Methodology for Software Interoperability in Networked Services: A Telemedicine Perspective

Pronab Ganguly
School of Information Systems, Technology and Management,
The 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

Classification: Information Processing
Application Area: Telemedicine
Keywords: Methodology, CORBA, Software Agents, Software Interoperability, Telemedicine, Networked Services, Electrocardiography, Distributed System

1. Introduction

Various networked services are offered through distributed systems. In these distributed systems, the computing environment is heterogeneous. Hence it is very important that pieces of software in the distributed systems interoperate effectively. Successful software interoperation will lead to effective integration. Incorporating emerging techniques such as software agents and code mobility, this research study proposes a software-agent enhanced methodology for development of interoperable distributed system. This methodology is applied and evaluated in telemedicine which involves the integration of information, human-machine and healthcare technologies.

The next section illustrates the software interoperability challenges through an example from telemedicine. The subsequent section outlines the proposed methodology. Section 4 addresses the evaluation issues of the methodology. The concluding section mentions the scope of further research in this area.

2. Networked Services and Telemedicine

The networked services are organized through wide area networks linking various organizational networks. The effectiveness of networked services and associated computing systems depend on the successful integration of its subsystems. Configuration management of this widely distributed networks involves the management of the network's hardware and software configuration. Management of software configuration ensures that various pieces of software deployed over a large system are interoperable. Thus software interoperability may be defined as "“the capability with two or more programs can share and process information irrespective of their implementation language and platform”[HOW96]. As heterogeneity in computing environment is common in networked services, software interoperability presents interesting challenges. The problem of software interoperability, in various forms, exists in every distributed information system in all application areas. Because of its complex nature, software interoperability needs special attention in healthcare information systems. Telemedicine is an integral aspect of distributed healthcare information systems. Telemedicine effectively utilizes integration of information systems and telecommunication technology to exchange medical information for diagnosis, therapy and education.

It is important to note that in USA “*The $1 trillion health-care industry is under siege due to new competitors and pressures to contain costs. There is reportedly 10 to 20 percent waste in this industry (this amounts $100 and $200 billion). To respond to these pressures, health-care companies are embracing a range of technologies such as the Internet, data mining tools, data warehouses, object-orientation, and high-speed private networks. These technologies are being employed to restructure the businesses, enter new markets, improve customer relations, develop products and services, streamline distribution, and reduce bottom line. The main idea is to interconnect hospitals, doctors, drug makers, and drug distributors through Internet and then* *provide powerful object-based querying and transactional capabilities to minimize paperwork.*" [GAM96] Hence it has been rightly pointed out - " The inability to share information across systems and between care organizations is just one of the major impediments in health care business's progress toward efficiency and cost-effectiveness" [GRI00].

The next section describes the software interoperability challenges through an example from telemedicine.

2.2 The Software Interoperability Problem

2.2.1 Distributed ECG Analysis

Electrocardiography has been used for many years as a key, non-invasive method in diagnoses and early detection of heart disease. One of the aspects of telecardiology known as tele-electrocardiography deploys ECG machines to transmit ECGs over networks. Various research studies show that tele-electrocardiography diagnosis and ECG interpretation is simple, reliable and substantially cheaper in cost in comparison to conventional referral systems [GAN200]. 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. The cost and accessibility can be improved by separating the ECG instrumentation from its present tightly integrated knowledge-base and sharing the knowledge-base over an enterprise network.

In a distributed framework for a tele-electrocardiography system, the point-of-need user with the help of a front-end ECG machine will access and invoke the distributed knowledge repository over the network. The interpretation repositories will diagnose the ECGs on-line, and forward the ECG data along with multiple interpretations to the on-line supervisory cardiologist. For normal cases, results will be distributed over the network by the Supervisory Cardiology Station to the point-of-need and Computerized Patient Record Repository. For abnormal cases, a cardiac care service provider will additionally be alerted.

The available mechanisms for software interoperability are [HOW96]:

- 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 machine and frequently in other application programs. Standard such as Health Level Seven (HL7)-based [HEA97] interoperability falls into this category.

- Specification-level Interoperability

In this approach, the interoperability is addressed at the specification level and the programs communicate at higher level of abstraction. Thus the applications sharing the data need not know the finer details of the aggregate data type information. In our application, ECG data can be stored either in a single-dimension array or in multi-dimensional array but the applications sharing the data need not know the finer details and treat the entity as a whole. Object Management Group's (OMG) Common Object Request Broker Architecture (CORBA) and Microsoft's Distributed Component Object Model (DCOM) 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. 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 [ROSS99], [HEI96].

Again in the last few years, following significant trends are emerging [GUL00]:

- Structured programming movement divided monolithic programs into separate compilation unit and subroutines. The object-oriented approach encapsulated state information (i.e., data) with the code. The activation (When does a unit run?), however, still remained external. This has been resolved recently by the software agent paradigm which allocates autonomy besides encapsulated code and (state) data.
- As the components are more and more encapsulated, they need capabilities to implement the tasks allocated to them or in other words they should be able to offer autonomous solution. This autonomous solution may be light-duty or heavy duty.

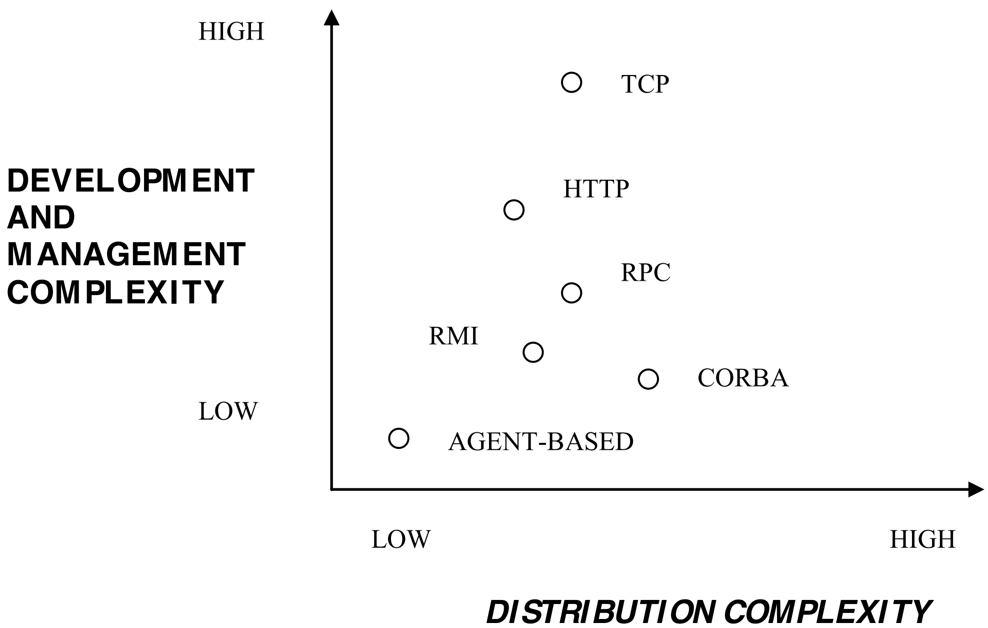

FIGURE 1: Empirical Evaluation of Some of the Distributed Systems Communication Protocols

TCP – Transport Control Protocol

HTTP – HyperText Transfer Protocol

RPC – Remote Procedure Call

RMI – Remote Method Invocation

CORBA – Common Request Broker Architecture

These developments have contributed to the growth of large scale distributed systems. But the methodology for small size distributed systems do not always scale up to address the issues of large scale systems [GUL00].

Traditional distributed systems may be perceived as a single virtual machine framework. This concept provides a virtual layer which conceals the details of network architecture from programming components. As the virtual layer provides the transparency of communication, the programming burden is reduced. But this may result in inefficient implementation as the programming modules cannot behave in a location dependent manner. Emerging trends adopt a different approach. Even though the concept of virtual machine framework is maintained, the software components become network aware i.e. the structure of the network is not hidden from the components. Thus similar events can be interpreted from a location dependent perspective.
Again an empirical evaluation , as shown in Fig 1, of some of the distributed systems communication protocols shows that the development and management complexity is lowest for software agent-based systems although it may not have high distribution complexity [MOU97]. 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 complexity. Therefore a combination of agent-based and CORBA (and similar)-based systems seem to be the best choice [GAN200]. Thus, a new methodology needs to be developed for software agent enhanced interoperable system. The next section describes an outline for the proposed methodology .

3. Methodology for Development of Software Agent Enhanced Interoperable Systems

The concept of a methodology has been debated in information systems literature for a long time. The definition [AVI97] provided by British Computer Society Information Systems Analysis and Design Working Group is probably most relevant here. The information system methodology is defined as “a recommended collection of philosophies, phases, procedures, rules, techniques, tools, documentation, management and training for developers of information systems”.
In information systems development, there are various types of methodologies [AVI97]. In the methodology, Structured Analysis and Design of Information Systems (STRADIS), the main techniques are the process-oriented ones of functional decomposition, data flow diagram, decision trees, decision tables and structured English. The basic approach in Information Engineering methodology is the data-oriented entity-relationship. Methodology, such as Process Innovation, focuses on business reengineering. The methodology, Object-oriented analysis, assumes that the essential building blocks of information systems are captured through objects and their attributes. Soft System Methodology is influenced by systems approach whereas Multiview is a hybrid methodology . Multiview combines aspects of many other methodologies. In this methodology, approach similar to Multiview will be adopted with specific emphasis on software interoperation. Process Innovation will be tied with Object Oriented analysis and subsequently integrated with emerging techniques such as software agent-based enhancement. An outline of the methodology is presented in figure 2. The initial approach in this methodology is similar to Co-operative Management of Enterprise Networks (CoMEN) [RAY00] though the emphasis is specifically on software interoperation rather than human co-operation. A brief description for each element is given below:

3.1 Requirement Gathering

This stage presents the overall picture of the system. The major elements are roles, activities and the environment.

3.2 Process Study

This stage involves a study of some representative scenarios in terms of activities and their interactions. The scenarios for study are chosen in consultation with the people actually working in the environment. This study identifies some of the fundamental sequences that strongly influences the overall efficiency of the process.

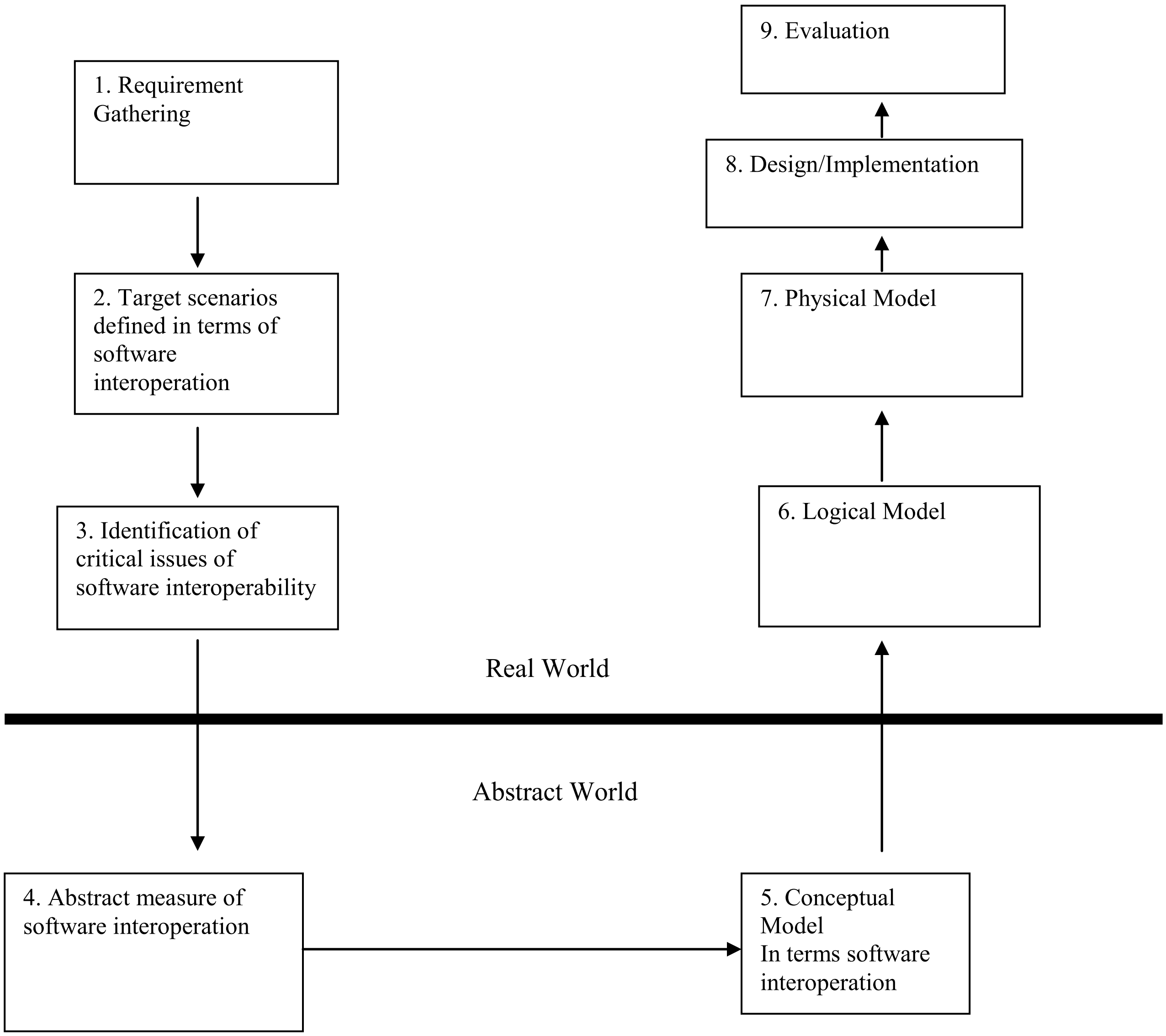

Figure 2 Methodology for Development of Interoperable System

3.3 Defining Software Interoperability Service Requirements
This stage identifies and examines the issues related to software interoperation. Every interactions and potential scope for improvement in terms of software interoperation in the scenario are identified .

3.4 Analysis
This stage (Stages 4 & 5 in Fig 2) carries out the analysis of the information collected during the previous steps. There is a need to define an abstract model for measuring software interoperation

for this purpose. An effective software interoperation should achieve seamless integration. The following dimensions are indicators of integration [DEG96]:

- Presentation

  Common appearance and behaviour i.e. ECG presentation.
- Data

  Perception of database as single entity i.e. Knowledge repository database and ECG database.
- Communication

  Standardization of exchanged messages i.e. standard ECG data format. ECG data format is under standardisation with SCP-ECG [RUB95]. But this format may not be universally accepted.
- Control

  Sequence of interworking i.e. sequence of operations for ECG analysis and diagnosis. In CORBAmed (Domain Task Force for healthcare informatics) [OMG97], relevant integration aspects 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.
- Semantic

Unified medical concept i.e. exchange of ECG data in its totality which includes desired granularity and presentation format.

These indirect measures of software interoperation determine the level or degree of software interoperation. Subsequently suitable mechanism of interoperation is adopted.

3.5 Logical Model

The stage provides a description of the system without any reference to the technology that may be used to implement it.

3.6 Physical Model

The stage provides a description of the system including the technology of the particular implementation.

3.7 Design & Implementation

This stage provides the technology specific detail design including programming and implementation of the system.

3.8 Evaluation

This stage involves the evaluation of the developed system from the perspective of software interoperation requirements.

Once a methodology is proposed, its effectiveness needs to be evaluated. The next section outlines a strategy to evaluate the effectiveness of the proposed methodology.

4. Evaluation of the Methodology

Though most of the methodologies in information system, such as STRADIS, claim to be general purpose [AVI97], Jackson System Development (JSD) has been most suitable for real-time processing application. Similarly, Soft Systems Methodology (SSM) has been developed to be applicable in human activity situations where very complex problem situations exist. The proposed methodology is specifically targeted towards Telemedicine though , with suitable modifications, it can be used in other areas as well..

The outline of the system design as per this methodology is shown in Fig 3.

- Step 1 involves business process reengineering through workflow in terms of emerging trends in technology.
- In step 2, feasibility for application of agent enhanced workflow may be studied to augment the identification of software interoperability issues.
- In step 3, conceptual design will be carried out through object analysis.

- In step 5, UML (Unified Modeling Language) may be used to design application object interface.
- In step 6, object realization is achieved through CORBA or DCOM IDL.

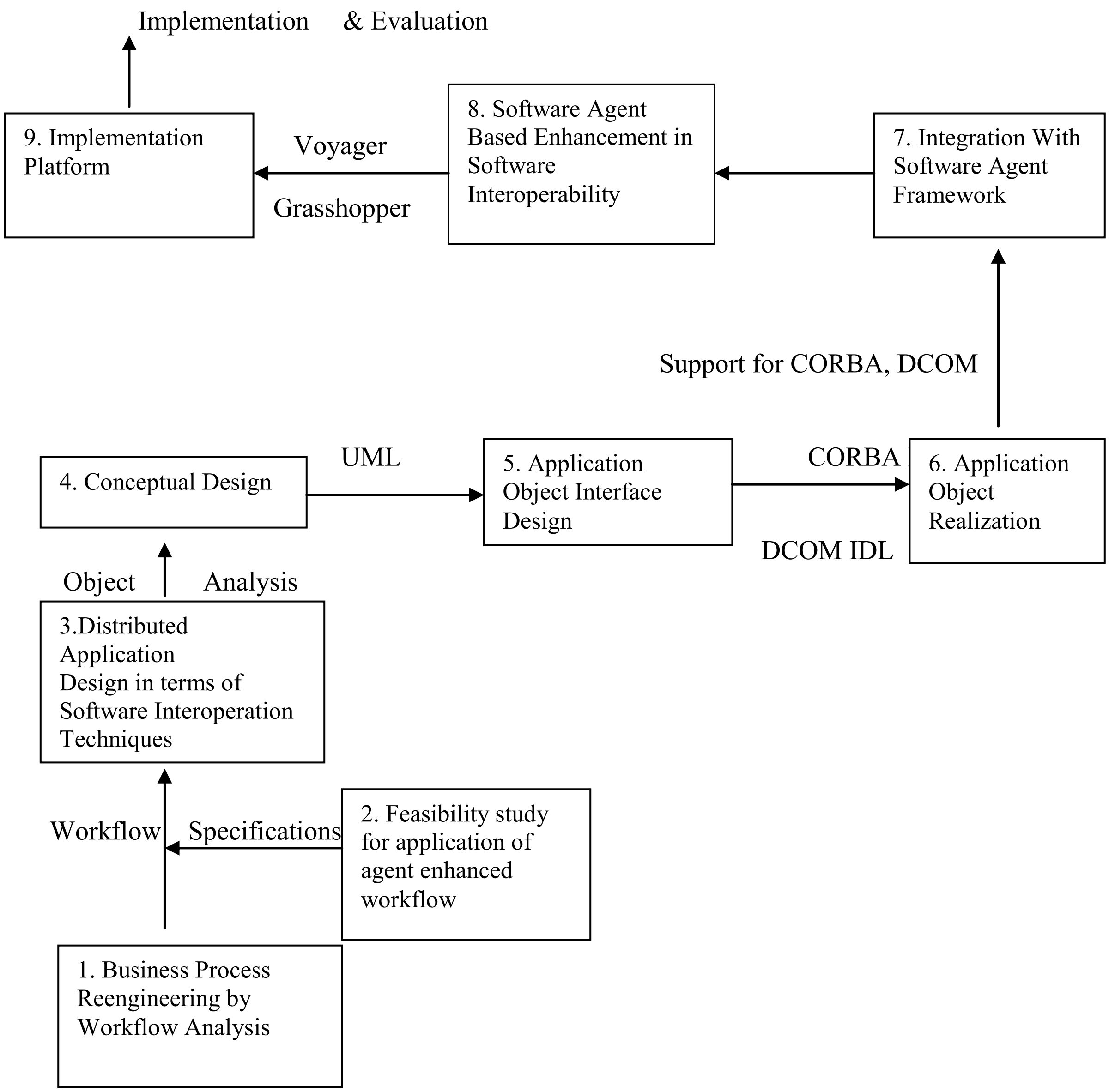

Fig 3 System Design Outline

- Step 7 is the integration with software agent framework.
- The outcome of the study will determine the implementation platform in step 9.
- This will be followed by programming, testing and evaluation.

The notations associated with tools and techniques at each step will be used for standardization. The next section briefly outlines proof-of-concept implementation of prototype system.

4.1 Proof-of-Concept Prototype System

For implementation of prototype, there is a need for a framework which provides basic infrastructure and communication protocol for implementation of inter-operable software agents. Due to space limitation, details of each stage of system design cannot be described here. 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)
- Better support for security with secure channels .
- 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.

In general, the agents and repositories are being developed with Voyager's special classes and interfaces. They include Agent, Messenger and Database [GAN200].

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 :

- Cooperation Agent (CA) coordinates high-level co-operation
- 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
- 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 .

As heuristic evaluation for telemedicine systems is found to be effective [LAT99], such an evaluation based on functional, technical, organizational, clinical and economic factors are under planning.

5. Conclusion

A software-agent enhanced methodology for interoperable systems in telemedicine is proposed. 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. Though our initial analysis looks quite 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 utilise this technology effectively more research is required into issues such as scalability, crash recovery, inconsistencies and security [GAN200]. Clinically effective telemedicine applications built on this technology will offer an efficient mechanism for distribution of medical expertise over the computer communication networks.

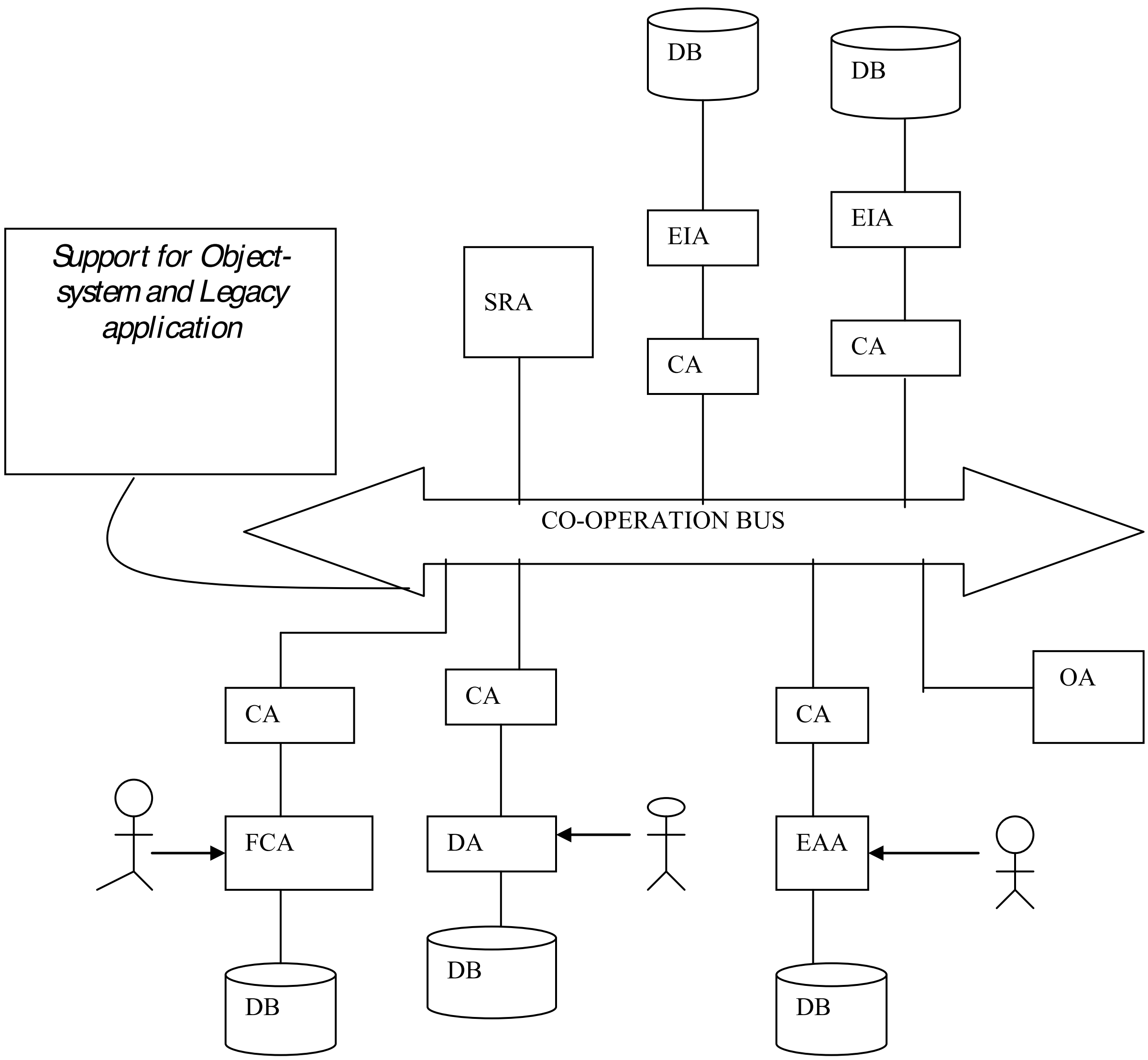

Fig 4: Architecture Schema for Distributed ECG Analysis Agent Framework

References

[AVI97] Avison D E & Fitzgerald G; Information Systems Development: Methodologies, Techniques and Tools; The McGraw-Hill Companies; 1997; page 280 – 294, 417 - 478

[DEG96] Degoulet P et al; Multimedia workstations: electronic assistants for the healthcare professional; Yearbook '96 of Medical Informatics; Jan H van Bemmel, Alexa T. McCray (eds); Stuttgart: Schattauer, 1996, page 65-75

[GAM 96] Gambon J; Healthcare Gets Technology Transfusion; Information Week; July 1, 1996; page 14 – 19

[GAN200] Ganguly P and Ray P; A Methodology for the Development of Software Agent Based Interoperable Telemedicine Systems: A Tele-Electrocardiography Perspective; Telemedicine Journal; Volume 6 Number 2 (15th July, 2000);
[GUL00] Gulhas Laszio, Kovacs Laszlo, Micsik Andras, Pataky Balazs & Zsamboky Istvan; Knowledge Shared XPS-based Research Network Using Multi-Agent Systems; EU-Inco No: 977113; 2000 (Unpublished Study)
[GRI00] Grimson Jane, Grimson William & Hasselbring Wilhelm; The SI Challenge in Health Care"; Communications of the ACM; June 2000 – Volume 43, Number 6; page 49 – 55
[HEA97] Health Level Seven; Standard Version 2.3; Standards Australia; May 13; 1997
[HEI96] Heiler S, Miller R J, 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 29/12/1999
[HOW96] Howie T C, Kunz J C, Law K H; Software Interoperability; Center of Integrated Facility engineering; Stanford University; 12/25/1996.
http://www.dacs.dtic.mil/techs/interop/title.shtml
[HUS00] Huston L T & Huston J L; Is Telemedicine a Practical Reality ? ; Communications of the ACM; June 2000 – Volume 43, Number 6; page 91 – 94
[LAT99] Lathan CE, Sebrechts MM, Newman DJ and Doran CR; Heuristic Evaluation of a Web-based Interface for Internet Telemedicine; Telemedicine Journal; Volume 5; Number 2; 1999
[MOU97] Mountzia M A, Benech D; Communication Requirements and Technologies for Multi-Agent Management Systems; Proceedings of DSOM 97, Sydney, 1997
[OMG97] OMG's CORBAmed Task Force; http:// www.omg.org/corbamed.htm; Accessed on 12.10.1997
[RAY00] Ray P; Cooperative Management of Enterprise Networks; Kluwer Academic/Plenum Publishers, New York; 2000; Page 47 – 63; 101-104; 125 - 127
[ROS99] Rosenthal A, Sciore E; Description, Conversion, and Planning for Semantic Interoperability;
http://www.mitre.org/pubs/data.mgt/papers/sem.pdf
Accessed on 29/12/1999
[RUB95] Rubel P, Fayn J, Zywietz C, Willems J L, Assanelli D, Malossi C, Florin C; Proceedings of Health Telematics '95; 1995; ISBN: 88-7958-003-5; page 381-386
[RUS98] Rusovick M Rik, Warner J David; The Globalization of Interventional Informatics Through Internet Mediated Distributed Medical Intelligence; New Medicine; Spring 1998; http://www.pulsur.org/febweb/papers/globalmed.html accessed on 5/12/99