Software Interoperability of Telemedicine Systems : A CSCW Perspective

Pronab Ganguly & Pradeep Ray
School of Information Systems, Technology and Management,
University of New South Wales, Kensington,
NSW 2052, Australia
**Tel: 61 2 9385 5803, Fax: 61 2 9662 4061; Email: p.ganguly@unsw.edu.au*
+Tel: 61 2 9385 5890, Fax: 61 2 9662 4061; Email: p.ray@unsw.edu.au

Abstract

Computer Supported Cooperative Work (CSCW) provides a fusion of the understanding of organisational processes with communication technologies. Telemedicine involves an integration of networking technologies with health-care processes. Since different modalities of patient care require applications running on heterogeneous computing environment, interoperability is a major issue in telemedicine. Software interoperability provides two distinctly classified benefits – benefits for the users of the system and benefits to the development and maintenance of the system. Software interoperability between different applications can be modeled at different levels of abstractions such as physical interoperability, data-type interoperability, specification-level interoperability and semantic interoperability. Various mechanisms exist to resolve the problem at different levels. This paper presents the design issues of an interoperable CSCW system in a distributed health-care environment through an illustrative study in the area of telecardiology.

1. Introduction

Computer Supported Cooperative Work (CSCW) provides a fusion of the understanding of organisational processes with communication technologies. Groupware, workflows and distributed object technologies are being used to design and implement CSCW systems. This paper presents design issues for a CSCW system in distributed health-care environment. Communication technologists study this area as "telemedicine". Telemedicine involves an integration of networking technologies with health-care processes. Telemedicine applications are classified as telecardiology, telepathology, telepsychology etc. depending on the discipline of health-care. Most computerised health-care applications are traditionally stand-alone applications without integration. In order to integrate telemedicine services to distributed framework, various healthcare applications running on heterogeneous platforms must interact through software interoperability mechanisms. Software interoperability may be defined as "the capability with which two or more programs can share and process information irrespective of their implementation language and platform." [4].

The interoperability problem in telemedicine is manifested in patient monitoring, diagnostic, decision support and communication systems needed at the point of care. For example, one can consider a typical distributed health-care scenario with radiology application supported by Digital Imaging and Communication in Medicine (DICOM) standard, the pathology laboratory application running on Health Level Seven (HL7) standard , the pharmacy application running on a Unix server, and the patient repository in general physician's practice in Windows NT environment. So heterogeneity in networked computing environment is common in telemedicine. Software interoperability provides necessary mechanism to integrate these disparate but related resources into a single computational environment to achieve effective resource utilization.

The organisation of the paper is as follows. The next section Section 2 describes the distributed ECG analysis framework. Section 3 presents an overview of the interoperability problem. Section 4 briefly describes the component object model and ActiveX for healthcare framework. The section 5 details the prototype implementation and the concluding section describes the scope of further research areas.

2. The Process of Distributed ECG– A Telecardiology Application

2.1 Electrocardiogram and Traditional Framework:

The ElectroCardioGram (ECG) is signature of the heart and records voltage changes transmitted to the body surface from electrical events in the heart muscle. It provides direct evidence of cardiac rhythm and conduction as well as indirect evidence of certain aspects of myocardial anatomy, blood supply and function. Electrocardiography has been used for many years as a key, non-invasive method in diagnoses and early detection of coronary heart disease, which is the leading cause of mortality in many countries.

One of the aspects of telecardiology known as tele-electrocardiography deploys ECG machines to transmit ECGs over a network. Various research studies show that tele-electrocardiography diagnosis and ECG interpretation is simple, reliable and substantially cheaper in cost in comparison to conventional referral system [7]. ECG computer processing can be reduced to four principal stages:

- Data acquisition
- Encoding, transmission and storage
- Pattern recognition and feature extraction
- Diagnostic classification:

In the last two stages, a knowledge repository is used by human specialists. Presently, such knowledge-repositories are made available with ECG equipment in vendor proprietary form in a tightly integrated manner. It has been suggested that the quality of computer assisted ECG interpretation is perhaps better than that of general physicians, and such computerised interpretation is comparable with a review provided by a professional cardiology service [8].

A typical framework [6] for the implementation of a telelectrocardiography system with a commercially available ECG machines involves the following activities:

- Acquisition of raw ECG data
- Presentation of ECG data in a proprietary format to a proprietary knowledge repository
- Transmission of ECG data to health informatics network
- Transmission of ECG diagnostic from the knowledge repository to a health network

But this scheme places the knowledge repository with the ECG data acquisition instrumentation. The higher the quality of the knowledge repository, the higher is the cost of the ECG machine. This cost and accessibility can be improved by separating the ECG instrumentation from its present tightly integrated knowledge-base and sharing the knowledge-base over a network.

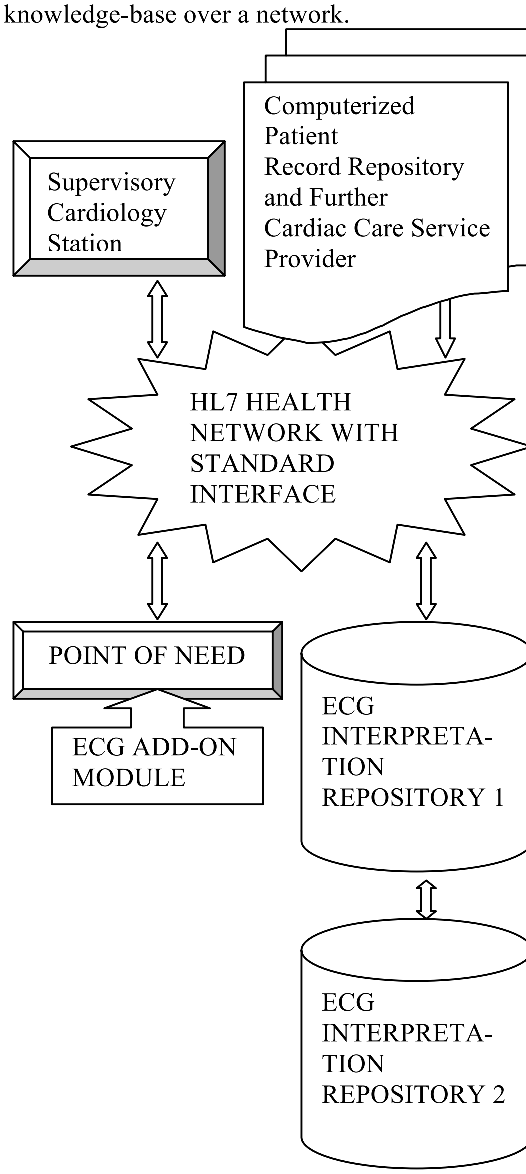

Figure 1: Distributed Framework for ECG Diagnostic System

2.2 Distributed Framework of ECG Analysis

Figure 1 shows a distributed framework for a tele-electrocardiography system. Here the point-of-need user with the help of front-end ECG machine accesses and invokes the distributed knowledge repository over the network. The interpretation repositories diagnose the ECGs on-line, and forward the ECG data along with multiple interpretations to the on-line supervisory cardiologist. For normal cases, results are distributed by the Supervisory Cardiology Station to the point-of-need and to the computerised patient record system over the network. For abnormal cases, further cardiac care service provider is also alerted. The various issues related to this distributed framework of ECG analysis and its advantages over traditional framework is reported in [6] .

In this framework, different users might use heterogeneous computing platforms, such as the point-of-need user might use Windows NT platform with a compatible set of application software whereas the Supervisory Cardiology Station might use Unix computing environment with a different set of application software. The other building blocks in the framework might use various other computing platforms as well. These different computing platforms need to interconnect, share data and act cooperatively with each other. Hence there is the need for interoperability in a telecardiology environment. In this application, the key issues related to interoperability are:

- How to support the interoperability between various platforms running different parts of the application?
- How to support interoperability with a pragmatic approach, using relevant standards for ECG data and knowledge repository?

Various issues related to healthcare informatics standards are reported in [8].

3. The Interoperability Problem

Different parts of a large telemedicine systems are from various vendors, who use different standards and information formats. These systems are also used by people with different levels of expertise and need. As a result, interoperability of these systems presents serious problem. Once the interoperability problem is solved, the development and maintenance of large telemedicine systems can be streamlined with data reuse, code reuse, application reuse and choice of an appropriate computing environment, using object-oriented technology.

A software interoperability problem can be viewed from following perspective [4]:

3.1. Physical Interoperability

In the first approach, the interoperability is achieved by physically transferring information through compatible electronic media such as floppy disks or magnetic tapes.

3.2. Data-type Interoperability

In the second approach, the focus is only on the content and structure of the information exchanged or shared. In our application, the format and presentation of the data (such as ECG results) will be such that the data stored in one program on one machine can be used in other machines and in other application programs. For example, all information may be stored in html for web based access.

3.3. Specification-level Interoperability

In specification-level interoperability, application sharing the data need not know the finer details of the aggregate data type information. For example, in our application, ECG data can be stored either in a single- dimension array or in a multi-dimensional array. Applications sharing the data need not know the finer details and treat the entity as a whole [4]. Object Management Group's (OMG) Common Object Request Broker Architecture (CORBA) and Microsoft Component Object Model (COM) provide means for achieving such an objective.

3.4. Semantic Interoperability

Applications can also exchange information at a semantic level. In this approach, a system is designed to use different abstract views of shared entities. This model inherently represents design intent, behavior and structured description of the entities. In case of our application, the different distributed applications should be able to share and use knowledge generated by other applications. Intelligent agents play an important role in this task. This is a subject of research worldwide.

4. Interoperability based on a Distributed Object Model

As discussed in Section 3.3, distributed object models, such as OMG Common Object Request Broker Architecture (CORBA), Sunsoft Java Remote Method Invocation (RMI), and Microsoft Distributed Common Object Model (DCOM) provide mechanisms for interoperability across systems using different operating systems, programming languages, and network protocols. This paper will now discuss our work based on Microsoft DCOM object environment. In this model [9], the interoperable software is encapsulated with object-oriented interfaces. DCOM is based on the Open Software Foundation's DCE-RPC specifications and uses Component Object Model (COM)[9],[10].

4.1 ActiveX for Healthcare

ActiveX for Healthcare is a collaborative work amongst Microsoft, Andover Working Group[2] and Health Level

Seven (HL7) [5] body. It is designed to provide easiest and less expensive interoperability between healthcare applications and systems.

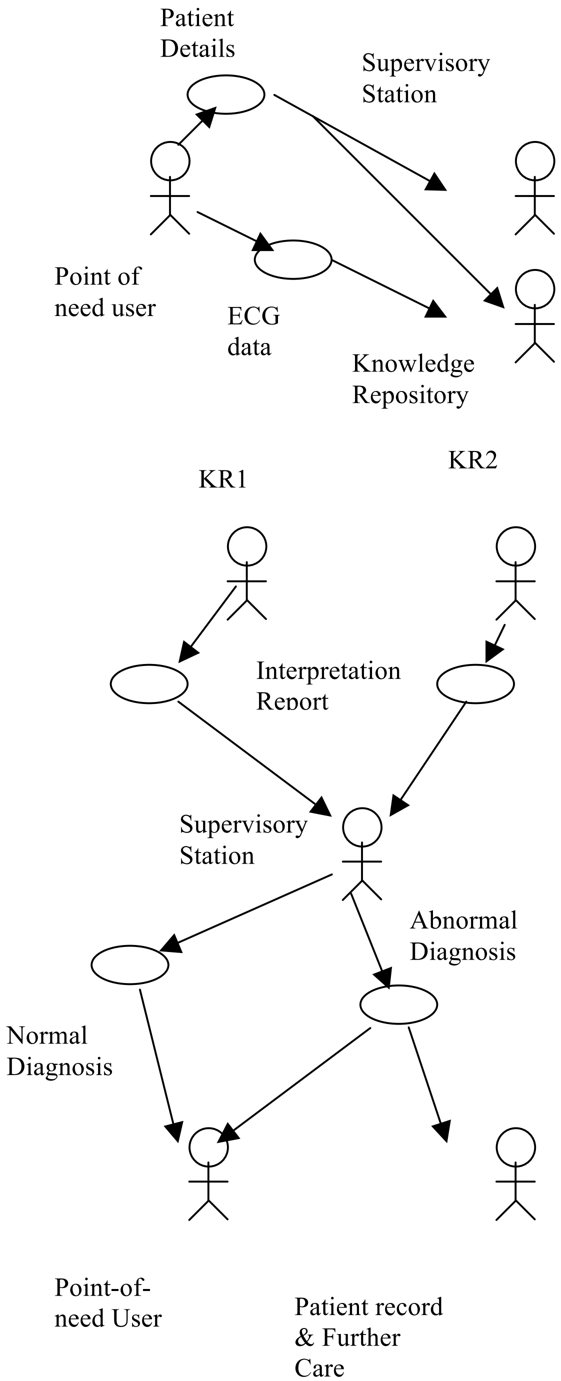

Figure 2 Brief Use Case Diagram for Distributed ECG Systems

Active X provides a framework involving DCOM communication environment, and different levels of infrastructure objects defined for the ease of implementation of a distributed object based systems. ActiveX is the key building block for interaction between applications and services – whether on the same machine, on a local network, or over the Internet. ActiveX components can be written in any language and deployed across multiple operating systems.

These freely licensed components and implementation guidelines are based on the work done by the Special Interest Group on Object Brokering Technologies (SIGBOT) of HL7 and specifies the HL7 message profiles and ActiveX-based version of the messaging application programming interface (API) developed by Andover Working Group. The relationship between Enterprise Communication Framework (ECF) developed by Andover Working Group [2] and ActiveX is represented by figure 4 [1].

The ActiveX for Healthcare (AHC) Messaging Components encapsulate HL7 2.3 message into objects and make it easy to connect various systems. Applications simply request the appropriate object (e.g., an admission or a medication order), add the appropriate data elements and transparently send the object to applications which are configured to receive it. A flexible architecture allows a migration path from legacy systems and interface engines to networked, component based applications. Applications can send messages via DCOM to another ActiveX application, or across a TCP/IP connection to an existing HL7 compatible application.

5. Prototype Implementation

5.1. Distributed Telecardiography in UML

The Unified Modelling Language (UML), considered to be the de-facto-standard object modelling language in industry, is a technique to specify, visualise and document the artifacts of an object-oriented system under development.

For this application , the following actors are identified:

- Point-of-need User (Creator)
- Knowledge Repository (KR) (Tracker)
- Supervisory Station (Manager)
  Patient Record & Further Care (Archivist)

The Use Cases for this application are Patient details, ECG data, Interpretation Report, Normal Diagnosis and Abnormal Diagnosis.

In UML, an actor is defined as the user of the system whereas A Use Case is a scenario that actor carries out. A brief description for each use case is as follows:

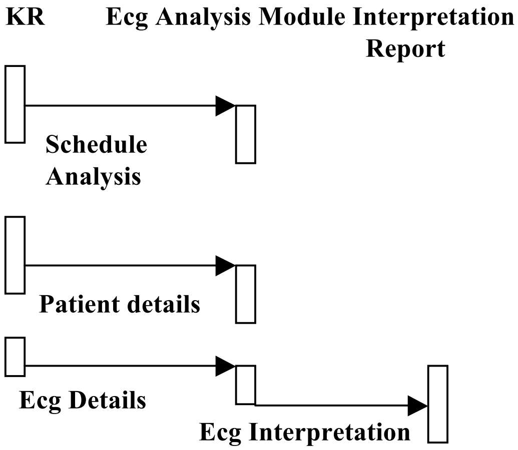

Figure 3: Brief Sequence Diagram for the Application

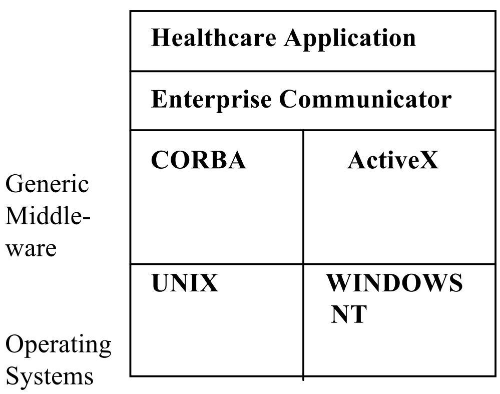

Figure 4: Relation between ECF and ActiveX

- Use Case: *Patient Details*

- Actors: Point-of-need User, Knowledge Repository, Supervisory Station

- Description: The point-of-need User acquires and submits patient details to Knowledge Repository and Supervisory Station. Both KR and Supervisory Station log this information.

- Use Case: *ECG data*

- Actors: Point-of-need User, Knowledge Repository

- Description: The point-of-need User acquires and submits the ECG data to Knowledge Repository. The KR logs and interprets this information.

- Use Case: *Interpretation Report*

- Actors: Knowledge Repositories, Supervisory Station

- Description: Once the ECG is interpreted, the ECG data and interpretation report are presented to Supervisory Station.

- Use Case: *Normal Diagnosis*

- Actors: Supervisory Station, Point-of-need User, Patient Record and Further Care

- Description: The Supervisory Station diagnoses the ECG and distributes the diagnosis to Point-of-need User and Patient Record &Further Care.

- *Use Case: Abnormal Diagnosis*

- Actors: Supervisory Station, Point-of-need User, Patient Record & Further Care.

- Description: Additionally Further Care is alerted to initiate action.

The Overviews of the Use Case and sequence diagram are shown in Figure 2 and 3.

The general scenario in ActiveX for HealthCare (AHC) for an application is[9]:

♦ Connect to the AHCMessenger component
♦ Request a blank message object (such as a HL7 2.3 admission or order)
♦ Add appropriate data elements
♦ Request message to be sent by the AHCMessenger component to a destination.
♦ Poll for incoming messages or handle callback requests.
♦ Disconnect from the AHCMessenger component

5.2. Prototype System

For our local area network prototype implementation, we have used Visual Basic version 5 with AHC version 1.1 in MS-Windows environment. Selected messages from HL7 2.3 have been incorporated in HL7 message factory. As this architecture allows for multiple message factories, other required messages can be implemented in a similar fashion.

5.3. Major Programming Modules

Following modules are used for the tele-electrocardiography application using the HL7 message objects wherever possible.

<u>Patient Demography</u> – This class used to exchange information about point-of-user need.

<u>ECG information</u> – This class is used for acquiring, defining and updating ECG information related to point-of-need user.

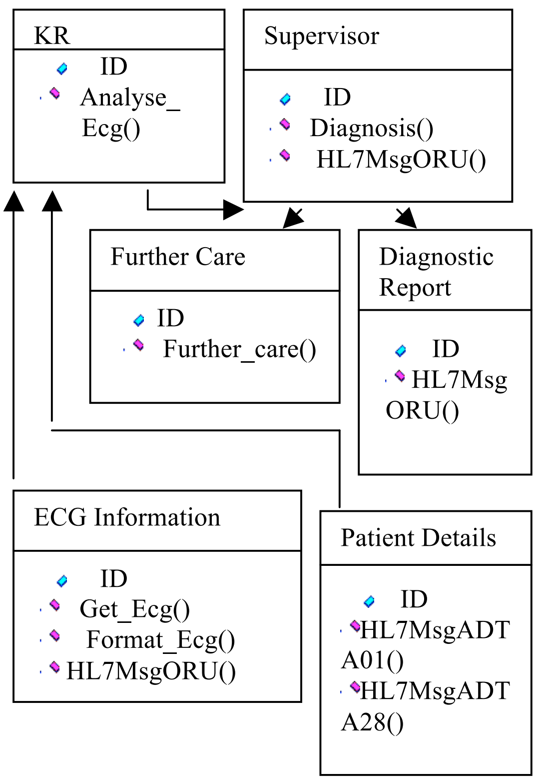

Figure 5: Major Modules Diagram for the Application

Knowledge Repository (KR) – This class is used for interpretation of ECG.
Supervisor – This class is used to generate diagnostic report of ECG.
Further care – This class is used to initiate further cardiac care service in case of abnormal ECG.
Diagnostic report – This class is used to generate diagnostic report.
Figure 5 shows the major programming modules and their interactions. In this diagram, the top box is the name of the class. The sign represents attribute whereas represents behavior. It may be noted that HL7MsgORU needs to be modified to include waveform.

5.4 Sample Code for Message Transmission

The sample code for sending a message in Visual Basic is given below [10]:

```
'Create an AHCMessageType object
        Dim msgType As New AHCMessageType
'Identify the type of message by filling in the properties of the object
        msgType.Version = "HL7 2.3"
        msgType.MessageTypeCode = "ADT"
        msgType.EventCode = "A28"
        'Declare a HCMessageEnvelope and call the CreateMessage method on 'HCMessageManager
        Dim msg As HCMessageEnvelope
        Set msg = msgman.CreateMessage(msgtype)
'Declare a message object of the type created in the previous 'step and get a handle to the 'message content.
        Dim a28 As HL7MsgADTA28
        Set a28 = msg.Content
'Fill in the message information.
        a28.PatientIdentification.Patient Name.Add(1)
        a28.PatientIdentification.Patient Name.Item(0).GivenName = "Viral"
        a28.PatientIdentification.Patient Name.Item(0).FamilyName = "Desai"
'Call the SendMessage method on the HCMessageManager interface passing it the 'HCMessageEnvelope interface. Either specify the destination application name or 'allow the HCMessenger object to use a default destination. Keep track of the message 'ID returned from the SendMessage method.
uuid = msgman.SendMessage(msg, "DestinationApp")
'Call the DoneWithMessage method for the message object.
        msgman.DoneWithMessage msg
        Set a28 = Nothing
        Set msg = Nothing
```

5.5. Experiences of Prototype Development

The following observations are made during the development of prototype system:

- There were memory leaks while the application is running for some time. This is a known problem in AHC version 1.1 which was subsequently rectified in later version.
- As blocking RPC is used to move information between networked machine, if the destination application is not available during sending of

information – the sending application sometimes hang. This was avoided by running the applications on different threads.

- AHC is still under evolution and not mature enough. Some new functionalities of HL7 are yet to be integrated into AHC. For example, we had to build our own messages to exchange ECG data – though the components follow general guidelines.

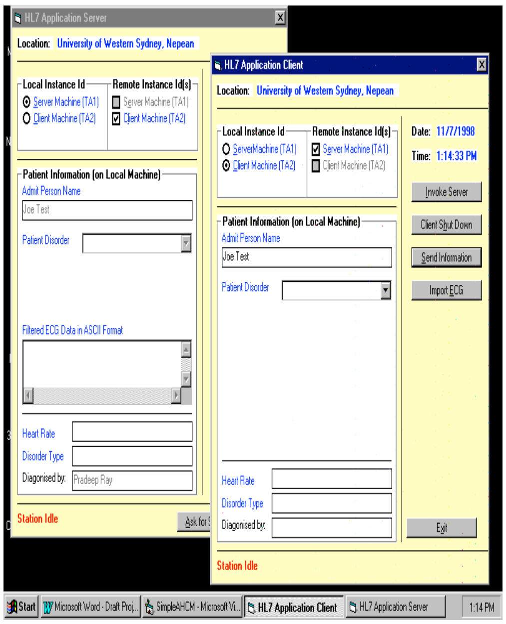

Figure 6: Snap-shot of a User Screen

- Since HL7 does not yet cover some application aspects, such as knowledge repository interface for this particular application scenario, we had to build our own message format as per the general guideline of HL7.
- Although DCOM is available on a few non-Microsoft platforms, it is realistic to conclude that at its present state Active X as an architecture is still evolving and is limited only to Microsoft platforms[11].
- Although the application model may be suitable for conveying the information, the system might need additional fine tuning to accommodate more appealing user interfaces.
- The system needs to tested more rigorously in a hospital environment.

5. Conclusion and Future Work

In the next stage of this research, we have undertaken the development of such systems in CORBA and Java platforms in collaboration with a major Australian hospital. Once these developments are completed, it will be possible to provide an interoperable telemedicine environment encompassing legacy, DCOM, CORBA and Java platforms. Such a system over wide area network then would be evaluated from following perspectives:

- Clinical adaptability
- Quality of service
- User acceptability
- Cooperative management of distributed intelligence
- Management of security

We are presently working on an evaluation strategy for this purpose.

6. References

[1] alSafadi Yasser et al; PACS/Information Systems Interoperability Using Enterprise Communication Framework; IEEE Transactions on Information Technology in Biomedicine; June 1998; Volume 2; Number 2; page 42-47

[2]Andover Working Group White Paper; Accelerating the movement towards standards-based interoperability in healthcare; http://www.interactive.hp.com/mpgawg/whitepaper.html ; 2nd January, 1999;

[3] Brandt S; The benefits of interoperability; http://www.bed.soton.ac.uk/sdis/interop.shtml; 10/02/99

[4] Cameron T. Howie, et al; Software Interoperability; Center for Integrated Facility Engineering, Stanford University, 25th November,1996; http:// www. dacs.dtic.mil/techs/interop/title.shtml; 01/09/1998

[5] Health Level Seven, Standard Version 2.3, Standards Australia, May 13,1997

[6] Ganguly P & Ray. P "Enterprise-wide Networking and Computing in Health-care Industry: A Case Study"; Proceedings of IEEE Enterprise Networking and Computing Conference (ENCOM'98), in conjunction with ICC'98, Atlanta USA, June 1998;

[7] Ganguly P & Ray P "Telecardiology over healthcare intranet"; Advances in Clinical Pathology; vol 2; No 2; 1998, 165-168; Proceedings of 4th European Congress on Telepathology; European Society of Telepathology, Udine, Italy, June 1998

[8] Ganguly P & Ray P "Telemedicine over Enterprise-wide Networks : A case study"; Proceedings of Globecom'98; Sydney, 8 –12 November, 1998

[9] Microsoft Corporation; The Component Object Model: Technical Overview; 2nd January, 1999

[10] Microsoft Healthcare Users Group, ActiveX for Healthcare Committee: ActiveX for Healthcare Messaging Components Programmer's Guide, Version 1.1, February 1998
[11] Sessions Roger; COM and DCOM: Microsoft's Vision For Distributed Objects; Wiley Computer Publishing; 1998; pages 102, 469 – 472.

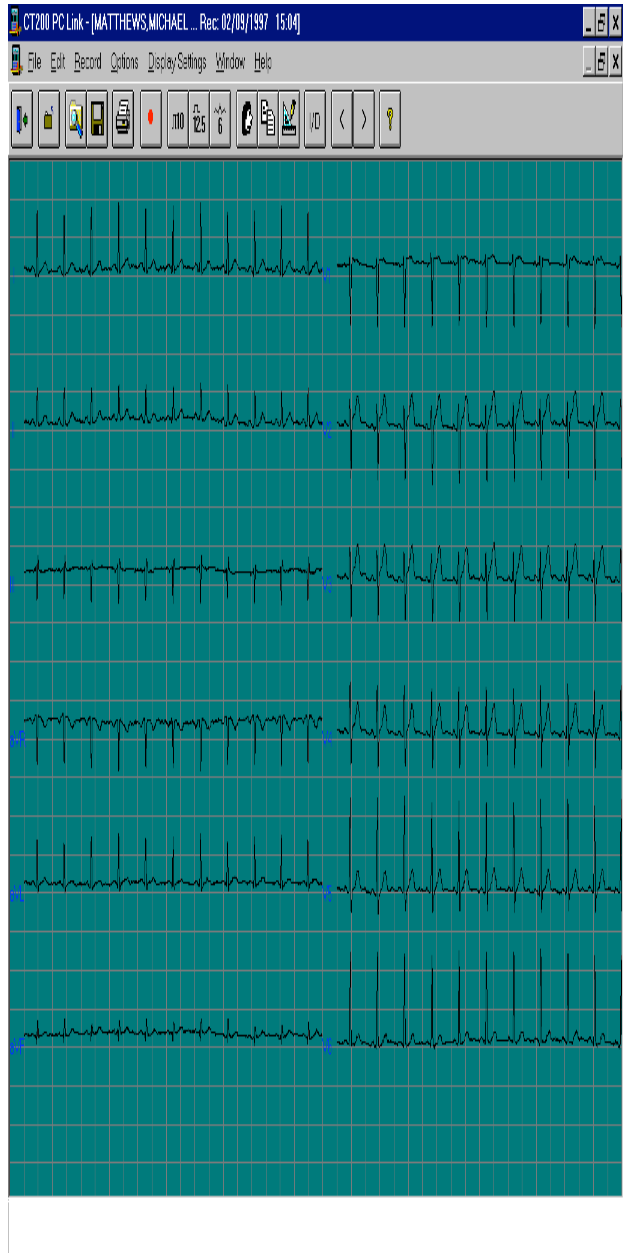

Figure 7: A Snapshot of ECG Screen