REPORT DOCUMENTATION PAGE	Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	7.Sep.00	THESIS

4. TITLE AND SUBTITLE
EXPLOITATION OF WEB TECHNOLOGIES FOR THE JOINT BATTLESPACE INFOSPHERE

5. FUNDING NUMBERS

6. AUTHOR(S)
CAPT WEBSTER PAUL T

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
NAVAL POSTGRADUATE SCHOOL

8. PERFORMING ORGANIZATION REPORT NUMBER
CY00371

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
THE DEPARTMENT OF THE AIR FORCE
AFIT/CIA, BLDG 125
2950 P STREET
WPAFB OH 45433

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION AVAILABILITY STATEMENT
Unlimited distribution
In Accordance With AFI 35-205/AFIT Sup 1

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

14. SUBJECT TERMS

15. NUMBER OF PAGES
100

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT

ABSTRACT

Today's military command and control (C2) systems provide much information to today's combatants, but because these legacy systems are disjointed, there exists an overload of poorly organized and incomplete data during operations. These systems tend to be "stovepiped," inflexible, difficult to integrate, and hard to use in building a common operational picture. The information exchange model for DoD C2 systems is migrating away from dedicated point-to-point and broadcast systems (information push) toward a model based partly on Internet and World Wide Web technologies (publish and subscribe). The USAF Scientific Advisory Board has created a visionary combat information management concept called the Joint Battlespace Infosphere in response to this movement.

The purpose of this thesis is to identify and evaluate emerging Internet/Web-based technologies that could be employed by the DoD to improve upon existing information exchange services. This survey will examine the strengths and weaknesses of technologies such as client-server architectures, search engines, middleware, intelligent software agents, and multicast delivery tools that could enhance the development of the Joint Battlespace Infosphere.

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

EXPLOITATION OF WEB TECHNOLOGIES FOR THE JOINT BATTLESPACE INFOSPHERE

by

Paul T. Webster

June 2000

Thesis Co-Advisors: William Kemple
Heather Dussault

Second Reader: Gary Porter

Approved for public release; distribution is unlimited.

EXPLOITATION OF WEB TECHNOLOGIES FOR THE JOINT BATTLESPACE INFOSPHERE

Paul T. Webster-Captain, United States Air Force
B.S., State University of New York at Buffalo, 1995
Master of Science in Systems Technology-June 2000
Advisor: Heather Dussault, Air Force Research Laboratory
Co-Advisor: William Kemple, Department of Command, Control, and Communications
Second Reader: Gary Porter, Department of Command, Control, and Communications

Today's military command and control (C2) systems provide much information to today's combatants, but because these legacy systems are disjointed, there exists an overload of poorly organized and incomplete data during operations. These systems tend to be "stovepiped," inflexible, difficult to integrate, and hard to use in building a common operational picture. The information exchange model for DoD C2 systems is migrating away from dedicated point-to-point and broadcast systems (information push) toward a model based partly on Internet and World Wide Web technologies (publish and subscribe). The USAF Scientific Advisory Board has created a visionary combat information management concept called the Joint Battlespace Infosphere in response to this movement.

The purpose of this thesis is to identify and evaluate emerging Internet/Web-based technologies that could be employed by the DoD to improve upon existing information exchange services. This survey will examine the strengths and weaknesses of technologies such as client-server architectures, search engines, middleware, intelligent software agents, and multicast delivery tools that could enhance the development of the Joint Battlespace Infosphere.

DoD KEY TECHNOLOGY AREA: Command and Control (C2) Systems, Wide Area Networks (WANs)

KEYWORDS: Joint Battlespace Infosphere, Command and Control Systems, Global Information Grid, World Wide Web, Internet, Search Engines, Software Agents, Middleware, Multicast Delivery

REPORT DOCUMENTATION PAGE		Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE June 2000	3. REPORT TYPE AND DATES COVERED Master's Thesis
TITLE AND SUBTITLE : Exploitation of Web Technologies for the Joint Battlespace Infosphere		5. FUNDING NUMBERS
6. AUTHOR(S) Webster, Paul T.		
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) Air Force Research Lab Rome, NY		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	12b. DISTRIBUTION CODE
Approved for public release; distribution is unlimited	

ABSTRACT ***(maximum 200 words)***

Today's military command and control (C2) systems provide much information to today's combatants, but because these legacy systems are disjointed, there exists an overload of poorly organized and incomplete data during operations. These systems tend to be "stovepiped," inflexible, difficult to integrate, and hard to use in building a common operational picture. The information exchange model for DoD C2 systems is migrating away from dedicated point-to-point and broadcast systems (information push) toward a model based partly on Internet and World Wide Web technologies (publish and subscribe). The USAF Scientific Advisory Board has created a visionary combat information management concept called the Joint Battlespace Infosphere in response to this movement. The purpose of this thesis is to identify and evaluate emerging Internet/Web-based technologies that could be employed by the DoD to improve upon existing information exchange services. This survey will examine the strengths and weaknesses of technologies such as client-server architectures, search engines, middleware, intelligent software agents, and multicast delivery tools that could enhance the development of the Joint Battlespace Infosphere.

14. SUBJECT TERMS Joint Battlespace Infosphere, Command and Control Systems, Global Information Grid, World Wide Web, Internet, Search Engines, Software Agents, Middleware, Multicast Delivery			15. NUMBER OF PAGES 106 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

EXPLOITATION OF WEB TECHNOLOGIES FOR THE JOINT BATTLESPACE INFOSPHERE

Paul T. Webster
Captain, USAF
B.S., State Unitversity of New York at Buffalo, 1995

Submitted in partial fulfillment of the
requirements for the degree of

**MASTER OF SCIENCE IN SYTEMS TECHNOLOGY
(COMMAND, CONTROL, AND COMMUNICATIONS)**

from the

NAVAL POSTGRADUATE SCHOOL
June 2000

Author: _____
Paul T. Webster

Approved by: _____
Heather Dussault, Thesis Co-Advisor

_____
William Kemple, Thesis Co-Advisor

_____
Gary Porter, Second Reader

_____
Daniel Boger, Chairman
Command and Control Academic Group

THIS PAGE INTENTIONALLY LEFT BLANK

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

EXECUTIVE SUMMARY

Today's military command and control (C2) systems provide much information to today's combatants, but because these legacy systems are disjointed, there exists an overload of poorly organized and incomplete data during operations. These systems tend to be "stovepiped," inflexible, difficult to integrate, and hard to use in building a common operational picture (COP). This results in very little interoperability among the services and even less with coalition forces. These systems give scattered, inconsistent snapshots of the battlespace. Today, only a small fraction of the data gathered in military operations is actually used.

An explosion in the amount of information that can be made available to operational commanders demands new and more innovative approaches to information collection, management, and dissemination. The information exchange model for DoD C2 systems is migrating away from dedicated point-to-point and broadcast systems (information push) toward a model based partly on Internet and World Wide Web technologies (publish and subscribe). The USAF Scientific Advisory Board (SAB) has created a visionary combat information management concept called the Joint Battlespace Infosphere (JBI) in response to this movement.

A. PURPOSE OF THESIS

The purpose of this thesis is to identify and evaluate emerging Internet/Web-based technologies that could be employed by the DoD to improve upon existing information exchange services. This survey will examine the strengths and weaknesses of technologies such as client-server architectures, search engines, middleware,

intelligent software agents, and multicast delivery tools that could enhance the development of the JBI.

B. THESIS ORGANIZATION

Chapter I provided a basic overview of the concepts that are the background for this thesis. It includes a short description of *Joint Vision 2010*, the objectives of C2 systems, current shortfalls of C2 systems, and the future requirements of such systems. Chapter II gives a brief description of the Joint Battlespace Infosphere, its characteristics, functions, and a detailed definition of its publish and subscribe architecture. Chapter III provides a description and analysis of the different Internet/Web-based tools previously mentioned that could be incorporated into the development of the JBI. Chapter IV addresses the issue of computer network security. It describes the scope of the problem within the DoD, the different types of threats to networks, what types of security controls exist, what are the security requirements for a C2 network, and what types of systems exist to increase the security of a network. Finally, Chapter V provides the conclusions and recommendations of this thesis in terms of the development and acquisition of the JBI.

I. BACKGROUND

This chapter provides the background and lays the foundation for this thesis by introducing *Joint Vision 2010*, the objectives of command and control systems, Network-Centric Warfare, and shortfalls and future requirements for C2 systems.

A. JOINT VISION (JV) 2010

JV 2010 is a vision that creates a framework for shaping military service programs and capabilities to meet future needs. It defines the four operational concepts of Precision Engagement, Dominant Maneuver, Focused Logistics, and Full Dimensional Protection, which combine to ensure that U.S. forces have the capability to dominate an opponent across the full range of military operations. This capability is known as full spectrum dominance. [Ref. 1]

The achievement of full spectrum dominance requires information superiority, which is the capability to collect, process, analyze, and disseminate information while denying an adversary the ability to do the same. Achievement of information superiority is absolutely essential in today's military operations. If information superiority is not achieved as a precondition, the operational concepts on which *JV 2010* is built are not achievable. [Ref. 1]

Today, information is gathered by the different services using a number of different assets. For the most part, these systems are not integrated to form a coherent view of the battlespace. In order to achieve information superiority, the U.S. must develop systems that fuse information from different assets to provide the right information to the right people at the right time. Future military operations will require

the application of new and innovative information technologies to enable decisions to be made and executed rapidly.

A notion that supports *JV 2010* is the C4I for the Warrior (C4IFTW) concept. C4IFTW sets forth a 21st century vision of a global information infrastructure consisting of a web of computer controlled telecommunication grids that transcends industry, media, government, military, and other non-government entities as well. C4IFTW provides a unifying theme, guiding principles, and milestones for achieving global C4I joint interoperability that is responsive, reliable, secure, affordable, and allows the warfighter to perform any mission. [Ref. 2]

The goal of the C4IFTW vision is to evolve an open systems architecture referred to as the global grid. This global grid is represented in Figure 1. It provides instantaneous connectivity to the warfighter. This type of architecture can support both vertical and horizontal information flow. [Ref. 2]

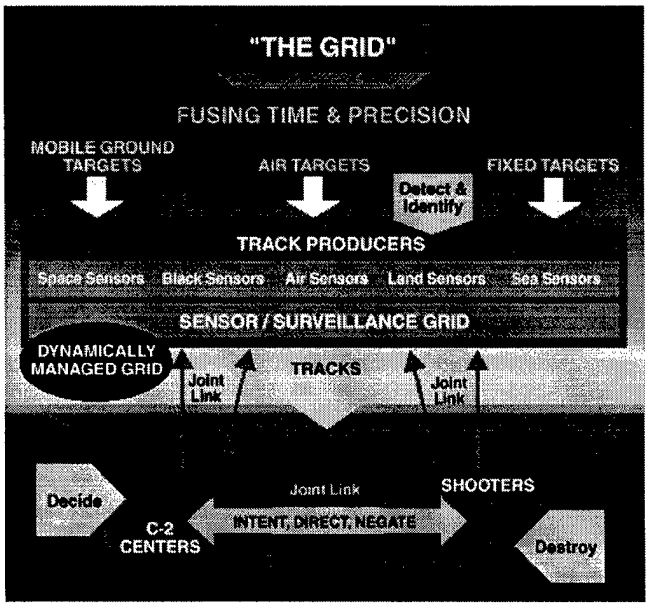

Figure 1. Global Information Grid From Ref. [2]

B. OBJECTIVES OF C2 SYSTEMS

There are four fundamental objectives of C2 systems, which are described in the following subsections.

1. Produce Unity of Effort

C2 systems should help a military force to coordinate and fuse the ideas of multiple commanders and warfighters. This allows the views of many experts to be brought to bear on a given task. [Ref. 2]

2. Exploit Total Force Capabilities

C2 systems must help people to form accurate perceptions, which lead to correct decisions and actions. These systems must be immediately responsive, simple, and easily understandable. [Ref. 2]

3. Properly Position Critical Information

C2 systems must be able to respond quickly to requests for information and deliver the minimum information where it is needed. [Ref. 2]

4. Information Fusion

The ultimate goal of C2 systems is the management of military forces and information in order to produce a picture of the battlespace that is comprehensive, accurate, and meets the needs of warfighters. This goal is achieved by fusing information and putting it in a form which is immediately useful. With concise, accurate, timely, and relevant information, unity of effort is improved and uncertainty is reduced, enabling the force as a whole to exploit opportunities. [Ref. 2]

C. NETWORK-CENTRIC WARFARE (NCW)

One of the challenges of *JV 2010* is understanding how information superiority can be exploited to devise an operational architecture that progresses beyond the weapon/platform-centric systems of today. This architecture needs to be able to coordinate the capabilities of sensors, command and control, and shooters.

Network-centric systems gain their operational advantage by integrating existing planning and warfighting systems, and interconnecting such systems via a communications network. These systems are function specific because there are specific systems devoted to planning, fusion, execution, and combat support. Improved communications and the ability to coordinate and self-synchronize operations come about through a common network capability. [Ref. 3]

Network centricity provides the electronic connections between different functional systems and allows the operators of these systems to share information more rapidly than ever before. NCW exploits these connections between systems operating in a combat environment. However, current network-centric systems fall short of the ultimate goal of full-spectrum battlespace awareness. They allow for only a snapshot view of the battlespace as seen by each existing functional system. The interaction of network-centric components requires human intervention. Existing systems cannot be integrated directly and do not provide for the direct sharing of information. Although it is essential that function-specific systems be able to communicate with each other, the U.S. needs an over-all architecture that allows which intelligent and comprehensive data transformation, information exchange, knowledge sharing, and processing. [Ref. 3]

D. SHORTFALLS OF CURRENT C2 SYSTEMS

The amount of information available to commanders has increased dramatically in recent years. This increase in the quantity of information has not necessarily led to improved situational awareness and better decision-making. The complexity of the modern battlefield is such that problems of interoperability, correlation, fusion, and overload are replacing the old problem of insufficient information. Information is so important in operations today that commanders treat it as a weapon. [Ref. 4] As a result, forces today are growing increasingly dependent on command, control, and communications. This can be viewed as a vulnerability which may be exploited by an adversary. Command and control warfare (C2W) seeks to deny the adversary the ability to command force disposition and employment while protecting its own force from similar efforts [Ref. 2].

Current military information systems need to be reoriented to be able to better accommodate time critical information such as information on an imminent attack, targeting information, combat identification information, etc. This type of information needs to be distributed in real-time to everyone in the mission who has a need for it. Current networks do not meet the speed or distribution requirements for time critical information.

Today's information systems are designed to accomplish singular functions. Planning systems, command and control systems, and execution systems are all designed to perform very specific functions. It is difficult to share information between them. Human intervention is involved for such things as file transfers. This induces a time delay for the other systems receiving the information. Network-Centric Warfare will

provide for the communication between these "stovepiped" systems. Again, this requires human intervention because NCW does not provide for direct sharing of information or the integration of existing systems.

There is no argument that current systems provide a great deal of data to the warfighter. However, these systems are disjointed as a result of not being able to interoperate, leading to an overflow of redundant and possibly conflicting data. This results in a great deal of needless effort expended on trying to organize and correlate the data of the different legacy systems. Therefore, the attention of the warfighters involved in this effort is diverted from the operation itself to the task of deconflicting the data [Ref. 3].

These "stovepiped" systems make it difficult to build a common operational picture. There is very little interoperability among the services and even less with coalition forces. As can be seen in Figure 2, today's systems give a scattered, inconsistent snapshot of the battlespace.

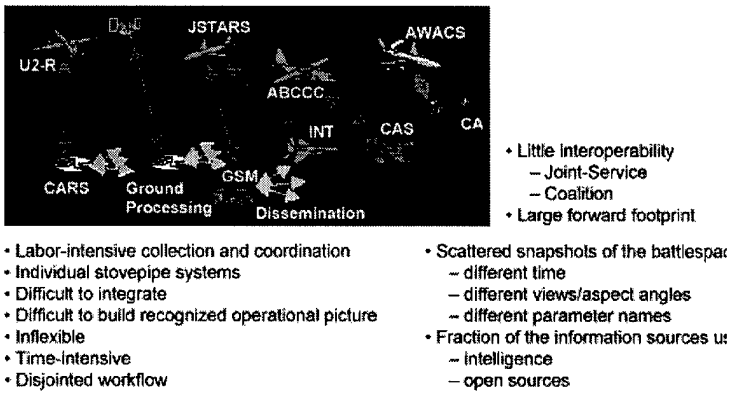

Figure 2. Data Overload and Information Starvation From Ref. [3]

These systems collect an enormous amount of data, a small amount of which is actually used for making command decisions. This situation results in both data overload and at the same time, information starvation, because the users cannot find what they need in the vast amount of available information. [Ref. 3]

Historically, the problems with current C2 systems arose, in large part, from the acquisition process. The military services traditionally have been organized around individual system programs. Each one of these systems has been designed and developed to perform a specific function as a closed system, with little effort devoted to how it will operate with other systems. The result, as previously stated, is a collection of "stovepiped" systems that are difficult to integrate which leads to little interoperability in joint and coalition environments. During operations, many different independent systems must be deployed and supported.

E. FUTURE REQUIREMENTS FOR C2 SYSTEMS

1. Requirements

As military operations become more dependent on interoperability, information processing, and connectivity, the potential for mission failure becomes more dependent on the reliability and availability of these capabilities. With the ever-increasing complexity of technological change and connectivity, new threats are continuously appearing. In fact, the number, type, and variation of these threats are quite overwhelming. Increased dependence on information technology inevitably leads to an increase in complexity, uncertainty, and unpredictable consequences. These demand

organizational and procedural changes in order for the U.S. to maintain its position as the most technologically advanced and capable military in the world. [Ref. 5]

In the future, commanders and warfighters must have the means to maximize the warfighting potential of information at a moment's notice. During the Cold War, the U.S. military had the luxury of focusing on a single major threat with only a few excursions. A forward-deployed U.S. posture in a predominantly bipolar world allowed the U.S. to statically plan and exercise for predictable crises. With the end of the Cold War and the emergence of competing national interests in a multi-polar world, U.S. warfighting requirements have become more expeditionary in nature. The new environment places demands on commanders at all levels to organize, deploy, and employ tailored forces to meet a wide variety of threats and situations, from counter-terrorism to major theaters of war. To meet this need, U.S. commanders and warfighters need a variety of capabilities such as total situational awareness, the ability to quickly develop appropriate responses to crises, agile combat capability to execute expeditionary missions, etc. As the world moves into the new millennium, commanders must be able to tailor combat power to the tasks at hand. [Ref. 4]

The warfighter of the 21st century needs an information management and exchange capability that supports tailorable, dynamic, and timely access to all required information. This will require a new capability for routing data and information to meet future battlespace requirements. This new capability stems from three essential prerequisites. First, the user must be pre-eminent in defining what information is needed. Second, efficient bandwidth utilization requires an information transmission and retrieval scheme that only transmits information that the user specifically needs or requests.

Third, the identification, dissemination instructions, and retrieval options must be sufficiently flexible to meet the user's rapidly changing mission requirements. [Ref. 5]

Meeting these prerequisites requires a battlespace network that can handle and process information based on the needs of a wide range of users. This network must be able to provide information based on the user's needs and in a format that is compatible with the user. The user must be able to quickly and easily instruct the network with respect to what information is needed and have the ability to change his/her requirements for information as the mission unfolds.

This battlespace network must maximize effective information exchange among multi-service and international systems. Interoperability of systems is needed for the successful achievement of timely and accurate information exchange among the warfighters to maximize information sharing and ensure information superiority.

2. C2 Architecture

C2 systems must be able to meet the demands of tomorrow's military forces and provide the necessary information where it is needed in time to be used effectively. The fundamental objective of C2 systems is to get the critical and relevant information to the right place in time to allow forces to seize an opportunity and meet the objectives across the range of military operations [Ref. 2]. Also, these systems must be able to support the instantaneous flow of information both vertically and horizontally. They must provide the rapid, reliable, and secure flow and processing of information to ensure continuous information exchange throughout the force.

Achieving information superiority in the information age requires sophisticated technology as well as continuous assessment of C2 organizational processes and procedures. Powerful computers are needed, communication paths with wider bandwidths are required to support the increase in computer power, and automated management systems are needed to transfer information, perform security functions, etc. These networks must be able to move enormous amounts of information to the warfighters at high data rates. Real-time dissemination of information such as high resolution imagery and video teleconferencing capabilities have contributed to the dramatic increase the decision cycle of military commanders and have become an important part of successful military operations.

One of the challenges of the future is to create an adaptive information architecture to provide decision makers and operators with superior battlespace awareness by consistently supplying the right information in sufficient detail and in enough time to make the best decisions at all levels of command [Ref. 6]. This will require a network of multispectral sensors, advanced automated processors, analysis and correlation tools, and distributed database storage capabilities. These systems must be integrated in order to process large amounts of data and provide the right information to the users.

The future C2 architecture is envisioned to consist of thousands of distributed nodes performing the full range of collection, data fusion, analysis, and command functions, all linked through a robust networking system. It is a standards-based architecture allowing modular upgrades without massive redesign. This architecture collects raw data, organizes it into useable information, analyzes and assimilates it, and

imparts it in a form that enhances the military decision maker's understanding of the battlespace. Figure 3 gives of functional depiction of this architecture. [Ref. 6]

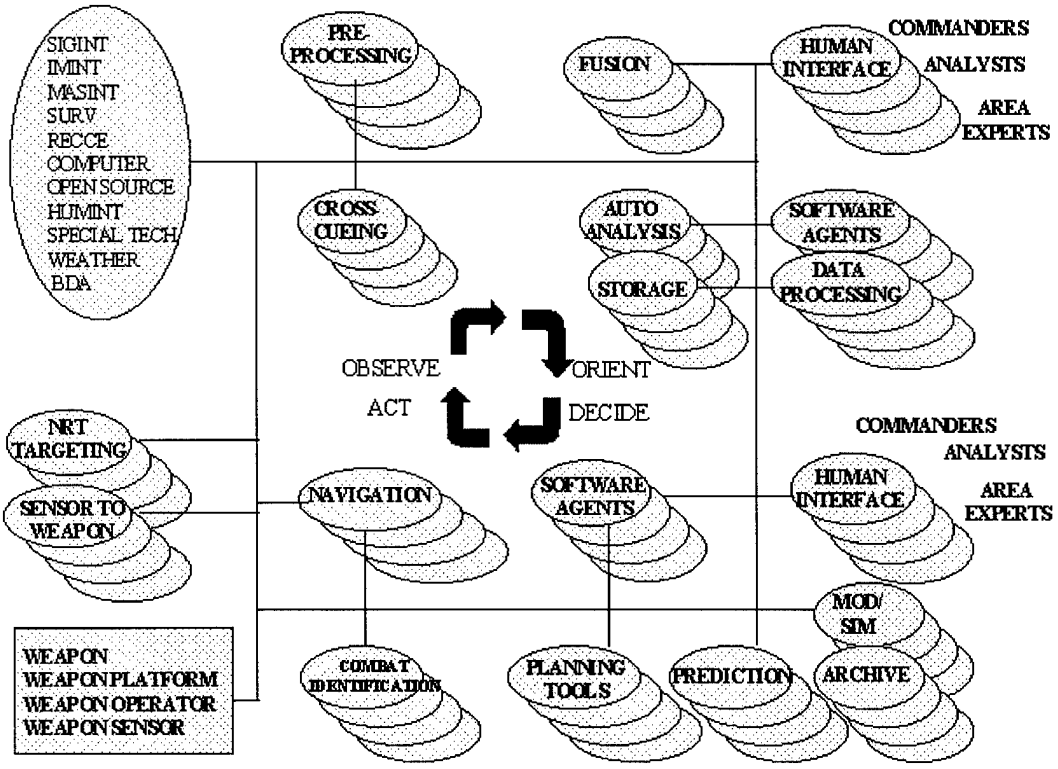

Figure 3. Command and Control From Ref. [6]

This global information grid can connect the stakeholders in a mission to all different kinds of information. A network that allows complete access to information, and puts that information where it is needed requires sophisticated technology. Advanced telecommunication networks are required to gather data, fuse the data to create useful information, correlate that information, and properly present it to the user. These systems must employ advanced parallel and distributive processing techniques and be compatible across all users. [Ref. 7]

One of the most difficult challenges in achieving these objectives will be the management of the information transmission spectrum. This challenge is due to the increased need in high bandwidth information products such as high-resolution imagery, and the increasing pressure to make reserved bandwidth available to the commercial world. Therefore, effective bandwidth utilization will become increasingly more important.

Information superiority requires the capability to collect, process, and disseminate an uninterrupted flow of information. New protocols will be required that permit data entry, sharing, and forwarding across all communication media. Effective information dissemination management is essential to providing the right information to the right place at the right time over the right communications path. It will require a new architecture to manage the routing of information that is more complex than those of today's systems. [Ref. 5]

With new systems constantly being added to the battlespace, legacy systems will be required to interconnect with these systems. During the development of new systems, plans must be made to ensure that information exchange requirements with legacy systems will be met. This may require modifying legacy systems to make them more compatible with new systems and making new systems backward compatible with certain essential, legacy systems.

To summarize, military operations in the present and future require C2 systems that are reliable, flexible, redundant, secure, and can meet the warfighter's needs to execute their missions. Future C2 systems will be designed to be interoperable and meet the requirements of *JV 2010*. These systems will need to be backward compatible with

legacy systems to provide interconnectivity until the legacy systems are phased out with the migration towards the vision of *JV 2010*. The lack of standard data formats, common structured representations, ontologies, and schemas in legacy systems has made it difficult to achieve information interoperability. Standards must be put in place reduce the diversity in ways to store, process, and present data that exists today.

THIS PAGE INTENTIONALLY LEFT BLANK

II. THE JOINT BATTLESPACE INFOSPHERE (JBI)

Chapter I laid the foundation for this thesis by providing background and historical information on C2 systems, the current problems with these systems, and the types of functions these systems will be required to provide commanders in the future. This chapter introduces a new DoD concept for managing the exchange of information, called the Joint Battlespace Infosphere.

A. BACKGROUND

To maintain information dominance, a commander must have a flexible and robust C2 system. To effectively plan an operation and make the right decisions, a timely and accurate exchange of information is required between each echelon of command and the warfighters. Future operations will require a globally distributed command and control structure that provides battlespace data from a vast array of sensors to the commanders and warfighters who need that data to enable rapid decision making and execution. The USAF SAB has created a visionary combat information management concept called the Joint Battlespace Infosphere as a means to develop this command and control architecture.

B. JBI DEFINITION

The JBI is the next step in the evolution from system-centric through network-centric to information-centric. Network-Centric Warfare is a first step towards forming a common view of the battlespace. Network-centric systems connect different functional platforms through a communications network. The JBI extends the capabilities of these

systems for intelligent data transformation, information exchange, and knowledge sharing. [Ref. 3]

The JBI is defined as a globally organized and integrated information system of systems that is designed to carry out the commander's intent. It is a dynamic, distributed, real-time system that provides database and communication services. It provides up-to-date information to all stakeholders associated with a military operation. The people involved in an operation will use the JBI to input, manipulate, and extract information pertinent to the mission. [Ref. 3]

The JBI is built on four key concepts. These are: [Ref. 4]

1. The exchange of information using a publish and subscribe architecture.
2. The transformation of data into knowledge.
3. Distributed collaboration using shared information objects.
4. Assigned unit incorporation through the use of force software templates

The JBI integrates the five essential elements of military operations, which include command, planning, execution, combat support, and information support. Each of these functions will interact with and be part of the JBI, while at the same time maintaining their own unique required actions. As a result, the JBI will serve as an integrating system. Figure 4 on the next page shows the relationships between these systems and the JBI. [Ref. 3]

The JBI provides a repository for information for everyone involved in the mission. It is intended to be a single place to which anyone contributing to the accomplishment of the mission can go to receive the information they need. These JBI users include weather, intelligence, logistics, planning, and operational staffs. This

system integrates all different kinds of data, automatically manipulates that data, and produces tailored information to support decision making within the operation. [Ref. 3]

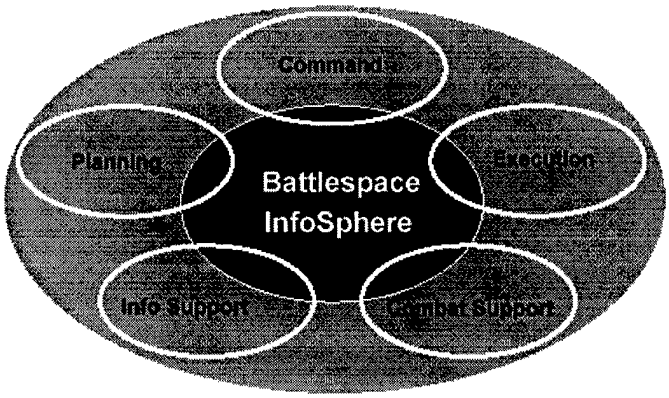

Figure 4. Relation of JBI to Military Operations Elements From Ref. [3]

Data and information are stored in the JBI as objects. These are not necessarily the same as the objects associated with object-oriented programming languages such as Java. They are attribute-value pairs which contain the represented information and metadata describing the information. Metadata is data that describes the structure of the data element itself, and must also conform to standard definitions, and is used to respond to queries and to determine whether an object should be passed to a subscriber. Objects are input into the JBI from various sources and made available for manipulation. Objects can also interact with entities outside the JBI, such as people, legacy systems, and external databases. These objects must conform to standard definitions. This is necessary so all the interconnecting computer systems and databases interpret each object in the same way. [Ref. 3, 4]

An object's tag, which contains its metadata, is stored within the JBI. The contents of the object are not necessarily stored in the JBI, especially if it is a very large

object. For large objects such as satellite imagery, the contents of the object will be retained by the original source from which the object was published. The JBI would retain a link, similar to a World Wide Web Uniform Resource Locator (URL). The link would used to get the object from the original source.

To summarize, the JBI is an enterprise information infrastructure that enables a new level of seamless information collection, aggregation, integration, and dissemination. It facilitates the rapid, functionally tailored, and decision-centric exchange of real-time information among all users to insure information dominance, mitigate uncertainty, and foster mission success. Vital components of this infrastructure include: advanced publish/subscribe/query models for information sharing; a virtual object space in which data objects are continuously created, enhanced, and propagated to consumers; sophisticated metadata services to discover, broker, and deliver precision-targeted information; readily accessible virtual repositories and digital libraries cataloging data elements; intelligent, knowledge-based control mechanisms to manage information flow; and multi-level security services. These features will be briefly described in the sections to follow. Security issues are covered in Chapter IV. [Ref. 8]

C. JBI CHARACTERISTICS

The JBI is envisioned to be an evolutionary system of systems that can grow to accommodate changing operational requirements. Capabilities and functionality will be gradually implemented within the JBI. It will be incrementally built by the introduction of new interoperable systems and the use of wrapper technology to gradually incorporate function specific, legacy systems into the JBI. It will make full utilization of the

commercial technology base and will fully exploit evolving technologies as they become available. The development of this infrastructure will rely heavily on existing and emerging commercial information technologies. This information management infrastructure will allow commanders in an area of responsibility (AOR) to build and modify a tailored JBI to meet their changing operational needs.

The technology requirements imposed by the JBI come from the need to input, manipulate, and interact with information. Many different types of technology are available to build the JBI. Wrapper technology is a technique to make information stored in legacy "stovepiped" systems available to other systems. Extraction technologies will obtain relevant information for the user. Data from different sources needs to be fused in order to extract the right information from raw data. The data will be fused for subsequent processing within the JBI using available fusion technologies. Storage of information will take place across a distributed architecture so the information will reside in numerous locations. This is approach is similar to how the Internet stores information. This will help to prevent the entire network from going down when a node experiences physical destruction, computer network attack, power failures, or other outages. The JBI must make information available to those who need it, and, at the same time, it must prevent information from being available to those who should not have it. This operational need requires the implementation of a multilevel security capability. Information must be layered so that wide dissemination of the information is possible, but access to the information is only granted to authorized users. These are just a few of the many examples of tools and information technologies that exist to help implement the JBI. [Ref. 4]

Interaction with the JBI involves communication within the JBI and techniques for presenting information in ways that the user can easily recognize. Automatic formatting and filtering of information provides ways to reduce the information overload problem and make relevant information easy to understand. Interactions between users and the JBI will take many sophisticated forms, but the need for information will be based on use-driven events, resulting in a dynamic flow of information. [Ref. 3]

The JBI routes information to the warfighters based on the specific needs of each user via the subscription and query mechanisms or at the commander's direction. Information objects can be sent automatically to people who have a need for the information. A user can also pull the information from the JBI using search techniques or they can browse for information objects that are not normally automatically routed to them because they do not have an explicit need to be continually updated with such information.

There are numerous benefits to the creation of this type of integrating infrastructure. It will provide the DoD with an affordable, secure, reliable, flexible, distributed, and interoperable information management capability. It can be rapidly tailored to meet the needs of theater commanders. The JBI will produce an enabling framework that is based upon the latest and best commercial practices in information management. It will incorporate added security and assurance features to meet the specific military requirements. It will simplify information management processes and the operation of the C2 architecture. Finally, it will integrate multiple sources of data, enable the automated manipulation of that data, provide rapid response times, and produce tailored information to the warfighter. [Ref. 8]

D. JBI FUNCTIONS

The functions within the JBI will fall into three main categories: input, manipulation, and interaction. Information must get into the JBI, it must be manipulated to produce knowledge, and users, which include humans and other software clients, must be able to interact with the results from the manipulation. These functions are shown in Figure 5. Each of these different functions will be described in the following subsections. [Ref. 4]

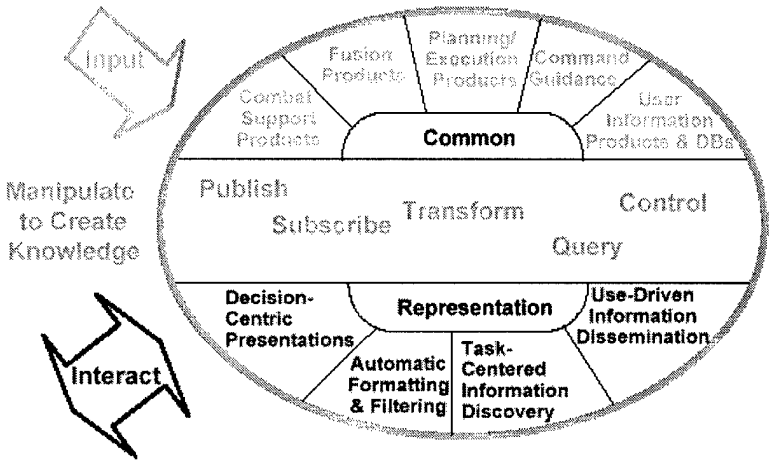

Figure 5. Components of the JBI From Ref. [4]

1. The Input Process

In order to be useful, information must be available to those who need it. Information will enter the JBI from a number of different sensors and other sources. As can be seen in Figure 5, these sources include combat support products, fusion products, planning/execution products, command guidance, and user information products and databases. Examples of these types of sources include logistics systems, imagery and intelligence data, commander's input, maps, weather reports, news feeds, etc. [Ref. 4]

2. Manipulation of Data

The core features of the JBI are based on the concept of manipulating data to create knowledge. This concept is composed of the ideas of publishing objects in the JBI so that they can be shared with others, subscribing to objects to be instantly provided with the most up-to-date information, transforming objects into new objects to create knowledge, conducting queries to find information within the JBI, and controlling the operation of the JBI to ensure it is correct and robust. Section E of this chapter is devoted to the concept of a publish and subscribe architecture and its associated functions. [Ref. 4]

3. The Interaction Process

Figure 5 shows that people and systems interact with the JBI in different ways. One way people interact with the JBI is through presentations geared toward the user. Displays can be made specific to the individual or to the types of decisions being made. Objects within the JBI are formatted so they are in the proper configuration for each user. Also, the information is automatically filtered based on the requirements of each user. The commander's intent and users' needs determine what information is disseminated in this architecture. The JBI forwards only the pieces of information that are needed by the users rather than broadcasting large amounts of information which requires searching by the user to find what he/she needs. This reduces the information overload experienced by users in a push/pull system. [Ref. 4]

E. PUBLISH & SUBSCRIBE ARCHITECTURE

The core features of the JBI are based on the concept of manipulating data to create knowledge. Again, objects within the JBI are manipulated through five actions: publish, subscribe, transform, query, and control. This section and the next focuses on the publish and subscribe functions. The last section will describe the remaining functions.

1. Definition

Information management is moving away from the concept of pushing and pulling information and moving towards the concept of presenting use-driven information to the warfighter. This is information demanded instantaneously during the conduct of the mission. In this type of scenario, the information dissemination will automatically occur as the mission progresses.

a. Publishing objects

One of the most important parts of the JBI is the concept of publish and subscribe. When new information is collected or becomes available, the object is published in the JBI repository. The object then becomes instantly available to users that access the repository. Therefore, this type of architecture forms the communication link between the providers of information and those that seek it. Objects are automatically routed to information processing functions based on the needs of those functions. This could lead to the publishing of new objects in the JBI. In this way, battlespace information is processed through automatic information flows. [Ref. 3]

b. Subscribing to Objects

A subscription is somewhat similar to a database query or search engine in that it specifies the kind of data the user seeks. Subscriptions are a standing request for objects whose metadata values match those specified in the subscription. Users subscribe to objects by specifying the properties of the information they need. When an information object is published in the JBI, those people who subscribe to the published information are immediately notified of the published object. Users are also allowed to change and update their subscriptions at any time, making this information exchange architecture use-driven. [Ref. 4]

2. Characteristics

The publish and subscribe mechanisms are the key to the JBI. They provide the means for communication amongst systems and people. These types of transactions can happen very fast in order to create real-time linkage and provide information from sensor to shooter. When a new object is published, the metadata is maintained in a catalog along with all other published objects. A catalog of all of the subscriptions is maintained as well. When a new object is published, it is matched against the list of subscriptions so that the appropriate subscriptions are triggered by the new object. This catalog will most likely be distributed across more than one server for purposes of performance optimization and redundancy in case part of the network goes down. [Ref. 3]

Once an object is defined, the publisher of the object can update the object on a continuing basis. Publishing change data in the JBI ensures that those who have subscribed to that object are continuously updated. Users subscribe to the objects they

need to carry out their responsibilities within an operation. They determine what specific information they need at the desired level of detail by subscribing to the object that provides that level of detail. Information that is posted within the JBI is instantly shared with subscribers. Authorized users may subscribe to as many objects as they need to carry out their mission. Users can also search for information using browsers and queries, but the need to search for information should be significantly reduced. [Ref. 3]

3. Associated Functions

This section describes the remaining actions within the area of manipulating data to create knowledge, which includes transformation of information, querying for information objects, and managing control of JBI functions.

a. Transformation of Information

The JBI performs a number of different information processing activities on data objects such as aggregation, filtering, and data fusion to create information. The USAF SAB calls these types of processes fuselets. Fuselets are computer programs that fuse data from several sources into information. They enter subscriptions and collect the information they need. Whenever a new object is published that matches a subscription, the fuselet process is triggered and executed. If the fuselet performs any processing on the data object, it publishes a new object in the JBI. This is depicted in Figure 6. [Ref. 3]

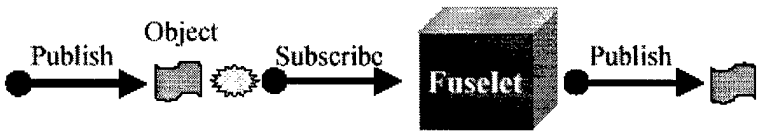

Figure 6. Information Flow Using Fuselets From Ref. [3]

Fuselets have many roles within the JBI. They can act as an interface to allow wrapped legacy code to interoperate with the JBI. They can use information gateways to take data from public and commercial sources and publish it in the JBI. Fuselets work with browsers to display objects that each user has subscribed to. The fuselet detects when an object has changed and the browser display is automatically updated. Figure 7 gives an example of how fuselets can be used to aggregate data from different military bases involved in an operation into one base status report.

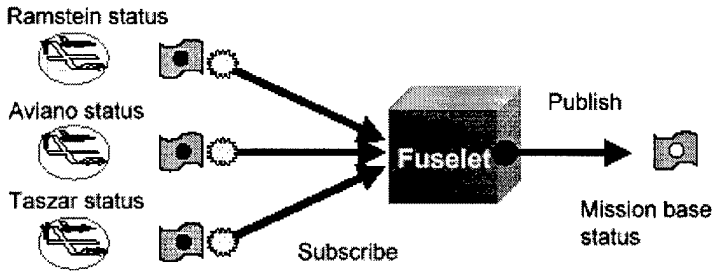

Figure 7. Fuselet Determines Base Mission Status From Ref. [F]

b. Query for Objects

The JBI can be searched for information in a manner similar to how the Internet or a database is searched. Query operations allow authorized users who do not subscribe to a particular object to access the object as necessary for the mission. These users may not need a subscription to that information object because they do not need to be updated continuously with the most current modifications. Alternatively, they may need to view certain objects only on a periodic basis to fulfill their mission.

c. Control of JBI Functions

The JBI needs a set of management and control functions to keep it operating correctly and efficiently. Aspects of the system requiring management and control include such areas as bandwidth allocation, quality of service, performance monitoring, data management, security, and configuration. [Ref. 3]

As an example to illustrate the importance of data management, separate channels must be used in the JBI to carry different types of traffic. When an object is published, a fuselet will be triggered to determine which channels to use for publishing and subscribing. Sensor-to-shooter loops will use channels that are devoted to time critical information. Material and supply information, on the other hand, can use a channel with a slower response time.

An important part of operating the JBI is the ability to control the flow of information. Critically time sensitive information must get to its recipients in time to be useful. Within the JBI, information of this nature will be tagged with a higher priority than other items, which will move more slowly within the system.

Monitoring the performance and maintaining the smooth running of a large distributed network is a difficult task. Objects that have many subscribers can create bottlenecks in the network. Since the network is a dynamic entity, it is difficult to predict when and where these bottlenecks will occur. It is possible to keep such objects outside the JBI repository and provide pointers to the location of those objects. The ability to visualize, analyze, and repair flow problems is a crucial requirement of the JBI. The result will be a network that is more efficient. [Ref. 3]

THIS PAGE INTENTIONALLY LEFT BLANK

III. INTERNET/WEB ENABLING TOOLS

The previous chapter described the Joint Battlespace Infosphere. It gave a definition of the JBI and described its components, functions, characteristics, and benefits. It discussed the heart of the JBI, the publish and subscribe architecture, which is the driving force for this thesis. This chapter will focus on the different types of World Wide Web and Internet based tools that can be used to create the architecture described in the last chapter. This chapter will describe each tool and assess its strengths and weaknesses.

A. BACKGROUND

Chapter I discussed the problems with current DoD C2 systems. To reiterate, these systems provide much information to today's combatants, but because these systems are disjointed, there exists an overload of poorly organized and incomplete information during operations. These "stovepiped" systems tend to be inflexible, not very time responsive, and difficult to integrate and use in building a common operational picture. The result is limited interoperability among the services and even less with coalition forces. Today's systems give a scattered, inconsistent snapshot of the battlespace.

Today's sensors are able to gather vast quantities of data, only a fraction of which is actually used. This situation leads to data overload and at the same time information starvation because users cannot find what they need in the available data. The consequence is that commanders and warfighters lack a common understanding of the

battlespace and a great deal of effort is spent deconflicting the data being presented instead of focusing on the mission.

An explosion in the amount of information that can be made available to operational commanders demands a new and more innovative approach to information collection, management, and dissemination. Thus, the USAF SAB has created the visionary combat information management concept called the Joint Battlespace Infosphere as a means to develop this command and control architecture. Within this type of information exchange model, C2 systems are migrating away from dedicated point-to-point and broadcast systems, toward a model based partly on Internet technologies to create a publish and subscribe architecture. The many forms of information storage and networked communications among the elements of the JBI will require exploitation of modern Internet technologies. There are numerous Internet technologies that can be employed in this type of architecture. This chapter will evaluate the strengths and weaknesses of the use of an Internet-like client-server architecture, search engines, middleware, intelligent agents, and multicast delivery tools that could be employed by the DoD to help create the Joint Battlespace Infosphere.

B. WEB-BASED CLIENT-SERVER ARCHITECURE

1. Description

The development of traditional client-server computing allowed for the distribution of applications and data across a network. The server is where information that a user accesses is stored. The client is software that enables the user to access the information. Application programs and data files are installed on a networked central

server and authorized users access it remotely over the network using client software designed for this purpose. The server sends only the files the client software requests instead of the entire application program. The development of the World Wide Web provided a means by which documents could be published, linked, stored, and retrieved much easier than in a traditional client-server network [Ref. 9, 10].

2. Strengths of a Web-Based Architecture

The development of the traditional client-server architecture optimized the use of networks, but it created a tremendous amount of administrative work with respect to managing users and client software. Traditional client-server architectures tend to be proprietary and do not interoperate. Web technology has vastly simplified the retrieval of information. In the traditional architecture, one had to log into the computer containing a file; the file had to be located in the proper directory; the file was then opened, downloaded, and closed; and the remote computer was logged off. With Web technology, all of these steps were reduced to the click of a mouse. [Ref. 9]

Other benefits of using a Web-based system include the ability of an end user to tailor their interface to the Web to fit their own needs using Hypertext Markup Language (HTML). Therefore, using HTML, an organization can create a uniform and standardized format for viewing and publishing information within their own intranet. Traditional client-server application interfaces are difficult to change due to the task of updating numerous copies of the client applications, which may be spread throughout the organization. With a Web-based system, the interface resides on the server, so it can be easily manipulated to meet the users' needs. With respect to network efficiency, the

traditional model requires a live connection between the client and server. A user logs on to the server and the connection remains open until the user logs off. On a Web-based system, there is no extended connection between the client and server. The client sends a request to the server, and the server sends the requested information back to the client. The server is occupied only as long as it takes to fulfill the user's request. [Ref. 9]

Two other advantages of Web-based systems are their open architecture and non-proprietary nature. A Web server by default allows access to anyone with a browser. Security features can be activated to protect against unauthorized users, but if they are not activated, the server is open to all users. In a traditional client-server system, a user must have an account established on that server in order to gain access. This leads to a great deal of administrative work. An account must be established for each user, and for a network consisting of thousands of users, this is a tremendous task. Also, the World Wide Web uses non-proprietary standards which allows for platform independence. In a traditional client-server architecture, separate client and server software must be purchased for each individual network application ("stovepiped" applications). These applications are developed to operate on a single platform (i.e., Windows operating system or UNIX). [Ref. 9]

3. Limitations of Architecture

One of the most obvious weaknesses in a Web-based system is network security. The threat of malicious attacks and intrusions from outsiders is always a concern, but these Web-based systems give users unprecedented access to an organization's data. Therefore, access to data by people within an organization is a legitimate concern. Only

people who have a need for specific data should be allowed to access it. On the other hand, no one should be excluded from having access to the data if they have a legitimate need for it. The use of identification and authentication, passwords, encryption, and firewalls will be addressed in Chapter IV.

A Web-based network requires a set of management and control functions that identify and enforce a hierarchical data structure and an interface which allows easy location of resources and standards for controlling information updates [Ref. 9]. Implementing and maintaining these functions requires significant investment in manpower and training. Without these functions, the organization of information on the network will quickly grow out of control. Rules and standards must be put in place for the provision of data objects on the network and the maintenance of data content to ensure accuracy.

4. JBI Applicability

There is no question that the JBI will be a derivative of the client-server model. The JBI infrastructure must be standards based and will most likely take advantage of and utilize such widely adopted standards as HTML and TCP/IP. However, these standards will not be sufficient to fully realize the JBI concept due to issues such as speed, security, and quality of service. The JBI must be standards based because the infrastructure needs to be platform independent since many different people using different applications and systems will be publishing and retrieving information that ranges from simple text messages to satellite imagery. The universal nature of the World Wide Web with its

many different users and many different platforms makes it an excellent starting point for the development of the JBI architecture.

This Web-based model is ideally suited for storing information objects in a distributed fashion. These information objects can be input from many different sources and stored on more than one server. If the objects reside in more than one location, they will still be available to the users if a node within the network goes down.

This type of architecture lends itself to the many different kinds of information to be used within the JBI. The JBI will most likely contain a database which stores all the metadata of the different data objects. For small objects, such as short text messages, the entire object can be stored within the database. For large objects, such as imagery, the JBI database would retain a URL link to a networked server that stores the imagery, facilitating the need for a distributed architecture.

C. SEARCH TOOLS

1. Search Engines

Search engines are a popular type of software with the ability to search for Internet/Web resources. They use automatic indexing software to discover, harvest, and index Web pages. They provide an interface for users to perform queries and return search results, which a user follows to actual documents on the Web.

The Web uses an application called a Common Gateway Interface (CGI) to allow data to pass in both directions between a client and server, which makes it possible for someone using a client workstation to control applications which reside on a server. This feature is what enables information content searches and database queries. The interface

is a type of interactive online form. Information is typed into the form on the client machine, which is sent to the server. The server hands the information to the CGI. The CGI passes the data to another application on the server such as a database. The response will be passed from the application to the CGI, and then to the server, which will send the information to the client. [Ref. 9]

2. Robots

Search engines use robots to find resources on a network. Within the domain of software engineering robots are programs that automatically traverse the Web's hypertext structure by retrieving a document and recursively retrieving all documents referenced by the original. Robots are also known as spiders, crawlers, and worms. What distinguishes robots from other resource discovery tools is their automated nature. These tools use software agents that are programmed to traverse a network and search for new resources and include them in a database of resources. Software agents will be discussed in a later subsection. Other search tools provide the capability of searching a network, but draw on a knowledge base compiled manually. Robot-based search tools generally have three components: [Ref. 11, 12]

1. The Wanderer - Tool that traverses the Web finding and extracting new resources for entry into the knowledge base. Each robot differs in the frequency of its search and the scope it traverses.
2. The Knowledge Base - Documents traversed by the robot are compiled into an index containing locations of resources specified by a URL and some context information about the resources.
3. The Search System - Users enter their search terms and the search engine finds data for the user.

There are four basic functions common to all robots that relate to these components: [Ref. 11]

1. Finding Web pages.
2. Harvesting Web pages and building an index.
3. Searching the index with a user query.
4. Providing the user's interface.

Robots work in many different ways. What is common among them is that they maintain a stateless connection to the servers they search. This means that there is not a continuous connection between the search engine and the server. The communication consists of a discrete request from the client machine and a response from the server. Today's robots can discover Web pages for themselves and can even harvest information from the text of those pages. They can be subject specific so they search for pages containing certain keywords, or they can be site specific, looking for Web pages in certain locations. Simple parameter variations allow robots to be customized to search for almost anything. It is possible for users to employ their own personal Web spider, which hunts for whatever information is requested. [Ref. 11]

Robots are used to index the URLs of Web pages they discover. Robots treat URLs as citations to Web pages. They begin with the first hyperlink and follow that link wherever it leads for a certain number of iterations, and then return to the original document to launch forth from the next hyperlink. They index the URLs by disassembling them into component words that are stored together so users can retrieve the associated pages using keyword searches. Therefore, a database of Web pages can be created with no human intervention. Then, a search by a user would produce results in the form of a unique set of URLs connected to the original documents on their home servers. [Ref. 11]

3. URL Index

When a user queries a search engine, the keywords are compared to an index or catalog. The keywords are linked to a second database containing the titles of Web pages, which become the user's search results. These tools constitute the part of the search engine with which the user interacts.

A robot (the wanderer) searches the Web and saves addresses of sites in a URL database. A second robot, called a harvester, checks the database of URLs and begins automatically revisiting each Web page the first robot discovered. Instead of just recording the URL, the second robot extracts part of the text of the page and integrates it into a master index of words from other Web documents. This becomes the index the user searches. [Ref. 11]

Figure 8 shows how the information flows.

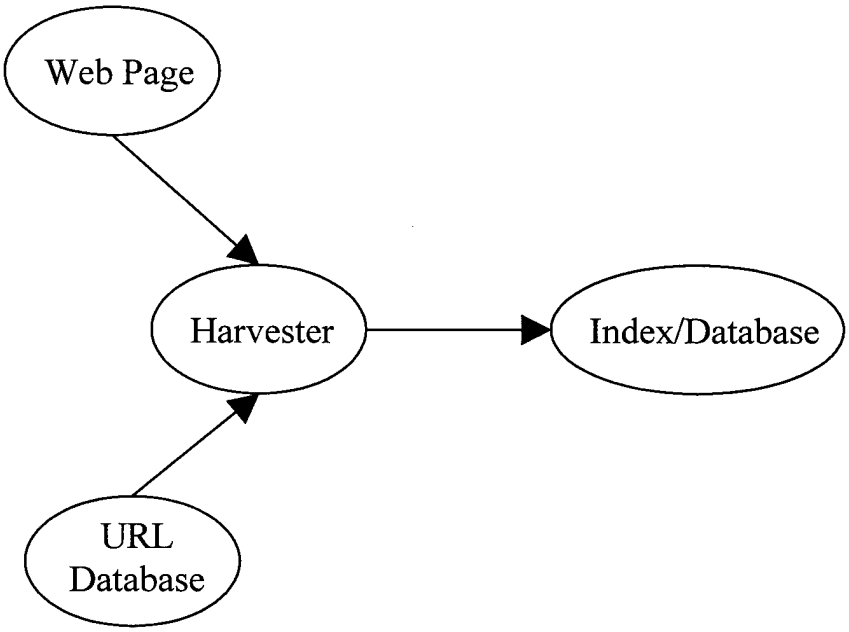

Figure 8. Information Flow Process for Robot Harvesting From Ref. [11]

When a user submits a query to a search engine, it compares the query to the index that the harvesting agents supply with Web page contents. The search results are based on the index contents, not on an actual search of the Web. This can be seen in Figure 9.

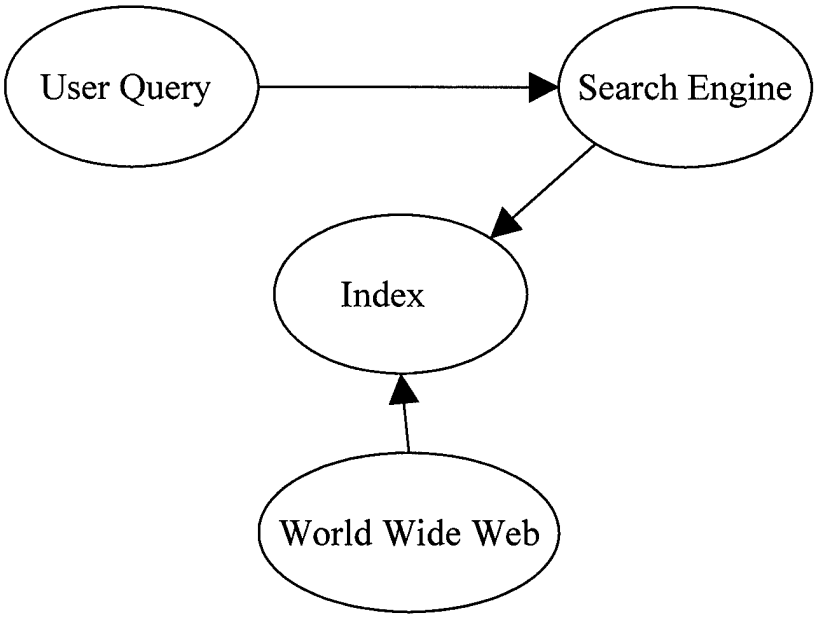

Figure 9. User Query of Index From Ref. [11]

A typical index has four fields based on a keyword search:

1. The URL of a page containing the word.
2. The word's position within the Web page.
3. The total number of words within the page.
4. The field in which the word occurs (title, URL, etc.).

Within the context of the Web, harvester robots continuously update the index, typically on a daily basis. As the harvester reads each word on the Web page, it creates an entry in the database. The text from each page is parsed into individual words. The indexing process transforms the information gleaned from the page into a set of words and their positions in the page. [Ref. 11]

When a user submits a query, the search engine matches the query to the words in the index. Once the engine finds matching words in the index, it compiles the URLs into a list and returns them to the user. The search engine breaks the user query into its component pieces and searches for the keywords. When it finds a match in the index, it pulls the record for that particular URL. It then presents the URL, title, and summary to the user.

4. Search Engine Interface

Robots, indexes, and search engines all reside and operate on the server. The interface is the part of the search tool with which the user interacts. It is the medium of communication between the user and the search engine. Typically, some form of Web browser is used as the interface. When a query is made using the browser, it prompts the search engine and translates the user's query into the proper format for the search engine. From here, the engine searches the index and reports back its findings. The interface presents the results to the user. [Ref. 12]

To initiate a search, most interfaces contain some sort of command line. Users enter keywords in the command line. The search statement is submitted to the server where the search engine transforms it and compares it to the index. Users must be aware of the appropriate syntax of the interface to a search tool in order to properly submit a search request. [Ref. 11]

The query page of the interface is a static file at a fixed address. When a search is conducted, the search tool creates a new temporary HTML document to display the results of the specific search. The results contain the URL, which links to the named

page itself, not a copy of the page stored in the search tool's database. The results of Web searches are pointers to locations that contain information. [Ref. 11]

5. Strengths of Search Tools

a. Search Engines

The most obvious benefit of having a search engine capability within the JBI is to be able to query for specific information. The JBI is envisioned to have a search capability similar to how the Internet is searched. On the Internet, when a search is conducted, the search results contain links to the appropriate pages. The pages themselves are stored on their own servers versus within the search tool's database. A search of the JBI would be analogous to the Internet search; the results would be links to the information residing on their own servers. Additionally, as mentioned in Chapter II, users sometimes do not need subscriptions to certain information objects because they have no need to be continuously updated with that particular information. However, they may need access to that information object periodically, and need the ability to query for that information.

b. Robots

Robots are an intricate part of the search engine and are a necessary tool to allow the JBI to be queried and to enable the subscription process. One of the integral features of the JBI is that once a particular information object is subscribed to, any time the object is republished with updated information, the user is automatically and instantly updated. JBI robots will operate in a manner similar to Web robots. They can be customized to search for almost anything. These robots will go out on the network and

search for information based on the specifications of subscriptions. Of course, for time-sensitive information, the robots need to traverse the network frequently, on the order of seconds, requiring multiple robots operating simultaneously. Therefore, the user can be updated in real-time with accurate information.

Robots provide other benefits to the JBI. It was stated earlier that Web robots maintain a stateless connection to the servers they search. This idea can be carried over to the JBI because one of the critical issues within this architecture is bandwidth utilization and management. Maintaining stateless connections equates to conserving bandwidth. Also, there needs to be a way to keep track of all the information posted to the JBI. Manually creating such databases is a tedious task and a waste of human resources that could be focusing on the mission. Robots are an automated means to do such large scale indexing.

c. Index

The JBI will require a repository to catalog information objects, but this repository will most likely be significantly different than the URL indexes created by Web search engines. However, the basic principles still apply. When an object is published to the JBI, the metadata and appropriate link to the information object is cataloged in a repository. A similar catalog of subscriptions exists. This is where information objects will be matched to subscriptions. In contrast to the indexes of Web pages created by search engines, which are by no means a complete listing of relevant pages on the Web, the JBI indexes will contain all of the information objects and subscriptions. Also, indexing within the JBI may be done in a completely different

manner than the Web, such as geospatial and temporal indexing. Web indexing is conducted based on keywords. The amount of information contained within the JBI is envisioned to be vast, but still manageable. These databases will still need to be distributed across multiple servers for purposes of redundancy and network recovery.

6. Limitations of Search Tools

One of the primary concerns in this type of architecture is the right people getting the right information and the ability to search and find the information you need. This means it is up to the publishers of the information objects to ensure that the metadata contains the correct information and the links to the data are accurate. The publishers need to employ descriptive URLs in order for the information objects to be properly indexed, or it would be nearly impossible for the user to find and retrieve the information.

A second issue is proper organization of the information within the JBI. The Web is a huge repository of data that is loosely organized and complex. Robots are software programs that attempt to create organization as they traverse the Web. This proves to be a very difficult task and therefore different robots have different levels of success in creating indexes. The JBI will not have the same amount of information as the World Wide Web, but nevertheless, it must be properly organized in a hierarchical fashion so the right information can be found by the users.

Lastly, users must be able to use the search tools properly so that the appropriate syntax is used when performing a search. This will most likely require some amount of training and familiarization so users are comfortable with the interface. The interface should be user friendly, but the users should still have some exposure to the system and

how it works prior to an actual operation. For example, the user should know whether the search can be conducted using natural language entries or if the search tool only accepts Boolean operators as entries.

D. MIDDLEWARE

Mission critical systems, including those of the DoD, make extensive use of distributed computing to share information over large, heterogeneous networks. One of the most difficult tasks of these systems is passing information reliably between many applications running on different computers. Network programming is the term used to define the software development process of building applications that must communicate with one another over a network. As information is passed across a network, it will go from one type of platform to another. In many instances these platforms will not have compatible formats for representing data. The network programmer must translate the different types of data so they are understood by each platform.

A typical network environment includes servers, mainframes, workstations, and laptops trying to communicate with each other over multiple network protocols, and trying to get information out to weapons platforms, such as aircraft, where the hardware component might be an avionics flight computer. The JBI is no exception. It will include multiple systems and platforms that all need to communicate with one another. Middleware is one of the tools that can help solve this difficult problem.

1. Description

Middleware is connectivity software that allows multiple processes running on one or more machines to interact across a network. It allows for applications to exchange

information regardless of the environment they are running in. Middleware is essential in providing communications across heterogeneous platforms. It provides interoperability in support of the migration to client-server architectures. [Ref. 13]

Middleware enables distributed functionality, which allows two or more application programs to operate through some intermediary application. It helps to solve the interoperability issue by providing the means for one component to communicate with another component to request some action be initiated or to relay information. [Ref. 14]

Middleware services are sets of distributed software that exist between the application and operating system, as depicted in Figure 10.

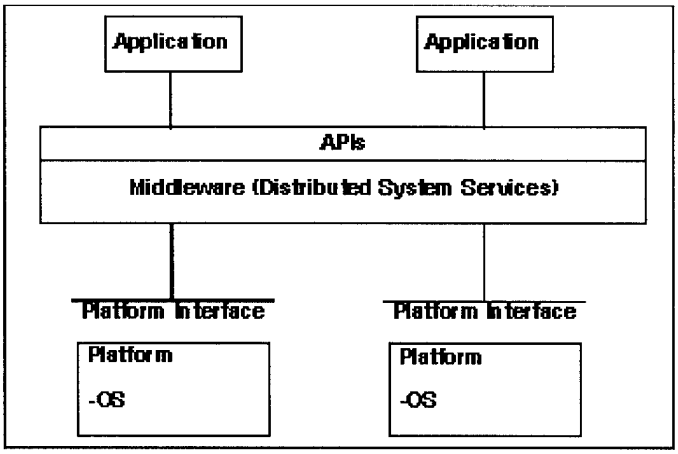

Figure 10. Use of Middleware within a Network From Ref. [13]

Middleware services provide a functional set of Application Programming Interfaces (APIs) to allow an application to interact with another application or service by locating it transparently across the network. [Ref. 13]

Middleware can be broken down into five segments that each use a different method to move information between software programs. These are: [Ref. 15]

1. Database Middleware
2. Transaction Processing Monitors (TPMs)
3. Remote Procedure Calls (RPCs)
4. Object Request Brokers (ORBs)
5. Message-Oriented Middleware (MOM)

Each of these types of middleware is described in the following subsections.

a. Database Middleware

Database middleware uses database gateways to perform its necessary functions. During operation, a client program issues a Structured Query Language (SQL) request to the gateway, which in turn ships the request to the target database. The gateway translates the request to the proper format for the target database. Database middleware requires synchronous, point-to-point communication. This approach does not work well in high performance applications because the database quickly becomes the bottleneck when many applications are trying to communicate. [Ref. 15] Within the JBI, there are certain instances where the use of this type of middleware would not be the optimum choice, such as for delivery of mission critical information. However database middleware could handle other types of information with the JBI, such as supply and logistics information.

b. Transaction Processing Monitors (TPMs)

TPMs are not used for general-purpose program-to-program communication. Instead, they provide a complete environment for transaction applications, usually involving relational databases. Using TPMs, clients invoke remote procedures that reside on servers, which also contain an SQL database engine. Procedural statements on the server execute a group of SQL statements (transactions),

which either all succeed or all fail as a unit. The applications based on transaction servers tend to be mission critical applications that require a rapid response and require tight controls over the security and integrity of the database. [Ref. 15]

c. Remote Procedure Calls (RPCs)

RPCs use a call and return method of communication similar to procedure calls used in software programming. The application components communicate with each other synchronously in a request-wait-reply manner. When an RPC function is activated, it is carried to completion and control is returned to the sender. RPCs work best with smaller, simple applications where asynchronous operation is not required and all communication is point-to-point. Due to their synchronous nature, RPCs are not a good choice to use in large, mission critical applications where high performance and high reliability are needed. [Ref. 14, 15]

d. Object Request Brokers (ORBs)

ORBs are object-oriented RPCs that allow objects to remotely invoke a function of another object. They execute distributed function communication and integration using a call and return paradigm. ORBs are designed to be used with applications that are programmed using an object-oriented approach. Object brokers unite new applications with existing applications in an object-oriented manner in which objects from one application invoke objects from other applications. An interface definition language (IDL) is used to define the interfaces between objects. Like RPCs, ORBs are generally synchronous and operate in a point-to-point manner, and therefore are not well suited for large, mission critical applications. However, from a security

perspective, an ORB can provide authentication, access control, audit, and message protection functionality. [Ref. 14, 15]

One popular ORB standard that has emerged is the Common Request Broker Architecture (CORBA), which has been incorporated into many legacy systems. This technology handles a client's request to perform some action on an object. However, the original standard left so much to the interpretation of the developer, that many CORBA compliant products do not interoperate. In addition, this standard does not function well within a dynamic environment. Middleware that depends on an IDL relies heavily on static definitions that do not change. Changes to an interface require changes to the interface definition and recompilation of the software code. Object Management Group, the creators of CORBA, have defined a dynamic invocation interface (DII) to handle this problem. However, DII is very difficult to understand and program. [Ref. 15]

e. Message-Oriented Middleware (MOM)

Message-oriented middleware provides the most flexibility to the developer of a network. It is considered process-centric because information flows between processes. MOM provides asynchronous message queuing for application to application communication, enabling ongoing processing while the message is being delivered [Ref. 4]. Applications can send, receive, and process messages with guaranteed message delivery [Ref. 4]. MOM products also include other services such as translating data, broadcasting data to multiple programs, error recovery, security, prioritization of messages, and location of resources on the network. This type of middleware is well

suited for large, mission critical, high performance applications, which makes it an excellent tool to assist in the construction of the JBI. [Ref. 15]

MOM software resides in both portions of a client-server architecture, supporting asynchronous calls between the client and server applications, as can be seen in Figure 11. Message queues provide temporary storage when the destination program is busy or not connected. The Message-Oriented Middleware Association (MOMA) was formed in the mid-1990's to promote standardization of this technology [Ref.14]. Members of this group include IBM, PeerLogic, Momentum Software, and Covia Technologies. However, there currently exists many different kinds of MOM software available.

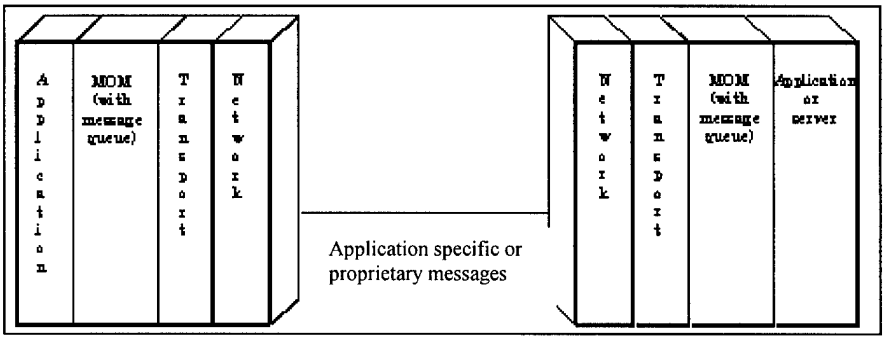

Figure 11. Message Oriented Middleware From Ref. [16]

One type of message-oriented middleware that has evolved is called message passing. It is popular for building large, distributed applications. It uses the principle of pushing information out to applications rather than forcing applications to go out and find the information they request. One model of communication used in message passing is a publish/subscribe scheme. Programs publish messages to subjects and also subscribe to subjects. Once a subject has been subscribed to by a program, the program

will receive any messages published to that subject in a distributed application. This type of middleware tool uses software agents for such things as message routing and fault tolerance. Software agents will be discussed in the next section of this chapter. [Ref. 15]

2. Strengths of Middleware Tools

The most obvious benefit of using middleware is to allow the interoperability of heterogeneous systems and the capability to retrieve information from legacy systems. The JBI needs to leverage legacy systems in order to take full advantage of the services these systems can provide without changing their software. Middleware can be used to wrap existing programming interfaces so other applications can use the information provided by these systems. Figure 12 on the next page depicts the vision of an interoperability "grid" that will allow systems to share information with each other even when different communication standards are used. In this figure, the middleware provides a set of interoperability services, allowing intercommunication between a wide range of systems and services. [Ref. 4]

The type of middleware best suited for integrating numerous applications residing on different platforms is message-oriented middleware. MOM increases the interoperability, portability, and flexibility of an application by allowing the application to be distributed over multiple, heterogeneous platforms. It increases the flexibility of an architecture by enabling applications to exchange messages with other programs without having to know what platform or processor the other application resides on within the network. It reduces the complexity of developing applications that span multiple

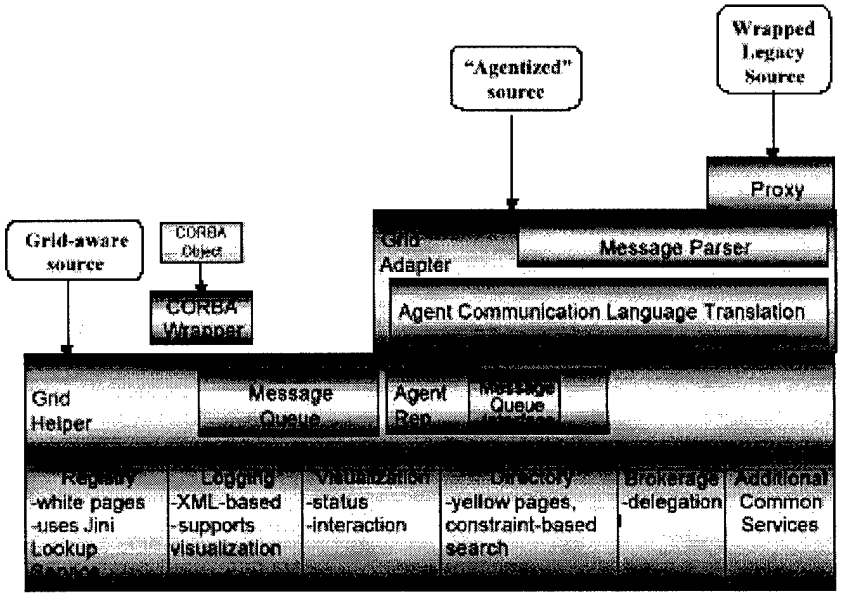

Figure 12. Interoperability Grid From Ref. [4]

operating systems and network protocols by insulating the application developer from the details of the various operating systems and network interfaces. [Ref. 16]

MOM's message passing scheme can be used to support the JBI's publish/subscribe architecture. In traditional network applications, when two processes need to communicate with one another, they need network addresses to begin communicating. If a process wants to send a message to many other processes, it needs to know the network addresses of the other processes and then create a connection to all of those processes. This type of architecture does not scale well in a dynamic network environment. The publish/subscribe communications model provides location transparency, allowing a program to send the message with a subject as the destination property while the middleware takes care of routing the message to all programs that have subscribed to that subject, creating a flexible, dynamic network. This addresses one

of the biggest challenges of the JBI, which is handling the changing, non-deterministic nature of this type of network. [Ref. 15]

Another benefit of using MOM is the fact that it can assign priorities to different messages. Messages are queued up in priority rather than time order. Therefore, messages with time critical data such as intelligence or targeting information can be assigned a higher priority and routed quicker through the JBI than less critical information such as supply replenishment. [Ref. 17]

3. Limitations of Middleware Tools

The primary purpose of middleware is to help solve many application connectivity problems. However, the development of middleware has created new problems. Many popular middleware services use proprietary implementations, meaning many product implementations are unique to the vendor. This makes network applications dependent on a single vendor's product and maintenance support for future enhancements. This reliance can have a negative effect on a system's flexibility and maintainability, portability, and interoperability. This leads to increases in costs. [Ref. 16]

Message-oriented middleware is not exempt from these problems. MOM is typically implemented as a proprietary product which means MOM implementations are usually incompatible with other MOM implementations. Using a single implementation of MOM results in a dependence on that particular vendor for services. Also, not all MOM implementations support all operating systems and protocols. The choice of a

certain MOM product depends on the application platforms and protocols supported. [Ref. 16]

Three additional issues are the number of different types of middleware products available and the necessity of having an administrator trained to maintain the software, and the security of data translated and passed using middleware. First, there are so many middleware services available today, it can become a barrier to using them. The network developer must decide in advance what type of functionality and platform coverage they need. Second, the addition of middleware software will increase the administrative and maintenance burden for a network manager in a large heterogeneous system. Third, this research did not discover any examples of middleware products that specifically address data security. This means the data must be secured by other hardware and software tools outside the realm of middleware, such as data encryption. Encryption will be discussed in Chapter IV.

Although MOM software can employ publish and subscribe mechanisms, the way current MOM software transports data is different than the concept of publish and subscribe envisioned within the JBI. Industry's middleware allows the delivery of small amounts of data in a message format across different platforms. This concept needs to be extended to allow the delivery of information objects within the JBI. The same holds true for subscriptions to information objects. However, there is widespread commercially availability of middleware that interoperates well with the Web. With such a large amount of middleware software available, the initial construction and demonstration of functionality within the JBI should be possible with Commercial Off-the-Shelf (COTS)

software. Figure 13 provides an example of the architecture of a typical commercially available middleware.

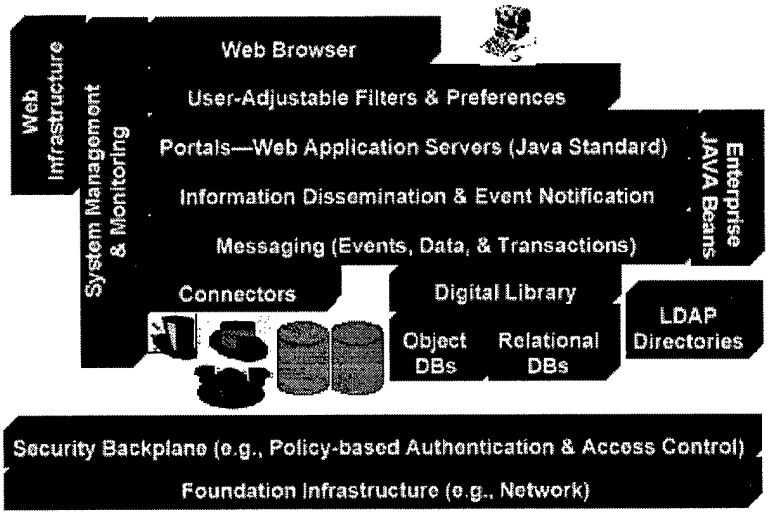

Figure 13. Architecture of Commercially Available Middleware From Ref. [4]

E. SOFTWARE AGENTS

New kinds of software are continually increasing the ability to access, manage, edit, and present information. However, many of these new systems and capabilities are actually making the provision of information more complex and increasing the workload of their users. Software technology is focusing on automating such processes using software agents. This type of technology can help the military to adapt decision-making processes quickly and cheaply to automate access to information, generate alternative courses of action, communicate ideas, and protect the information infrastructure [Ref. 18].

1. Description

Software agents are used in many different ways and there is no one single definition that everyone agrees on. One way to describe software agents is that they are software entities that function continuously and autonomously in a particular environment, often inhabited by other agents and processes. Agents that inhabit an environment with other agents must be able to communicate and cooperate with each other. Since most software agents carry out very specific functions, the term software agent can be viewed as an umbrella term that covers a wide range of other specific agent types. These agents have limited functionality by themselves, but in aggregate they can accomplish complex functions. [Ref. 19]

Software agents simplify the complexities of distributed computing. Intelligent interoperability in software systems refers to cooperation among systems to optimally achieve specified goals. Future computing environments will consist of distributed software systems running on multiple heterogeneous platforms, but many of the systems that exist today do not communicate well. Fostering this communication is one of the roles of software agents. [Ref. 19]

In order for software agents to communicate and interoperate properly, they require a common language, a common understanding of the information exchanged, and the ability to exchange information. This requires an interaction protocol, an agent communication language (ACL), and a transport protocol, respectively. The interaction protocol is the high level strategy pursued by an agent that governs its interaction with other agents. The ACL is the medium through which the content of the exchange is

communicated. The transport protocol is the actual transport mechanism used for communication. [Ref. 19]

There are numerous ways to classify software agents. One classification scheme identifies the attributes software agents possess, such as autonomy, cooperation, and learning. Autonomy is the ability of an agent to act on its own without the need for human guidance. Since different agents perform different functions, they must be able to cooperate with each other. In order to cooperate, agents must have the ability to interact with each other and humans via a communication language. Finally, as agents react and interact with their environment, they learn. [Ref. 20]

In addition to software attributes, agents can be classified their mobility, their ability to move around a network. A third way to classify agents is as either deliberative or reactive. Deliberative agents possess an internal, symbolic, reasoning model, and they engage in planning and negotiation in order to achieve coordination with other agents. They act using a stimulus/response type of behavior by responding to the present state of the environment in which they are embedded. A fourth way to classify agents is by their ability to manage large amounts of information in large networks. A fifth classification category combines the ability of two or more kinds of agents in a single entity. [Ref. 20]

Within this classification scheme, there exist seven different types of software agents. These are depicted in Figure 14 on the following page. These different types of agents will be described in the subsections to follow.

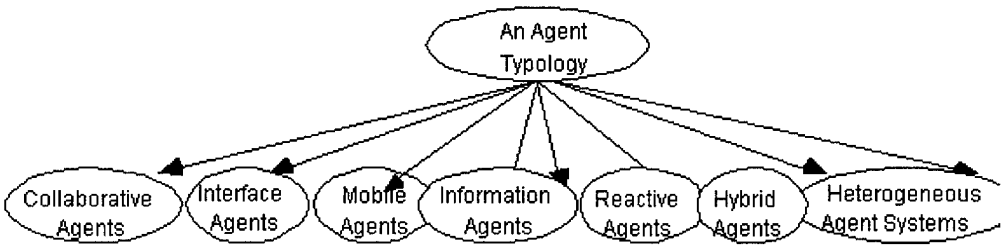

Figure 14. Classification of Software Agents From Ref. [20]

a. Collaborative Agents

Collaborative agents emphasize autonomy and cooperation with other agents in order to perform tasks for their owners. The general characteristics of these agents include autonomy, social ability, responsiveness, and proactiveness. They are able to act autonomously in open and time-constrained multi-agent environments, and interact with other agents using an agent communication language. [Ref. 20]

b. Interface Agents

Interface agents act on behalf of a user in a virtual environment. Their usefulness can range from managing mundane tasks like scheduling, to performing customized information searches which combine filtering and the production of alternative representations, to providing help and advice in interactive contexts. [Ref. 19]

Interface agents emphasize autonomy and learning in order to perform tasks for their owners. Interface agents collaborate with a human user, as opposed to collaborative agents, which collaborate with other agents. Interface agents do not require an explicit agent communication language to collaborate with a user. [Ref. 20]

Interface agents support and provide assistance to a user learning to use a particular application. They observe and monitor the actions taken by the user, learn

short-cuts, and suggest better ways of doing tasks. They act in a manner similar to an autonomous personal assistant, which cooperates with the user in accomplishing some task in the application. [Ref. 20]

c. Mobile Agents

Mobile agents are software processes capable of roaming large networks, interacting with foreign hosts, and gathering information on behalf of their owners. However, mobility is not a sufficient condition for being categorized as a software agent. Mobile agents are agents because they are autonomous and they cooperate. [Ref. 20]

d. Information Agents

Information agents perform the role of managing, manipulating, and collating information from many distributed sources. There is no fine line distinguishing the difference between information agents and collaborative and interface agents. There is a significant degree of overlap in functions. However, information agents are defined by what the do, in contrast to collaborative and interface agents, which are defined by what the attributes they possess. Information agents have varying characteristics, i.e., they may be non-cooperative or social, and they may or may not learn. [Ref. 20]

e. Reactive Agents

Reactive agents are a special category of agents that do not possess internal, symbolic models of their environments. Instead, they act in response to a stimulus in the environment in which they are embedded. Reactive agents possess emergent functionality, which means there is no prior specification of the behavior of

them. Reactive agents are viewed as a collection of modules, which operate autonomously and are responsible for specific tasks such as sensing and motor control. Each module is described in a language based on augmented finite state machines (AFSM). An AFSM is triggered to perform some action if its input signal exceeds some threshold. Reactive agents tend to operate on raw data in contrast to the high level symbolic representations that are required for other types of agents. There is no standard mode to reactive agent operation and therefore the attributes it possesses depends on the specific application. [Ref. 20]

f. Hybrid agents

Hybrid agents try to combine the characteristics of two or more different agent types into one agent. They attempt to maximize the strengths and minimize the weaknesses of these agents. The attributes a hybrid agent possesses depends on what combination of characteristics the agent is attempting to encapsulate (i.e. collaborative, interface, mobile, information, or reactive agents). [Ref. 20]

g. Heterogeneous Agents

Heterogeneous agent systems refer to an integrated setup of at least two or more agents which belong to two or more different agent classes. They require an agent communication language so different software agents can communicate with each other. These systems may also contain hybrid agents. Once again, the attributes a heterogeneous systems possesses depends on what combinations of software agents are used within the system. [Ref. 20]

2. Strengths of Software Agents

An agent-based infrastructure will allow the JBI to evolve towards a seamless integration of existing and evolving systems. Software agents can be used for a wide range of information tasks that include searching, filtering, and collating data from military systems, and monitoring information in these systems. Figure 15 shows how existing and evolving systems will be integrated into the JBI. [Ref. 4]

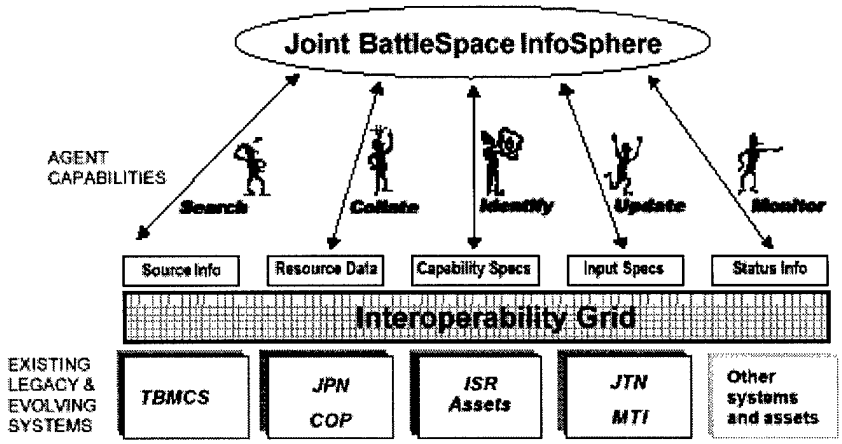

Figure 15. Integration of Systems Using Software Agents From Ref. [4]

A major issue with networks today, and surely one that is relevant to the JBI, is how to actively keep only the most relevant information at the forefront of user interaction. The use of software agents is one way to better cope with the increasing volume and complexity of information available. In this context, agents work to select the right data, fuse the applicable components of data, and format and present the information in the best way for a specific user. The JBI needs software that can fuse data to create new information. Information fusion will be an integral part of the JBI and

software agents will be the tools that make it possible. Advantages specific to the different kinds of software agents are described in the following subsections.

a. Collaborative Agents

Collaborative agents allow for the interconnecting and interoperation of multiple existing legacy systems. Collaborative agents enhance modularity which reduces complexity, enhances speed, reliability, flexibility, and reusability. [Ref. 3]

b. Interface Agents

The primary method for users to communicate with the JBI will be through some sort of interface agent. Interface agents ease the workload for the user and application developer. The agent can adapt over time to its user's preferences and habits. These agents help the user identify and find the resources they need to perform their mission, support translation of information to proper formats for their respective users by displaying the information objects that each user has subscribed to, and collaborate with the user to help solve a particular problem. In other words, interface agents can interact with humans at a problem-solving level. These agents can provide advice and recommendations to the user.

These agents allow communication among users and can be used to manipulate distributed information objects that are published to various subscribers. Thus, interface agents can be used in conjunction with the publish and subscribe mechanisms that will be part of the JBI services.

c. Mobile Agents

There may be a large amount of information in a network that needs to be examined to determine relevance, such as new information objects posted to the JBI. Transferring this information can be very time consuming and slow down the network. Mobile agents can go to a location, do a local search, and transfer only those information objects that are relevant. [Ref. 20] Therefore, the use of mobile agents can help to preserve bandwidth and maintain the performance of a network.

Mobile agents allow for asynchronous computing through multitasking. You can task these agents to do something and then work on something else as the agents perform their tasks and return results upon completion.

d. Information Agents

One of the main benefits of using information agents is that, along with collaborative agents, they allow for access and interaction with legacy systems, which will be an important part of the JBI. The JBI needs to be able to retrieve and process information manipulated by legacy military systems. These legacy systems accept information and commands from the JBI and provide information through Application Programming Interfaces. Information agents contribute to mapping the existing APIs to the proper formats within the JBI. Therefore, these legacy systems appear to be part of the JBI itself to the user.

e. Reactive Agents

Reactive agents are simple and easy to understand because their actions only depend on what happens at the present moment. Reactive agents are found to be

more robust and fault-tolerant than other agent-based systems. An agent may be lost, but without catastrophic effects in contrast to some of the other types of agents. They are also flexible and easily adaptable. [Ref. 20]

Reactive agents can work at the raw data level (i.e., directly from sensors), collecting this data for further processing in such tasks as information fusion, routing, formatting for presentation, etc.

f. Hybrid Agents

The main reason to have hybrid agents is that the benefits from having a combination of characteristics within a singular agent is greater than the gains obtained from the same agent based entirely on a single type. The benefits are the union of the benefits of the individual characteristics of different agents. [Ref. 20]

g. Heterogeneous Agents

The driving force for heterogeneous agents is interoperability. Many software products exist that can provide many services for the JBI. These programs work well in isolation, but they need to interoperate as a whole in this type of command and control architecture.

Programs designed for standalone applications can provide value-added services by enhancing them in order to participate and interoperate in a cooperative heterogeneous setup [Ref. 20]. An example would be a suite of collaborative and information agents that wrap legacy software into the JBI architecture by injecting code into that software to allow it to communicate in an ACL. This helps to solve the legacy

software problem by eliminating the need for costly program rewrites by getting these applications to interoperate with others.

3. Limitations of Software Agents

With respect to software agents in general, it is absolutely essential that robustness be a key factor in their design. Otherwise, there are many opportunities for software agents to fail in their tasks. For example, an agent that is designed to take an action when it recognizes a relevant event, may fail to deem some event relevant when in fact it was, or it may misinterpret the event altogether. On the other hand, it may correctly identify and interpret an event, but it may respond incorrectly. [Ref. 19]

Currently, there are many difficulties getting different kinds of software agents to communicate with each other. Many vendors have developed proprietary ways to get their software agents to communicate. There needs to be standardization in this area and a focus of effort on common agent communication languages. The current lack of standards and supporting infrastructure has hindered agent interoperability, which is the trait most desired in software agents. Many researchers are addressing this by trying to develop open, distributed architectures for software agents. One on-going effort in this area is the development of the Knowledge Query and Manipulation Language (KQML). It is a language that helps software agents in identifying, connecting with, and exchanging information with other agents. [Ref. 19]

Software agents produced by different developers cannot cooperate in any meaningful way. This is due primarily to the lack of standardization that exists in the development of software agents and the difficulty of creating a test environment to verify

and validate their performance. Cooperation among agents is critical to building powerful applications that support military capability because without cooperation, each new task must be handled by a new agent designed for it. Control strategies are needed to build teams of agents that can cooperate. Also, it is difficult to predict and control the behavior of current software agents. There are no algorithms or mechanisms that prevent a large heterogeneous set of agents from exhibiting chaotic behavior on a network. This lack of control can lead to degraded networks, poor performance, system crashes, and security vulnerabilities. The limitations specific to different types of agents are addressed in the subsections to follow. [Ref. 18]

a. Collaborative Agents

Coordination is essential to enabling groups of agents to solve problems effectively. Currently, there is no standardized inter-agent coordination scheme. This can lead to deadlock in collaborative agent systems. In addition, there is no established way to evaluate collaborative agent systems. This makes it difficult to verify and validate these systems to ensure they meet their functional specifications. [Ref. 20]

b. Interface Agents

There are three primary issues with respect to interface agents. First, it is difficult to demonstrate that the knowledge learned with interface agents can truly be used to reduce user workload [Ref. 20]. Second, it is difficult to get interface agents to be able to negotiate with other peer agents [Ref. 20]. Third, there is a concern for the user becoming too dependent on the interface agent, which has limited capabilities, for assistance in problem solving.

c. Mobile Agents

Mobile agents have unresolved issues in terms of authentication, secrecy and security. These topics will be discussed more in Chapter IV, Computer Network Security. Also, it is unclear what would be the effect of having huge numbers of such agents in a wide area network (WAN). [Ref. 20]

d. Information Agents

If information agents are static in nature, the same challenges that apply to interface agents in terms of reducing user workload and peer agent negotiation also apply to information agents. If the information agents are mobile, then the challenges for mobile agents are applicable. [Ref. 20]

e. Reactive Agents

Right now, there are few applications that span a narrow range which are based on reactive agents. There needs to be a clearer methodology to facilitate the development of reactive software agent applications. Much of the current work in this area uses simple trial and error. [Ref. 20]

f. Hybrid Agents

Traditionally, network architectures based on hybrid software agents translate to ad hoc designs, which create numerous problems in themselves (standardization, interoperability, maintainability, etc.). Historically, architectures utilizing hybrid agents traditionally tend to be very application specific. [Ref. 20]

g. Heterogeneous Agents

Presently, the work on heterogeneous agent systems in its infancy. There exists a need for tools, methodologies, techniques, and standards for achieving interoperability among heterogeneous sources. [Ref. 20]

F. MULTICASTING

1. Description

There are three basic ways to transfer data in a network. In unicast data transfer, a unicast data packet is addressed to a single node on the network. Each of these nodes has a unique address, an IP address for example. A second way to transfer data is to broadcast the data. In this method, each bridge or router forwards packets to all other paths to which it is connected less the path from which the packet came. This uses a large amount of bandwidth because all destinations will receive the packets, which travel all available transmission routes. A third way to transfer data is through multicast delivery. Multicasting facilitates simultaneous dissemination of information to specific receivers within the network over a single connection. A multicast packet is addressed to a subset of nodes on the network. Only a single copy of the data is sent by the source, and routers within the transmission path generate the required number of copies for delivery to all recipients. Figure 16 on the next page gives a graphical view of how multicasting routes packets through a network. [Ref. 21]

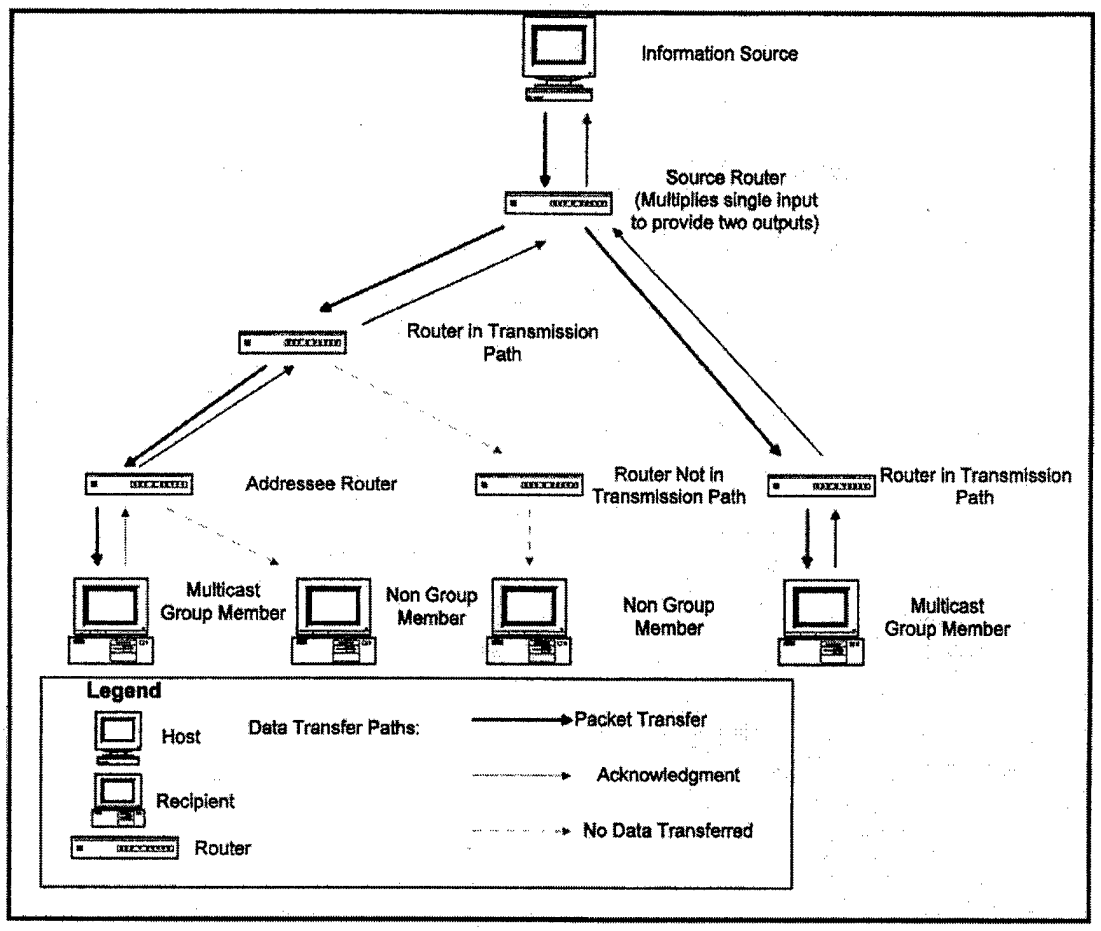

Figure 16. Multicast Delivery of Packets Through a Network From Ref. [21]

Multicast distribution is based on multicast groups. A group is a collection of one or more senders and receivers, which are considered a single entity and have a single locating address. Therefore, the dynamics of the group can change and the senders of data do not have to change the destination address of packets as a result of this. A host indicates that it is interested in receiving transmissions through a group subscription. Routers keep track of these groups and dynamically build distribution trees to chart paths from senders to receivers. One multicast router in each network queries all hosts

periodically to refresh group membership. Each host belonging to a group responds with a report for each group to which they belong. [Ref. 22]

There are two types of multicast distribution trees. These are called source based trees and shared trees, also known as core based trees. Source based trees rely on periodic broadcasting to maintain the tree. Multicast packets are periodically broadcast across the network to advertise multicast data. To reduce transmitting duplicate packets across the network, multicast broadcasting is done only for packets that arrive on ports that the router considers to be the shortest path back to the source of the packet. In contrast, shared trees create a rendezvous point that becomes the center of the multicast group. Each host that wants to receive multicast data from a different group sends a request to the rendezvous point to join those groups. [Ref. 22]

Tunneling is used to transfer data across regions where there are routers that do not support multicast traffic. This technique is popular with the Internet. A tunnel encapsulates an IP multicast packet into a unicast packet and then un-encapsulates it at the end of the tunnel. The following subsections give examples of four protocols that use multicasting. [Ref. 22]

a. Distant-Vector Multicast Routing Protocol (DVMRP)

DVMRP uses a process called reverse path forwarding for data transmission. In this process, the first packet, which contains the source/group pair, is broadcast to all routers within the network. If a router does not have any relevant subscriptions, it returns a message stating this and is removed as a path from the group. These routers can become part of the multicast architecture at a later time if subscribers

are added within their subnet. DVMRP implements its own routing table to maintain the current state of the group. [Ref. 22]

DVMRP algorithms are straightforward and easy to implement, making this type of architecture easy to set up. DVRMP is widely used in the Multicast Backbone (MBone), so a network implementing DVRMP can use the functionality of MBone to deliver global information. MBone is an experimental collection of multicast router islands that are interconnected by tunneling on top of the Internet. Therefore, a network using DVRMP can tap into global information resources available on the Internet. Also, DVRMP supports tunneling to connect across routers that do not implement multicast. [Ref. 22]

DVMRP relies on periodic broadcasting to maintain routing tables, and this utilizes precious bandwidth. It also utilizes a great deal of routing table memory to store state records. As a result, distant-vector algorithms do not scale well and are therefore suitable only for small networks. [Ref. 22]

b. Multicast Open Shortest Path First (MOSPF)

MOSPF uses source/group pairs to establish multicast traffic. It uses an Open Shortest Path First (OSPF) algorithm to determine the shortest reverse path through a network. It maintains a local database of group memberships through network monitoring. One MOSPF router on each subnet is responsible for subscriptions and sends host membership reports to all other MOSPF routers. [Ref. 22]

MOSPF provides scalability to a network and thus, is well suited for large dynamic networks such as the JBI. MOSPF performs route calculations on demand, and,

therefore, there is no need to broadcast to build the initial distribution tree. It also has the capability to integrate with DVMRP by using a special DVMRP multicast tunnel that provides a way to integrate MOSPF into MBone. This allows MOSPF to be used internally within a domain, such as a subnet, utilizing DVMRP as a gateway protocol between domains. [Ref. 22]

The main problem with MOSPF is that it requires OSPF to run on every router participating in multicasting. OSPF is a link-state protocol that enables the network manager to configure routes based on different metrics, such as the speed of transmission or the most fault-tolerant link. MOSPF does not support tunneling across routers not capable of multicasting unless the router is running OSPF. [Ref. 22]

c. Multicast Transport Protocol (MTP-2)

MTP-2 uses the concept of a multicast master and subordinate senders and receivers. A sender who wishes to transmit a packet sends a unicast request for a token to the master. Once approved, the sender uses the token to multicast the packet. The token is then returned to the master. [Ref. 21]

MTP-2 is a protocol that requires minimal overhead compared to other protocols. MTP-2 allows for error correction if the network starts to incur losses by migrating the master from one machine to another or designating a new master, thereby making the network more robust. Also, MTP-2 has a prioritization scheme that can be enacted for token requests so more critical information, such as intelligence, can be delivered first. [Ref. 21]

The use of MPT-2 is not without its drawbacks. MPT-2 relies on a receiver initiated error recovery scheme that restricts scalability. Retransmission requests for missing packets might be made by multiple receivers, which could lead to multiple retransmissions of the same packets. Also, after a sender transmits a packet of data, it waits for a period of time to receive retransmission requests and then discards that data. If a request comes in after the allotted time, that request will be unfulfilled, which means some intended recipient did not get the information, and this condition is unacceptable in a combat network. [Ref. 21]

d. Xpress Transport Protocol (XTP)

XTP is a general-purpose protocol that can provide many communication protocol needs such as reliable multicast connections. This general-purpose approach provides greater flexibility and support for reliability. It is a high performance protocol designed to meet the needs of distributed, real-time, and multimedia systems in both unicast and multicast environments. [Ref. 21]

Within XTP, there is no requirement for data to have one particular structure. This leads to adaptability to the communication needs within a specific architecture. At the core of XTP is a set of mechanisms whose functionality is orthogonal. XTP separates communication paradigms from the error control policy employed. Flow control, rate control, and error handling can be tailored to the communication needs. [Ref. 23]

2. Strengths of Multicasting

Multicasting is well suited for real-time applications such as video or audio broadcasting as well as transferring a single file to many locations at once. It is also used for sending out simultaneous updates to multiple PCs, which is similar to the subscription process within the JBI. Multicasting leads to a more efficient use of bandwidth since there is no need to make numerous copies of the packet to send to multiple recipients. Furthermore, it allows for near concurrent receipt of information, and packets are routed to only those interested in receiving the information, which is an important function of the JBI. All of these characteristics make multicast delivery an ideal tool to be utilized by the JBI. [Ref. 21, 22]

Other benefits of multicasting depend upon whether the multicast distribution tree is source based or share based. Source based trees are resilient to network failures since separate trees are maintained for each multicast recipient. This makes them ideal for networks in a military combat environment. They are also efficient, since packets follow the shortest path to their destination. This makes them especially well suited for time critical information such as targeting, which could change while strike aircraft are en-route. However, source based trees do not scale well in a large network due to broadcasting. Shared trees, on the other hand, reduce broadcasting and flooding the network and, therefore, scale better than source based trees. This makes them a good choice for a large network, but they also utilize less efficient paths and are more vulnerable to network failure, which means they are less robust in a military environment. [Ref. 22]

The use of reliable multicast in a system of systems such as the JBI would be beneficial in traffic periods characteristic of crisis situations. It can be used to transmit any information where assured delivery is critical. The JBI needs a multicast protocol that is flexible enough to handle traffic that requires a high priority while at the same time, handling more routine, less critical traffic.

3. Limitations of Multicasting

One of the principal problems with most multicast protocols today is their inability to scale, which is a necessity in the widespread deployment of multicast technology. MOSPF and XTP are exceptions. Keeping a large distribution tree intact across a huge network can flood the network with updates and requires extremely large routing tables. Research must look at hierarchical routing techniques as being key to multicast since this is how the Internet has scaled to service so many users. [Ref. 21]

Management of multicast groups becomes a tedious task when the number of receivers of information and the number of multicast groups in a network increases. The result is that personnel involved in a military operation focus more effort on managing the network itself instead of mission tasks. The greater the number of these receivers, the lower the throughput, since each receiver must provide the requisite acknowledgements. Multicast throughput is highly dependent upon the performance of the slowest receiver. If a receiver that exhibits fast performance is added to the group of receivers, throughput will slow a small amount because one additional acknowledgement must make it back to the transmitter prior to packet transfer. However, a receiver with slow performance has

the potential to significantly delay packet transfer. Despite this, research and testing in this area has found the degradation to be negligible in many instances. [Ref. 21]

IV. COMPUTER NETWORK SECURITY

The last chapter described different types of tools that can be used to help create the core services for the Joint Battlespace Infosphere. It provided descriptions of different types of software tools popular with the Internet/World Wide Web that can be used to create a dynamic C2 network architecture, and assessed their strengths and weaknesses. This chapter is devoted to the issue of security within networks. It will discuss the background of the problems with network security, the threats to networks, the network security controls that exist, the security requirements of a network such as the JBI, and example systems used for security.

A. BACKGROUND

Attacks on DoD computer systems are a serious and growing threat. The exact number of attacks is unknown because only a small portion are detected and reported. The Defense Information Systems Agency (DISA) has estimated that the DoD has experienced as many as 250,000 attacks in 1995. In testing its own systems, DISA attacks and successfully penetrates defense systems 65% of the time, and their data show that the number of attacks is doubling each year. These attacks cost the DoD millions of dollars each year and have the potential to be a serious threat to national security if attackers successfully corrupt sensitive information or deny service from critical communications backbones or power systems. Attackers have seized control of entire DoD networks, many of which support critical functions such as weapon systems research and development. Attackers have also stolen, modified, and destroyed data and software. They have shut down and crashed entire systems and networks. They have

installed unwanted files and back doors that allow the attacker continued unauthorized access in the future. [Ref. 24]

The task of preventing unauthorized users from compromising the confidentiality, integrity, or availability of sensitive information is becoming more and more difficult, especially with the increased skill of the attackers and the technological advances in their tools and methods of attack. The DoD is attempting to react to successful attacks, but it is lacking in uniform policy for assessing risks, protecting its systems, responding to incidents, and assessing damage. [Ref. 24]

B. THREATS TO NETWORK SECURITY

There are many different kinds of threats to which networks are exposed. Beside the threat of natural disasters, the most significant threats are man made in the form of attacks. The most dangerous forms of attack are those designed to corrupt, distort, or implant false information into a network. The attack methods expected to be directed against military networks include the full range of countermeasures designed to disrupt, degrade, deny, and destroy the information functions provided to military forces. [Ref. 6]

The basic threats to the security of a network, not including natural disasters and intentional physical destruction, include wiretapping, impersonation, data confidentiality violations, data integrity violations, hacking, code integrity violations, and denial of service attacks. These threats can be malicious in the form of attacks, or they can be unintentional, as in operator error.

1. Wiretapping

Wiretapping is defined as intercepting communications. It can be done covertly, so neither the sender nor receiver know the communication has been intercepted. A passive wiretapper just listens, but an active wiretapper could modify the communication. [Ref. 25]

2. Impersonation

Impersonation is a significant threat in large networks. When one person impersonates another, they are able to obtain information from the network directly. In this type of threat, an attacker can get in by using a target whose authentication data is known, guess the authentication details of the target, use a target that does not require authentication, or circumvent the authentication mechanism altogether. [Ref. 25]

3. Data Confidentiality Violations

Sometimes data is misdelivered due to some flaw in the network hardware or software. Sometimes a destination address is modified or a message is delivered to someone other than the intended recipient. It is also common for humans to mistype network addresses of recipients. [Ref. 25]

4. Data Integrity Violations

When data is sent between hosts on a network, attackers can change the content of data, replace the data entirely, reuse old data, change the source of the data, redirect the data, or delete the data. [Ref. 25]

5. Hacking

Hackers can use methods of attack other than those already described, such as achieving access to a host through a back door or security flaw and then sending false messages or mounting a denial of service attack by flooding the network. [Ref. 25]

6. Code Integrity Violations

A serious threat in networks is damage to executable code. This is usually accomplished through the use of viruses, worms, Trojan horses, and other types of malicious software. This software is transferred to a network when an unsuspecting user downloads a file that contains the malicious code. [Ref. 25]

7. Denial of Service

The impact of a denial of service attack grows as people become increasingly dependent on computer communications for interaction. Denial of service is caused by connectivity problems due to a failed link or host, generation of spurious messages that flood a network, routing problems which could be caused by routing table modifications, or the launching of malicious software on a network which forces the network to shut down for repairs and removal of the software. [Ref. 25]

C. NETWORK SECURITY CONTROLS

There are numerous security controls that can protect against network exposures. These include encryption, access control, authentication, traffic control, data integrity, firewalls, and trusted network interfaces, to name a few.

1. Encryption

Encryption is a powerful tool that can provide privacy, authenticity, integrity, and limited access to data. In a network, encryption can be applied between two hosts (link encryption) or between two applications (end-to-end encryption). [Ref. 25]

With link encryption, data is encrypted just before the system places it on the physical communications link. Decryption occurs just as the data enters the receiving computer. The message is encrypted in transit between two systems, but the data is in plaintext inside the hosts. If the message passes through an intermediate host, it may be transformed to plaintext for routing and addressing purposes before being re-encrypted and sent on its way. These intermediate hosts need to be trustworthy. The host must also have a cryptographic facility in order to decrypt the data. This encryption method is performed at a low level protocol layer and is therefore invisible to the user. Link encryption is appropriate in networks where the transmission line is the point of greatest vulnerability and all hosts are reasonably secure. Link encryption is also fast and easy to use. [Ref. 25]

End-to-end encryption provides security from one end of a transmission through the other. The encryption precedes all transmissions and routing. The data is therefore transmitted in encrypted form all throughout the network. Data sent through several intermediate hosts is still protected since the content of the message is still encrypted. Because intermediate hosts do not need to encrypt or decrypt the message, they have no need for cryptographic facilities. End-to-end encryption is useful when there is a need to supply encryption selectively to different applications. [Ref. 25]

Public-key cryptography was invented primarily to solve the problem of distribution of secret message keys, but introduced the problem of use of compromised keys. This issue was addressed by the creation of public-key certificates, signed by a certificate authority, which vouch for the authenticity of a public key. The public-key infrastructure (PKI) is a network of services that includes certificate authorities, certificate repositories and directories for finding and storing public key certificates, and certificate revocation lists for managing keys that expire or are revoked. [Ref. 26]

2. Access Control

Access to data, programs, and other resources on a network is a serious concern in network security. When a computing system is part of a network, users may not know which other users may be connected to the same network. In a network environment, access control must protect each system of the network and also prevent unauthorized users from passing through one system of a network to access other systems. Tools such as automatic call-back systems and silent modems help to maintain access control. [Ref. 25]

3. Authentication in Distributed Systems

Host-to-host and user-to-host authentication is important in a network so that hosts are assured of the authenticity of a remote host or a user on a remote host. Digital distributed authentication is an example of an architecture that was developed by Digital Equipment Corp. to authenticate non-human entities within a computing system, such as two processes that need to exchange data. This architecture is effective against the threats of impersonation of a server by a rogue process, interception or modification of

data exchanged between two servers, and replay of a previous authentication. Distributed Computing Environment (DCE) is an example of a set of software tools and services that make it easier to operate distributed, heterogeneous, computer applications. It presents a complete support environment for building distributed applications through managing controlled, shared access to remote and distributed sources. Biometric identification technologies use physiological traits such as voice, fingerprint, or eye recognition to provide an identity check of all users of the network. [Ref. 6, 25]

User names and passwords are probably the most popular authentication tool used in networks. When a user enters a username and password, the network compares this information against allowable names and passwords, and grants access if there is a match. Passwords need to be well chosen and changed periodically. Poorly chosen passwords can be easily deciphered using readily available password cracking programs. If users have too many passwords to remember, they will use a few simple ones that are easy to break. A potential solution to the problem is using security servers, which map all access permissions to a single encrypted username and password authentication system for the entire network. [Ref. 9]

4. Traffic Control

Sometimes it is not just the content but the mere existence of data being sent between users that is sensitive. When an interceptor monitors only the existence of message traffic on a network, this is known as traffic flow analysis. The standard control for such is the introduction of spurious messages between points of low traffic so that message traffic between all nodes appears symmetrical. It will then be hard to point out

heavy communication between two hosts that can make an interceptor suspicious. This method would be ideal for a deception plan by creating false traffic between two hosts to make the enemy think an operation may be impending in a location different than the actual location. The drawback to this method is, of course, the added traffic places an extra burden on the network. [Ref. 25]

5. Data Integrity

Data integrity is a function of the correct generation, storage and transmission of data. To help preserve the integrity of the data in a network, the network should use protocols that accommodate some form of encryption so an intruder cannot modify packets in transmission. A second way to guard against message tampering is to use a cryptographic checksum. This is check data built into a message to detect and sometimes to correct failures. A digital signature, on the other hand, is a means to certify the authenticity of a set of data and ensure the data was not modified. It is a mark only the sender can make, and it is intended to be unforgeable and not alterable. [Ref. 25]

6. Firewalls

The easiest way to protect sensitive resources is to not connect them to any system accessible from outside the organization's security perimeter. However, C2 networks today demand global resources; so this is not a feasible solution. These networks need filters that let through only those interactions which are needed. The most popular method of providing such filtering is through the use of firewalls. A firewall is a process that filters all traffic between a protected network and a less trustworthy network. Firewalls implement a network's security policy. All network accesses from the outside

should be controlled by the firewall and should pass through it. In addition, firewalls are well isolated, making them highly immune to modification. [Ref. 25]

It should be emphasized that firewalls are not a "catch-all" security solution for network security. The can protect an environment only if the firewalls control the entire perimeter, i.e., there are no connections through the perimeter not mediated by the firewall. They do not protect data outside the perimeter. Firewalls are targets of penetrators, and although they are designed to withstand attack, they are not impenetrable. Therefore several different layers of protection is better than relying on the strengths of a single firewall. [Ref. 25]

There are three basic types of firewalls:

1. Screening routers
2. Proxy gateway
3. Guard

a. Screening Router

A screening router is nothing more than a computer that routes communications toward a target. Screening routers become the connection from a network to other outside networks. They apply screening rules to the packets passing in and out of the network. They attempt to determine whether a packet from the outside is attempting to forge an inside address. They allow into the network only packets from allowable addresses, and allow out only communications destined for permissible outside addresses. [Ref. 25]

b. Proxy Gateway

Screening routers only look at the header information of a packet, not the data inside the packet. A proxy gateway forces an application to act properly upon receipt of requests. It intervenes in the communication protocol between the outside sender and inside receiver. To the sender, the proxy gateway appears to be the destination of the communication. To the receiver, it appears to be the sender of the communication. It is typically an isolated machine with very limited capability beyond implementing the network security policy. It controls actions through the firewall based on data within the packet, not just external header data. [Ref. 25]

c. Guard

A guard receives protocol data packets, interprets them, and then passes through the same packets or different packets that achieve the same result as the originals. A guard screens data going in both directions by interpretation of message content. It decides what services to perform on the user's behalf. The security policy the guard implements is somewhat more complex than the action of a proxy gateway. However, since it is a more complex firewall that has greater functionality, there are also more ways for it to fail. [Ref. 25]

6. Trusted Network Interface

One way to achieve multilevel security certification of a computer network is to form a trusted network base called the trusted network interface. Figure 17 on the next page shows how a trusted network interface is established.

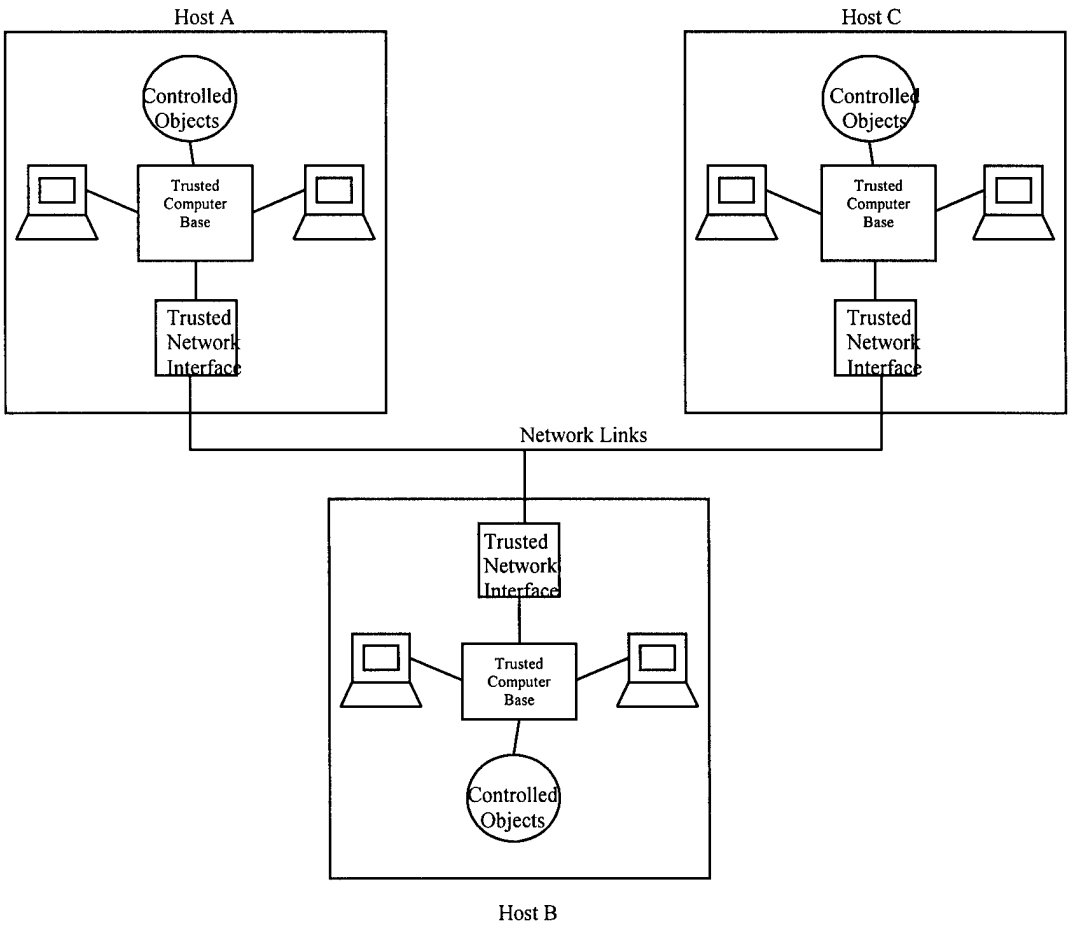

Figure 17. Trusted Network Interface From Ref. [25]

Each host has a trusted network interface. These hosts are cautious of other hosts that join the network without a trusted network interface. Each interface is responsible for maintaining the security of the resources it controls. The interface performs activities such as preserving the security of its host, checking classification level before releasing data, ensuring data integrity, and others. Multilevel security will be discussed in the next section. [Ref. 25]

D. SECURITY REQUIREMENTS OF C2 NETWORKS

The world is increasingly relying on electronic networks for the secure and uninterrupted flow of digital information. Protecting critical information systems and the data on them is the key to running successful military operations. The four key properties that information systems must exhibit to be robust and survivable are resistance to attacks, recognition of attacks and the extent of damages, recovery of full and essential services after an attack, and adaptation and evolution to reduce the effectiveness of future attacks. [Ref. 27, 28]

The military must be able to protect its information systems, prevent unauthorized intrusions, detect attacks in a timely fashion, and react to attacks in an appropriate manner. The difficulty with an information infrastructure such as the JBI is that it provides a wide range of services, and the more services a system provides, the higher the probability that there will be security vulnerabilities. Security must continually be addressed all through the development of the JBI and not just an afterthought. [Ref. 4]

Traditional methods of attack such as electronic jamming are countered through hardening, dispersal, and redundancy. The JBI will be a system-of-systems with multiple nodes and distributed processing which eliminates the possibility of a single point of failure if a node is attacked. If a node is effectively cut off from the rest of the system, the immediate effect is felt only at those isolated points and not across the entire network. Information flow will be rerouted around the disrupted node. [Ref. 6]

The JBI must manage a variety of information from different sources. It must distribute this information only to authorized people and systems. The levels of authorization may vary for different users and different information objects. The JBI

architecture must provide a multilevel security environment within multiple layers to protect the information. In a multilevel secure network, two or more people want to share network access at different classification levels. A multilevel security network must preserve the property that no user may read data at a level higher than that for which the person is authorized. It must also preserve the property that no user may write data to a level lower than the level the person has accessed. [Ref. 4, 25]

The military nature of the information within the JBI demands strict ability to control and access data. The JBI needs access control lists for the different types of information within the network. These lists are used to define who may retrieve or subscribe to which information objects and at what level of classification. The commander of an operation uses access control services to control the kinds of information flowing within and outside the theater of operation. These access controls pertain not only to users, but also to software agents that operate autonomously. It is essential to authenticate all users and all information in the JBI. However, it is extremely difficult to manage and modify access lists when the value of the information changes from a security perspective. Therefore, in addition to access lists, information objects need to be tagged such that it is possible for authorized users to change the security value of objects belonging only to them. No other users or systems should be allowed to modify the security value of those objects. [Ref. 4]

Other tools such as firewalls and intrusion detection software will be an important part of the JBI. Firewalls were discussed in detail in the last section. Intrusion detection systems set off alarms when unusual events are detected. Many intrusion detection tools provide some form of automated intruder response. Rapid response to intrusion detection

is needed to maintain system availability. Current research needs to focus on technology for automated response selection. [Ref. 4]

In terms of acquiring secure systems for the JBI, the competition in the commercial market will drive the development of more secure systems in the future. The JBI will evolve from the most secure commercial operating systems available. Building a proprietary secure system from the ground up will cost too much in terms of time, research, and development, and is not necessary with the abundance of commercial technology available. The JBI developers need to know the security requirements of the system, understand and evaluate commercially available products, utilize the most appropriate commercial products, and augment the capabilities of those products with additional security capabilities as required.

E. SYSTEMS FOR SECURITY

There are numerous research and development programs aimed at creating more secure networks. The Information Assurance Program at DARPA is developing security and survivability solutions that will reduce vulnerability and allow increased interoperability of network systems. The Dynamic Coalitions Program, also a DARPA project, is developing technologies to enable secure collaboration with coalition partners. DARPA's Information Assurance Science and Engineering Tools program will allow both the DoD and commercial developers to create systems with specified assurance properties and measurable effectiveness against attack. Information Server Support Environment Guard System is a system developed by the Air Force Research Lab that provides a trusted interface for high-speed, digital transfer of intelligence information

including text, imagery, graphics, and fusion products [Ref. 29]. These are just a few examples of the many efforts to create more secure networks and the technologies that exist to protect the security of the JBI. [Ref. 4]

There also exist a number of available COTS software tools that check a network for weaknesses. CRACK is a collection of password-checking tools. Tripwire is a tool to use after a suspected penetration. It is a file integrity checker that compares active versions of files against a backup to determine which files have been changed. COPS is a set of programs that check important system files, user configurations, and permissions settings to list potential security flaws. These are a few of the many available software tools that analyze the security of a network, tools the JBI should incorporate into its information management services. [Ref. 6]

F. CONCLUSIONS

The security environment of a network must be viewed as a whole. All possible exposures should be considered, and each security tool must fit into a larger comprehensive security strategy.

Increased security comes with a price. More stringent security measures cost money and increase overhead with respect to network performance. Increased access control mechanisms increase the complexity of the network. However, building reliable security measures is less costly than suffering the theft of critical information. The more widespread the use of effective security technology, the lower the cost in the long run. [Ref. 25]

THIS PAGE INTENTIONALLY LEFT BLANK

V. RECOMMENDATIONS AND CONCLUSIONS

A. RECOMMENDATIONS

Building a C2 system-of-systems of the complexity of the JBI requires a great deal of research, design, and implementation effort. However, this type of system does not need to be built from scratch. The existing base of commercial information technologies can be leveraged to the maximum extent possible. The DoD must leverage the commercial market's lead in information technology development. The exponential growth of communications and networking technology in the commercial sector will provide the military with cost effective information technology tools to create the JBI. Therefore, the DoD will not need to invest substantial sums to create this type of C2 architecture. However, the DoD needs to invest in technology that is not being actively pursued in the commercial market, but is needed to complete development of the JBI. Examples are unique application interfaces, network aware-intelligent agents, hardened, survivable systems, and multi-source fusion technologies. [Ref. 4, 6]

A software system such as the JBI should be viewed as an evolving system-of-systems that will provide capabilities and services far into the future. These capabilities will be delivered incrementally over the lifetime of the system. They will be implemented in an evolutionary fashion such that each additional function added to the system provides a measurable piece of the requisite performance desired. The first increment of the design should provide only the minimum useful capability, and each successive increment provides additional capability. Therefore, an investment and procurement strategy needs to be developed based on these attributes that spans the life of the program. [Ref. 4]

The current methods for defense acquisition are inadequate for procuring information technology systems. The traditional acquisition cycle moves too slow to take advantage of the latest available technologies. A performance-driven approach is needed that specifies the functionality desired in contrast to a detailed technical specification of the architecture. A performance-based specification states requirements in terms of the required results and provides criteria for verifying compliance, but it does not state how to achieve those results. [Ref. 4]

The spiral acquisition model is more appropriate for procurement of systems to incrementally achieve the JBI. The spiral acquisition model allows for the evolution of a system from its initial capabilities. The development of a system such as the JBI needs to follow an evolutionary process. The spiral model is an evolutionary process model that provides for rapid development of incremental versions of information technologies. Development of a system is in a series of incremental releases. During early iterations, the incremental release might be a prototype. During later iterations, more complex versions of the engineered system are produced. [Ref. 30]

The spiral model is divided into a number of task regions, which can include such activities as planning, risk analysis, engineering, testing, and construction and release. Each of these activities contains a number of work tasks that are adapted to the specific project.

As the process begins, the developers move around a development spiral starting at the core. The first cycle around the spiral may result in the development of a product specification. Subsequent cycles around the spiral may be used to develop a prototype

and then a progressively more sophisticated version of the system. This model is adapted to apply throughout the life of the system. [Ref. 30]

The technologies that are needed to create the JBI are, for the most part, being thoroughly researched within the government and commercial sector. New technologies emerge out of both sectors on a continuing basis which can impact the acquisition of JBI systems. Technology that was not feasible during the first increment of development may be incorporated in the JBI in later increments. These technologies must be assessed on a continuing basis for possible enhancement of the functionality of the JBI. [Ref. 3, 4]

Chapter III provided an analysis of a number of commercial Internet/Web-based tools that could provide many benefits to the JBI. Chapter IV discussed the issues related to security in networks and some of the commercial products available to promote and enhance network security. Based on this research, there is a need for continuing research in the design, test, evaluation, and operation of message-oriented middleware, agent-based systems, and commercial security and information assurance products in order to fully realize the potential of an information infrastructure such as the JBI.

B. CONCLUSIONS

The military can reap great benefits from having a C2 system-of-systems such as the JBI. The benefits of such a system fall into the categories of survival, effectiveness, and efficiency. Survival benefits are due to avoidance of friendly force losses because of timely information dissemination leading to information superiority. Effectiveness benefits result from being able to make decisions to achieve desired effects with the acquired information. The JBI concept is an information management infrastructure that

can improve unity of effort, exploit total force capabilities, properly position critical information, and produce a comprehensive and accurate picture of the battlespace, thereby maximizing the effectiveness of military forces. Efficiency benefits result from achieving desired effects with low effort and cost. [Ref. 4]

How well the JBI performs depends on providing information that can span the vast needs of today's missions. The information objects will be of many different varieties coming from many different information systems. It was stated in the last section that the commercial technology base should be leveraged in order to acquire many of these tools/systems. There are many development toolkits currently available to create Web-based applications, such as inexpensive reliable code generators for HTML that run on multiple platforms. By taking advantage of such tools, the benefits to the end user will be seen in cross-platform interoperability, common user interfaces, operational scalability from a small network to a worldwide enterprise, and platform scalability from hand held PCs to high performance workstations [Ref. 31].

The development of a command and control system such as the JBI will be a monumental milestone for the DoD. Such a system will provide previously unattainable levels of interoperability and information systems integration within and among the services, thereby bringing the DoD closer to the goals outlined in *JV 2010*. Focusing on the necessary research efforts, adopting the appropriate acquisition strategy, and leveraging the commercial technology base will make the JBI a reality.

LIST OF REFERENCES

1. "The Air Force & Joint Vision 2010." [http://www.af.mil/lib/afissues/1997/issues22.html], 1997.

2. Joint Publication 6-0, *Joint Doctrine for Command, Control, Communications, and Computer (C4) Systems Support to Joint Operations*, pp. I-1 to II-12, 30 May 1995.

3. USAF Scientific Advisory Board, SAB-TR-98-02, *Information Management to Support the Warrior*, pp. 1-72, Unclassified, December 1998.

4. USAF Scientific Advisory Board, SAB-TR-99-02, *Building the Joint Battlespace Infosphere*, pp. 1-138, Unclassified, September 1999.

5. United States Atlantic Command, *Capstone Requirements White Paper for Joint C4 to Meet the Needs of 2010 and Beyond*, pp. 3-22.

6. Murphy, E., LtCol, and others, "Information Operations: Wisdom Warfare for 2025," [http://www.au.af.mil/au/2025/volume1/chap01/v1c1-1.htm], April 1996.

7. Atkins, R., LtCol, and others, "2025 In-Time Information Integration System (I3S)," [http://www.au.af.mil/au/2025/volume1/chap03/v1c3-1.htm], April 1996.

8. Air Force Research Lab/IF, *Joint Battlespace Infosphere Initiative*, Unclassified.

9. Morgan, Reece., *An Intranet For the Systems Management Curricular Office*, Master's Thesis, Naval Postgraduate School, Monterey, California, September 1997.

10. Baber, R. and Meyer, M., *Computers in Your Future 98*, pp.4-9 to 4-10, Macmillan Publishing, 1998.

11. Maze, S., Moxley, D., and Smith, D., *Authoritative Guide to Web Search Engines*, pp. 7-39 , Neal-Schuman Publishers Inc., 1997.

12. Clement, G., *Science and Technology on the Internet*, pp. 128-191, Library Solutions Press, 1995.

13. "Middleware," [http://www.sei.cmu.edu/str/descriptions/middleware_body.html].

14. Weiss, K., *Integrating Middleware Software into Open-System Client/Server Systems*, Master's Thesis, Naval Postgraduate School, Monterey, California, September 1995.

15. "Everything You Need to Know About Middleware: Mission-Critical Interprocess Communication, A Talarian Whitepaper," [http://www.talarian.com/industry/middleware/whitepaper.shtml], 1997.

16. "Message-Oriented Middleware," [http://www.sei.cmu.edu/str/descriptions/momt_body.html].

17. "Publish & Subscribe: Middleware With a Mission, A Talarian Whitepaper," [http://www.talarian.com/products/smartsockets/whitepaper.shtml], April 1998.

18. Hendler, James, "Control of Agent-Based Systems," [http://dtsn.darpa.mil/iso/programtemp.asp?mode=126].

19. Bradshaw, J.M., *Software Agents*, pp. 7-392, AAAI Press, 1997.

20. Nwana, Hyacinth, "Software Agents: An Overview," [http://agents.umbc.edu/introduction/ao/], September 1996.

21. Johnstone, G.S., Glenn, W.D., *Applied Reliable Multicast Using the Xpress Transport Protocol*, Master's Thesis, Naval Postgraduate School, Monterey, California, March 1997.

22. Black, Darryl, *Building Switched Networks*, pp. 193-209, Addison-Wesley, 1999.

23. "A Brief Introduction to the Xpress Transport Protocol," [http://www.ca.sandia.gov/xtp/xtpintro.html].

24. GAO Report on Pentagon Computer Security, "Information Security - Computer Attacks at Department of Defense Pose Increasing Risks," [http://epic.org/security/GAO_DOD_security.html], May 1996.

25. Pfleeger, Charles P., *Security in Computing*, Second Edition, pp.378-442, Prentice Hall PTR, 1996.

26. Denning, Dorothy E., *Information Warfare and Security*, pp.336-338, Addison Wesley, Longman, Inc., 1999.

27. Tenet, George J., "Remarks by DCI George J. Tenet - Sam Nunn Nations Bank Policy Forum," [http://www.odci.gov/cia/public_affairs/speeches/archives/1998/dci_speech_040698.html], April 1998.

28. Brutch, P., Brutch, T., and Pooch, U., "Electronic Quarantine: An Automated Intruder Response Tool," [http://www.cert.org/research/isw98/all_the_papaers/no6.html], 1998.

29. "Information Support Sever Environment Guard - Secure Communications," [http://www.if.afrl.af.mil/tech/programs/isse/].

30. Pressman, Roger S., *Software Engineering*, Fourth Edition, pp. 26-41, McGraw-Hill, 1997.

31. Naval Research Laboratory and NetSpace Corporation, *Exploitation of Web Technologies for C2 Networks*, Gardner, S. and others, pp. 2-3.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center 2
   8725 John J. Kingman Road, Suite 0944
   Ft. Belvoir, VA 22060-6218

2. Dudley Knox Library 2
   Naval Postgraduate School
   411 Dyer Road
   Monterey, CA 93943-5101

3. Professor William Kemple 1
   Code CC/KE
   Naval Postgraduate School
   Monterey, CA 93943

4. Professor Gary Porter 1
   Code CC/PO
   Naval Postgraduate School
   Monterey, CA 93943

5. Chairman, C4I Academic Group 1
   Code CC
   Naval Postgraduate School
   Monterey, CA 93943

6. Mr. Arison 1
   JCS/J6
   Pentagon
   Washington, D.C. 20350-2000

7. Capt Maslowsky 1
   CNO/N62
   Pentagon
   Washington, D.C. 20350-2000

8. Lt Col Ziegenfuss 1
   HQMC C4I Plans and Policy Division
   2 Navy Annex
   Washington, D.C. 20380-1775

9. Lt Col Jacques 1
   HQ AFCIC/XPF
   Pentagon
   Washington, D.C. 20350-2000

10. Jerry Dussault.....1
AFRL/IFSE
525 Brooks Rd.
Rome, NY 13441-4505

11. Dr. Heather Dussault.....1
AFRL/IFS
525 Brooks Rd.
Rome, NY 13441-4505

12. Capt Paul Webster.....2
3329 Misty Cove Circle
Baldwinsville, NY 13027