ADAPTATION OF AN ELECTRONIC MEDICAL RECORD SYSTEM FOR REMOTE MONITORING APPLICATIONS

Sonny Hernandez
Computer Engineering and Computer Science
University of Southern California
e-mail: sonnyher@usc.edu

J. Mikael Eklund
Faculty of Engineering and Applied Science
University of Ontario Institute of Technology
e-mail: mikael.eklund@uoit.ca

Arsalan Tavakoli
Department of Electrical Engineering and Computer Sciences
University of California, Berkeley
e-mail: arsalan@eecs.berkeley.edu

Ruzena Bajcsy
Department of Electrical Engineering and Computer Sciences
University of California, Berkeley
e-mail: bajcsy@eecs.berkeley.edu

Shankar Sastry
Department of Electrical Engineering and Computer Sciences
University of California, Berkeley
e-mail: sastry@eecs.berkeley.edu

Abstract

The Electronic Medical Records system (EMR) is a method of storing information kept in a traditional medical record in an electronic database for distributed access to medical professionals. This paper focuses on how an EMR system can be designed to allow for the integration of an assisted living at home system and which together can allow for the automated data entry into the EMR from home medical sensors. This paper looks at the system architecture and implementation details of such an integrated system. An electronic medical record has numerous advantages over the traditional medical record, such as integrating dynamic environmental data into the system, and a customizable interface on both the client and doctors end, outlined in this paper. The system is designed on a existing set of components from the open source project OpenEMed and is implemented as a distributed system written in Java. This paper presents the motivation for the project and it concludes with a discussion of the impact of such a project and future directions.

1 Introduction

With the aging baby-boomer generation, providing health care for such a large part of the population is going to be a very real and difficult problem. With the increase of the elderly population, health care costs will rise and put further strain on our health care system. The development of information technology for assisted living at home are important as they will help alleviate some of this strain and are in current progress.

The electronic medical record (EMR) system can be is a critical part of the assisted living at home by allowing the medical data collected at home to be made available to medical professionals in an asynchronous and archival manner. The ITALH project incorporates camera, sensor and accelerometer data to detect falls and the EMR system is where it all comes together. The EMR system serves as a storage area for necessary information such as movement patterns, medication status, and past medical information. The key differentiating factor of this system is the fact that it monitors dynamic data from the sensors at the users home. This fills in the gap for doctors whom once could only rely on data gathered from an occasional visit, where with this system, they can have up-to-date data on the patients status to provide better care and a more accurate diagnosis. This would also help ease the strain on the health care system as the users system will be able to analyze sensor readings and notify doctors of potential problems, thus saving the user trips to the doctor and relieving some of the strain on the doctors. The system will analyze sensor readings at the users end to find possible problems a patient may be experiencing and can trigger alerts to notify the doctor.

By having these live-readings coupled with past medical history, everything a doctor needs to monitor the patients status will be available through an intuitive interface, which can increase efficiency and let the doctor focus be on helping the patient instead of looking for records. The doctor will also be able to customize this interface to decide what conditions will trigger an alert and they will be able to query datasets from specified time spans. The distributed nature of this system though, raises some implications that cannot be overlooked, the biggest being security. To be practical, the sensors must be wireless and the sensor readings must be transmitted to the system without compromising the integrity of the data and must also prevent eavesdropping. The data, once recorded in the database, must also be made secure so only authorized users can access the sensor data and patient history. Despite the security issues, the successful implementation of such as system will have many positive repercussions, as the system gives elderly people an alternate to being in a retirement home: it gives them the option to stay at home and maintain their independence while still providing doctors with the data they need to monitor the status of their well-being.
2 Method

2.1 System Analysis

This project builds on the open source EMR system OpenEMed [4], which is quite large and has many dependencies and consisted of many (4500+) source files. Examining this legacy system, which provided a good example of other legacy systems, including a lack of adequate documentation, took a considerable amount of time to determine the source files dependencies, etc. This was a vital and necessary step because without a good knowledge of the source files, it would be impossible to modify and build the EMR system. This process required several few weeks of thorough analysis, but resulted in a good knowledge of the integral components were and where the modifications were to be carried out.

2.2 Design

Upon determining what the overall system was doing and how each of the components contributed to our project, we had to lay out a design and lay out a plan for how and where the system required changes. This required laying out specifications that gave an outline of the requirements which we had to set forth and how we planned to implement and carry out these changes.

2.2.1 Architecture

With the system analysis done, we gathered that the system was built using Java and utilized HSQL databases for data storage. We were also now prepared to decide where software changes needed to be made. Most of these changes were made in the Personal Identification Service (outlined below). This was where the medical record was to be added as all other pertinent data regarding the patient was located here as well. The other changes regarding sensor data were likely to have been added into the Clinical Observation Access Service, although it was not accomplished during this project.

2.2.2 Requirements

The requirements of the EMR system design were as follows:

- Security was of utmost importance. In a distributed system, it is easier for data to get in the wrong hands so access control should be enforced and medical records should only be modifiable by authorized users and should be viewable by the patient.
- EMR should contain all relevant information regarding the patient, such as medication information, medical history, and relevant sensor readings.
- The doctor should be able to submit queries to obtain information such as the medical record, sensor readings, or whatever other data is necessary.
- The interfaces should be customizable to some degree, such as allowing doctors to set the conditions to trigger alerts.
- The system should provide some patient/doctor interactions such as allowing the doctor to sync up patient data or being able to update the medical information for the patient.
- The system should be scalable so it can be expanded as the amount of patients increase yet still remain functionality.
- Fault tolerance and accuracy is very important. The system must minimize errors to be dependable, otherwise alerts may go unnoticed or may be triggered too often when not needed.

2.3 OpenEMed

OpenEMed serves as the foundation for the EMR system. It is an open-source project containing numerous services and applications built for utilization in the health care industry. Among the critical ones for this project are the Personal Identification System (PIDS), the Resource Access Decision Service (RADS), and the Clinical Observation Access Service (COAS).
3.5 Modifications

After the system was running correctly and we had gained an in-depth knowledge of the system, then the system could be modified to meet the specifications that we needed. Most of the files that needed to be modified were identified in the design document period but there were a few exceptions which caused the process to be longer than anticipated, as it required spending more time going through the source files and attempting to see where else change was necessary. There are also interdependencies between files so if one line changed in one file, that meant many lines could have changed in another group of source files. Along with the modifications came the inevitable errors. Although the files would often compile without a problem, there were runtime errors that caused a lot of trouble which had to be fixed.

4 Results

I have finalized a design map of sorts for the system so although we did not finish the system entirely, whomever picks up where we left off will be able to follow a plan to complete the system, as it will lay out what modifications need to be made and mention where they can be made. This will assist any future researchers to focus on the development of the software instead of trying to figure out what the system does by sifting through the numerous source files and where to make modifications. The imperative change that was made to the PIDS client was that it now contains a functional saving/retrieving medical record area that is associated with the patient entries. Whenever one looks up a patient, the system brings up their contact and personal information, as well as the newly added medical record (See Figure 6).

2.3.1 PIDS

The Personal Identification Service is currently the main interface that the doctor would use to look up, maintain, and edit their patient data. The interface is fairly basic and allows queries by any one of many fields, among them: first and last name, address, patient identification number, date of birth, etc. This component initially had only fields to keep contact information and allowed for storage and retrieval of patient contact information.

2.3.2 RADS

The Resource Access Decision Service is an OpenEMed component for changing the permissions that the users have when accessing the various other services. The service utilizes a simple interface where the users names are kept and their permissions can be changed to read, edit, add, or other. The RADS service is not connected to any particular service at the moment, so although permissions can be changed, it does not affect the other components yet.

2.3.3 COAS

The Clinical Observation Access Service allows for queries to be carried out based on clinical data. This is a more in-depth service that allows you to query specific record types, such as immunology, and find a users immunology record. While quite useful, this component is stand-alone and not interconnected with PIDS or RADS.

2.4 Running Components

Even after we had a better idea of the functions of the various components, getting them to run properly was an equally challenging task. As previously mentioned, the software was poorly documented so a lot of the errors were not mentioned, and we had to do a lot of troubleshooting. A majority of the problems came from the PIDS client which was supposed to generate a file for the server but was not, and therefore, we would get an error message that it could not connect to the server as seen in Fig. 4.
3.3.2 RADS

The Resource Access Decision Service is an OpenEMed component for changing the permissions that the users have when accessing the various other services. The service utilizes a simple interface where the user’s names are kept and their permissions can be changed to read, edit, add, or other. The RADS service is not connected to any particular service at the moment, so although permissions can be changed, it does not affect the other components yet.

Figure 2. The RADS user interface.

3.3.3 COAS

The Clinical Observation Access Service allows for queries to be carried out based on clinical data. This is a more in-depth service that allows you to query specific record types, such as immunology, and find a user’s immunology record. While quite useful, this component is stand-alone and not interconnected with PIDS or RADS.

Figure 3. The COAS user interface.

3.4 Running Components

Even after we had a better idea of the functions of the various components, getting them to run properly was an equally challenging task. As previously mentioned, the software was poorly documented so a lot of the errors were not mentioned, and we had to do a lot of troubleshooting. A majority of the problems came from the PIDS client which was supposed to generate a file for the server but was not, and therefore, we would get an error message that it could not connect to the server (See Figure 5).

Figure 4. The PIDS client running.

For large queries it is much more efficient to do a <Count> to see how many records would come back.

Figure 5. The COAS user interface.
3.3.2 RADS

The Resource Access Decision Service is an OpenEMed component for changing the permissions that the users have when accessing the various other services. The service utilizes a simple interface where the user's names are kept and their permissions can be changed to read, edit, add, or other. The RADS service is not connected to any particular service at the moment, so although permissions can be changed, it does not affect the other components yet.

4 Conclusions and Future Work

As EMR systems are being developed and entering into practice, there are parallel advances being made in telehealth applications such as assisted living at home. Many potential benefits of this information technology will not be achieved unless these two streams of development are integrated. In this paper we have presented our initial work at modifying and experimental EMR system to allow for this integration. This can serve as model for integration of such functionality into legacy systems, and for the design of newer systems.

The development of such a system is no simple task but it is one that can be greatly successful when implemented correctly. Several other considerations must enter into the design process, such as security, but these are already partially addressed in the
Conclusions and Future Work

Through my research and time working on the system, I have come to realize the need for the EMR system. The development of such a system is no simple task but it is one that can be greatly successful when implemented correctly. The work I have done serves as a base for the remainder of the project. Security is still an imperative factor, and although secure servers are part of OpenEMed, I was not able to get them working properly. Testing the system for robustness and how it holds up in a distributed environment is important as well.

I was not able to set up a computer to be the server with many other client computers accessing it so the testing and assurance that it will hold up is very important, as it is a distributed system that will have server with many clients accessing them. The medical record that was added, while functional, is rather simple and can be expanded further by splitting it up into sections or adding different fields. Last, the interconnectivity of the various components can be further explored to see how one can be applied to the other, the development of an overhead interface to access all the various services from one place would make accessing them simpler.

Acknowledgements

This research was conducted as part of the Summer Undergraduate Program for Engineering Research and Berkeley (SUPERB) and supported by the Team for Research in Ubiquitous Secure Technology (TRUST), a National Science Foundation Science and Technology Center.

References


