NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

MULTI-AGENT ARCHITECTURE FOR INTEGRATING REMOTE DATABASES AND EXPERT SOURCES WITH SITUATIONAL AWARENESS TOOLS: HUMANITARIAN OPERATIONS SCENARIO

by

Cantemir Ahciarliu

March 2004

Thesis Advisor: Alex Bordetsky
Co-Advisor: Glenn Cook

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			*Form Approved OMB No. 0704-0188*
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
1. AGENCY USE ONLY ***(Leave blank)***	**2. REPORT DATE** March 2004	**3. REPORT TYPE AND DATES COVERED** Master's Thesis	
4. TITLE AND SUBTITLE: Multi-Agent Architecture for Integrating Remote Databases and Expert Sources with Situational Awareness Tools: Humanitarian Operations Scenario			**5. FUNDING NUMBERS**
6. AUTHOR(S) Cantemir Ahciarliu			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited			**12b. DISTRIBUTION CODE**

13. ABSTRACT (maximum 200 words)

Complex Humanitarian Emergencies are usually military conducted activities where participants must be able to react to a very dynamic and unfriendly environment. National and international participating forces require cooperation and coordination between civilian and military entities. The continuous need to share huge amounts of information requires a technological framework to allow legacy and new hardware and software interconnection, rapid network installation and flexible bandwidth availability. To improve the speed and the quality of the decision-making a scientific approach must be applied to the process. Maximizing both the effectiveness and efficiency in decision-making can be obtained by developing decision support systems capable of providing access to existing databases and expert systems. Databases usually contain raw information available for retrieval and processing according to the needs of the decision makers. Expert systems embed human expertise and allow the propagation of scarce expert resources throughout an organization to increase the consistency and quality of the decisions. Sharing access to these types of information within a Complex Humanitarian Emergency environment provides for better situational awareness and improves the decision-making process. This thesis will gather and combine the information from different sources and will suggest a model for integrating remote databases and expert sources with situational awareness tools.

14. SUBJECT TERMS Complex Humanitarian Emergencies, Peer-to-Peer Collaborative Software, Decision-Making, Databases, Situational Awareness Tools, Agents, Web-Enabled Databases			**15. NUMBER OF PAGES** 95
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UL

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

MULTI-AGENT ARCHITECTURE FOR INTEGRATING REMOTE DATABASES AND EXPERT SOURCES WITH SITUATIONAL AWARENESS TOOLS: HUMANITARIAN OPERATIONS SCENARIO

Cantemir M. Ahciarliu
Lieutenant Colonel, Romanian Air Force
B.S., Military Technical Academy, Bucharest, 1983

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN INFORMATION TECHNOLOGY MANAGEMENT

from the

NAVAL POSTGRADUATE SCHOOL
March 2004

Author: Cantemir Ahciarliu

Approved by: Alex Bordetsky
Thesis Advisor

Glenn Cook
Co-Advisor

Dan C. Boger
Chairman, Department of Information Sciences

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Complex Humanitarian Emergencies are usually military conducted activities where participants must be able to react to a very dynamic and unfriendly environment. National and international participating forces require cooperation and coordination between civilian and military entities. The continuous need to share huge amounts of information requires a technological framework to allow legacy and new hardware and software interconnection, rapid network installation and flexible bandwidth availability. To improve the speed and the quality of the decision-making a scientific approach must be applied to the process. Maximizing both the effectiveness and efficiency in decision-making can be obtained by developing decision support systems capable of providing access to existing databases and expert systems. Databases usually contain raw information available for retrieval and processing according to the needs of the decision makers. Expert systems embed human expertise and allow the propagation of scarce expert resources throughout an organization to increase the consistency and quality of the decisions. Sharing access to these types of information within a Complex Humanitarian Emergency environment provides for better situational awareness and improves the decision-making process. This thesis will gather and combine the information from different sources and will suggest a model for integrating remote databases and expert sources with situational awareness tools.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I would like to express my sincere appreciation to my advisors, Dr. Alex Bordetsky and Glenn Cook for their guidance throughout my experience at NPS. Their advice and direction paved the way for the accomplishment of this thesis.

I wish to express my gratitude to all my teachers in Romania who have laid the foundation bricks of my education, as well as my professors at NPS.

I would like to honor the memory of my parents who made all my life’s achievements possible by teaching me the most important lessons in life. Finally, I wish to thank my son, Titus, for giving me the greatest reason to succeed.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. AREA OF RESEARCH

Natural or human induced tragedies are permanent challenges for our societies, which more than ever strive to find a proper means to relieve human suffering. Complex Humanitarian Emergencies (CHE) are an ongoing international effort sustained by civilian and military forces willing to better understand each other and to harmonize their cooperation.

This research is focused on the information requirements and technology implementations designed to help the actors playing a part in humanitarian operations. The rapid evolution of Information and Communication Technology (ICT) extends the capability and the means to access and to share information. The information revolution imposes the qualitative transformation of the information by increasing both its quality and its availability. This study will not analyze the means of how to obtain the required level of information richness but how to make it available to those who need it.

Consequently, different sources of information like: data bases, web pages, expert systems have to be integrated with the existing collaborative software in order to generate improved situational awareness in CHE. This study will develop a multi-agent architecture based on DARPA's Control of Agent-Based Systems (CoABS) methodology which will improve the situational awareness by sharing access to remote databases and expert sources.

B. RESEARCH QUESTIONS

- **Primary research question:** How to improve situational awareness in Complex Humanitarian Emergencies and Peace Operations by sharing access to remote databases and expert sources?
- **Secondary research question:** How can the proposed architecture enable information sharing in a peer-to-peer collaborative environment?

C. DISCUSSION

Complex Humanitarian Emergencies are usually militarily conducted activities where participants must be able to react to a very dynamic and unfriendly environment. These involve large numbers of refugees or internally displaced persons, gross violations of human rights, and generate international response from different organizations or agencies. As a result, CHE's involve national and international participation that require proper cooperation and coordination between civilian and military entities. The need to share huge amounts of raw information requires a technological framework to allow legacy and new hardware and software interconnection, rapid network installation and flexible bandwidth availability.

Former Naval Postgraduate School (NPS) students have explored the use of technology to enhance collaboration and coordination among participants in a CHE by developing a web based application used as a testbed for off-the-shelf technologies. As a continuation of these efforts, another thesis has focused on the development of a collaborative, Information Technology-based, operational support system designed to facilitate information sharing at the field level during CHE and Peace Operations. The gathered information applies to the existing technology used by different civil and military organizations in the process of information sharing during a CHE.

To improve the speed and the quality of the decision-making a scientific approach must be applied to the process. Maximizing both the effectiveness and efficiency in decision-making can be obtained by developing decision support systems capable of providing access to existing databases and expert systems. Databases usually contain raw information available for retrieval and processing according to the needs of decision makers. Expert systems embed human expertise and "allow the propagation of scarce expert resources throughout an organization to increase the consistency and quality of the decisions". Sharing access to these types of information within a CHE environment provides for better situational awareness and improves the decision-making process.

A vast domain of research was opened by the DARPA initiated Control of Agent-Based Systems (CoABS) program which is focused "to develop and demonstrate techniques to safely control, coordinate, and manage large systems of autonomous software agents" [Ref. 1]. Based on this theoretical approach and on the information gathered by the former researches, this thesis will propose a model to facilitate situational awareness by accessing remote databases and expert sources using multi-agent architecture based on DARPA CoABS.

D. SCOPE OF THE THESIS

The goal of this thesis is to gather and combine the information known from different sources and to define a model for integrating remote databases and expert sources with situational awareness tools. The role of the CoABS solution is that of middleware used to access the different types of databases which contain the required data in a raw form. Evaluations and proposals will be based on the information and results gathered during former NPS conducted experiments and from the Exercise Cobra Gold 2003. The results will document future implementations, which may be deployed and tested during future Cobra Gold exercises.

E. METHODOLOGY

The research process will start by defining key concepts needed to understand the main characteristics of the Humanitarian Operations Scenario. The terms and definitions used are the ones agreed upon in official United Nations documents that tackle the interaction between civil and military entities working together within humanitarian operations. This documenting effort will serve the need to expose facets and phases where the use of information technology may increase the quality of the results, better address intrinsic risks and decrease financial burden.

The next step will explore several capabilities and models that facilitate information sharing across the network. Database technology and web-enabled solutions, collaborative tools and the Groove System, intelligent agents and the Control of Agent-Based Systems (CoABS) methodology are the foundation bricks to serve the proposed multi-agent architecture. Software products and in-house developed applications used during NPS conducted experiments and projects will provide examples for the theoretical notions.

Carried out under the auspices of DARPA's CoABS program, the Coalition Agents Experiment (CoAX) will serve as the main example of a technology based experiment focused on command support systems for Coalition Operations.

In order to classify the information exchanged during CHE a careful analysis will cover the *2003 Cobra Gold Exercise* which is an annual joint/combined, multilateral exercise. Its main focus in 2003 was on peace enforcement, humanitarian assistance and disaster relief operations.

Based on the foundation concepts of the Complex Humanitarian Emergencies and Peace Operations, using the theoretical support, the main effort will require applying CoAX experiments philosophy to the CHE environments like the 2003 Cobra Gold Exercise.

F. BENEFITS OF THE STUDY

By overlapping the CoAX experiments with the 2003 Cobra Gold Exercise we may extend the experience gained in a two years internationally conducted experiment proposing a framework applicable in future CHE environments.

The proposed solution will combine the multi-agent architecture built on the CoABS technology with the peer-to-peer model merging the advantages brought by both technologies towards improving the situational awareness. The possible multi-agent architecture is designed to integrate remote databases with situational awareness tools using CoABS, ROCC and Groove.

II. FOUNDATION CONCEPTS OVERVIEW

A. BACKGROUND

Modern world realities bring the necessity to improve the way our societies react and address natural or human induced tragedies. Large scale conflicts, regional wars, peace keeping operations and search and rescue missions bring together multiple actors who join their efforts in finding proper means to relieve human suffering. It is more likely that they represent not only different nations but different segments of the human society belonging to both civil and military organizations. The need to ensure the proper level of understanding between participants led to the adoption of a special set of key terms and their definitions, which create the foundation for intercommunication.

This paper explores information requirements and technology implementations designed to assist participants in data-sharing during humanitarian operations; for that reason Chapter II introduces key concepts and their meaning, and discusses main challenges and difficulties belonging to this area. This documenting approach is not supposed to turn into a domain dictionary for the key terms used, but to reveal facets and phases where the use of information technology may increase the quality of the results, better address intrinsic risks and decrease financial burden. Terms and definitions are the ones agreed upon in official United Nations documents that tackle the interaction between civil and military entities working together within humanitarian operations.

B. KEY TERMS AND DEFINITIONS

Complex Emergency: A complex emergency, as defined by the Inter-Agency Standing Committee (IASC), is “a humanitarian crisis in a country, region or society where there is total or considerable breakdown of authority resulting from internal or external conflict and which requires an international response that goes beyond the mandate or capacity of any single and/or ongoing United Nations country program.” [Ref. 2]

Humanitarian Assistance: Humanitarian assistance is aid to an affected population that seeks, as its primary purpose, to save lives and alleviate suffering of a crisis-affected population. Humanitarian assistance must be provided in accordance with the basic humanitarian principles of humanity, impartiality and neutrality. Based on the degree of contact with the affected population assistance can be divided into three categories. These categories are important because they help define which types of humanitarian activities might be appropriate to support with international military resources under different conditions:

- *Direct Assistance* is the face-to-face distribution of goods and services.
- *Indirect Assistance* is at least one-step removed from the population and involves such activities as transporting relief goods or relief personnel.
- *Infrastructure Support* involves providing general services, such as road repair, airspace management and power generation that facilitate relief, but are not necessarily visible to or solely for the benefit of the affected population.

An attribute of Complex Emergencies is that they bring together humanitarian agencies along with military forces which leads to different, sometimes opposing goals: military forces may be used to provide support and security for the activity of the humanitarian agencies; alternatively humanitarian workers become a target in conflict settings as the distinction between humanitarian and military forces become blurred; sometimes humanitarian objectives may be used as a justification for military intervention. Thus, an important task is to define guiding principles that cover all the aspects of civil-military interaction in Complex Emergencies.

The Humanitarian Imperative Comes First is the principle that requires all possible steps should be taken to prevent or ease human suffering arising out of a conflict or calamity and that civilians so affected deserve protection and assistance; therefore all civil-military interaction has to serve this humanitarian goal.

According to the 1949 Geneva Convention, *Respect for International Humanitarian Law and Instruments* means that humanitarian agencies must ensure that their activities maintain a clear distinction between combatants and non-combatants. Furthermore, humanitarian agencies must ensure that the civil/military interaction does not compromise *impartiality* which is their ability to deliver assistance regardless of race, creed or nationality and without adverse distinction. These organizations must also respect the neutrality principle which states that they should not act as instruments of government foreign policy.

A key element for humanitarian agencies and organizations when they deploy, consists of establishing and maintaining a so-called *humanitarian operating environment* (also referred as "humanitarian space"). Applying the key operating principles of neutrality and impartiality ensures that suffering is addressed wherever it is found; therefore a main task is to properly define roles and functions of humanitarian actors understanding that the generally agreed upon scenario offers the military the role of creating an operating environment in which humanitarian organizations can perform their tasks both effectively and safely.

As defined in the 1994 Oslo Guidelines, *Military and Civil Defense Assets* (MCDA) comprise relief personnel, equipment, supplies and services provided by foreign military and civil defense organizations for international humanitarian assistance. Furthermore, civil defense organization means any organization that, under the control of a Government, performs the functions enumerated in Article 61, paragraph (1), of Additional Protocol I to the Geneva Conventions of 1949. When under UN control, these assets are referred to as UN MCDA and they are used according to the guidelines that define the way they can support United Nations Humanitarian Activities in Complex Emergencies [Ref. 3].

C. INFORMATION SHARING DURING COMPLEX HUMANITARIAN EMERGENCIES

In the context of humanitarian assistance, the involvement of national and multinational armed forces has been extended considerably following the end of the Cold War. United Nations and non-UN humanitarian partners have decided to join efforts with the military to ensure coordinated and effective response to CHE. Between 1991 and 2001, the United Nations Security Council deployed a total of 36 peace missions, almost three times as many as over the previous period of 40 years. A real challenge when dealing with an unfriendly type of environment is the need to ensure security and coordination between field-based humanitarian and military organizations which leads to the necessity of information sharing between them. It is the common concern for security and the desire for stability that creates the base for communication and information sharing on security issues that include: locations of landmines and unexploded ordinance, direction and size of population movements, hostile activities or violence, inaccessible areas around vulnerable populations, incident and emergency situation reports, evacuation planning, and other kinds of events that potentially affect planning and delivery of relief.

This mutual interest for cooperation may be hindered by the different ways that reality may be perceived: a situation that might be a security concern for the military may not be one for the rest of the relief organizations involved. Interpreting the situation and transmitting an assessment about the nature and perception of threats and vulnerabilities to relief organizations is an important aspect to address properly in order to make this cooperation feasible.

Information sharing between humanitarian relief and military organizations requires that each entity clearly communicates its set of activities, assets, and limitations in order to build credibility, manage expectations, and build good, dependable relationships across organizational lines. At national and international level, policies/rules are developed in order to frame civil-military relationship under various conditions which ensure credibility and impartiality to the humanitarian joint effort.

Analyzing information flow between civilian and military communities, there are factors which limit information sharing such as: military cannot disclose classified information; humanitarian relief organizations are reluctant to share information that may look like "intelligence" about the conflict and thus threaten the security of their staff and operations in the field. Actually, humanitarian workers and agencies must not engage in gathering or provision of military sensitive information to military forces. The key to this problem is the development of humanitarian-military relationship based on the *civil-affairs personnel* whose role is basically as civil-military liaisons. [Ref. 4] They can filter classified from unclassified and "intelligence" from logistical information and deliver appropriately useful information to field-based relief organizations. Though they may be seen as not belonging to either of the two communities, by acting as a human interface, they can effectively improve humanitarian-military relationship.

Presuming the organizational part of this relation solved, the entire focus should go on the value of the shared information. Joint Vision 2010 emphasizes the need for developing information superiority, which can be understood as a state that is achieved when competitive advantage is derived from the ability to exploit a superior information position. [Ref. 5]. Relevance, accuracy and timeliness are dimensions of information, which are important for both military and commercial sectors; therefore the quality and timeliness of well-formulated data can be crucial for the success of both the humanitarian and military operations.

Pioneered in the commercial sector, network-centric concepts are the means to obtain information superiority by allowing the organizations to achieve shared awareness and self-synchronization. In the military, the generic term C4I was introduced to put together the process of decision-making (command and control) with the information technology assets (computers and communication). C4I systems provide information about the location and status of enemy and friendly forces but they cannot guarantee superior decision making.

When joining the C4I structure of the military community with a similar existing structure in the humanitarian community, the most critical and challenging element is communication. Instead of trying to plan and coordinate every action before deployment, a far more useful and effective approach would be to share the information collected in the field between participants, consequently formulating and adapting plans according to the reality on the ground. The main achievement would be actors’ capability to engage in more dynamic operations.

Efforts have been made by the United Nations, following the end of the Cold War to start a targeted effort towards coordination of humanitarian assistance. Different new created organizations and structures share responsibilities in defining and organizing the humanitarian assistance domain.

United Nations Office for the Coordination of Humanitarian Affairs (OCHA) was established: to approach policy development and coordination functions designed to address all humanitarian issues; to advocate humanitarian issues with political organs, notably the Security Council; and to ensure the coordination of humanitarian emergency response [Ref. 6].

OCHA fulfills its coordination function through the *Inter-Agency Standing Committee* which provides a forum that brings together a broad range of UN and non-UN humanitarian partners including UN humanitarian agencies, the International Organization for Migration (IOM), three consortia of major international non-governmental organizations (NGO) and the Red Cross movement. The primary role of the IASC is to formulate humanitarian policy to ensure coordinated and effective humanitarian response to both complex emergency and to natural disasters.

OCHA tried to address the problem of creating a common information-sharing framework meant to assist all the parts involved in the process of comparing, reconciling and evaluating information for decision-making. The so-called *SHARE* (Structured Humanitarian Assistance REporting) approach requires that important data used in emergency relief and recovery operations should always entail:

- Geo-referenced information signifying where the data was collected, such as the latitude and longitude.
- A time-stamp, particularly important in fast-moving emergencies where information validity is relative to time, indicating when data was collected and at what frequency.
- Information about the data itself defining its credibility such as: the source of the information, the measurement standards and indicators used for the applied methodology.

The great benefit of collecting and organizing information following these principles allows the efficient use of databases which can be used, updated not only by a single source, the originator. Other entitled organizations may benefit from this dissemination of information and it was through implementations like the ReliefWeb, that valuable and time-critical information was made available in a more accessible and convenient way than through fax or mail. This proposed framework enables a friendlier representation of the available information using basic maps enriched with overlapped information, which can be kept up-to-date. Traditional form of sharing information-coordination meetings is replaced by a more efficient procedure that ensures needed information accessibility, regardless of time and schedules.

Back to the humanitarian – military communication issues, information sharing can be improved through real-time communication methods. An old-fashioned approach is to use radio links for implementing a 911 facility used to alert the actors of significant unknown/unexpected events or evolutions.

Other ways to obtain effective civil-military coordination rely on dedicated organizations such as Civil Military Cooperation Centers (CIMICs), Civil-Military Operations Center (CMOCs), and Humanitarian (Community) Information Centers (HCICs and HICs) which base their activity on open communication and information sharing between the humanitarian and local communities and the military community. These are physical sites including personnel specially trained to fulfill this mission.

In contrast, new solutions take advantage of the information and communication products by creating virtual information-sharing environments such as intranets, web sites, listservs, and electronic bulletin boards. New developed technologies manage to integrate situational awareness tools with existing databases and expert tools and they will be introduced in the following chapters.

Physical and virtual implementations are influenced in their effectiveness by both their location and their networking capacity. CIMICs or CMOCs are usually military-organized entities but they have to be available to the humanitarian community, outside the military compound; otherwise they discourage open cooperation due to inner military regulations. They must also easily obtain access to military command and logistics; therefore their position has to be adjacent to military facilities. Civilian-organized HCICs or HICs have to enclose an inner military component to enable information sharing between the two communities.

Under these circumstances, solutions that take advantage of cutting edge technologies, do not exclude the need for specialized personnel capable to provide a needed interpretive function between humanitarian and military entities by sorting, ordering, and establishing the relative value of information that should be shared between the various entities in a complex emergency.

III. INFORMATION SHARING WITHIN A NETWORK

A. BACKGROUND

Rapidly evolving network and computer technology, coupled with the exponential growth of services and information available on the Internet, brought us to the point where hundreds of millions of people have fast, pervasive access to a phenomenal amount of information. Through desktop machines at work, school and home, through televisions, phones, pagers, and car dashboards, from anywhere and everywhere. Today it is widely acknowledged that the major result of the information revolution is the qualitative transformation of the information domain.

Information is subject to a complex and continuous enrichment process which changes its most important attributes: information richness and information reach. Information richness is normally a measure of the quality of information, as opposed to information reach, which is a measure of the degree to which information can be shared. The entire evolution of the domain may be analyzed as a sequence of information technologies, which aimed to improve either or both information attributes.

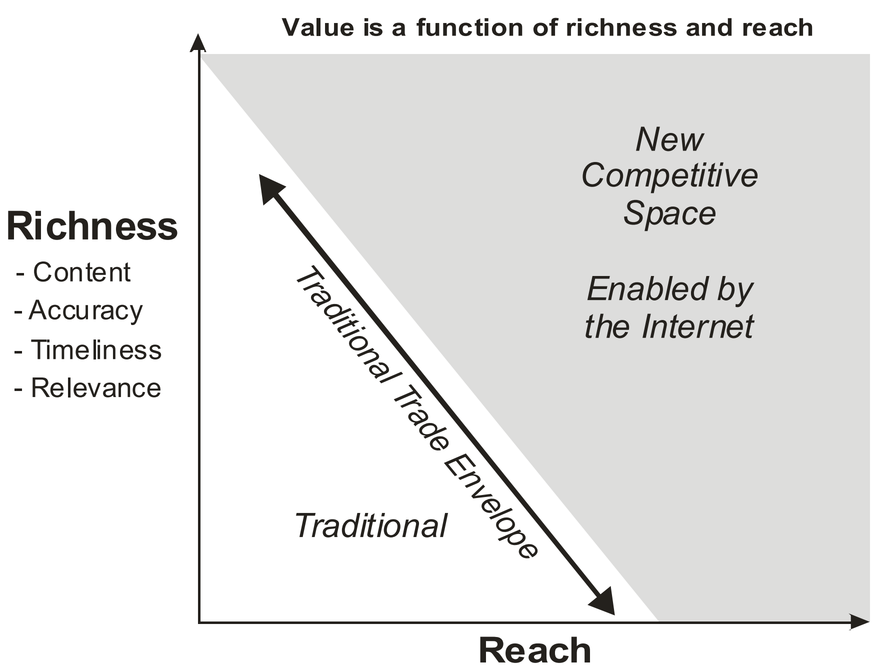


Figure 1. Information as Value Creation [After: Ref. 7:p. 74]

The intent of this paper is not to analyze the means of how to obtain the required level of information richness but to reveal existing opportunities to make it available to network users. It is the purpose of this chapter to present several capabilities and models that facilitate information sharing across the network. The main part of it brings an evolutionary approach to database technology centered towards the web-enabled existent solutions, and a glimpse on existing collaborative tools. A final focus is on Groove Workspace, which was the main collaborative tool in NPS experiments and projects

B. WEB-ENABLED DATABASES

The rise of the computers era made it clear that any future development was to be related to the capability of sharing valuable information between interconnected computers. To facilitate this inter-communication necessity, standardized protocols have been developed based on the newly developed networking technologies. In the sixties DARPA created a public network of computers that used the *Transmission Control Program/Internet Protocol (TCP/IP)* to communicate - the ARPANET. Although it was designed to connect major research centers, universities and military facilities, the continuous evolution of this network made it available to civilian organizations too. This is what we call today *the Internet*.

The Internet is currently serving millions of people connected through millions of computers, almost every business and government institution has a web page, and the web and web browsing are becoming the primary source of information in the human society. Its development has given rise to networking technologies that enable information sharing between remotely connected computers. Several important moments and architectures that have influenced database applications development will be introduced.

Data is an important organizational asset, which must be properly used, therefore the first system architectures tried to address enterprise database processing. This method used to allow multi-user access to database systems was *teleprocessing* which uses a single computer and one Central Processing Unit which performs all the processing. Users operate on dumb terminals (or microcomputers that emulate dumb terminals) that transmit transaction messages and data to the central computer. The operating system includes: a communication control portion (connected via communication lines to the terminals) which distributes the data to the appropriate running application, and a data management portion that is used by the Data Base Management System (DBMS) to process the database according to the requests made by the applications. The results of the completed transaction are sent back to the users using the same communication control portion and communication lines. [Ref. 8:p. 516). Figures 2-7 are based on diagrams presented in [Ref. 8].

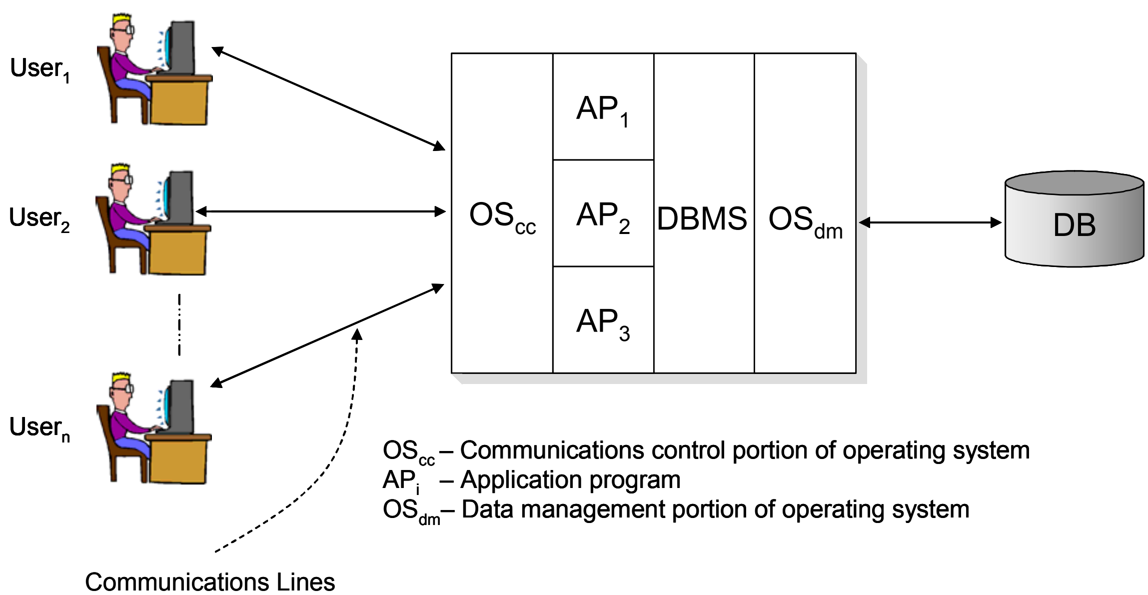


Figure 2. Teleprocessing System Architecture.

Although the use of dumb terminals determined a primitive users' interface which was generally character-oriented, teleprocessing systems were for a long period of time the most common type of multi-user database system.

The development of microcomputers led to a new type of architecture, called the *client-server system*, which is based on a number of computers, connected in a LAN or a WAN. In the beginning the microcomputers were used to process application programs and were called *clients* as opposed to the mainframe system, which processed the database and was called *server*.

The architecture may contain multiple servers but each server has to process different applications in order to respect the client-server architecture definition; otherwise we are referring to a distributed database system. Main advantages of this also called “two tier architecture” are the improved scalability and flexibility. Its simplicity led to frequent employment in non-complex, non-time critical information processing systems.

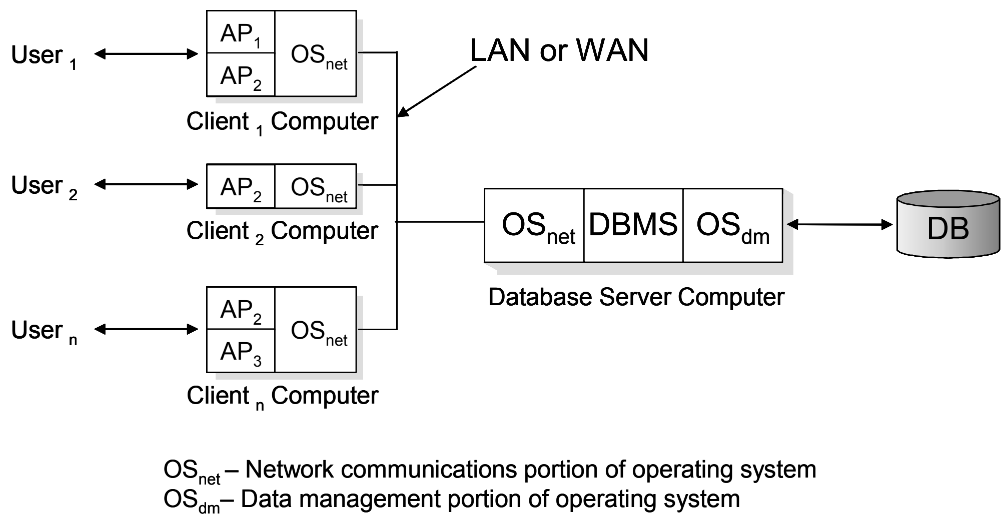
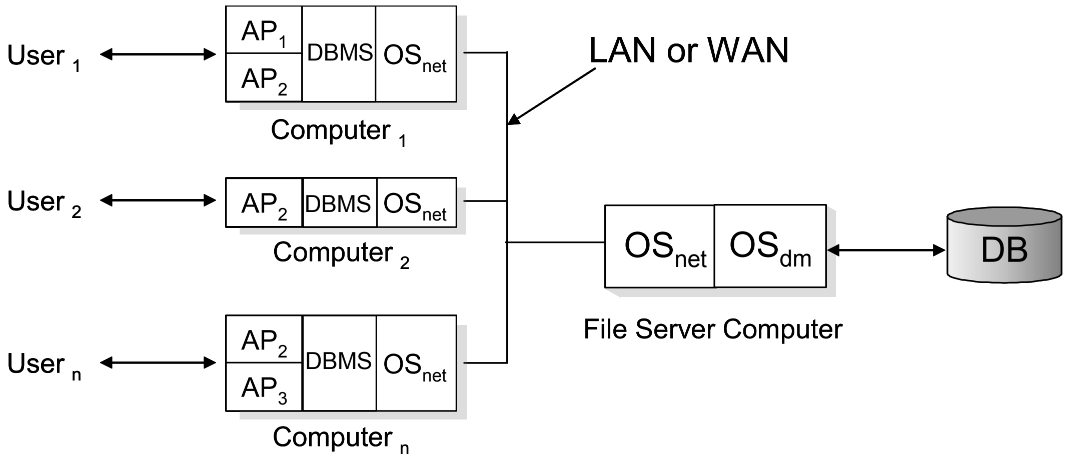


Figure 3. Client-Server Architecture

Another system architecture called *file-sharing* is obtained by distributing both the applications and the DBMS on the clients, therefore the server is no longer a Database Server but it is a File server. The entire process is very resource intensive since all database operations are executed on the client after receiving all the necessary files from the server; therefore this technology is seldom used when high volume processing is required but it is a good option when used to share expensive peripheral equipment or for non-database applications that need to store large files on large and fast disks.



OS_{net} – Network communications portion of operating system
OS_{dm}– Data management portion of operating system

Figure 4. File-Sharing Architecture

A major step forward in the growth of information sharing within the Internet was the development of the *hypertext transfer protocol (HTTP)* which is a TCP/IP-based protocol that enables the sharing of documents over TCP/IP networks. There are two characteristics that make this protocol noteworthy for database applications. First, HTTP is *request-oriented* meaning that HTTP servers wait for requests to arrive; when they do, the HTTP server takes some action and then, possibly, generates a response. Unlike other Management Information Systems (MIS), they neither poll nor solicit other application activity. Second, HTTP applications are *stateless*, signifying that HTTP server receives a request, processes it and then forgets about the client who made the request. [Ref. 6:p. 408] As a result, Web applications that provide static content are not affected but this turns to be a problem for database applications which process transactions between client and server; therefore, a main concern was to define a proper mechanism that had to maintain state during the time the transactions were processed.

The development of the World Wide Web (WWW) posed the problem of defining a universal markup language to ensure the display of information and the linkage between different pieces of information. The idea was that any Hyper Text Markup Language document (or *web page*) would be presentable in any application that was capable of understanding HTML (commonly known as *web browsers*). An important evolutionary trend came with languages like Java and JavaScript which were designed to embed programs into HTML documents. Java applets, which are designed to be downloaded from the web and run directly by the Java virtual machine within a browser, are also increasingly being included in web pages to provide more sophisticated animation and other desirable features.

Although HTML is very successful, its limited scope, the fact that it was designed to properly display documents in a browser, made it unsuitable for more complex database applications. Its main disadvantages are that it does not provide means to process data content and it is not extensible, meaning that it cannot be modified to meet specific needs.

As a result, the World Wide Web Consortium (W3C) defined the design goals of a new markup language, XML, designed to provide an extensible standard for materializing documents on web pages. Furthermore, it became a standard for data interchange since it is useful for transmitting database views.

The evolution of the Internet as a public network was accompanied by the development of *intranets*, which are private, local or wide-area networks (LANs or WANs) that use TCP/IP, HTML and related browser technology on client computers and Web server technology on servers. Besides the fact that intranets are usually private networks they also permit fast transfers of data between computers. This is a main aspect that differentiates them from the Internet and it requires a different approach when dealing with database applications. It is understandable that Internet application design must take into account transmission requirements, which have to be tailored according to the available transmission bandwidth. In contrast, due to the existing capabilities, intranet applications may allow the transfer of large amounts of data from the server to the clients; therefore most of the processing may be hosted by client machines.

Based on the capabilities of the Internet environment several architectures have been developed to optimize the database applications throughput. The three-tier architecture emerged to overcome the limitations of the two tier architecture. In the three-tier architecture, a middle tier was added between the user system interface client environment and the database management server environment. There are a variety of ways of implementing this middle tier, such as transaction processing monitors, message servers, or application servers. The structural concept of multiple tiers facilitates data exchange between different types of servers that may run different operating systems. Typical three-tier architecture may include: the database server, the Web server and the browser, which is installed on the client computer.

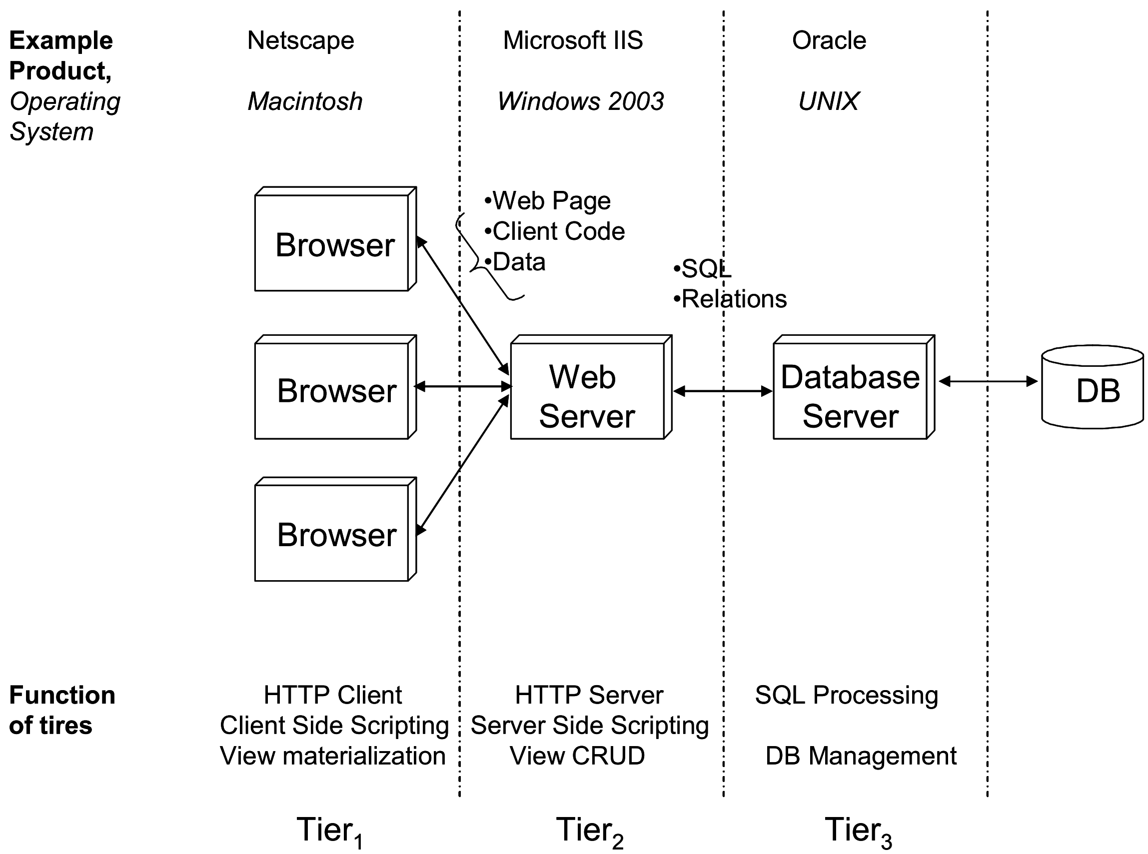


Figure 5. Three-Tier Architecture

In either architecture the data "traffic" is highest between database and database server, which means that the network infrastructure that connects them needs to ensure high bandwidth, which makes it expensive. The advantages of a multi-tier architecture is: It forces the separation between the user interface and the business logic and it requires low bandwidth network between them, making it suitable for Internet use.

Based on the operating systems running on the Web servers major software developers have created their own web server environments. Microsoft has designed its own set of standards and languages when using Windows 2000 Web Server Environment. IIS as part of the Windows Operating System is the HTTP server, which provides a special program interface (ISAPI) used by other programs (like ASP) to trap and process HTTP messages.

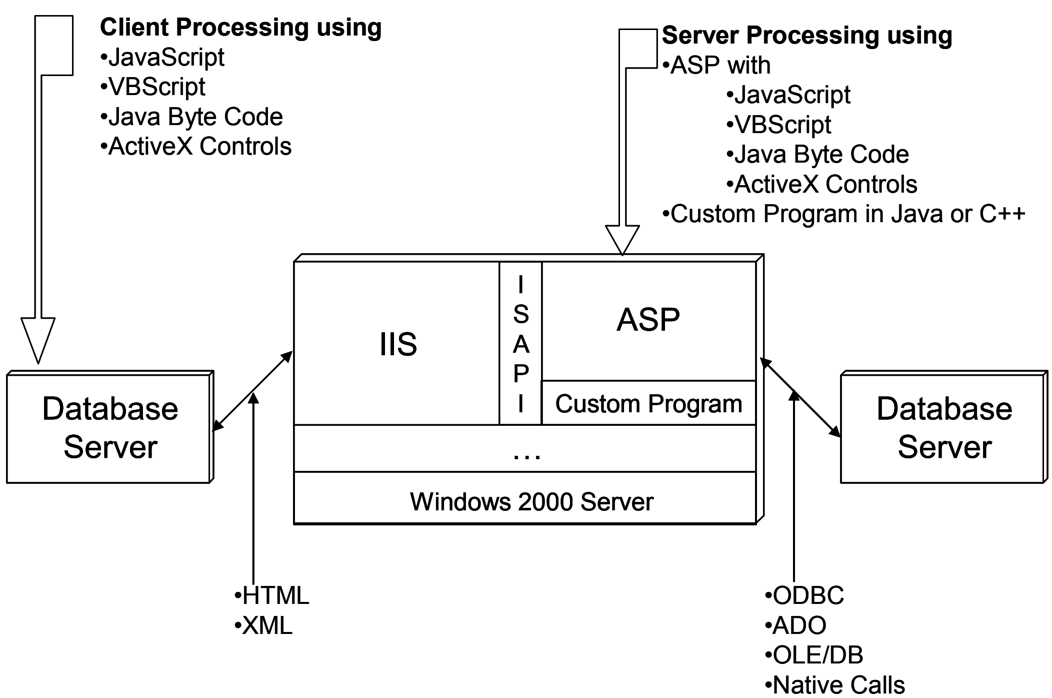


Figure 6. Standards and Languages Common with Microsoft Web Server

The process is initiated when the IIS receives an ".asp Web page" which is sent to the ASP who processes the page and sends the required response to the client via the IIS. Since the ASP hosts scripting languages, all the statements contained along with the embedded ActiveX controls will be executed when ASP processes the page. Besides ASP, other programs may be developed using C++ and Java in order to process HTTP messages.

The second major implementation is the Linux Web Server environment which uses instead of IIS, either Apache or Netscape Server. Three different types of interfaces (CGI, ISAPI/NSAPI and Apache Tomcat) are available to programs written in different languages to trap HTTP messages. Depending on the language used to write the Server processing program the data access standard used is either JDBC when accessing the databases from Java programs or ODBC.

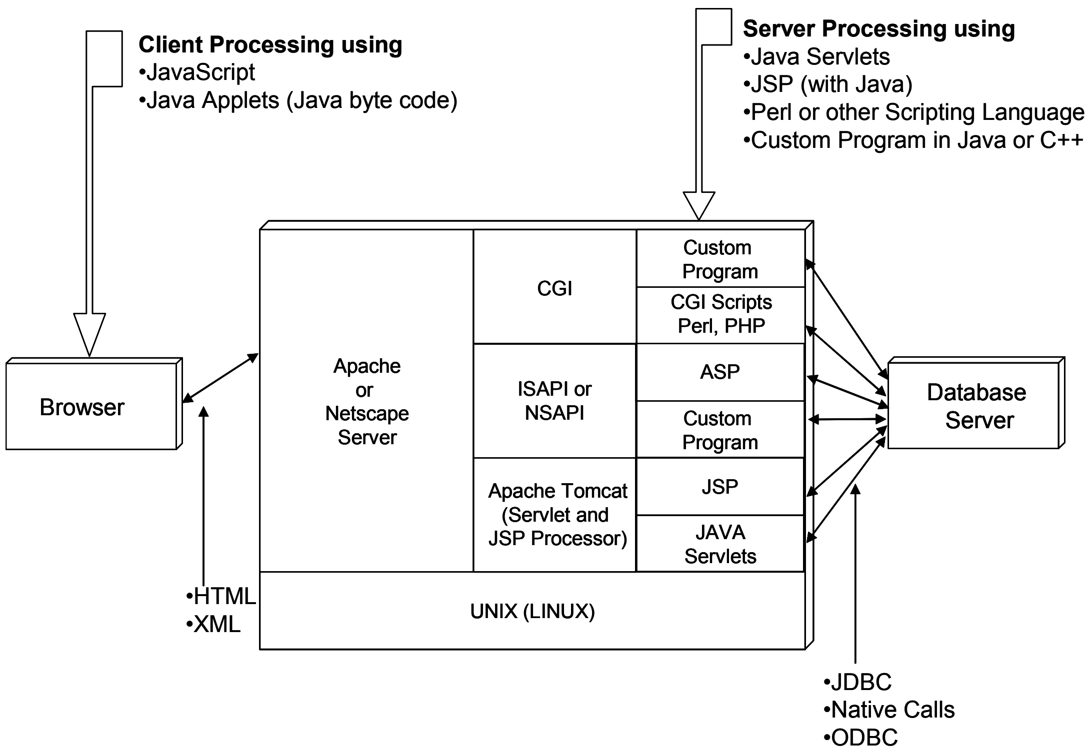


Figure 7. Standards and Languages Common with UNIX Server

As it has been mentioned before due to the stateless property of the HTTP protocol, the two presented architectures have included the capability to track the state of a initiated session.

The development of wireless technology claimed the need for an application protocol to make possible the use of Internet technology on wireless devices. The development of the *Wireless Application Protocol* (WAP) was followed by the creation of the Wireless Markup Language (WML), a variant of

XML, which offered the means to display Web pages on wireless devices. XSLT offered the means to convert XML documents into WML documents and vice versa. Other faced challenges dealt with the need for special *Micro Browsers* needed to address the problem of dividing the entire web page into small parts (called *cards*) properly tailored to fit to the screen of wireless devices. A new architecture was designed to allow access to database services through the Internet using wireless devices.

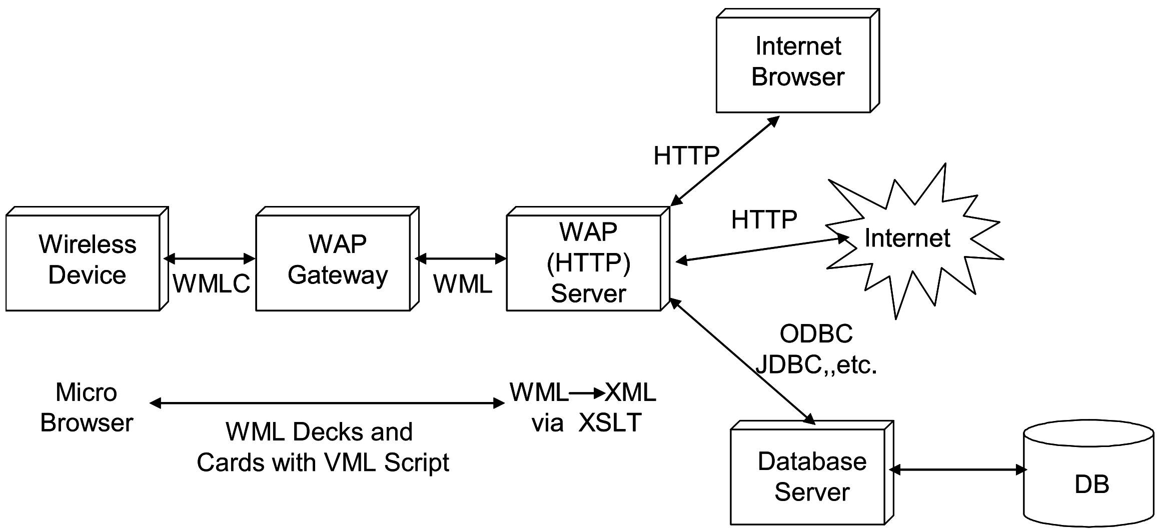


Figure 8. Use of Wireless Application Protocol [From: Ref. 8:p. 432]

Summarizing former introduced architectures, they all share a common philosophy where application processing is dispersed between different systems keeping the database hosted in a single location (on a single computer). Another approach is the so-called *distributed database architecture* where the entire database is spread among the interconnected computers.

Analyzing the principles used in partitioning the database there are *vertical fragment* implementations where the original database table is broken in two or more sets of columns, and *horizontal fragment* implementations, which deal with groups of rows from the original table. Different implementations may be obtained not only by distributing the original database on different computers but by duplicating it or parts of it on different computers too. As a result, there are advantages and drawbacks that come with each possible alternative design and here is how they may be presented in a comprehensive manner.

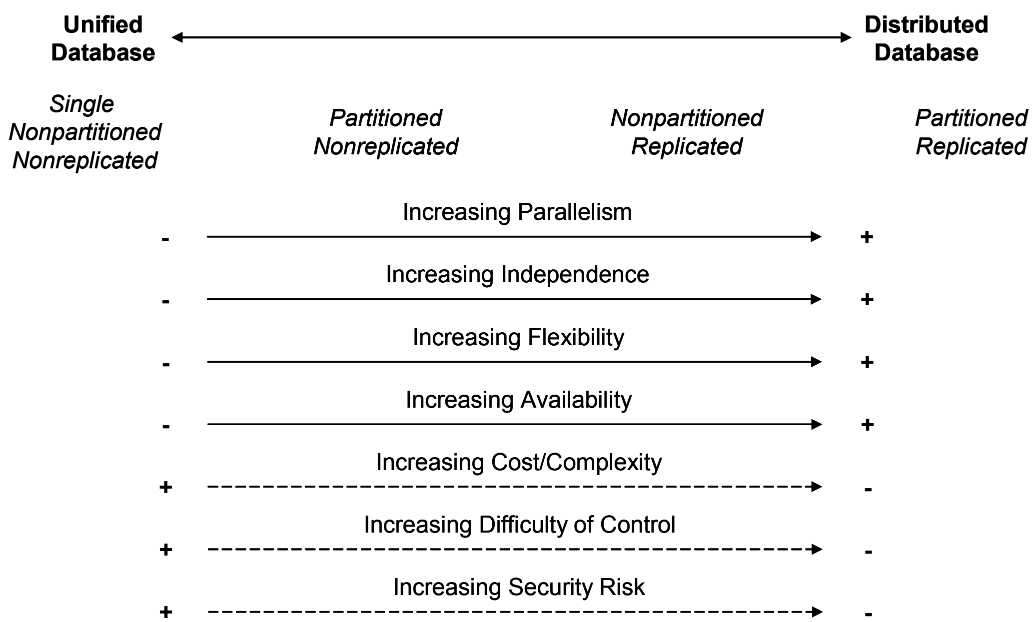


Figure 9. Database Distribution Alternatives - Advantages and Drawbacks [From: Ref. 8:p. 521]

As a result of the continuous development of personal computers, large quantities of data may be downloaded from the database servers and used locally for queries and reports. The process doesn't provide for simple capabilities to update the downloaded data since this process might generate problems, which have to do with the coordination, consistency and access control. An easier method to make downloaded data available to the users and to reduce the traffic within the network is to consider the use of a web server. The main idea is to publish the downloaded info from the database server on the web server where it can be accessed by the browsers. This way information is easier kept up-to-date since the database server has "a single client", the web server.

C. COLLABORATIVE TOOLS

Information sharing across a network also includes *Collaborative Computing*, an idiom which covers two main aspects of human collaboration: the *asynchronous aspect* represented by e-mail, discussion groups, news servers and similar software products which provide the basis for the so-called *“groupware”,* and the *real-time aspect* where interaction between people and specialized hardware and software is involved. Based on handling data and representing information, the person-to-person communication is made possible by the ability to share, modify and collaboratively create data and information.

The two existing technological methodologies for developing collaborative software are based on *the client-server* and *the peer-to-peer* models.

Client/server applications enable communication between clients but only after they connect to the server, which acts as middleman, keeping the master copy of all the information, running nearly all the application logic, and downloading the results to the client.

Peer-to-peer applications demand almost all the application logic and information reside on the client, which communicates directly with other clients without server intermediation. Peer-to-peer is also called *decentralized computing*, which doesn’t entirely exclude the use of the servers.

Based on its capabilities “collaborative software” can be segmented into two major categories: perceptual collaborative tools, collaborative workspaces.

Perceptual collaborative tools imply the existence of a session leader or initiator, all the other participants sharing this person’s view, that meaning only one participant at a time can perform collaborative actions. As a result, these tools require high bandwidth and are implemented using the client-server model. Collaborative interactions are limited to what can be viewed at one time. In addition, anything one participant can see or modify all participants can see or modify which leads to difficulties in supporting data control and basic security. Examples of currently available perceptual collaborative tools are NetMeeting and WebEx.

NetMeeting permits both audio and video communication; it allows collaboration in Windows-based programs, graphics exchange on an electronic whiteboard, files transfer, or the use of a text-based chat program. Common uses of NetMeeting include real-time document collaboration, technical support in a Helpdesk environment, training and distance learning, and conducting remote meetings." It is based on the Real Time Protocol (RTP) which works over User Datagram Protocol (UDP) and H.323 for audio and video transmissions and call control. The Light weight Directory Access Protocol (LDAP) is used by the Internet Locator Servers (ILS) which in turn are used by the clients to locate and communicate other clients - a type of directory access system.

Collaborative workspaces, which are also called affinity communities, provide asynchronous collaboration and team management utilities to a synchronous group of participants, allowing multiple people to work on a single file and communicate with each other, but not at the same time. Extensive file sharing and document management capabilities are essential to collaborative workspaces, because shared simultaneous access to the same file is not possible. These workspaces provide low-level connectivity functions that enable the transport of raw data. eRoom and Groove are two of the currently available collaborative workspace products.

D. GROOVE SYSTEM

There are certain reasons that have influenced the decision to choose Groove Workspace as the collaborative tool in former NPS collaborative experiments:

- Groove has an outstanding ability to work securely both within and across organizational boundaries
- It can be used whether online or offline;
- Its has a decentralized architecture that enables groups to form quickly, to add/remove members, to tailor their workspace to needs, and to easily disestablish the workspace when it is no longer needed, all without requiring an IT administrator.

The required Operating Environment includes:

- Operating System: MS Windows XP, 2000, NT 4.0, 98, ME
- Connection to the INTERNET: Min 58K dial-up (recommended DSL, ISDN or LAN)
- Microphone & Speakers for the audio features

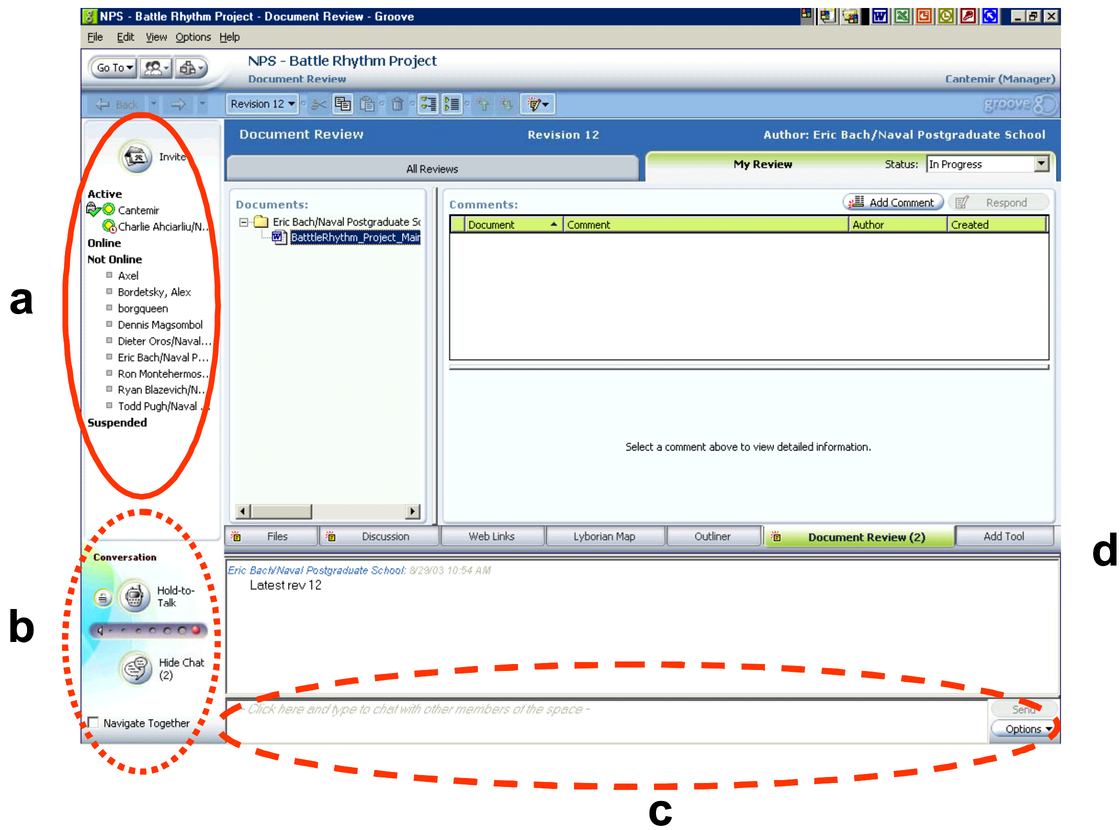


Figure 10. Basic Layout of Groove Workspace 2.5 (Communication Features and Tools Area)

The description of Groove features and functions is presented as found on the basic layout of Groove Workspace 2.5 [Ref. 9:pp. 1-5]

The Extended Presence Detection Section (**a**) shows information on:

- The active collaborative team members who are in the current workspace, and which tools they are currently using
- Who is online, but not in the current workspace
- How long they have been inactive

The Audio Conversations/Voice Messaging Section (**b**) allows half-duplex or full duplex connections.

The Public Chat Section (**c**) displays the Instant Messages to all workspace members.

Tools Area (**d**) may be customized for the needs of each shared space.

There are a large number of tools that are available either from Groove or from third-party vendors (Discussion, Calendar, Meetings, SketchPad, Contact Manager, Forms). In order to satisfy specific needs, software developers can create customized tools. Tools of specific interest for document collaboration include:

- Files Tool. Store, organize, and share any type of computer file. Microsoft Office’s Word files can be co-edited by two or more shared-space members; PowerPoint files can be shown in presentation mode with multiple members viewing.
- Document Review Tool. Post documents for review by multiple members, review and merge changes, publish a new revision, and start the process again if desired. By using threaded discussions comments and replies may be captured. This is an advantage over Microsoft Office’s features because the product adds the ability to organize, manage, and track progress throughout the review process.

Offline Use. This ability differentiates Groove from other server-based virtual workspaces. After working offline all updates and additions are automatically synchronized with those of other workspace members when connecting again

Encryption and Security. Groove data is encrypted and compressed both when it is stored on a hard drive and while transmitted over the Internet. Whenever possible, data is transmitted directly to the other people in the same shared space. A copy of this data is stored on each user's computer. At times, Groove does need to store data on a server. For example, if data is transmitted to someone who is not online, that data is stored temporarily on a server but then relayed to the person's computer when it comes online. In this case, that data is encrypted so that it is unreadable on the server.

Integration with Other Products. A growing number of Groove Workspace features are integrated with Microsoft products, including Word, PowerPoint, Outlook, Project, and SharePoint. This integration allows the user to shift more easily between the work environments currently used and Groove's collaborative environment.

Specifically, the user can:

- Collaboratively edit Microsoft Word documents
- Collaboratively view Microsoft PowerPoint presentations
- Send a copy of an email message or calendar entry from Outlook to a Groove shared space

Architecture. Groove's distributed peer-based architecture may be regarded as hybrid peer-to-peer and client/server architecture. Groove Workspace is the client piece installed on users' machines, encompassing both the client software (the transceiver) as well as user data sets (the shared spaces). Groove transceivers can communicate directly with each other in peer-to-peer mode. The reason to use a relay server resides in the need to enable initial contact between two users, to provide presence information, firewall navigation services and synchronization of offline users.

Figure 11 shows details for one Groove USER C, who is a member of two different shared spaces (X and Y). Membership for these shared spaces could virtually include any combination of clients shown on the diagram who can be either directly connected to the Internet or behind a firewall. They can work from more than one location, online/offline all workspaces being automatically synchronized when users reconnect.

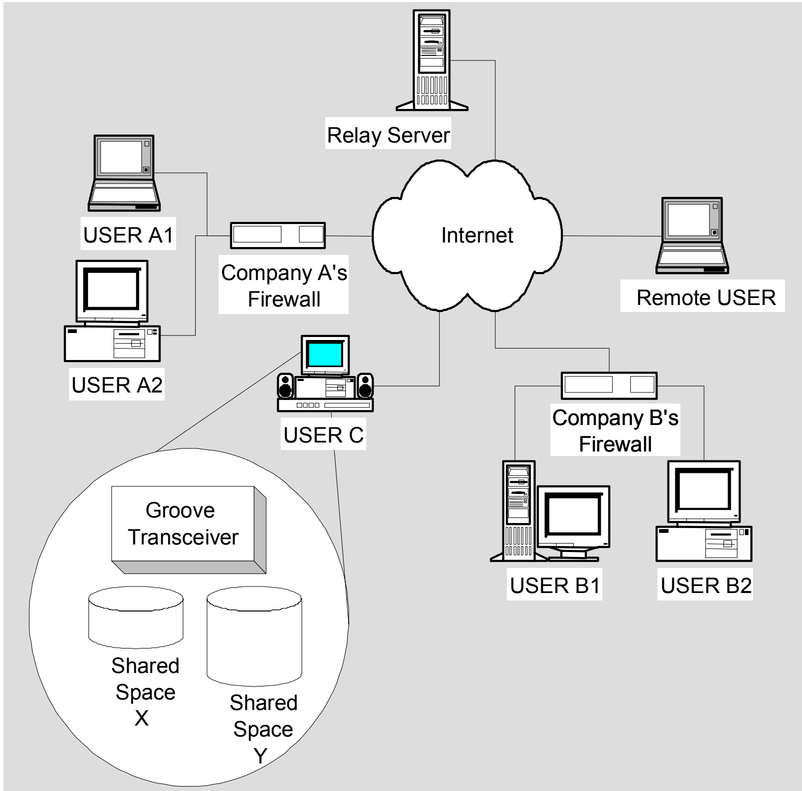


Figure 11. Groove's Peer-Based Architecture As a Blend of Peer-to-Peer and Client/Server Aspects [From: Ref. 9:p. 4]

For a better understanding of its features, examine Groove's peer-to-peer and client server aspects.

Peer-to-Peer Aspects. Each copy of Groove Workspace maintains encrypted local copies of its shared spaces on the local PC's hard drive. It keeps these shared spaces synchronized with other members' copies by transferring and receiving small packets of changes, or deltas. For example, when one member edits a file, those changes (not the entire file) are transferred to other members. Groove creates deltas for all user activities, including presence detection and instant messages.

Groove uses its proprietary Simple Symmetric Transfer Protocol (SSTP) to compress and encrypt all deltas for efficient and secure storage and transport over the wire across port 2492. SSTP also provides the routing mechanism that allows deltas to traverse network firewalls and Network Address Translation (NAT) routers.

Client/Server Aspects. While two Workspace clients can communicate with each other directly in peer-to-peer mode, Groove uses relay servers extensively to pass information in certain cases. The resulting impact of these situations is that workspace information is at least temporarily stored on Groove's relay server.

- If even one workspace member is offline when another makes changes, Groove stores all deltas on the relay server until the offline user reconnects. This feature allows the formerly offline user to receive all changes regardless of the online status of the other members.
- If a user suffers from a slow connection, then Groove Workspace will publish a delta to a relay server instead of that user sending deltas directly to all other members. The server will then fan out the single delta message into multiple messages, one for each workspace member.
- If Groove's SSTP is blocked by a firewall, Hypertext Transfer Protocol (HTTP) tunneling is employed. Groove sends the compressed, encrypted deltas to the relay server where they are wrapped in HTTP and routed via port 80 on the destination machine.
- That is why a given Groove shared space, with many members behind tightly managed firewalls would depend on the relay server for 100% of its communication needs. It is also important to remember that the data typically stored on these servers consists of compressed, encrypted deltas, rather than entire shared spaces.

Further on, in Chapter IV, a subchapter will include a presentation of former NPS experiments and projects that have used Collaborative Software (GROOVE) and agent technologies within in-house developed software applications.

IV. MULTIAGENT ARCHITECTURES

A. BACKGROUND

Chapter III analyzed the evolution of information sharing within a network relating it to important moments in the development of the Internet. Based on its evolution characteristics there are a number of trends which affect data sharing over the Internet [Ref. 10:p. 2]:

- *Bandwidth Utilization* – Despite the continuous development of the backbone of the Internet, the available bandwidth for many users will continue to limit their processing capabilities. There will be an increasing gap between the quality of the backbone and the drawbacks caused by the limitations of the end-user available technologies.
- *Mobile Devices* may be considered the hottest area of development in the computer industry. It is presumed to have unreliable, high-latency telephone or wireless network connections.
- *Mobile Users* will look for full access to all of their files and applications from any terminal. Despite the foreseen proliferation of the mobile devices, Web terminals will not be available everywhere the users might find themselves.
- *Information Overload* – the continuously increasing volume of available information requires adapted methods to ensure customized filtering. The role of proxy sites will increase due to their capabilities of decreasing information overload and customizing service access.

Then again, the World Wide Web (WWW) has already made interaction between people and web pages possible. The next wave of interaction promised by the web is between software applications operating without the need for human intervention. These web-based applications could be federated databases, intelligent agents, grid services, Web services and other kinds of distributed applications.

The purpose of this chapter is to introduce the basics on Intelligent Agents and to focus on the vision of the Control of Agent-Based Systems (CoABS) developed by DARPA. The implementation of the Sensor - Decision Maker Grid is an important facilitator of the functionality of this concept and it will be explained in connection with the development of the CoABS Grid concept.

A presentation of existing implementations described in the literature will enable a better understanding of the concepts and of the approaches used in different experiments based on the CoABS. A final focus is on those software products and in-house developed applications that have been used in former NPS experiments and projects as means of information sharing.

B. INTELLIGENT AGENTS

A rapidly evolving technology that provides answers to a great variety of computer related problems including collaboration aspects too, is the concept of “intelligent agents”. Diverse approaches have generated a lot of different definitions of the term “agent” which stress some important properties pending to the domain where the definition comes from. Two shared properties can be found analyzing a list of the current definitions of the intelligent software agents: *(1) an agent is one who acts or who can act,* and *(2) an agent is one who acts in place of another person with permission from that person.* [Ref. 11:p. 457]. To formally approach the problem, an agent should be defined as a software system, operating in a permanent *Perceive-Reason-Act* (PRA) cycle as represented in Figure 12. Thus, the agent receives some stimulus from the environment, perceives it, starts a reasoning process that combines the newly incorporated information with its former existing knowledge and goals and finally selects one of the possible actions to be executed. As a result of this action, the state of the environment changes which in turn generates new perceptions for the next cycle.

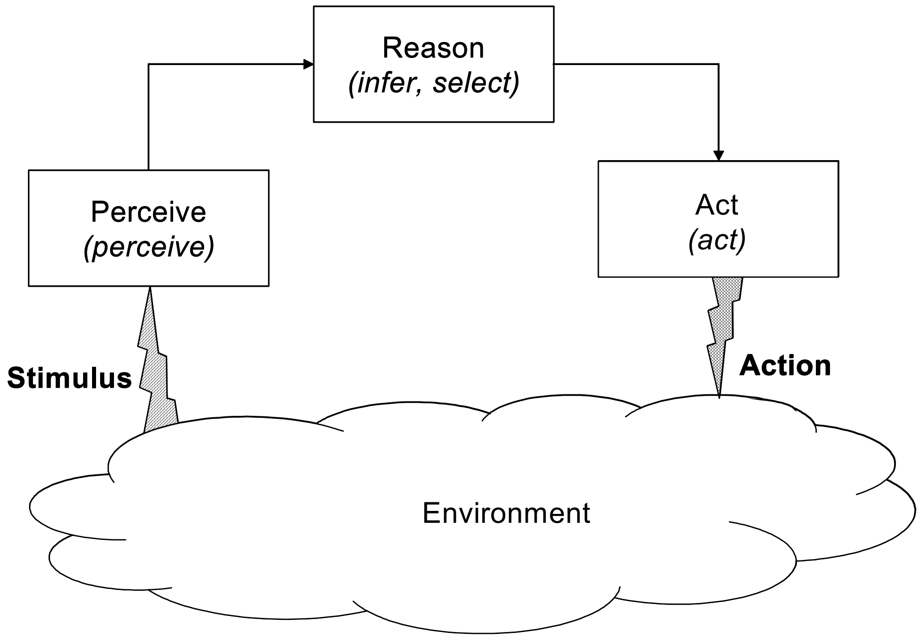


Figure 12. Perceive–Reason-Act Cycle [After: Ref. 12]

The PRA cycle may is concisely and formally described by the notation proposed by [Ref. 12]. A basic assumption is that the agent is located in some environment and (*S*) will represent the set of external states. The agent may be characterized by a database of acquired knowledge (*D*), a set of possible perceptions of the environment (*T*) and a set of possible actions (*A*) in conjunction with four functions: *perceive, infer, select,* and *act* which may be described as follows:

perceive : $S \rightarrow T$ is the function which determines how the agent actually perceives the state of the environment and is limiting the amount of provided information to a partial view of the entire state.

infer : $D \times T \rightarrow D$ is the function used by the agent to update its internal knowledge base using the new perceptions.

select : $D \times T \rightarrow A$ is the function used to determine the best action for the current cycle.

act : $A \times S \rightarrow S$ is the function used to change the state of the environment consequently.

Figure 13 represents the 7-tuple *{ D, T, A, perceive, infer, select, act }* which defines an agent and the information flow between these elements.

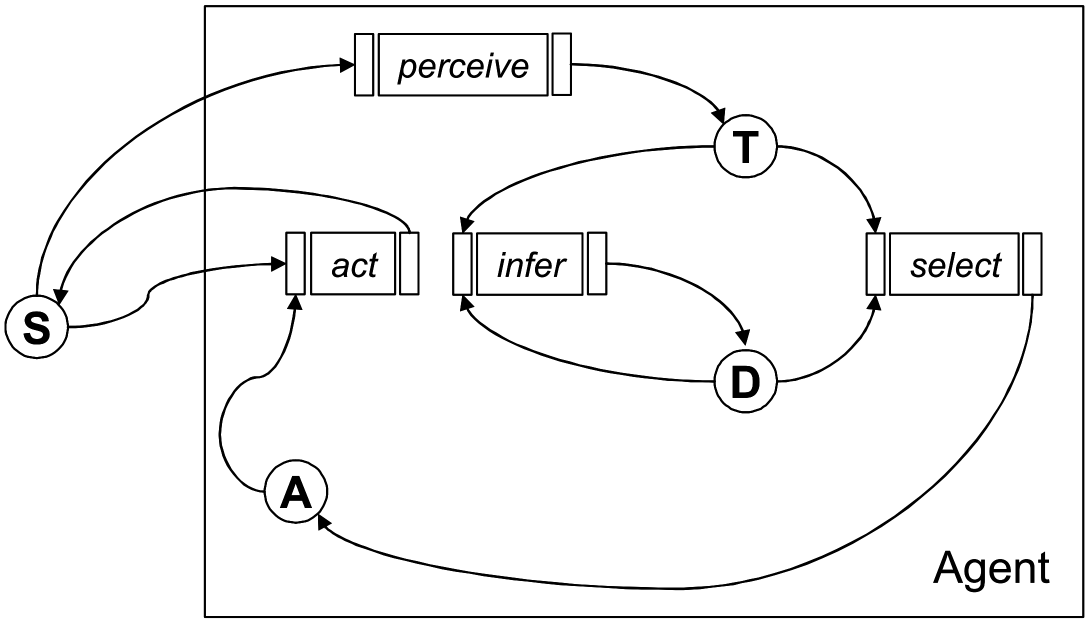


Figure 13. The 7-Tuple Defined Agent.

As described before the agents' functionality cannot be understood ignoring the environment to which they belong or the nature of their interaction. To classify the environment it is necessary to know whether it is: accessible or inaccessible, deterministic or non-deterministic, static or dynamic, discrete or continuous. [Ref. 13:p. 18].

Alternatively, it is important to depict those capabilities that agents should enclose in order to differentiate them from other computer-based programs. Depending on the author who covered the topic, there are different characteristics and classifications, which may not overlap due to the chosen criteria or the chosen level of granularity. A distilled list of characteristics may be found in [Ref. 11:pp. 456 - 461]:

- *Autonomy* understood as the capability of an agent to decide himself when and under which conditions it will perform what actions;
- *Cooperation* defined as the ability of communicating, working together and negotiating with other agents. The combination of autonomy and cooperation leads to a class of agents called "*collaborative agents*" which seek to achieve common goals through cooperation.
- *Mobility* is the quality of the so-called "mobile agents" which allows them to migrate from host to host in a network, at times and to places of their own choosing. By saving the state of the running program, it can be transported to a new host where it is restored and the program starts from where it stopped.
- *Personalizability* defined as the capacity of a learning agent to "self-educate" by monitoring its user's actions and consequently reacting properly.
- Alternatively a more general set of capabilities to define an agent's "intelligence" are introduced in [Ref. 13:p. 18]:
- *Reactivity* – the quality of an agent to properly perceive and react to environmental changes
- *Proactiveness* – the ability of an agent to *take the initiative* with the purpose of satisfying its design objectives
- *Social ability* – relates to the capacity of an agent to interact with other agents or humans in order to accomplish its mission.

To summarize, an intelligent agent is one which is able to make rational decisions, i.e., blending proactiveness and reactiveness, showing rational commitment to decisions made, and exhibiting flexibility in the face of an uncertain and changing environment. Here is where the main differences between agents and expert systems need to be defined: an expert system needs the inputs coming from a user which acts as the "man in the middle" between the environment and the system; it cannot provide proactiveness and reactiveness and it is not capable of cooperation, coordination, and negotiation.

Intelligent Agents represent a powerful Artificial Intelligence technology whose advantage comes from its resemblance to our natural view of the world and from the belief that agent–oriented programming is often considered as a natural successor of object-oriented-programming. [Ref. 14:p. 35]

There are certain well-established approaches used in building AI systems such as: deductive reasoning agents, practical reasoning agents and reactive and hybrid agents. As inferred by their name deductive reasoning agents prove intelligent behavior by working with symbolic representations and being capable of logical deductions. Practical reasoning agents are designed to be able to decide what to do and are usually based on the BDI model introduced by M.E. Bratman more than 15 years ago and which basically sustains many already implemented systems (PRS, JAM, JACK). This cognitive model defines the mental attitudes of Belief, Desire, and Intention as its central concepts; thus an agent is described by its *Beliefs* that determine the agents' current world knowledge, its *Desires* that determine the goals of the agent and finally, the *Intentions* that are generated from reasoning about the current beliefs and goals and thereby determine the best possible actions. The first agent architecture to explicitly implement this model was the *Procedural Reasoning System (PRS)* – presented in Figure 14, developed at Stanford Research Institute and applied in many successful multi-agent implementations, including the OASIS air–traffic control system.

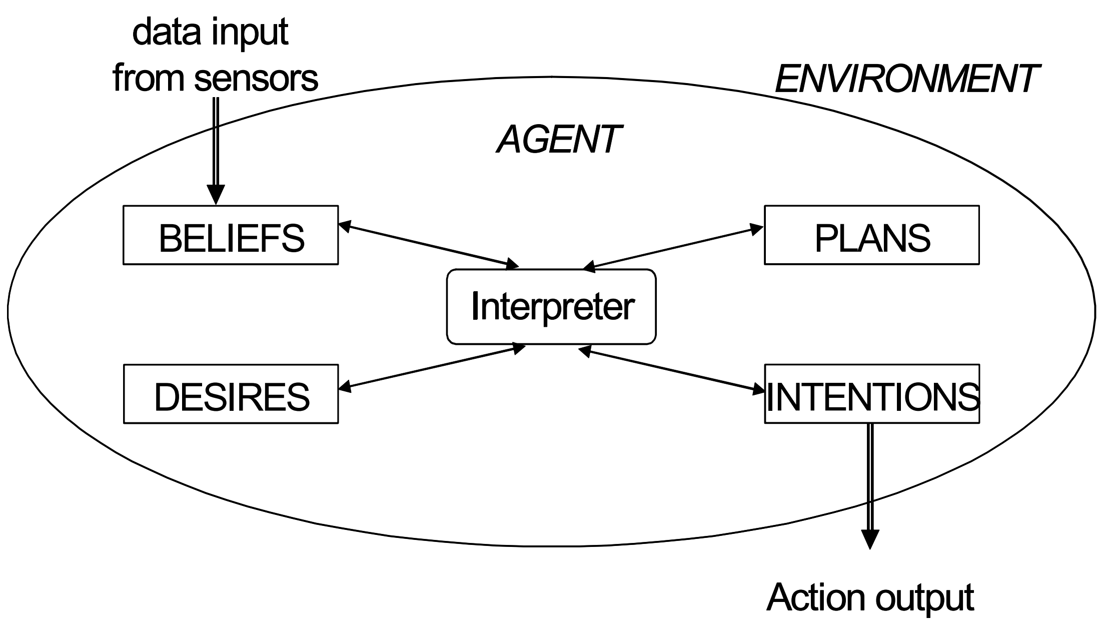


Figure 14. Procedural Reasoning System (PRS) [From: Ref. 13:p. 81]

As mentioned before, intelligent agents are both proactive and reactive. Being proactive means that the agent follows its agenda over time, it implies the use of desires, which may modify the agent's internal execution cycle. Instead of choosing intentions (actions) the chosen desires (goals) are persistent and restrict the selection of intentions.

Conversely, a reactive agent is one which will change its behavior in response to changes in the data received from the environment so an important aspect in decision making is balancing proactive and reactive aspects. On one hand the agent should follow its desires by default; on the other hand it should consider changes in the environment. The key is to properly define a threshold that identifies when the input data coming from the environment sensors is considerably changed turning the perceptions into a significant event. An event may trigger new desires, may generate changes in the information about the environment and may lead to immediate action. Since in real situations agents are provided limited computational power and limited ability to sense the environment, beliefs are important as repositories of perceived information from the environment and plans enclose the means to pursue the desires. Furthermore, a library of plans will eliminate the need to define each action's preconditions and effects.

By extension, we may analyze an entire system by decomposing it into subsystems characterized by either one of the two behaviors [Ref. 13:pp. 97-99]. As a result, there will be classes of architectures with different subsystems grouped into a hierarchy of interacting layers. There are two types of control flow within layered architectures: the horizontal layering, where each software layer acts like an agent because each one is directly connected to the data inputs and outputs; and vertical layering where inputs and outputs are performed by only one layer. More complex examples of layered architectures are the Touring Machines – a horizontally layered architecture and the InteRRap – a two-pass vertically layered architecture.

Previously defined concepts introduced a single-agent view, which has to be extended towards models capable to explain multi-agent systems functionality. Usual structures of multiagent systems contain a set of agents placed in a common environment, having spheres of influence, which sometimes overlap, and being able to establish relationships among them – like in Figure 15.

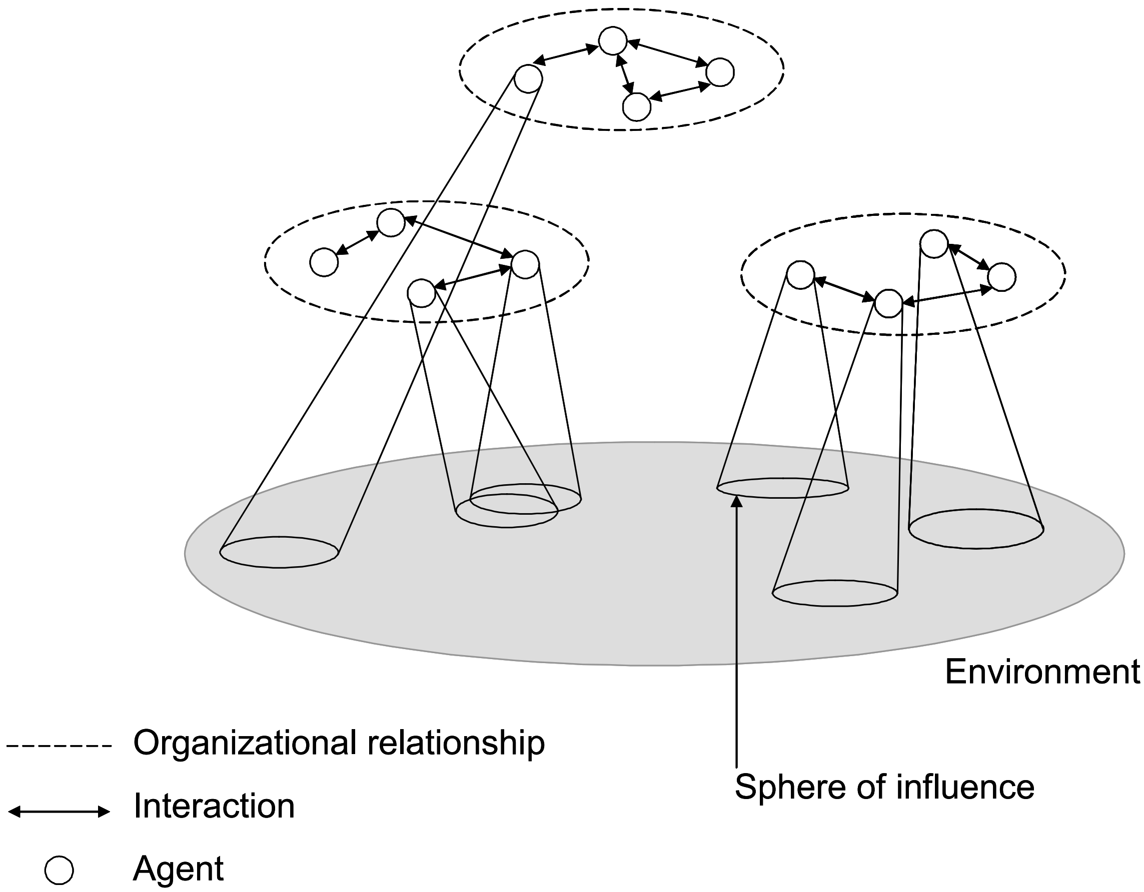


Figure 15. Procedural Reasoning System (PRS) [From: Ref. 15:p. 117]

A multiagent system (MAS) architecture needs to be structured which means to define every agent's role and the relationships between them; therefore the architecture is to reflect the capability of the system to deal with complex problems and to determine the operational mechanisms. In MAS, agents can undertake learning, perception, reasoning, judgment and decision in term of their knowledge about themselves and their environment. Moreover, agents can cooperate and coordinate by communication.

Based on the chosen criteria, a MAS architecture can be categorized as pipes-and-filter, event-based and layered (*in term of the mechanism*); or hierarchical, distributed, open, reconfigurable mobile and fault-tolerant (in *term of their characteristics*). [Ref. 16:pp. 94-95.] However, these categories do not completely describe the common characteristics and the essence of MAS. In a general view multi-agent system architectures can be classified as information-flow oriented, role-oriented and control-oriented architectures. Role–oriented architectures of MAS are adapted to meet the specific requirements of the functional decomposed application domains and they are the overwhelming majority of MAS based on the studied literature. Finally, hierarchical architecture is a typical control-oriented architecture of MAS.

The hierarchical solution to a global problem is built up of modules which form stable sub solutions, and allow one to construct a complex system out of less complex components [Ref. 17: pp. 3-13]. Classical hierarchical architecture lies in the fact that complex tasks can be decomposed into several levels of abstraction and assigned to distinct hierarchical levels. The specialization of each level facilitates the implementation and management tasks. Hierarchy is an essential paradigm to tackle complexity. Hierarchy is a proper equivalent of human organization/management procedures to solve complex problems and provides effective feedbacks and control capabilities. In contrast, there is a major disadvantage of the hierarchical architectures that has to do with their rigidity and their impossibility to adapt to unexpected disturbances.

Former presented architectures form the theoretical foundation for the developed agent technologies, which greatly impact the development of computer science today. Agent technologies have spread within different application domains and are expected to play a crucial role in the development of: *Ambient Intelligence, Grid Computing, Electronic Business, Semantic Web, Bioinformatics and Computational Biology.* Some of the new developed technologies try to efficiently take advantage of the distributed and relatively structured information resources found on the World Wide Web. The use of information agents is based on their capability to accesses, collect and manage different information sources following queries posed by human users and other agents. This approach is not as easy as it might seem due to the enormous quantity of available information and the immaturity of the existing search techniques, which are not able to depict only the valuable data. The *passive* approach implemented in search engines and crawlers can be improved by wrapping information sources with agent capabilities as presented in Figure 16. [Ref. 13:p. 254]

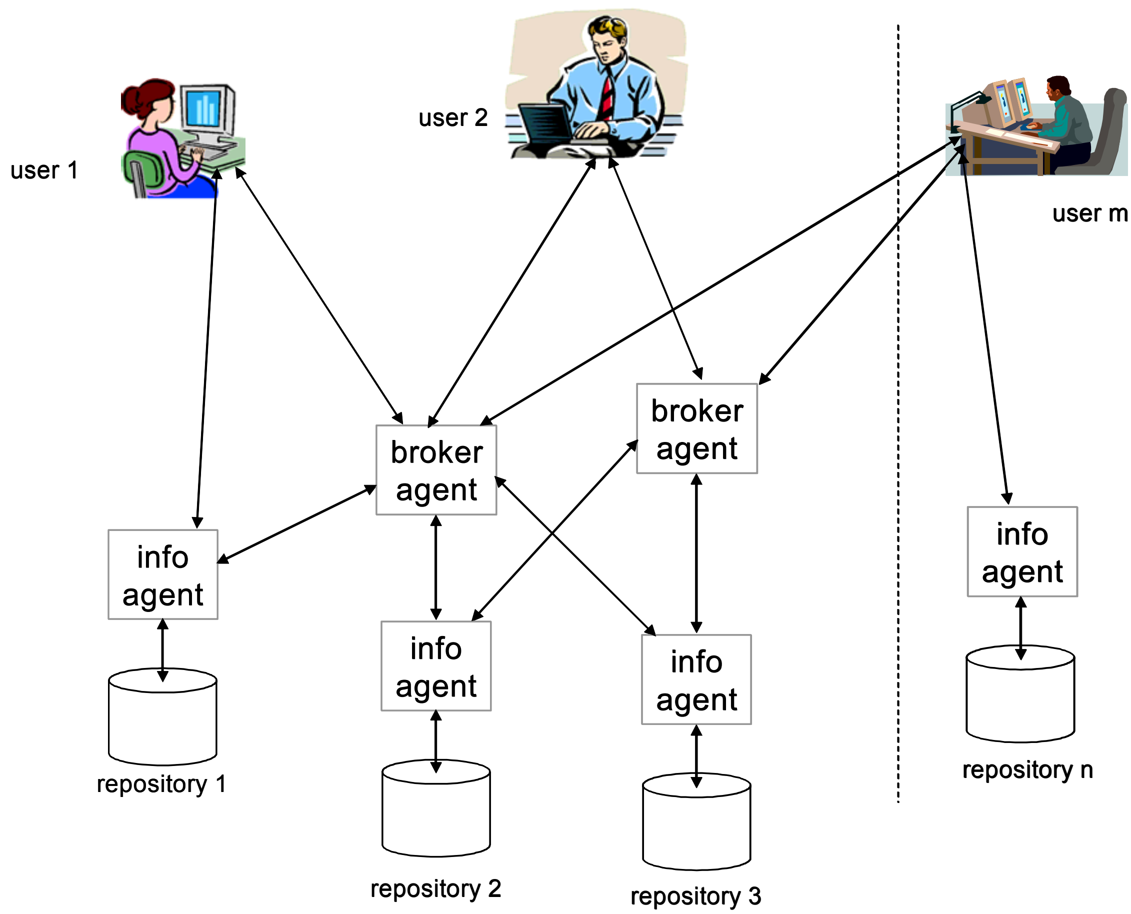


Figure 16. Multiagent Information System Architecture

Each repository is accessed by an information agent which serves as an expert capable to extract the necessary information based on the provided "meta-level" queries and to communicate with other agents advertising their capabilities. A middle structure can reside between users and information agents, the so-called *brokers*, to keep the track of the existing advertised capabilities. Broker agents may either match requests with advertised capabilities or simply collect and make available requests acting as a blackboard agent. The system in figure 15 includes agents that reside on users' PCs where they gather the issued queries and send them to information agents or to broker agents. If used, broker agents may contact and collect specific data from different information agents.

The relatively new Grid computing concept focuses on transparent resource sharing in a dynamic distributed environment, where sharing refers to direct access to remote computers and their associated resources, as opposed to conventional file sharing as seen in the Internet. Resources are shared by attaching them to a grid enabled network, allowing clients to communicate with resources without knowing their location. The model tries to solve the problems in the same manner as electrical energy is distributed through national power grids. [Ref. 18]. The grid middleware has to mask the heterogeneous nature of the resources and to provide:

- A secure execution environment that is scalable in terms of the number of users and resources
- Virtual organizations, formed through the federation of real resources. Resource owners will only contribute their resources to these federations if they are able to ensure access to their own local community. [Ref. 19];
- Information relating to the grid's resources, the application's performance and behavior and the requirements coming from the user and the resource provider;
- Effective resource exploitation by making use of information relating to the structure, behavior and performance of the application.

The great advantage of the Grid technology comes from the ability to obtain remarkable processing power without using expensive high performance computing methods such as massively parallel processors, supercomputers etc. A number of middleware platforms have been developed to support the development of Grid Computing applications. New trends started to be exploited in a Grid environment for instance: distributed object computing, component-based software development, resource discovery platforms, etc.

Middleware concepts like DARPA's CoABS Grid is based on Jini^{TM} Network Technology [Ref. 20] which was developed by Sun Microsystems that allows dynamic (I.e. services can be created and removed at any time) distributed environments to be created, which allow clients and services to communicate with each other whilst possessing only a minimal knowledge about each other. Any service wanting to participate in a Jini organization must find and join a Jini lookup service. For redundancy, more than one lookup service may be available; in this case the service may choose to join some or all of the available lookup services. A client wishing to make use of a service finds all lookup services and retrieves a list of available services. The client can then choose and make use of whichever service he wants to use. Details on how Jini Layer was used by DARPA's CoABS Grid along with details of implementation will be presented later this chapter.

C. DARPA'S CONTROL OF AGENT BASED SYSTEMS PROGRAM

The need to provide the right information to the right participants at the right time is the main concern of current military command and control systems. The challenge of being able to ensure interoperability between different information systems belonging to different technologies translates itself into providing a better information management, which is the key to obtaining the information superiority, needs as described in "Implementing Joint Visions 2010." It is a required quality for modern military to efficiently and rapidly interconnect unlike information systems providing a structure capable to operate as a whole. An important intricacy is caused by the need to join systems with different characteristics, designed to work in different environments implementing no predefined standard and belonging to different structures, organizations or even countries. It is the purpose of the Control of Agent Based Systems (CoABS) program to explore *the technical underpinnings of such run-time interoperability of heterogeneous systems, and develops new tools for facilitating rapid system integration in practice.* The research program is lead by the US Defense Advanced Research Projects Agency and the US Air Force Rome Labs and is targeted towards the use of agent technology to improve military command, control, communication and intelligence gathering. Agent-based technologies are intended to facilitate multi-system integration, to optimize the information traffic between them and thus prevent or reduce bandwidth saturation and improve the quality of service in the entire system.

In order to obtain a comprehensive and scalable approach to software agent interoperability, the CoABS program was focused on the accomplishment of three major tasks:

- The Agent Grid;
- Agent Interoperability Standards
- Scaling of Agent Control Strategies

A so-called *grid adapter* concept applies agent technology aiming to make simpler the upgrade of military legacy systems by focusing on the connection mechanisms instead of the client components. The middleware approach is sustained by defining new standards on agent-human interaction, agent-agent communication, agent software interfaces, and agent management and control. A powerful agent-based solution requires capabilities for monitoring, coordinating, controlling, and managing agent collections, ranging from simple tasks involving the cooperation of small agent teams to highly complex interactions involving thousands of individual agents. This task also provides guaranteed behaviors for agents, even in unreliable networks. Areas of interest include knowledge sharing techniques; team formation and coordination through modeling of plans, commitments, and intentions; and computational markets including protocols for auctions and voting. [Ref. 21]

Part of the overall CoABS Program, the CoABS Grid provides a method-based application programming interface used to register agents, advertise their capabilities, discover agents based on their capabilities and ensures communication between agents. Adding new available agents to the network or purging failed or unavailable ones is made without reconfiguring the network. As mentioned before, CoABS Grid was built using the Jini^{TM} Network Technology which was developed by Sun Microsystems. There are few features of Jini that have made this technological solution attractive [Ref. 22:p. 2]:

- The Jini concept of service, which is used to represent an agent;
- The use of Jini's discovery/ lookup/ join protocol to support dynamic registration and discovery of agents, along with processing resources and data resources;
- The use of Jini™ Entries, which are used to advertise an agent's capabilities;
- The use of Jini's event service to keep track of changes in available GRID resources;
- The use of Java for cross platform compatibility;

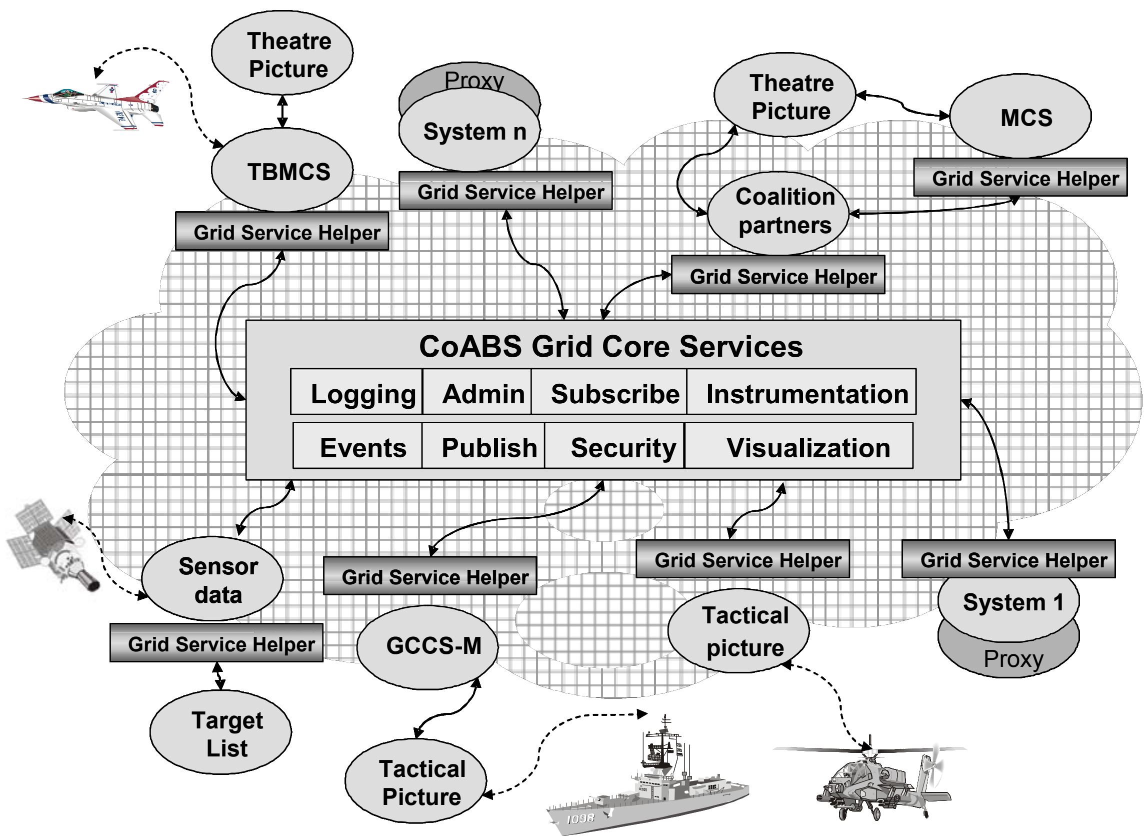


Figure 17. CoABS Grid Core Services [From: Ref. 23]

CoABS Grid's unique features include: it is transport neutral in terms of agent communication, it provides a general message delivery interface, which must be used by the employed proxy inside the grid but the agent may use any transport to communicate with its proxy. Communication within the CoABS Grid is fully distributed, meaning that each agent sending a message communicates directly with the receiver, using the proxy registered by the receiver. All the communication between agents is point-to-point its performance being influenced only by the agents' distribution within the network, the available bandwidth and by the proxy's implementation.

Due to its versatility, the CoABS vision started to be actively used for integrating applications across military and commercial projects developed by more than 35 commercial and Government organizations. A short list of the most important military implementations includes: *The Expeditionary Sensor Grid* developed by the Navy Warfare Development Command, *The Airborne Manned / Unmanned System Technology Program* conducted by the Army Aviation Applied Technology Directorate, *The Joint Battlespace Infosphere* developed by the Information Directorate of the Air Force Research Lab or the CoAX Experiment funded by DARPA and the UK Ministry of Defense.

D. THE COALITION AGENTS EXPERIMENT (COAX)

The Coalition Agents Experiment (CoAX) represents an international collaborative research effort carried out under the auspices of DARPA's CoABS program which focused on supporting the concept of *Network Enabled Capability (NEC)* within coalition operations. The main research hypothesis of the experiment was that the emerging technology of software agents and the Semantic Web could help to construct coherent command support systems for Coalition Operations.

Present military operations require a great amount of valuable information to be made available to different actors, which must be able to join their efforts into a coherent force. Coalition operations, which may be multi-national, require rapid integration of systems that have not been designed to work together but need to do it in an unfriendly or unknown environment and in a cohesive manner.

The CoAX Experiment joint the efforts of twenty–six formal partners from the UK, the US and Australia in a program that ran from February 2000 to October 2002.

The overall goal was to prove that an agent-enabled infrastructure could help the construction of a Coalition ‘command support system’ and improve its effectiveness. More specifically, as defined in the official documents [Ref. 24:pp. 26-35], the operational and technical objectives of CoAX were to:

- Show how flexible, timely interaction between different types of potentially incompatible systems and information 'objects' could be effectively mediated by agents, leading to agile command and control and improved interoperability;
- Show how ease of composition, dynamic reconfiguration and proactive co-ordination of Coalition entities leads to adaptive responses to unexpected events at 'run-time', providing robustness in the face of uncertainty;
- Show how loosely-coupled agent architectures, where behaviors and information are 'exposed' to the community, are more efficient and effective than monolithic programs;
- Show how agent policies and domain management help facilitate: selective sharing of information between Coalition partners, leading to coherent operations and control of appropriate agent behavior, leading to an assured and secure agent computing environment

Based on the agent computing model, other new emerging technologies were used to provide better data sharing within different distributed entities: the CoABS Grid infrastructure, the Semantic web, the Human-cyberspace Interface.

Agents are dispersed entities within the entire environment; they communicate and share information in digital format in an asynchronous manner their main task being to improve collaboration between different human structures, helping them to fulfill their missions. Figure 18 overlaps the representation of a network, a view of the employed agents along with their built-in functionality.

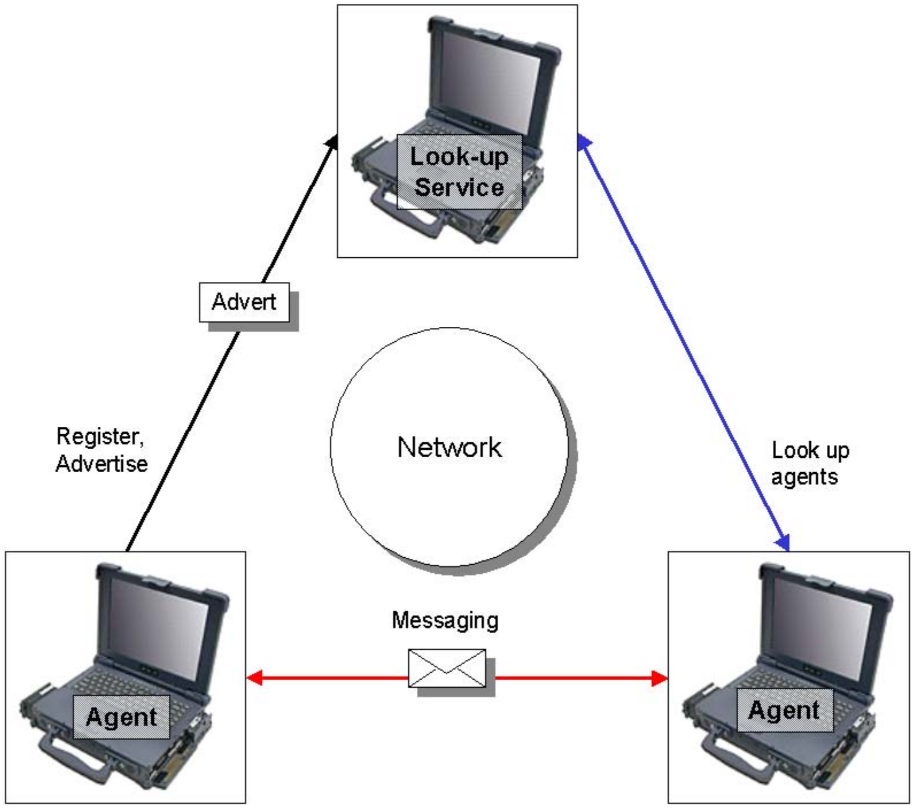


Figure 18. Computer Network, Agents and Implemented Services [From: Ref. 24:p. 133]

The CoABS Grid is the structure that ensured the capability for agent discovery and for transferring messages between the subscribed agents.

The unattended use of agents to perform tasks, which cannot be fulfilled by common software, may turn into a potential danger when malicious agents may attempt to harm the proper flow of actions. Therefore, a so-called Knowledgeable Agent Oriented System (KAOS) was developed to monitor that all the agents, working on different platforms always operated according to a predefined policy and that their action was subject to permanent human control. The policies were established according to the requirements imposed by the different military structures belonging to a hierarchical structure. Agents were organized into logical groups that corresponded to the existing human organizational structures, administrative groups and task-oriented teams.

The CoAX Experiment also tested methods to obtain the implementation of a new concept *The Semantic Web.* Commonly used Web technologies do not provide support for machine understanding, thus limiting the processing capabilities for huge volumes of web-hosted information. The idea is to find a convenient way to define and link web data in order to facilitate its usage not only for display purposes but for automation, integration, inference and reuse across various applications [Ref. 25]. The semantic web model requires the usage of various technologies, tools and methodologies for instance:

- Uniform Resource Identifiers (URI) are used to identify resources on the web,
- Extensible Mark-up Language (XML) provides a well defined and structured format for data exchange,
- Schemas and ontologies provide the means to describe the meaning of terms in a domain and they use The Resource Description Framework and the DARPA Agent Mark-up Language (DAML).

Within CoAX, XML was one of the languages to facilitate inter-agent messaging and DAML was used to encode and reason about domain entities, domain policies and agent message contents.

The CoAX experiment also provided some relevant conclusions about how agent technologies can support Network Enabled Capability and consequently enhancing military capacity by better usage of existing information. Due to their capabilities (their behavior is not fixed at “design time”) agents may enable military commanders to either behave unpredictably or to adapt to the unexpected changes in the military environment at runtime. A main difference between agents and traditional software is that their behavior is proactive and predictive as opposed to being reactive. Real battle space can be characterized as event-driven, high-tempo, uncertain, diverse and dynamically varying; therefore agents, which can adapt and react to different types of triggers, such as events or messages, are robust and perfectly suitable within such environments. Finally, software agents can work with humans in a so-called mixed initiative manner such that, as the humans click, type and speak, they are triggering agent actions. Agents can sense certain real-world events and report back to the humans, humans and agents working as a collaborating distributed team [24]. The activities are not determined at ‘design-time’ but are free flowing and natural. In this case, software agents are supporting the core themes of shared awareness and fully networked support.

As a final point, the CoAX experiment has shown that agent-based systems are capable to adapt to varying levels of resource availability, their access to resources can be controlled by pre-established policies and that the entire architecture of software agents and the CoABS Grid did not hamper processing and communication [Ref. 24:pp. 26-35].

E. COLLABORATIVE SOFTWARE AND AGENT TECHNOLOGIES USED IN FORMER NPS EXPERIMENTS AND PROJECTS

Extensive collaboration capabilities, intelligent tailoring and dissemination of knowledge were the requirements for the new defined CoABS methodology based on the usage of a multi-agent architecture residing within the global information grid and using it as the information exchange infrastructure. Through a better usage of technology, the concept can offer the means to improve communication between International Organizations (IO), Non-Governmental Organizations (NGO) and the military during humanitarian and peace operations.

An important theoretical concept is the creation of a so-called habitat where all the actors must be able to share resources (information, services, etc.) in a way that optimizes their ability to carry out their assigned missions effectively within the imposed security or policy constraints. As a result, this is a dynamically created concept meant to support a specific operational mission having interfaces, which ensure its connection with other habitats as well as with all other “legacy” systems, assets, organizations, or individuals.

Former experiments conducted by NPS have been focused on creating a Tactical Humanitarian Relief Operations habitat based on Groove - Groove Workspace providing the virtual space for immediate and direct connection between its users which can join efforts in performing a wide variety of collaborative activities.

This ensures a better communication within the team members, which benefit from the enhanced awareness of the other members sharing the workspace and consequently the speed and the quality of decision-making are increased. The Tactical Humanitarian Relief Operations habitat was created using a web-based application called the Relief Operation Coordination Center (ROCC) which is based on the use of HTML and active server pages (ASP) to interface with a database to insert, edit, view, delete and manipulate information to enhance multi-participant information sharing. The application is web-based in order to be mobile and accessible via internet connectivity and can be embedded in Groove in order to enhance the ability of geographically distributed users to plan, organize, and collaborate for problem solving.

Using the principles revealed by the DARPA CoABS program, which deals with the techniques used to safely control, coordinate, and manage large systems of autonomous software agents, the NPS has developed an agent-based application, also known as the Complex Humanitarian Emergency Situational Awareness Tool (CHESAT). Its main mission is to give the users self-aware capability to maintain situational awareness on each other's location and have a common knowledge of events in their area of operations.

The tool manages to integrate a series of in-house developed agents with the ROCC web-based application and with the Groove client. A short description of the functions covered by the two most important agents is:

The SA Management Agent provides the visual interface display for all participants through their web browser and is intended to support the shared situational awareness for all the tool's users. It provides display capability for a great amount of information, which allows a user to make informed decisions on how to assist in a particular event, and also provides the necessary information to coordinate assistance.

The Tracking Agent provides position-location information to the SA Management Agent for display in the browser. Data collected by the Tracking Agent comes from one of two input sources. One source uses manual inputs from the user who clicks and drags a user icon to a location on the display. The icon is then dynamically displayed to everyone accessing the CHESAT. A second input source is from a GPS receiver. This is accomplished by enabling a software agent that takes the GPS receiver input and transmits it to the SA Management Agent in the CHESAT, which subsequently moves the user icon to the correct location on the display. This method is much more accurate and requires no user input to adjust position information. This method of input is obviously hindered when a participant is obstructed from GPS detection (e.g. inside a building) or does not have a GPS receiver. In this situation, the user can easily switch to manual inputs by clicking the appropriate button on the CHESAT display.

Finally, the Complex Humanitarian Emergency Situational Awareness Tool may exist in two different spaces at the same time:

- Web server – that means it is accessible to all the users that can access the server where it resides;
- CoABS Grid – which can be understood as the infrastructure layer that has all the of the agents and services running on it.

Formerly gained experience in using GROOVE and ROCC applications, served the 2003 Tactical Battle Rhythm Experiment which analyzed and developed a model of real world TBR in the context of a notional Humanitarian Assistance Operation (HAO). Analyzing the battle rhythm at the Battalion/Squadron/Combat Service Support Group level, the experiment explored the enabling effects of collaborative tools over the battle rhythm at tactical level. After this experiment, it became obvious that collaboration may turn into a force multiplier if properly used in order to speed OODA loops and decision-making. The solution, due to its technical nature, is susceptible to incur a risk that proportionally increases to the dependence upon it; therefore any anomalous behaviors may sensitively influence the battle rhythm at tactical level. Collaborative solutions have to be robust, reliable and redundant in order to provide effectiveness and to ensure maximum safety for all who rely upon them.

At the tactical level, then, collaboration becomes a force multiplier if utilized when and where needed at a level robust enough to speed OODA loops and decision processes. As with any technology solution, however, the risk of network/technical issues increases proportionally to the dependence upon it. Tactical battle rhythm is much more sensitive to anomalous behaviors than are either operational or strategic battle rhythms. This sensitivity requires that collaborative solutions be well-implemented, robust and redundant.

V. INFORMATION EXCHANGED DURING COMPLEX HUMANITARIAN EMERGENCIES

A. BACKGROUND

Success in CHE as in military operations require executing high-tempo, coherent, decisive actions faster than the opponent can react, using command agility to obtain the envisioned results. Command agility, which is a human-developed quality, means flexibility to grasp fleeting opportunities, the power to be creative and innovative in real-time situations. On the other hand, this entire process depends on good data, on how it is obtained, processed and shared. Information superiority is generated by adopting network-centric concepts that allow shared awareness and self-synchronization. [Ref. 5:p. 55]

Based on the foundation concepts that define Complex Humanitarian Emergencies and Peace Operations introduced in Chapter II, using the technical theoretical support built in Chapters III and IV, Chapter V will approach the topic of exchanged information during CHE based on the raw information obtained from Cobra Gold 2003 Exercise. The main target is to use CoAX experiments philosophy and to try to apply it to CHE environments like Cobra Gold. By overlapping the two experiments we may extend the experience gained in a two year internationally conducted experiment proposing a framework applicable in future CHE environments. The solution is built on the official documents of the Cobra Gold 2003 Exercise provided by Col. (Ret.) Peter Leentjes working for APAN in the Center of Excellence in Disaster Management and Humanitarian Assistance by applying the lessons learned from the CoAX experiment.

B. COBRA GOLD 2003

Cobra Gold is an annual joint/combined, multilateral exercise, scheduled in Thailand involving U.S., Thai and Singaporean armed forces. In 2003, the exercise was focused on peace enforcement, humanitarian assistance and disaster relief operations. Throughout the entire exercise, military personnel from the United States and Thailand also conducted multiple medical, dental, veterinary and construction projects meant to improve the quality of life of local residents.

In order to obtain a realistic situation, the Center of Excellence in Disaster Management and Humanitarian Assistance exercise has developed suitable scenarios properly tailored for the players included in the Cobra Gold 2003 exercise. The center having responsibilities in preparing senior leadership for real operations was able to properly define roles for the broad political and humanitarian community involved, and was able to define the inter-relationships between the humanitarian and military communities.

The first step implied defining the scene, and creating a pseudo historical background for the exercise. A brief presentation of the road to the conflict.

1. Historical Background

Age-old ethnic and economic antagonisms between Greenland and Country X in the disputed border have defied repeated negotiation attempts. The roots for this conflict can be found in the mutual oppression of the minorities; authorities in Country X posed intolerable economic, education, health, and social service discrimination on Greenland ethnic personnel in the disputed border region.

Mounting ethnic and economic tensions, together with recognition that Greenland's autonomy within Country X was not viable, led to Greenland's invasion of parts of Country X. This military action has resulted in significant displacement of people within and a major humanitarian crisis in the region due to mass migration of refugees from both sides and internally displaced into neighboring Countries, primarily Country X.

The UN Security Council has condemned Greenland's invasion and Country X's oppression. It attempted negotiated withdrawal and diplomatic settlement, and has responded to requests from Country X to separate forces and supervise a cease-fire agreement.

Figure 19 presents a map of the area of operations and Figure 20 the Road to conflict timeline.

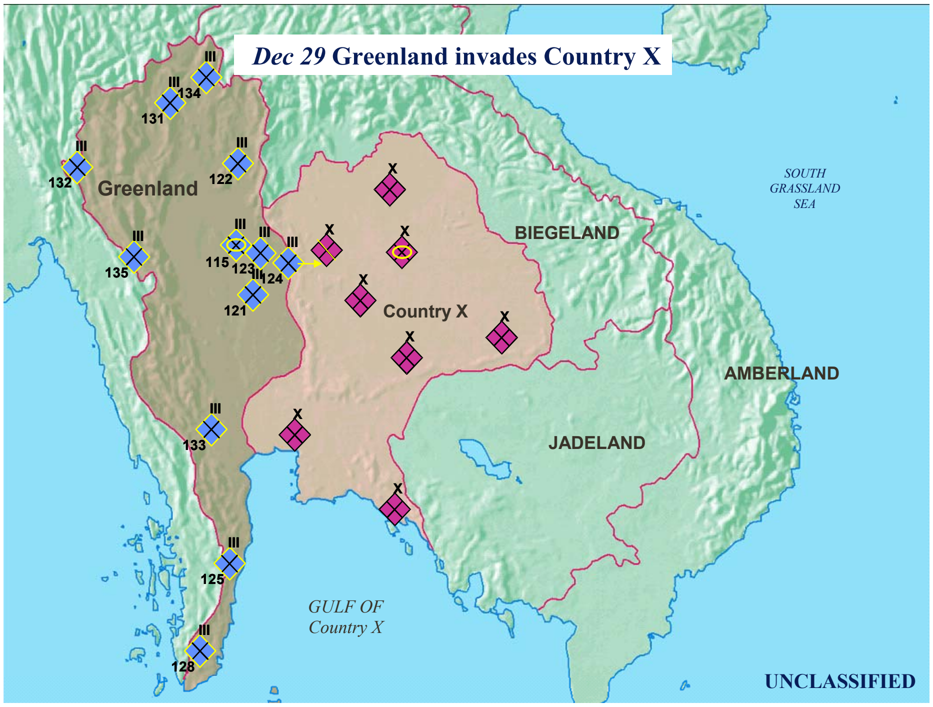


Figure 19. The Area of Operations

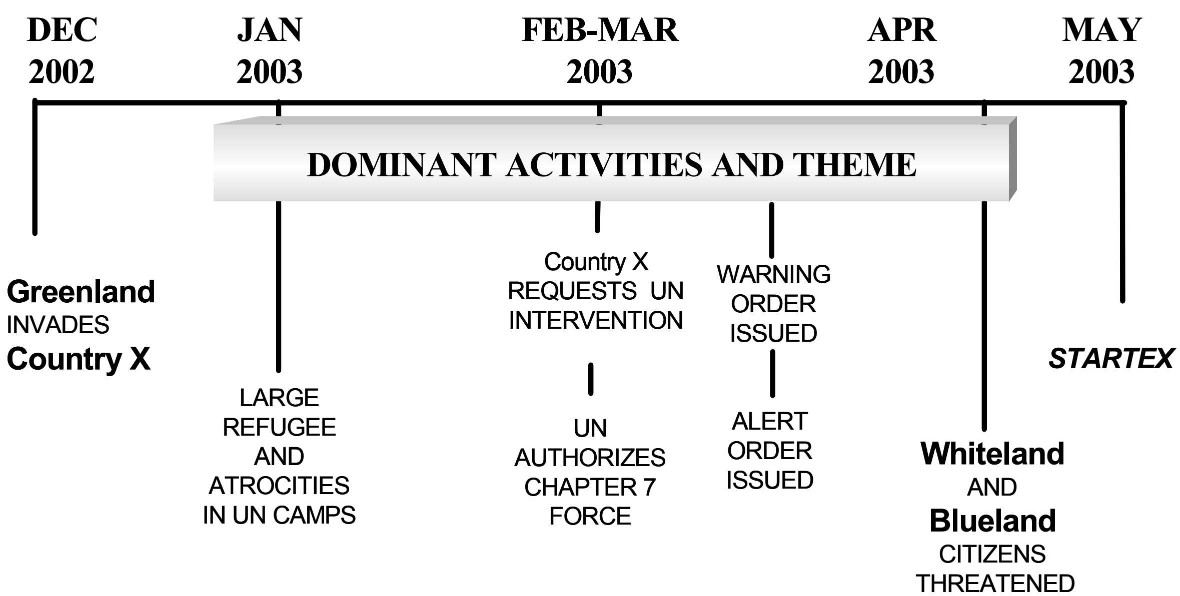


Figure 20. Road to Conflict Timeline

Following Country's X request UN decides to intervene by creating a Coalition/Combined Task Force (CTF), a multinational, multi service force including military forces of Whiteland, Blueland and Singland under the auspices of the United Nations designated to conduct a peace enforcement operation.

Since Cobra Gold 2003 was focused on peace enforcement, humanitarian assistance and disaster relief operations here are few of the exercise's civil-military themes:

- Security within the area of responsibility for humanitarian action;
- Shared access for Humanitarian Relief to facilities
- Generating a common operational picture
- Emergency support to humanitarian activities
- Managing a buffer zone of separation

Here is a list of the scene setting documents issued by the Center of Excellence and their designed impact within the given situation:

- **UN Resolutions***:* The Security Council Resolution calling peaceful resolution of conflict and peace agreement brokered by UN in the region
- *Impact*: Provided legal status to the force and to its operation
- **UN Peace Accord -- Military Aspects**: This is the agreement being enforced by the CTF to which military forces have to comply too.
- *Impact*: Permits CTF to establish enforcement criteria.
- **UN Secretary General Guidance Letter to CTF Commanders**: These are personal letters sent by the UN Secretary General to the CTF commanders containing the necessary guidance from UN perspective.

2. Impact: Expectations of Mission

The following documents define the baseline and the end-state for humanitarian activity, consequently requiring CTF analysis:

- **UN Consolidated Appeal for Humanitarian Assistance in the Region**: The document gives the preliminary needs assessment for Greenland and Country X, defines the chosen strategy, a list of specific needs, and provides the rough budgetary requirements evaluation. It also includes a list of implementing partners.
- **UN Campaign Plan**: defined the short and long range humanitarian concept of operations regarding the Greenland contingency

- **Summary of NGOs and agencies in the Area** contained the list of all NGOs and Agencies operating in Greenland and Country X including all their capabilities, assets, and personnel
- **Names, descriptions & locations of IDP/ Refugee Camps**
- **Political-Military Plan for Regional Contingency Operation** Peace Enforcement Operations (PEO) represent the application of military force, or threat of its use, normally pursuant to international authorization, to compel compliance with resolutions or sanctions designed to maintain or restore peace and order. This document issued by the Whiteland government has a strategic objective by putting the PEO / Humanitarian Assistance (HA) mission in context for CTF.

Based on the information contained in these documents a map of the existing situation at the beginning of the exercise is presented in Figure 21.

Derived from the United Nations Peacekeeping Mission Structure, the Coalition or Lead Nation Nations Peacekeeping Mission Structure is presented in Figure 22.

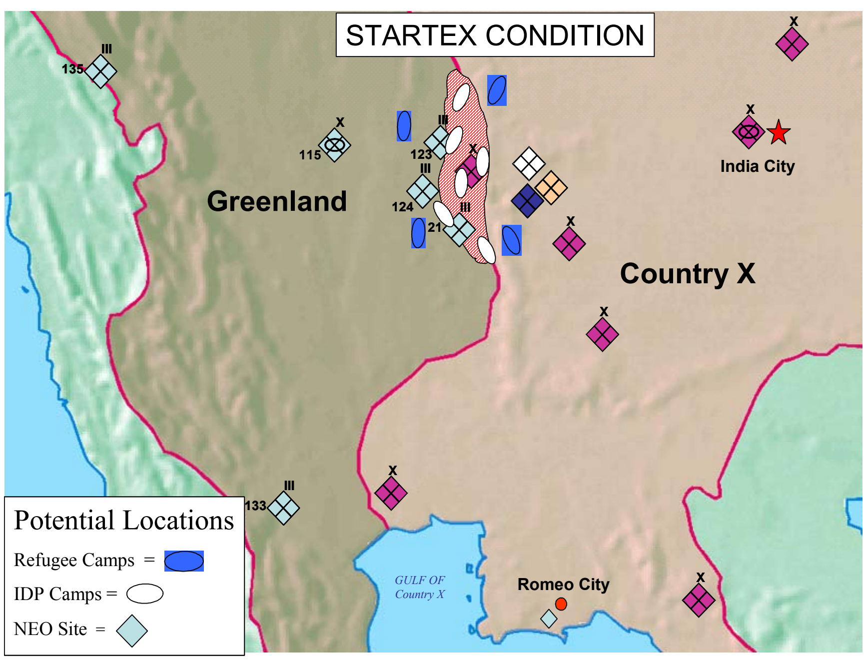


Figure 21. The Situation at the Beginning of the Exercise

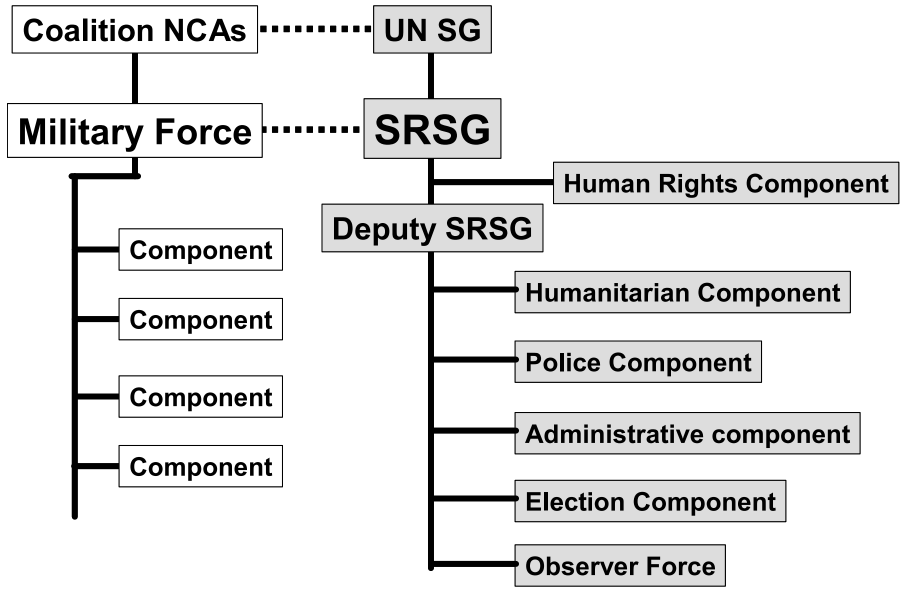


Figure 22. The Coalition Peacekeeping Mission Structure

Figure 23 defines the Multi-National Force roles in peace operations. MNF is an overarching term describing the broader force of participating nations, governments and agencies which unite their efforts in supporting their shared interests.

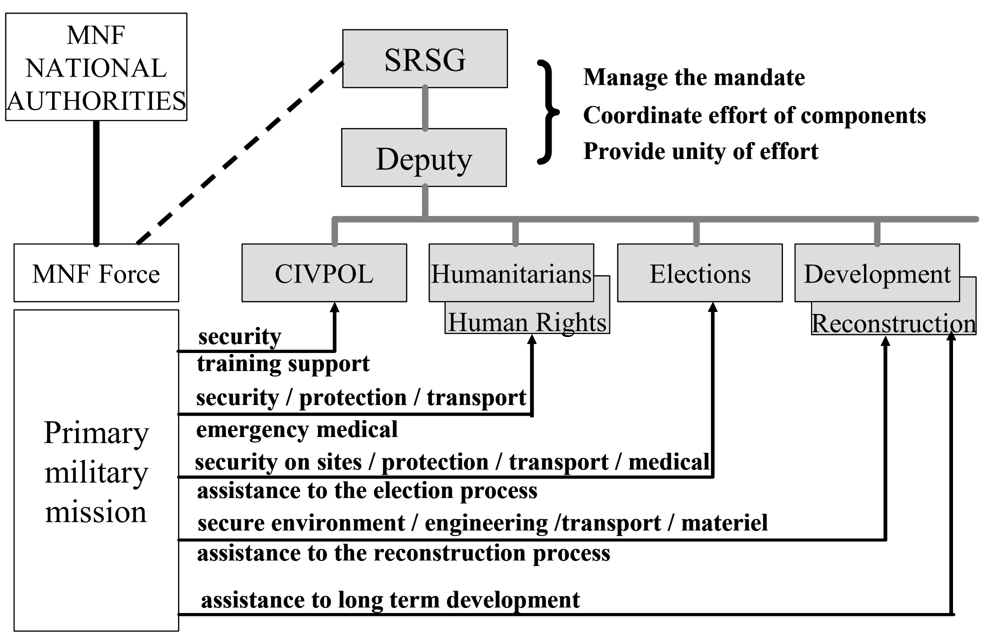


Figure 23. The Multi-National Force Roles in Peace Operations

Logically following this step is the need to define the Civil-Military Coordination Structure. Properly executed, Civil-Military Operations (CMO) provide optimum synergy between military and civilian International Organizations (IO)/Non-Governmental Organizations (NGO) crisis responders; and can reduce friction between the host nation governmental agencies, the civilian population and the participating military forces. The chosen architecture organizes the Civil Military Operations Center which is a coordination center formed from military and/or civilian assets that serves as the primary interface between military forces and the local population, IO, NGO, UN agencies and Foreign Governments.

As depicted by the organizers, there are two distinct levels of interface which exist between the CTF and the civilians:

- *The policy and coordination level* where the issues relate to defining policy and priorities, solving security issues, addressing political considerations and how to properly address parties, coordinating sectoral plans
- *The field operations level* which covers matters like: information exchange, coordination of day to day operations, provision of local support addressing security issues

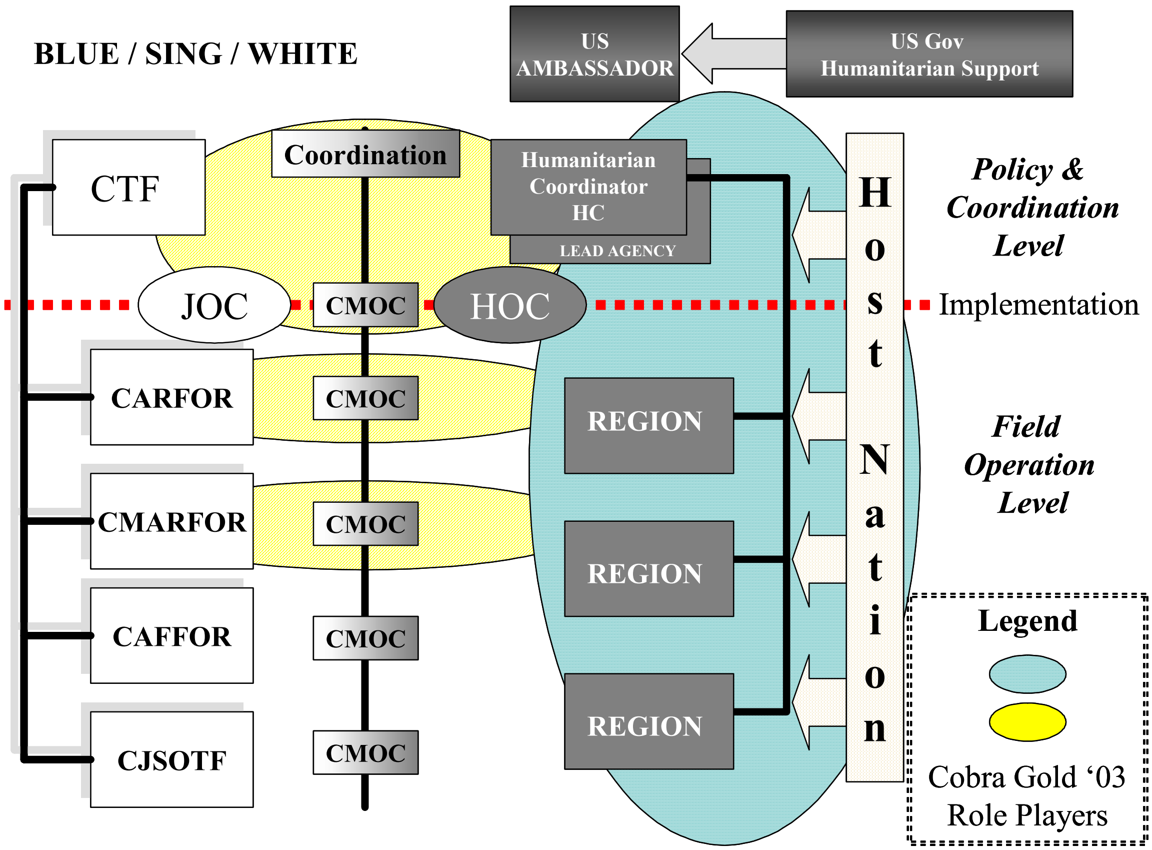


Figure 24. Cobra Gold 2003 Role Players Model

The exercise was built using the know-how developed by the Joint Warfighting Center, a Department of Defense (DoD) organization using computer simulations to support staff-level training. Since the actual trend requires scenarios depicting crises requiring humanitarian assistance, disaster relief, or similar emergency response (ER), JWFC developed a simulation suite using existing Joint Simulations and the High Level Architecture (HLA) which was designed to facilitate interoperability among simulations. The methodology used in ER training relies on an accepted practice to maximize training transfer: exercises maximize training transfer by assessing staff performance while "on the job".

There are four principles, which are applied to ensure quality training:

- Identify the training audience and the training objectives.
- Develop the exercise support infrastructure to support the training, to include plans for performance assessment.

- Conduct the exercise and gather performance assessment data.
- Conduct during and post exercise performance reviews to improve performance.

The infrastructure to enable exercise support includes simulation(s) that adequately represent the crisis forming the basis for the exercise, communications tools and networks to support information flows, and an exercise support staff. Pre-exercise support staff is required for exercise planning, assessment planning, simulation database development, scripted event development, network leasing and engineering, etc.

During the exercise the simulation allows the majority of the scenario to unfold in a coherent sequence. Scenario elements not amenable to simulation are introduced to the exercise using scripted events managed by a Master Scenario Events List (MSEL). MSELs are designed to initiate actions to cover most of the training objectives, but they represent only a part of the play, the rest being developed by the role players Small groups of people who are the interface between the simulation and the training audience are grouped in the “response cells” and they are in charge for introducing these scripted events.

Cobra Gold 2003, like other Pacific Command (PACOM) conducted exercises, used the Joint Theater Level Simulation (JTLS) system created by Rolands and Associates. JTLS is an interactive, multi-sided war-gaming system that models joint and coalition force warfare at the Operational Level. It models air, ground, naval, and special operations forces, including the simulation of movement, combat, logistics, and intelligence. Figure 25 shows the JTLS System as part of a more ample architecture.

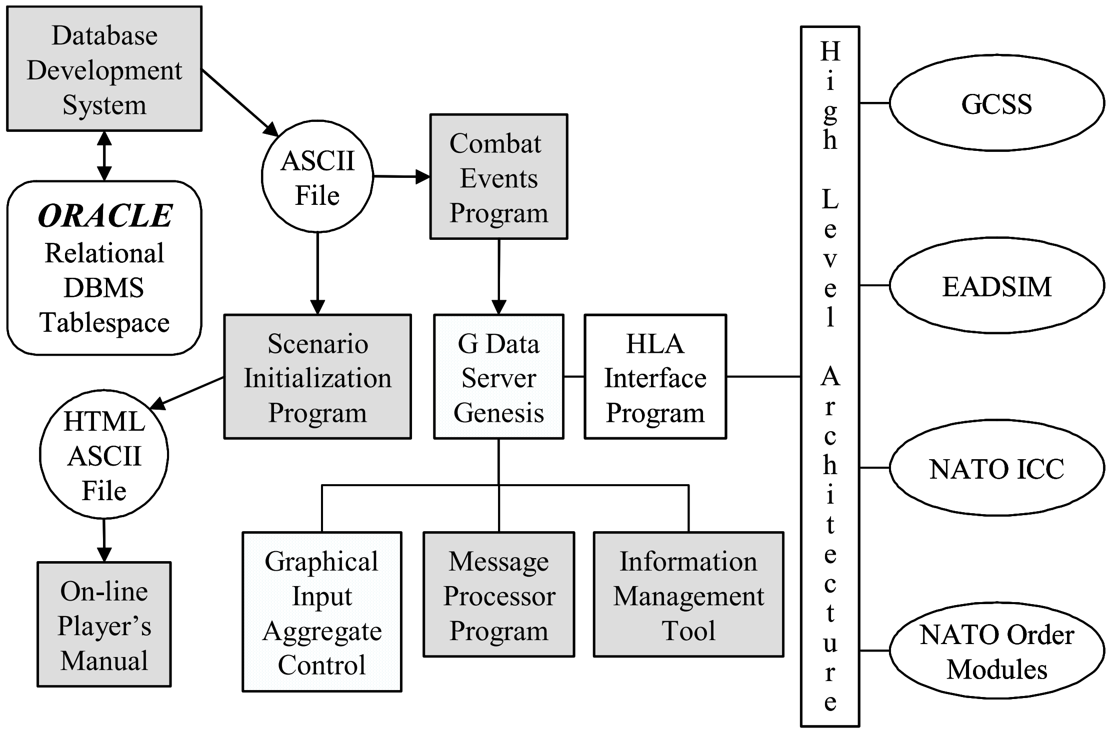


Figure 25. The JTLS System [From: Ref. 27]

The Cobra Gold Exercise Model was built by assigning different roles to the different player teams:

- *The Combined Exercise Control Group (CECG)* is assigned the entire exercise control. Its main tasks include managing the MESL input, coordinating the Political-Military play and the model activities (medical, supply, etc)
- *The Combined Task Force headquarters and staff (CTF HQ)* is assigned high level interaction tasks with the Special Representative for the Secretary General (SRSG), Humanitarian Coordinator (HC), Lead Agency, Human Rights Component). It injects the MESLs, coordinates the Political-Military play and conducts policy level play at CTF level, which means it synchronizes the Civilian and Military humanitarian activities at the strategic level to achieve the intended mission.
- *The Combined Marine Forces (CMARFOR)* and *The Combined Army Forces (CARFOR)* besides interacting with commanders and staff must also initiate MESLs.

The applied Coalition/Combined Task Force concept of operations utilizes a sequential approach, employing coercion early followed by rapid, decisive operations against Greenland forces, achieving an earlier cessation of hostilities and allowing the CCTF to then focus resources on humanitarian assistance (HA). CCTF conducts operations in four phases and two stages:

PHASE 1: Deploy to area of operations and establish lodgment

PHASE 2: Removal of Greenland forces from Country X

STAGE A: Information operations

STAGE B: Force compliance with overwhelming force

PHASE 3: Transition to HA and buffer zone maintenance

PHASE 4: Transition to UN authorized follow-on forces

Figure 26 presents the concept of operations and its timeline:

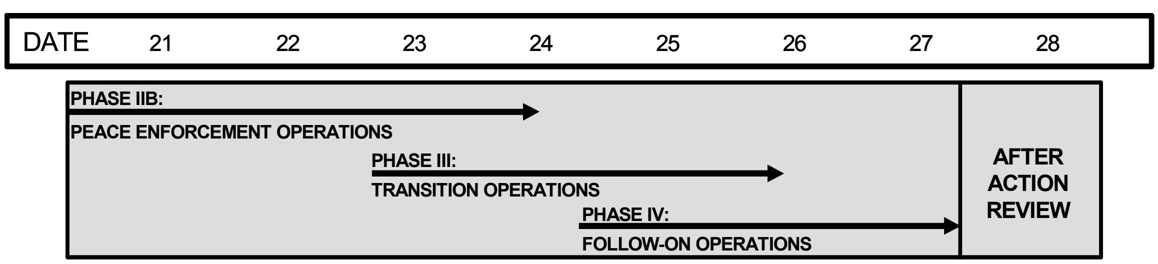


Figure 26. The Concept of Operations/CPX Timeline

Presenting the main aspects of the 2003 Cobra Gold Exercise was necessary to uncover the common features that allow us to extend the use of the know-how obtained from the CoAX Experiments within the Humanitarian Operations Scenario. Consequently, this approach provides an answer to the primary research question of this paper which looks for convenient means to improve situational awareness in CHE and peace operations by sharing access to remote databases and expert sources.

C. APPLYING THE COAX METHODOLOGY

As mentioned in Chapter IV, the collaborative effort entitled CoAX was a Technology Integration Experiment carried out under the auspices of DARPA's Control of Agent Based Systems (CoABS) Program. Using the CoABS Grid framework, the CoAX agent infrastructure groups agents into domains reflecting real-world organizational, functional and national boundaries; therefore, security and access to agents and information can be governed by policies at multiple levels. [Ref. 24:pp. 26-35].

Similarities between the CoAX Experiment and the 2003 Cobra Gold Exercise permit a parallel analysis and a shared implemented solution. Both exercises:

- Involve creating and operating a Coalition of Forces. There are various incompatible, "come-as-you-are" or foreign systems that need to be rapidly configured into a cohesive whole, within an open, heterogeneous and dispersed environment. Since there are no coordinated systems, Coalition scenarios require a rapid, flexible approach that allows capabilities to be assembled at run-time.
- Define a complex Coalition Command Structure involving UN, Governments, Non-Government Organizations, representatives of all Coalition countries, the Coalition Headquarters and subordinate fighting forces.

The difficult task in integrating information across a coalition requires human-level interoperability and proper usage of technology. Using both perspectives, the CoAX approach defines Human domains and software agent domains. From the human perspective applied to Cobra Gold Exercise the four types of domains may include:

- *Organizational domains* such as the joint task force headquarters
- *Country domains* (Blue, Sing, White) with each national command chain a separate, self-contained domain
- *Functional domains*, where entities collaborate on common tasks as human rights, humanitarian, police, etc.
- *Individual human domains of responsibility* where commanders have responsibility for their own headquarters and all subordinate ones

Since these domains overlap or interact at different levels depending on the chosen perspective, technical systems encounter difficulties in dealing with them.

From the technology perspective, the challenge when trying to integrate agents with systems is the need to define a basic infrastructure allowing discovery of new agents and the messaging between them. The CoABS Grid solution allows registration and advertisement of agent capabilities, and communication by message-passing. Other newly implemented facilities provide extra functionality to the Grid such as: logging, visualization, encryption of messages and agent authentication.

An important feature introduced by the CoAX experiments, addresses the dangers coming with the benefit of using agents. Agent autonomy and absence of human supervision create opportunities for malicious agents to destructively act producing severe damage to military operations. The so-called Knowledgeable Agent-oriented System (KAoS) provides services that help assure that agents from different developers and running on diverse platforms will operate within the bounds of established policies and will be continually responsive to human control so that they can be safely deployed in operational settings [Ref. 28:pp. 375-418]. KAoS services and tools are designed to allow for the specification, management, conflict resolution, and enforcement of policies within the specific contexts established by complex military organizational structures.

As a result, CHE environments may benefit from using KAoS domain management services by gathering agents into logical domains that reflect the existing organizational structures, the administrative groups or the task-oriented teams. Within the Cobra Gold Exercise we may use the previously defined logical domains allowing the existence of complex hierarchical and overlapping structures. Like any CoAX agent domain, they should consist of a unique instance of a domain manager (DM) in conjunction with the agents registered to it. Agents sharing one or more common properties may be grouped in connotation-defined domains. The Domain Manager manages agent registration and serves as a single point of administration and enforcement of policies across domains, hosts, Virtual-machines.

By defining a policy, the behavior of one or more agents is constrained by an imposed requirement. The policy acts on agents who may not be domain-aware, not by imposing on them how to perform their task but more likely by defining the conditions under which the task may be executed. As previously mentioned, KAoS domain management comprises those policies that govern processes like authorization, encryption, access control, and resource control. KAoS also comprises an important facility that allows forming, maintaining and separating teams of agents through the process of agent-to-agent communication using appropriate semantics. In order to discover and resolve policy conflicts and to be able to perform other kinds of policy analysis, DARPA's Agent Markup Language is used to represent all the defined policies in ontologies and domain manager's reasoning is accomplished by using an efficient description logic-based approach. Policies are more flexible, fine-grained, extensible, and can be dynamically reconfigured because policy specification is separated from policy enforcement mechanisms. As a result, changes in the policies will not necessarily lead to changes in the source code.

Considerations included up to this moment permitted a conceptual overlapping the principles used in CoAX experiments on the 2003 COBRA GOLD Exercise. In contrast, a CoAX experiment was conducted using a pool of predefined agents and technologies designed and developed by different agencies to assist actors involved in the exercise which is not the case of our CHE exercise. In order to provide a real understanding of the intricacy implied, Figure 27 presents a typical CoAX domain structure and Figure 28 provides an overview of technologies and agents.

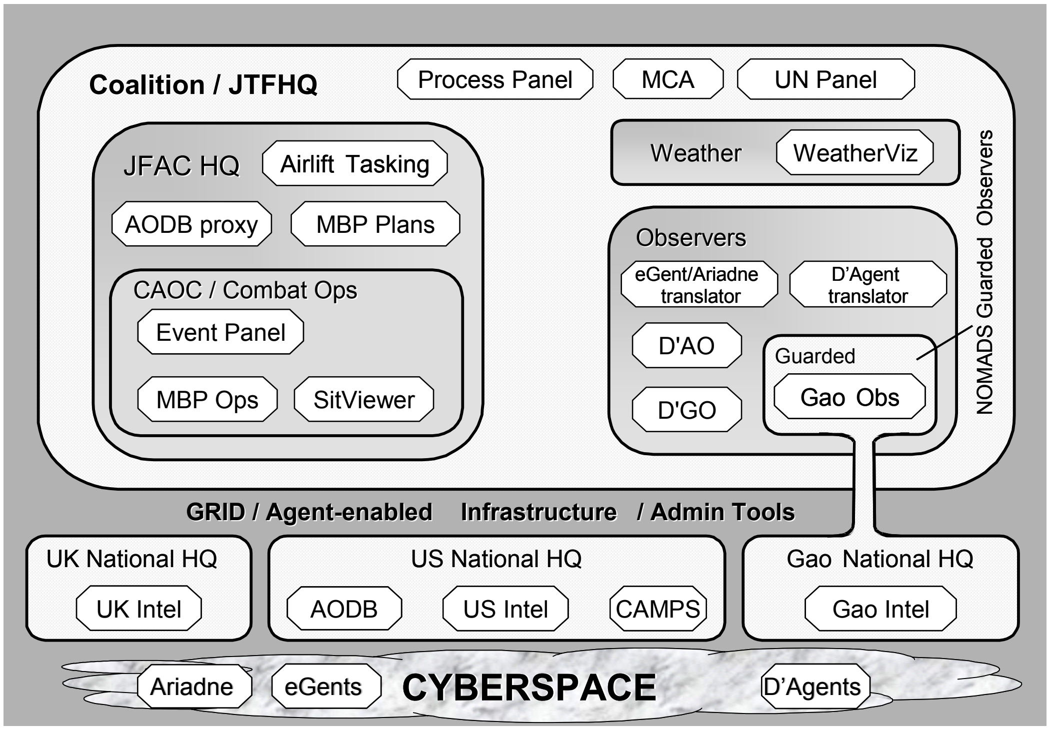


Figure 27. Typical CoAX Domain Structure; Domains Are Indicated by Rounded Rectangles; Agents by Angled Rectangles [From: Ref. 24:pp. 26-35]

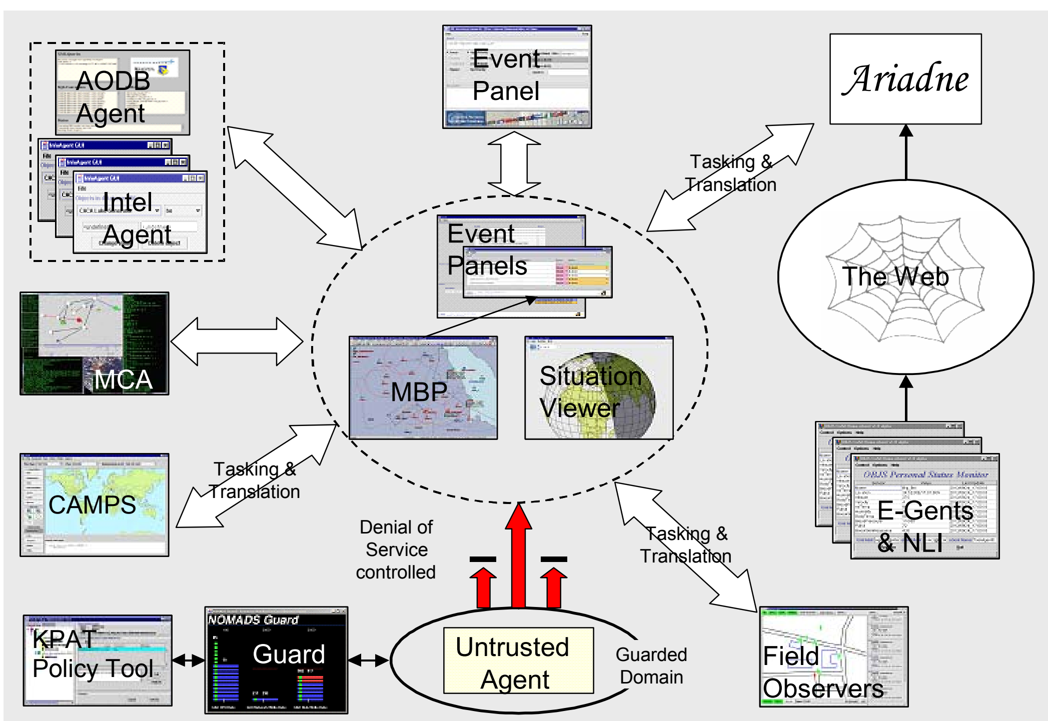


Figure 28. Technologies and Agents [From: Ref. 24:pp. 26-35]

The central visualization and planning tools (Figure 28) find and acquire data and services from the other agents and systems either directly or using intermediate tasking and translation agents.

Used in the CoAX at the JFAC level, the Master Battle Planner (MBP) is characterized as a highly effective visual planning tool for air operations. Since the Joint Theater Level Simulation (JTLS) system used in the 2003 Cobra Gold Exercise was designed as an interactive, multi-sided war-gaming system capable to be integrated as part of a more ample architecture, one might consider using both systems interconnected in the Humanitarian Scenario. The output coming from the JTLS would serve as input in the MBP this way assisting operators by providing them with an intuitive visualization of the data introduced by exercise planners. The map-based graphical user interface would be populated with scenario-generated elements, which can be used by the operator in performing his assigned task. Not to be neglected in this interconnection are the possible incompatibilities between the systems coming from different understanding of the military terms (such as: how is an air mission defined).

The MBP also uses translated and forwarded information coming from systems, mobile devices or observers. The Ariadne system gathers weather information from the sites and adjusts it to the format it can be included as part of the normal picture of the air situation. *Wrappers/mediators* solve queries by accessing different web-based data sources as if they form a single database. The XML technology is used to shift data from one system to the other.

Central visualization tools include event panels where important incoming issues are divided into sub-tasks in order to facilitate their control. Dynamic data gathered by mobile devices and observers along with information feeds provided by agents are joined on the Situation-viewer agent used to monitor an almost real-time situation. The information gathered from agents residing on mobile devices, from eGents agents which use e-mail for communication is gathered on websites used as data source by the Ariadne system. A translator agent using XSLT, forms a message stream which is sent to the MBP and to the Situation Viewer agent. This method can be easily applied within CHE where rapidly evolving situations require shared situational awareness an immediate response. Figure 29 graphically describes the following sequence of steps:

- An eGent client subscribes to eGents (agents) running on mobile devices placed in the less accessible zones of interest.
- The information gathered from these agents is published by the client on a web page
- The Ariadne system gathers the information from the web pages and produces an XML document
- The XML document is transformed using XSLT into the needed format to be used by other agents like MBP and Situation Viewer for visualization.

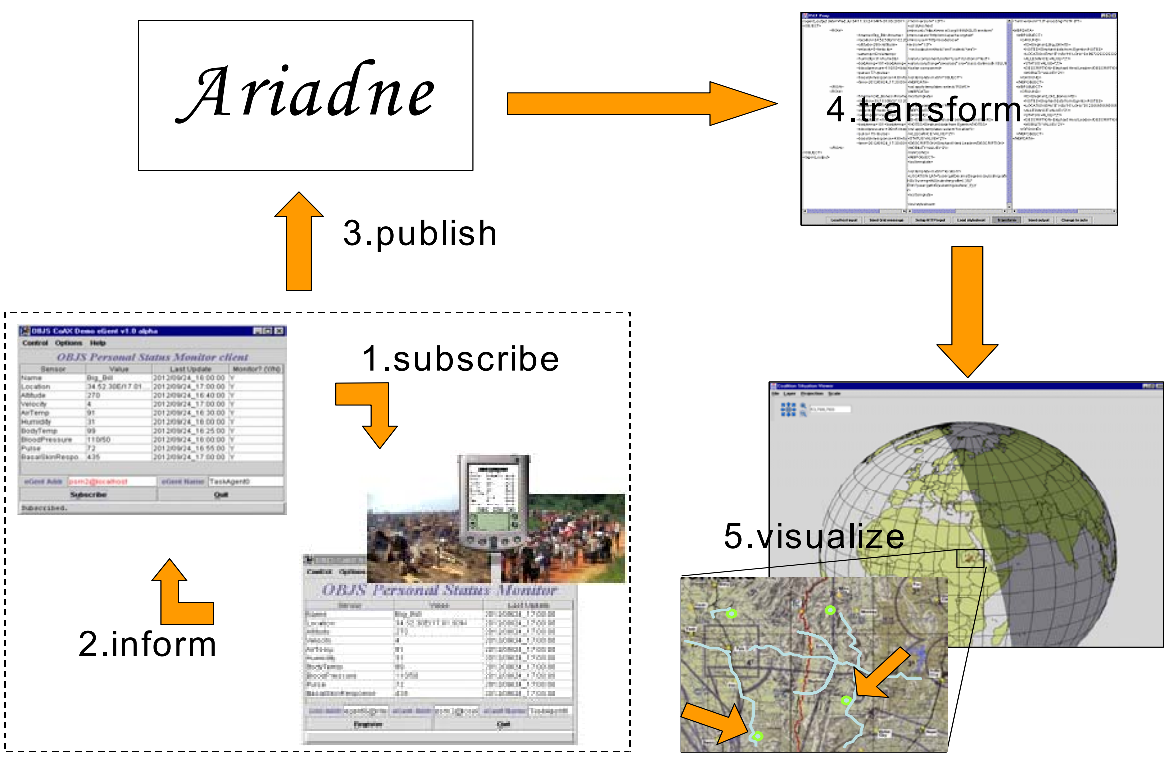


Figure 29. Data Feeds from Mobile Devices and Observers [After: Ref. 22:pp. 26-35]

Another situation where CoAX experience may be applied is when an important asset hosting observer agents becomes temporarily unavailable. As a precondition, there has to be a second facility capable of hosting the observer agents. In that case, by using a system like NOMADS, the full execution state of the agents captured from the unavailable platform may be moved to the new one using the Aroma VM mechanism. The advantage is that on the new platform agents may be restarted without losing their ongoing computations and coalition forces may benefit from the continuous flow of information.

CoAX experiments have demonstrated the benefits of using agent technology in Coalition Operations. Similar environments like Complex Humanitarian Emergencies and Peace Operations may benefit from the overlapped use of agent technology with the peer-to-peer collaborative tools.

Consider a possible scenario. Within the network, there are several computers hosting software applications that must collect, process, and share their results with other network stations. At the same time, each station may host agents capable to provide network awareness capabilities. According to the collaboration scenario, station X and Station Y need to share information regardless of the role they might need to take (server or client). Agents residing on both stations are network aware and help establishing the intercommunication. This scenario is consistent with the ever increasing requirement for ubiquitous and pervasive computing. As a result agents are required to fulfill different functions and to provide services on the platforms where they are embedded.

Based on the idea of building a Tactical Humanitarian Relief Habitat which has been experimented in former NPS conducted projects and experiments, Figure 30 presents a possible multi-agent architecture integrating remote databases with situational awareness tools using CoABS, ROCC and Groove.

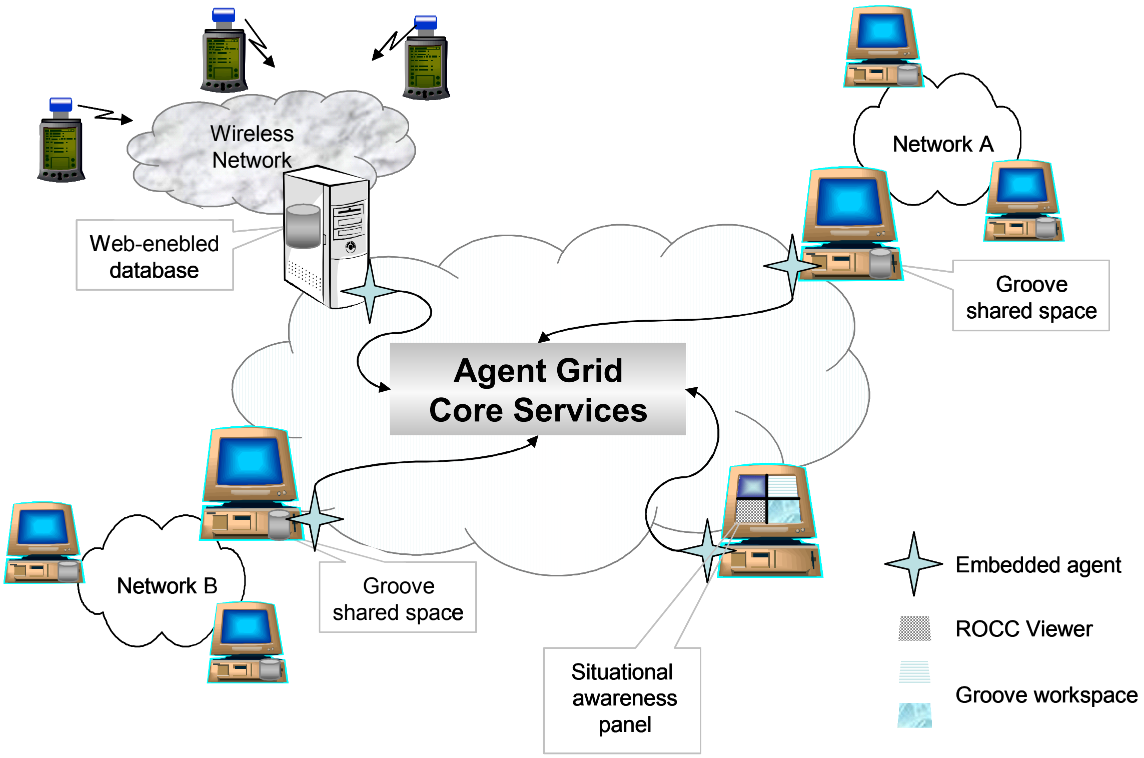


Figure 30. Proposed Multiagent Architecture

Finally, Chapter VI will review the basic parts covered in the paper trying to state a direct answer to the research questions and to suggest future courses of investigation.

THIS PAGE INTENTIONALLY LEFT BLANK

VI. CONCLUSIONS

This study is focused on possible approaches to permit employing a multi-agent architecture as the way to integrate remote databases and expert sources with situational awareness tools. The research effort was centered on the Humanitarian Operations Scenario and implied gathering and combining information from different sources.

A primary goal was to select and define the theoretical foundation bricks, both military and technical, to allow future architectural developments. Practical approaches cited in the literature along with implementations developed in the NPS experiments and projects served the task of better explaining these concepts.

Web-enabled databases, expert systems, collaborative software and even mobile-agent technologies have gained the confidence of software developers and have been used across different projects. Based on the phenomenal growth of Internet and the World Wide Web, the scientific community joined effort in promoting the combined use of these technologies in unfriendly environments, across different platforms.

Internationally conducted research, like the DARPA's CoAX experiments, provided valuable expertise in putting technologies to work together. Using a software-agent testbed the research provided enough evidence to sustain the exercise's main hypothesis, which asserted the idea that an agent-computing paradigm presents a good way of building complex software systems in general, and therefore, is considered a valuable solution for the technical issues, when establishing coherent Command and Control in a Coalition setting.

The Humanitarian Scenario provides the appropriate environment where theoretical principles and technologies may be combined in order to improve the outcome of the joint effort involving many actors belonging to different segments of the human society either civil or military. This study continued the previously NPS initiated research towards documenting the means to develop a collaborative, Information Technology-based, operation support system capable of improving information sharing during Complex Humanitarian Emergencies. This thesis tried to develop a sense of understanding on technological aspects that have not been thoroughly covered in former studies. Its centered task was to convey documented answers to the proposed research questions and here are the conclusions of this guided research:

Primary research question: *How to improve situational awareness in Complex Humanitarian Emergencies and Peace Operations by sharing access to remote databases and expert sources?*

The answer to this question can be found in the development of multi-agent architectures based on the DARPA CoABS philosophy. The CoAX experiments have shown how software agents can accomplish functions within a Coalition Scenario where information systems and infrastructure services must be shared between participating forces. Based on the assigned roles, agents can provide the necessary interoperability between Coalition systems despite the continuously changing environment.

Agents may be used not only to ensure proper communication between systems belonging to different technologies and different Coalition Forces, but they may also serve human operators' needs of information. The development of human-agent collaboration guarantees better situational awareness since human needs for information are solved in an almost transparent manner.

Agents and technologies are linked within this architecture in order to provide the required level of shared understanding:

- As randomly distributed processes within the existing network, agents communicate, share information taking advantage of the facilities offered by a middleware layer, such as the CoABS Grid;
- Software agents can be grouped into logical domains which follow human organizational structure using KAoS domain management services according to the defined policies. Policies frame agents' behavior, defining *what* they are supposed to do but not *how*.
- Web-hosted information can be processed as if contained in databases by using the wrapper approach

- XML and XSLT are the applied technologies that ensure data semantics and portability between different platforms and products
- Human operators benefit from consistent organized processes and event panels that put together meaningful information gained from different sources

Since real-world problems are usually distributed in nature multi-agent architectures are suitable to fit real-world situations because they provide a distributed model of computing.

Subsidiary research question: *How can the proposed architecture enable information sharing in a peer-to-peer collaborative environment?*

As previously defined, the peer-to-peer collaborative model, also called decentralized computing, is based on powerful clients that directly exchange information without server assistance. As a result of the advances in communication technology, such as mobile wireless, peer-to-peer solutions became more helpful. Combining the multi-agent architecture with the peer-to-peer model we may merge the advantages brought by both technologies and as a result obtain improved situational awareness. Chapter V has introduced a possible multi-agent architecture integrating remote databases with situational awareness tools using CoABS, ROCC and Groove.

Future developments should include joining existing NPS developed implementations, such as ROCC, and overlap them to the CoABS philosophy. Developing a collection of dedicated applications could serve the needs of the actors within the CHE and consequently improve their situational awareness. The central requirement would be to allow dynamic, automatic and trusted composition of services using heterogeneous technologies joint by the agent-based architecture.

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

1. DARPA Internet Site. What is CoABS?, [http://coabs.globalinfotek.com]. Date accessed: March 19, 2004.

2. Office for the Coordination of Humanitarian Affairs: Orientation Handbook On Complex Emergencies, August 1999, [http://www.reliefweb.int/library/documents/ocha__orientation__handbook_on__.htm]. Date accessed: March 19, 2004.

3. Guidelines on “The Use of Military and Civil Defence Assets to Support United Nations Humanitarian Activities in Complex Emergencies”, 2003, [http://www.reliefweb.int/mcdls/mcdu/mcdu.html]. Date accessed: March 19, 2004.

4. US Institute of Peace Good Practices: Information Sharing in Complex Emergencies, 2002, [http://www.usip.org/virtualdiplomacy/publications/reports/11.html]. Date accessed: March 19, 2004.

5. Alberts, S. David, Garstka, J. John, and Stein, P. Frederick, Network Centric Warfare – Developing and Leveraging Information Superiority – 2nd Edition (revised), 1999, CCRP, [http://www.ida.org/DIVISIONS/sctr/cpof/2ndNCW.pdf]. Date accessed: March 19, 2004.

6. UN Office for the Coordination of Humanitarian Affairs OCHA-Online, 2004, [http://www.reliefweb.int/ocha_ol/about/facts.html]. Date accessed: March 19, 2004.

7. Evans, P. B. and Wurster, T. S., "Strategy and the New Economics of Information", Harvard Business Review, September - October 1987, pp. 71 – 82.

8. Kroenke, David M., Database Processing. Fundamentals, Design and Implementation, Prentice Hall, Eighth Edition, 2001, pp. 516-521.

9. Dolan, Dana, Groove Workspace 2.5 - Detailed Evaluation Paper Software Productivity Consortium, 2003, [http://www.mygroove.biz/Wissen/GRV%20Paper.pdf]. Date accessed: March 19, 2004.

10. Kotz, David and Gray, Robert S., Mobile Agents and the Future of the Internet ACM Operating Systems Review 33(3), August 2000, pp. 2-13.

11. Marakas, M. G., Decision Support Systems in the 21st Century Prentice Hall, 2001, p. 457.

12. Genesereth, M. and Nilsson, N., *Logical Foundations of Artificial Intelligence*, Morgan Kaufman, 1987.

13. Wooldridge, Michael, An Introduction to MultiAgent Systems John Wiley & Sons Ltd., 2001.

14. Jennings, N. R., An Agent-Based Approach for building complex software systems, Communications of the ACM, 44(4): 35-41, 2001.

15. Jennings N. R., On Agent Based Software Engineering – Artificial Intelligence, 2000, p. 117.

16. Kolp, M., Giorgini, P. and Mylopoulos, J. (2002). Organisational Multi-Agent Architectures: A Mobile Robot Example. Proceedings of AAMAS, 2002, Bologna, Italy, pp. 94-95.

17. Fischer, K., Agent-Based Design of Holonic Manufacturing Systems. Robotics and Autonomous Systems, Vol. 27, 1999, pp. 3-13.

18. Foster I. and Kesselman C., editors. The Grid: Blueprint for a new Computing Infrastructure, Morgan Kaufmann, July 1998.

19. Foster I. and Kesselman C., editors. The Anatomy of the Grid: Enabling scalable Virtual Organizations, Intl. J Supercomputer Applications, 2001.

20. Jini Technology Community Resource [http://www.jini.org]. Date accessed: March 19, 2004.

21. LCDR Dylan Schmorrow, The DARPA Control of Agent Based Systems (CoABS) Program and Challenges for Collaborative Coalition. [http://www.aiai.ed.ac.uk/project/coalition/ksco/ksco-2002/pdf-parts/W-ksco-2002-paper-30-schmorrow.pdf]. Date accessed: March 19, 2004.

22. Kahn, L. Martha and Cicalese, Cynthia, The CoABS Grid, Springer-Verlag, Berlin Heidelberg, 2003.

23. DARPA Internet Site [http://coabs.globalinfotek.com/public/Grid/What_is_Grid.htm]. Date accessed: March 19, 2004.

24. Allsopp,D. N., Beautement, P., Bradshaw J. M., Durfee, E. H., Kirton, M. J., Knoblock, C. A., Suri, N., Tate A., and Thompson, C. W., 2002 Coalition Agents Experiment: Multi-Agent Cooperation in an International Coalition Setting, IEEE Intelligent Systems, May/June, 26-35.

25. Semantic Web Activity Statement Web Site: [http://www.w3.org/2001/sw/Activity]. Date accessed: March 19, 2004.

26. Bradshaw, J. M. et al., *Making Agents Acceptable to People.* In Handbook of Intelligent Information Technology. IOS Press, Amsterdam, Holland (editors N. Zhong and J. Liu) (2003).

27. Rolands, JTLS Executive Overview, 2003. Available Online Via [http://www.rolands.com/].

28. Bradshaw, J. M., Dutfield, S., Benoit, P. and Woolley, J. D., “KAoS: Toward an Industrial-Strength Generic Agent Architecture,” Software Agents, AAAI Press/The MIT Press, Cambridge, Massachusetts, 1997, pp. 375-418.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Dan Boger
   Naval Postgraduate School
   Monterey, California

4. Alex Bordetsky
   Naval Postgraduate School
   Monterey, California

5. Glenn Cook
   Naval Postgraduate School
   Monterey, California

6. Cantemir Ahciarliu
   Romanian Air Force
   Bucharest, Romania