

A Methodology for the Development of Software Agent Based Interoperable Telemedicine Systems: A Tele-Electrocardiography Perspective

PRONAB GANGULY, M.Eng.Sc., and PRADEEP RAY, Ph.D.

ABSTRACT

Telemedicine involves the integration of information, human-machine, and healthcare technologies. Because different modalities of patient care require applications running on heterogeneous computing environment, software interoperability is a major issue in telemedicine. Software agent technology provides a range of promising techniques to solve this problem. This article discusses the development of a methodology for the design of interoperable telemedicine systems (illustrated with a tele-electrocardiography application). Software interoperability between different applications can be modeled at different levels of abstraction such as physical interoperability, data-type interoperability, specification-level interoperability, and semantic interoperability. Software agents address the issue of software interoperability at semantic level. A popular object-oriented software development methodology—unified modeling language (UML)—has been used for this development. This research has demonstrated the feasibility of the development of agent-based interoperable telemedicine systems. More research is needed before widespread deployment of such systems can take place.

INTRODUCTION

TELEMEDICINE INVOLVES the integration of information, telecommunication, human-machine, and healthcare technologies. Presently, these telemedicine services are provided in proprietary form (by different vendors) in isolated (different equipment for different services) environments. It is necessary to integrate these diverse telemedicine service mechanisms to reduce some of the inefficiencies that exist today. Because many of these services are software-based, we need to investigate standards and

techniques for the interoperability of software in telemedicine systems. 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.”¹ The interoperability problem in telemedicine is manifested in systems needed for patient monitoring, diagnostic, decision support, and communication systems. For example in a typical hospital environment, the radiology application is supported by Digital Imaging and Communication in Medicine (DICOM) standard whereas the

School of Information Systems, Technology and Management, University of New South Wales, Sydney, NSW, Australia.

pathology laboratory application could be running on Health Level Seven (HL7) standard.² While the pharmacy application is running on Unix environment, the patient repository in general physician's practice is in MS-Windows environment. So heterogeneity in computing environment is common in telemedicine. Software interoperability provides a mechanism to integrate these disparate but connected resources into a single computational environment to achieve effective resource utilization.

The next section describes distributed framework for tele-electrocardiography application and associated software interoperability mechanisms. The features of software agents and their potential application areas in telemedicine are also briefly outlined in this section. The subsequent section describes a unified modeling language (UML)-based methodology to develop a software agent-based system for the tele-electrocardiography application. This is followed by a description of a prototype system and its implementation. The concluding section discusses practical issues related to the deployment of a software agent based system in clinical environment and some unresolved issues for further research.

MATERIALS AND METHODS

Distributed ECG analysis: a tele-electrocardiography application

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, noninvasive method in diagnoses and early detection of coronary heart disease, which is the leading cause of mortality in many countries. Tele-electrocardiography deploys ECG machines to transmit ECGs over network. Various research studies show that tele-electrocardiography diagnosis and

ECG interpretation are simple, reliable and substantially cost-effective in comparison to conventional referral systems.³ A brief summary of studies indicating effectiveness of tele-electrocardiography in countries such as the United States, Canada, Italy, Israel, Greece, and other parts of the world is included in Table 1. The major benefits of tele-electrocardiography are^{4,5} faster access to diagnosis, improved quality of care, reduction in length of hospitalization and associated cost, and better patient management. ECG computer processing can be reduced to four principal stages: data acquisition; encoding, transmission and storage; pattern recognition and feature extraction; and diagnostic classification.

In the last two stages, a knowledge repository is used by human specialist.⁶ 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 is comparable with the review provided by a specialised cardiology service.⁴ The existing framework³ for the implementation of a teleelectrocardiography system 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; and transmission of ECG diagnostic from the knowledge repository to a health network.

However, 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 in-

TABLE 1. A BRIEF LIST OF EFFECTIVE TELE-ELECTROCARDIOGRAPHY PROGRAMS

<i>Location of patient</i>	<i>Location of medical service provider</i>
Inside a moving ambulance ³⁰	Hospital
Rural health center ³¹	Urban hospital
Community outreach ^{5,32,33}	Hospital
Patients' homes ³⁴	Physician's home

strumentation from its present tightly integrated knowledge-base and sharing that over a secure network.

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 distributed knowledge repositories over the network. The interpretation repositories diagnose ECGs online, and forward the ECG data along with interpretations to the online supervisory cardiologist. For normal cases, results are distributed over the network by the Supervisory Cardiology Station to the point-of-need and Computerized Patient Record Repository. For abnormal cases, more specialized cardiac care service provider is additionally alerted.

In this framework, different users might use heterogeneous computing platforms. For example, 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

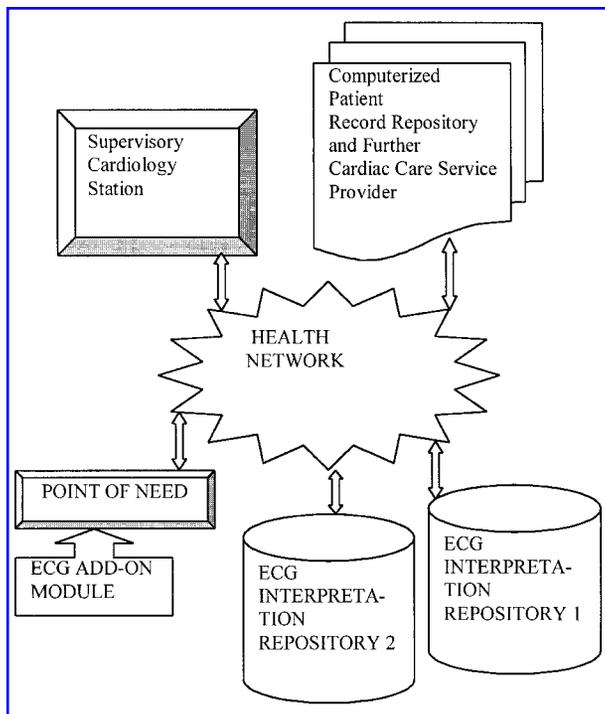


FIG. 1. Distributed framework for electrocardiogram (ECG) system.

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.⁷ In this application, the key issues related to interoperability are:

- Interoperability between various platforms as different parts of the application can run on heterogeneous platforms.
- Standard communication protocol between ECG data and knowledge repositories.
- SCP-ECG Standard Communication Protocol (from Comite European de Normalisation (CEN)/TC251) specifies the format of ECG data to be exchanged with interpretation repositories.⁶ It ensures that meaningful computerized interpretation is acceptable only if minimum data quality (in terms of sampling rate, no of leads, resolution, etc.) is assured.
- Conceptual model for ECG data storage and retrieval in terms of application areas. For example, ECG data varies depending on the testing environment such as rest, stress testing etc.⁶ Each mode requires specific formatting and interpretation details.
- Presentation of data in self-configurable graphical interfaces.⁶

The interoperability problem

Different parts of a large telemedicine systems are from various vendors. They use different standards and information format. These systems are also used by people with various need and different levels of expertise. As a result, interoperability of these systems presents interesting challenges. Once the interoperability problem is solved, the development and maintenance of large telemedicine systems can be streamlined with data reuse, code reuse and choice of various computing environment.

Thus, software interoperability can be viewed from following perspectives^{1,7}:

Physical interoperability. In this approach, the interoperability is achieved by physically transferring the information through electronic me-

dia such as floppy disks or magnetic tapes. In this case, information is transferred from one application to another by manually reentering the output data of the first application into other application. In our tele-electrocardiography application, this mode of interoperability means that data (in a known format) has to be transferred from one station to other by physical transportation of magnetic tape or floppy disk. This methodology is primitive, time consuming and expensive.

Data-type interoperability. In this approach, the focus is only on the content and structure of the information exchanged/shared. In our application, the format and presentation of the data (such as ECG readings) will be in such a manner so that the data stored in one program on one machine can be used in other machines and frequently in other application programs. In this case, standardized ECG data—for example either in single-dimensional array or multidimensional array in a particular format—must be exchanged between two applications. Standard such as Health Level Seven (HL7)²-based interoperability falls in this category. In this approach, the design of the application and its internal behavior are not important. Interface engines or protocol translators are necessary for communication between different standards. But the interface engines are often expensive, complex, and difficult to design. They are error prone. They require continuous and costly upgradation to accommodate new developments in standards and new features in medical devices. They are, however, capable of resolving problems of managing multiple point-to-point and proprietary interfaces.

Specification-level interoperability. In specification-level interoperability, applications 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 multidimensional array but the applications sharing the data need not know the finer details and treat the entity as a whole.¹ This approach is the extension of object-oriented or software component methodology to distribute objects across the network and still constitute a logical application. This is at a finer level of granular-

ity and tackles the technical issues of integration fairly well. In this approach the interoperability is addressed at specification level and the programs communicate at higher level of abstraction and increase the degree of information hiding. Object Management Group's (OMG) Common Object Request Broker Architecture (CORBA)⁸ and Microsoft's Component Object Model (COM) fall into this category.

Semantic interoperability. Semantic interoperability assumes that the components of the distributed application will have different expectations of the data. In this approach, a system is designed to use different abstract views of shared entities. In case of our application, the built-in knowledge resources in the system can exchange ECG data in a program-independent manner. This model inherently represent design intent, behavior, and structured description of the entities. Usually human intelligence is required to assess semantic information. Trends are emerging to automate this process. Some components of this trend are ontology, repositories and computer-aided software engineering tools.⁹ Software agent-based technology support semantic interoperability.¹⁰ A software agent exchanges knowledge with other agents through intermediate mechanism called "facilitator." Agent systems use a single agent communication language (ACL) to achieve interoperability. Knowledge interchange format (KIF) is a standard for defining the content of messages within the ACL structure. KIF is designed for interchange of knowledge among disparate programs.

We will briefly review the software agents in the context of telemedicine in the next section.

Software agents and telemedicine

Software agent is a proactive object. A software entity is an agent if it has¹¹:

- the data and code encapsulation of a software object;
- its own thread of control (active);
- the ability to execute autonomously without being invoked externally.

Thus, "an autonomous agent is a system situated within and a part of an environment that senses that environment and acts on it, over

TABLE 2. APPLICATION AREAS OF AGENT TECHNOLOGY IN TELEMEDICINE

User assistant	<ul style="list-style-type: none"> • User profile learning systems • Personal digital assistant 	Remote diagnostic assistance
Information retrieval	<ul style="list-style-type: none"> • Directory services • Database enquiry • Information brokerage 	Patient demography and medical expertise distribution
Entertainment	<ul style="list-style-type: none"> • Games • Videoproduction 	Multimedia image exchange
Service management	<ul style="list-style-type: none"> • Multimedia • Electronic commerce 	Health insurance and consultation fees
Business management	<ul style="list-style-type: none"> • Network management • Workflow management 	Computer supported collaborative work amongst healthcare entities
Manufacturing management	<ul style="list-style-type: none"> • Virtual factory management • Factory manufacturing robot 	Remote control of diagnostic laboratory
Service robotics	<ul style="list-style-type: none"> • Office delivery robots • House cleaning robots 	Remote cleaning and monitoring of lab and healthcare facilities

time, in pursuit of its own agenda and so as to effect it senses in the future."¹¹ For example, human beings are more sophisticated agents, whereas a thermostat is an oversimplified version of agent. A spell checker for a word-processor is not a software agent but a spell checker that monitors typing and corrects on the fly might be an agent.¹¹ Thus all software agents are programs, but not all programs are software agents.

Some of the applications areas for agent technology and their applicability in telemedicine are given in Table 2.¹² The special feature of agent-based technology and their relevance to telemedicine is summarized in Table 3.^{13,14}

Healthcare informatics and telemedicine researchers have started investigating this technology for real-life applications. Our research indicates that two such projects are Ginkgo by IBM and Multi Agent Network for Telemedi-

cine and Hypermedia Authoring (MANTHA) by University of Udine, Italy.

Ginkgo¹⁵ is written in Java and is being used to develop a decision support system named Physician's Assistant. This system can learn about drugs a doctor prescribes for a given situation and recalls the doctor's own practice pattern, providing an intelligent default suggestion. IBM is also working on a prototype system named Physician's Consultation that uses Ginkgo to intelligently mediate communication among doctors and make recommendations based on what recognized experts have done in a particular situation.

The goals of MANTHA¹⁶ are to allow authoring, publishing, and recording of hypertext documents. Because telemedicine involves collaborative work on multimedia medical data, this project aims to develop a federation of agents (called agency) skilled for telemedicine

TABLE 3. FEATURES OF AGENT TECHNOLOGY RELEVANT TO TELEMEDICINE

<i>Issues</i>	<i>Telemedicine application</i>
Bandwidth limitation can be improved through agent technology	Multimedia data exchange in telemedicine applications makes them suitable for agent technology
Network congestion can be better handled by agent technology	Document prefetch and higher quality of service can be achieved in telemedicine application
Fault tolerance and robustness is available with agent technology	As agent technology reacts dynamically to adverse conditions, it is suitable for development of fault tolerant distributed complex systems in telemedicine
Support for electronics service is provided by agent technology	As agent technology represents the intentions, desires and resources of the participants, it can be used to build electronic commerce for the telemedicine market
Heterogeneous computing environment is supported in agent technology	As distributed telemedicine application should be computing platform neutral, agent technology could be effectively used

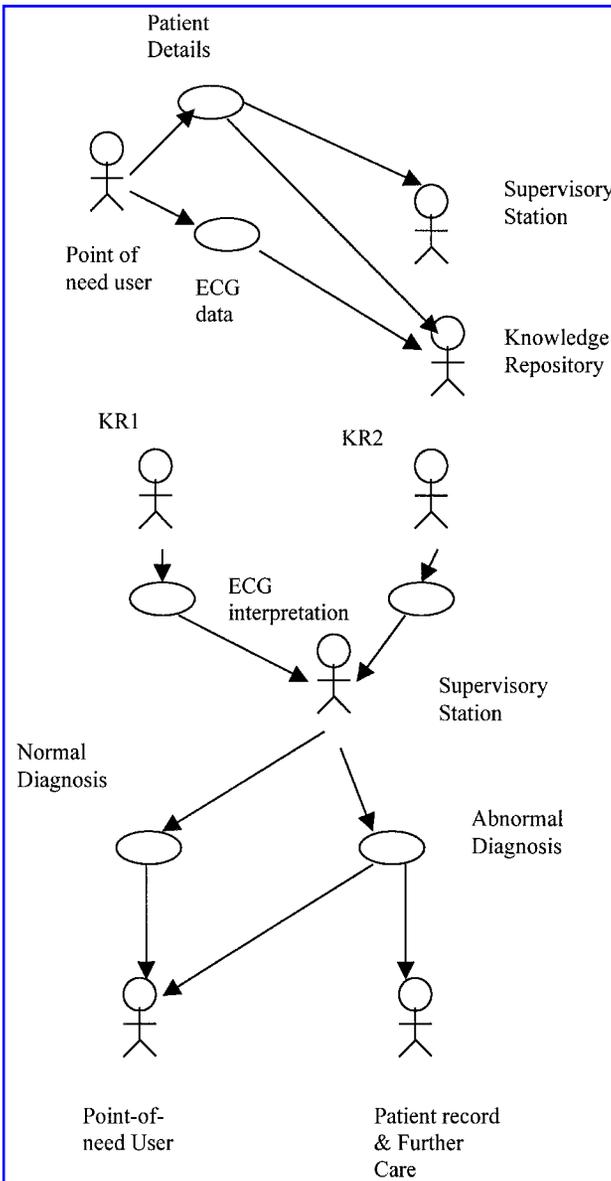


FIG. 2. Use case diagram for distributed electrocardiogram (ECG) system.

purposes. The agency manages hypermedia archives and supports collaborative authoring and network interactions.

RESULTS

Design of agent-based system

The techniques of development of the agent-based systems include conceptual analysis, role interactions, formal analysis, and implementation design.^{14,17} The discipline of information systems development specifies method-

ologies (or strategies) for the effective development of software systems. Various methodologies based on data-oriented (Information Engineering), function-oriented (Structured System Analysis & Design Methodology [SSADM]) and object-oriented UML techniques are available.¹⁸ Our research has used distributed object-based design framework. The proposed methodology for the development of software agent-based tele-electrocardiography system is based on UML. UML is a *de facto* standard in software system development. It has been extensively used by the software industry and adopted by Object Management Group (OMG). The UML specifies the following steps for the development of object-based system¹⁹:

- Analysis Use Case Model—Use Cases help in the specifications of the requirements of the system. It involves actors and their interactions. An actor is defined as the user of the system whereas a Use Case is a scenario that actor carries out.
- Conceptual Model—in UML a conceptual model is explained with a set of static structure diagrams. It does not show the details of operation.
- System Behavior Model—Collaboration diagrams explain object interactions in a graph or network format whereas system sequence diagram illustrates events from actors to systems.
- Class Model—this involves design of class diagrams
- Design State Model—this involves state diagrams for classes.

The next section summarizes the system design based on UML Use Cases and sequence diagrams.

Distributed tele-electrocardiography in UML.

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)

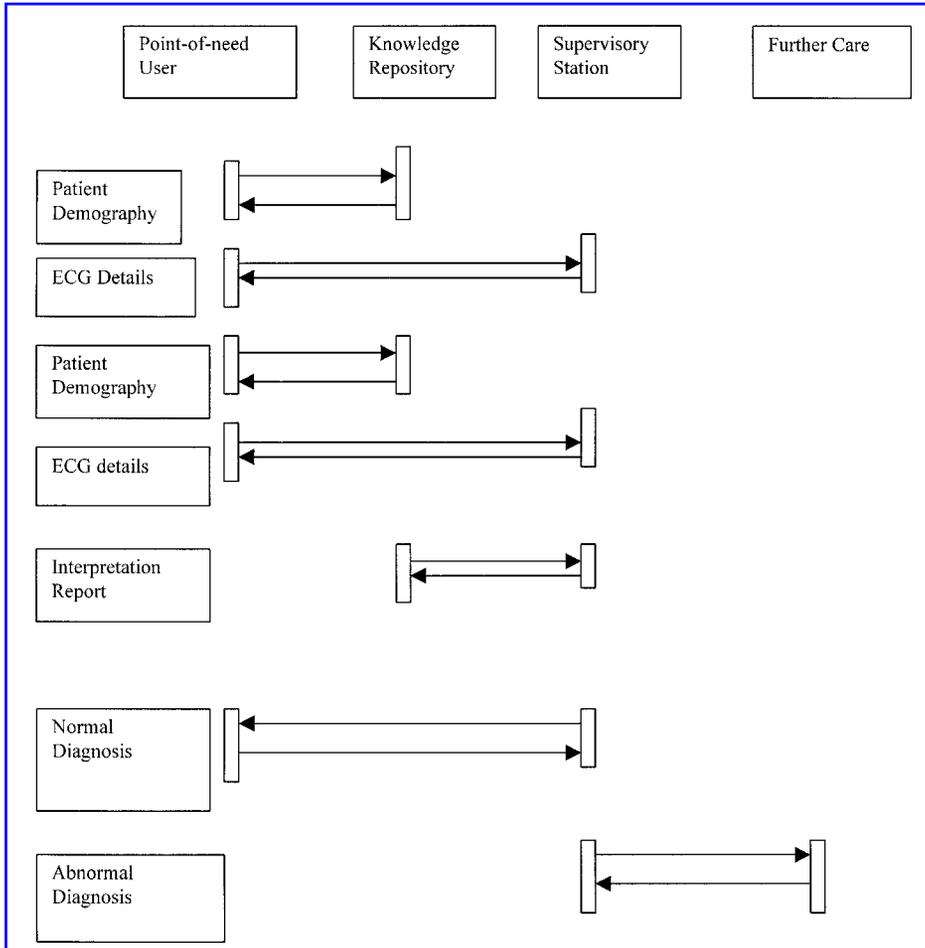


FIG. 3. Sequence diagram for the application.

The Use Cases for this application are patient details, ECG data, interpretation report, normal diagnosis, and abnormal diagnosis.

Figure 2 presents a brief Use Case Diagram for the application and Figure 3 presents a brief sequence diagram of the application.

A brief description for each use case is as follows:

- Use Case: Patient Details
 - Actors: Point-of-need user, Knowledge Repository, and Supervisory Station
 - Description: The Point-of-Need user acquires and submits patient details to Knowledge Repository and Supervisory Station. Both Knowledge Repository 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: ECG 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

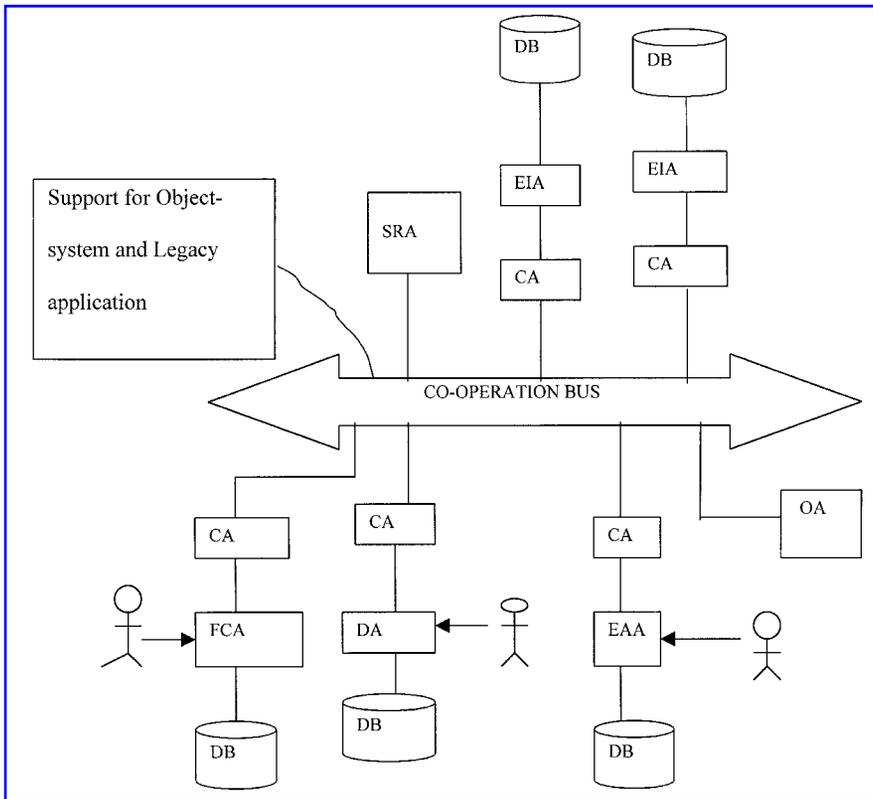


FIG. 4. Architecture schema for distributed electrocardiogram (ECG) analysis agent framework. DB, database.

—Actors: Supervisory Station, Point-of-need User,

- Patient Record & Further Care.

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

Design of agent-based system from use case scenario. Use cases define how users interact with a system with each actor as a user in a certain role. For a full requirement analysis, use cases must be specified for each scenario of the application. At this level of detail each use case can be mapped to an actor's or agent's plan. When multiple actors appear in one use case,

they collaborate and negotiate. In the agent-based system, the actors and resources become the agents and their negotiation becomes a co-ordination protocol. With this correspondence,²⁰ the application in UML is mapped into agent-based system as shown in Figure 4.

Distributed ECG and semantic interoperability

A brief list of system solutions for existing tele-electrocardiography programs is given in Table 4. But in a distributed healthcare informatics system, the end-users enjoy the benefits of a single virtual health enterprise. This type

TABLE 4. SYSTEM SOLUTIONS FOR EXISTING TELE-ELECTROCARDIOGRAPHY PROGRAMS

<i>Method of information exchange</i>	<i>Interoperability paradigm</i>
ECG data & interpretation by fax ³⁵	Physical/Data-type
Store and forward of ECG file	Data-type
PC based networked system ³⁴	Data-type
Client-server architecture ³⁶	Data-type
Notebook & wireless transmission ³⁰	Data-type
Open European Electrocardiological Data Interchange Project ⁶	Data-type
Active X-based Project ³⁷	Specification-level

TABLE 5. METRICS FOR INTEGRATION REQUIREMENTS VERSUS INTEROPERABILITY PARADIGMS

Integration Requirements	Interoperability paradigms			
	Physical	Data-type	Specification-level	Semantic
Presentation—common appearance and behavior	Yes	Yes	Yes	Yes
Data—Perception of database as single entity	No	No	Yes	Yes
Communication—standardization of exchanged messages	No	Yes	Yes	Yes
Interworking of applications	No	No	Yes	Yes
Unified medical concept	No	No	Yes (limited)	Yes

of information relies on seamless integration of geographically distributed heterogeneous components. The following dimensions indicate a measure of integration²¹:

- Presentation integration assures that different tools used by the applications have a common appearance and behavior.
- Data integration relates to perception of information as if it is stored in a single database.
- Communication integration refers to standardization of the formats of the messages exchanged between applications.
- Objective of control integration determines sequence of interworking of different application.

- Semantic integration refers to unified medical concept so as to enable the applications exchange information meaningfully.

Table 5 presents a comparison of integration requirements against software interoperability paradigms. In our distributed ECG-analysis framework, ECG data format is under standardization with SCP-ECG. But this format may not be universally accepted. ECG-storage model varies as per application areas such as stress testing or resting condition. Similarly, presentation of ECG on graphical user interfaces are difficult to standardize as they are dependent on ECG lead configuration. In CORBAMED (Domain Task Force for healthcare informatics),²² relevant interoperability services such as Person Identification Service, Lexical Query Service, etc., are either developed or under development. But SCP-ECG data format, ECG storage, and presentation formats are yet to be mapped in interoperability services. Thus, it seems that in its present state, object-brokers are limited in their expressiveness.²³ Again the framework of distributed ECG analysis needs to integrate with distributed object-oriented systems as well as legacy application. An empirical evaluation²⁴ of some of the distributed systems communication protocols is presented in Figure 5. This evaluation shows that the development and management complexity is lowest for agent-based systems although it may not have high distribution complexity. On the other hand, CORBA-based system has a slightly higher level of development and management complexity but they would support higher level of distribution com-

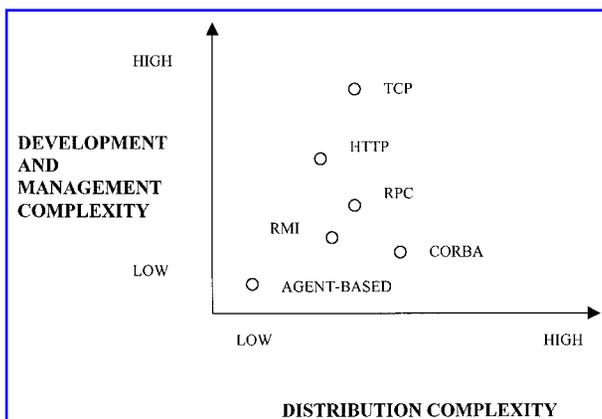


FIG. 5. Empirical evaluation of some of the distributed systems communications protocols. TCP, transport control protocol; HTTP, hypertext transfer protocol; RPC, remote procedure call; RMI, remote method invocation; CORBA, common broker architecture.

TABLE 6. DEVELOPMENT TOOLS FOR AGENT-BASED SYSTEMS

<i>Features</i>	<i>Development tools</i>
Agent as a single reactive process Agents as travelers	IBM's Agent Building Environment—C++ <ul style="list-style-type: none"> • General Magic's Telescript • IBM's Aglet—Java objects • General Magic's Odyssey—Java version of Telescript • ObjectSpace's Voyager—Java-based Object request broker for mobile agents
Agents as members of a community	Gensym's Agent Development Environment (ADE)
Agents as intelligent process	<ul style="list-style-type: none"> • DMARS • D-Muse • Agent Building Shell (ABS)
Infrastructure of typed-message agents	JATLite

plexity as well. Therefore, a combination of agent-based and CORBA (and similar)-based architecture seems to be the most effective means of solving our problem. Therefore our methodology for the development of agent based tele-electrocardiography system would consist of:

- UML-based Use Cases and Sequence diagrams;
- Mapping of UML-based system to agent-based system;
- Selection of suitable agent platform supporting CORBA, DCOM and EJB (Enterprise Java Beans);
- Implementation design based on platform specific aspect;
- Coding and implementation;
- Development and roll-out of the system in a chosen telemedicine environment in a cautious manner; and
- Evaluation of the system from the point of views of benefits and drawbacks.

This loop can be iterative starting from the first step depending on the feedback. This article illustrates the above methodology. The first two steps have already been explained in previous section. The next section will show the feasibility of implementation phase with a proof-of-concept illustration.

Proof-of-concept implementation

For implementation of prototype, there is a need for a framework that provides basic infrastructure and communication protocol for im-

plementation of interoperable software agents. The development frameworks available for agent-based systems are listed¹⁴ in Table 6. Although we started with JATLite²⁵ agent platform, ObjectSpaces's interoperable framework Voyager is preferred for prototype implementation for the following reasons:

- Seamless integration of agent framework with distributed computing;
- Supporting communication architecture for CORBA, COM, RMI (Remote Method Invocation), and EJB (Enterprise Java Beans); and
- Better support for security with secure channels.¹³

In Voyager, secondary objects, known as facets, can be attached to primary objects in run-time. The primary object with its facets is known as aggregate. An agent is made up of one or more objects. An object can be turned into an agent by¹³:

- Agent.of() to obtain object's facet.
- IAgent Interface.

In general, the agents and repositories are being developed with Voyager's special classes and interfaces. They include Agent, Messenger and Database.²⁶

The proposed architecture shown in Figure 4 is under implementation. This approach involves development of a collaborative community where each agent is specialized in specific services as below^{27,28}:

- Cooperation Agent (CA) coordinates high-level cooperation;
- Ontology Agent (OA) allows communication between various types of agents;
- Semantic Router Agent (SRA) stores ontological agreements;
- ECG Interpretation Agent (EIA) presents the ECG to Knowledge Repository and obtains automated ECG interpretation;
- ECG Acquisition Agent (EAA) obtains the ECG data at point of need;
- Diagnostic Agent (DA) manages ECG diagnostic in the Supervisory station; and
- Further Care Agent (FCA) manages the further cardiac care services.

In the above approach, the burden of interoperation is borne by software agents and communication facilitators and thereby the load on application programmers is reduced.²³

As heuristic evaluation for telemedicine systems is found to be effective,²⁹ such an evaluation based on functional, technical, organizational, clinical, and economic factors is being planned.

DISCUSSION

Various types of interoperability such as physical, data-type, etc., have been discussed. Our research has addressed the issue of semantic interoperability. A methodology based on UML has been proposed for the development of distributed, object-oriented, agent-based interoperable telemedicine systems. A proof of concept implementation plan based on ObjectSpace's Voyager has been discussed. All concepts have been illustrated with a distributed ECG system. We feel the same idea can be extended to other types of telemedicine systems. Although our initial analysis looks promising, it is necessary to evaluate these designs in a wide variety of clinical environments before extensive deployment of such systems.

Software agent-based technology is developing rapidly. In order to utilize this technology effectively, more research is required into issues³³ such as scalability, crash recovery, inconsistencies, and security. Clinically effective telemedicine applications built on this technol-

ogy will offer an efficient mechanism for distribution medical expertise over the computer communication networks.

REFERENCES

1. Howie TC, Kunz JC, Law KH. Software interoperability: center of integrated facility engineering. Stanford University; 12/25/1996.<http://www.dacs.dtic.mil/techs/interop/title.shtml>
2. *Health Level Seven*. Standard Version 2.3; Standards Australia; May 13, 1997.
3. 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 GA: 1998.
4. Ganguly P, Ray P. Telecardiology over healthcare intranet. *Proceedings of 4th European Congress on Telepathology; European Society of Telepathology*, Udine, Italy, June 1998. *Adv Clin Pathol* 1998;2:165-168.
5. Shanit D, Greenbaum RA. Towards a comprehensive telecardiology monitoring centre for community-based services. *J Telemed Telecare* 1997;3(Suppl 1):60-62.
6. Rubel P, Fayn J, Zywiets C, Willems JL, Assanelli D, Malossi C, Florin C. *Proceedings of Health Telematics '95*; 1995, pp. 381-386
7. Brandt S. The benefits of interoperability <http://www.bed.soton.ac.uk/sdis/interop.shtml>; 10/02/99.
8. Orfali R, et al. *Instant Corba*. Wiley Computer Publishing, 1997, pp 22-24.
9. Heiler S, Miller RJ, Ventrone V. Using metadata to address Problems of Semantic Interoperability in Large Object Systems. <http://www.computer.org/conferences/meta96/rmiller/paper.html>; Accessed on 12/29/1999
10. Rosenthal A, Sciore E. Description, conversion, and planning for semantic interoperability. <http://www.mitre.org/pubs/data.mgt/papers/sem.pdf>. Accessed on 12/29/1999.
11. Franklin S, Graesser A. Is it an agent, or just a program? A Taxonomy for autonomous agents. *Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages*. Springer-Verlag; 1996.
12. Azarme N, et al. FIPA application types, Rev 2.0; <http://drogo.cselt.it/fipa/yorktown/nyapplication.htm>; 12/2/99
13. Mahmoud QH. *Distributed programming with Java*; Manning Publications Co, 1999, pp 238-239; 240-241; 255-289.
14. Van Dyke P. Practical and industrial applications of agent-based systems, Industrial Technology Institute, 1998; <http://www.iti.org/~van/;03/04/99>.

15. Bareker D. The Secret Agent Man. <http://www.botsport.com/pcai/article2.html>; 10/2/99
16. MANTHA: University of Udine, http://www.telemud.uniud.id/mantha/ptj_descr.html; 2/15/99.
17. Bui T, et al. Software agents for telemedicine; <http://www.vacets.org/vtic97/txbui2.htm>; 9/04/99.
18. Avison DE, Fitzgerald G. *Information Systems Development: Methodologies, Techniques and Tools*. The McGraw-Hill Companies, **1997**;280–294.
19. Larman C. *Applying UML and patterns: An introduction to object-oriented analysis and design*. Prentice-Hall, Inc., **1997**, 144.
20. Kendall EA, et al. A methodology for developing agent based systems for enterprise integration: Modeling and methodologies for enterprise integration. Peter Bernus; Chapman & Hall, **1996**, pp 333–344.
21. Degoulet P. Multimedia workstations: Electronics assistants for the health care professionals. In: van Bemmel JM, Mccray AT, eds. *Yearbook '96 of Medical Informatics* Stuttgart: Schattauer; **1996**, pp 65–75.
22. OMG's CORBAMED Task Force; <http://www.omg.org/corbamed.htm>; Accessed on 12/10/1997.
23. Genesreth RM. An agent-based framework for interoperability, software agents. Jeffery M. Bradshaw; AAAI/MIT Press; 1997, pp 317–345.
24. Mountzia MA, Benech D. Communication requirements and technologies for multi-agent management systems. *Proceedings of DSOM 97*, Sydney, **1997**.
25. JATLite Introductory FAQ; <http://cdr.stanford.edu/ProcessLink/papers/JATL.html> 2/15/99.
26. Falchuk B, Karmouch A. Visual modeling for agent-based applications. *Computer* **1998**;31–37.
27. Leclercq E, Benslimane D, Yetongnon K. ISIS: A semantic mediation model and an agent based architecture for GIS interoperability. Proceedings of 1999 International Database Engineering and Applications Symposium, <http://church.computer.org/proceedings/ideas/0265/026500087.pdf>. Accessed on 12/29/1999.
28. Ramesh V, Canfield K, Quirologico S, Silva M. An intelligent agent-based architecture for interoperability among heterogeneous medical databases. <http://hsb.baylor.edu/ramsower/ais.ac.96/papers/ramesh2.html>. Accessed on 12/23/1999.
29. Lathan EC, Sebrechts MM, Newman DJ, Doarn CR. Heuristic evaluation of a web-based interface for Internet telemedicine. *Telemed J* 1999; 5:177–185.
30. Giovas P, Papadoyannis D, Thomakos D, Papazachos G, Rallidis M, Soulis D, Stamatopoloulos C, Mavrogeni S, Katsilambros N. Transmission of electrocardiograms from a moving ambulance. *J Telemed Telecare* **1998**;4(Suppl 1):5–7.
31. Chin TL. Straight from the heart. *Health Data Manage* **1998**;6:108–109.
32. Bertazzoni G, et al. Telecar: An Italian telecardiology project. *Telemed Telecare* **1996**;2(3):132–135
33. Rissam IIS, Kishore S, Bhatia ML, Trehan N. Trans-telephonic electrocardiographic monitoring-experience in India. *J Telemed Telecare* 1998;4(Suppl 1):8–11.
34. Berdusis K. The state of the art: Tele-echocardiography and telecardiology. *Telemed Today* **1999**;7:25–35.
35. Ong K, et al. A telemedicine system for high-quality transmission of paper electrocardiographic reports. *J Telemed Telecare* **1995**;1:27–33.
36. Tong DA, Gajjala V, Widman LE. A client/server system for remote diagnosis of cardiac arrhythmias; *Proceedings of the Annual Symposium on Computer Applications in Medical Care*. 1996, pp 601–605.
37. Ganguly P, Ray P. Telemedicine over enterprise-wide networks: A case study. *Proceedings of Globecom'98*; Sydney: 8–12 November, **1998**.

Address reprint requests to:

Pronab Ganguly
School of Information Systems, Technology
and Management
The University of New South Wales
Kensington NSW 2052, Australia

E-mail: p.ganguly@unsw.edu.au

This article has been cited by:

1. Probnab Ganguly , Pradeep Ray , N. Parameswaran . 2005. Semantic Interoperability in Telemedicine through Ontology-Driven Services. *Telemedicine and e-Health* **11**:3, 405-412. [[Citation](#)] [[PDF](#)] [[PDF Plus](#)]
2. Dr. Cheryl Reed , Ralph Burr , Ted Melcer . 2004. Navy Telemedicine: A Review of Current and Emerging Research Models. *Telemedicine Journal and e-Health* **10**:3, 343-356. [[Abstract](#)] [[PDF](#)] [[PDF Plus](#)]
3. USHA SINHA, ALEX BUI, RICKY TAIRA, JOHN DIONISIO, CRAIG MORIOKA, DAVID JOHNSON, HOOSHANG KANGARLOO. 2002. A Review of Medical Imaging Informatics. *Annals of the New York Academy of Sciences* **980**:1, 168-197. [[CrossRef](#)]