Defence Science Journal, Vol. 55, No. 4, October 2005, pp. 477-485

SHORT COMMUNICATION

Distributed Computing and its Scope in Defence Applications

B.V. George, R.V. Raghavan, S. Pooja, and N.K. Agarwal

Defence Laboratory, Jodhpur-342 011

ABSTRACT

Distributed computing is one of the paradigms in the world of information technology. Middleware is the essential tool for implementing distributed computing for overtaking the heterogeneity of platform and language. DRDO's intranet, DRONA, has the potential of hosting distributed applications across the network. This paper deals with the essentials of distributed computing, architecture of DRONA network, and the scope of distributed computing in Defence applications. It also suggests a few possible applications of distributed computing.

Keywords: Component, distributed computing, middleware, CORBA, DRONA, distributed geo-processing, open-geographical information system, common object request broker architechture

1. INTRODUCTION

Internet access over server and client architecture have crossed the boundary of information exchange long back and stepped into the territory of communication between remote executable codes across the web for all practical applications. This major advancement in the practical use of the web is known as distributed computing. The purpose of distributed computing is to provide a global environment for all business, organisational, and research-related activities, cutting across the barriers of location, network, platform, and operating systems. One of the successful implementations of distributed computing is the Human Genome Project, wherein 16 major research organisations and innumerable systems across the world were utilised to map the human genome[1]. The project was initiated in 1990 and the draft report submitted by Human Genome Consortium and Celera Genomics in June 2000. It is the most computation-intensive project ever executed in the history of computational biology. The data and their analyses were voluminous and it might have been difficult to complete the project within the stipulated time frame without using distributed computing.

Implementation of distributed computing requires an essential tool called middleware, which overcomes the complexities and heterogeneity of platforms, operating systems, programming languages, and networks. This paper deals with the introductory aspects of middleware, an intranet for defence applications (DRONA), and utility of distributed computing in defence.

2. MIDDLEWARE

The objects designed for a single system with rigid features will not be useful. The need of the hour is to exploit the full potential of the worldwide network and to have intelligent and more powerful objects to cater to the heterogeneity of computer

Revised 18 February 2005

system, the operating system, and the network. Objects of this kind are called components. While objects are bound to a single application and platform, components are implemented over networks of multiple platforms. Components are developed in dynamic languages like JAVA.

In distributed computing, objects implemented in a system need to communicate with those implemented in other systems across several networks. The platforms, the operating systems, and the networks involved may differ from one another. The situation is similar to that of two persons in conversation without a common language of expression. They require an interpreter to converse with each other. The objects across networks also require one such interpreter to realise the communication among them. But, in this case, the interpreter is not a simple entity, but an integrated architecture and serves a wide variety of tasks in addition to the basic communication. The distributed object architecture, which makes distributed computing possible, is called middleware. The middleware provides an object model to design reusable components while hiding communication details behind the interfaces.

3. COMMON OBJECT REQUEST BROKER ARCHITECTURE

Some of the prevailing distributed systems are remote procedure call, distributed common object model, remote method invocation, and common object request broker architecture (CORBA). Each of these adopts unique methodology and architecture to tackle the heterogeneity in the distributed computing environment[2]. But, one chooses a CORBA relevant distributed system, because of the features that CORBA provides. CORBA combines the advantages of both platform and language independence. Platform independence means that CORBA objects can be used on any platform for which there is a CORBA object reuest broker (ORB) implementation. Language independence means that CORBA objects and client can be implemented in just any programming language. Moreover, CORBA objects need not know which language was used to implement other CORBA objects that these talk to[3].

The distributed systems rely on the definition of interfaces between components and on the existence of various services available to an application. CORBA provides a standard mechanism for defining the interfaces between components as well as some tools to facilitate the implementation of those interfaces using the developer's choice of languages. In addition, a wealth of standard services, such as directory and naming services, persistent object services, and transaction services are also specified. Each of these services is well defined in a CORBA-compliant manner, so these are available to all CORBA applications. Finally, CORBA provides all the plumbing that allows various components of an application, or of many applications, to communicate with each other. While the concept of CORBA had been in existence since 1990, it was standardised by the Object Management Group (OMG), a consortium of about 500 industries[4]. CORBA provides distributed infrastructure and object services for distributed computing.

3.1 Metholody of CORBA

.CORBA facilitates distributed computing by creating interface specification in interface definition language (IDL). The specification includes various services for creating and deleting objects, accessing these by name, storing these, etc. These are achieved in totality by making the environment completely IDL-specified. This process involves two steps. In the first step, component provider specifies the interface and structure of objects in IDL. The definition of all these interfaces is included in the CORBA interface repository. In the second step, the service provider supplies the distributed services, which determine the objects in a network, methods related to these, and interface adapter supported by these. This makes the location of the objects transparent to the clients, and the components discover each other dynamically at run time'. A server object can be implemented in any language, viz., ADA, C++, or JAVA and its specifications need to be defined in IDL. These specifications serve as the binding contract between the clients and the servers.

3.2 Architecture of CORBA

As per the Object Management Architecture Guide (1992-95), CORBA architecture consists of four main elements, viz., object request broker (ORB), CORBA services, CORBA facilities, and application objects. Figure 1 illustrates the model of the OMG reference architecture[6].

3.2.1 *Object Request Broker*

The object request broker consists of a platform and a language-specific library. It is provided by various vendors and it defines the object bus. ORB receives requests and responses from local and remote objects. This makes the location of objects transparent and helps to cross the barrier

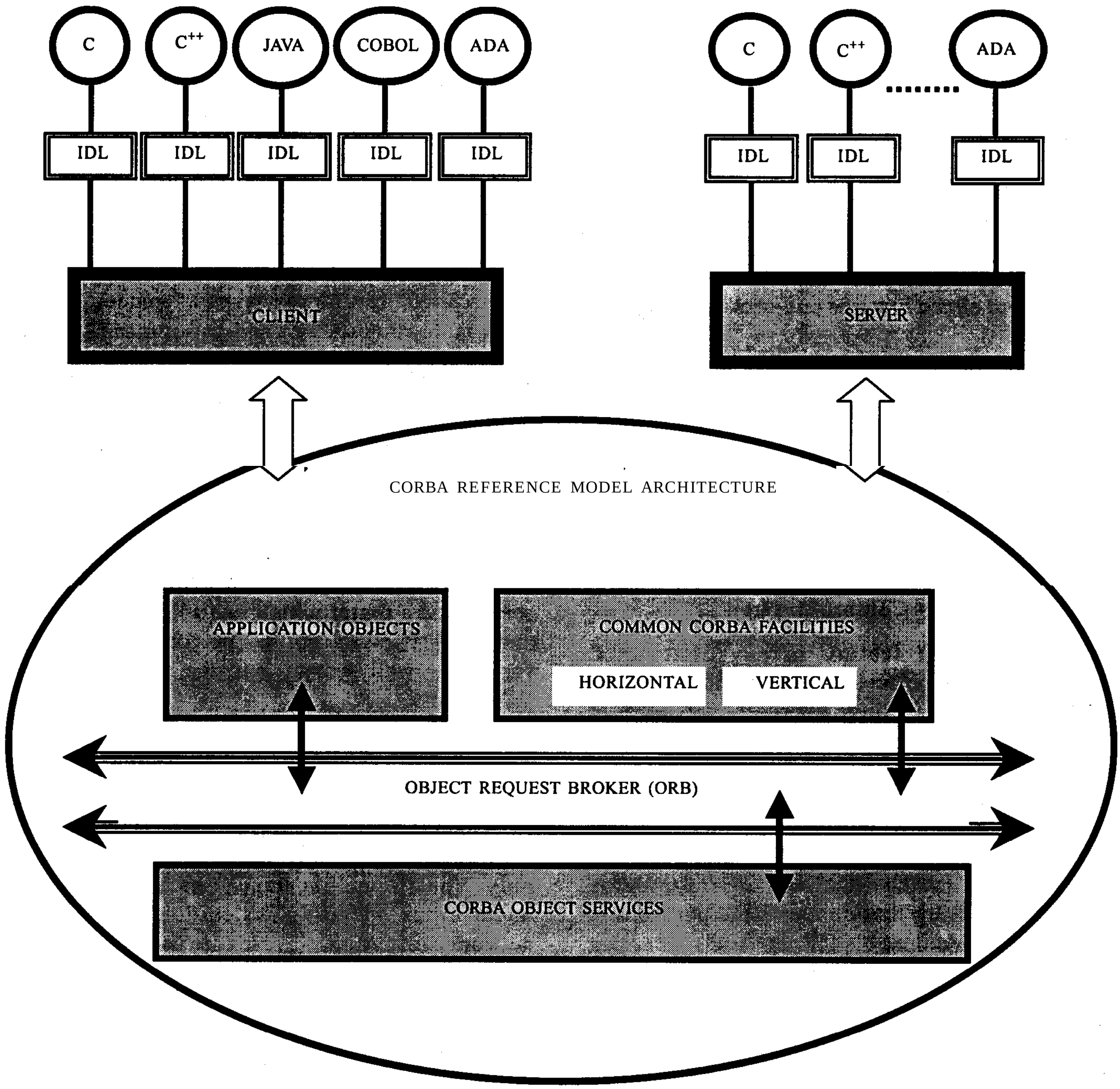

Figure 1. OMG's CORBA reference model architecture and IDL bindings

of heterogeneity of the platform, language, and the network.

3.2.2 CORBA Services

CORBA services are domain-independent interfaces that are used by distributed object programs. These define the object framework at the system-level that extends the ORB bus. Naming service, for example, allows clients to find object based on names. Lifecycle management, security, transaction, and event notification are some of the essential features of *CORBA* services.

3.2.3 CORBA *Facilities*

Common CORBA facility defines the horizontal and vertical application frameworks used directly by business objects. Like CORBA object service interfaces, these interfaces are also horizontally oriented, but aimed towards end-user applications.

3.2.4 Application Objects

Application objects are business objects, the ultimate consumers of CORBA. The interface definition of these objects can be made specific to either domain or application. Domain interfaces are similar to object services and common facilities, but are oriented towards specific application domains[7, 8].

3.3 Client-server Communication

An important feature in any distributed environment is the object location for communication among the objects. Since all the server objects are registered in the naming service, the client can easily locate the server object method (SOM) and can get an interoperable object reference (IOR) to the SOM[9]. With this IOR, the client executes the required SOM. The parameters to be passed on to the SOM undergo a process called marshalling that converts the parameters into a network-compatible format. At the receiving end, these parameters are converted into the relevant machine format. This process is called demarshalling. The result of the SOM execution is passed on to the client object through the same route using marshalling and demarshalling.

4. DRONA-THE INFORMATION HIGHWAY OF DRDO

DRONA (DRDO's rapid online network access) has been setup to provide a secured network to connect all DRDO laboratories and the DRDO Hqrs[5]. It is based on integrated service digital network (ISDN) services provided by the Department of Telecommunication (DoT). Various DRDO laboratories are connected to this network, and locations where ISDN facilities do not exist will be brought on to the network through very small aperture terminal (VSAT) links or other alternatives. Initially, connectivity will be provided using ISDN's basic rate interface (BRI) facility, which gives a data rate of 128 kbps.

4.1 Organisation of DRONA

The DRONA network is organised in a two-tier architecture with two types of routers, referred to as main router (MR) and sub-router (SR). The MR will have more ports for connectivity and supports SR in the vicinity. The sub-router (SR) caters to the requirement of a single laboratory. There are six MRs situated at different locations called main router centers (MRCs). Rest of the laboratories have SRs and these are called sub-router centers (SRCs). Figure 2 illustrates the DRONAs network layout. The MRCs have a firewall, a main server, a SUN server which is configured as DNS/web server, and a router with 8 to 10 ISDN WAN ports and 8 PSTN WAN ports. The SRCs have a mail server, firewall system, and the router with two WAN ports.

4.2 Security Mechanisms & Management

Since DRONA is based on public services such as ISDN, necessary security mechanism is adopted to restrict the access to authorised users only. The firewall system is an Intel-based PC under Linux operating system loaded with three network cards. Multiple cards are used to physically separate LAN from the DRONA. Firewall system filters the network packets based on the protocols, services, and IP address. Apart from this, more security is ensured in the DRONA network by implementing features like challenge handshake

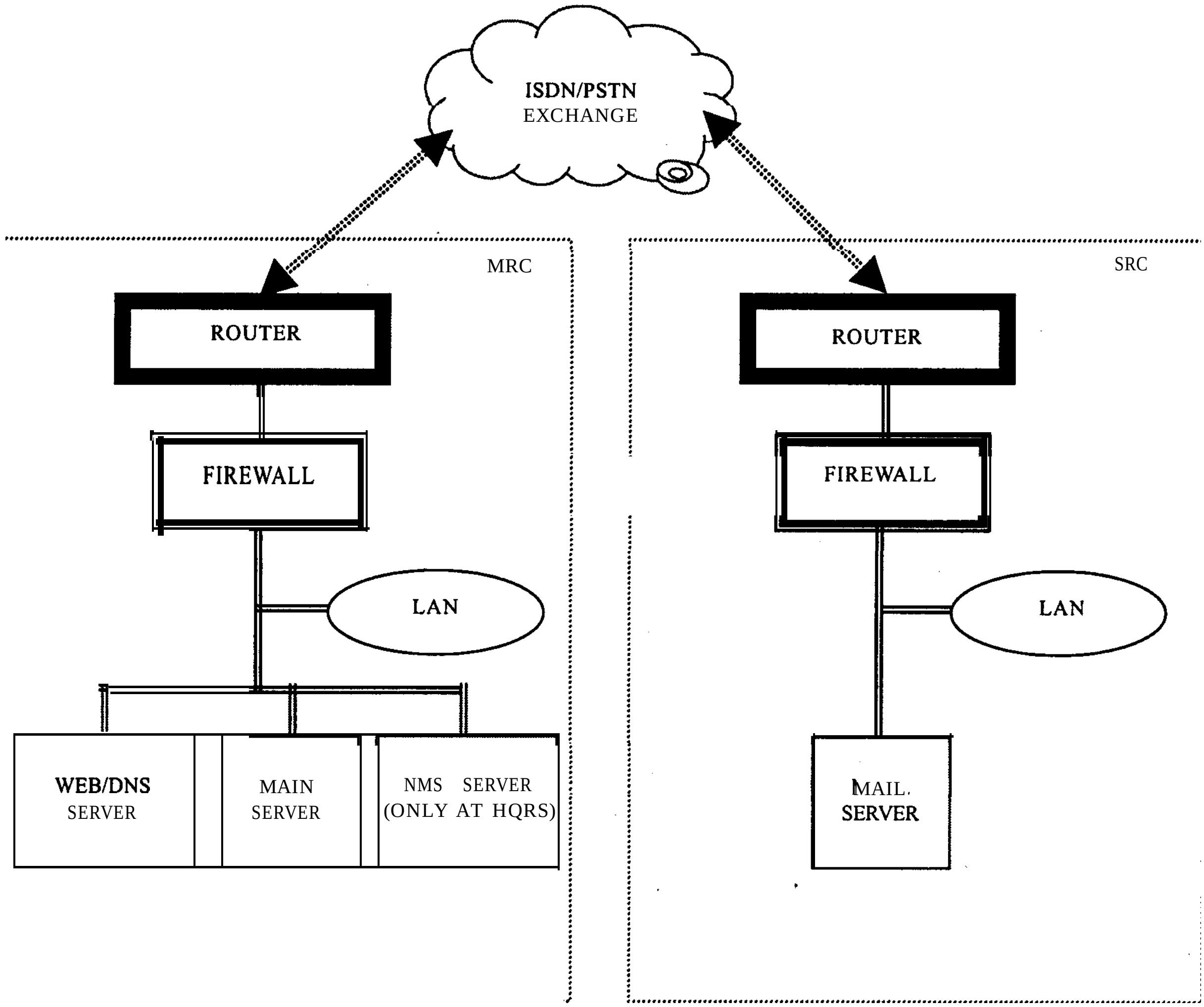

Figure 2. DRONA network layout

authentication protocol (CHAP) added to PPP protocol, ISDN caller identification, drop call and call back, etc.

For the DRDO HQrs MRCs, an additional server is used to manage the entire DRONA network and is called network management system (NMS) by which a single point of control to the network is achieved. The NMS allows monitoring and control of the traffic on the whole DRONA network.

4.3 Services on DRONA

The DRONA network provides services that are commonly available on the internet and other intranets. E-Mail, file transfer protocol (FTP), bulletin boards, home page and voice mail, video conferencing, etc are some of the services. For enhanced security, e-mail and FTP services can be encrypted. The FTP service is meant to transfer large files from one system to the other system. Information of general interest to all the users can be placed in the bulletin board. Each laboratory can establish and maintain its home page on the web server attached to its MRC. A new web mail service has also been introduced by which mail can be accessed from any DRONA terminal across DRDO. Voice mail can be used for interactive communication between the two laboratories. Online meeting among various laboratories will be possible through video conferencing. Services like remote procedure call, remote execution, Telnet, etc will also be provided on the DRONA network.

5. SCOPE OF DISTRIBUTED COMPUTING OVER DRONA NETWORK

Consider an application, which requires database and skill base from different laboratories of DRDO and needs to be implemented over the intranet. The application can be split into a number of modules and development of modules can be assigned to each of the concerned laboratories. It can so happen that the laboratories have programmers specialised in a particular language and operating system. They tend to develop the module assigned to them in that operating system and language. There may be restrictions imposed by format and type of data used also. To integrate all these modules and the data, and to run the application over the intranet, there is a need for a middleware, which can understand and tackle the heterogeneous modules. As evident from the earlier sections, a middleware relevant to CORBA architecture can serve this purpose and facilitate distributed computing over DRONA.

Distributed computing has far reaching applications in the defence sector of our country. Vast expanse of territory poses the problem of protecting the complete border with defence installations, and support facilities. At the time of any emergency or war-like scenario, it should be possible to mobilise and deploy forces within a short notice. This requires ready access to information on terrain and related factors. Further, information on the availability of civilian infrastructure helps to utilise the military resources economically and efficiently. A subset of distributed computing, which deals with the geo-information processing, is called distributed geo-processing. As a peace-time exercise, it will be useful to completely organise and update information on the resources in various parts of the country in a distributed geo-processing environment. Such information about the adversaries will help in the tactical and strategic operations during wartime.

The Open GIS Consortium (OGC) of industries and academic institutes founded in 1994, brought out a software specification called Open GIS Specification (OGS), a comprehensive specification of a software framework for distributed access to geodata and geo-processing resources[10]. The specification includes the implementation details of distributed geo-processing on various distributed computing platforms (DCPs), viz., CORBA, DCOM, JAVA/RMI, and others. Figure 3 shows a possible architecture of the distributed geo-processing application relevant to CORBA. The services and ORB will have the same feature as required by any other distributed application. Facilities are application-specific and are of two types, viz., horizontal and vertical. While horizontal facility provides services of general nature, such 'as graphic-user interface or task management, vertical facility is meant for a particular industry or domain. In the case of distributed geo-processing, it will have GIS features specified by OGC.

Interoperability is the most essential requirement of distributed geo-processing. Programmers developing applications achieve interoperability by application programming interfaces (API), which conform to a common, agreed upon set of criteria. Popular GIS packages from Bentley Systems, ESRI, Intergraph, MapInfo, etc, have already released APIs for distributed access to various GIS features and processing. These APIs provide primitive-level facilities, establishment of linear and angular units, spheroids, datum, and map projections. At intermediate level, these enable the construction and manipulation of geometric elements such as points, lines, curves, strings, rings, polygons, and surfaces. At the GIS feature-level, the APIs provide for the creation and management of feature collections, and the ability to access a feature from such collections using geometric, topological, or attributional modifiers. Further, APIs for spatial reference systems, grid coverage access, catalog service, and querry service, are also being made available. One of these existing packages can be used to develop, maintain, and access geographic information distributed over a network. Else, a package can be developed to cater to the requirements of distributed geo-processing.

The distributed geo-processing environment relevant to defence requirements may contain three major components, viz., terrain analysis, infrastructure, and environmental characterstics".

Terrain analysis database can have layers related to hydrology, geology, and vegetation. These data can be collected, updated, stored, and maintained

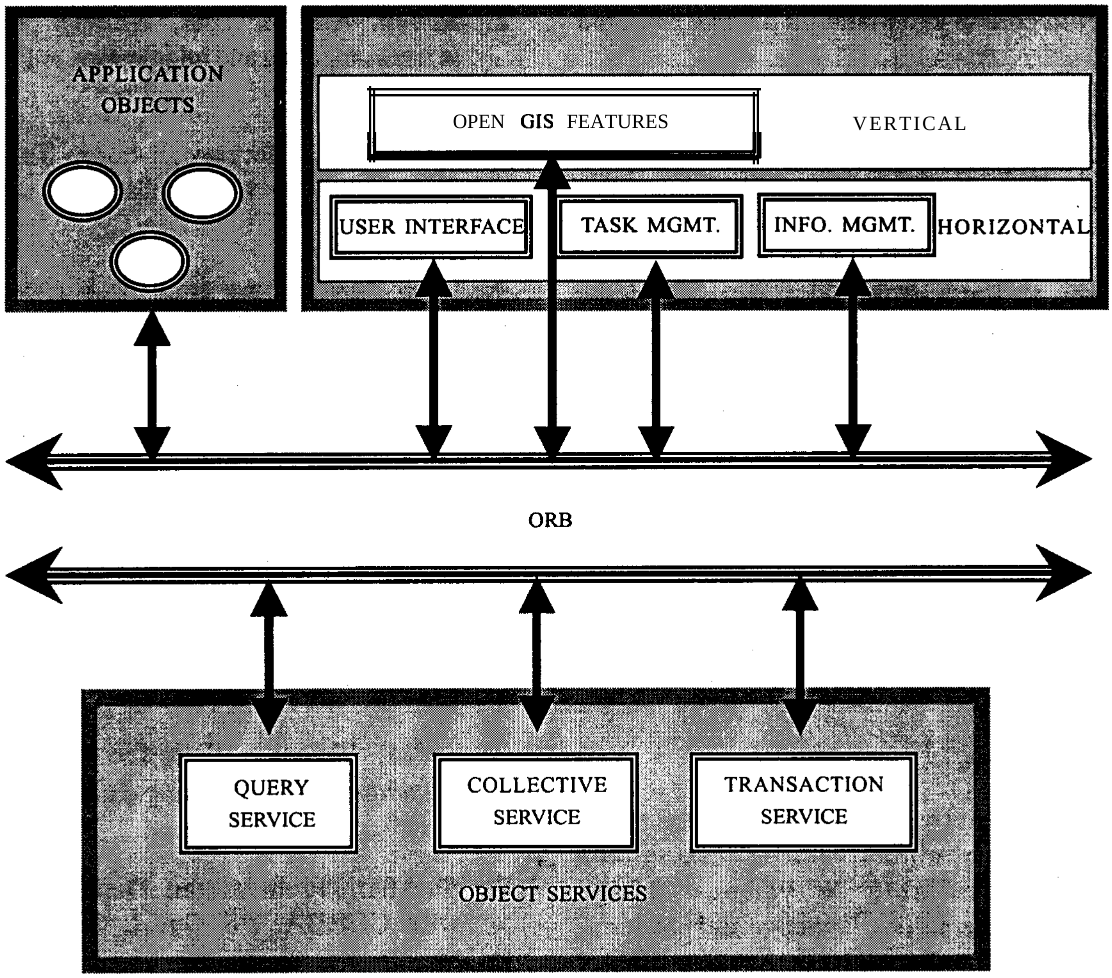

Figure 3. Architecture for a conceptual Open GIS Consortium over DRONA

by the concerned laboratories. The data collection can make use of traditional methods or by interpretation of remotely sensed data. The databases can be remotely accessed, at the time of requirement, to derive trafficability map, cover and concealment map, etc.

The infrastructure database can have layers on transportation network, airfields, communications, ports, harbour, etc. Information derived from these databases will be useful in effective operational planning. The database on environment can have layers on temperature and rainfall details, which affect any military operation.

The availability of data, which can be part of these databases, has increased manifolds over the past three decades. This is due to the developments in the field of sensors, communication and platform, and computation techniques. It is extremely difficult to have these on a single system and provide a monolithic solution to all defence-related needs. The advent of distributed geo-processing makes it possible for parts of a database to be stored and maintained at different locations. It is economical in comparison to the monolithic system. Each of the laboratories need not store and maintain a complete spectrum of databases. They can access the remote database and make use of the specialised processing at remote sites as and when required. For decision-makers, large archives of data across the network help to make on-the-spot decisions and offer these to anyone connected to the network. It enables distributed custodianship to various laboratories to collect and maintain database related to them.

Interoperability provided by distributed computing platform is useful in distributed digital library (DDL) also. In the DRDO setup with a number of laboratories working in closely associated areas of technology, the distributed digital library is highly relevant. Each of the laboratory can have its own digital library service oriented towards its research area. Any laboratory, which needs to have knowledge on a particular topic, can remotely access distributed digital library service implemented by other laboratory over the network. In addition to the service of information repository, the distributed digital library can have automated research summarisation and trend analysis also[12]. The distributed collection of such services will be helpful in performing information-extensive tasks.

6. CONCLUSION

The DRONA network can be utilised much beyond its proposed objective of accessing data and resources available in various DRDO laboratories. Interoperability, language and platform-independence, will enhance its utility to a greater extent. A middleware relevant to CORBA architecture can be employed to achieve interoperability and facilitate distributed computing. Effective implementation of a distributed application like distributed geo-processing over DRONA can be very useful to national defence.

ACKNOWLEDGEMENT

The authors are grateful to Shri R.K. Syal, Director, Defence Laboratory, Jodhpur, for granting permission to publish this paper.

REFERENCES

1. Singh, S. Kumar. The human genome : A first look at the blueprint of human inheritance. *Current Science,* July 2000, **79(2),** 25.
2. Simon, A.R. & Tom, Wheeler. Open client/server computing and middleware. A.P. Professional, New York, 1995.
3. Rosenberger, Jeremy. E-book: Teach yourself CORBA in 14 days. Macmillan Computer Publishing, USA, 2004.
4. Object Management group. weblink www.omg.org
5. Steve, Vinoski. Distributed object computing with CORBA. *C++ Report Magazine,* July/August 1993.
6. Randy, Otte *et.al.* Understanding CORBA. Prentice Hall, New Jersey, 1995.
7. Steve, Vinoski. CORBA: Integrative drivers applications within distributed heterogeneous environment. *IEEE Comm. Mag.,* February 1997, **32(2).**
8. Steve, Vinoski. New features of CORBA. *Communications of the ACM,* October 1998, **41(10).**
9. Shekhar, Govindarajan. Distributed computing with CORBA. *PC Quest,* Oct 2000, 158
10. Open GIS Consortium. web link – www.opengis.org
11. Desmond, Ball & Ross, Babbage. Geographic information systems: Defence applications. Pergamon Press, Australia, 1989.
12. Using distributed objects for digital library interoperability. Web link – www.computer.org/computer/dli/r50061.html

Contributors

Mr B.V. George obtained his MSc (Information Technology) from the AAIDU, Allahabad and DAC from C-DAC, Pune University. He is a certified software quality professional from STQC, New Delhi. He has been working at the Defence Laboratory (DL), Jodhpur, since 1999. His areas of interest are: Network administration and distributed computing.

Ms S. Pooja obtained her MCA from the JNVU, Jodhpur. She has been working at the DL, Jodhpur, since 2003. Her areas of interest are: Database management and internet technologies.