

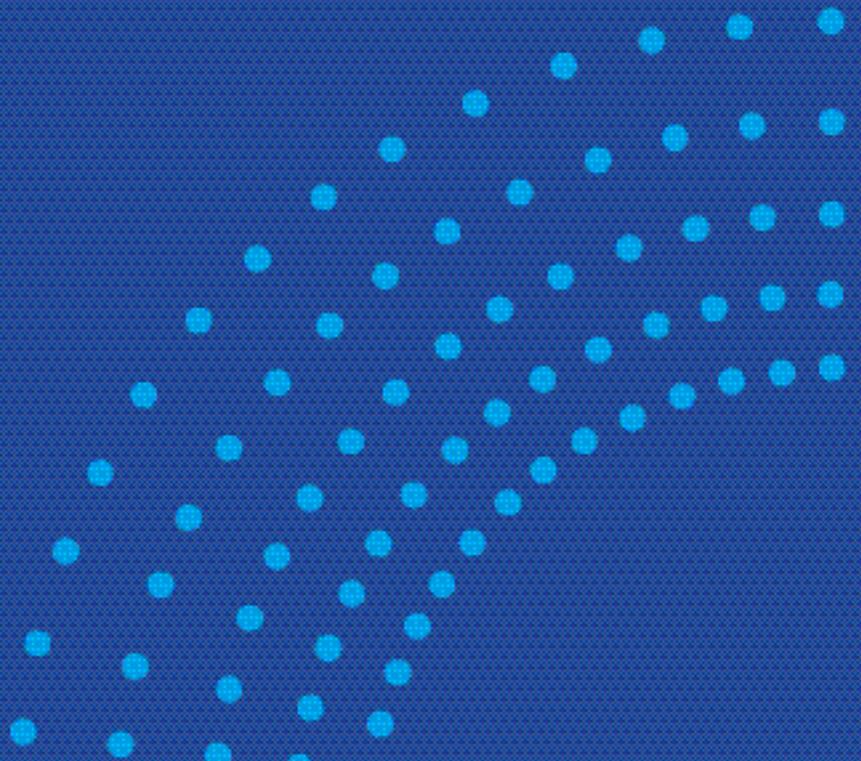
Master Thesis in Software Engineering and Management
**A SYSTEM DESIGN FOR A TELEMEDICINE
HEALTH CARE SYSTEM**

Asadullah Shaikh & Muhammad Misbahuddin
Göteborg, Sweden 2007



IT University
of Göteborg
CHALMERS | GÖTEBORGS UNIVERSITET

Department of Applied Information Technology



REPORT NO. 2007/78

A SYSTEM DESIGN FOR A TELE-MEDICINE HEALTH CARE SYSTEM

Asadullah Shaikh & Muhammad Misbahuddin



Department of Applied Information Technology
IT UNIVERSITY OF GÖTEBORG
GÖTEBORG UNIVERSITY AND CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2007

A SYSTEM DESIGN FOR A TELE-MEDICINE HEALTH CARE SYSTEM

Asadullah Shaikh & Misbahuddin Muhammad

© Asadullah Shaikh & Muhammad Misbahuddin, 2007

Report no 2007:78

ISSN: 1651-4769

Department of Applied Information Technology

IT University of Göteborg

Göteborg University and Chalmers University of Technology

P O Box 8718

SE – 402 75 Göteborg

Sweden

Telephone + 46 (0)31-772 4895

Göteborg, Sweden 2007

PREFACE

We would like to express our gratitude to all those who gave us the possibility to complete this master thesis. We want to thank the Department of Software Engineering and Management of I.T University of Göteborg for giving us permission to commence this thesis in the first instance, to do the necessary research work and to use all necessary resources.

We are deeply indebted to our supervisor Dr. Thomas Lundqvist for stimulating suggestions and encouragement helped us in all the time of research for and writing of this thesis. Our colleagues from the Department of Software Engineering and Management supported us in our research work. We want to thank them for all their help, support, interest and valuable hints. Our supervisor, Thomas Lundqvist looked at the final version of the thesis for English style and grammar, correcting both and offering suggestions for improvement.

Especially, we would like to give our special thanks to Dr. Agneta Nilsson whose precious advice enabled us to complete this work.

Asadullah Shaikh

M.Sc Student

I.T University of Gothenburg, Sweden

E-Mail: shaikhasad@hotmail.com

Muhammad Misbahuddin

M.Sc Student

I.T University of Gothenburg, Sweden

E-Mail: mizbaah@gmail.com

A SYSTEM DESIGN FOR A TELE-MEDICINE HEALTH CARE SYSTEM

Asadullah Shaikh & Misbahuddin Muhammad

Department of Applied Information Technology

IT University of Göteborg

Göteborg University and Chalmers University of Technology

ABSTRACT

Telemedicine systems have been steadily increasing in number and scale over the past decades. This increasing trend intensifies the need for telemedicine in the era of national health care system. The increasing size of distributed telemedicine systems creates problem of data integration, vendor lock-in and interoperability.

In this thesis, we state the above problems and solutions of current telemedicine system by showing an example of Tele-WoundTM architecture. To achieve this goal, we have set of requirements from a company named as A-B medicine in order to build a telemedicine application for them. Those set of requirements will be named as Tele-WoundTM application, which is the part of telemedicine system. This thesis presents mainly architecture of Tele-WoundTM which is being build as Service Oriented Architecture (SOA), because we believe that by the adoption of SOA several problems in today's telemedicine system can be resolved. Currently several resources are complaining that users become limited to a single vendor due to announcement of new technology, so this thesis can at least presents the few solution of users problem.

We provide a detailed description of requirement of Tele-WoundTM system, and then proceed to design of an architecture that could be used as an example to other related application. The particular architecture that we have designed is comprises of several web services.

In addition to the specific results that we present, we feel that the collective wisdom provided by the theory, requirement, evaluation and discussion throughout this thesis provides valuable insight for others in the same area of research.

Keywords: Tele-Wound, Architectural design of Telemedicine application, Service Oriented Architecture in telemedicine system, SOA Tele-Wound.

Contents

PREFACE	IV
1. INTRODUCTION	1
1.1 BACKGROUND.....	1
1.2 DELIMITATION OF THE PROBLEM.....	2
1.3 PROBLEM FORMULATION	2
1.4 PURPOSE	3
2. METHODOLOGY	4
2.1 DESIGN RESEARCH.....	4
2.2 DESIGN REQUIREMENTS.....	4
2.3 ASKED QUESTION	4
3. THEORY	6
3.1 SOFTWARE DEVELOPMENT	6
3.2 VENDOR LOCK-IN	7
3.3 INTEROPERABILITY	7
1.1.1.1 3.3.1 Physical interoperability	8
1.1.1.2 3.3.2 Data-type interoperability.....	8
1.1.1.3 3.3.3 Specification level interoperability	8
3.4 SERVICE ORIENTED ARCHITECTURE	8
3.5 WEB SERVICES	8
3.5.1 SOAP.....	9
3.5.2 Security in SOAP.....	9
1.1.1.4 3.5.2.1 Why we need SOAP layer security.....	10
1.1.1.5 3.5.2.2 End-to-end Security.....	10
1.1.1.6 3.5.2.3 Security Requirements.....	10
3.5.3 WSDL	11
3.5.4 UDDI.....	11
3.5.5 XML Web Services.....	11
4. REQUIREMENTS.....	12
4.1 REQUIREMENT SPECIFICATION.....	13
4.1.1 Functional and Data Requirements.....	13
4.1.2 Non-Functional Requirements.....	13
5. PROPOSED ARCHITECTURE FOR TELE-WOUND™	14
5.1 DESCRIPTION	14
5.2 DESIGN DESCRIPTION	15
5.2.1 System Component	15
5.2.2 System Sequence.....	16
5.3 IMPLEMENTATION OF WEB SERVICES.....	18
5.3.1 Interoperability through Web Services.....	19
6. EVALUATION	20
6.1 TECHNOLOGIES USED IN TELE-WOUND™	20
6.2 WHY SERVICE ORIENTED ARCHITECTURE	21
6.3 SECURITY IN TELE-WOUND.....	23
7. DISCUSSION	24
7.1 PROPOSED SOLUTION.....	24
7.2 RELIABILITY IN WEB SERVICES.....	24

8 FUTURE WORK..... 26

 8.1 USABILITY 26

 8.2 AVAILABILITY AND RELIABILITY 26

 8.3 SAFETY AND SECURITY 27

 8.4 EFFICIENCY 27

 8.1 SCALABILITY 27

9. CONCLUSION 28

10. REFERENCE 29

APPENDIX A: ACRONYMS 32

APPENDIX B: QUESTIONIARE RESPONSE..... 323

1. INTRODUCTION

This chapter gives an introduction to this report. Section 1.1 explains the description of the chosen subject as a background; Section 1.2 is about delimitation of problem; Section 1.3 presents the problem formulation and derived research questions, whereas Section 1.4 describes the purpose of the report.

1.1 Background

Healthcare is about the prevention and healing from illness. The term health care system refers to a country's system of delivering services for the prevention and treatment of disease. The Swedish health care system is built on local authority and their services are funded through taxes by the government of Sweden.

Health care systems are now changing due to the dynamic nature of technological and scientific medical practices. The health care providers are now swiftly adopting the technology into their health care procedures for example modern surgery helped to create the modern hospitals. Due to the advancement in medical technology the cost of treatments becomes higher. Lots of health care provider, especially hospitals cannot manage to pay for high cost equipment to take care of diseases and wounds. Some complex kind of treatments like bypass surgery of heart, surgeries, trauma care and other complex procedures need expert medical teams including equipment and facilities. Such resources are only available at hospitals and surgical centers with expert doctors. Therefore people living in rural areas need to travel huge distances to access more costly and complex levels of care.

Telehealth is a method of treatment by using telecommunication technology such as telephones, cell phones, text messages, PDA's, the Internet and videoconferencing. Telehealth care systems are becoming more popular due to its mobility. Nowadays the word telemedicine is well-known among physicians, medical experts, biomedical engineers, and others. But what exactly does telemedicine mean? The following statement on telemedicine is fairly informative [20].

“Telemedicine utilizes information and telecommunications technology to transfer medical information for diagnosis, therapy, and education”

Telemedicine is a part of telehealth, which is based on the technologies by using telecommunication for the interaction between health professionals and patients in order to execute medical actions at distance. Telemedicine is a very vast field in today's world, which is widely used to reshape the systems in the health care. Telemedicine is helping different healthcare system to solve the problems in many ways.

This thesis presents results from a development project concerned with a telemedicine application called Tele-Wound TM. It is a design of architecture in order to help the present and future telemedicine systems.

A-B medicine is a company in Gothenburg Sweden that makes innovative application of telemedicine for the Swedish health care system. Presently their focus is on the Tele-WoundTM application which will follow the healing of patient's leg ulcer or wound with the help of MMS messages. These MMS messages consist of patient's picture fetched by the nurse(s) in order to send into perspective database at the clinic where the patient's responsible doctor works. In this way the patient can save his/her unnecessary trips to the doctor or one can ask the patient to come for a visit if the ulcer is worsening.

1.2 Delimitation of the problem

Telemedicine applications need to run in heterogeneous computing environments. Data integration, vendor lock-in and interoperability is major concern in telemedicine applications [1]. In the current situation, most of the telemedicine systems are designed by the consideration of today's needs of companies. Due to this, lots of problems are occurring in the area of telemedicine.

Most of the telemedicine systems vary from each other. Also the infrastructure of current telemedicine system is quite different then previous systems. It means the client must be limited to the same vendor in order to maintain the integration and interoperability between the old and new telemedicine system. Including all these things, those vendors can not convert on new technology, which requires a heavier investment because of the integration to previous systems. Let's take an example that if a vendor wants to integrate one telemedicine system to another telemedicine system but it could be also quite hard to implement due to extra cost of new technology announced. Here in this regard many issues are described above but still there is a matter of vendor lock-in and interoperability, which stands on top in few industries. Even there is question in industry while working on each telemedicine system components that could it be interoperable in future? [2]

Now the problem is that how the user can be sure that his telemedicine system will not create the problem of interoperability along with vendor lock-in and data integration. What will happen when he wants to integrate his one of telemedicine component to other?

With the consideration of all aspects, we can at least give an example design of a telemedicine system, which suggests a few solutions to those people who are already working on this area or intended to work in future.

1.3 Problem Formulation

The research was conducted in order to provide an answer of the following question:

“How to build a system architecture for Tele-WoundTM that can resolve the problem of vendor lock-in and interoperability”

1.4 Purpose

The purpose of this report is to show example system architecture of telemedicine system along with preferred design choice. The lack of previous published system architectures in this subject further motivates the study as well as the importance of our design choice in telemedicine systems. The results have possibility to affect the ongoing discussions about the research field of telemedicine system by presenting how the vendor lock-in, data integration and interoperability.

Another motivation for this type of study is to raise the interest of all those persons, who are afraid of not having the vendor support in the area of telemedicine. Telemedicine systems are most discussed and debated topic in today's society.

To achieve above purpose, we have some basic requirements of Tele-Wound™ application, which is a part of telemedicine system that contains detailed descriptions of Tele-Wound™ application. Our intention is to build a system architecture that suggests the solution of vendor lock-in, data integration and interoperability in the area of telemedicine.

This thesis also consists of future requirements in order to suggest a better design than existing one.

Furthermore this thesis contains theory chapter (see chapter 3) in order to give general concept to a reader about Service Oriented Architecture (SOA), vendor lock-in, interoperability and web services. By reading this chapter, the reader can understand the terms and technologies used in entire thesis. Theory chapter gives motivation to our design choice i.e. SOA.

2. METHODOLOGY

In this chapter we discussed about our methodology used in this study. Section 2.1 describe the design research 2.2 defines about design requirements and 2.3 present our asked question.

2.1 Design Research

Design Research is also called as 'Improvement Research' [31]. It put emphasis on the problem solving. In the case of our research for A-B medicine, we worked on existing software architectures of telemedicine, in which there are few common problems like data interchanging and interoperability. To evaluate these problems we considered previous and existing architecture of telemedicine in order to capture the problem point.

In order to carry out this research, the study of the previous work in the field of telemedicine should be needed a survey of the existing applications and comparison of those existing applications would also be vital for proposing better and suitable software architecture for the telemedicine system. After considered previous architectures exist in A-B medicine and decided to draw SOA based architecture.

2.2 Design Requirements

In order to build system architecture as SOA, it is always good to collect the design requirements from different companies to know that how many organizations are influences with SOA. In order to do that, we sent an e-mail with the questionnaires (see section 2.3) to a perspective companies in order to collect their view points. The purpose of sent questionnaires was to understand and analyze the data for building such new system architecture by avoiding the issue of interoperability, vendor lock-in and data interchanging.

In a short period of time, we chose 5 companies. All the organizations chosen were first studied in order to see their level of developing the telemedicine and distributed products. The selected companies were initially approached with e-mail along with questionnaires. Fortunately we got reply from 3 companies out of 5 companies.

2.3 Asked Question

The questions were designed to be open ended and close ended. The formulation of the questions was based on the number of developed software product in the organization.

The following questions were asked from selected companies.

1. How many software systems you have developed in your organization?

Ans. 1) Less then 50 2) between 50 to 100 3) More then 100

2. How many Telemedicine and distributed system you have developed so far?

Ans. 1) Less then 50 2) between 50 to 100 3) More then 100

3. How many customers came back to you for further extension of their software product?

Ans. 1) Less then 25 2) between 25 to 50 3) More then 50

4. Have you developed any of the software system as a Services Oriented Architecture (SOA)?

Ans. 1) Yes 2) No

5) Are there any customers who are limited to your organization?

Ans. 1) Yes 2) No

6) What are the methods you provide to customers for interoperability of their product?

Ans.

7) Do you think SOA can help to resolve the problems of Software Interoperability?

Ans. 1) Yes 2) No

3. THEORY

In this chapter we discussed some basic theories related to background. It also gives a brief description of study required. Section 3.1 explains the basic concept of software development; Section 3.2 is about general theory of vendor lock-in; Section 3.3 presents the interoperability and its levels; Section 3.4 is about Service Oriented Architecture and section 3.5 is about web services

3.1 Software Development

Software development is the conversion of the user requirements in to products [21]. It is basically a process of software engineering joint with objective of the market to develop a software product [22]. Software Engineering is responsible for the organization of software principles and methods in order to create a reliable and functional software product. More over it is a systematic and disciplined approach to developing software; it applies computer science and engineering discipline.

Software architecture of software is rigorous system that represents the physical structure of the system, which encompasses software components, software properties and their relationship. Architecture develops a bridge between requirements and design which performs the essential steps to satisfy the customer needs. In other words the architecture definition is placed between requirements analysis and software construction that could be design, code or test.

In the below figure (figure 1) the three triangles (the peaks) showing the major software development activities for requirement analysis and construction; the increasing of all three shapes are representing the system development as time goes on. The curling arrows shows the progressive movement of steps towards requirements, architecture and construction

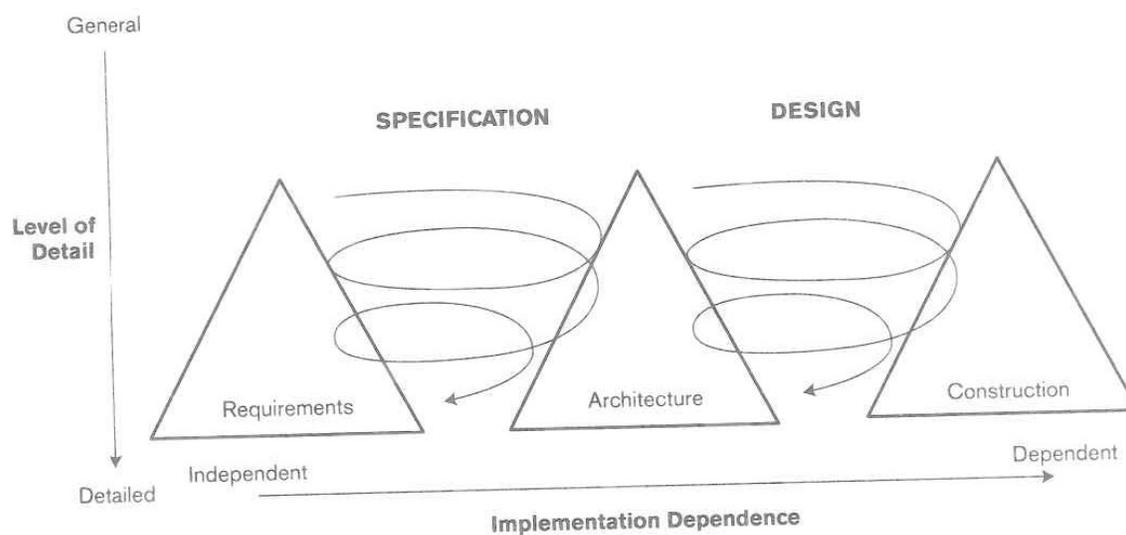


Figure 1: Three Peaks Model [24]

Requirement analysis provides the overview of the scope and desired functionality of the system, while software construction is the collection of incremental deliveries along with the useful set of function. The construction of each phase gives feedback to architecture definition. Requirement analysis, architecture definition and software construction are strongly related in the mean of software architecture because requirement analysis gives an idea about early context for architecture definition. Architecture definition gives the implementation process of the product [24].

3.2 Vendor Lock-in

Vendor lock-in also known as customer lock-in, where clients are dependent to a single vendor for products and services and cannot move to another vendor without investing heavy costs. Lets take an example of Microsoft. Microsoft carries big level of vendor lock-in, based on its wide set of Application Program Interface.

“The European Commission, in its March 24, 2004 decision on Microsoft's business practices, quotes, in paragraph 463, Microsoft general manager for C++ development Aaron Contorer as stating in a 1997-02-21 internal Microsoft memo drafted for Bill Gates

"The Windows API is so broad, so deep, and so functional that most ISVs would be crazy not to use it. And it is so deeply embedded in the source code of many Windows applications that there is a huge switching cost to using a different operating system instead.

"It is this switching cost that has given the customers the patience to stick with Windows through all our mistakes, our buggy drivers, our high TCO, our lack of a sexy vision at times, and many other difficulties [35] Customers constantly evaluate other desktop platforms, [but] it would be so much work to move over that they hope we just improve Windows rather than force them to move.”

"In short, without this exclusive franchise called the Windows API, we would have been dead a long time ago."

Software of Microsoft Application demonstrate lock-in through file fomrats. “For example, use proprietary datastore files and interfaces which are impossible to read without being parsed, and such parsers may in turn not be able to exist legally without performing reverse engineering. For example, to access data contained in Outlook's '.PST' files, the application must process the request through Outlook instead of directly handling the file”. [35]

3.3 Interoperability

Interoperability is capability to transmit data and exchange information between two or more systems and permitting each other to process information independently [33]. The current challenge with exchanging and processing medical data from the patient’s end to the doctor’s end is that different vendors have their own proprietary standards for data formats. The platform should give as much as possible solutions for providing data exchange making the same standard of data format.

The transformations of the patients records through patient to doctor's end are based on a predefined communication protocol for data interchange, which is used to control and maintain connection between the two sites, thus ensuring portability, interoperability and security of the transmitted data [5]. The interoperability could be evaluated by the following perspective.

3.3.1 Physical interoperability

Physical interoperability is usually attained by transferring information through electronic media such as electronic files on any storage media. In our Tele-WoundTM system, the problem of keeping the patient records (pictures of their wound) would be occur when the data will be exceed and there will be no alternative volumes for replacing. The above approach of interoperability is very important for the application as performance point of view, as the information must be entered into the applications using different interface.

3.3.2 Data-type interoperability

Data-type interoperability focuses on the type or the structure of the data which would be transferred from user end into the application end. The Tele-WoundTM application is a web based application and design for the leg ulcer patients so that the pictures of the infected area of the patients should be presented which would be accessible on web.

3.3.3 Specification level interoperability

Specification level interoperability gives us high-level, representation-independent approach to combining software components that are written in different languages or that are run on different machines. Specification level interoperability approach thereby attains information hiding for interoperating programs, encouraging the use of entity descriptions (i.e., type definitions) that promote the overall organization of a software system. [10].

3.4 Service Oriented Architecture

Services oriented architecture SOA is a new and developing model for building distributed applications [36]. It is a collection of services, and those services components reside in different domains.

3.5 Web services

Web services are web based applications used for exchange the data with clients through transport protocols. These are XML-based standards designed for interoperable machine to machine interaction over a network.

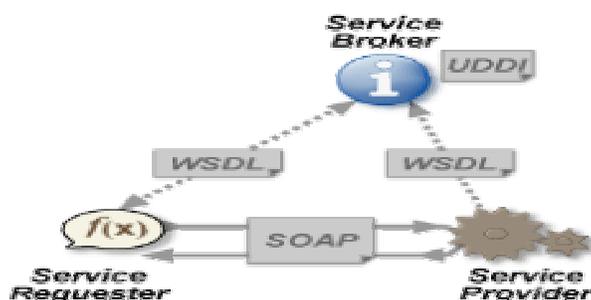


Figure 2 Web Services

3.5.1 SOAP

Simple object access protocol (SOAP) is an XML based message binding protocol that defines how messages can be well ordered and processed by system to system. SOAP use HTTP for communicating with different platforms. It indicates exactly how to encode an HTTP header and an XML file so that a program in one computer can call a program in another computer and pass it information. It also specifies how the called program can return a response [37]

SOAP is easy, flexible and extendable mechanism for interchanging prearranged data. SOAP was designed for Remote Procedure Call (RPC) mechanism for extensively used internet standards such as HTTP and XML. SOAP has thin layer enveloping a message expressed in XML. SOAP envelop is stated in XML and allow us to add large variety of information to the message. SOAP enveloped consists of Header and body. Header is used to add a feature to a SOAP messages. The body is a container for data, so SOAP can be another layer between transport layer and application layer.

Layer	Examples	Comments
Application	Business data (e.g., purchase order)	In general, the SOAP Body is used to carry application data.
Message	SOAP	Transaction, message routing, security, etc.
Transport	HTTP, SMTP	Deals with hop-by-hop data delivery.

Table 1. SOAP communication layering (37)

3.5.2 Security in SOAP

The header of SOAP offers an adaptable mechanism for increasing SOAP messages. While to add a security in SOAP header is a great option to deal with. In our system design we are concern with following issues:

- **Encryption:** How to perform encryption for SOAP messages.
- **Digital Signature:** How to digitally sign SOAP messages.
- **Authorization:** How to authorize senders of SOAP messages

Here we will define the syntax and the semantics of header entries (called SecTags) that will transmit data associated to above issues.

3.5.2.1 Why we need SOAP layer security

Initially, any one can ask that “why we need the SOAP message security?” Is transport security is not enough? Well we believe that transport security is not enough, so we required SOAP layer security. There are several reasons to argue about SOAP layer security.

3.5.2.2 End-to-end Security

SOAP can be used with several types of transport protocol like HTTP and SMTP. SOAP messages travel from one origin destination to desired destination by crossing several intermediate paths. Telewound is an application that is able to send and receive the messages. When TeleWound application will receive a soap message, it deal with header entries and then can forward the message further. It may also add a alike header entry intended for another Telewound.

SSL/TLS are secure transport protocols and these are not good enough to be trusted for the security of messages during broadcast. Although messages will received and sent by telewound, so safe communication is quite hard if application is not fully trusted even though links between them are trusted.

3.5.2.3 Security Requirements

Our aim is to protect soap messags including its header and body. Here in this regard we have following security requirement for message broadcast

- **Confidentiality** that is an unauthorize person can not read the message.
- **Authorization** that the sender is authorized to send a message.
- **Data integrity** that is unauthorized modification of data.
- **Message origin authentication** assures that the message is sent by authorized sender and is not a replay of a previously transmitted message.
- **Non-repudiation** assures that the sender not deny that he/she has sent the message.

In order to complete above five requirements, we talked about encryption, digital signature and authorization.

- **Encryption:** XML-Encryption [XENC] can be used for this encryption. It make able the user to encrypt any data in the SOAP message.
- **Digital Signature:** W3C XML Signature standard [XSSP] is usefull for this kind of requirement. It suitable requirement because SOAP is XML-based protocol. The heapder of SOAP can have XML signature within the message. To validate it, actor attribute might be usefull for it

- **Authorization:** In the case of authorization, the user may not be permitted to request a service. To satisfy the requirement of authorization in SOAP layer is the use of authorization authority (AA) to avail the centralized authorization. In this case every user needs to get certain services from AA. Firstly the provider of services will verify the user's identity, and then makes an access control decision.

Table 2. Cryptographic Techniques Used

Requirement	Cryptographic Technique
Encryption	XML-Encryption [XENC]
Digital Signature	W3C XML Signature [XSSP]
Authorization	Attribute certificate ("push" model)

Table 2 summarizes the cryptographic techniques

3.5.3 WSDL

Web services description language (WSDL) is an XML based language and provides a model for describing a web services. It is derived from SOAP and it is used in combination with SOAP and XML to provide web services over the Internet [38]

3.5.4 UDDI

Universal Description, Discovery and Integration (UDDI) is also an XML based web services standard. It is designed to be interrogated by SOAP messages and to provide access to WSDL documents describing the protocol bindings and message formats required to interact with the web services listed in its directory [39]

3.5.5 XML Web Services

In most of cases, there is a misconception that ALL web services employ XML messages with Simple Object Access Protocol (SOAP) over HTTP (Hypertext Transfer Protocol). That's false. Web services messages can employ XML; it's done mostly through SOAP headers, but usually that's not necessary to use SOAP indoctrination for body of message. For the web services there are two well distinct principles: WSDL (Web Services Definition Language) and UDDI (Universal Discovery, Description and Integration). With the consideration of all above services, Tele-WoundTM has flexibility in formats and transport protocol with interoperability of web services.

4. REQUIREMENTS

In this chapter we provide a brief description of requirements specification of Tele-Wound™ application given by A-B medicine. Moreover the chapter is consists of the real requirement provided by the A-B medicine, which need to be implementing in initial Tele-Wound™ application.

The Tele-Wound™ application can follow the procedure for healing of leg ulcer patients or wounds with the help of MMS messages. These MMS consists of patients' pictures fetched by the nurse in order to send these pictures to the perspective database. In this way the patient can save an unnecessary trip to the doctor or one can ask the patient to come for a visit if the ulcer is worsening. The application should be programmed which is compatible with windows platform and shall include database (a database that is cheap and easy to run on the server, i.e. MySQL). The framework of application shall have an easy to use layout. The picture of the patient's ulcer shall be presented in such a way that they can be compared on the computer screens. In the database you can search on the bases of social security number, general description and telephone number. The below figure gives you better idea about Tele-Wound™ application.

Tele-Wound™ system consists of two main parts: one part will be on the user (patients end) and other part will be on the doctor's end. Form the first end, (nurse) will be responsible for collecting and transmitting still images of the patients from the happening place to the doctor's location while the doctor's end is responsible for receiving and displaying incoming data. Communication between the two ends completed by sending MMS's and emails through mobile network which ensures safe data transmission

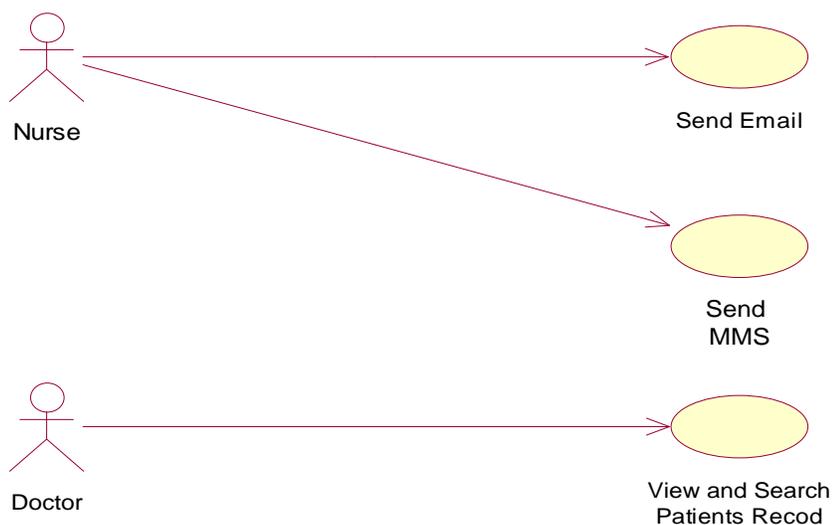


Figure 3 High level use case

4.1 Requirement Specification

This section is consist of requirements specification i.e. functional and non-functional provided by A-B medicine, these are the basic requirement of the company that should be fulfilled in working prototype of Tele-Wound™ application.

4.1.1 Functional and Data Requirements

- The application shall be receiving MMS to the application server (somewhere in database) containing the picture of the leg ulcer patients.
- The application shall be receiving Email to the application server (somewhere in database) containing the picture as attachment file of the leg ulcer patients.
- The application shall be able the store the records of the patients in the form of text (containing the social security number followed by some description) and pictures of infected areas of their leg ulcer or wound.
- The application shall be able search the patients record on the bases of social security number, date and telephone number
- The doctors can view the patient's previous record from the application by supplying their social security number.

4.1.2 Non-Functional Requirements

- The Tele-Wound™ application should have a good user interface, so it will be quite easy to use for the actors of the system in order to communicate with each other.
- The Tele-Wound™ application must be fast, throughput, or response time constraints on the system. So the Tele-Wound™ application will have the ability to keep a given level of performance in case of fault or problem occurred.
- The Tele-Wound™ application should be error free, so the application should have capability of reestablish level of performance after error occurred.

5. PROPOSED ARCHITECTURE FOR TELE-WOUND™

Chapter 5 gives brief description of proposed architectural description of Tele-Wound™. It consists of design description containing system components and system sequence, implementation of web services in Tele-Wound™ and interoperability through web services.

5.1 Description

In the world of telemedicine, lots of telemedicine applications are exist in the environment and lots of similar architecture has been proposed for ambulatory for patient monitoring but our Tele-Wound™ application is a sort of different application influenced by Services oriented Architecture. The targeted audience of this application will be leg ulcer patients and the main purpose of this application is to serve those leg ulcer patients who can not move freely. Their pictures of infected areas will be fetched through camera (cell phone camera or digital camera) by the nurse from their location, after fetching the picture he/she will send those pictures into our application server along with little description of patient.

This Web based architecture of Tele-Wound™ is utilizing telecommunication (GSM/3G) and internet technology and their major's components/modules are more dependent on those technologies. This architecture is also confirming and assuring the availability of medical data from the other resources. Specialists in the doctor's end would be needed with internet access and a browser so that they can help out patients from any place or city or even country for serving the patients/nations.

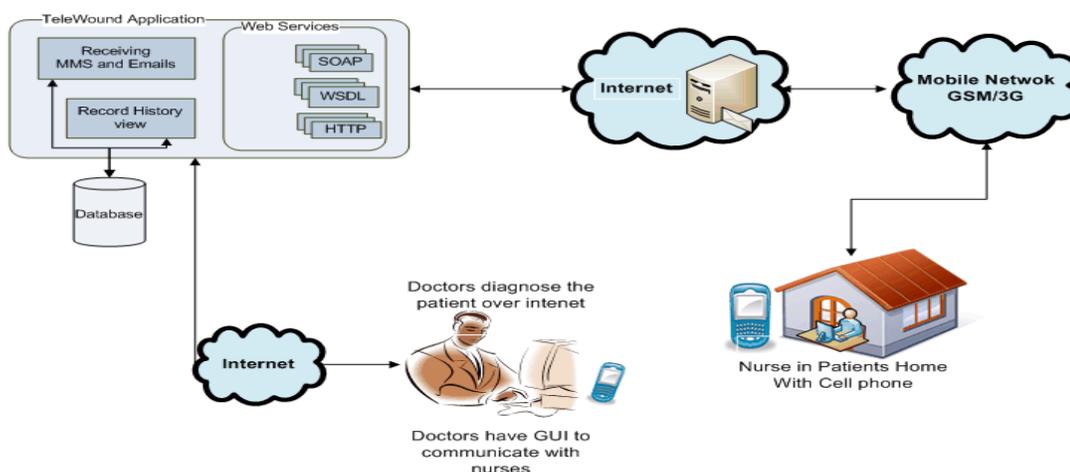


Figure 4 Context diagram of Tele-Wound

The above context diagram presents sending and receiving MMS and Emails from the Nurses. The MMS messages will be consists of patient's picture fetched by the nurse(s), which will be store on database at the clinic/hospital, where the patient's responsible

doctor can work on the patient's records and make some prescription on the basis of received data.

Application would be deployed on Visual Studio .Net Framework with Internet Information Server and MySQL will be used as a database. Receiving MMS module will receive the MMS from our premium MMS service provider and will separate images from it. After the separation, the images will be saved in to perspective database including the description of patient. Receiving Email will also work same like receiving MMS but it receives email from external email server and will store information in to databases. Once MMS and Emails received on application server and stored into the data base, then doctor and specialist will be able to look on patient's data according to their latest pictures of wound so that they can be able to make prescription. The picture of the patient's infected area of ulcer shall be presented in such a way that they can be compared on the computer screen according to latest to previous. Doctors can also search for the patient's history record on the bases of their social security number, general description and telephone number.

The application server may cater to many patients concurrently but it depends on the availability of the doctors/specialist/consultants having different areas of expertise. The specialist may use any computer with Internet access to offer his or her expertise. Standard browser interface is required to connect the application server for accessing the patient's data and giving feedback to them through text, voice or video chat or SMS messages. In this way the patient can save an unnecessary trip to the doctor or one can ask the patient to come for a visit if the ulcer is worsening.

5.2 Design Description

The designing of architecture of Tele-WoundTM application will be SOA based architecture. Below we describe about our design description of Tele-WoundTM application, we describe system component and system sequence to provide the technical design of a system.

5.2.1 System Component

This component diagram of Tele-WoundTM represents the components of the Tele-WoundTM application. Doctor's component represents the doctor's activities of application. This component also contact with the external email server for receiving emails. The method read_emails() request the email from the email pop3 server and give it back to doctors component where the doctor's component receive all emails and separate patient's images and text descriptions and save it into databases.

This component also need patients records so that the doctors can be able to treat their patients after viewing and analyzing their previous history records, fetch_record() will fetch the records from database to the doctors components by using web services.

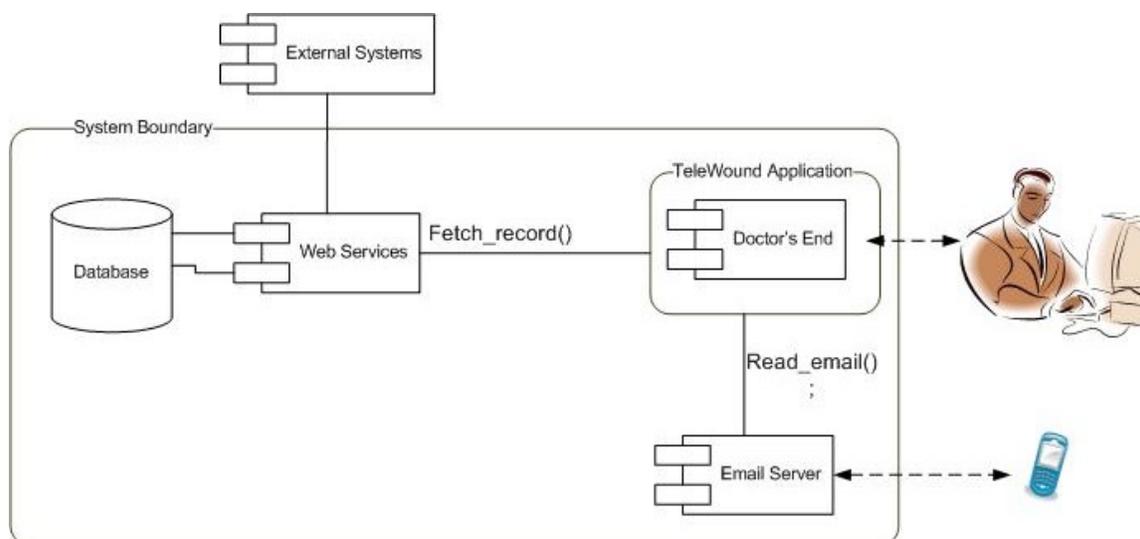


Figure 5 Component Diagram of Tele-Wound™ application

Services	CRUD Type	Description
DeleteRecords(intID)	Delete	Delete the patients records
Form1_Load()	Read	Load many parameters and variables when form load (i.e pwd of email accounts and user and pwd of database server)
InsertEmailDataIntoDatabase(strEmailID, strEmailTxt, strImagePath)	Create	Take email/MMS data from email server and then parse them and insert it into database
InsertUniqueEmailIDIntoDatabase(strEmailID)	Create	Insert the unique email id assigned by email server into database.
PopulateGrid(sql)	Read	Populate the present the patient's records against the patient's personal number in webpage.
Read_emails()	Read	Read email from server in a assigned time interval
StartMedicineEngine()	Read	
UpdateRecords(intID)	Update	Update patients records

Table 3. List of methods used in Tele-Wound™ Application.

5.2.2 System Sequence

A below sequence diagram explains how groups of objects work together in achieving the system behavior. Sequence diagram contains one main object for doctor's end. From this component doctor will receive the pictures from email server. In case of error occurred, the message will be displayed on doctor's end. Doctor can also check the patient's record in order to see entire patient's history. That history will be fetched from perspective database. When any error occurred, the doctors will get message on their

screen, otherwise doctor will receive entire patient’s history, so that they can make prescription for their patients.

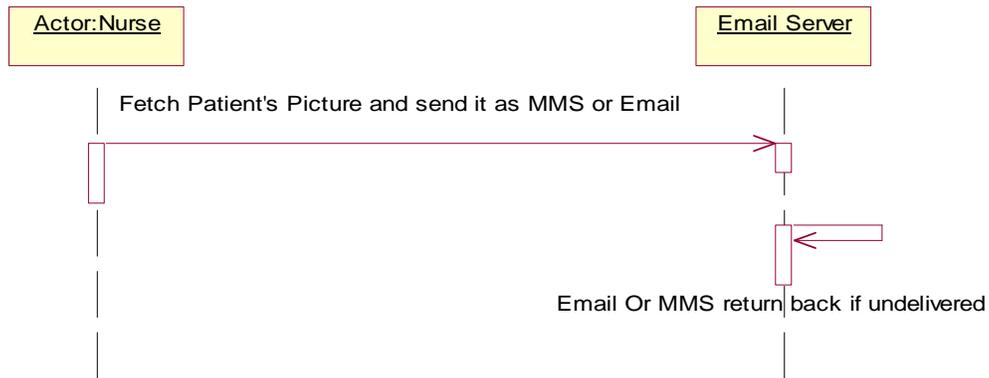


Figure 6 Sequence Diagram of Tele-Wound™ Application of Nurse’s End

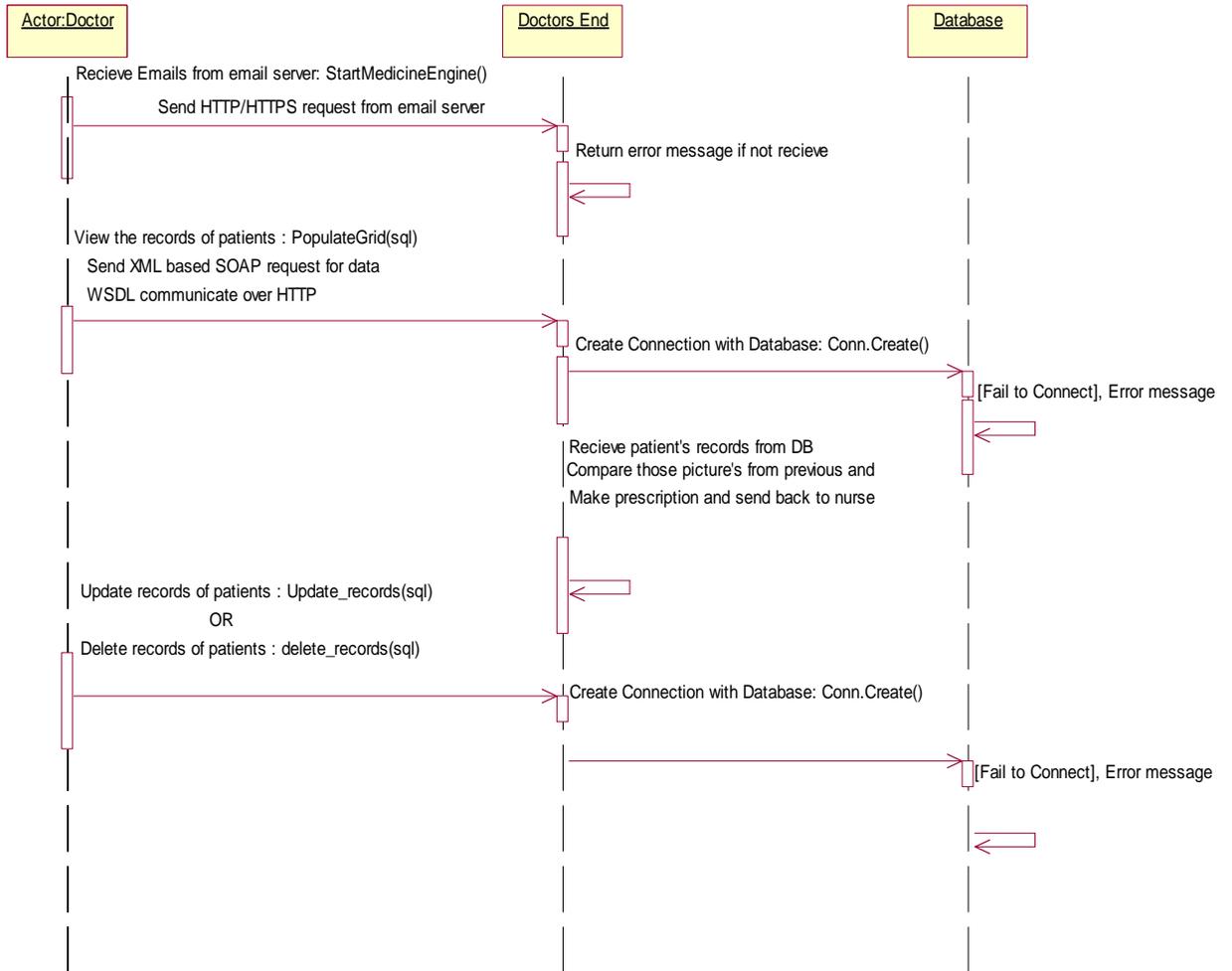


Figure 7 Sequence Diagram of Tele-Wound™ Application of Doctor’s End

5.3 Implementation of web services

The application will communicate with web services. Web services technologies can be used to create a contract between different software systems [27]. We have planned to use different web services. The main advantage of this approach is that it allows organizations to cost-effectively reusable component without adopting expensive technique and technologies.

Here we used different web services in Tele-Wound TM application, these web services are allowing to communicate and interchange the data between similar or dissimilar telemedicine systems. Simple Object Access Protocol (SOAP) has been used as an XML based message binding protocol, the primary protocols of SOAP are HTTP and HTTPS, and POP3 is used for communicating with email server. Here SOAP defines how messages can be well ordered and processed by system that facilitates cross platform independency in different programming language or platform, and thus interoperability between applications written in different programming languages and running on different operating systems can be achieved.

Figure 8 shows that how WSDL interchange the different platforms, share the web services contract and communicate using SOAP over HTTP, it will help to resolve the problem data interchange from one platform to another, for make it possible both ends need to understand XML to interact data with in different platform. By considering the figure 8, we can resolve the interoperability problem in cross platforms with-in the similar telemedicine systems and if the two or more different telemedicine application from different vendors use these service then the application can able to communicate and interact their data. By this process the user will not be dependent to a

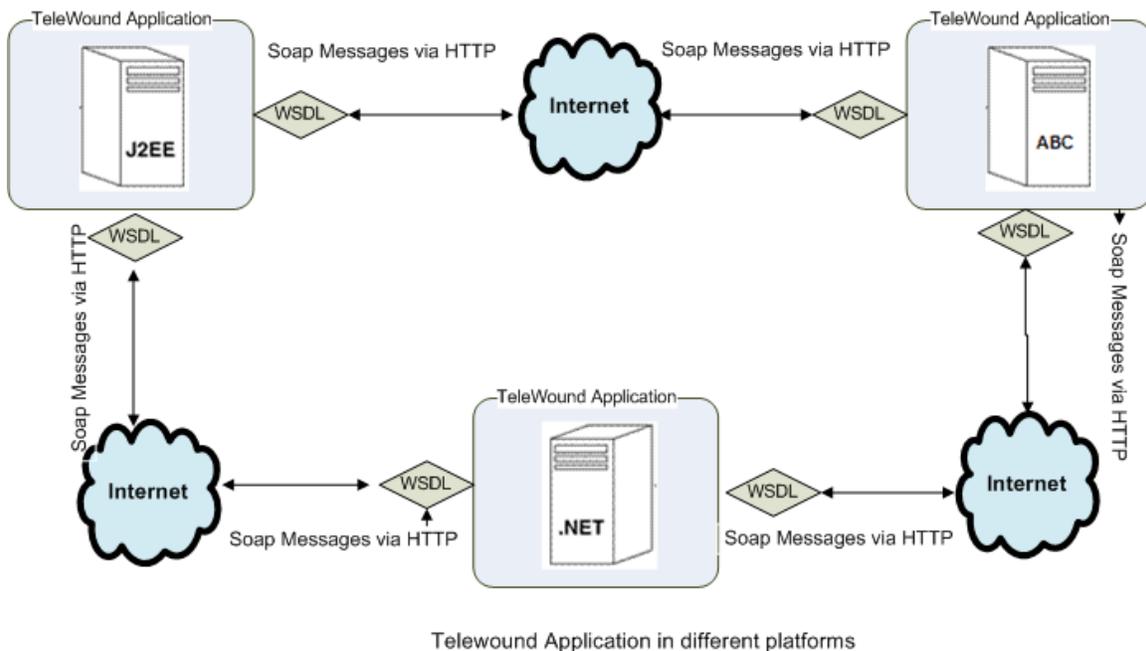


Figure 8 Tele-Wound TM applications in different platforms

single vendor because if the technologies are changed then user can easily change his telemedicine application according to his new requirements.

5.3.1 Interoperability through Web Services

Interoperability, that is ultimately required now a days can be achieved in our system by adopting the services oriented architecture. Interoperability is one of the important achievements that are gained from implementing those web services according to the services oriented interoperability. It is difficult to integrate two or more dissimilar system, but a web services layer between application and client can significantly remove these difficulties. Figure 8 and 9 shows two dissimilar applications communicate through XML based SOAP message over HTTP and they communicate by interpreting WSDL, through this service, dissimilar Tele-Wound™ application can communicate through SOA based architecture by using web services. All above services are the part of Tele-Wound™ in order to support all other telemedicine applications that are wonder to be connected to different telemedicine application.

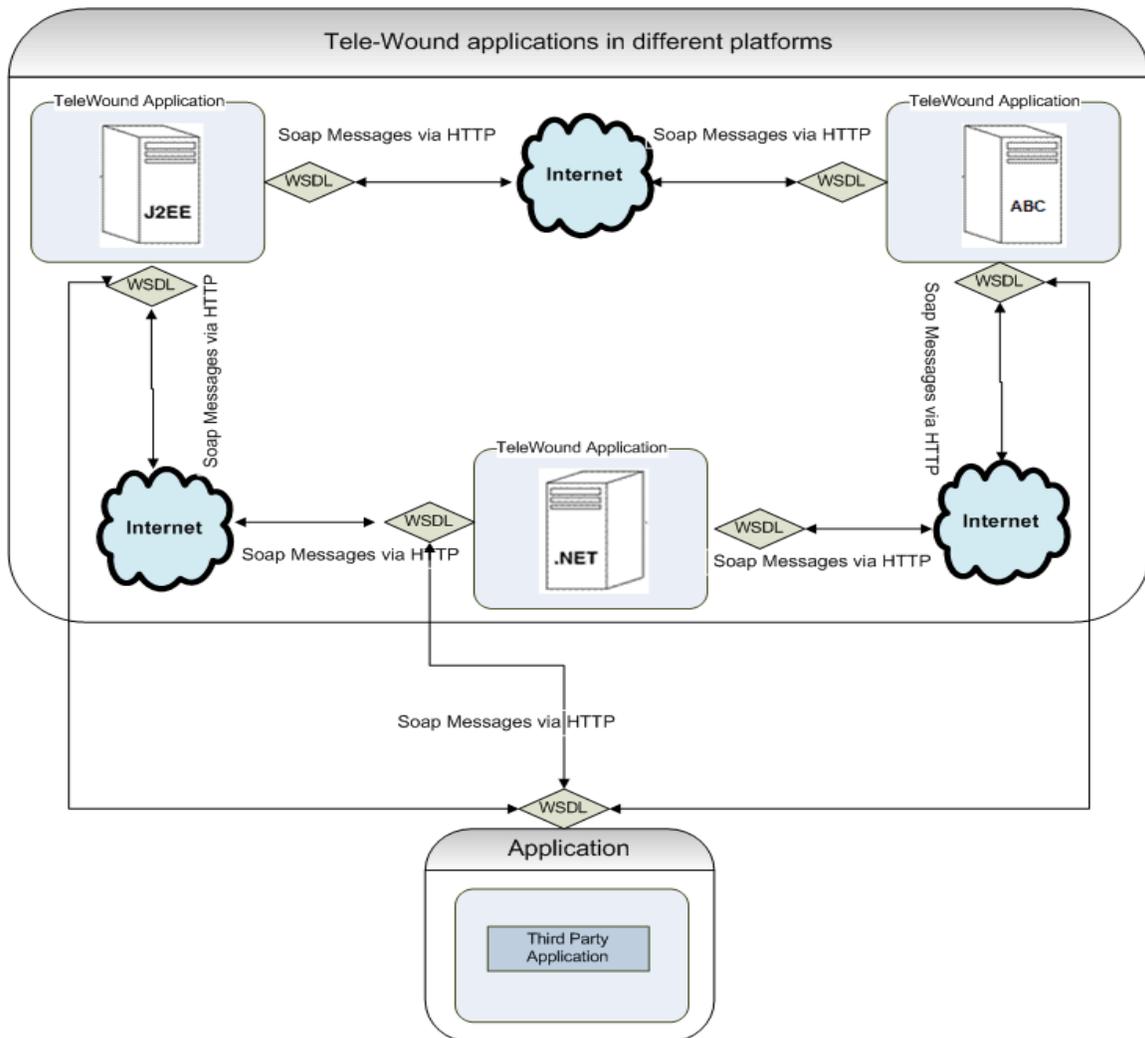


Figure 9 Interoperability with dissimilar application

6. EVALUATION

Evaluation chapter describe the evaluation of our proposed system of Tele-Wound.

6.1 Technologies used in Tele-Wound™

Being that architecture as services oriented architecture; the application will communicate with external web services. Web services technologies can be used to create a contract between disparate software systems [27]. We have planned to use different web services. The main advantage of this approach is that it allows organizations to cost-effectively reusable component without adopting expensive technique. Following are the little description and advantages of the applied web services.

By using web services description language (WSDL), the communication protocols and messaging formats has been standardized between external services in the Tele-Wound™ application. Figure 10 show the how the WSDL interchange the different platforms share the web services contract and communicate using SOAP, it will help to resolve the problem data interchange from one platform to another, for make it possible both ends need to understand XML to interact data with in different plate form. Web Services-Metadata Exchange (WSME) facilitates to know what the other end of application wants to know which type of information and how to interact between them. By using the above web service may be we can achieve the data interchange problem and if the two or more different telemedicine application from different vender use these service then the application can able to communicate and interact their data. By this process the user will not be dependent to a single vendor because if the technologies are changed then user can easily change his telemedicine application according to his new needs.

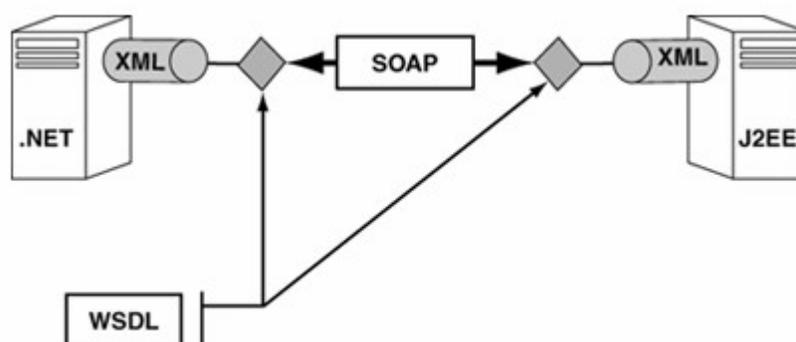


Figure 10 WSDL describe how to process SOAP message [27]

Now consider the figure 11, which illustrates the fact that when .NET platform and J2EE distributes the web service agreement, they communicate using SOAP messages. Basically these SOAP messages are XML documents [27] because each contributor requires the understanding of XML. The WSDL documents are approved by both parties

in order to produce the SOAP messages. For example if the data type incompatibilities are different, then both parties understand and interchange the data to each other because it is understandable to them. That connection shows that both ends should have access to same WSDL definition, so that both SOAP nodes can interchange the messages to the common WSDL definition. This process will certainly resolve the problem of connectivity and data interchanging in case of same WSDL definition. Tele-Wound™ is consisting of all above service in order to provide a better solution when the application is running on heterogeneous environment.

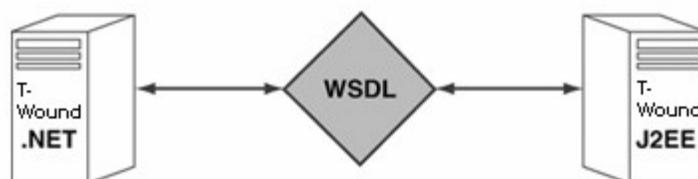


Figure 11 : The WSDL agreement permits dissimilar systems to connect[27].

Hence it's clearly stated that dissimilar Tele-Wound™ application can integrate, connect and communicate through SOA based architecture by using web services. All above services are the part of Tele-Wound™ in order to support all other telemedicine applications that are wonder to be connected to different telemedicine application. Only the difference is the usage of same web service and same WSDL contract in order to go for further communication.

6.2 Why Service Oriented Architecture

We want choose service oriented architecture for Tele-Wound™ because in today's world most of telemedicine systems are designed for as turnkey solution. Normally, it's quite hard to design such system architecture by different manufacturers to solve the problem of interoperability and vendor lock-in. Such as organization 'A' cannot manages the data interchange to organization 'B' and also cant retrieve the patient's data. In the same way telemedicine system frequently doesn't integrate the clinical data with other organization's infrastructure. Moreover the types of different telemedicine systems design makes hard to increase them with further clinical capabilities. This means the users of system may be lock-in to particular vendor solution and can not think of further procession of their system. Except these issues, those vendors are not able to integrate telemedicine systems immediately because adoption of new technology takes time and also required heavy investment [2].

Furthermore doctors and physicians documented the history of patients in order to give better treatment for future. Firstly, numerous specialists from multiple clinical communities share the data of patient's record. Secondly, electronic communication is more popular rather than manual communication and it is also reliable. So physicians rely on electronic data interpretation rather than human interpretation. Thirdly, the intercommunication of different telemedicine creates the issue of interoperability

including data integration. This issue became big and important when the procedure of health care system becomes distributed with rising communication between professionals from one sector to another. In this time, there is a need of such system which can easily handle the issues efficiently. In general the purpose of telemedicine is the arranging and integrating the data in such way that the people of healthcare sector should access that data outside the surroundings. According to medical informatics handbook [14], by the consideration of figure 1 there are such rules that say:

- The communication should be managed in better way between dissimilar organizations.
- There should be a certain rights on external resources in order to integrate the patient's data over the public networks.

The complication and vagueness starts from natural language and that makes integration of knowledge approximately impossible [15].

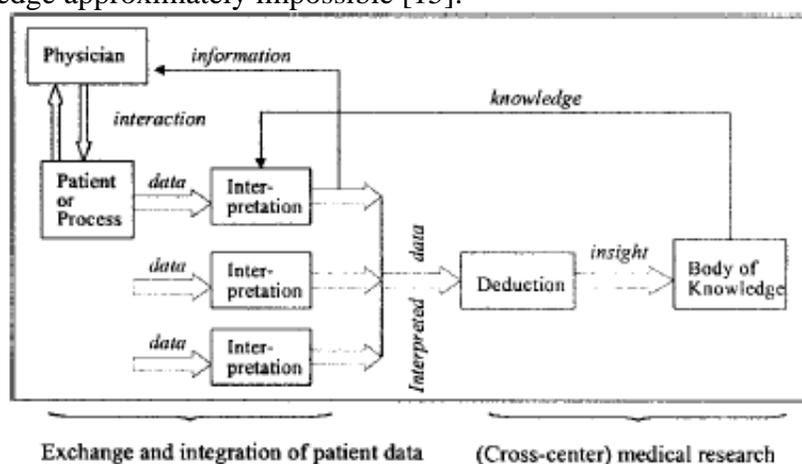


Figure 12. Data, information and knowledge (see [14])

Due to the all above issues, we wanted to draw the architecture of Tele-Wound™ system as Service Oriented Architecture and Web Services for Integration. It will make Tele-Wound™ architecture to integrate with business drivers, common business and technical goals, and recurring technical challenges Business driver.

Tele-Wound™ application is using common programming interface and interoperability protocol. By using service oriented development, Tele-Wound™ application will get following benefits”.

- **Reuse:** Through this feature the Tele-Wound™ application will create the service which can be used for other applications.
- **Efficiency:** By this benefit, Tele-Wound™ application gets the ability to make the services and new application using a mixture of new and old services. This service provides the facility to the application to focus on data to be joint instead of implementation underneath.
- **Loose technology coupling** Through this benefit the Tele-Wound™ application can model services separately of their implementation environment and creates messages that can be sent to any other services [27].

The services in Tele-Wound™ basically implemented as a software entity that relies on a single instance and interrelates with application through loosely coupled, message based communication model. The terminology behind the services is quite same as component based development, only the different is of defining the terms used in elements with in web services. Below figure will give you some idea about the concept used in web services

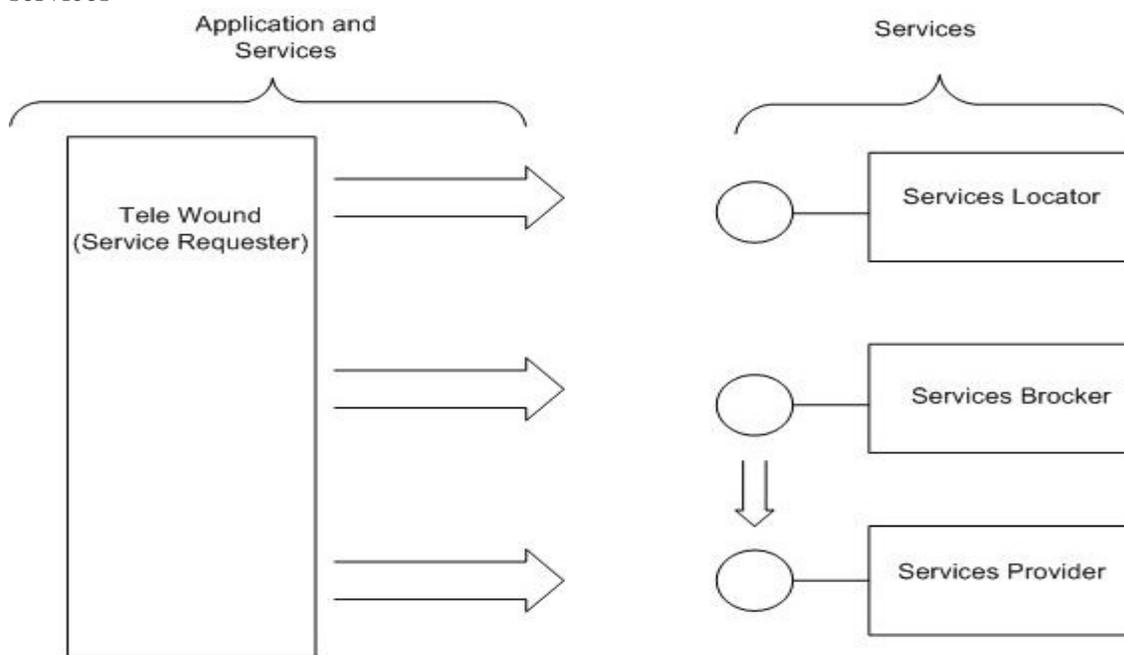


Figure 13. Services Terminology used in Tele-Wound™ [32]

- **Service:** A logical unit; which is used for published interfaces.
- **Service Provider:** A software unit that acquires a service requirement.
- **Service Broker:** A particular type of service supplier used in Tele-Wound™ application to pass service request to one or more providers.
- **Service Locator:** A particular type of service supplier used in Tele-Wound™ application to perform like registry and look up for service providers and service location.
- **Tele-Wound™:** The software that requests services. Normally it's a client which calls the service provider.

6.3 SECURITY IN TELE-WOUND

Security is an important issue when we talk about web services because these web services are publicly available. In order to maintain the security in SOAP for Tele-Wound™, we used well known standards of security. See section 3.5.2 for further details.

7. DISCUSSION

In this chapter we discuss different aspects of the chosen technology and web services. We also explain why we did certain choices and what problems we phased and some reflection upon them.

7.1 Proposed Solution

Extremely disperse nature of healthcare services makes it not easy to develop a single solution to serve all needs, so making the decision of chosen technologies for the proposed application is challenging and complex job, some time we face some difficulty and sometime not and we learn more thing. We started our work from the literature reviews from the previous related works and related researched. From the previous research, we came to know that there is no Telemedicine application that follow the services oriented architecture style and not design according to that standard. With the evolution of software design model, (e.g. SOA [26]), middle wares (e.g. COM and APIs), and development tools (e.g. Visual Studio .Net), the difficulties for the development of complicated applications are reduced.

A service-oriented architecture is basically a collection of services. These services communicate with each other. The communication between the services can involve data interchange. Now, the technology of Web services built on existing and emerging standards such as HTTP, Extensible Markup Language (XML), Simple Object Access Protocol (SOAP), Web Services Description Language (WSDL) and Universal Description, Discovery and Integration (UDDI). Designing an effective architecture especially for the problems seems like a promising solution. [32]

7.2 Reliability in web services

While designing Tele-WoundTM application as web based services, many questions raised in mind about reliability and performance of web services in Tele-WoundTM. Web service performance, reliability and scalability can be supplied by architecture on the basis of HTTP and SOAP, which is quite time consuming and untrustworthy. Initially, time consuming and untrustworthy must be distinguished, and then it must be understand that even reliable transport can give unreliable results at end of the session. When ever we design the application on a enterprise level, it's our duty to consider all functional and non functional requirement in order to support the business goal.

Let's consider the example, when using SOAP over HTTP it is always achievable to construct application-level protocols and connections that give extra abilities for message response and security. Consider the example in following figure:

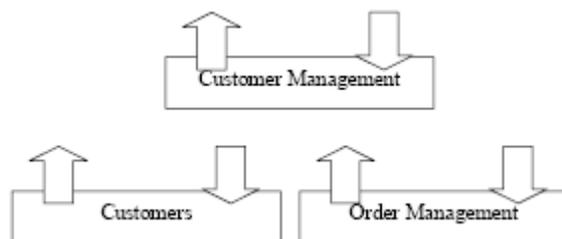


Figure 15: service dependencies [32]

In the above figure, all external clients communicate with customer management; it interrelates with two internal services. The Question is why we need the flexibility of HTTP and SOAP for the communication of internal services? If we suppose that performance and reliability is our major obligation for the communication between customer management and customers. So it is also possible to use a component Remote Procedure Call (RPC) communication (i.e. .NET remoting) that offers binary encoding and higher performance uniqueness. Some times it's essential to understand that through web services we have simple model, adaptable protocols, even though we are not restricted to use these choices. Barely as WSDL is compulsory in both SOAP and HTTP, it's also significant to supply requestors with extra choices[32].

In short security is the set of choices, collection of simple and inexpensive, the design objectives of performance, reliability and scalability came through certain set of decision

8. FUTURE WORK

For future Tele-Wound™ application, we want to achieve quality requirement. Also as a future challenge, we would like to take a challenging task in the field of Semantic interoperability [34]. Semantic Interoperability is a hot research topic around the world and after designing Tele-Wound™ as a SOA based, it is also possible to resolve the issue of semantic interoperability through several web services. So our next work of the research will about Semantic interoperability, below are the quality requirements which we targeted to achieve in our future work.

8.1 Usability

The usability of any given system means how easy it is for an end-user to operate it successfully. When designing a system human-computer interaction aspects among others should be taken into account. Systems are used by both beginner and expert users, and the design should take into account the needs of both. It should be straightforward and easy for users to operate a system successfully. The processes of the application system should be automated as much as possible. In case of attention is needed, the system should be supportive to guide the users on how to operate the other operations. In case of errors occur, friendly messages should be provided.

8.2 Availability and Reliability

The availability is a probability that the system is available for use at a given time, similarly availability of 0.998 means that in every 1000 time units the system is likely to be unavailable only twice [18]. The reliability of a system relates to the availability and is an important thing to consider when formulating requirement

The technical platform of Tele-Wound™ system should provide a reliable solution especially in doctor's end and it also ensure reliability both from patient's and doctors ends. This approach will improve technical quality support and will improve mobility. In the patients end we must assure that the devices that they are using for communicating purpose with the doctor's end should work properly. From the Doctors end perspective, the arrival of data, storage and retrieval of medical data and audio/video conversation should be performed in a reliable fashion. In case malfunctioning, an alert message or SMS message must be sent to mobile phones of Doctor's end technical staff explaining the cause of the problem and the possible solution. However any failure can result may lose the interest of users in the telemedicine service offered. At least the technical platform must assure that no harm is made to users in case of failure and no information is lost due to disaster within the system.

The technical platform should provide redundancy, in order to avoid single point of failures. When designing a network infrastructure, redundancy issues must be considered both from the patient's and doctor's perspective. The more levels of redundancy the technical platform has, the greater the source availability.

8.3 safety and Security

Security is a combination of confidentiality, integrity, and privacy. It determines the quality or to being protected from uncontrolled losses or effects. However the cost of implementing security measures and safeguards is considerably lower than the potential cost of a security incident. After all effective security pays and does not cost [19].

The patient's descriptive data to be sent from the patient's side to the doctor's side is considered to be confidential. It is important that those data is not read by unauthorized people, since it can violate patients' confidentiality and privacy and decrease the trust in using the telemedicine service. Hence it is of outmost importance to protect medical data from eavesdropping and possible modification by intruders or unauthorized persons. The technical platform must provide high level of confidentiality, privacy and data integrity. It is based on the estimation that network security cannot be a hundred percent guaranteed. Therefore, Tele-WoundTM should only remotely monitor the medical data obtained.

8.4 Efficiency

Efficiency relates to a system's use of resources to perform its tasks. Optimum efficiency is impossible to achieve, so we are concern about a pragmatic approach to minimizing the time taken to carry out each operation. Processing time usually has more effect on users' perceptions of a system. The platform should take into account the resources being used in the users' and the doctor's end, and try to minimize the load of work from the users' end. In some situations efficiency may differ from time of the day and availability of bandwidth, but the latency and performance of transferring medical data from users to doctor's end should be reduced as much as possible.

8.1 Scalability

Storage is an ability of to deal with an amount of data. The medical data obtained from the patient's end can be very important for clinical and statistical analysis in the doctor's end. The platform design must present storage planning for data backup and disaster recovery among others. It is expected that the number of patients using the service of Tele-WoundTM will increase by five to ten percent every year. It means that the system can perform functions efficiently and reliably even after three or four years of the technical platforms existence.

9. CONCLUSION

We have designed and implemented a service oriented architecture in our Tele-Wound[™] application in order to avoid the problem of interoperability, vendor lock-in and data interchange. The problems that appear when discussing the interoperability, data interchange and vendor lock-in among telemedicine application, then SOA suggested the better solution. This thesis presents the proposed architecture of Tele-Wound[™] application with the implementation of web services based on the experience gained with some previously developed applications. We tried to resolve the problem of interoperability in general; vendor lock-in and data interchange in our Tele-Wound[™] architecture, so that our Tele-Wound[™] users will not be dependent on single vendor and can also easily interchange the data between similar and dissimilar application by understanding same WSDL definition.

10. REFERENCE

- [1] Pronab Ganguly & Pradeep Ray, Software Interoperability of Telemedicine Systems, IEEE CNF 4-7 July 2000, available on
[<http://ieeexplore.ieee.org/iel5/6934/18631/00857717.pdf?tp=&arnumber=857717&isnumber=18631>]
- [2] Rick Craft, Introduction to the Telemedicine System Interoperability Architecture, Sandia National Laboratories, available at [<http://telemedicine.sandia.gov/>]
- [3] Cameron T. Howie, et al; Software Interoperability; Center for Integrated Facility Engineering, Stanford University, November,1996; available on
[<http://www.dacs.dtic.mil/techs/interodtitle.shtml>]
- [4] J. C. Wileden, Specification level interoperability, ACM digital library 1990, available on [<http://portal.acm.org/citation.cfm?id=100304&coll=portal&dl=ACM>]
- [5] Tayab D Memon¹, B.S Chowdhry, Mohammad S Memon, Potential of Telemedicine System: National Conference on Emerging Technologies 2004.
- [6] Thomas Jepsen, Interoperability for the NHIN, IEEE-USA MTPC, July 5, 2005
- [7]. Supavadee Aramvith and Ming-Ting Sun, MPEG-1 AND MPEG-2 Video Standards, 1999 Academic Press, available on
[<http://www.ee.eng.chula.ac.th/~supava/doc/mpeg12.pdf>]
- [8]. J. R Epps and W.H. Holmen, A New Very Low Bit Rate Wideband Speech Coder with a Sinusoidal High band Model, IEEE, 2001 available on
[<http://ieeexplore.ieee.org/iel5/7344/19923/00921079.pdf>]
- [9]. Magneto-optical drive [http://en.wikipedia.org/wiki/Magneto-optical_drive]
- [10]. David A. Clunie, Lossless Compression of Grayscale Medical Images - Effectiveness of Traditional and State of the Art Approaches, available on
[http://www.dclunie.com/papers/spie_mi_2000_compression.pdf]
- [11] JPEG compression, 2001 L. Leurs , available on
[<http://www.prepressure.com/techno/compressionjpeg.htm>].
- [12]. International Telecommunication Union, Broadband audiovisual communication systems and terminals, November 1996, available on [<http://www2.informatik.hu-berlin.de/~holz/H.310.pdf>]
- [13] Surekha Durvasula, Why Services-Oriented Architecture?, September 2006, available on [<http://www.soablueprint.com/whitepapers/SOAPGPart1.pdf>]

- [14] J. H. van Bommel and M. A. Musen, Handbook of Medical Informatics. Houten/Diegem: Bohn Stafleu Van Loghum, 1997.
- [15] Josef Ingenerf, "Telemedicine and Terminology: Different needs of context information", IEEE Transaction on I.T in biomedicine, Vol. 3, No. 2, June 1999, available on [<http://ieeexplore.ieee.org/iel5/4233/16626/00767084.pdf>]
- [16] Cynthia LeRouge and et al, Quality Attributes in Telemedicine Video Conferencing, Proceedings of the 35th Hawaii International Conference on System Sciences – 2002
- [17] P. Venkat Rangan and Daniel C. Swinehart, Software Architecture for Integration of Video Services in the Etherphone System, IEEE Journal In Communications. Vol 9. No 9. December 1991
- [18] Sommerville I. Software Engineering, 6th edition. Harlow, UK: Addison-Wesley, 2001, ISBN: 0321210263
- [19] Gritzalis D.A., Enhancing security and improving interoperability in healthcare information systems, Med Info (London), 1998
- [20] American Telemedicine Association, Telemedicine: A Brief Overview, Congressional Telehealth Briefing, Washington, DC, 1999. See also the web page at
- [21] Birrell, N.D., A Practical Handbook for Software Development. Cambridge University Press. ISBN 0-521-25462-0., 1985
- [22] DRM Associates (2002). New Product Development Glossary. Available on [<http://www.npd-solutions.com/glossary.html>], 2006
- [23] MSDN, Service Oriented Architecture, available on[<http://msdn2.microsoft.com/en-gb/architecture/aa948857.aspx>]
- [24] Nick Rozanski and Eoin Woods, Software System Architecture, Addison Wesley Professional 2005, ISBN-10: 0-321-11229-6
- [25] Uyar Ahmet et al, Service-Oriented Architecture for a Scalable Videoconferencing System, Pervasive Services, 2005. ICPS '05
- [26] Heather Kreger, Web services conceptual architecture, IBM white paper and software group, 2001
- [27] Greg Lomow at. al., Understanding SOA with Web Services, Addison Wesley December 2004, ISBN 0-321-18086-0
- [28] Web-based architecture for low-cost cost versatile telemedicine Jain, A.K.; Sharma, M.G.India Annual Conference, 2004. Proceedings of the IEEE INDICON 2004.

[29] Telemedicine Architecture, available on
[<http://www.bandmmedical.com/telemedicine.asp>]

[31] Design Research, available on
[<http://www.isworld.org/Researchdesign/drisISworld.htm>]

[32] Alan Brown and et. al., Using Service-Oriented Architecture and Component- Based Development to Build Web Service Applications, A Rational Software Whitepapers 2002

[33] Kevin Heubusch, “Interoperability, what it means and why it matters”, available on
[http://library.ahima.org/xpedio/groups/public/documents/ahima/bok1_028957.hcsp?dDocName=bok1_028957]

[34] Jeff Heflin and et. al., Semantic Interoperability on the Web, Extreme Markup Languages 2000, available on
[<http://www.cs.umd.edu/projects/plus/SHOE/pubs/extreme2000.pdf.%5d>]

[35] http://en.wikipedia.org/wiki/Vendor_lock-in

[36] SOA, Jeffrey Hasan, Services oriented architecture and Wikipedia

[37] SOAP, Duraisamy Sivaram, Whatis.com

[38] WSDL, <http://en.wikipedia.org/wiki/wsdl>

Note: All URL’s were last accessed on 28st August, 2007

APPENDIX A: ACRONYMS

API	Application Program Interface
CRUD	Create Read Update & Delete
GSM	Global System for Mobile Communication
EHR	Electronic Health Record
HTTP	Hypertext transfer Protocol
HTTPS	Hypertext transfer Protocol Secure
J2EE	JAVA 2 Enterprise Edition
JPEG	Joint Photographic Experts Group
MB/s	Mega Bit Per Second
MMS	Multimedia Message Services
MPEG	Moving Picture Experts Group
NHIN	National Health Information Network
PDA	Personal Digital Assistance
POP3	Post Office Protocol Version 3
RPC	Remote Procedure Call
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
TM	Trade Mark
UDDI	Universal Description Discovery Integration
WSDL	Web Service Description Language
WSME	Web Service Metadata Exchange
XML	Extensible Markup Language

APPENDIX B: QUESTIONAIRES RESPONSE

Questionnaire 1

Organization Name: _Mirus Information AB Lindesberg Sweden

Designation: _Project Manager R&D

Name: Rauf Agha

E-mail: rauf.agha@gmail.com

1. How many software systems you have developed in your organization?

Ans. 1) Less then 50 2) between 50 to 100 3) More then 100

2. How many Telemedicine and distributed system you have developed so far?

Ans. 1) Less then 50 2) between 50 to 100 3) More then 100

3. How many customers came back to you for further extension of their software product?

Ans. 1) Less then 25 2) between 25 to 50 3) More then 50

4. Have you developed any of the software system as a Services Oriented Architecture (SOA)?

Ans. 1) Yes 2) No

5) Are there any customers who are limited to your organization?

Ans. 1) Yes 2) No

6) What are the methods you provide to customers for interoperability of their product?

Ans.

- x Quick Integration
- x Different ways of getting input and output (in several formats)
- x Provide internal and external service interfaces of our own product
- x Create Adapters for the legacy applications to integrate them to our system

7) Do you think SOA can help to resolve the problems of Software Interoperability?

Ans. 1) Yes

2) No

Questionnaire 2

Organization Name: Northstar Technolgies, Inc

Designation: Software Engineer II

Name: Muhammad Jaffar Ramay

E-mail jaffar.ramay@globalnorthstar.com

1. How many software systems you have developed in your organization?

Ans. 1) Less then 50 2) between 50 to 100 3) More then 100

2. How many Telemedicine and distributed system you have developed so far?

Ans. 1) Less then 50 2) between 50 to 100 3) More then 100

3. How many customers came back to you for further extension of their software product?

Ans. 1) Less then 25 2) between 25 to 50 3) More then 50

4. Have you developed any of the software system as a Services Oriented Architecture (SOA)?

Ans. 1) Yes 2) No

5) Are there any customers who are limited to your organization?

Ans. 1) Yes 2) No

6) What are the methods you provide to customers for interoperability of their product?

Ans.

Data Merging

7) Do you think SOA can help to resolve the problems of Software Interoperability?

Ans. 1) Yes

2) No

Questionnaire 3

Organization Name: ISolution

Designation: Project Manager

Name: Moazam Nabi

E-mail moazam@isolution.org

1. How many software systems you have developed in your organization?

Ans. 1) Less then 50 2) between 50 to 100 3) More then 100

2. How many Telemedicine and distributed system you have developed so far?

Ans. 1) Less then 50 2) between 50 to 100 3) More then 100

3. How many customers came back to you for further extension of their software product?

Ans. 1) Less then 25 2) between 25 to 50 3) More then 50

4. Have you developed any of the software system as a Services Oriented Architecture (SOA)?

Ans. 1) Yes 2) No

5) Are there any customers who are limited to your organization?

Ans. 1) Yes 2) No

6) What are the methods you provide to customers for interoperability of their product?

Ans.

Interpretability is of great importance when developing software. We study first whether client wants open solution or not (usually open solutions are best for commercial products) we plan the project, analyse it and produce the solution.

7) Do you think SOA can help to resolve the problems of Software Interoperability?

Ans. 1) Yes

2) No