“e-Arogya” – An e-Health Solution for Sri Lankan Government Hospitals

G. S. Niroshan, K. G. Lanerolle, L. C. T. Silva, W. M. U. S. Wijekoon, S. Weerawarane
Department of Computer Science and Engineering,
University of Moratuwa
Moratuwa, Sri Lanka

Abstract— **Sri Lanka is one of the few countries that provide free health services. But majority of hospitals in Sri Lanka still use the traditional paper based methods when registering patients and other administrative functions. The efficiency of the hospitals has decreased largely due to this. “e-Arogya” is introduced as a solution to this problem. This paper explores the possibilities achievable for the introduction of ICT in Sri Lankan healthcare system, problems faced when achieving that, the proposed solution with the use of open source software, and open standards in healthcare, with a new user interface design that follows Human Computer Interaction (HCI) principles.**

Keywords – Healthcare, HL7, SOA,

I. INTRODUCTION

A healthy population increases the productivity of a country by great proportion. Therefore, any country provides an effective healthcare system to their citizens. Some countries like Sri Lanka, provides free healthcare services at government hospitals.

At present, e-Health is a leading topic in the ICT industry. It aims to deliver efficient healthcare services to the community. This new technology is commenced by the ICT and healthcare institutions. Now e-Health solutions for healthcare institutions are available worldwide. Unfortunately the acquisition cost of such solutions is not affordable for developing countries like Sri Lanka, and trying to overcome this financial barrier continues to be a challenge. Furthermore, most such solutions are not tailored for the specific needs of healthcare institutions in developing countries, which themselves, have evolved towards providing affordable healthcare with highly limited resources in challenging environments.

This research is to provide an e – health solution for Sri Lankan healthcare service sector with concept of paperless hospital [1]. A clinical unit – the Emergency Treatment Unit (ETU) of the Colombo South Teaching Hospital (CSTH) - which is one of the largest teaching hospitals in Sri Lanka was chosen based on the research conducted by Dr. M.N. Karunathilake on Business Process Re-Engineering [2] for the Emergency Treatment Unit at the Colombo South Teaching Hospital. As the first step we introduce a system to the ETU [3].

II. CURRENT SITUATION

Most of the activities in Sri Lankan government hospitals are still using the traditional manual methods based on paper work. These methods lead to decrease the efficiency in a hospital in a considerable manner. Patients have to wait for a long time to take medicine from government hospitals. It also leads to wastage of resources that can be used for the benefit of the society. For example, lots of medicines are thrown away from the CSTH due to exceeding of the expiry date. This happens because the hospital orders more than the required medicine in order to prevent shortage. This can be easily prevented if an efficient system can be used to calculate the amount of medicine that should be ordered for a given period of time. Even patient registrations have to be done several times if he/she has been transferred to a different ward because previous records are not reused [3].

In developed countries a paperless hospital is a well accepted concept which incorporates new technologies aimed at improving efficient delivery of healthcare. Therefore we provide the first step towards the concept of paperless hospitals in Sri Lanka, by proposing the e-Health solution “e – Arogya”.

III. SOLUTION

The “e – Arogya” project is to develop a software system which adheres to software engineering principals such as usability, extendibility, and maintainability. Also this system is to be developed such that it adheres to the standards in the e-health domain, thus communicating with existing e-health systems will be standardized.

Furthermore, the final system developed for the ETU should be capable of making the functions of ETU more efficient than the existing paper based system. Since the main objective is to re-engineer the current process in the hospital, the design should ensure that the system is well accepted by the users in the hospital. Users should be able to use the system with minimum technical knowledge and with minimum knowledge of laws and regulations. All the required legal restrictions must be handled within the system. For example illegal access to the data and functions in the system should be prevented by managing a proper authentication system.

We mainly focus on achieving the following targets in relation to the ETU as detailed in [3].

- To provide fast patient care services.
- To develop an easy patient tracking system.
- To manage the medical records with efficient retrieving and single registration number.
- To eliminate unnecessary wastage of drugs, surgical items and meals.

A. An Open Source Solution

Since Sri Lanka is a developing country and the health service is provided for free, the government cannot invest a large budget on developing the system. Therefore developing the system has to be of low cost. Free and Open Source Software (FOSS) matches perfectly with these requirements.

B. Ripple Effect

Even though the system in developed for the ETU, it is designed such that it can be expanded to other departments and then to other hospitals. This expansion can be compared to a ripple effect on a water surface. We initiate the ripple by introducing "e-Arogya" to the ETU. "Ripple Effect" is a term used in software metrics as a complexity measure. It explains how software can be extended without adding any unnecessary complexity to the system [4].

C. A SOA based Solution

There are many other systems in Sri Lanka which are not in the e-Health domain, but can be connected to provide a better service to the society. "ePopulation Registry", a project currently under development by ICTA [5] is one such solution. This "e-Arogya" designed in such a way that it can be expanded as a ripple effect to other departments in the CSTH and to other government hospitals.

To design an SOA based solution, first information domains within the organization have to be identified. In the ETU of the CSTH the information domains are Electronic Health Records (EHR), Labs, Pharmacy, Ambulance service etc. Then for each of these information domains a web service is developed which only exposes the information that the domain is willing to expose. This results in many web services and independent solutions.

D. Enterprise Service Bus

Since each part of the e-health solution is developed as a web services to achieve interoperability, there should be a mechanism to connect these web services and allow them to communicate with each other. "An **Enterprise Service Bus** (ESB) [6, 7] is a software architecture construct which provides fundamental services for complex architectures via an event-driven and standards-based messaging engine (the bus)". An ESB plays a major role in the performance of a SOA [8, 9] solution. Therefore an ESB that matches the requirements of the system should be selected with care. There are many free and open source ESBs available. Some of the ESB that are considered as suitable for the system is mentioned below.

1) Mule ESB

This is a widely used ESB with 1.5 million downloads. This is a light weight ESB developed by Mule Soft and started in 2003. It runs on JVM but can broker interaction between other platforms like .Net using web services [10, 11].

2) Open ESB

This is java based ESB initially developed by SUN Microsystems. But after Oracle acquired SUN Microsystems they decided to cancel their sponsorship to the project. But still the community development of the project continues [7].

3) JBoss

JBoss is a FOSS developed by Red Hat. It is not only an ESB, but also an application server. JBoss is developed using Java-EE. This middleware runs on any Java VM so that it can be run on multiple platforms [12].

Considering all the features of the above mentioned ESBs and the main requirement of the system, which is that the system should be able to run even in a single machine and the support provided by the ESB for HL7 messaging standards, which is discussed below, Mule ESB we selected MuleESB for e-Arogya. Figure 2 illustrates the intended SOA architecture of the e-Arogya solution, featuring an ESB.

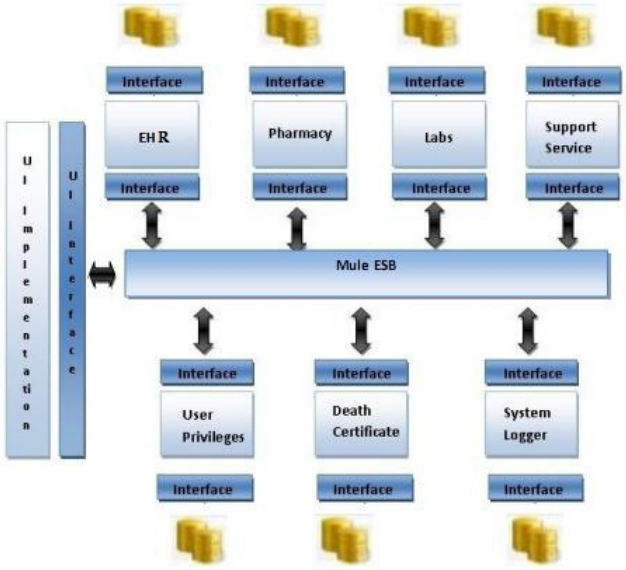

Figure 1. SOA Architecture of "e-Arogya"

E. Open Standards in e-Health Domain

In a single organization there can be many healthcare solutions that provide different functionalities. In most of the cases these solutions may be provided by different vendors. For an organization to deliver a more effective service these separate solutions should be able to communicate among each other. Since these solutions are provided by different vendors there should be a mutual agreement on how they should

communicate. Open standards give the solution for this problem. Open standards provide a format to communicate such that any solution can understand what is sent to it from another solution.

There are many e-Health related open standards widely used in the world. Among these open standards, the following e-Health standards are widely recognized.

1) HL7 (Health Level Seven Standards)

HL7 is a US-based organization which is involved in developing standards in the e-Health domain. These standards are approved by both ANSI and ISO and have been widely adopted by many countries in the world. For exchanging medical records between systems HL7 provides standard message structure which can be adapted to the needs of the organization [13].

2) OpenEHR

This is an open standard in health informatics that describes the management, storage, retrieval and exchange of health data in Electronics Health Records (EHR). This standard is not concerned with the exchange of data between other EHR systems which is the main focus of other systems such as HL7, EN 13 606 [14].

3) EN 13606

The Electronic Health Record Communication Standard is a European standard which defines information architecture to communicate part or all of the Electronic Health Records (EHR) of a patient. "OpenEHR" is the basis of this standard. The objective of this standard is to preserve the original meaning intended by the author and reflecting the confidentiality of data intended by the author or the patient [15].

4) ISO 18308

This is a standard provided by ISO/TC 215, the International Organization for Standardization's Technical Committee. This committee works on the standardization of Health Information and Communication Technology. This standard describes a pragmatic classification of the health records [14].

F. HL7 Standards

Unlike many standards, which mainly focus only on Electronic Health Records (EHR), HL7 focus on additional functionalities in the e-health domain such as pharmacies and labs. Therefore, a single standard can be used to connect all departments involved in providing health services. HL7 provides a standard way to communicate with all these systems [13].

HL7 has two versions of standards, HL7 V2.x and HL7 V3. HL7 V2.x only focuses on the internal hospital workflows and it only has a set of message specifications. This messaging standard provides interoperability among Electronic Patient Management Systems, Laboratory Information Systems, Dietary, Pharmacy and Billing Systems and Electronic Health Record Systems (EHR) or Electronic Medical Record Systems (EMR) [16, 17].

After identifying many issues in HL7 V2.x such as segments being reused in many messages and message definitions reused in many trigger events, many data fields were made optional in order to accommodate reuse; HL7 organization has introduced a new version called HL7 V3. This new version incorporates more trigger events and message formats than the previous version [18].

When choosing between the two versions many things were considered. HL7 V2.x has been used by many systems and is fairly stable. Also since the HL7 is involved in sending messages in a given format, these messages have to be parsed to extract the data. There are many open source parsers available for HL7 V2.x. Java HAPI, NHAPI, Nule Light HL7 are some of these parsers. Among these parsers Java HAPI is a stands out because it uses an object oriented structure where each trigger event, segment and data type are objects. This concept makes identifying data in a message as well as creating a new message fairly easy. In contrast HL7 V3 is fairly new and it has been subject to changes since the first release. In addition currently there are no proper parsers available for HL7 V3 standard messages. Therefore HL7 V2.x was selected as the messaging standard in the e-Arogya system.

Since HL7 is a globally accepted standard it contains many message formats and segments, which a not useful in the Sri Lankan context. HL7 has defined 317 message formats and 170 message segments. HL7 has provided with the flexibility to choose the segments in a message as needed. Since each country follows their own process in health service, many of the messages defined are not useful. As an example, the message DFT_P03 – Post Detail Financial Transaction, which carries billing details of a patient's visit, is not required since the government hospitals in Sri Lanka provide services free of charge. The segment FT1 carries the billing details of a patient and since the services are free of charge this segment can be ignored whenever it is found in a message format.

G. Mirth Connect

Mirth Connect is an ESB built to send HL7 messages. This is built using the Mule ESB and contains connectors to support web services, HTTP, FTP and SMTP protocols. Mirth Connect is a free and open source project. Within Mirth Connect the HL7 message received are validated so that messages which do not adhere to HL7 standards cannot be sent and an error message it sent back to the original system [19].

Mirth connect also provides message transformation from HL7 V2.x to HL7 V3 and vice versa. Thus adding new parts which support different HL7 versions to the system do not require any extra work. In addition Mirth Connect has an inbuilt logging mechanism which stores all the details about messages which are sent back and forth such as sending application, receiving application, whether the message was

successfully sent, and the error message if an error was caused while sending the message.

Installing and using Mirth Connect is fairly easy. It provides an all in one package which installs all the required parts at once. After installing a user can create a channel simply by entering the required information. Then the channel created can be deployed and it is ready to route any message received. For a person to use Mirth Connect they will only require a very small amount of training, so that high skilled IT professionals are not required to manage the system, saving a lot of expenses. Figure 2 illustrates the software architecture of the e-Arogya solution, featuring Mirth Connect.

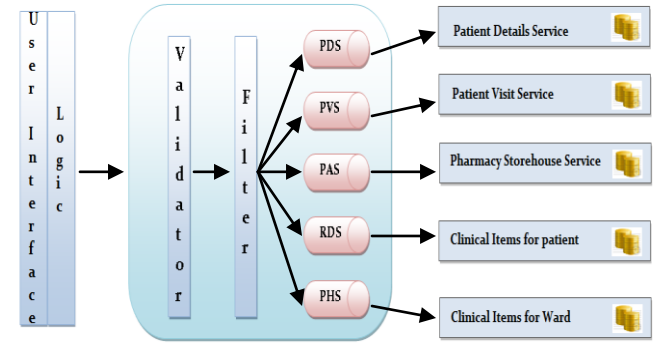

Figure 2. Software Architecture of "e-Arogya"

IV. SYSTEM FUNCTIONALITY AND IMPLEMENTATION

As shown in Figure 2. "e-Arogya" has three main parts. The front end, communication and the back end. Both the front end and back end uses a parser called HAPI parser which is used to create as well as to parse HL7 messages [20].

A. Front End

The front end of "e-Arogya" consists of the user interface and the logic that is necessary to create a message and send to mirth connect, which is the ESB. We designed the user interface such that it adheres to the user's domain of knowledge. For example, in Sri Lankan government hospitals the doctors, nurses in a given ward refer to a patient using the bed number. Therefore, we designed a user interface to display an overview of the ward, showing the occupied beds and unoccupied beds. Clicking on an occupied bed allows a user to view medical records, prescribed drugs and personal information such as name, date of birth etc. Figure 3 shows the user interface of the above mentioned ward view.

In order to achieve security we provide role based access. In role based access, the user is only displayed the user interfaces that are needed to perform the tasks a user is assigned. By this the user will not perform tasks he/she is not allowed to do.

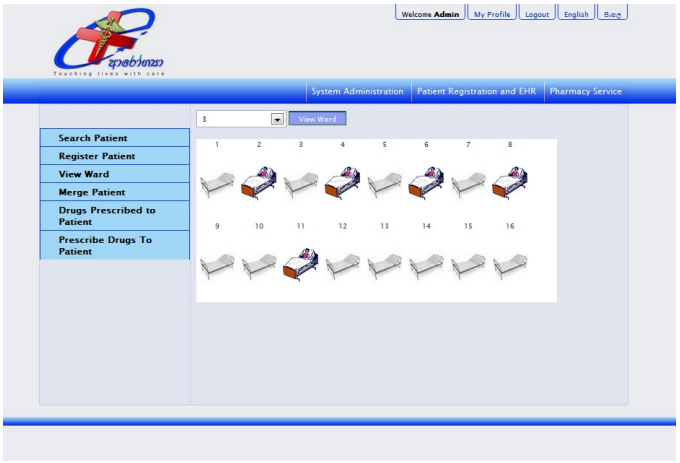

Figure 3. Ward View

B. Back End

Based on the information domains identified back end contains five web services. Each web service provides a specific functionality and maintains a separate database. This is because an information domain is the owner of its information and if that information is required, then it has to be requested from the information domain. If all information is stored in a single database, then this functionality cannot be achieved.

The functionalities of the five web services in the back end is described below.

1. Patient Details Service

This web service contains personal information of the patient such as name, date of birth, religion, address etc. In addition to storing this information, this web service provides functionality to register a patient, update patient details, search a patient, merge two patients etc.

These functionalities are handled using HL7 messages ADT A04, ADT A08, ADT A60 and QRY A19. Messages related to patient details and admission starts with ADT and QRY indicates querying information.

2. Patient Visit Service

This web service contains admission and medical information of a patient from the time a patient is admitted to the hospital until the patient is discharged such as admission date, admitting officer, admission reason, diagnosis, observations, discharge date etc. This web service provides functionalities such as admitting a patient add medical records, discharge etc.

These functionalities are handled using HL7 messages ADT A01, ADT A03, ADT A08, ADT A11, ADT A17, ADT A61and QRY A19

3. Pharmacy Storehouse Service

This web service contains information about the drugs stored in the pharmacy. Also it records information

about the drugs ordered to the pharmacy, and drugs issued by the pharmacy.

These functionalities are handled using HL7 messages ORM O01, RDE O11, RDS O13, RGV 015, EQQ Q04 and TBR R08. Messages related to pharmacy and drugs starts with ORM, RDE, RDS, RGV, EQQ and TBR.

4. Clinical Items for Patient Service

   This contains information about the drugs given to a patient. When a doctor prescribes drugs to a patient, a request is sent to this web service and it is recorded along with the date and the id of the person, who issued the drugs to the patient.

   The functionalities of this web service are handled by the same set of messages as in the Pharmacy Storehouse Service.

5. Clinical Items for Ward Service

   This contains information about the drugs ordered and maintained by a ward. The ward order drugs from the pharmacy and issue to patients.

   The functionality of this web service are handled by the same set of messages as in the Pharmacy Storehouse Service.

C. Comminucation

The communication part consists of Mirth Connect. A channel is created inside Mirth Connect to with web service listener as the source connector type and web service sender as the destination connector type. A single channel can have many destinations. Since the back end have five web services, the channel created contains five destinations.

Since all the messages are received by a single source, which is connected to the front end, filters are used to identify which message should be sent to which destination. An HL7 message contains a field in the header segment that contains the destination of the message. The messages are filtered based on this destination.

Also the reply coming from all destinations has to be sent back to the front end to inform the user the success or the failure of the function performed by the user. But Mirth Connect only allows reply from one destination to be sent to the front end via the web service listener (source) connector at a time. If the destination needed to be changed it has to be done manually. To overcome this issue we created another destination of the type “channel writer” and re-direct the reply messages from all the destinations to this new destination. The new destination is used to send the reply message to the front end via the source connector.

V. CONCLUSION AND FUTURE WORK

“e – Arogya”, is an e - Health solution for the ETU section at the Colombo South Teaching Hospital. It is developed leveraging FOSS with HL7 healthcare standards.

Based on the ripple effect, “e – Arogya” can be used in the other sections of the Colombo South Teaching Hospital and other government/private hospital of the Sri Lanka with minimum amount of changes.

REFERENCES

[1]. Harper R.H.R., O’Hara K.P.A., Sellen A.J., and Duthie D.J.R. “Toward the paperless Hospital” *British Journal of Anaesthesia*. 78:762-767. 1997. J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[2]. Subramanian M., Whitman L., and Cheraghi S. H. “*Business Process Reengineering: A Consolidated Methodolog*” Dept. of Industrial and Manufacturing Engineering Wichita State University Wichita, KS-67260 0035, US, Nov. 1999.

[3]. Karunatilaka M. N., *Business Process Re-Engineering for the Emergency Treatment Unit at the Colombo South Teaching Hospital* M.B.A thesis, CSE, UOM, Moratuwa, 2009.

[4]. Black, Sue (2001). "Computing ripple effect for software maintenance". *Journal of Software Maintenance and Evolution: Research and Practice*. ISSN 1532-060X.

[5]. ICTA (2011) [Online] Available: http://www.icta.lk/en/programmes/re-engineering-government/131-main-projects/251-epopulation-registry-.html

[6]. Woolf B. “ESB-oriented architecture: The wrong approach to adopting SOA”. WebSphere SOA and J2EE Consultant, IBM. Sep. 2007.

[7]. Flurry G, Reinitz R. “Exploring the Enterprise Service Bus, Part 2: Why the ESB is a fundamental part of SOA”.Software Group, IBM. Sep. 2007.

[8]. Lewis G. A., Smith D. B., Kontogianni K. “*A Research Agenda for Service-Oriented Architecture (SOA): Maintenance and Evolution of Service-Oriented Systems*” TECHNICAL NOTE, CMU/SEI-2010-TN-00, SEI Administrative Agent, ESC/XPK, 5 Eglin Street, Hanscom AFB, MA 01731-210, Mar. 2010, http://www.sei.cmu.edu

[9]. IBM Corporation. “*Own your process, own your information, own your business with SOA*”. Software Group, Route 100, Somers, NY 10589 U.S.A., Dec. 2010, http://public.dhe.ibm.com.

[10]. MuleSoft. (2011). [Online] Available : http://www.mulesoft.com/mule-esb-features

[11]. Kamran S., Yarmand M. H., and Down D. G., “*Mined-knowledge and Decision Support Services in Electronic Health*” Dept. Computing and Software, McMaster University Hamilton, ON, L8S 4K1, Canada

[12]. Marrs, Tom; Davis, Scott (July 1, 2009). *JBoss At Work: A Practical Guide*. O'Reilly. ISBN 0596007345

[13]. Health Level Seven International (2011) [Online] Available : http://www.hl7.org/

[14]. OpenEHR. (2007). *ISO EHR standards* [Online]. Available : http://www.openehr.org/standards/iso.html

[15]. EN 13606 Association. (2011). *EN 13606 Association - Home* [Online]. Available : http://www.en13606.org/

[16]. Health Level Seven International. (2011), *HL7 Standards.* [Online] Available: http://www.hl7.org/implement/standards/index.cfm?ref=nav

[17]. HL7Book. (2011, May 12). *HL7 Version 2* [Online]. Available : http://hl7book.net/index.php?title=HL7_version_2

[18]. HL7 International. (2010, June 04), *HL7 Version 3 May 2010 Ballot.* [Online] Available: http://www.hl7.org/v3ballot2010MAY/html/welcome/environment/index.htm

[19].Mirth. (2011). *Products.* [Online]. Available: http://www.mirthcorp.com/products

[20]. HAPI, (2011, June 26) [Online] Available :http://hl7api.sourceforge.net/