AR-009-460

DSTO-CR-0008

Development of a Defence Distributed Computing Environment (DCE) Database Demonstrator

B. McClure and J. Mansfield

APPROVED FOR PUBLIC RELEASE

DEPARTMENT OF DEFENCE
DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION

Development of a Defence Distributed Computing Environment (DCE) Database Demonstrator

B. McClure and J.Mansfield

Information Technology Division
Electronics and Surveillance Research Laboratory

DSTO-CR-0008

ABSTRACT

This report discusses the development of a Defence Distributed Computing Environment (DCE) database demonstrator program. The Demonstrator program showcases the interoperability, portability, survivability and security features of Open Software Foundation's Distributed Computing Environment.

APPROVED FOR PUBLIC RELEASE

DEPARTMENT OF DEFENCE

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION

Published by

DSTO Electronics and Surveillance Research Laboratory
PO Box 1500
Salisbury. South Australia, Australia 5108

Telephone: (08) 259 7053
Fax: (08) 259 5561

AR-009-460
November 1995

APPROVED FOR PUBLIC RELEASE

Development of a Defence Distributed Computing Environment (DCE) Database Demonstrator

EXECUTIVE SUMMARY

DSTO Task ADF 94/151 (Distributed Systems Technology) includes the development of a prototype that demonstrates the potential of the Open Software Foundation's (OSF) Distributed Computing Environment (DCE) technology for addressing the interoperability, security, survivability, portability and reliability requirements of modern C^3I systems (working definitions for these terms are given in the body of the report). One of two such demonstrators is a database retrieval system based upon notional enemy ORBAT data. This report describes the results of developing and testing this demonstrator and the relevance for deployment of DCE by the ADF.

From the user's perspective the Demonstrator is a simple data retrieval system based upon notional enemy ORBAT data. There are several searches that can be performed based on selections made by the user. DCE security is implemented through three main features: Mutual authentication of clients and servers; Authorisation of clients by servers, based on security group membership; Integrity and sequence number of packets guaranteed.

Interoperability is demonstrated by allowing the user to run the client on HP, DEC Alpha, Windows 3.1, or OS/2 machines, and obtain services from a server running on HP, DEC, SUN, or OS/2 (the server is currently being ported to an IBM MVS mainframe). Just as important, the client and server programs are unaware of which type of computer they are interacting with or where that computer resides. Furthermore, the Demonstrator communications are built around a DCE interface specification, which enables transparent integration of new applications.

Survivability is demonstrated by enabling the client program to find an alternative server and continue processing in the event of server or communication failure. The client changes to a new server (if available) without operator intervention.

From an architectural perspective the Demonstrator uses a three tier architecture: Graphical User Interface (GUI) based DCE clients, application servers and Database Management Systems (DBMS). Of interest is the fact that the Database Management System (DBMS) is seen as a legacy system to which the application servers must interface. All of the features of this system could still be obtained if the data resided on a legacy mainframe computer.

A comparison of the development process for three different communications methods is made, based upon the authors' experience: OSF DCE, database client/server development tools, and TCP/IP sockets programming. Summarising this comparison, it is seen that the database client/server development tools would have produced a working system in less time and with less training or expertise but would not meet all

requirements. The sockets mechanism is flexible and could have met all requirements, but some features would have to be crafted from scratch or third party systems integrated. This would have taken longer and may be difficult to scale up at a later time. Thus DCE was the only mechanism that could satisfy all the requirements.

The features of survivability, reliability, security, portability and interoperability have been demonstrated by the Defence DCE demonstrator. The Demonstrator verifies many of the claims made about DCE. Although the development process for DCE applications is more complex for a simple system such as the Demonstrator compared with a traditional client/server development, the result is a more flexible and open system. The experience gained in developing the Defence DCE database demonstrator will enhance further research, reports and ad-hoc advice provided to the ADF under Task ADF 94/151.

CONTENTS

FIGURES

GLOSSARY

ADF	Australian Defence Force
C^3I	Command, Control, Communications and Intelligence
CDS	Cell Directory Service
CITI	Center for Information Technology Integration
DBMS	Database Management System
DCE	Distributed Computing Environment
DGFD	Director General Force Development
DSTC	Distributed Systems Technology Centre
DSTO	Defence Science and Technology
GUI	Graphical User Interface
INFOSEC	Information Security
IP	Internet Protocol
IT	Information Technology
K95	Exercise Kangaroo 1995
LAN	Local Area Network
ODBC	Open Database Connect
ORBAT	Order of Battle
OSF	Open Software Foundation
TCP	Transport Control Protocol
WAN	Wide Area Network

1. INTRODUCTION

Military users of today's Information Technology (IT) are calling for secure, survivable and reliable systems that communicate with each other, at a lower cost and with shorter lead times [1,2]. Meanwhile, IT project managers and system managers require open and scalable systems that are able to be centrally managed by a small team of administrators.

DSTO Task ADF 94/151 (Distributed Systems Technology) includes the development of a prototype that demonstrates the potential of the Open Software Foundation's (OSF) Distributed Computing Environment (DCE) technology for addressing the above requirements. One of two such demonstrators is a database retrieval system based upon notional enemy ORBAT data. This report describes the results of developing and testing this demonstrator and the relevance for deployment of DCE by the ADF.

A second DCE demonstrator is under development at the time of writing. It is based upon an existing system called the Geographic Hypermedia Information System (GHIS) and is being re-engineered into a client/server architecture and ported to DCE. A separate report is expected to be published at the conclusion of this activity.

The Defence DCE database demonstrator is referred to in this report as 'the Demonstrator'. Production of the Demonstrator has allowed DSTO staff to gain a valuable insight into the issues associated with DCE application development, verify some of the features offered by the technology, as well as produce a working prototype which enables members of the ADF to see the technology in a Defence environment.

Although this task is sponsored by DGFD(Joint) the results are of wider interest as many Defence systems are becoming distributed. Further, if interoperability is to be achieved across systems then a standardised approach to inter-system communication needs to be taken.

2. OVERVIEW OF THE DEMONSTRATOR

This section gives an overview of the Defence DCE database demonstrator. The reasons for its development, the design goals for the Demonstrator, the system as the user sees it and the chosen software architecture are discussed.

2.1 Purpose of the Demonstrator

Why have a Defence DCE demonstrator? Many non-Defence related organisations (eg: Center for Information Technology Integration, Michigan and the Cooperative Research Centre for Distributed Systems Technology, Brisbane) and some Defence related organisations (eg: MITRE corp) have already constructed DCE prototypes. However, although DSTO is in possession of some reports related to these activities, this does not transfer experience to DSTO nor does it provide the ADF with a working prototype of the DCE technology. The purpose of this demonstrator then, is threefold.

- To demonstrate and verify some of the features of DCE.
- To provide the ADF with a better understanding of how DCE could be used in the Defence environment.
- To gain experience with the DCE technology.

2.2 From the User's Perspective

From the user's perspective the Demonstrator is a simple data retrieval system based upon notional enemy ORBAT data. There are several searches that can be performed based on selections made by the user. For example, assume that during K95, a detachment of soldiers is fired upon by the enemy forces. A search can be conducted for units equipped with guns (the type of ammunition can be specified if known). A list of units along with summary data is then displayed. The user can then select one of these units to display data associated with that unit. Using this process, some possible scenarios could be constructed. Figure 1 depicts the main screen of the Demonstrator client program.

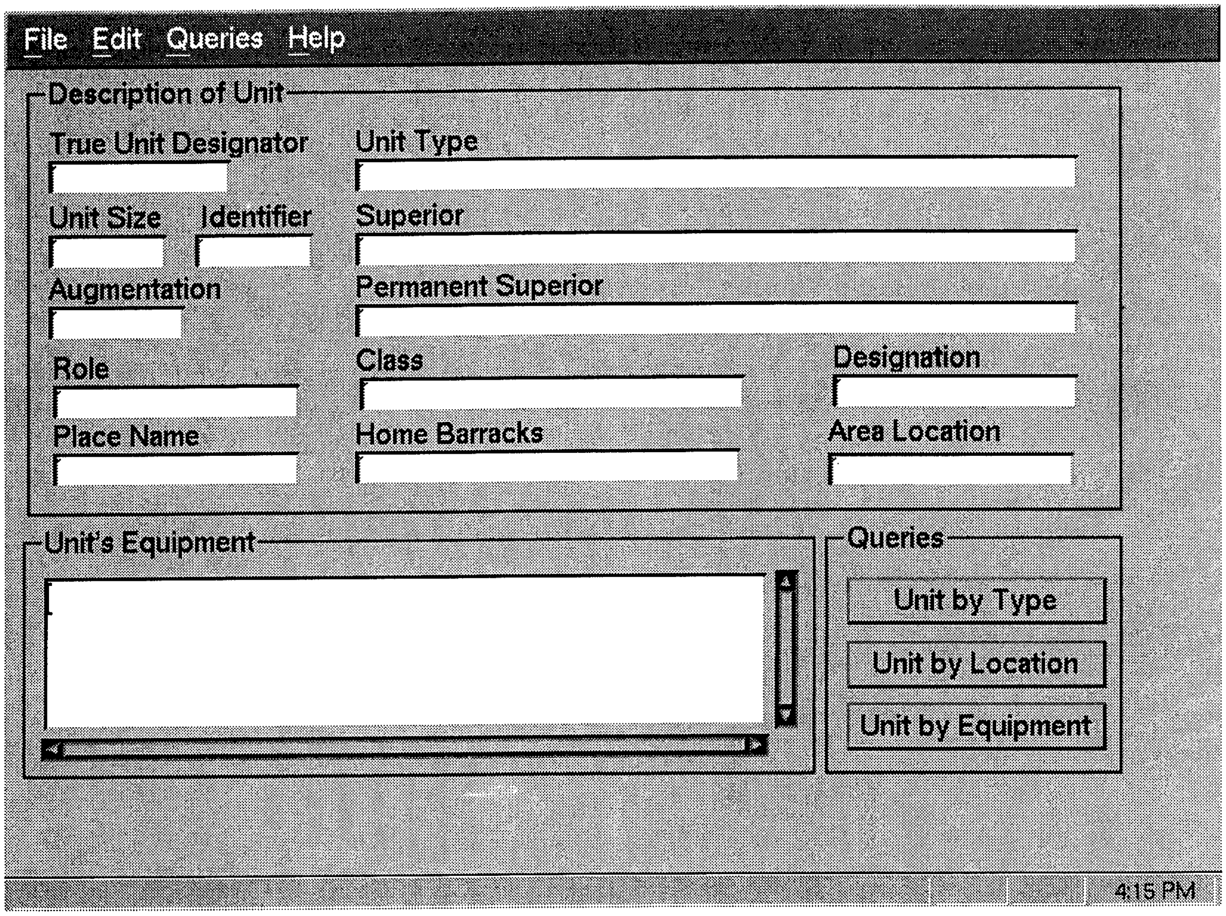

Figure 1. The main screen of the Demonstrator client program

Figure 1. is the screen that the user sees upon running the Demonstrator. Note that the equipment associated with the last unit displayed is listed in the Equipment window. Apart from the fact that the user must log in before running the Demonstrator, and the nature of some error messages, there are no visual cues to remind the user that the DCE services are being used by the application.

2.3 Demonstrator Architecture

The Demonstrator uses a three tier architecture: Graphical User Interface (GUI) based DCE clients, application servers and Database Management Systems (DBMS). Figure 2 depicts this architecture.

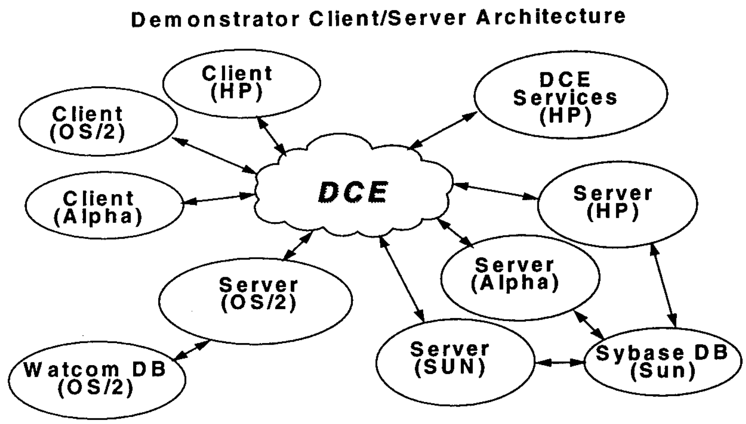

Figure 2. The software architecture of the Demonstrator system

Figure 2. emphasises the communication paths between the software components of the demonstrator. In comparison with a more traditional client server architecture, there is an extra layer of software (Figure 3. below); and there is overhead associated with each layer of software. However, it is this extra layer of software that enables features such as interoperability, survivability and scalability to be achieved.

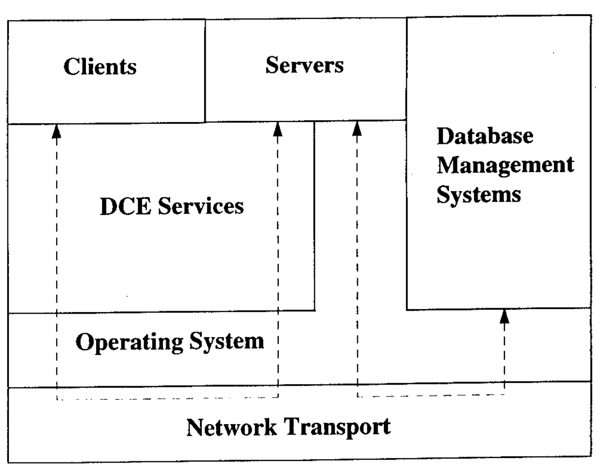

Figure 3. Relationship of software layers of the Demonstrator system

Of interest is the fact that any one of the Database Management Systems (DBMS) can be seen as a legacy system to which the application servers must interface. All of the features of this system could still be obtained if the data resided on a legacy mainframe computer. So for example, the Sybase DBMS in Figure 2. could be replaced by a legacy system, provided that the application servers had been modified to communicate with the legacy system. In fact, the Demonstrator is in the process of being ported to an IBM Mainframe under MVS OpenEdition.

3. FEATURES OF THE DEMONSTRATOR

The features demonstrated by the Defence DCE demonstrator are survivability against server failure, reliability, scalability, security, portability and interoperability. For the purposes of this report, the following definitions will apply.

- **Survivability** requires that a client program that utilises the services of a server program can continue processing in the event of a server failure or a communications failure.

- **Reliability** requires that the system is capable of running for long periods of time and with high processing loads, without crashing or producing incorrect results.

- **Scalability** requires that a system can be extended by adding more servers and/or processing power in an incremental way to allow more users (client programs) to use the system. The system should not become unmanageable with increasing size. Ideally, the performance of a scalable system should increase in a linear fashion with the number and processing power of the servers; in practice there are many factors affecting system performance.

- **Security,** in the context of this report, refers to the commercial level security provided by DCE. This has no direct relation to any Defence information security (INFOSEC) requirements; however, it is significantly stronger than UNIX system security. The main requirements are authentication (verifying that users, client software and server software are valid) and authorisation (verifying that users, clients and servers are allowed access to requested resources).

- **Portability** requires that a program developed on one type of computer can be recompiled on another type of computer with a minimum of changes to the source code associated with that program. Ideally, no changes should be required in the source code.

- **Interoperability** requires that a program running on one type of computer can communicate with a program running on another type of computer, without any consideration or knowledge of what type of computer it is communicating with.

3.1 Survivability

The Demonstrator provides survivability against server failure and communications failure between client and server. When a failure occurs such that the client/server communications are broken, the client program automatically establishes communications with another server (if available) and continues processing, with a maximum of 10 minutes delay. Note that no user or administrator is required to intervene to re-establish communications. Typical recovery time is about 1 minute and 40 seconds when the communication link is destroyed by literally disconnecting the server machine from the LAN/WAN.

Survivability has been enabled both by the DCE services and the software design of the Demonstrator system. The DCE services allow the client program to find another server program easily. The design of the system is that the ORBAT data is replicated on several servers. Since this system allows data retrieval only, and no updates, replicating the data is a simple process. This means that it is straightforward to have multiple servers running at the same time, to which clients can connect. It should be stated that this is simplistic and that many applications require update capability. Distributed updates of a replicated database presents a number of difficult problems in terms of performance and accuracy of data. The problems arise because it is difficult to guarantee that a distributed transaction, such as an update, has completed successfully on *every* database. Refer to [3,4] for a discussion of some proposed solutions. The distributed database update problem is not addressed in this report for reasons which follow.

- The problem does not relate directly to DCE.
- The solution is a performance / capability tradeoff and is very dependent on the situation to which it is applied.
- The problem has been solved for specific sets of requirements including some commercial products.

3.2 Reliability

Reliability as defined above depends upon each component in the entire system. In the case of the Demonstrator, this means that there is a dependence upon the operating system, DCE services and the application code to deliver reliability. The Demonstrator has undergone stress/endurance tests for more than 24 hours without failing. The stress / endurance tests consisted of 4 independent clients which continuously retrieve data from the server. These test clients have 5 preselected queries which are then selected at random and data is retrieved. This system puts both the database server and the DCE application server under some stress. During the 24 hour period, several million rows of text were retrieved from the database. The successfully completed tests give a degree of confidence in the reliability of the Demonstrator, and hence the DCE services. It should be stated that the Demonstrator has been developed as a prototype and not an operational system, hence more comprehensive testing is not required.

3.3 Scalability

DCE is managed and administered on the basis of cells. A DCE cell is a group of DCE enabled computers that share a security service, a Cell Directory Service (CDS which provides the naming and locating function), a time service and possibly a Distributed File Service (DFS). For more background information about DCE refer to [5].

The DSTO DCE cell currently consists of some five DCE enabled computers which limits the ability to test the scalability of the system. However, the architecture of the system is such that it is expected to scale well. The Demonstrator clients are configured to connect to the first server returned from the Cell Directory Service (CDS). Each server can be set up to only accept a number of clients commensurate with the processing power available to that server. If a client is refused service on a particular server then it will attempt to connect to another server. This means that provided a sufficient number of servers are running to support the user base, the system could scale up to thousands or tens of thousands of users.

DCE services are known to scale up to tens of thousands of users [6,7].

3.4 Security

The level of security exhibited by the Demonstrator is that provided by the DCE security services. The version of DCE in use at DSTO was acquired through normal commercial channels and as such does not provide encryption of user data, which is subject to US export restriction laws. Defence is able to acquire the full version where required. All other security features are implemented by the Demonstrator, including the features that follow.

- Mutual authentication of clients and servers via the security service.
- Authorisation of clients by servers, based on security group membership.
- Integrity and sequence number of packets guaranteed.

The Demonstrator is able to detect imposters, both clients or servers and valid users whose login context is invalid. In DCE the default login context time is 8 hours, after which users need to re-authenticate themselves by running a utility and typing in their passwords.

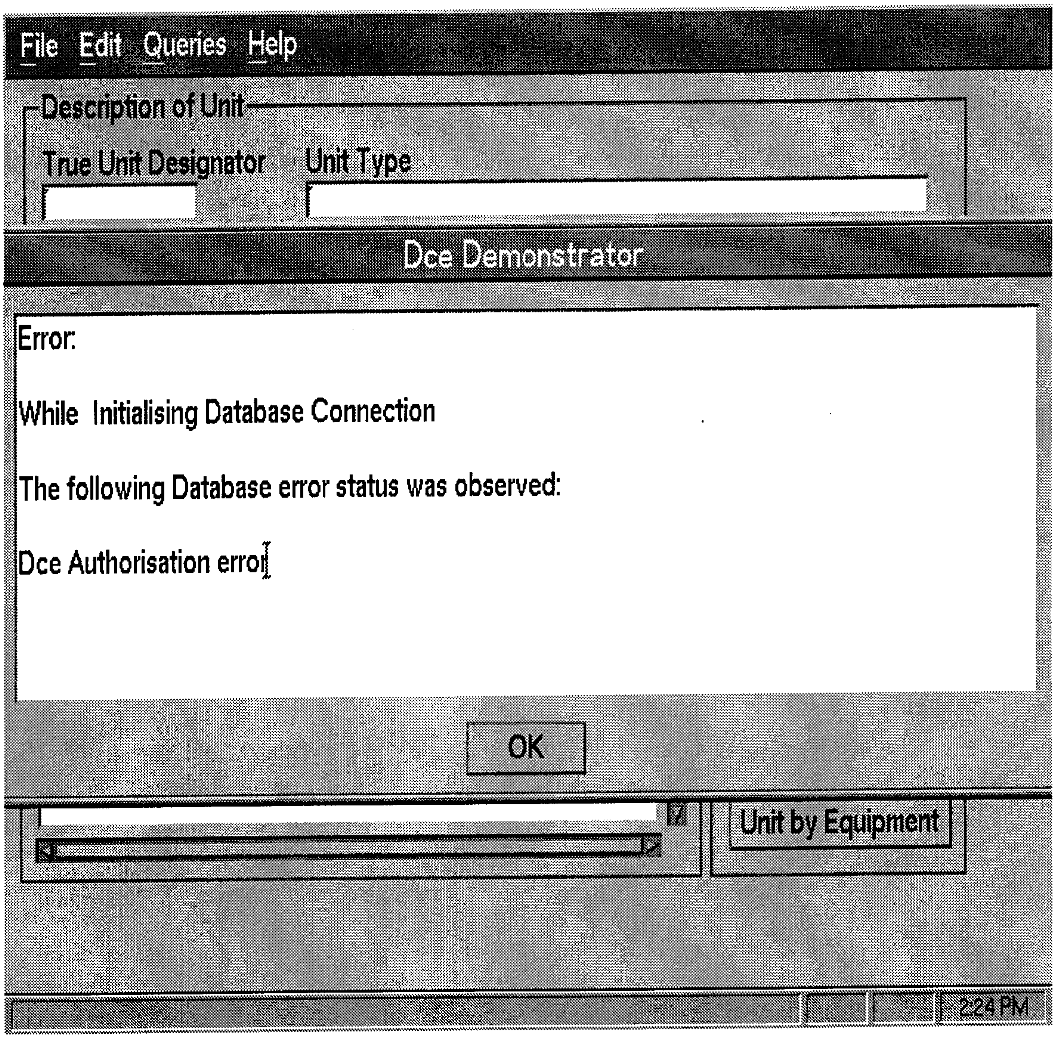

Figure 4. Demonstrator client program when user unauthorised

When a user who is not logged in with the appropriate privileges attempts to use the system, the server will detect this and the result at the client will be that displayed in Figure 4. above. At the server, a message "Client not Authorised" is printed out. It would be possible to modify the server so that, for example, an email message was automatically sent to the administrator.

It should be noted that there is at least one security weakness in the Demonstrator architecture. That is the connection between the server and the DBMS. The Sybase database that was used is available on the network, and the communications between the server and the DBMS are not DCE communications. One solution to this problem is to place the DBMS and the server on the same computer, and disable network access to the database directly. Another approach is to use security mechanisms provided by the database vendor. The latter approach has the disadvantage of being non-portable.

3.5 Portability

The HP, DEC Alpha and OS/2 client programs were all source code compatible. There were some changes necessary to port the server code from OS/2 to the DEC Alpha. The Alpha, HP and SUN servers remain source code compatible. The changes required to port from OS/2 to DEC Alpha can be categorised as follows:

(i) Database related

(ii) Signal handling related

The database related changes are a direct result of the fact that the OS/2 server gets its data from the Watcom Database Management System (DBMS), whereas the DEC Alpha, HP and SUN servers retrieve data from a Sybase DBMS running on a remote SUN. These changes were of the order of 150 lines of 'C' code. The Signal handling changes are peculiar to the OS/2 implementation of DCE and are described in some detail in [8]. The order of these changes is approximately 10 lines of 'C' code.

It can be seen that both DCE and the Demonstrator afford a high degree of portability.

3.6 Interoperability

The Demonstrator allows the user to run the client on HP, DEC Alpha, or OS/2 machines, and obtain services from a server running on any of those platforms. Just as important, the client and server programs are unaware of which type of computer they are interacting with or where that computer resides. This represents a high degree of interoperability and was attained with relatively little effort. Further, by conforming with the DCE requirement for interface definitions, the Demonstrator has the potential for more interoperability. For example, assume that another DCE application was to be developed, say for another group of users that would be connected to the same network and required access to the same data. This new application could access the ORBAT data through the predefined interface of the Demonstrator rather than directly interacting with the DBMS. The DBMS could later be changed (eg: from Sybase to Oracle) and the Demonstrator's server code updated if necessary to reflect the new DBMS. Provided no change was made to the Demonstrator client / server interface, neither the new application nor the Demonstrator client program would require any modification.

4. THE DEVELOPMENT PROCESS

This section describes the process used to develop the Demonstrator and the effort required to implement the various software modules. Figure 5. below depicts the development of each major software module.

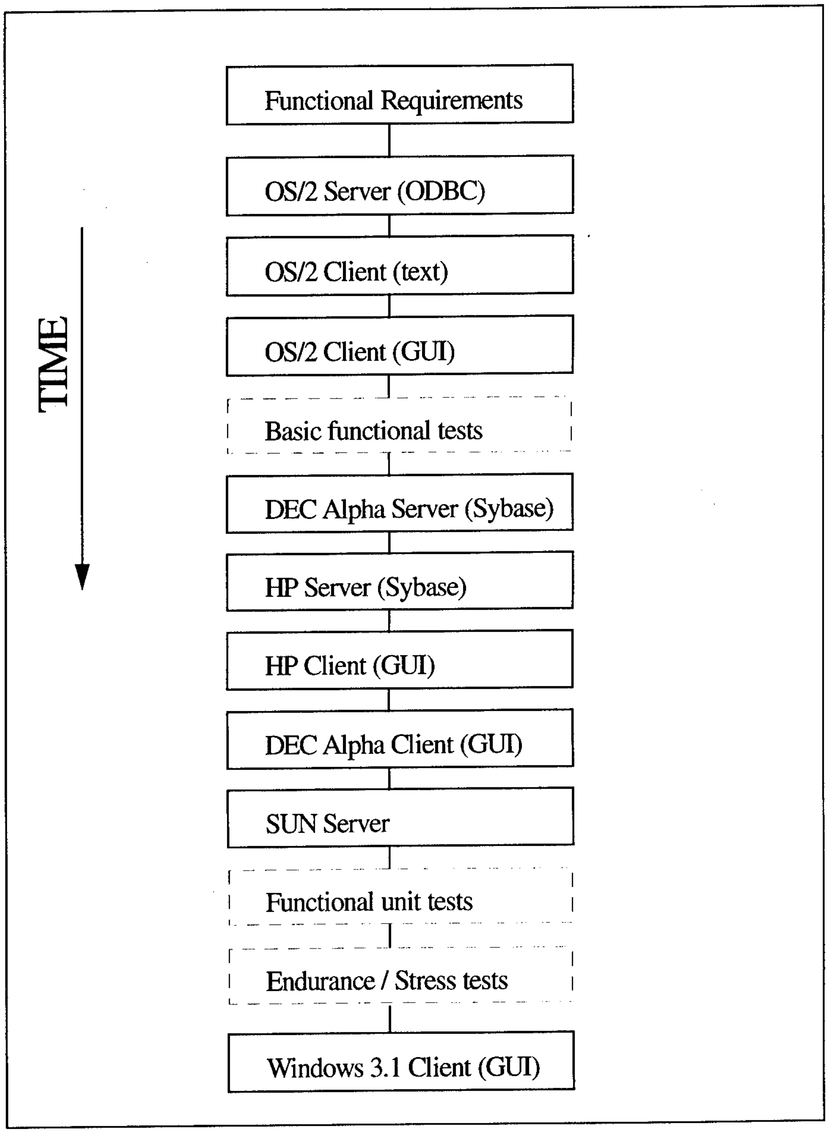

Figure 5. The Software Development Process

4.1 Software Development

The development of the software began with the implementation of the interface definition file. From there, an application server which interacts with a Watcom DBMS via Open Database Connect (ODBC) was developed. A text based client was then developed to allow initial testing of the system.

The Z App development suite for OS/2 and Motif was then procured. This package was selected because it supports source code compatibility across OS/2, UNIX/Motif and Windows 3.1. In addition, the source code for the libraries was provided, which enables them to be compiled with the same options as the rest of the system. A GUI version of the client was then developed (OS/2). This system was then extended to incorporate the security features. This process was slightly complicated by the requirement to understand DCE threads and exceptions mechanisms.

Following some testing, the clients and servers were ported to the DEC Alpha, Sun and HP platforms. A client only was ported to Windows 3.1. See Section 3.5 for a description of problems encountered during porting. It is estimated that approximately 50 programmer-days of effort was expended in developing, porting and testing the Demonstrator. This includes time to develop a security class, and some survivability code that can be reused. The estimate does not include all DCE training time.

In approximately 50 programmer-days, the Demonstrator was developed from an ASCII text database file and a functional requirement. This process involved designing the client/server interface, developing database code, building a GUI client that could interact with DCE, porting to several platforms and testing the resulting system. This process involved some system-level programming.

5. A COMPARISON WITH OTHER DEVELOPMENT APPROACHES

This section compares the development process described in the section above with what might be required if DCE was not used. In particular, two of the more popular communication mechanisms: Database connectivity tools and Sockets programming are compared with the DCE approach.

5.1 Database Client/Server Development Tools

Database client/server development tools such as Informix Hyperscript Tools, Paradox for Windows and Visual Basic are becoming common. These systems typically come with integrated GUI development systems and high level languages which enable access to a DBMS. A system such as the Demonstrator, which is essentially a client server database is likely to have been built in less time, with less training requirements if a client/server development tool was used. The fact that no server would have to be developed would produce a significant saving in terms of development effort.

However, some of the features such as security, survivability and scalability are less likely to have been provided. In particular, provision of security and survivability will probably preclude interoperability with other database vendors. Furthermore, although the Demonstrator is database-centric, many applications have other forms of information such as documents and geographic information which need to be distributed.

It is clear that database client/server development tools are very useful when applied to certain sets of requirements, but are unable to provide the full range of features that DCE supports.

5.2 TCP/IP Sockets Programming

TCP/IP Sockets programming has been available for a long time, and is a flexible way to achieve system wide communication. It should be noted that the entire DCE services are built upon the sockets facility, but the details are hidden from the developer. The sockets communication mechanism consists of establishing a connection between the two computers and sending messages from the local socket to the remote one. The development cycle would be similar to that used for DCE, but may take longer depending on the features provided. A similar architecture could have been used to achieve the goals of surviveability and security.

There are several drawbacks to this approach.

- The developer must co-ordinate addresses and socket numbers.
- The developer must define the meaning and format of the message content.
- The security mechanism must be sourced or developed from scratch.

- The byte order and character set differences could cause problems in porting the system to some platforms.

These drawbacks imply that it is likely that the development of the Demonstrator would have taken longer using sockets than DCE. Further, the solutions to security and resource locating (some form of naming is required if survivability is to be achieved) are unlikely to scale well.

5.3 Summary of the comparison

Figure 6. provides a summary of the features that can easily be provided by the various approaches to distributed systems development.

Feature Provided	Database Connectivity	TCP/IP Sockets	DCE
Portability	✓	✓	✓
Interoperability	✓[1]	X	✓
Security	✓[2]	✓[3]	✓
Scalability	✓[2]	X	✓
Reliability	✓	✓[4]	✓[4]
Survivability	✓[2]	✓[3]	✓[4]

Figure 6. A Comparison of Features

NOTES:

1 Only provides database interfaces.
2 Not a standard feature, and must be provided by the database vendor. Use of such features may reduce interoperability.
3 Is possible but must be provided by the system developers and can be difficult.
4 Can be provided and is dependent on the competence of system developer.

In summary, the database client/server development tools would have produced a working system in less time and with less training or expertise but would not meet all requirements. The sockets mechanism is flexible and could have met all requirements, but some features would have to be crafted from scratch or third party systems integrated. This would have taken longer and may be difficult to scale up at a later time.

6. CONCLUSIONS

The features of survivability, reliability, security, portability and interoperability have been demonstrated by the Defence DCE demonstrator. This amounts to a verification of many of the claims made about DCE. Although the development process for DCE systems is more in depth for a simple system such as the Demonstrator compared with a traditional client/server development, the result is a more flexible and open system. For large distributed systems these features could actually reduce the development/testing cycle. The comparison of the development process using DCE, database client server development tools, and TCP/IP socket programming emphasises the features offered by DCE. The experience gained in developing the Defence DCE database demonstrator will enhance further research, reports and ad-hoc advice provided to the ADF under the Distributed Systems Technology Task (ADF 94/151).

REFERENCES

1. JCSE Request For Tender (JP2030-01/94) Volume 2, Statement of Requirement, April 1994.

2. ADFDIS Implementation Reference Model version 2.0, 15 August 1995.

3. R. Elmasri and S. Navathe, 'Fundamentals of Database Systems', The Benjamin/Cummings Publishing Company, Inc., 1989.

4. M. Ozsu and P. Valduriez. 'Principles of Distributed Database Systems', Prentice-Hall International Inc., 1991.

5. Ward Rosenberry, David Kenney and Gerry Fisher,'Understanding DCE',O'Reilly and Associates, Inc., 1992.

6. Mark Carter, 'Adding 50,200 Users to a DCE Registry: A comparison of OSF DCE V1.0.2 and IBM DCE/6000 V1.2' CITI report February 1994.

7. Personal visit (J.Mansfield) to Project Pilgrim, May 1995.

8. B.McClure and D.O'Dea, 'An Experience in Cross Platform DCE Development', DSTO Technical Note 1995.

Development of a Defence Distributed Computing Environment (DCE) Database Demonstrator

B. McClure and J.Mansfield

(DSTO-CR-0008)

DISTRIBUTION LIST

	Number of Copies
Defence Science and Technology Organisation	
Chief Defence Scientist and members of the)	1 shared copy
DSTO Central Office Executive)	for circulation
Counsellor, Defence Science, London	(Document Control sheet)
Counsellor, Defence Science, Washington	(Document Control sheet)
Senior Defence Scientific Adviser)	1 shared copy
Scientific Adviser - POLCOM)	
Assistant Secretary Science Industry Interaction	1
Director, Aeronautical & Maritime Research Laboratory	1
Electronics and Surveillance Research Laboratory	
Chief Information Technology Division	1
Research Leader Command & Control and Intelligence Systems	1
Research Leader Military Computing Systems	1
Research Leader Command, Control and Communications	1
Head Information Management and Fusion Group	1
Executive Officer, Information Technology Division	(Document Control sheet)
Manager, Human Systems Integration Group	(Document Control sheet)
Head, Software Engineering Group	(Document Control sheet)
Head, Trusted Computer Systems Group	1
Head, Advanced Computer Capabilities Group	(Document Control sheet)
Head, Intelligence Systems Group	(Document Control sheet)
Head, Systems Simulation and Assessment Group	(Document Control sheet)
Head, Exercise Analysis Group	(Document Control sheet)
Head, C3I Systems Engineering Group	(Document Control sheet)
Head, Computer Systems Architecture Group	(Document Control sheet)
Head Trusted Computer Systems Group	1
Head Command Support Systems Group	1
Mr B.McClure, Command Support Systems Group	1
Mr J. Mansfield, Command Support Systems Group	1
Publications and Publicity Officer, ITD	1
Defence Intelligence Organisation	
Principal Research Scientist (R&D)	1
Assistant Secretary, Information Technology	1
HQADF	
Director General, Force Development (JOINT)	1
Director General, Force Development (SEA)	1
Director General, Force Development (LAND)	1
Director General, Force Development (AIR)	1
Director, Operational Information Systems	1
DD-EW	1
Navy	
Navy Scientific Adviser (NSA)	1
Army	
Scientific Adviser, Army (SA-A)	1
Project Director, AUSTACCS	1

Air	
Air Force Scientific Adviser (AFSA)	1
CO, Electronic Warfare SQN	1
ACQUISITION AND LOGISTICS PROGRAM	
Director General, Information Management and Communications Engineering	1
Project Director, ADFDIS	1
Project Director, JP2030	1
Libraries and Information Services	
Defence Central Library - Technical Reports Centre	1
Manager Document Exchange Centre (MDEC) (for retention)	1
Additional copies which are to be sent through MDEC	
DIS for distribution:	
National Technical Information Centre. United States	2
Defence Research Information Centre, United Kingdom	2
Director Scientific Information Services, Canada	1
Ministry of Defence, New Zealand	1
National Library of Australia	1
Defence Science and Technology Organisation Salisbury, Research Library	2
Library Defence Signals Directorate Canberra	1
AGPS	1
British Library Document Supply Centre	1
Parliamentary Library of South Australia	1
The State Library of South Australia	1
Spares	
Defence Science and Technology Organisation Salisbury, Research Library	6

Page classification: UNCLASSIFIED

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION
DOCUMENT CONTROL DATA

1. PRIVACY MARKING/CAVEAT (OF DOCUMENT)	N/A
2. TITLE	Development of a Defence Distributed Computing Environment (DCE) Database Demonstrator
3. SECURITY CLASSIFICATION (FOR UNCLASSIFIED REPORTS THAT ARE LIMITED RELEASE USE (L) NEXT TO DOCUMENT CLASSIFICATION)	Document (U) Title (U) Abstract (U)
4. AUTHOR(S)	Bruce McClure and John Mansfield
5. CORPORATE AUTHOR	Electronics and Surveillance Research Laboratory PO Box 1500 Salisbury SA 5108
6a. DSTO NUMBER	DSTO-CR-0008
6b. AR NUMBER	AR-009-460
6c. TYPE OF REPORT	Client Report
7. DOCUMENT DATE	January 1996
8. FILE NUMBER	
9. TASK NUMBER	ADF 93/315
10. TASK SPONSOR	DGFD (Joint)
11. NO. OF PAGES	26
12. NO. OF REFERENCES	8
13. DOWNGRADING/DELIMITING INSTRUCTIONS	N/A
14. RELEASE AUTHORITY	Chief, Information Technology Division

15. SECONDARY RELEASE STATEMENT OF THIS DOCUMENT

APPROVED FOR PUBLIC RELEASE

OVERSEAS ENQUIRIES OUTSIDE STATED LIMITATIONS SHOULD BE REFERRED THROUGH DOCUMENT EXCHANGE CENTRE, DIS NETWORK OFFICE, DEPT OF DEFENCE, CAMPBELL PARK OFFICES, CANBERRA ACT 2600

16. DELIBERATE ANNOUNCEMENT

No limitation

17. CASUAL ANNOUNCEMENT — No limitation

18. DEFTEST DESCRIPTORS

- Distributed computer systems
- Computer programs
- Command control communications and intelligence
- Interoperability
- Computer program portability
- Computer security

19. ABSTRACT

This report discusses the development of a Defence Distributed Computing Environment (DCE) database demonstrator program. The Demonstrator program showcases the interoperability, portability, survivability and security features of Open Software Foundation's Distributed Computing Environment.

Page classification: UNCLASSIFIED