Transparent Public Access to Medical Services

Bartosz Kwolek, Dominik Radziszowski, Pawel Rzepa

AGH University of Science and Technology Department of Computer Science
Al. Mickiewicza 30, 30-051 Kraków
{bkvolek, dr, rzepa}@agh.edu.pl.

Abstract. The paper describes disadvantages and limitations of common Polish medical computer systems both from technical and end-user point of view. It presents the public transparent access to medical services as a solution for all postulated expectations. Besides legal and ethic problems, technical ones are described in details. State-of-the-art technology for web-enable public access is presented and CAS (Clinical Appointment System) case study is discussed. The paper ends with conclusions.

Introduction

Modern hospitals use computer systems to access patient's medical record or information about hospital resources. However most of the existing medical systems expose their interfaces only to hospital staff, but usually not to wide public. Therefore the detailed information about the services (availability, preconditions, awaiting time) is not widespread and assistance of specialized medical staff is often mandatory. The progress in Internet technologies, including web-services, allows making almost the whole functionality available for wide public not only preserving it, but even extending it and, what is most important in medical systems – maintaining the security. Modern technologies permit to build secure systems easy to get from anywhere, anything and anytime.

Contemporary Clinical Systems

Modern hospitals use computer systems to access patient's medical record or information about hospital resources. These systems usually contain different kind of information. Mostly, it is personal patient data, patient contact data and data concerning patient's stay in a point of care i.e. hospitalisations as well as medical consultation and procedures carried out. More complex systems contains also information about medicines applied during hospitalisation accompanied by temporal and quantitative characteristic. Additionally they are supplied with very complex, integrated accounting modules including typical financial, warehouse and other specific modules e.g. Public Insurance Accounting module.

High complexity of those systems, high module dependency cause that technological innovations appears in this area relatively late and that most of Polish points of care system architectures are based on out-of-date, no scalable client-server paradigm. Common solution uses one central data base server and hundreds of work stations equipped with client application (so called fat client) that connect to the server.

Accessibility of Medical Services

Most of the existing medical systems expose their interfaces only to hospital staff, but usually not to wide public. Therefore the detailed information about the services (availability, preconditions, awaiting time) is not widespread and assistance of specialized medical staff is often mandatory.

Problem of making some information available to outside hospital users is very complex. Security of this legacy systems, unauthorized access control in particular, is mainly based on total access blockade to the systems for the outside world (firewalls, non routable IP addresses etc.). An attempt to allow access to a part of the data for wide group of external users within the confines of existing systems cause a huge risk of security reduction in access to the data, that are required to remain confidential. Hence we can safely say that common medical systems can not be accessed by the public.

Doesn't information society of the beginning of XXI century deserve public transparent access to medical services?

Public Transparent Access To Medical Services

Expectations concerning public transparent access to medical services are very wide. Public access means not only access for patients of a point of care, it should also cover potential patients and external physicians (i.e. GP doctors). Such circumstances bring lots of ambiguity and decrease medical service confidence level. Public Transparent Access is an access to information about medical services, equipment, hospital personnel and examination results, first and foremost is a method for booking a medical service selected from all served by hospital or a consultation with preferred medical specialist chosen form all working at hospital. Additionally it requires setting the clear rules in access to a short supply services (e.g. transplant). Creation of various standards of services access makes the procedures complex and unclear. This leads to patients confusion and necessity of clarifying rules arises. In this context transparent and public access to medical services begins to play more and more important role.

Many sociological, ethic, technical and legal issues have to be overcome to bring it to a proper level. In authors' opinion the current solutions give the opportunity to rise above technical limits, when others still remain unsolved.

Another problem is an appropriate information selection and its classification to the right access group – public accessible, for stuff only, confidential etc. Here hence the organisational and legal problems are more difficult to overcome then technical ones.

Technologies

Technologies and architectures used for constructing existing systems prevents from creating solutions that live up to all described expectations. However there are technologies that allow making these systems functionality transparently available for wide public.

Nowadays public access means access through an internet browser what impose use techniques and architectures proper for web technologies. Using internet browser as a client application introduces another facts:

? systems must be multi-layered to let thin clients take advantage of functionality accessible on business logic servers

? systems should be accessible through various end-devices including laptops, PDA-s and cellular phones.

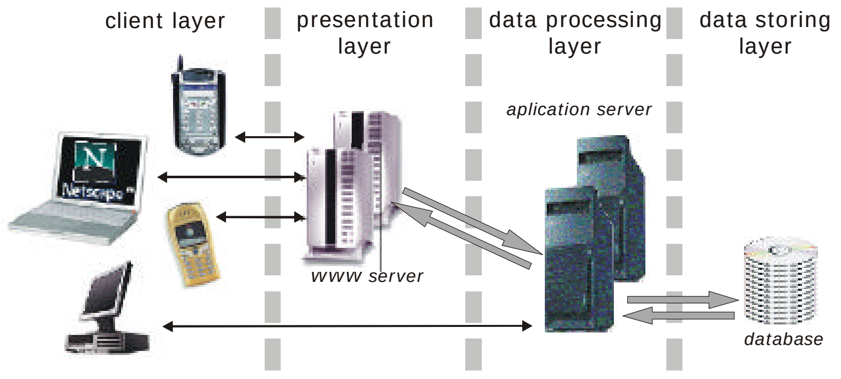


Fig. 1. Multi-layered architecture of web-based systems

One of the most advanced technologies used in web environments is Sun’s J2EE [1] architecture. This is an open standard that defines technologies and APIs used for creating reliable, enterprise-scale applications.

The main component of J2EE architecture is an application server that bases on EJB (Enterprise JavaBeans) technology which is a component model dedicated for server layer. The application server creates an infrastructure that makes services like security and transactionality available for business logic components called Enterprise JavaBeans. The application server covers also another issues like scalability, reliability, load balancing or optimised access to data storing layers [4].

Security policy, the way that reliability is ensured or transactions performed are defined in XML-based deployment descriptors what makes applications written in J2EE compatible way easily reconfigured and adopted to different requirements.

Enterprise JavaBeans components can be accessible through Java RMI invocations. The assumption stating that system functionality should be presented to end-users through internet browsers creates the need for mechanisms capable of transforming standard HTTP get and post methods into RMI calls. J2EE infrastructure defines Java Servlet [8, 9] and JavaServer Pages (JSP) [6,7] technologies in order to cover this area.

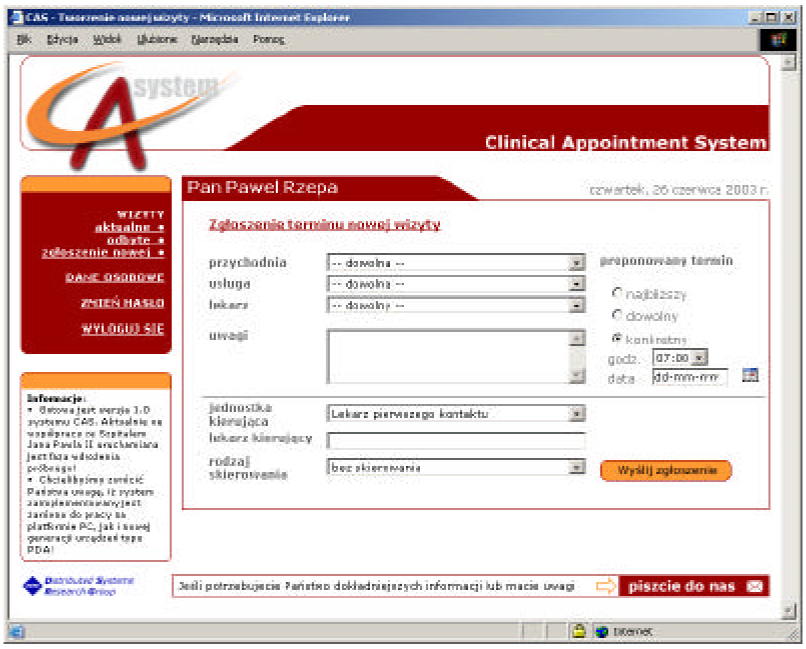
Java Servlets are server-side components that provides Web developers simple, consistent mechanism for expanding the web servers functionality and for accessing existing business systems. Servlets are able to catch HTTP requests, process them e.g. leveraging functionality implemented in Enterprise JavaBeans components, and return a proper response as an HTML, XML or other document. JavaServer Pages technology is an extension of Java Servlet technology that reorganizes the way servlets are implemented by programmers. It separates the user interface from content generation enabling designers to change the overall page layout without altering the underlying dynamic content.

Systems that assume access from diverse end-devices prefers XML as a document format because it describes document structure rather than the way document is displayed, what is characteristic for HTML. To present XML documents they must be transformed into a form proper for an actual display using e.g. XSLT transformation [11]. J2EE architecture is fully compatible with this approach.

CAS

In order to validate the usability of J2EE architecture in the area of creating web-based systems that transparently expose their functionality for wide public Clinical Appointment System (CAS) has been developed. The works has been carried out by Dept. of Computer Science, AGH University of Science and Technology within 6WINIT grant.

CAS is an application that gives patients possibility to make an appointment with a doctor for specified examination in a selected clinic using web browser. The system works as following: patients issue requests specifying a type of an examination, a doctor, a clinic and the time when they want the examination to occur. Requests are served by appointments clerks who validate patients requirements and approve the requests, modify or refuse them in accordance with doctor’s timetables. If a clerk modifies appointment’s prerequisites the patient is able to accept or rearrange them to suit his needs. This dialog last until either sides of conversation are satisfied.


CAS - Tworzenie nowej wizyty - Microsoft Internet Explorer

Plik Edycja Widok Ulubione Narzędzia Pomoc

Clinical Appointment System

WIZYTY
aktualne
odbyte
zgłoszenie nowej
DANE OSOBOWE
ZMIEŃ HASŁO
WYLOGUJ SIĘ

Informacje:
- Gotowa jest wersja 1.0 systemu CAS. Aktualnie we współpracy ze Szpitalem Jana Pawła II uruchamiana jest faza wdrożenia próbnego!
- Chcielibyśmy zwrócić Państwa uwagę, iż system zaimplementowany jest zarówno do pracy na platformie PC, jak i nowej generacji urządzeń typu PDA!

Pan Paweł Rzepa

czwartek, 26 czerwca 2003 r.

Zgłoszenie terminu nowej wizyty

przychodnia	-- dowolna --
usługa	-- dowolna --
lekarz	-- dowolny --
uwagi	
jednostka kierująca	Lekarz pierwszego kontaktu
lekarz kierujący	
rodzaj skierowania	bez skierowania

proponowany termin
( ) najbliższy
( ) dowolny
(•) konkretny
godz. 07:00
data dd-mm-rrrr

Wyślij zgłoszenie

Distributed Systems Research Group

Jeśli potrzebujecie Państwo dokładniejszych informacji lub macie uwagi piszcie do nas

Internet

Fig. 2. Creation of a new appointment

Application distinguishes three separate group of users that access its functionality. Besides patients and appointment clerks described previously there are administrators who handle all issues concerning managing the system e.g. adding clinics, doctors, appointment clerks etc.

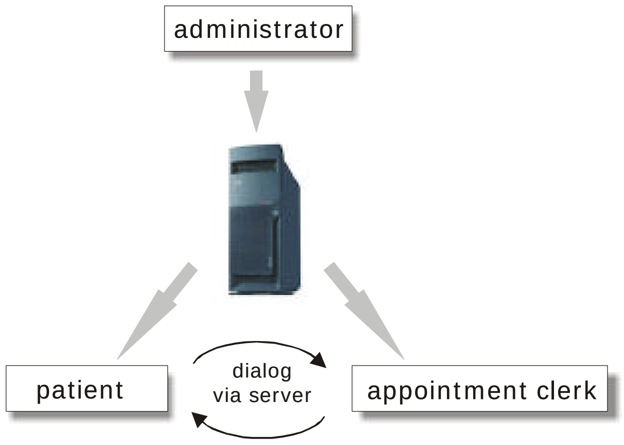


Fig. 3. Interaction of groups of CAS users with the system

The mentioned basic functionality is accompanied by several extensions:

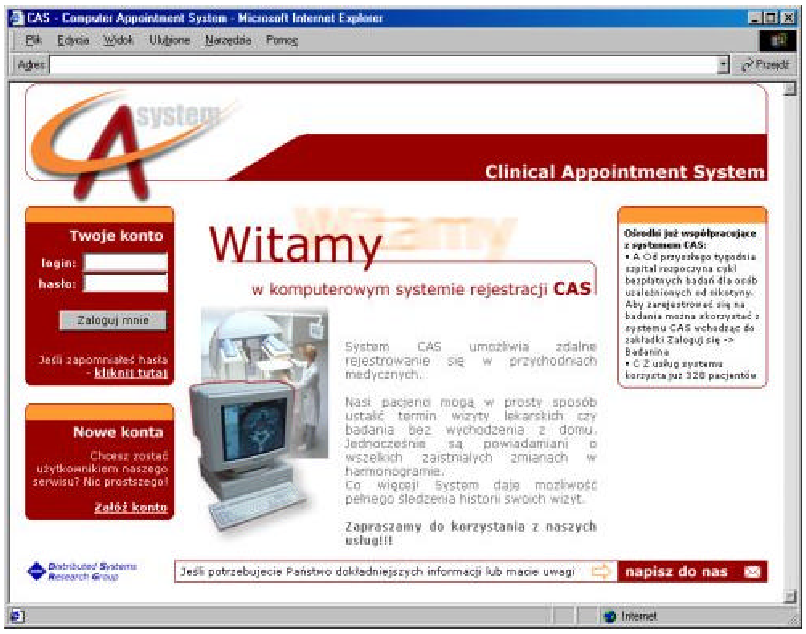
? patients can review the schedule of their visits
? patients can review the information about completed visits
? patients can review their personal data
? appointment clerks are able to create new visits in behalf of patient
? appointment clerks validate and correct patients personal data

One of the important characteristics of the system is to give appointment clerks possibility to act as patient. In this way the system can be used in a traditional way when a patient calls clerks or makes an appointment at appointment desk. This feature smooth the progress towards using web-based functionality.

The user interface of CAS has been designed to make use of the system from PCs and PDAs. Other client devices can access the system as well, but the overall page layout wouldn't look as good as it does on displays of these two types end machines.

Fig. 4. CAS on a PDA display

CAS functionality is accessible only for authenticated users. Every system client is equipped with login and password and must use it to make use of the system. Authorization mechanisms allows every patient to use only this part of the sensitive information gathered by the system that concerns him. Appointment clerks can read all the information connected with appointments and personal data. To prevent data exchanged between web browser and web server from being vulnerable to malicious users attacks secured HTTP protocol is used.


CAS - Computer Appointment System - Microsoft Internet Explorer
Plik Edycja Widok Ulubione Narzędzia Pomoc
Adres
Przejdź
system
Clinical Appointment System
Twoje konto
login:
hasło:
Zaloguj mnie
Nowe konta
Załóż konto
napisz do nas
Distributed Systems Research Group
Internet

Witamy

w komputerowym systemie rejestracji CAS

System CAS umożliwia zdalne rejestrowanie się w przychodniach medycznych.

Nasi pacjenci mogą w prosty sposób ustalić termin wizyty lekarskich czy badania bez wychodzenia z domu. Jednocześnie są powiadamiani o wszelkich zaistniałych zmianach w harmonogramie.
Co więcej! System daje możliwość pełnego śledzenia histori swoich wizyt.

Zapraszamy do korzystania z naszych usług!!!

Jeśli zapomniałeś hasła - **kliknij tutaj**

Chcesz zostać użytkownikiem naszego serwisu? Nic prostszego!

Ośrodki już współpracujące z systemem CAS:
• A Od przyszłego tygodnia szpital rozpoczyna cykl bezpłatnych badań dla osób uzależnionych od nikotyny. Aby zarejestrować się na badania można skorzystać z systemu CAS wchodząc do zakładki Zaloguj się -> Badania
• C Z usług systemu korzysta już 320 pacjentów

Jeśli potrzebujecie Państwo dokładniejszych informacji lub macie uwagi

Fig. 5. CAS on a PC display

CAS was deployed in a John Paul II hospital in Cracow in order to validate its usability and usefulness.

Conclusions

CAS system proved that there is no technical limits to create transparent public access to medical services. The pilot deployment revealed significant demand from both patient and medical staff for this kind of systems. Unfortunately current law regulations effectively prevent from progressing toward systems fulfilling goals pointed out by the authors.

References

1. http://java.sun.com/j2ee
2. Sridharn P.: Java Beans – developer’s resources. Prentice Hall PTR
3. Roman E.: Mastering Enterprise JavaBeans and the Java 2 Enterprise Edition. Wrox Press Ltd, 2001
4. Monson-Haefel R.: Enterprise Java Bean. O’Reilly, 1999
5. Sun Microsystems: Core Java. Sun Microsystems Press 1999
6. Bergsten H.: Java Server Pages. O’Reilly, 2000
7. http://java.sun.com/products/jsp

8. http://java.sun.com/products/servlet
9. Hunter J.: Java Servlet Programming, 2nd Edition. O'Reilly, 2001
10. Husted T.: Struts in Action. Manning Publications Co. 2003
11. XML: http://www.xml.com, http://www.xml.org, http://www.xml.pl
12. Radziszowski D., Rzepa P.: Enterprise Java Beans. TelenetForum, April 2002