data service. System is implemented on Linux platform. The architecture of the system is shown in Figure 1.

Procedures considering the system shown in Figure 1 can be described as follows. User can access web pages

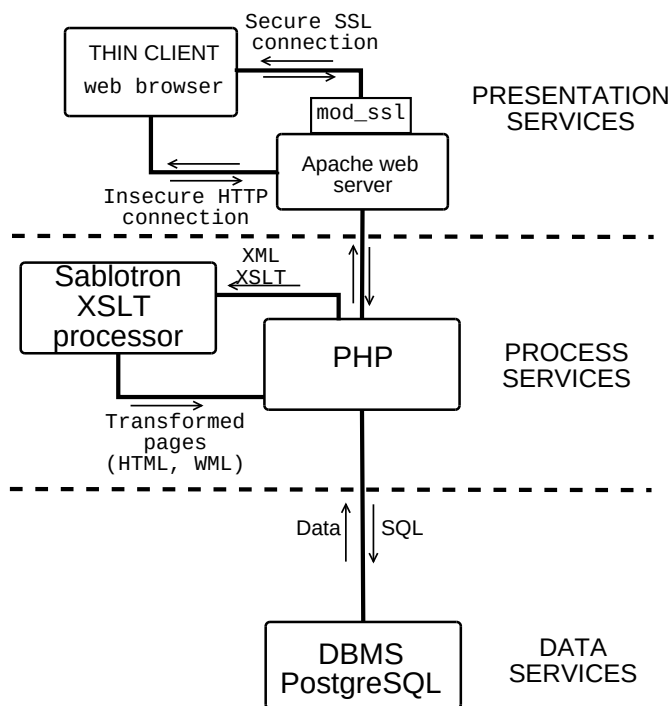


Figure 1. System architecture

using SSL protocol (Secure Socket Layer) or plain http. Through Apache web server PHP scripts are called, using scripts database management system is accessed, and received data is processed in a way that XML data is formed. PHP scripts are also used to call Sablotron XSLT processor using XML data which was formed from SQL queries or XML data stored on disk, and XSLT scripts written to form a HTML (currently just HTML is produced, but whole system is designed in a way that other formats such as WML can be produced with minimal additions).

2.1. Presentation services

Presentation services, as it can be seen on Figure 1, consists of two main parts. First part is user interface, which is web browser. So it is obvious why it is called *thin-client*. Second part of presentation services is web server. Apache web server has been used, to achieve secure client-server communication it was compiled with support for SSL protocol, mod_SSL is compiled as DSO (Dynamic Shared Object), and it represents an interface between Apache and OpenSSL library. Requests to database are made through web server, so support for PHP was also compiled as DSO.

Currently, web pages can be accessed through plain http, server listens for requests on port 80, or through SSL protocol, in this case server handles requests received on port 443. System is still under development, but when test phase is over only secure connections will be allowed. Popular

web browsers such as Internet Explorer and Netscape have support for SSL, so clients can access data using secure connection. Details about mod_ssl, and which browsers currently support can be found in the User Manual located at [5].

To summarize, presentation services must provide user interface to the system and functions of web server with expanded possibilities, such as secure client-server communication, means to handle data from HTML forms, and interface to the database. Data is passed to process services, and then result of the process is presented.

Advantages of the *thin-client* approach are:

- platform independence
- easier system maintenance
- better network communication performance

One can notice that presentation services could include just web browser as user interface, and the web server then belongs to the middle tier, or process services.

2.2. Process services

Process services are the second tier in the system architecture - Figure. This tier represents interface between presentation and data layer. Because access to data is done through this layer, complex business rules must be implemented, and transformation of data in the format appropriate for presentation service is performed. As it can be seen on Figure 1 this layer consists of PHP scripts, XML data, XSLT scripts.

Most of the programming considering the web application described in this article has been done using PHP scripts. PHP¹ scripts are used to coordinate all procedures performed in the system. PHP handles data which was passed from HTML forms in the way that SQL queries are formed, queries are sent to the database and then results of the queries are processed. Process of the query results is basically creation of the XML data. Sablotron, which is compiled as PHP module, is used to transform XML data according to XSLT scripts. Result of the Sablotron processing is HTML, this result is passed to presentation services.

Development of web-based application in general is continuous process, so maintainability is very important aspect. Following that principles, object-oriented possibilities of PHP was used, abstraction of code from HTML was achieved using FastTemplate. FastTemplate provide means to separate page structure from content. Furthermore, to separate content and visual rendition CSS(Cascading Style

¹PHP was previously abbreviation of the "Personal Home Page", then "Professional Home Page", now PHP is defined recursively as PHP - Hypertext Preprocessor.

Sheet) language was used (to ensure maximum possible portability only CSS Language Level 1 properties were used). Another important consideration related to maintainability is connected with writing XSLT scripts. XSLT scripts are written in a way that independence in relation to XML element type name, and number of elements is achieved. This is important, because changes in database can be performed, but changes of XSLT scripts are not required. To summarize, it can be said that process services implements all business logic of the system, process services act as connection between presentation and data layer.

2.3. Data services

Data services are the third tier in the system architecture. This tier represents database management system, and we have used PostgreSQL to provide required functionality. Functionality of this layer is to:

- Store data
- Provide means to preserve data integrity
- Ensure that users is authorized to perform certain task
- control parallel access
- provide means to restore database in case of failure

Data integrity is achieved by implementing integrity rules, triggers, and stores procedures. Data service provides means to unale unauthorized access to data, such as grant user perform of only specified tasks, creation of views, period of access, etc. Restoration of database is achieved by backup of data, and logging transactions.

3. Database description

In this section we describe structure of the database, which is used in the system. The database can be divided in two main parts. The first part store teleconsultation data, while the second part contains electronic health record data. The database contains:

- User (physician) data which contains user ID, personal information, medical specialization, login, and password
- Audit information about user access to the system (user identity, actions taken)
- Teleconsultation requests which are being sent by physicians
- Patient records which consist of: personal information, laboratory results(text, images), allergies, anamnesis(case history), status

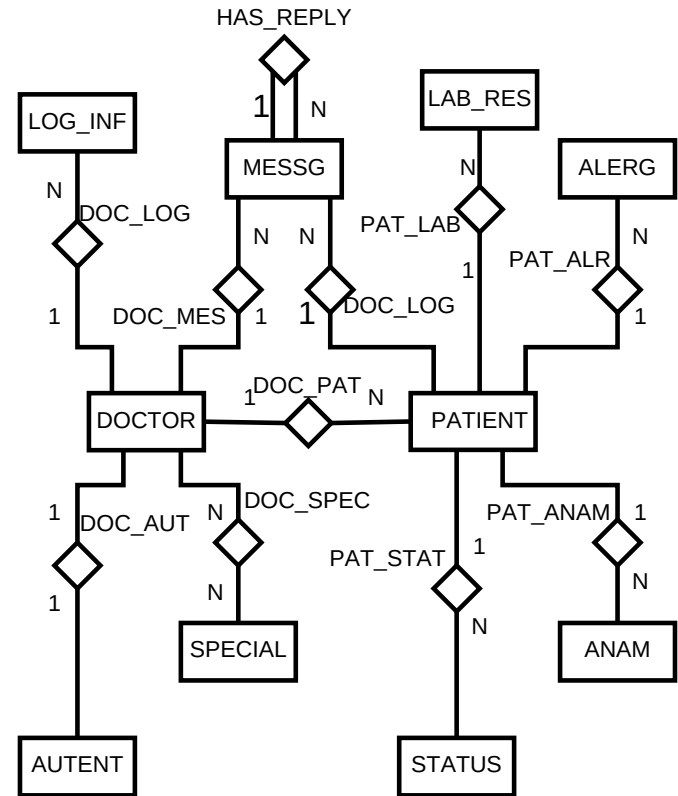


Figure 2. Entity-relationship model

E-R diagram shows entities and how entities are mutually related, e.g. entity *SPECIAL* (stands for specialization) and entity *DOCTOR* are related through relationship *DOC_SPEC*; mapping between entities is N:N, which means that a physician can be specialist in more than one area, and several physicians can specialize in a single area. In the same manner we can describe most of the other relations between entities.

One special relation is reflexive relationship, in our E-R diagram reflexive relationship is *HAS_REPLY*, and it is connected with entity *MESSAGE*. In this case reflexive relationship is used to show that entity *MESSAGE* can have replies, replies have same attributes as *MESSAGE*, so entity *MESSAGE* has two roles, the first as message, and the second as reply.

4. User interface

User interface is currently based on HTML and CSS, so only browser is required to use system.

The system is accessed by opening initial web page where user authentication is performed. User provides password to access following pages. This is achieved using session handling provided by PHP (PHP 4.0.4 PL1). Session can be stored as cookies, if browser is set to not accept cookies, session is handled through URL.

Logically we can divide system in two main parts, teleconsultation and electronic patient record. Teleconsultation part provides user with functionality required to conduct teleconsultation. A teleconsultation request contains information about the patient which is located in electronic patient record and a note from referring physician describing the request. To achieve minimal response time the physician is also notified of a pending consultation request by means of GSM SMS (Short Message Service).

Patient record as a data structure that helps to describe important facts considering patient health has been used since the fifth century B.C. when Hippocrates proposed purpose of the patient record. His observations were noted in chronological order, so this type of record is called time-oriented medical record. Improvement of the medical record organization lead to source-oriented and problem-oriented medical record where data within sections are in chronological order.

Our implementation of an electronic patient record is based on the source-oriented concept. According to that concept the record content is structured according to the method by which it was acquired, e.g. laboratory results, allergy information, etc. Data within each source is organized in chronological order.

Electronic patient record in our implementation provides functionality for maintaining patient records. It enables user to add, edit, and delete data in patient record.

5. Security considerations

Security is very important aspect of Web-based applications especially for e-commerce applications and telemedicine-based services. System architecture, as it was shown in second section Figure 1, can be represented as three-tiered. Security issues exist within each tier, so every system component must be secured to achieve comfortable level of security.

Important security aspects are:

- Authentication
- Access control
- Audit trail
- Confidentiality
- Integrity
- Availability

Authentication is handled using simple html form, that let user to input user-name and password (alternative is to use browser dialog). The security of this process is ensured using SSL services.

Access control ensures that system resources are available only to authorized entities, in our application this aspect is ensured through middle layer and database management system (DBMS). DBMS through granting only specific action, and views enables access to exactly specified system resource.

Audit trail is chronological record of system activities, in our system it is performed using session handling mechanism (provided by php) and database.

Confidentiality is very broad security aspect, in our system this aspect is ensured through session handling mechanism which provides that sensitive private informations (login, password, etc) are not stored in the cookies (on the client machine) or in the URL. User input is checked, to ensure that commands in a manner:

```
system("/bin/mail  
attacker@somwhere.org < /etc/passwd")
```

(providing password file to attacker) could not be issued.

Integrity aspect is primarily achieved through defining integrity checks provided by DBMS, such as integrity rules, triggers, and stored procedures.

Availability aspect depends on the stability of system components in every tier. In our system web-server and DBMS will restart in case of crash.

Development of the secure Web-based application is continuous process, but still completely securing a computer against unauthorized access is extremely difficult.

6. Conclusion

In this article we have presented a Web-based application for medical teleconsultation. The application is under development. System aims at providing medical teleconsultation between physicians (on island and other rural areas) and large medical centers. It reduces risk to patient who cannot be transported due to medical problems, discomfort due to travel, and health care costs (eliminates the need for travel).

References

- [1] I. Bashir, E. Serafini, and K. Wall. Securing network software applications. *Communications of the ACM*, 44(2):29–30, 2001.
- [2] J. Bosak, J. Clark, T. Bray, and J. P. et al. Extensible markup language (xml) 1.0 second edition. URL: <http://www.w3.org/TR/2000/REC-xml-20001006>, accessed 2001-04-02, 2000. W3C Recommendation 6 October 2000.
- [3] P. Chen. *Database Modeling & Design*. Morgan-Kaufmann, 1999.
- [4] R. S. Engelschall. Openssl: The open source toolkit for ssl/tls. URL: <http://www.openssl.org>, 2001. accessed 2001-04-06.
- [5] R. S. Engelschall. User manual. URL: <http://www.modssl.org>, 2001. accessed 2001-04-08.
- [6] C. B. et al. Electronic medical consultation: A new zealand perspective. URL: <http://www.jmir.org/2001/1/e13/>, 2001. Journal Of Medical Internet Research 2001;3(1),e13, accessed 2001-04-08.
- [7] G. B. et al. Internet patient records: new techniques. URL: <http://www.jmir.org/2001/1/e8>, 2001. Journal Of Medical Internet Research 2001;3(1),e8, accessed 2001-04-04.
- [8] J. C. et al. *Professional PHP Programming*. Wrox Press Ltd., 2000.
- [9] R. A. et al. *Professiona XML*. Wrox Press Ltd., 2000.
- [10] S. A. et al. Xsl transformation (xslt) version 1.0. URL: <http://www.w3.org/TR/1999/REC-xslt-19991116>, accessed 2001-04-04, 1999. W3C Recommendation 16 November 1999.
- [11] S. B. et al. Php manual. URL: <http://www.php.net/manual/>, 2001. accessed 2001-04-06.
- [12] T. A. S. Foundation. Apache http server version 1.3 documentation. URL: <http://httpd.apache.org/docs/>, 2001. accessed 2001-04-06.
- [13] J. Joshi, W. Aref, A. Ghafoor, and E. Spafford. Security models for web-based applications. *Communications of the ACM*, 44(2):38–44, 2001.
- [14] H. Lie and B. Bos. Cascading style sheets, level 1. URL: <http://www.w3.org/TR/1999/REC-CSS1-19990111>, accessed 2001-04-04, 1999. W3C Recommendation 17 Dec 1996, revised 11 Jan 1999.
- [15] S. Loncaric, M. Silovic, E. Pek, and A. Margan. A web-based system for medical teleconsultation: First results. In *Proceedings of @ITM: Telemedicine and Health Informatics*, pages 168–171, 2000.
- [16] B. Momjian. *PostgreSQL Introduction and Concepts*. Addison-Wesley, 2001.
- [17] G. Mourani. *Securing and Optimizing Linux: RedHat Edition*. OpenDocs, 2000.
- [18] S. Pemberton, M. Altheim, and D. A. et al. Xhtml 1.0: The extensible hypertext markup language. URL: <http://www.w3.org/TR/2000/REC-xhtml1-20000126>, accessed 2001-04-05, 2000. W3C Recommendation 26 January 2000.
- [19] H. Tatsumi. Internet medical usage in japan: Current situation and issues. URL: <http://www.jmir.org/2001/1/e12>, 2001. Journal Of Medical Internet Research 2001;3(1),e12, accessed 2001-04-05.
- [20] A. van Ginneken and P. Moorman. *Handbook of Medical Informatics*. Springer-Verlag, 1997.
- [21] J. Viega, T. Kohno, and B. Potter. Trust (and mistrust) in secure applications. *Communications of the ACM*, 44(2):31–36, 2001.