UNIVERSITY OF WESTMINSTER

WestminsterResearch
http://www.wmin.ac.uk/westminsterresearch

Distributed online doctor surgery.

Hamid Jahankhani
East London Business School, University of East London

Pari Jahankhani
Harrow School of Computer Science, University of Westminster

This is an electronic version of a paper presented at the 4th International Conference On Enterprise Information Systems, 03-06 Apr 2002, Ciudad Real, Spain.

The WestminsterResearch online digital archive at the University of Westminster aims to make the research output of the University available to a wider audience. Copyright and Moral Rights remain with the authors and/or copyright owners. Users are permitted to download and/or print one copy for non-commercial private study or research. Further distribution and any use of material from within this archive for profit-making enterprises or for commercial gain is strictly forbidden.

Whilst further distribution of specific materials from within this archive is forbidden, you may freely distribute the URL of WestminsterResearch. (http://www.wmin.ac.uk/westminsterresearch).

In case of abuse or copyright appearing without permission e-mail wattsn@wmin.ac.uk.

DISTRIBUTED ONLINE DOCTOR SURGERY

Hamid Jahankhani
University of East London, East London Business School, Longbridge Road, Dagenham Essex RM82AS,UK
Email: hamid.jahankhani@uel.ac.uk

Pari Jahankhani
University of Westminster, Harrow School of Computer Science, Watford Road, Harrow HA1 3TP, UK
Email: parij@westminster.ac.uk

Key words: World Wide Web, OO, UML, RMI, Servlet, Information systems, Oracle 7, Server, NHS

Abstract: This paper reports on redesign of the existing manual system of a Doctor Surgery, to a computerised system, which takes the advantage of the latest technologies and allows the patients to have better interaction with the system.
The Doctor surgery plays a major role in human life, over the years we have seen the drastic changes in the treatment of patient in surgery, however we haven't really seen much changes on structure of the system as a whole. Many surgeries still use a manual paper based system for their transaction. The recent rapid development in web technology and growth of distributed processing seems to be only applicable for commercial business and field such as medical treatment seems to have fallen behind in the technology and as consequence, inefficient and ineffective services provided to the patients. The new prototype system has been designed using Object Oriented Methodology and implemented by using mainly JAVA (RMI, SQL, SERVLET and other Java packages) for creating the communication server and the web site. Also, for the end user interface of the database in the surgery ORACLE 7 and Developer 2000 application was used.
The implementation of the system allows the patient to carry out appointment transaction (create, query, delete) and communicate with the doctor via the web site, which is connected to the oracle server in the surgery. The web site provides all the necessary details and information about the surgery and practice. The final prototype utilises distributed technology and built upon the research carried out.

1. Introduction

At the end of 2000, the National Health Authority launched a major pilot national electronic library for health. It contains the most up to date knowledge about illnesses, medicines and treatments. Patients primarily benefit at present through its contribution to NHS Direct Online. Though the Library is currently geared towards the professionals, it is hoped that the pilot gives information about how patients and the public will want to use it in the future. Feedback from users (there is a form to fill out online) will inform the Library's evolution, NHS Information Authority site, 2001.
Selected patients at two pioneering GP practices in March 2001 were the first in the UK to have direct on-line, secure access to their own medical records. The pilots represent a large step forward in reaching the target set in the NHS Plan to make electronic health records available to everyone by 2004. The patients took part in these groundbreaking pilots at GP practices in Derbyshire and Oxford. This was organised to explore how patients can make best use of access to their personal medical records. The pilots were to look at not only patient's views and attitudes to the type and format of information available but also those of the GPs and other practice staff.
Electronic patient records will help to put patients in control of their own health and health care. Patients will be able to look at their medical records prior to seeing their GP ensuring that they are fully informed about their health and have time to think of the questions that they wish to ask. This will make consultations faster and clearer as both the doctor and the patient will have access to the same information

prior to consultation. The NHS Information Strategy made a clear commitment to providing electronic patient records for everyone in UK, NHS direct site October 2001 and EDDIP (2000).
In recent years, the object-oriented approach has received attention from the computer and information systems communities. Many authors state that object-orientation allows us to intuitively and naturally model the real world, Borgida et al, 1985, Martin and Odell, 1992, McGinnes, 1992, Booch et al, 1998. Proctor and Creech 2001, conducted research to explore the development of an object-oriented model to support inter-operation of simulations within a federation for the purpose of conducting medical analysis and training over a distributed infrastructure. The medial federation is referred to, as the Combat Trauma Patient Simulation System and was composed using high level architecture. The use of object-oriented technology has increased significantly in the area of integrated information systems through the use of programming tools and methods such as Unified Modelling Language (UML) and more recently as a formal notation to support analysis, design and implementation of information systems Monarchi and Purh, 1992.
This project has been carried out on the behalf of a Doctor Surgery, to redesign the existing manual system to a computerised system.

2 Surgery background

This doctor surgery has been around for more then 20 years, it is one of the oldest and well known around in Chingford, Essex, UK. The primary object of the surgery is to provide the patient with best possible health care service, with the minimal waiting time. It consists of two doctors, a nurse, two staffs and surgery manager. It provides services such as child vaccinations and health surveillance, chronic disease management, maternity services etc. Making an appointment in advance carries all these out. Appointment are made or cancelled by either phoning or by visiting the surgery within the surgery time.

2.1 Analysis of the existing system

Conducting a fact-finding mission to get a better understanding of the system carried the feasibility study of the existing system. From the interviews of the staffs, observation of the system and general survey for patients it was found that the existing system uses mainly a manual transaction apart from the storing customer details on the computer.
Appointment booking can be made by phoning or visiting the surgery within the opening hours which are Monday to Friday 9-11AM and 4-6PM. No appointment can be made outside this time, because there is no one present in the surgery.
The date/time of the appointment and the name of the patient are required when making an appointment, which is written down in the appointment book.
When attending the appointment**,** patients come into the surgery and mention their name to the secretary. The secretary verifies the information and asks the patient to sit in the waiting room, meanwhile the secretary searches through a filing cabinet for the patient record file and forwards to the doctor when the patient is ready to be seen. The patients are advised to come 5-10 minutes before the appointment, so the patient history can be traced and ready to be seen by the doctor.
When the patient is ready to be seen by the doctor, the secretary takes the medical history file to the doctor and their name is called out. Once the patient sees the doctor depending on the status of the patient the doctor prescribe him/her with the necessary medicine, this information is written in the medical history file.
To cancel an appointment patient ask the staff by giving the date/time and the name. The cancellation notification should be given within 24 hour before the appointment time and during surgery hours.
To registering a new patient, there are several steps, these are:

- Patient enquires about joining the surgery in the reception.
- The receptionist consult the surgery manager, for confirmation
- Patient is given a form to fill in. This form includes all the necessary information about the patient and important medical history.
- The information is then stored in the computer as a record.
- The patient is then ready to make appointment and see the doctor.

A new medical card will be sent to the person by post and provide confirmation of the change of the doctor.
Once a patient has decided to change the doctor, the patient's history is removed from the existing filing system and forwarded to the NHS centre. Patient details are then deleted from the computer. Repeat prescription, is currently done by patient writing their

name and the name of the medicine on a piece of paper and when it is ready it can either be collected from the reception or sent to home address. Then in the reception there is a box called "repeat prescription" where the paper is placed and two days advance notice is needed. Due to number of drawbacks, including data duplication, large storage spaces, etc. Therefore, it was clear that the existing system is no longer live up to the expectation of the today's demands, and a new system is needed which can deliver all the necessary needs.

2.2 Functionality of the new system

The new system will provide the following:

- All the existing record will be stored in the relational database which includes all the patient's history and personal details, staff.
- A Server will be implemented to carry out all the transaction in the surgery, the server will communicate with the database. This will allows distributed processing since the client is transparent to the server, the client can be located anywhere.
- A local area network using four computers linking, one computer with the server, database and other two computers for doctor and receptionist.
- A website will be available for the surgery and will allow online booking system. This will allow patient to book, cancel, query appointment. The website will also display all the necessary information such as any latest crisis, outbreaks etc.
- The patients will be able to communicate with the doctor using a customised server in the surgery. This facility will be available for patients only.
- Different access mode will be provided for different users on the net and with in the surgery. This ensures that only the required information is available to the user.

2.3 Designing the new system

Object-Oriented methodology (UML) and Java programming language was used for the systems development. Java was used rather then C /C++ or other languages for the implementation, since it is platform independent, which means once the Java source file is compiled the class file can be taken to any platform and run the class file without having to recompile the class files again.
This is useful in distributed system where a system is developed in heterogeneous environment, the client can be sitting at one location and the server can be located at different location. The other reason for choosing Java is that, it has lot of build in classes ready to use, this allows faster development of application. And finally Java provides a better security than other language, Java™ Security site, 2001. The new system was designed purposely as a distributed system. It was designed that way so that the users are transparent to the implementation of the server, where located, how it's set up, etc.

2.3.1 Section One: Internet

This is where the end user interaction will take place and was designed using HTML for the static and dynamic content of the web pages, Java applet for the doctor communicating with the patient and Java script for the validation purpose. HTML was used rather than XML, since it is supported by virtually all the browsers and relatively straight forward to develop. Since a communication channel is created between the doctor and the patient, we know that HTTP (Hypertext Transfer Protocol) is state-less which is no use for communication purpose, we need to create a socket connection to the host port. Applet can make socket connection to the host, which it was loaded from. Therefore a communication channel is possible using applet. The applet allows the users to make remote method calls to the RMI server over a network; it also provides dynamic content to a web page. Since applets are not allowed to write to and read from files and can only make socket connection to the host it was loaded from, makes it more secure to use.

2.3.2 Section Two: Middleware

This is without a doubt the most critical part of the system, and need to be implemented properly. This section is the manager of the system. It deals with the entire request and uses the appropriate resource to give feed back to the all the users to the system. There are three servers and other components sitting in the middware, these are:

- RMI chat server
- JWS web server
- Oracle database Server, JDBC driver and Java Servlet

The RMI chat server allows the patient to talk to the doctor over the network. It deals with the authentication of the patient and doctor over the network. The JWS is a pure Java web server and it provides all security of Java and is also platform independent.
ASP (Active Server Pages) by Microsoft for creating dynamic web pages cannot be used because, it is platform dependent and used in windows environment, CGI-script is an alternative, but the problem with CGI-script is the security issue. If it's not written properly hackers can break into the system easily. Java Servlets deals with both of these problems Servlets are protocol- and platform-independent server side components that can be dynamically loaded like applets. And since Servlets is part of Java it also implements the Java security. RMI (Remote Method Invocation) was used for distributed application. RMI implements thread, therefore it deals with the concurrent clients.
Oracle is set up as either a server or client. When it is set up as server the database resides on the server side and allows multiple clients to connect concurrently and also it allows different access right to different part of the database. This was the reason for choosing Oracle.

2.3.3 Section Three: Hierarchical DBMS

Hierarchical database was used since information like patient history needs to be protected so that only authorised users can access the information.

2.3.4 Section Four: LAN (Local Area Network)

This is used with in the surgery. A computer will be set up as the main server. The rest of the other computer in the surgery will be set up as the clients. In the surgery each client will log in to the Oracle database server and will be given a different access right to the database system. The end user application in the surgery was developed in developer 2000.

3 Implementation

The complete system was developed in components. Each component provides adds-on functionality for the complete system. The sequence for building the system were:

- Creating a database in oracle from the class diagram from UML.
- Developing end user interface of the database in developer 2000.
- Set up the JDBC driver and connection
- Developing the static structure of the website.
- Developing RMI server for the communication channel over the Internet.
- Connect the Database to Java classes used by the Servlets and RMI server.
- Creating dynamic web pages using Java Servlets.

As each component developed, it was tested using dummy data to see if it is functioning the way it should be. From the class diagram there were altogether five classes, these are: Patient, appointment, medical staff, general staff and medical record. Each class represents a table in relation database. So the five tables are: Patient table, Appointment table, Medical staff table, General staff table and Medical record table. The attributes in the class diagram for each class becomes the attributes of that table. The aggregation between the classes represents the relationship between the tables.

3.1 Patient Table

This table will store all the required information about the patients in to the system. The NHS_No was used as the logging name and the password field will be used as the password for logging into the website. Each patient has their own unique NHS_No, so the primary key for this table is the NHS_No.

3.2 Medical Staff Table

This table stores the information about the medical staff. The medical staff is considered to be someone with the medical profession like doctor, nurse etc. Medical_Staff_Id is the unique identifier for each medical staff. Only the doctor will be able to use the net for communicating with the patients. The password should be no less then five digits. It should also contain mixture of numeric and alphanumeric characters, to ensure security.

3.3 General Staff Table

This table stores the information about the general staff. The general staff was considered to be

someone without medical profession like secretary. Medical_Staff_Id is the unique identifier for each medical staff.

3.4 Appointment Table

The appointment table was used for storing appointment details. This table allows the patient to query/insert/delete appointment at their own convince. This is the only table that patients have direct access to it through the website.

3.5 Medical Record Table

This tables holds each of the patient medical details. Each patient will have a medical record associated with them, and will be updated each time the patient visits the surgery.
This table is set up as the slave table and the patient table as the master table this is because each time a patient visits the doctor the medical record will be updated. The master slave tables allows the user to do complex queries and also maintains the data.
To Create and customise the layout of tables, and add additional functionality to ease the browsing and make it as user friendly as possible, Developer 2000 was used. There is a high level of navigation between tables in order to make the usability easier.

4 Main menu

Figure 3, below shows the main menu that the medical staff and general staff will be using. Different colour scheme is used to make the staff aware of which table they are about to browse also the tables are in alphabetical order of name, which will make it easier to find the tables. Each of the options uses a single click push button except from the exit button, which requires double click, this is to ensure that the form isn't terminated accidentally. When the application is on running mode, the very first thing will appear is the main menu. From the menu depending on the need, one of the options will be selected.

5 Oracle Database Connectivity Using JDBC

Since the oracle database is running in Unix environment, ODBC (Open Database Connectivity) developed by Microsoft for database connectivity cannot be used because it is platform dependent, it only works on windows platform.
The other alternative is to use JDBC (Java Database Connectivity) developed by Sun Microsystem. JDBC™ technology is an API that lets you access virtually any tabular data source from the Java™ programming language.
It provides cross-DBMS connectivity to a wide range of SQL databases, and now, with the new JDBC API, it also provides access to other tabular data sources, such as spreadsheets or flat files.
The JDBC API allows developers to take advantage of the Java platform's "Write Once, Run Any where" and capabilities for industrial strength, cross-platform applications that require access to enterprise data. With a JDBC technology-enabled driver, a developer can easily connect all corporate data even in a heterogeneous environment, Java™ product site, 2001.
JDBC, like most driver specifications, is low-level and functional. The basics of the functionality are contained within the *java.sql.** classes. The basic JDBC driver is loaded with a *java.sql.DriverManager* class. Once you have a loaded driver it is possible to open a connection using the *java.sql.Connection* class. The connection is perhaps the most vital part of the process, as the rest of the procedure centres around the use of the connection object. After the connection is created, most clients will begin to transmit queries. Any return values from queries are stored in a *java.sql.ResultSet object*. JDBC also allows stored procedures to be called using the *java.sql.Statement*, *java.sql.PreparedStatement* and *java.sql.CallableStatement* classes.
Two pieces of information are needed in order to create a connection to the database, these are: Name of the JDBC package driver class and the URL that specifies which server and the database to use. The driver allows the connection between the Java application and the database. API communicates with the JDBC manager driver API, sending it various SQL statements. The manager should (transparently to the programmer) communicate with the various third-party drivers that actually connect to the database and return the information from the query or perform the action specified by the query.

6 WebSite Development

The website is the end interface of the system from the patient point of view. This is where the patient

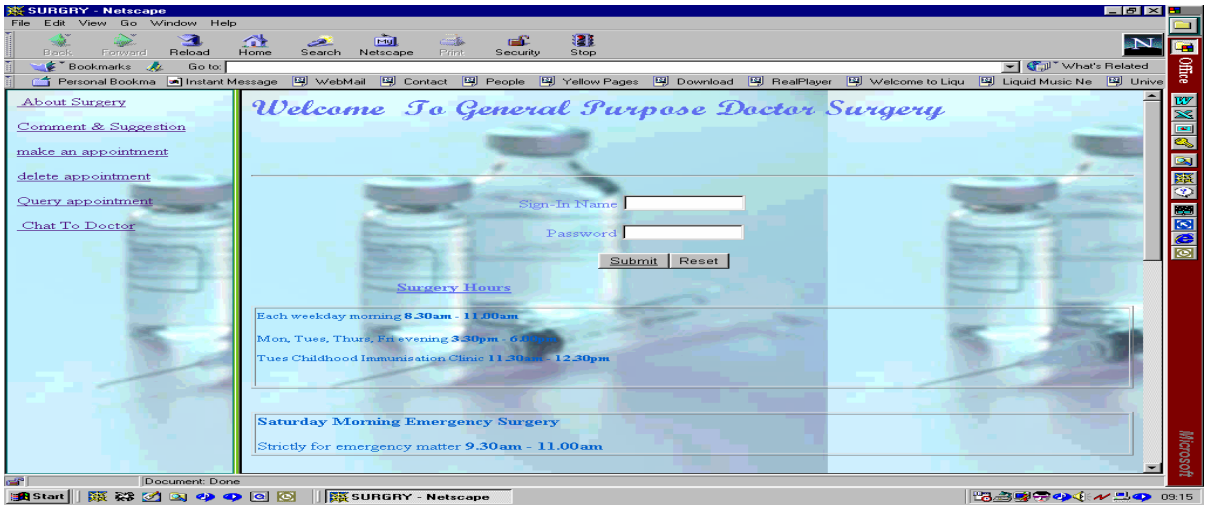

Figure 1, The website

and other users will interact with the system. It consists of several components.

The web site needs to generate dynamic web pages. Dynamic pages are customised page depending on the request different result will be displayed at different times. Simple dynamic content can be displayed using Java script such as the validations. But for creating a complete dynamic page, this is done by the server side using server side scripting. Java Servlets was used for this purpose Figure 1 shows the website that the patients will interact with the system. This site combines all of the components together.

7 Conclusions

The primary aim was to explore the concepts of distributed system and e-commerce, and the most popular programming language "JAVA", all this were combine to create a computerised system from the existing system, which will allow multiple transaction simultaneously. The main objectives of this prototype is to derive a system that will serve the following:

- Efficient and reliable service to the patient and the staff.
- Minimise the unnecessary transaction made by the staff where possible.
- Minimise the user error where possible.
- All the patients to interact with the system through a website, allowing them to book, create cancel appointment and communicate with the doctor.

Although this prototype satisfies all the basic need for the surgery, there are always spaces to improve and add extra features, which will make the system more robust.

References

NHS Information Authority web cited Sep. 2001, URL: http://www.nhsia.nhs.uk/text

NHS direct web cited Sept. 2001, URL: http://www.nhsdirect.nhs.uk

ERDIP Demonstrator Awareness Forum, (31st Oct. 2000), NHS Information Authority, Coventry, UK.

Borgida, A., Greenspan, S. Mylopoulos, J., (1985), Knowledge representation as the basis for requirements specifications, IEEE computer, 18,4:82-91.

Booch, G, Rumbaugh, J., Jacobson, I., (1998), The Unified Modelling Language User Guide, Addison-Wesley, Reading, MA.

Java™ Security web cited November 2001, URL:http://java.sun.com/security/

Java™ product web cited November, 2001URL:http://java.sun.com/products/jdbc/

Martin, J., Odell, J.J., (1992) Object-oriented analysis and design, Prentice-Hall, eEnglewood Cliffs, NJ.

Monarchi, D.E., Purh, G.I. (1992), A research typology for object-oriented analysis and design, Communications of the ACM35, 9:35-47.

McGinnes, S. (1992) How objective is object-oriented analysis?, Proceedings of CaiSE, Manchester, UK, 1-16.

Proctor, M.D., Creech, G.S., (Sept. 2001) object-oriented modelling of patients in a medical federation, IEEE Transactions on Information Technology in Biomedicine: a publication of the IEEE Engineering in medicine and Biology Society, 5,3:244-247.