Theses and Dissertations   1. Thesis and Dissertation Collection, all items

2004-03

Integrating XML and RDF concepts to achieve automation within a tactical knowledge management environment

McCarty, George E., Jr.

Monterey, California. Naval Postgraduate School

http://hdl.handle.net/10945/1648

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

INTEGRATING XML AND RDF CONCEPTS TO ACHIEVE AUTOMATION WITHIN A TACTICAL KNOWLEDGE MANAGEMENT ENVIRONMENT

by

George E. McCarty, Jr

March 2004

Thesis Advisor: Man-Tak Shing
Second Reader: Michael Crawford

THIS PAGE INTENTIONALLY LEFT BLANK

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

13. ABSTRACT *(maximum 200 words)*

Since the advent of Naval Warfare, Tactical Knowledge Management (KM) has been critical to the success of the On Scene Commander. Today's Tactical Knowledge Manager typically operates in a high stressed environment with a multitude of knowledge sources including detailed sensor deployment plans, rules of engagement contingencies, and weapon delivery assignments. However the WarFighter has placed a heavy reliance on delivering this data with traditional messaging processes while focusing on information organization vice knowledge management. This information oriented paradigm results in a continuation of data overload due to the manual intervention of human resources. Focusing on the data archiving aspect of information management overlooks the advantages of computational processing while delaying the employment of the processor as an automated decision making tool.

Resource Description Framework (RDF) and XML provide the potential of increased machine reasoning within a KM design allowing the WarFighter to migrate from the dependency on manual information systems to a more computational intensive Knowledge Management environment. However the unique environment of a tactical platform requires innovative solutions to automate the existing naval message architecture while improving the knowledge management process. This thesis captures the key aspects of building a prototype Knowledge Management Model while providing an implementation example for evaluation. The model developed for this analysis was instantiated to evaluate the use of RDF and XML technologies in the Knowledge Management domain. The goal for the prototype included:

1. Processing required technical links in RDF/XML for feeding the KM model from multiple information sources.

2. Experimentation with the visualization of Knowledge Management processing vice traditional Information Resource Display techniques.

The results from working with the prototype KM Model demonstrated the flexibility of processing all information data under an XML context. Furthermore the RDF attribute format provided a convenient structure for automated decision making based on multiple information sources. Additional research utilizing RDF/XML technologies has the potential of enabling the WarFighter to effectively make decisions under a Knowledge Management Environment.

14. SUBJECT TERMS Resource Description Framework (RDF), Extensible Markup Language (XML), Emission Control (EMCON), Task Force (TF) Web, User Datagram Protocol (UDP), Forward Error Correction (FEC), Knowledge Management (KM)			**15. NUMBER OF PAGES** 119
			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

THIS PAGE INTENTIONALLY LEFT BLANK

**INTEGRATING XML AND RDF CONCEPTS TO ACHIEVE AUTOMATION
WITHIN A TACTICAL KNOWLEDGE MANAGEMENT ENVIRONMENT**

George E. McCarty, Jr.
Civilian, SPAWAR Systems Center San Diego
B.S., National University, 1998

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN SOFTWARE ENGINEERING

from the

**NAVAL POSTGRADUATE SCHOOL
March 2004**

Author:　　　　　George E. McCarty, Jr.

Approved by:　　　Man-Tak Shing
　　　　　　　　Thesis Advisor

　　　　　　　　Michael Crawford
　　　　　　　　Second Reader

　　　　　　　　Peter J. Denning
　　　　　　　　Chairman, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Since the advent of Naval Warfare, Tactical Knowledge Management (KM) has been critical to the success of the On Scene Commander. Today's Tactical Knowledge Manager typically operates in a high stressed environment with a multitude of knowledge sources including detailed sensor deployment plans, rules of engagement contingencies, and weapon delivery assignments. However the WarFighter has placed a heavy reliance on delivering this data with traditional messaging processes while focusing on information organization vice knowledge management. This information oriented paradigm results in a continuation of data overload due to the manual intervention of human resources. Focusing on the data archiving aspect of information management overlooks the advantages of computational processing while delaying the empowerment of the processor as an automated decision making tool.

Resource Description Framework (RDF) and XML provide the potential of increased machine reasoning within a KM design allowing the WarFighter to migrate from the dependency on manual information systems to a more computational intensive Knowledge Management environment. However the unique environment of a tactical platform requires innovative solutions to automate the existing naval message architecture while improving the knowledge management process. This thesis captures the key aspects for building a prototype Knowledge Management Model and provides an implementation example for evaluation. The model developed for this analysis was instantiated to evaluate the use of RDF and XML technologies in the Knowledge Management domain. The goal for the prototype included:

1. Processing required technical links in RDF/XML for feeding the KM model from multiple information sources.

2. Experiment with the visualization of Knowledge Management processing vice traditional Information Resource Display techniques.

The results from working with the prototype KM Model demonstrated the flexibility of processing all information data under an XML context. Furthermore the RDF attribute format provided a convenient structure for automated decision making

based on multiple information sources. Additional research utilizing RDF/XML technologies will eventually enable the WarFighter to effectively make decisions under a Knowledge Management Environment.

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

Many individuals deserve a great deal of credit for any achievements associated with this work as well as the completion of a major milestone in my career. First of all, my Mom sacrificed her entire life for my education and this work is dedicated to her memory. Also I would not have made it this far without the support of my wife and our loyal companion, Nick. There are many other special folks that deserve recognition as follows.

For the patience of all my existing and former bosses, I am eternally grateful.

For my present and former employees, who contributed that extra effort during my absences, I am in their debt.

To the professors, technical advisors and experts who gave of themselves for my benefit. Without all of you, this would not have been possible.

To my children, who provided a source of inspiration as well as an opportunity to demonstrate leadership by example.

Finally for all the men and women of the Submarine Force who have been a special part of my life for the past 38 years. THANK YOU!!!

THIS PAGE INTENTIONALLY LEFT BLANK

I.   INTRODUCTION

Since the advent of naval warfare, the challenge for the Commanding Officer has been to expediently employ both tacit and explicit knowledge in pursuit of a tactical advantage. The source of this knowledge has come in many forms including data and information flow. However despite many advances over hundreds of years, the manipulation of both knowledge and information has been a relatively slow manual process.

Progress in this area, when measured over time, can be represented by terms used to describe the naval information flow systems. Prior to the electronics era, signaling was a primary means of information exchange and was represented by systems such as flag hoist and semaphore. Relatively slow from a data flow standpoint, signaling provided reliable connectivity between units within line of sight but lacked long range capability. However the advent of electronics brought many new changes including the ability to communicate when out of line of sight. The initial phases of the electronics era included the invention of Morse code, which brought Communications in the form of radio frequency transmissions and flashing light.

Over the years despite improvements in both transfer speed and transmission types, the basic Communications Architecture has remained essentially the same for about 50 years. Then around 1980, Command and Control were combined with Communications to form C3 thus representing one of the initial links between information flow and the decision making process. Although not formally associated with Knowledge Management (KM), the combination of Control, Command and Communications (C3) included all the aspects of the knowledge process including creation, management and exchange. Furthermore as discussed below, KM processes were very relevant prior to the genesis of C3.

By definition [1], KM is the systemic and organizationally specified process for acquiring, organizing, and communicating knowledge of employees so that other employees may make use of it to be more effective and productive in their work. While the WWII Tactical Commander would not have used the term Knowledge Management

to describe Warfare Planning and Execution, the process was just as relevant in those days as it is now. Furthermore the art of warfare has been heavily influenced by information processing for Knowledge Management purposes. Therefore if one accepts the theory that KM has been an element since the inception of warfare, then over time the Knowledge Management Process could be used as a model to evaluate the strengths and weaknesses of warfare execution.

Since World War II, Knowledge Management was heavily influence by information processing since most Naval Forces utilized the Naval Message as the prime delivery of critical warfare directives. The importance of this text-based document has been demonstrated by events such as the Battle of Midway whereby the United States Navy maintained a tactical advantage through surreptitious entry into the Japanese Naval Message System. Even today, under limited bandwidth or Emission Control (EMCON) conditions, the naval message remains the optimum method of information delivery.

Today's Naval Command, Control, Communications and Computer (C4) Architecture still reflects much of the past configuration including the Naval Message System that combines or stovepipes the application/transport/network/physical layer into a stand-alone architecture. However, in a Network Centric Warfare environment, the WarFighter does not have the luxury of reviewing thousands of messages to make a time critical decision. Only recently have the advances in technology resulted in the delivery of Internet Protocol (IP) capability allowing the Naval WarFighter to engage network centric warfare as a tactical advantage

Furthermore, the implementation of secure web based access as well as a continuation of traditional message based data requirements has created an information overload situation complicating the organization's Knowledge Management practices. Fortunately, recent improvements in software technology may provide solutions for reduction of data overload as well as the integration of disadvantage messaging or limited bandwidth users in a network centric environment.

Now that Network Centric Warfare is potentially a reality for all platforms, the challenge is to adapt and implement new technologies that provide a tactical advantage vice further overloading the Commander's KM processing. The WarFighter's focus must

shift from information processing to knowledge management. Maintaining the existing Naval Message oriented architecture in a Network Centric environment will not enable the transformation from a legacy information based configuration to a knowledge based IP oriented design. Now is the time to demonstrate that the Tactical Naval Commander will be able to employ tools such as Extensible Markup Language (XML) and Resource Description Framework (RDF) as knowledge enablers for both IP and Naval Message based configurations.

The XML and the RDF models display both file type translation and text-based properties that readily apply to a transformation in message processing. Additionally the power of the Resource Description Framework, implemented within XML, has already demonstrated a potential to improve web based data processing while selectively providing the means to screen the desired database information.   Typically associated with Semantic Web Technology, these models possess the software power to improve the implementation of the WarFighter's Knowledge Management capability and transform the architecture of Network Centric Warfare. The remainder of this document will be focused on applying XML and RDF to accelerate the transition process.

THIS PAGE INTENTIONALLY LEFT BLANK

II.   BACKGROUND

A.   LEGACY AUTOMATION VIA EMERGING TECHNOLOGIES

Legacy is a term that is thrown around these days and tends to apply to any naval information system or architecture that is not IP oriented. It would seem that this term does a disservice to many viable and IP capable programs while signaling the death sentence for some very good technology. Then why automate legacy? Bottom line is that the number of platforms and amount of hardware currently deployed does not lead to a total transformation overnight. It has been demonstrated that a viable transition plan is required to ensure that all Naval Forces, including Allies, have the ability to fight in this Coalition oriented environment.

The next question would be what should be automated? A list of systems could be provided but, in practical terms, all fall under the classification "Information Exchange Systems" or IXS as known in the programmatic world. Most of these systems have the similar qualities for example:

- Relatively low data rate in the order of 2400 or 4800 bits per seconds

- Well established, reliable and generally trouble free

- Single Path

- Unique efficient protocols designed to provide maximum compression

- Low overhead

- Covert, Emission Control (EMCON) or One Way Broadcast capability included

- Unique Radio Frequency Band Assignment (i.e. Ultra High Frequency (UHF))

- Satellite Capable

- Stand Alone with dedicated hardware and resource assets

- Fixed data rate resulting in an inefficient use of available bandwidth

- Manually monitored

- Designed to support NAVAL MESSAGING

- Specified data format

- Software design based on 1970s technology

In contrast these same requirements supported by the equivalent IP architecture would have the following characteristics:

- High data rate (limited by connectivity bandwidth)

- Complex but mostly reliable

- Multiple Paths

- Common Protocols for universal access

- High Overhead, Inefficient in Limited Bandwidth Situations

- Multicast capable but not designed for EMCON

- Radio Frequency Independent subject to bandwidth limitations

- Satellite Capable

- Hardware/Resource Independent

- Automated Monitoring

- Electronic data exchange oriented

- Format Independent

- Based on most current technology

While this legacy/IP comparison addresses several important aspects, it does not provide sufficient details to formulate an automation strategy. Therefore identification of Naval Tactical Information Processing requirements are essential to proceed with an automation strategy.

The following requirements were specified by a platform sponsor as being critical to future information processing:

- EMCON Requirement – UDP Based Protocol

- Support of Naval Messaging

- Policy Specifications – Automated Implementation of Doctrine

- Limited Bandwidth – Efficient Transfer of Data

- Task Force Web Implementation – Naval Web Based Application

Although it is too early to draw conclusions, from the legacy/IP comparison outlined above, legacy appears to provide better support under limited bandwidth, Naval Messaging and EMCON conditions, whereas automation and web requirements are better supported by IP and larger bandwidth conditions. From a strategy standpoint it is clear that IP implementation needs to address the legacy strengths and therefore one focus of this study will be to address the Naval Messaging requirement as it relates to automation, the web and the ability to more efficiently implement IP and improve Tactical Knowledge Management. The next several paragraphs will provide more detail on Naval Messaging and specifically the impact on Knowledge Management.

1. Background Information
a. *Naval Messaging*

Since the advent of electronic warfare the Naval Message has been essential for transmitting textual data in a standardized format that was easily interpreted by receiving Naval Commanders. As information flow between the various military services became more common and with a focus on joint missions in the early 1970s, the message format also became more standard resulting in the establishment of the US Message Text Format (USMTF). As the joint standard for message-based information exchange, USMTF enabled efficient and effective employment of US forces in joint and combined operations. This information standard has been an agreed set of character oriented message formats, protocols, vocabulary, and procedures which enhance command, control, communications, computers, and intelligence (C4I) interoperability.

Current USMTF users include all Department of Defense (DoD) services/agencies, all Joint Commands and most Allied Nations. Therefore adapting the USMTF product to an IP based system has been a high interest item and under close scrutiny by various organizations. A drawback into transitioning the USMTF system into

an IP environment has been the substantial economic investment made into existing legacy hardware that remains in War-fighters inventory. However the decreasing cost of IP hardware along with innovative software solutions has the potential to overcome this current impasse. One of the most promising efforts has been an XML-to-USMTF-to-XML (XML2MTF) prototype software capability which will be discussed in more depth later in this paper. However, the translation of USMTF documents to an XML format does not imply a Knowledge Management capability but does provide an enabler for migration into a KM environment. The following paragraph provides a discussion on the scope of Knowledge Management implementation in the process of establishing a Knowledge Centric Organization.

b.   *Knowledge Management (Ontology for a Military Domain)*

What is Knowledge Management and why is it important to the modern WarFighter? Broken down to simple terms, Knowledge Management is about delivering the right knowledge to the right people at the right time [2]. For example, putting weapons on target is all about delivering the right knowledge, such as weather conditions, target location, etc. to the War-fighter at the desired moment of weapons launch. Therefore if one accepts the theory that Knowledge Management is critical to the WarFighter; the question then becomes where do we go next? The Department of the Navy's Chief Information Officer (DoNCIO) has taken the lead towards Knowledge Management implementation with the development of a Information Management/Information Technology Strategic Plan to build a knowledge sharing culture and exploit new IT tools to facilitate knowledge transfer across the organization. In support of the DoNCIO's initiative Dr. Geoffrey Malafsky (Tech I2 LLC) has captured the problem with the following excerpt from his technical paper titled Knowledge Taxonomy: "Achieving Knowledge Superiority, both for the War-fighter and support forces, requires us to capture, organize and disseminate critical knowledge in a timely and succinct manner. We cannot merely expand access to knowledge, information, and data by building large repositories, since without a clear and easy method to find exactly what people need at any given moment our forces will continue to succumb to information overload and not achieve the objectives of Knowledge Superiority." Doctor Malafsky goes on later in his paper to say that a key part of the Knowledge Management

strategy is the methods and tools used to organize and classify the vast volume of Knowledge Information Data throughout the Department of the Navy enterprise. Therefore the purpose of this document is to examine software tools with the capability to provide potential solutions for achieving Knowledge Superiority by supplying the WarFighter with the right information at the right time.

2.   Introduction of Supporting Technology

During recent conflicts, the use of leading edge technology has been demonstrated as a key enabler for maintaining military superiority. Focusing on the software aspect of this enabler, it has been necessary to develop an in-depth understanding of several key technologies in order to demonstrate the potential of shifting the military domain from a Naval Messaging to a Knowledge Management environment. In the thesis development phase, this research has been able to take advantage of public available development code and applications to evaluate use of new technology for implementation in non-traditional military applications. The remainder of this paragraph describes some of the key technologies used contributing to this study.

a.   JAVA Development Tools

During the initial phases of research for this thesis, it became apparent that many knowledge management tools relied on JAVA technology to provide the fundamental elements for development and execution of the desired application. Java has gained popularity as a general-purpose programming language and Sun's supporting JAVA technology has become increasing capable as evidenced by the following products:

- Java 2 Platform, Standard Edition (J2SE)

- Java Development Kit (JDK)

- Java Runtime Environment (JRE)

- Java Web Services Development Pack

- Sun ONE Studio (formerly Forte for Java)

- Community Edition

- IDE includes XML support, GUI Editor, & Source Editor

At a very reasonable cost ($0.00) to the graduate student, the author is indebted to SUN for allowing the download of leading edge tools, source code and accompanying tutorials that were employed in this project. For example, the Sun ONE Studio Integrated Development Environment (IDE) provided a platform comparable with that of a fully functional commercial IDE. One major technical advantage of employing the Sun IDE is the embedded XML module that supports all related functions required for processing XML documents. It also supported creation of the following related documents:

- Document type definition (DTD) files
- Cascading style sheets (CSS)
- Extensible style sheets (XSL)
- SAX and DOM classes
- XHMTL documentation

In addition, Sun's IDE provided XML file editing by the following methods:

- Source Editor enabled manipulation of text files.
- XML Editor allowed creation and editing of the XML documents hierarchical set of nodes and node properties.

The IDE also allowed validation and format checking of the XML document to a basic set of grammatical rules, including:

- Every start tag must have an end tag.
- Elements cannot overlap.
- There must be exactly one root element.
- Attribute values must be quoted.
- An element may not have two attributes with the same name.
- Comments and processing instructions cannot appear inside tags.

- No unescaped <or &signs can occur in the element's or attribute's character data.

Additional features that made the SUN IDE a full package included the ability to create JAR files and the ability to draw on the resident functionality of the Java Runtime Environment and embedded DOM/SAX parsers.

Several other JAVA based tools utilized in this research included:

ISAViz: Visual authoring and browser tool for the Resource Description Framework. ISAViz is a Java tool that provided a visual interface for browsing and creating RDF models represented as graphs. The tool allowed RDF models to be imported and exported from both the RDF/XML and N-triple syntaxes. Graphs could also be exported as SVG and PNG files. This tool will be discussed in further detail later in this paper.

Protégé-2000:   Integrated Java tool used for the creation of customized knowledge-based tools. Protege-2000 provided an extensible architecture for employment as a Semantic Web language editor capable of knowledge modeling and acquisition. This tool will be discussed below under the Ontology Tools paragraph.

b.   *eXtensible Markup Language (XML)*

What is XML? A World Wide Web Consortium (W3C) search will yield responses such as "XML is a meta-language" or "XML is a flexible text format" or "XML is a standard that provides the context for transforming data into information". Fulfilling various roles for multiple users XML is a powerful textual based software tool that provides data structure for information exchange among systems having dissimilar internal formats. Recognizing the power of XML, DOD and the U.S. Navy have been very proactive towards implementation and standardization of this text based software capability. As quoted from the Joint Chiefs of Staff (JCS) MSG 121615Z APR 99:

XML IS A NEW INTERNATIONAL INDUSTRY STANDARD
APPROVED BY THE WORLD WIDE WEB CONSORTIUM (W3C) IN
FEB 98 FOR DESCRIBING AND SHARING STRUCTURED
INFORMATION. XML IS PLATFORM INDEPENDENT AND

ALLOWS OPERATORS USING DIFFERENT COMPUTER
HARDWARE, SOFTWARE, DATABASES AND
COMMUNICATIONS PROTOCOLS TO EXCHANGE
INFORMATION. INDUSTRY LEADERS ARE EMBRACING AND
IMPLEMENTING XML IN THE MOST POPULAR OFFICE
APPLICATIONS, WEB BROWSERS, DATABASES AND
OPERATING SYSTEMS. DATA CONTENT IS SEPARATED FROM
ITS PRESENTATION FORMAT, ENABLING OPERATORS TO
DEFINE CUSTOMIZED VIEWS OF DATA TAILORED TO SUPPORT
SPECIFIC WARFIGHTER NEEDS.

As defined in the U.S. Navy's policy statement # 20 of 2002, the overall
goals of Navy XML policy are to:

- Encourage and promote the use of XML as an enabling technology to help
  achieve enterprise interoperability

- Establish multiple assets that will assist the Navy in adopting and
  implementing XML

- Support interoperability between the Department of the Navy (DON),
  DOD and other agencies

- Actively influence XML and XML related standards bodies to facilitate
  the creation and adoption of XML specifications that support DON
  requirements.

The fundamental premise of this research has been the application of
XML as a universal translator in support of Knowledge Management architecture design
and development. While this role does not mirror the traditional web based XML
implementation, it does draw on development efforts within the DOD and more
specifically the U.S. Navy.

Figure 1 below provides a representation of the potential conversion
power of XML:

XML TRANSLATOR

DATA TYPE

OUTPUT
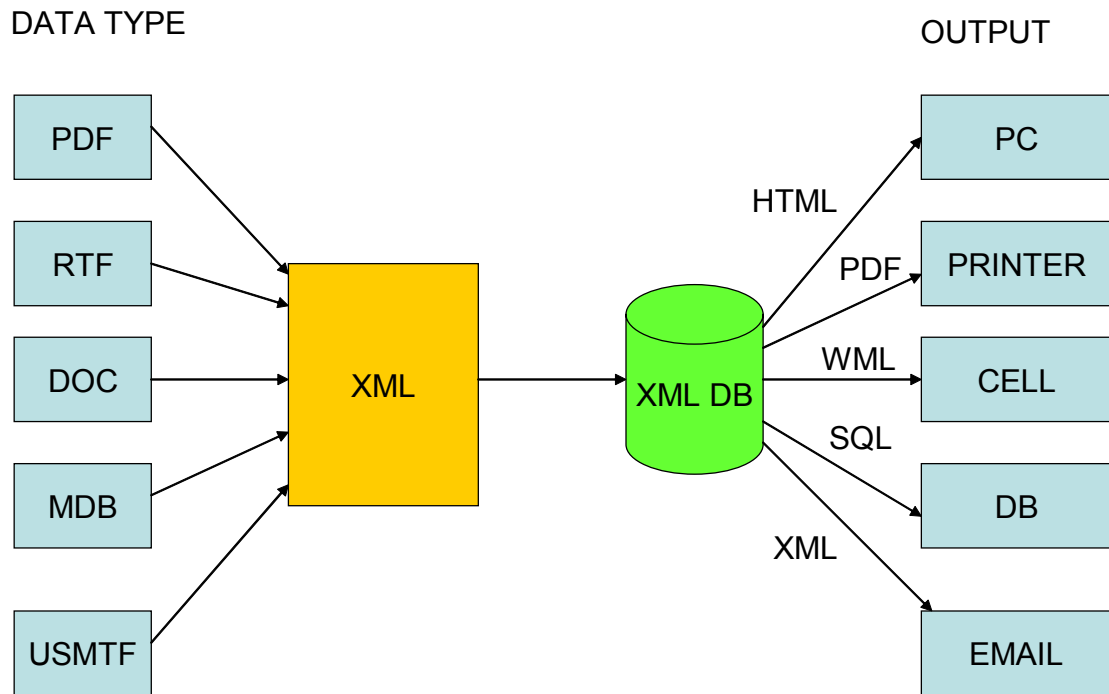
Figure 1.   XML Conversion Strategy ([After Ref. [3].)

The USMTF type conversion to XML will be discussed in depth under the chapter titled XMLMTF. Providing a foundation for semantic web development, XML also acts as a building block for new technologies such as the Resource Description Framework, which is discussed in the next paragraph.

c. *Resource Description Framework (RDF)*

Just as with XML, a web search will reveal several definitions for RDF such as "methodology", "framework", "metadata standard" and "XML based markup language". The strength of RDF is reflected in the definition that includes metadata or more simply defined as data about data or information about information. Created in 1997, the RDF model initiated a process whereby a distinction could be drawn between data and metadata. This was important as XML provided the structure to accommodate syntax requirements but XML did not adequately address the issue of the semantic capability required for development of a smart web. The creation of RDF opened up the potential of addressing semantics by allowing metadata management with a common vocabulary. However, as XML and RDF are viewed as two separate activities [4], there remains a considerable amount of effort to ensure that future development is unified to enable the realization of a semantic web. Figure 2 below proposes a foundation for the semantic web and is based on a similar layered approach presented by Tim Berners-Lee during XML 2000. This representation builds on a XML foundation with a goal of achieving a logical web through the capabilities provided by RDF and Ontology, the subject of the next paragraph.

14
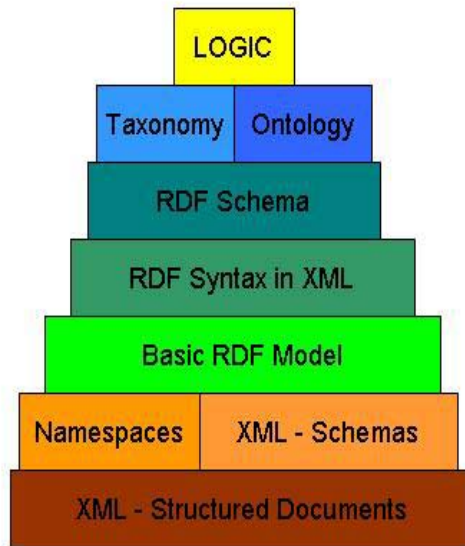
Figure 2.   Syntactic & Semantic Development Model ([After Ref ([5].)

As previously mentioned one of the key contributions to this research has been the availability of toolsets required for implementation at each level of a Figure 2 layer. In pursuit of a providing a visual model, the author was able to obtain ISAviz, a RDF Modeling tool, from the following URL: [http://www.w3.org/2001/11/IsaViz/]. Feb 2004.

It should be noted that a separate product, Graphviz, also requires downloading and installation to enable viewing of the ISAviz modeling window. However both of these tools were available without cost and functioned properly in a Windows XP system. Discussed briefly in the Java Development tools paragraph, ISAviz's visual environment represents RDF Models as directed graphs. The following figure depicts the tools desktop environment:
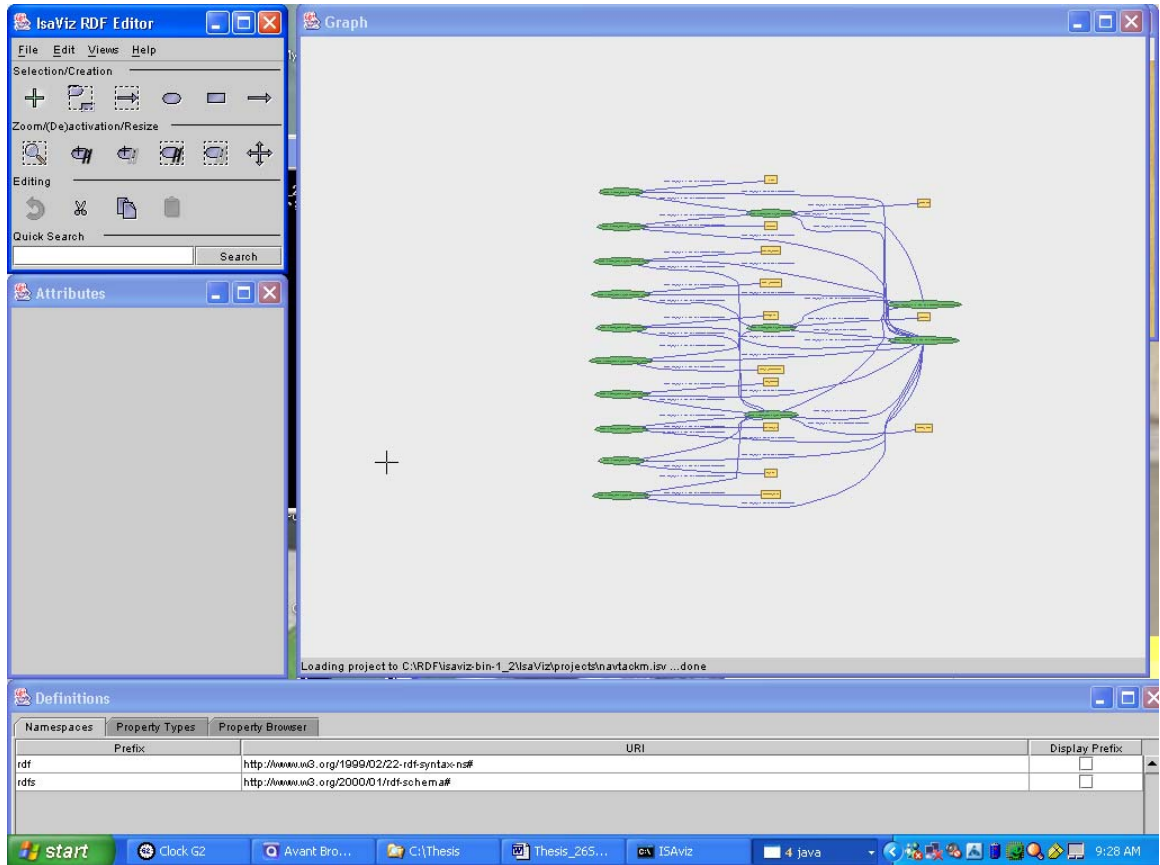
Figure 3.   ISAviz (RDF Modeling Tool)

The nodes of the Figure 3 graph correspond to resources (ellipses) and literals (rectangles) while graph edges symbolize properties. The ISAviz environment [6] consists of four main windows as follows:

- ISAviz RDF Editor: Contains main command window with a palette of available tools.

- Graph: The RDF Model is represented as a 2D graph and uses a ZTVM view displaying a region of infinite virtual space as seen through a camera.

16

Movement of the camera through the visual space is enhanced by an altitude change capability that results in a 2.5D Graphical User Interface with super zooming capabilities.

- Radar View: Graph overview in a separate window with an embedded rectangle outlining region corresponding to large graph zoom view.

- Property Browser: A textual browser that displays detailed properties of the selected resource and includes a hyperlink capability.

The RDF Models displayed in this thesis are built with the ISAviz tool and the resulting graphs displayed as a figure later in the paper. The next paragraph will go into further detail for a similar tool used in development of the ontology architecture.

d. Ontology Tools

What is ontology and why is ontology important? As was experienced with XML and RDF, there is no shortage of definitions for ontology. For the purpose of this research, a brief and very accurate description [7] refers to ontology as a specification of a conceptualization or in more detail as a formal specification of the concepts and relationships written as a set of definitions of formal vocabulary. While the definition and application can be adapted to varied environments, the effective implementation of knowledge management by the WarFighter will depend on a successful ontology design. Therefore the importance of ontology will be to address the military domain's knowledge sharing complexities by the inclusion of a metadata schema for use in a stressed environment. The recent maturity of Ontology Tools [8] has not yet addressed the manual nature of ontology acquirement and in the current toolset there remains an inability to rapidly redefine the knowledge base. However one of the more promising leading efforts in this area has been the Protégé-2000 Project ([http://protege.stanford.edu]. Feb 2004) developed by Mark Musen's group at Stanford Medical Informatics. This tool may be downloaded free of charge from the following URL: [http://protege.stanford.edu/download/release/index.html]. Feb 2004.

It is highly recommended that several of the additional plug ins such as OWL, RDF and OntoViz are downloaded from the URL. These tools as well as the users guide will enable the recipient to more effectively construct an ontology model. As extracted from the users guide: "Protégé-2000 is an integrated software tool used by system developers and domain experts to develop knowledge-based systems. Applications developed with Protégé-2000 are used in problem-solving and decision-making in a particular domain." The Protégé desktop environment is illustrated in the following figure.
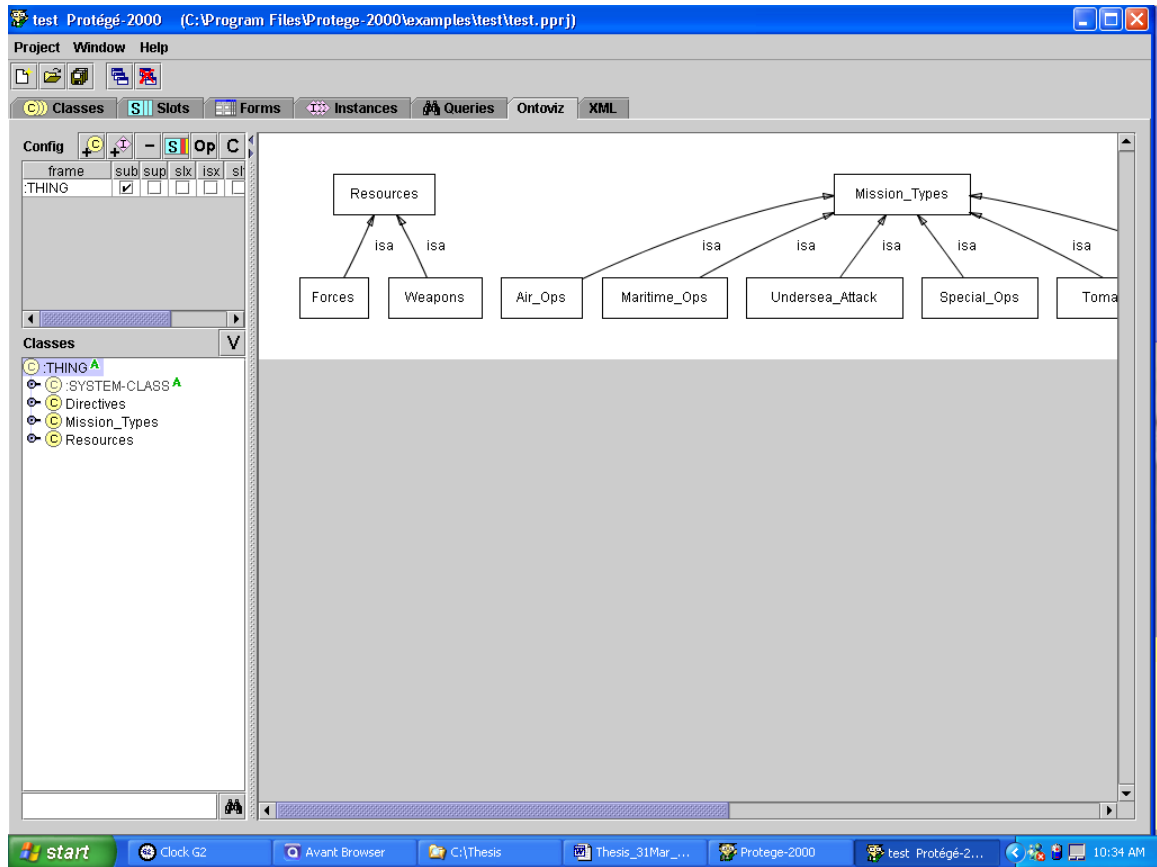
Figure 4.   Protégé 2000 (Ontology Modeling Tool)

The Protégé-2000 software provides a Graphical User Interface with a row of selectable tabs allowing integration of the following capabilities [9]:

- Modeling of an ontology of classes describing the domain
- Creation of a knowledge-acquisition tool for knowledge collection
- Entry of specific data instances and creation of a knowledge base
- Application execution

The class modeling is significant, as classes become the focus of most ontology's. Furthermore there are several steps towards ontology development including the following:

- Ontology class definition

19

- Class arrangement in a taxonomy

- Slot definition and values description for the slots

- Assignment of slots values for the instances

The knowledge base is then created [9] by defining individual instances of these classes filling in specific slot value information and additional slot restrictions. A step forward in domain ontology development, Protégé-2000 promotes the reuse domain ontology and problem solving methods. This recycling approach [10] enables multiple applications to use the same domain ontology to solve different problems and the same problem solving method for different ontology's. Reuse of existing products will result in a reduction of development time and less program maintenance.

The background information presented above was designed to briefly address legacy automation as well as emerging technologies capable of providing solutions. The Process section will address requirements and specific implementations envisioned as potential capabilities in the tactical knowledge management environment.

III.   PROCESS

In this section, the analysis will begin to investigate the integration of developmental and future concepts into the WarFighter's knowledge management architecture. As with any software engineering development, the first phase of the process is the identification of requirements, which is just as significant in the military domain as it is in the commercial market. Recently, several naval platform sponsors have instituted an assessment process that is focused on matching operational missions with information technology capabilities to develop a requirements matrix. The results of this requirements identification process will be identified in paragraph A below. In addition, this section will include a review of developments potentially capable of enabling a swifter transition towards the Tactical Knowledge Management architecture.

A.   IDENTIFY TACTICAL NAVAL REQUIREMENTS

The goals for Knowledge Management within the Department of the Navy can be obtained from the Chief Information Officer's (CIO) website which provides the following summary:

Information technology (IT) and information management (IM) are essential but insufficient to achieve information superiority. Knowledge Management (KM) offers the potential to significantly leverage the value of our IT investment. It is the link between technology and people.

Based on this guidance, an implied requirement of achieving Knowledge Superiority has been specified. This direction also implies that Knowledge Management will be the enabler to help achieve the knowledge superiority requirement. Although KM guidance for the Tactical Commander remains to be explicitly specified, by definition the inherited requirement of knowledge superiority will be necessary to get the right knowledge for the right tactical situation at the right time. The challenge will be to translate the Tactical Naval Commander's Knowledge Management requirements into an

architecture that supports all war fighting domains. The DON CIO site also offers the following concise depiction for the relationship between Knowledge Management and the Naval Service:

What is Knowledge Management in the DoN?

Value, Relevancy, Currency

CONTENT

Enabling, Facilitating, Promoting Innovations

TECHNOLOGY

PROCESS

Making Explicit, Capturing, Categorizing, Clumping, Synchronizing, Analyzing, Disseminating

"Balanced" Knowledge Management

LEARNING

CULTURE

Building Context, Creating, Growing, Thinking Strategically

Commitment, Sharing, Exchanging, Building Relationships

Figure 5.   DON CIO's Guidance on KM and the U.S. Navy ([From Ref. [11].)

This diagram captures the essence of Knowledge Management not only for the Navy but also for a much broader application. The next section will examine potential developments related to establishment of the required architecture.

B.   SPECIFY RELATED POTENTIAL DEVELOPMENTS

A significant shortcoming towards arriving at a common architecture is the inability to maintain a cohesive approach during the transition process. This lack of focus is traditionally based on historical acquisition strategies as well as a fragmented funding process. Therefore the diverse program sponsors typically focus on fulfilling individual needs vice embarking on a coordinated end to end development approach. The advantages and disadvantages of this strategy result in independent solutions to the same problem. While inefficient from a funding aspect, the ability to evaluate multiple developments will most likely result in more capability as well as a better solution for the WarFighter. This independent acquisition process is especially true in environments where a multitude of unique platforms are involved. Transitioning from legacy messaging towards a Network Centric Architecture has been especially difficult in a tactical environment. Furthermore full implementation of Tactical Knowledge Management at the platform level is not feasible until issues such as bandwidth limitations, IP overhead and Emission control solutions are addressed. Seeing as this transition effort remains work in progress, the developments discussed below provide a direction rather than a final design for this difficult problem. The following analysis will review the contributions that this areas are envisioned to contribute to the transition process.

1.   IP Based Messaging

Although IP messaging could apply to numerous developments, for discussion purposes this review will focus on the Defense Messaging System (DMS). Baselined in 1989, DMS was envisioned to modernize DoD messaging. The following DMS description was extracted from Naval Tactical Publication NTP-21(A) dated June 1997 ([http://www.nctsfe.navy.mil]. Feb 2004).

> The DMS Program encompasses the hardware, software, procedures, personnel, and facilities required for electronic delivery of messages among organizations and individuals in the DOD. It also includes interfaces to tactical, afloat, and Allied systems. The DMS Program is under the oversight of the Major Automated Information System Review Council (MAISRC). MAISRC's milestones dictate that DMS be

implemented progressively, first with the deployment of transitional components, then the deployment of components to provide Unclassified-but-Sensitive (SBU) messaging, and finally the deployment of components to provide Classified messaging. The DMS shall provide message service to all DOD users, to include deployed tactical users, access to and from worldwide DOD locations, and interface to other US government, allied, and Defense contractor users as needed. To minimize delay, this service shall be direct to the end user whenever possible. The DMS shall reliably handle information of all classification levels (unclassified to TOP SECRET), compartments, and handling instructions. In addition to maintaining high reliability and availability, the DMS must interoperate with current message systems as it evolves from the current configuration to full implementation. The DMS shall be a vehicle for planned growth and technology enhancement that does not exist today. It shall be based upon the principles of standardization and interoperability, while preserving adaptability to implement Service and agency unique functions. The major elements of the current collection of subsystems upon which the DMS will be built include the Automatic Digital Network (AUTODIN) system (including tactical and base level support systems) and the electronic mail systems on the DOD internet (principally within the Defense Data Network (DDN) and associated local area networks (LANs)). DMS is standards-based and adheres to X.400 and X.500 international standards with approved extensions to meet military messaging requirements. These military messaging requirements have been accepted and approved by the U.S. allies and are formally approved in Allied Communications Publication 123 (ACP 123). DMS provides a uniform, seamless messaging system with full interoperability among the messaging assets of all DOD parties.

While this description would leave one to believe that DMS is the final IP messaging solution for tactical forces, the implementation process reveals a different picture. One of the first shortcomings of DMS was its inability to meet stringent time delivery and precedent requirements for certain classes of organizational messages. These deficiencies are still being addressed at senior DOD levels and hybrid arrangements are envisioned for the near future. At the tactical level, DMS has a fundamental major deficiency due to a requirement for higher bandwidth than normally available for several platform types. This limitation is being addressed by various sponsors with different strategies. However, a full end-to-end network centric architecture has not yet been finalized and many issues remain to be resolved. From a technical aspect the DMS Program consists of 3 systems:

- Message Transfer System (MTS)

The MTS contains a three-tier design of Message Transfer Agents (MTA) that include Subordinate Message Transfer Agents (SMTA), Intermediate Message Transfer Agents (IMTA) and Backbone Message Transfer Agents (BMTA). Subordinate Message Transfer Agents (SMTA) are at the lowest level of the tier architecture. SMTAs have connectivity with the Intermediate Message Transfer Agent (IMTA) but will not normally have peer-to-peer (SMTA-SMTA) connectivity. The next level up is the Intermediate Message Transfer Agent (IMTA). Similar to the SMTA, the IMTA will not normally have peer-to-peer connectivity. As described previously, the IMTA will support one or more SMTAs and will also connect with the upper level of the tier or the Backbone Message Transfer Agent (BMTA). Typically, the SMTA resides at the organizational level (i.e. military command) whereas the BMTA and IMTA fall under the regional node.

- Message Handling System (MHS)

The Message Handling System is responsible for the preparation, receipt and transmission of messages within the DMS. The MHS [12] consists of the Message Transfer Agents (MTAs), User Agents (UAs), Message Stores (MSs), Profiling User Agents (PUAs), Mail List Agents (MLAs), and the Multi-Function Interpreters (MFIs). The MHS employs the messaging directives and protocol derived from the ITU-T X.400 recommendation.

- Directory Services

Based on the ITU-T X.500 recommendation, DMS employs a distributed hierarchical X.500 Directory distributed among a number of components, called Directory System Agents (DSAs). DSAs are at two levels; a Global and a Local level. The upper portion of the Directory Information Base (DIB) is contained in the Global DSA at DISA Head-quarters, with the other Global DSAs containing copies of the information. Individual DMS components, such as the Directory User Agents (DUAs), will not normally access the Global DSA, which contains the highest levels of the Directory structure.

The combination of the three Transfer, Handling and Directory Systems have been planned to provide a worldwide capability and enable a decommissioning of the existed legacy Autodin Messaging system. Although shortfalls, identified above, will slow the transition from a legacy architecture, the overall system has demonstrated the potential to enable activation of the IP messaging capability ASHORE. The real challenge begins where DMS ends and Afloat Tactical Messaging begins. Similar to tackling world hunger, the mechanism to transfer both messaging data and overhead associated with an IP system is not a trivial matter. For example, most of the existing legacy naval tactical messaging systems exchange data at 2400bps or 2.4kps and previous attempts at prototyping an equivalent IP capability have received lukewarm support from most War Fighters. Therefore most efforts have since focused on expanding the existing bandwidth through techniques such as asymmetric connectivity or shift to other Radio Frequency (RF) spectrums such as Super High Frequency (SHF). While these efforts are enabling bandwidths from 32K to 256K, the challenge of implementing a fully capable network centric capability for all platforms has not yet been achieved. However in the quest of Tactical Knowledge Management, many hurdles have been overcome. Most notable there now exists an IP EMCON broadcast where as covert units maintain their status with a Space and Naval Warfare Systems Center San Diego developed system known as ISDS or the Information Screening and Delivery Subsystem. The metrics for ISDS will be examined in a discussion later in this paper. Other challenges such as DMS to legacy interface will also be reviewed. In summary, organizational IP messaging is now available and will increasingly become more capable of fulfilling gaps in the attainment of a Tactical Knowledge Management Warfare system.

2. Chief of Naval Operations (CNO) Task Force Web

This section looks at the Task Force Web (TF Web) initiative planned by CNO for the transformation of most naval data systems into a web environment. Although TF Web is currently focusing on non-tactical implementations, this section looks at TF Web from the more challenging environment of tactical knowledge management. Since its inception, the overall goal of TF Web has been to achieve a transition from Legacy/Stovepipe Systems to an Interoperable Semantic Web. While the focus of this

paper most certainly advocates a similar roadmap, experience has proven that a successful outcome is highly dependent on the specific transition strategy. Therefore in order to evaluate the TF Web transition strategy for a tactical environment, it is important to define terms such as "IP Capable", "Web Enabled" and "Organizational Messaging". For the purposes of this discussion, IP Capable is better defined as a facilitation of network interoperability vice the desire to shift to a web enabled environment. Web Enabled describes the shift from stovepipe systems to a Navy Portal architecture facilitating access to web centric applications. The following portal configuration [13] depicts a potential design envisioned for Navy implementation:

Web-Service Access Via Portal
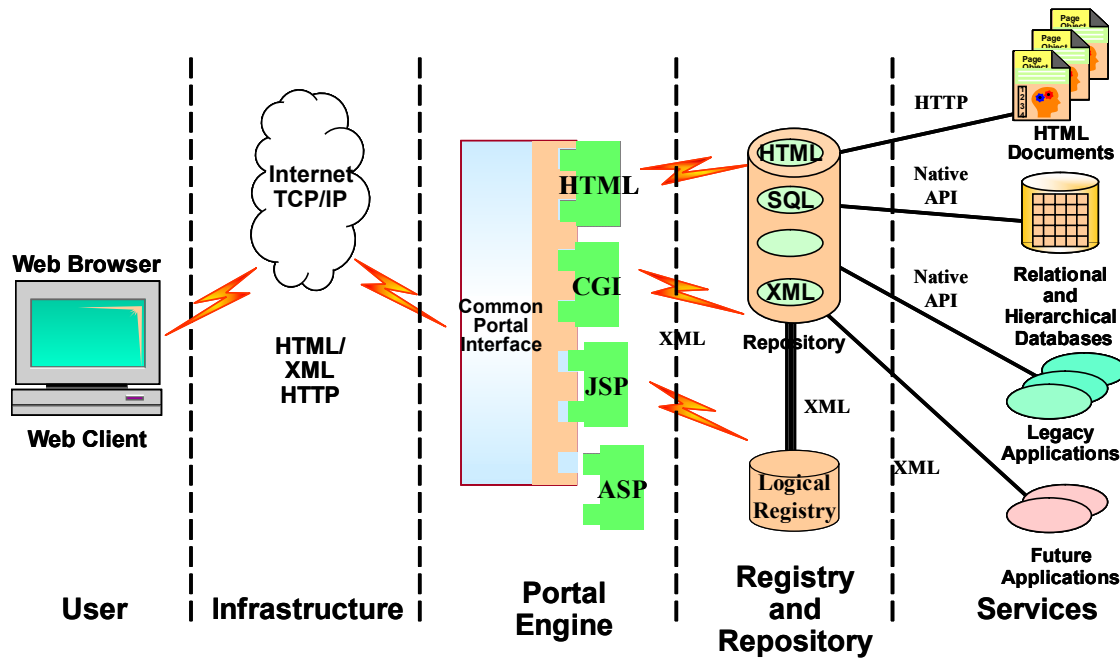
Figure 6.   Portal Architecture ([From Ref. [13].)

Finally, organizational messaging is best defined as a formalized method of sending and receiving doctrine in order to execute mission assignments. Combining the IP Capable and Web Enabled attributes in order to support tactical Organizational Messaging increases the level of complexity when considering factors such as priority packet handling, data accountability, existing interface requirements, allied interoperability and limited bandwidth. The Defense Messaging System (DMS), described in the previous section, was designed as the long-term programmatic solution to provide the messaging capability ashore. However DMS does not address factors issues such as priority packet handling, bandwidth constraints or the limitations of afloat tactical organization messaging. Therefore due to the inherent costs associated with resolving these issues, there have been several proposals to eliminate or modify the organizational message requirement. Regardless of the final outcome the immediate lack of a clear transition strategy, with continued reliance on stove-piped legacy systems, will continue to delay progress and lengthen the time required to achieve information superiority through effective Knowledge Management. Thus far, prototyped efforts at the implementation of tactical IP organizational messaging has taken on varied formats including Outlook email or in some cases XML instantiated browser templates. Each of these attempts requires additional development and has not addressed all key issues cited above. In addition, feedback from TF Web Phase I has indicated a need for more analysis. Phase I comments including "Reduce Graphics" and "Bandwidth extremely limited for most ships underway" were early indications that unique software and processing efforts were required to address data overload concerns. Thus far for non-tactical systems, the TF Web concept appears to have mixed reviews as demonstrated by the following metrics.

- Total Number of Systems: 153

- Number of Systems capable of being Web Enabled: 43

- Number of Systems planned for elimination: 37

- Number of Systems evaluated as infeasible for Web Enabled: 73

While these metrics provide neither an endorsement nor a rejection of TF Web, the concept of a hardware independent interoperable XML centric architecture is a sound

technical approach and emulates the design of current semantic web models. As discussed above, the successful outcome of TF Web will ultimately depend on the transition strategy. The remainder of this section will address TF Web strategy and the technical issues associated with the transition. As can be seen in the Figure 7 below, TF Web is incorporating new technologies [13] as enablers for shifting from traditional legacy systems to a semantic web environment:

Emerging Technologies

From	To
• Proprietary systems	• Internet standard based systems
• Wired	• Wireless
• HTML	• XML
• Limited browser use	• Universal portal use
• Manual use	• Automation
• Limited specific use	• Service Centric Usage
• Diverse message formats	• Common message format (XML)
• Diverse database exchanges	• XML based database exchanges
• Low bandwidth	• More efficient use of bandwidth
• Desktop	• Network appliance / Handheld
• MHz computers	• GHz computers and PDAs

HTML

XML

Next Gen XML??

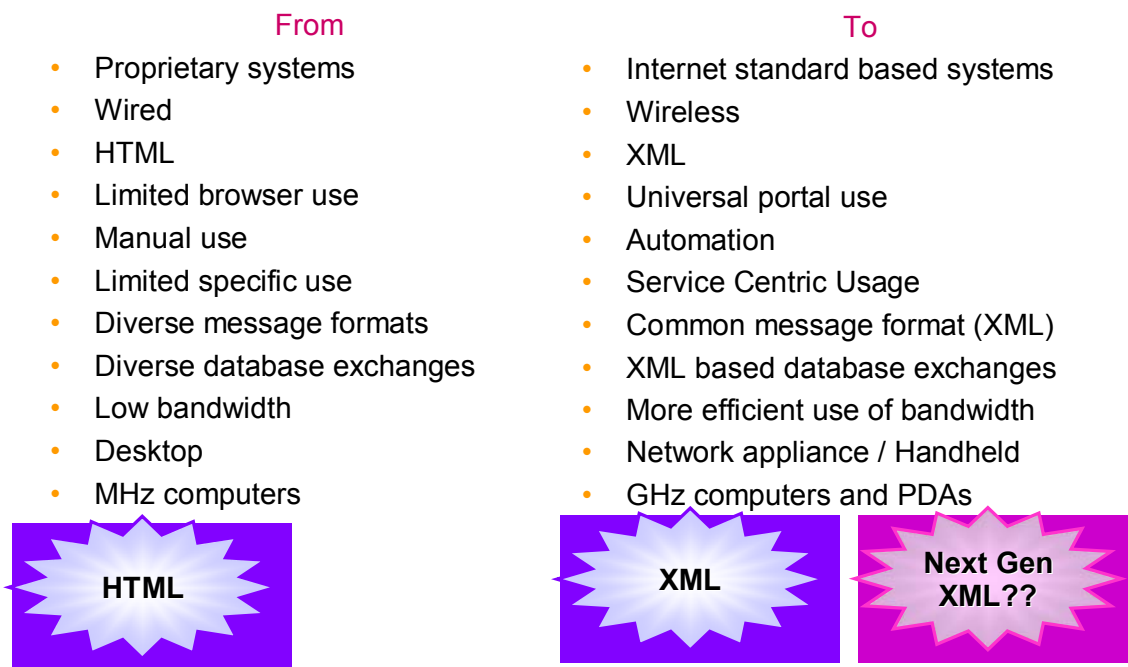Figure 7.   Task Force Web Transition Strategy ([From Ref. [13].)

The main technical issue relating to TF Web or any other Web based system is bandwidth. Similar to the days when 19.2k dial up was viewed as a major step forward, the attainment of 32k in a 25 kilohertz satellite channel has become the minimum standard. Rather than assume that additional resources will become available to alleviate

an already crowded network, several TF Web bandwidth saving techniques currently under study include:

- Selective Database Replication. The goal is to establish a database architecture that replicates changes vice the traditional method of replicating the entire database. This is a critical element with intensive database architecture designs such as TF Web.

- Assignment of High Bandwidth Receive Only Systems (i.e. Global Broadcast System (GBS)) to pass large data files. Potential use of Web based systems via GBS could open up another resource for the portal design.

- Static vice multimedia video. During the recent Iraq conflict, the news media demonstrated the capability to make extensive use of minimum framing videos via satellite phone. While typical Video Teleconferences (VTC) employ a 2X64 connection, the lower bandwidth framing requirements of a static media system are essential to enable a basic capability under TF Web's afloat architecture.

- Asymmetric networking technique. Already deployed in operational low bandwidth satellite circuits, the asymmetric method has provided a performance level consistent with text based web pages and minimum graphics. Most typically employed as 32k shore to ship with a ship to shore reach back of 4.8k or 9.6k via low bandwidth circuits, asymmetric networking has the potential of Gigabyte shore to ship GBS transmissions with a lower data rate afloat reach back.

- Improved compression techniques. While TCP/IP has become the net standard for interoperability, this protocol alignment has a significant bandwidth issue and has not reflected the efficient design of prior legacy systems. Incorporated in most legacy protocols, compression needs to be reevaluated at all levels of the OSI layer model for more effective use of scarce Radio Frequency resources.

Complexities associated with bandwidth not withstanding, the TF Web architecture must be sufficiently robust to support the extensive non-tactical requirements of a large architecture ashore. This three tiered design [13] supporting a broad number of applications and users is depicted in Figure 8 below:
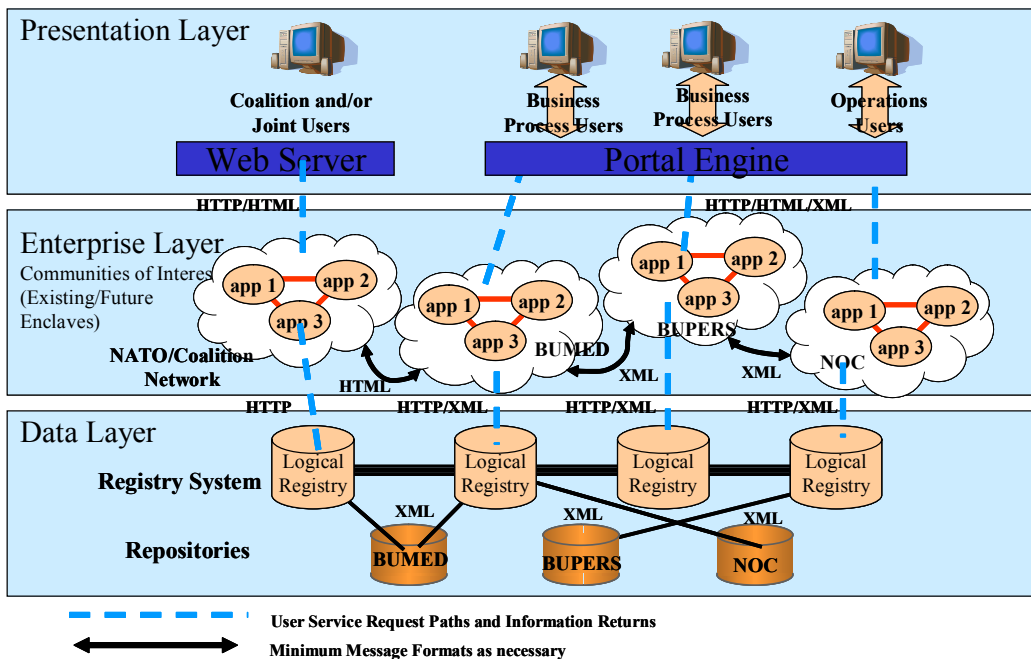

Figure 8.   Task Force Web Tiered Architecture ([From Ref. [13].)

As displayed above in Figure 5 there are 3 levels to the tiered design. They are the Presentation Layer, the Enterprise Layer and the Data Layer. Each layer serves an important role in linking the user to the right application and supporting data. In the Data

Layer, objects and modules that interface with databases and applications are contained within Repositories while the list of services is contained in the Logical Registry. The Enterprise Layer includes the applications servers with the modules used to interface with the portal. User interface is provided by the Presentation Layer and includes the browsers and portal engine for application/information access. This design focuses on connecting both ashore and afloat users in an interoperable seamless architecture capable of meeting joint war fighter requirements. As previously noted Task Force Web is work in progress and the next few years will determine if the transition strategy results in a successful implementation.

3.   XMLMTF

On 11 March 1999, the Joint US Message Text Format (USMTF) Standard Configuration Control Board (CCB) agreed to adopt XML as part of the USMTF standard (MIL-STD-6040). This powerful combination of military and industry standards, called XML-MTF [14] is expected to drastically improve the WarFighter's ability to find, retrieve, process and exchange tremendous amounts of information easily across system, organizational and format boundaries (i.e., the right information at the right time in the right formal). Since the adoption of XML, one of the most interesting initiatives for a military domain has been the XML2MTF project undertaken by the MITRE Corporation. This effort enabled the feasibility of a transition strategy that repackaged the existing US Message Text Format (USMTF) product into a XML file for display in a browser environment or conversion to other formats. Of more significance was the potential of direct database access via the newly formed XML document for updating and providing the most recent record updates. Prior to achieving a translation capability, the development team embarked on an XML-MTF mapping task that resulted in the required specifications defining the relationship between XML-MTF messages and the respective MTF Messages. The composition of the MITRE team is provided below to recognize the individuals associated with this significant accomplishment:

Mike Cokus, project leader, exposed the data in the CDBS as XML. Roger Costello and James Garriss wrote the XSL. Roger Costello added the Java wrapper. Jasen Jacobsen performed the testing.

The overall design goals [14] for the XML-MTF mapping development included:

- XML-MTF shall be easy to read, use, and understand. Descriptive names and logical structures that resemble as much as possible the structure of MTF standards shall be favored over terse abbreviations and clever shortcuts.

- XML-MTF shall be designed to ensure widespread military adoption. In keeping with this goal, XML-MTF shall be designed to accommodate current MTF standards.

- XML-MTF messages should be easy to construct from basic rules mapping it to MTF formats. Transformation of XML-MTF to formats such as USMTF, ADatP-3, and OTH-T Gold should be as simple as possible.

- XML-MTF schemas should be easy to construct; drawing from the logical structure of MTF Message standard databases, such as those defined for USMTF and ADatP-3.

- Operations on XML-MTF messages, such as a query, should be resilient to schema changes.

- XML-MTF shall as much as possible draw on industry adopted standards and technologies to save time and money.

The keystone product of the XML-MTF development effort [14] was the XML-MTF mapping. The purpose of the mapping was to convey a standard means of making MTF Message information available in a Commercial Off-The Shelf (COTS) supported data format while preserving the rich meta-data described in MTF standards. The XML-MTF mapping describes the composition of an XML-MTF message that is a rendering an MTF Message in XML format. The importance of this effort was immediately evident as once the USMTF message was translated into a universal format it enabled the exchange of XML data between heterogeneous systems. This new capability corrected a major limitation that previously restricted message transmissions between USMTF capable hardware systems. With XML it has been possible to exchange data with almost any

33

platform of interest without worrying if the data attributes were the same or if the receiving hardware was capable of processing the US Message Text Format. Another significant advantage has been the opportunity to design and display the data in the manner that the user most desires. Furthermore, the following detailed discussion of the XML-MTF mapping specification is provided to highlight the power of this type of development. The structure of an XML-MTF message, like that of an MTF Message, is hierarchical. The following table illustrates the relationships:

XML-MTF Message	MTF Message
Root Element contains Child Elements	Sets/Segments
Child Elements	Sets/Segments
Segment Element contains Child set & Segment Elements	Segment
Components	Element Names
Field Content	Element Content Characters
Field Element	Composite Field

For example a mapping of the MTF Message composite field corresponding to the XML-MTF Message field elements is shown below:

MTF Message	XML-MTF Message
Composite Field	**Field Element "<field-name>elementals</field-name>"**
ARRIVAL/20030606//	<date>
	<year>2003</year>
	<month_numeric>06</month_numeric>
	<day>06<day>
	</date>

The example shown above demonstrates that a relative short composite field such as 20030606 may convert to several elementals with the associated beginning/ending field-

34

names. Although this translation results in a much larger data element, it does facilitate the required formatting to support XML processing. The specific structure of a particular XML-MTF message will depend upon the characteristics of the MTF message. The following rules [14] describe the general form of an XML-MTF message using a modified Backus Naur Form (BNF) syntax:

XML-MTF MESSAGE -> MESSAGE

MESSAGE -> <message-text-format-name> (SET|SEGMENT)+ </message-text-format-name>

SEGMENT -> <first-set-format-name_segment> (SET|SEGMENT)+ </first-set-format-name_segment>

SET -> LINEAR-SET| COLUMNAR-SET | FREE-TEXT-SET

LINEAR-SET-> <set-format-name setid = 'set-format-identifier' position = 'set-position' amplification = 'FREE-TEXT' narrative = 'FREE-TEXT'> FIELD-FORMAT* FIELD-GRP* </set-format-name>

COLUMNAR-SET -> <set-format-name setid = 'set-format-identifier' position = 'set-position' amplification = 'FREE-TEXT' narrative = 'FREE-TEXT' > FIELD-GRP* </set-format-name>

FREE-TEXT-SET -> AMPN-SET| NARR-SET | RMKS-SET | GENTEXT-SET

AMPN-SET -> amplification = 'FREE-TEXT'

NARR-SET -> narrative = 'FREE-TEXT'

RMKS-SET -> <remarks setid = 'RMKS > FREE-TEXT-FIELD </remarks>

GENTEXT-SET -> <general_text setid = 'GENTEXT' position = 'set-position' > GENTEXT-IND-FIELD FREE-TEXT-FIELD

</general_text>

GENTEXT-IND-FIELD -> <text_indicator>

DATA

</text_indicator>

FIELD-GRP -> <set-format-name_group_of_fields>FIELD-FORMAT+

</set-format-name_group_of_fields>

FIELD-FORMAT -> ELEMENTAL-FIELD | COMPOSITE-FIELD |

ALTERNATIVE-CONTENT-FIELD | EMPTY-FIELD

FREE-TEXT-FIELD -> <free_text xml:space = 'preserve'>FREE-TEXT</free_text>

ELEMENTAL-FIELD -> <FUD-name> DATA </FUD-name>

COMPOSITE-FIELD -> <FUD-name> ELEMENTAL-FIELD+ </FUD-name>

ALTERNATIVE-CONTENT-FIELD -> <field-position-name-plus-set-format-identifier>
ELEMENTAL-FIELD | COMPOSITE-FIELD

</field-position-name-plus-set-format-identifier>

EMPTY-FIELD -> <FUD-name/> |

DATA -> MTF field data with XML illegal data characters escaped and field descriptors
removed

FREE-TEXT -> MTF free-text data

The importance of the above BNF syntax is to provide the reader with an understanding
of the format required to achieve translation from the MTF Message to XML-MTF
Message. The mapping rules will be used later in this research to demonstrate use of the
XML2MTF capability in an operational scenario. . It is noteworthy that XML-MTF has
now become an official part of MIL-STD-6040 (USMTF) and is currently in the staffing
process for inclusion in ADatP-3 (NATO MTF). Additional information concerning the
XML-MTF capability can be obtained from Mike Cokus at msc@mitre.org.

Thus far we have looked at IP Based Messaging, TF Web, and XML-MTF. Each
of these items only address a particular aspect of the potential architecture. The next
subject will begin to look at an integrated architecture capable of supporting Tactical War
fighter requirements.

4. Knowledge Management (KM)

Per the author's academic advisor, Doctor Shing, Knowledge Management is a popular buzz word that does not yet translate to a true identity. As a loyal student the author tends to support the academic advisor's guidance however in this case I would modify the guidance to include that the general misuse of the term Knowledge Management has had a significant impact on the ability to establish an identify within the cognizant domain. For example, the author's first exposure to Knowledge Management resulted in a misdirected focus on information vice knowledge. This is a paradigm that practitioners have to set aside in order to understand the true meaning of knowledge management. Even today this preoccupation towards substituting information management for knowledge management continues for very "knowledgeable individuals". Therefore it is important to clearly define knowledge management prior to proceeding. There were 2 definitions cited earlier in the paper and they will be repeated here for further examination:

1. Knowledge Management is the systemic and organizationally specified process for acquiring, organizing, and communicating knowledge of employees so that other employees may make use of it to be more effective and productive in their work [1].

2. Knowledge Management is about delivering the right knowledge to the right people at the right time. [2].

A combination of the 2 definitions from above results in the following standard: Knowledge Management is the systemic and organizationally specified process for delivering the right knowledge to the right people at the right time. Of the multitudes of knowledge management definitions, this inherited definition most aptly captures the magnitude of KM for any military organization. It should be noted that neither the inherited definition nor the 2 original definitions include information as part of the vocabulary. The oversight is significant as this lack of association between knowledge and information could be interpreted to represent a distinction at the most fundamental level. The following quote from Les Alberthal, 1995, clarifies the distinction in the understanding of the hierarchical nature of information and knowledge:

> Like water, this rising tide of data can be viewed as an abundant, vital and necessary resource. With enough preparation, we should be able to tap

into that reservoir -- and ride the wave -- by utilizing new ways to channel raw data into meaningful information. That information, in turn, can then become the knowledge that leads to wisdom.

To further understand the sequential relationship knowledge has with data and information, the following statements [15] are provided:

- A collection of data is not information.
- A collection of information is not knowledge.
- A collection of knowledge is not wisdom.
- A collection of wisdom is not truth.

The synopsis of these points is that information, knowledge, and wisdom are more than simply collections. Rather, the whole represents more than the sum of its parts and has a synergy of its own. To categorize how information and knowledge would fit into the knowledge management needs the following associations can be made:

- **Information** relates to description, definition, or perspective (what, who, when, where).
- **Knowledge** comprises strategy, practice, method, or approach (how). It has also been described as the eye of desire that can become the pilot of the soul.

From a War Fighter's perspective, most organizations have become very good from an information perspective or the what, who, when, and where. However the challenge is to incorporate the how for achieving a successful knowledge oriented organization. Now that the relationship between information and knowledge is discernable, the process of transitioning from a data mentality to a decision-oriented environment can begin. The next stage of a transition process requires a model to measure the amount of progress towards KM implementation. IBM has a Knowledge Continuum model [16] that identifies 8 stages of organizational implementation commencing at the Beginner stage and advancing towards the highest maturity level. The table depicted below highlights the author's adaptation of the IBM model.

STAGE	ATTRIBUTES	CRITICAL ENABLER
1. Beginner	No organized efforts to capture, protect and share knowledge	Leadership commitment to protecting intellectual capital and getting educated in the Knowledge Management discipline
2. Knowledge Laggard	Little or no investment in developing people and fostering collaboration.	Dedicating full time resources to Knowledge Management (KM) activities
3. Knowledge Loser	Recognizes KM value but no process in place to protect loss of knowledge from retirements, transfers etc.	Targeted initiatives that leverage technology investments to maintain corporate memory
4. Knowledge Gatherer	Data collection focused and oriented towards information technology	Measures tangible outcome related to KM initiatives
5. Knowledge Leverager	Value of KM well understood. Strategies are in place to apply KM principals	Equivalent to level 3 in the capability maturity model. Repeatable processes in place and systems in place to support
6. Knowledge Innovator	KM integrated into mission. Collaboration underway	Internal culture transforming towards KM practices
7. Learning Organization	KM processes are pervasive throughout the organization. Culture learns and adapts rapidly	KM activities expanding to outside the organization with wide spread adoption of KM processes
8. Knowledge Enterprise	Knowledge Management extends beyond the Enterprise level and is a standard model among multiple agencies	

Table 1.   Authors Adaptation of the IBM CONTINUUM ([After Ref. [16].)

From a software perspective, the Table 1 model is similar to the Capability Maturity Model (CMM) in that repeatable processes are important to development of KM practices. The relationship between the IBM model and CMM can be viewed by comparing Table I with the Maturity model [17] provided in the following table:

LEVEL #	LEVEL TITLE	CHARACTERISTICS
1	Initial	Ad Hoc Processes, occasionally chaotic, poor definition with success based on individual heroic efforts.
2	Repeatable	Basic management tracking processes established. Process discipline has expanded to repeat previous successes.
3	Defined	Organization has integrated a standard process across projects for all software related activities.
4	Managed	Software measurements incorporated. Metrics are used to control both the software process and products.
5	Optimizing	Continuous process improvement with external and internal feedback from innovative ideas & technologies.

Table 2.   CAPABILITY MATURITY MODEL (CMM) ([After Ref. [17].)

Upon closer examination the relationship between the KM model and the CMM model is noticeable in other areas such as both models possess the following characteristics:

a. Layered approach
b. Organizational oriented
c. Heavy reliance on standards

It is also envisioned that, similar to the CMM model, the KM model will place an emphasis on software tools for successful implementation of a knowledge enterprise. As advertised by IBM, the purpose of knowledge management software is to capture, manage, evaluate and reuse knowledge -- driving responsiveness, innovation, efficiency and learning to make better decisions faster.

Prior to discussing required toolsets, it is useful to review the rationale behind a knowledge management approach. The requirement exists for the Naval Warfare Tactical Commander to execute time critical decisions based on the right knowledge at the right time. This time related requirement is not supported by an extensive web search or by perusing several hundred naval messages to support the decision process. It is also not satisfied by extensive discussions with the domain subject matter expert prior to execution. The requirement is to have the right knowledge immediately available. Therefore breaking the details of this requirement into workable tasks requires identification of the software development processes that address each aspect of the knowledge management environment. For the purpose of this analysis, the following task elements have been chosen:

    a. Integrate a knowledge management solution from a Graphical User Interface that incorporates the ontology, semantics and syntax for the Naval Tactical Warfare Commanders domain.

    b. Formally specify the Commander's domain such that a given ontology expression can be processed within the time critical criteria.

    c. Develop the RDF metadata model for integration within the ontology specification.

    d. Build the syntax within the XML design to provide a foundation for both the RDF and Ontology efforts.

Breaking down these task elements into a single product or multiple products is not as simple as implementing an integrated software tool that incorporates the required syntax and semantic functionality. For example, addressing both tacit and explicit knowledge is an effort requiring extensive development and constant updating. Therefore the key to achieving a successful integration of the various KM elements is the development of an approach that reduces the challenge into an integration of existing applications. While there are available KM toolsets designed for specific applications, the maturity level of many existing products remain in a development or initial deployment stage. Current applications at the XML layer have demonstrated a higher mature level than at the RDF and Ontology tiers. For the purposes of this thesis the author has located the following

toolsets to allow development of a prototype Tactical Naval Knowledge Management System:

XML: XMLSPY Enterprise Edition Version 2004, Altova Inc.

RDF: IsaViz RDF Editor Version 1.2 W3C 2001; RDFedt Version 1.02, 12-01-01, Jan Winkler

ONTOLOGY: Protégé 2000 Version 1.8, Stanford University

As stated previously, the maturity level of these products go from a full production model (i.e. XMLSPY) to an Ontology prototype development toolset (i.e. Protégé 2000). The next few sections will address employment of these tools in the design stage to achieve an integrated knowledge management toolset.

C.   MAPPING REQUIREMENTS TO DESIGN

DoD's strategic vision for the 21st century is to ensure that U.S. forces have information superiority in every mission area. A related requirement previously articulated by DoD was to achieve dominant battlespace awareness through advanced information technology. As previously discussed above, information technology and information management (IM) are essential but insufficient to achieve information superiority. Therefore, Knowledge Management (KM) offers the potential to significantly leverage the value of our IT investment as well as providing the link between technology and people.

Drawing upon the information superiority in every mission area requirement this design will focus on providing a KM architecture for information integration in support of mission resource management by the Naval Warfare Tactical Commander. Normally the translation of requirements into a design is a lengthy process that includes preparation of a Software Requirements Specification (SRS), translation into formal design specifications and completion of other documents. However due to the thesis goal of evaluating semantic qualities, the design specifications for this prototype have been adjusted to address the following requirements:

- Provide a decision oriented architecture to manage information resources for each mission area.

- Include a scenario driven approach for the following mission types:

o   Land Combat Mission

o   Tomahawk Strike Mission

o   Air Operations Mission

o   Maritime Operations Mission

o   Undersea Attack Mission

o   Special Operations Mission

- Identify the required force and weapon resources for each mission type

- Provide an operational picture capable of generic mission support

These overall design specific requirements fulfill the anticipated attributes found in a typical environment supporting a Naval Tactical Warfare Commander. They also have the potential of allowing further development to expand and evaluate additional capabilities. This section completes the introductory, background, process and requirements discussion. The remainder of this document will address the specifics of a prototype knowledge management software product and present the results from evaluating the semantic qualities in relationship to flexibility and timeliness.

THIS PAGE INTENTIONALLY LEFT BLANK

IV.   DESIGN STRATEGY

Drawing upon the above design specifications, this prototype will be based on a top down design approach addressing each element of the semantic architecture. This approach should be familiar to software designers and developers who have worked Unified Modeling Language (UML) Tools such as Rationale Rose, etc. Similar to the UML design strategy, the desired goal is to build a model that translates directly to a usable source code product. Therefore the author started at the Ontology model to address the design at the decision level prior to proceeding to the next semantic levels of Resource Description Framework (RDF) and XML. Although many KM toolsets are in the prototype phase, at least one tool possesses a plug in functionality that provides the potential of combining all elements into one process. However in this prototype, separate tools were used for each element. The separate efforts will be described in detail below

A.   KNOWLEDGE MANAGEMENT APPROACH

Similar to other decision oriented tools, this development attempts to come to a conclusion that would normally be determined by a WarFighter with the equivalent information resources. A basic practical example is the Submarine Diving Panel. Utilizing electro-magnetic technology, each critical hull opening provides a signal to the panel for displaying a open or closed condition. Although the Submarine Commander directs the crew to dive the submarine, an important input to this decision is the visual circle/bar display from the diving panel. Bottom line is an all bar display generally results in a decision to dive whereby a circle indication always requires additional research. An example panel is shown in the following figure:
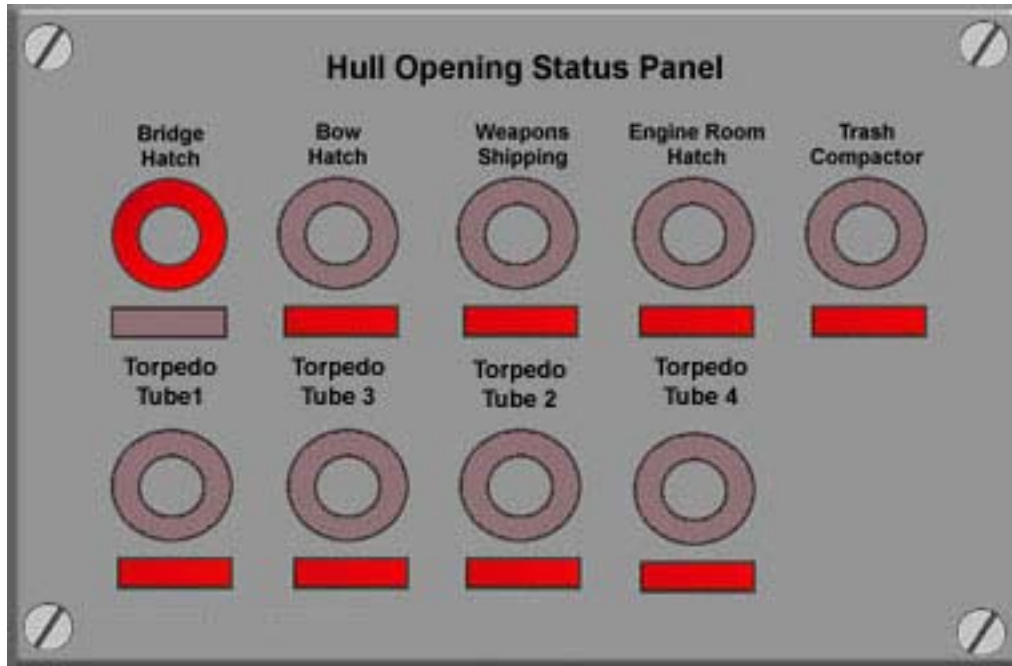
Figure 9.   Submarine Diving Control Panel ([From Ref. [18].)

Following a Diving Control Panel like process and based on input from a Resource
Description Framework file, this tool will produce a stoplight display for each mission
type. A green result will signal the Tactical Naval Warfare Commander that all resources
are in place to proceed with the tasking. A yellow result will indicate that a deficiency
exists but further research is appropriate prior to proceeding. Finally, any major resource
deficiency will result in a red display indicating that the mission is not ready to proceed.
The next part of the process will be to graphically address the KM design in an Ontology
tool.

1.   ONTOLOGY Development

As described in a previous section, Protégé 2000 was used in developing the
Ontology for this project. Utilizing the Ontoviz plug-in, a graphical representation was
created to define the relationships between classes. Due to the size of the graphs, each
relationship will be shown below in separate figures:

Figure 10.   Protégé 2000 Graph of Mission Relationships

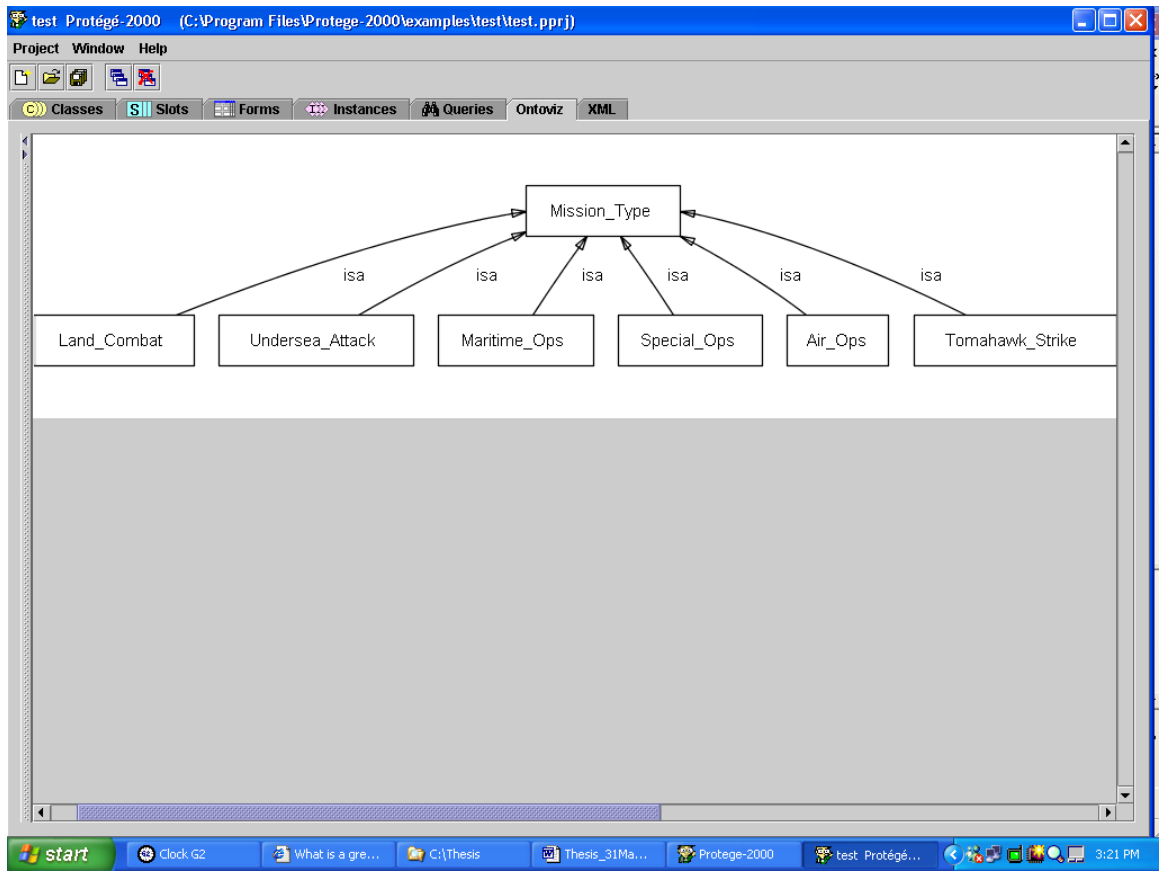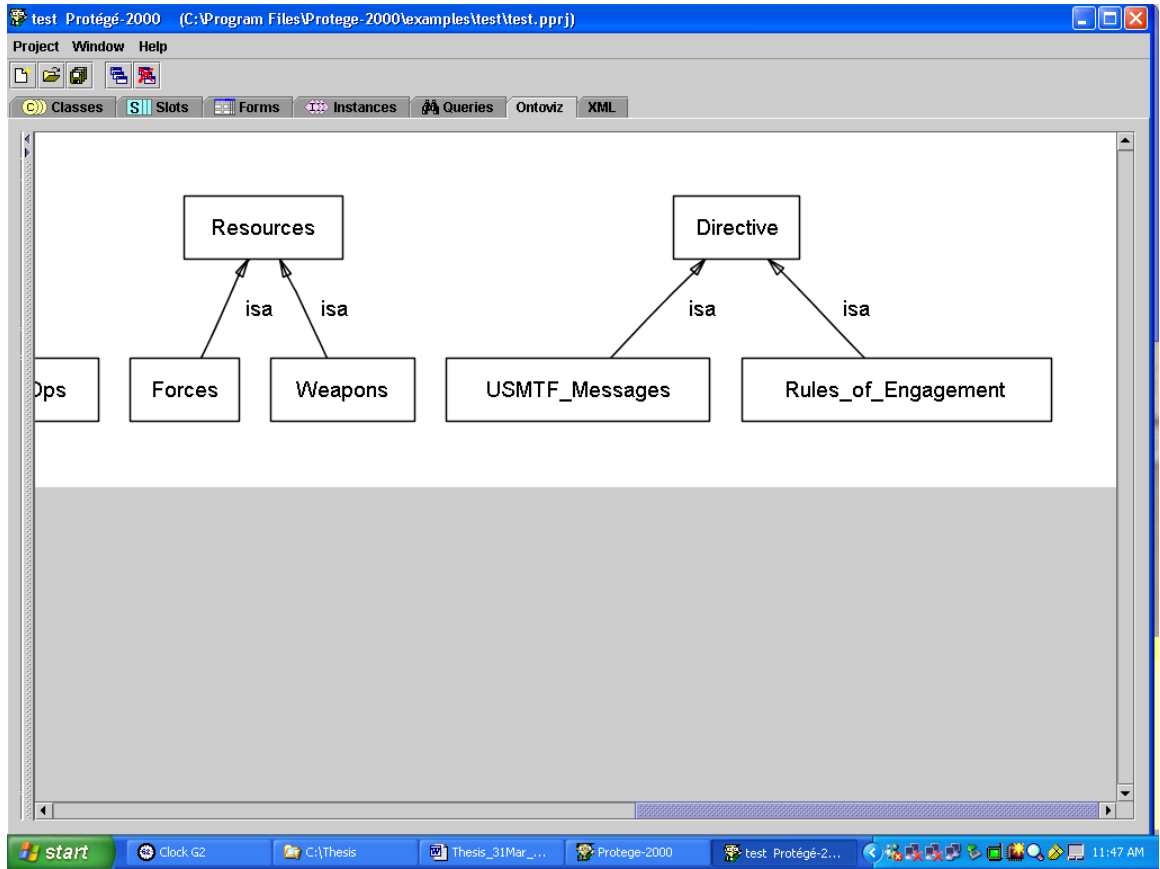Figure 11.   Resources & Directives Relationship Graph

The above graphs show the following relationships:

Mission Types:

> Land Combat
>
> Tomahawk Strike
>
> Air Operations
>
> Maritime Operations
>
> Undersea Attack
>
> Special Operations

Resources:

    Forces

    Weapons

Directives:

    USMTF Messages

    Rules of Engagement

These relationships form the architecture of the Naval Tactical Warfare Commander's Manager and will be fundamental to software development. Similar to UML toolsets, Protégé 2000 enables the user to export the graph into a file format for use in source code development. In this case the graphs created above were exported to a RDF file format as shown below:

```
<?xml version='1.0' encoding='ISO-8859-1'?>
<!DOCTYPE rdf:RDF [
        <!ENTITY rdf 'http://www.w3.org/1999/02/22-rdf-syntax-ns#'>
        <!ENTITY test 'http://protege.stanford.edu/test#'>
        <!ENTITY rdfs 'http://www.w3.org/TR/1999/PR-rdf-schema-
19990303#'>
]>
<rdf:RDF xmlns:rdf="&rdf;"
        xmlns:test="&test;"
        xmlns:rdfs="&rdfs;">
<rdfs:Class rdf:about="&test;Air_Ops"
        rdfs:label="Air_Ops">
        <rdfs:subClassOf rdf:resource="&test;Mission_Type"/>
</rdfs:Class>
<rdfs:Class rdf:about="&test;Directive"
        rdfs:label="Directive">
        <rdfs:subClassOf rdf:resource="&rdfs;Resource"/>
</rdfs:Class>
<rdfs:Class rdf:about="&test;Forces"
        rdfs:label="Forces">
        <rdfs:subClassOf rdf:resource="&test;Resources"/>
</rdfs:Class>
<rdfs:Class rdf:about="&test;Land_Combat"
        rdfs:label="Land_Combat">
        <rdfs:subClassOf rdf:resource="&test;Mission_Type"/>
</rdfs:Class>
<rdfs:Class rdf:about="&test;Maritime_Ops"
        rdfs:label="Maritime_Ops">
        <rdfs:subClassOf rdf:resource="&test;Mission_Type"/>
</rdfs:Class>
<rdfs:Class rdf:about="&test;Mission_Type"
```

```
        rdfs:label="Mission_Type">
      <rdfs:subClassOf rdf:resource="&rdfs;Resource"/>
</rdfs:Class>
<rdfs:Class rdf:about="&test;Resources"
        rdfs:label="Resources">
      <rdfs:subClassOf rdf:resource="&rdfs;Resource"/>
</rdfs:Class>
<rdfs:Class rdf:about="&test;Rules_of_Engagement"
        rdfs:label="Rules_of_Engagement">
      <rdfs:subClassOf rdf:resource="&test;Directive"/>
</rdfs:Class>
<rdfs:Class rdf:about="&test;Special_Ops"
        rdfs:label="Special_Ops">
      <rdfs:subClassOf rdf:resource="&test;Mission_Type"/>
</rdfs:Class>
<rdfs:Class rdf:about="&test;Tomahawk_Strike"
        rdfs:label="Tomahawk_Strike">
      <rdfs:subClassOf rdf:resource="&test;Mission_Type"/>
</rdfs:Class>
<rdfs:Class rdf:about="&test;USMTF_Messages"
        rdfs:label="USMTF_Messages">
      <rdfs:subClassOf rdf:resource="&test;Directive"/>
</rdfs:Class>
<rdfs:Class rdf:about="&test;Undersea_Attack"
        rdfs:label="Undersea_Attack">
      <rdfs:subClassOf rdf:resource="&test;Mission_Type"/>
</rdfs:Class>
<rdfs:Class rdf:about="&test;Weapons"
        rdfs:label="Weapons">
      <rdfs:subClassOf rdf:resource="&test;Resources"/>
</rdfs:Class>
</rdf:RDF>
```

This exported RDF Ontology file will be used as the building block for further software development of the Naval Tactical Warfare Commander's Manger. The next step in the building process will utilize the above Ontology generated file for additional processing by the RDF toolset.

2.   RDF Design

This stage of the process bridges the semantic levels of Ontology and Resource Description Framework. The stage will begin by importing the Ontology file into the ISAviz RDF toolset. Similar to Protégé 2000, the imported file produced the following RDF graph:
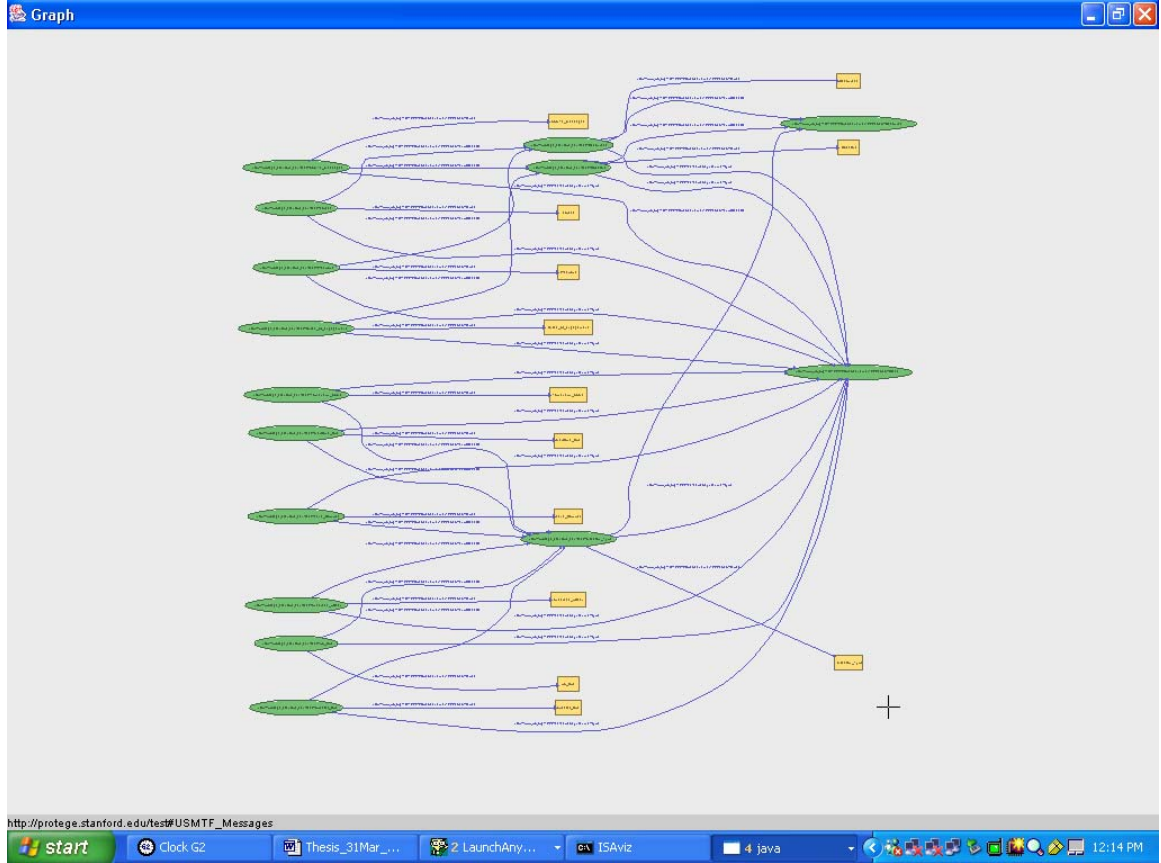
Figure 12.   ISAviz RDF Graph

Although the ISAviz RDF Graph has taken on a busier appearance, an important transformation has taken place during the conversion. This will become apparent upon exporting the new graph into an RDF/XML format which will be discussed in the next section.

3.   XML Implementation

Exporting the ISAviz graph into an RDF/XML format might seem redundant as the definition of RDF includes XML based syntax. However this additional step not only bridges the RDF and XML semantic levels but under the ISAviz export process a URL location is added to the file with the following results:

```xml
<?xml version="1.0" ?>
<rdf:RDF xmlns:RDFNsId1="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#" xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
<RDFNsId1:Class rdf:about="http://protege.stanford.edu/test#Weapons" RDFNsId1:label="Weapons">
<RDFNsId1:subClassOf>
<RDFNsId1:Class rdf:about="http://protege.stanford.edu/test#Resources" RDFNsId1:label="Resources">
<RDFNsId1:subClassOf rdf:resource="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Resource" />
</RDFNsId1:Class>
</RDFNsId1:subClassOf>
</RDFNsId1:Class>
<RDFNsId1:Class rdf:about="http://protege.stanford.edu/test#Special_Ops" RDFNsId1:label="Special_Ops">
<RDFNsId1:subClassOf>
<RDFNsId1:Class rdf:about="http://protege.stanford.edu/test#Mission_Type" RDFNsId1:label="Mission_Type">
<RDFNsId1:subClassOf rdf:resource="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Resource" />
</RDFNsId1:Class>
</RDFNsId1:subClassOf>
</RDFNsId1:Class>
<RDFNsId1:Class rdf:about="http://protege.stanford.edu/test#Rules_of_Engagement" RDFNsId1:label="Rules_of_Engagement">
<RDFNsId1:subClassOf>
<RDFNsId1:Class rdf:about="http://protege.stanford.edu/test#Directive" RDFNsId1:label="Directive">
<RDFNsId1:subClassOf rdf:resource="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Resource" />
</RDFNsId1:Class>
</RDFNsId1:subClassOf>
</RDFNsId1:Class>
<RDFNsId1:Class rdf:about="http://protege.stanford.edu/test#Land_Combat" RDFNsId1:label="Land_Combat">

<RDFNsId1:subClassOf rdf:resource="http://protege.stanford.edu/test#Mission_Type" />
</RDFNsId1:Class>
<RDFNsId1:Class rdf:about="http://protege.stanford.edu/test#Tomahawk_Strike" RDFNsId1:label="Tomahawk_Strike">
```

```
  <RDFNsId1:subClassOf
    rdf:resource="http://protege.stanford.edu/test#Mission_Type" />
    </RDFNsId1:Class>
- <RDFNsId1:Class rdf:about="http://protege.stanford.edu/test#Forces"
    RDFNsId1:label="Forces">
  <RDFNsId1:subClassOf
    rdf:resource="http://protege.stanford.edu/test#Resources" />
    </RDFNsId1:Class>
- <RDFNsId1:Class
    rdf:about="http://protege.stanford.edu/test#Maritime_Ops"
    RDFNsId1:label="Maritime_Ops">
  <RDFNsId1:subClassOf
    rdf:resource="http://protege.stanford.edu/test#Mission_Type" />
    </RDFNsId1:Class>
- <RDFNsId1:Class
    rdf:about="http://protege.stanford.edu/test#USMTF_Messages"
    RDFNsId1:label="USMTF_Messages">
  <RDFNsId1:subClassOf
    rdf:resource="http://protege.stanford.edu/test#Directive" />
    </RDFNsId1:Class>
- <RDFNsId1:Class rdf:about="http://protege.stanford.edu/test#Air_Ops"
    RDFNsId1:label="Air_Ops">
  <RDFNsId1:subClassOf
    rdf:resource="http://protege.stanford.edu/test#Mission_Type" />
    </RDFNsId1:Class>
- <RDFNsId1:Class
    rdf:about="http://protege.stanford.edu/test#Undersea_Attack"
    RDFNsId1:label="Undersea_Attack">
  <RDFNsId1:subClassOf
    rdf:resource="http://protege.stanford.edu/test#Mission_Type" />
    </RDFNsId1:Class>
    </rdf:RDF>
```

The design of this prototype will be based on the class relationship specified in above process. However due to the application's standalone design, the added URL output will not be implemented. In a true semantic architecture, the URL would provide access to the required data sources for timely decision execution. In the prototype the required data will be simulated in a XML formatted file on the supporting hard drive.

In addition, the XML syntax will be used extensively in the application to merge dissimilar formats into a common standard for processing by the software. Regardless of the data source, this XML feature will provide the default semantic format while enabling additional data manipulation to achieve a timely decision.

4.   GUI Integration

Following the Submarine Diving Panel example, the GUI design of this decision oriented prototype had to blend the simplicity of a stoplight display with the complexity of various information sources. This combination drove the author towards a JAVA based GUI in order to integrate the required display with the functionality of information processing. During the JAVA education process one of the learning tools provided by Sun had many of the qualities required for this project. Therefore the Sun application served as the template for GUI Development. In recognition of this contribution, the following Sun license release statement is provided:

Copyright (c) 1997-1999 by Sun Microsystems, Inc. All Rights Reserved.

Sun grants you ("Licensee") a non-exclusive, royalty free, license to use, modify and redistribute this software in source and binary code form, provided that

i) this copyright notice and license appear on all copies of the software; and

ii) Licensee does not utilize the software in a manner which is disparaging to Sun. This software is provided "AS IS," without a warranty of any kind.

ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES,INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE HEREBY EXCLUDED. SUN AND ITS LICENSORS SHALL NOT BE LIABLE FOR ANY DAMAGES SUFFERED BY LICENSEE AS A RESULT OF USING, MODIFYING OR DISTRIBUTING THE SOFTWARE OR ITS DERIVATIVES. IN NO EVENT WILL SUN OR ITS LICENSORS BE LIABLE FOR ANY LOST REVENUE, PROFIT OR DATA, OR FOR DIRECT, INDIRECT, SPECIAL, CONSEQUENTIAL, INCIDENTAL OR PUNITIVE DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF THE USE OF OR INABILITY TO USE SOFTWARE, EVEN IF SUN HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

This software is not designed or intended for use in on-line control of aircraft, air traffic, aircraft navigation or aircraft communications; or in the design, construction, operation or maintenance of any nuclear facility. Licensee represents and warrants that it will not use or redistribute the Software for such purposes.

During prototype development, the benefits of employing JAVA were realized not only from a GUI aspect but also from an integration perspective. As the prototype was required to interface with other applications, the JAVA calls were implemented efficiently and resulted in a smooth transition from one environment to another. The resulting GUI pages are provided below. The application opens with Figure 13 which displays instructions for this prototype and general usage. This page is called the KM Introduction GUI and it provides an overview of the various missions along with an explanation of the supporting databases and mission related data. Depending on mission selection, the user will be taken to a panel that provides information specific to the selected task. The Introduction GUI and 6 Mission GUIs are displayed in the following pages.
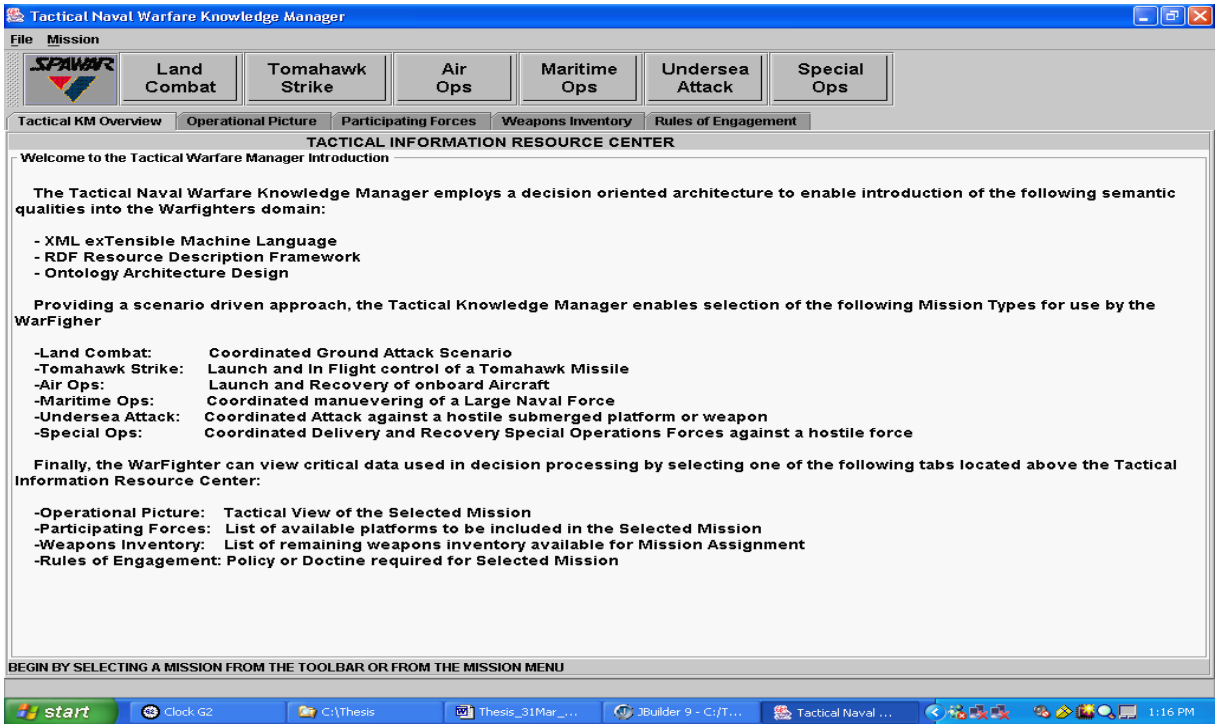
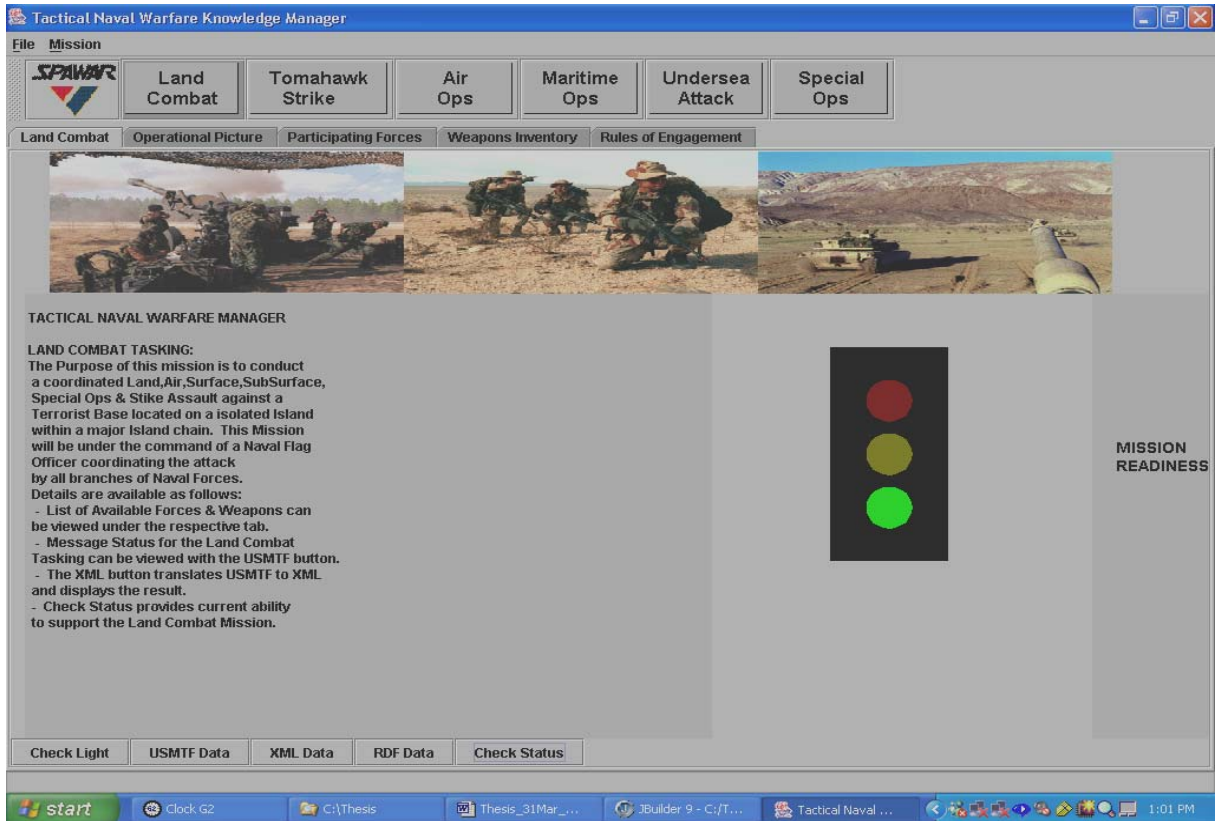Figure 13.   KM Introduction GUI

Figure 14.   Land Combat Mission GUI

56

Figure 15.   Tomahawk Strike Mission GUI

Figure 16.   Air Operations Mission GUI

Figure 17.   Maritime Operations Mission GUI

Figure 18.   Undersea Attack Mission GUI

58

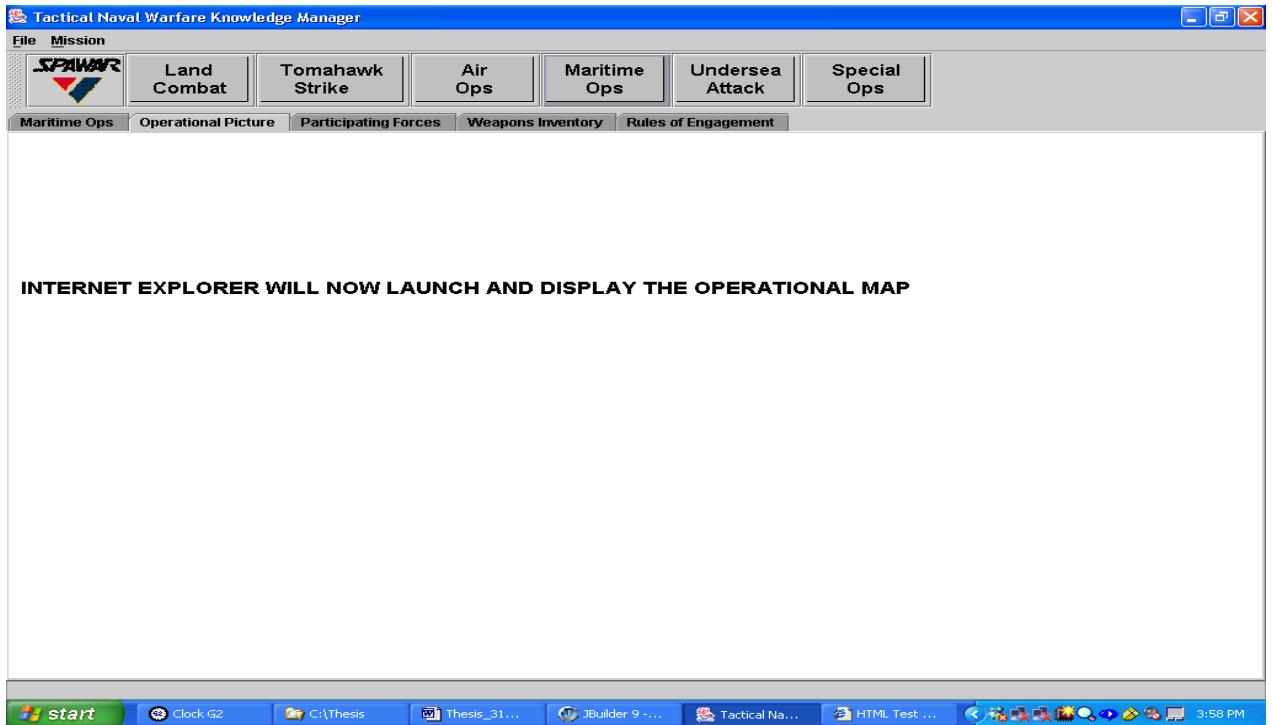Figure 19.   Special Operations Mission GUI

Figure 20.   Operational Picture Launch GUI

Figures 14 through 19 provide a graphical user interface for each Mission Type. This GUI design approach enables the Naval Tactical Warfare Commander to evaluate the readiness of each mission area independently or to review mission preparedness from a single application. Each figure provides the following:

a. *Textual Explanation of the Specific Mission Type.*

b. *Access to Required Mission Related Message Sources.*

c. *Access to Personnel and Weapons Database Sources.*

d. *Access to XML Translation of Message & Database Data.*

e. *Stoplight Representation of Mission Readiness Status.*

f. *Access to Other Mission Types.*

g. *Access to Operational Picture of Mission Type.*

Tabbed Panes and Buttons are provided for easy access to other Mission Types or any data resources. Two Important Mission Panel GUI Buttons are "Check Light" and "Check Status". The Check Light Button provides a visual check on the stoplight to ensure the lights are working properly and the "Check Status" button on each Mission Type will display a stoplight result to indicate readiness to perform the designated Mission. Details behind the "Check Status" and remaining GUI functionality will be discussed in the next section of this analysis.

The next page will display the GUI employed for access to the current operational picture. The Operational Picture is tabbed launched and initially displays a notification that the Digital Nautical Picture is being launched by the associated browser. See Figure 20 above. A Worldwide map [19] will subsequently be displayed in the browser for selection of the area specific map (Figure 21). In this example the appropriate Maritime Area map is selected for the operational area (Figure 22).
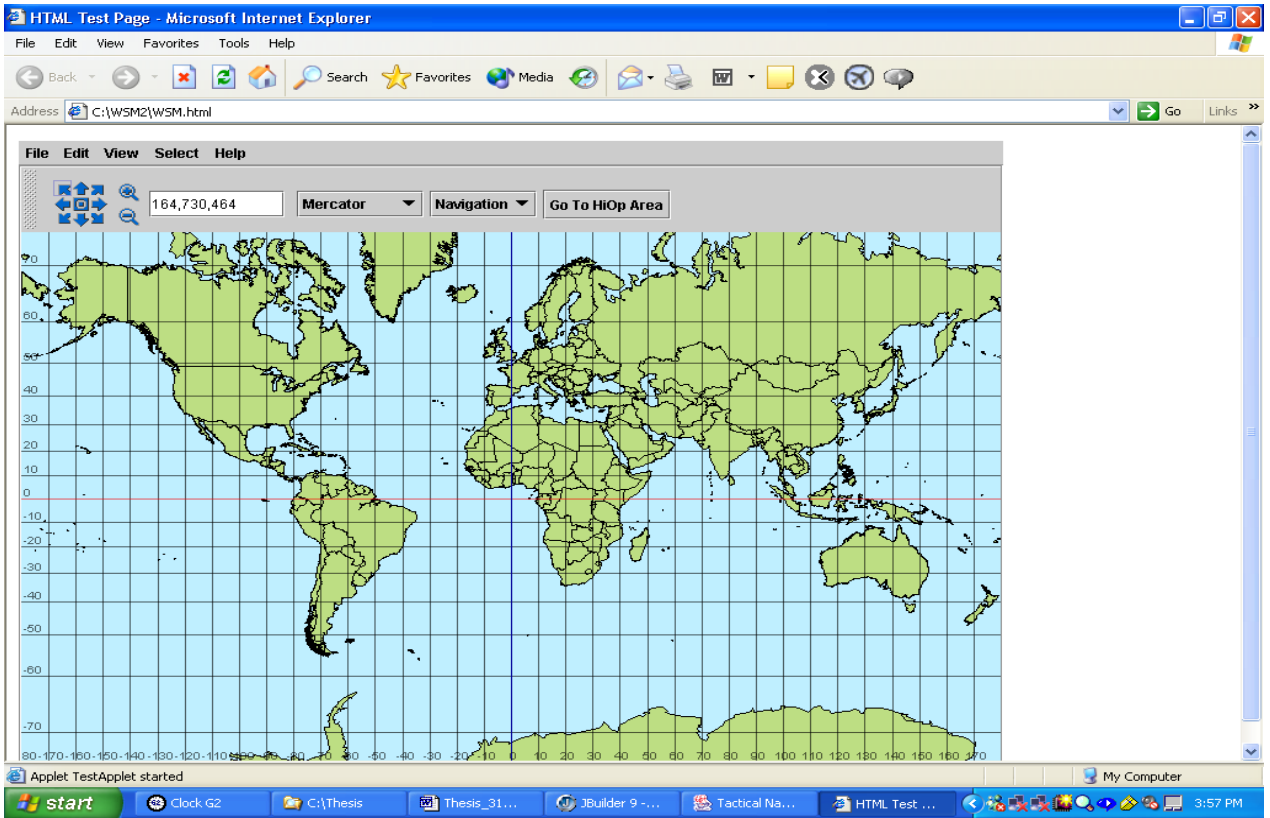
Figure 21.   Operation Worldwide Picture GUI
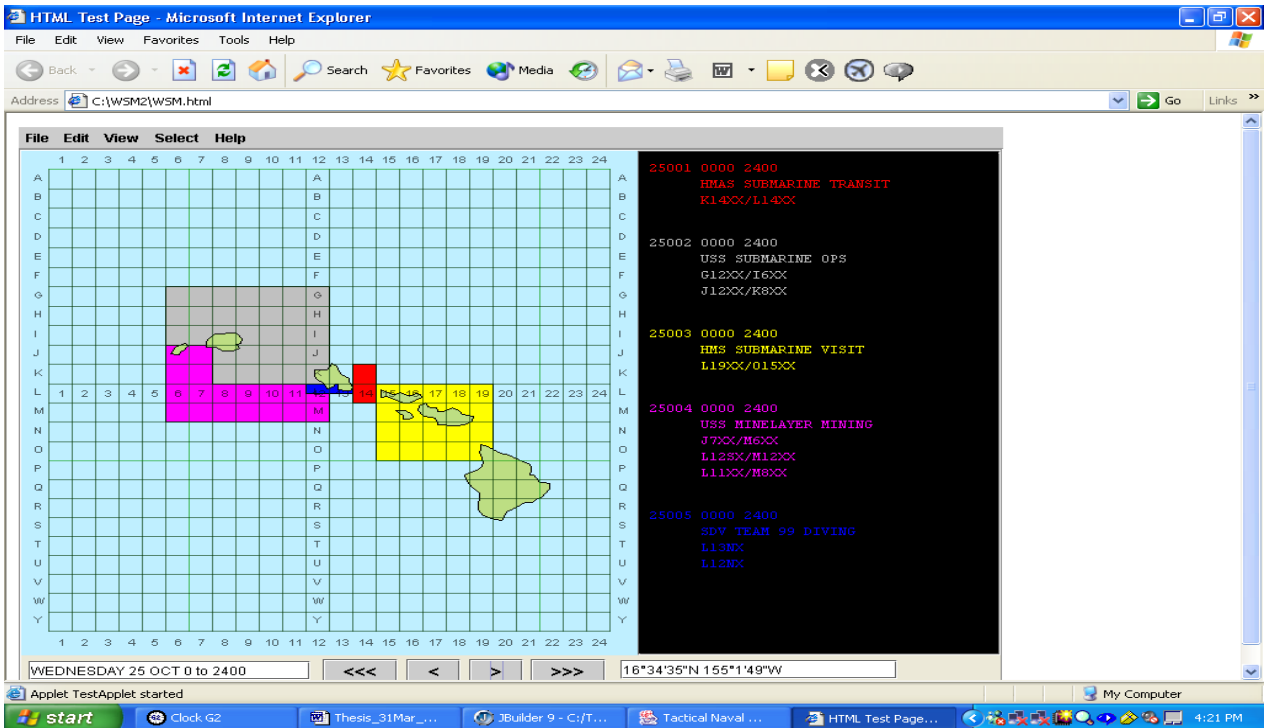
Figure 22.   Operational Detailed Picture GUI

61

Personnel and Weapons database sources are called from the tabbed panes labeled Personnel Forces and Weapons Inventory respectively (Figures 23 and 25). The GUI also allows the database data to be viewed in a XML format by pressing the appropriate button "Display XML ….." (Figures 24 and 26). Although display of raw XML and RDF data would not normally be available, for the purposes of this prototype, it was desired to visualize the converted data. The USMTF Data Button will display each Mission's message directive via a call to the registered textual display application (Figure 27). Selecting the adjacent "XML Data" Button will translate the USMTF directive into an XML format and then display the resulting file by the registered XML capable browser (Figure 28). The next step in the decision process provides a visual display of the critical Mission data in a RDF format by depressing the "RDF Data" Button in the selected Mission Type (Figure 29). Another critical information resource is the Mission specific Rules of Engagement directive that is called from the respective Mission Type tabbed pane annotated Rules of Engagement (Figure 30). The following figures display the various examples cited above.
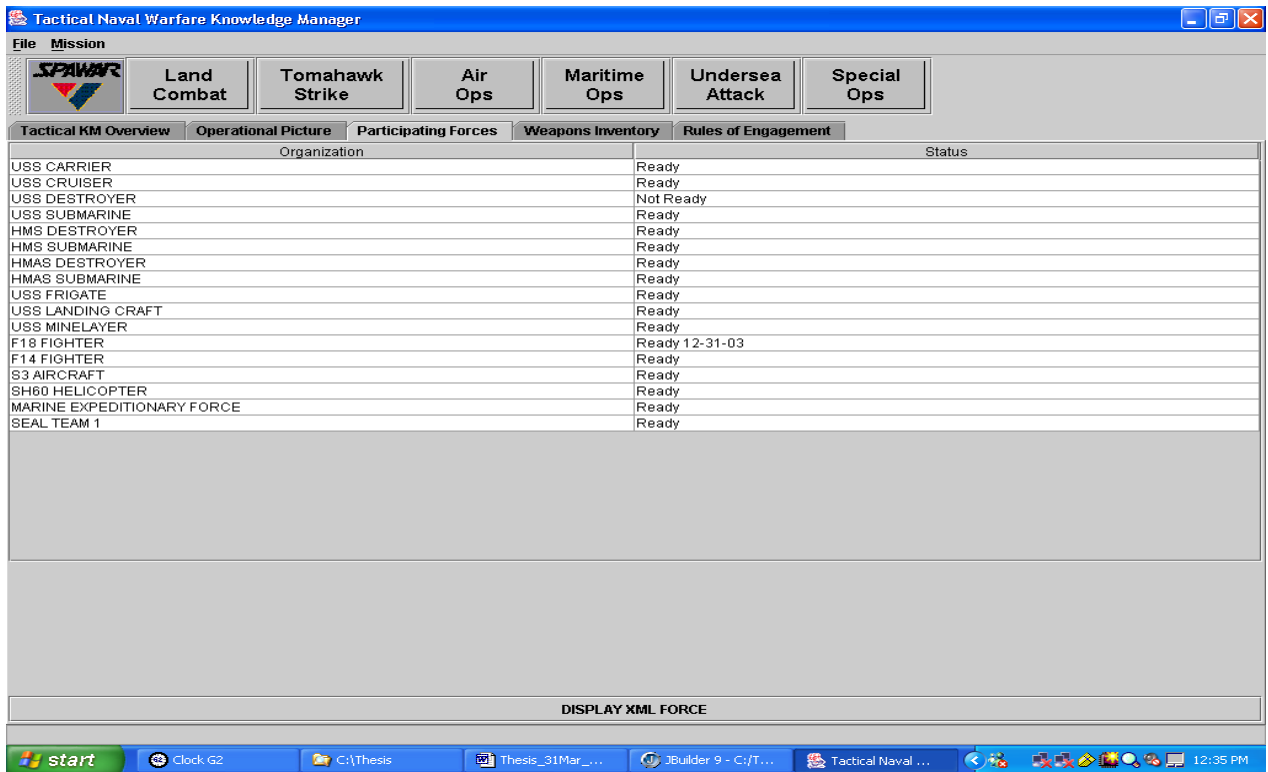
Figure 23.   Participating Forces GUI
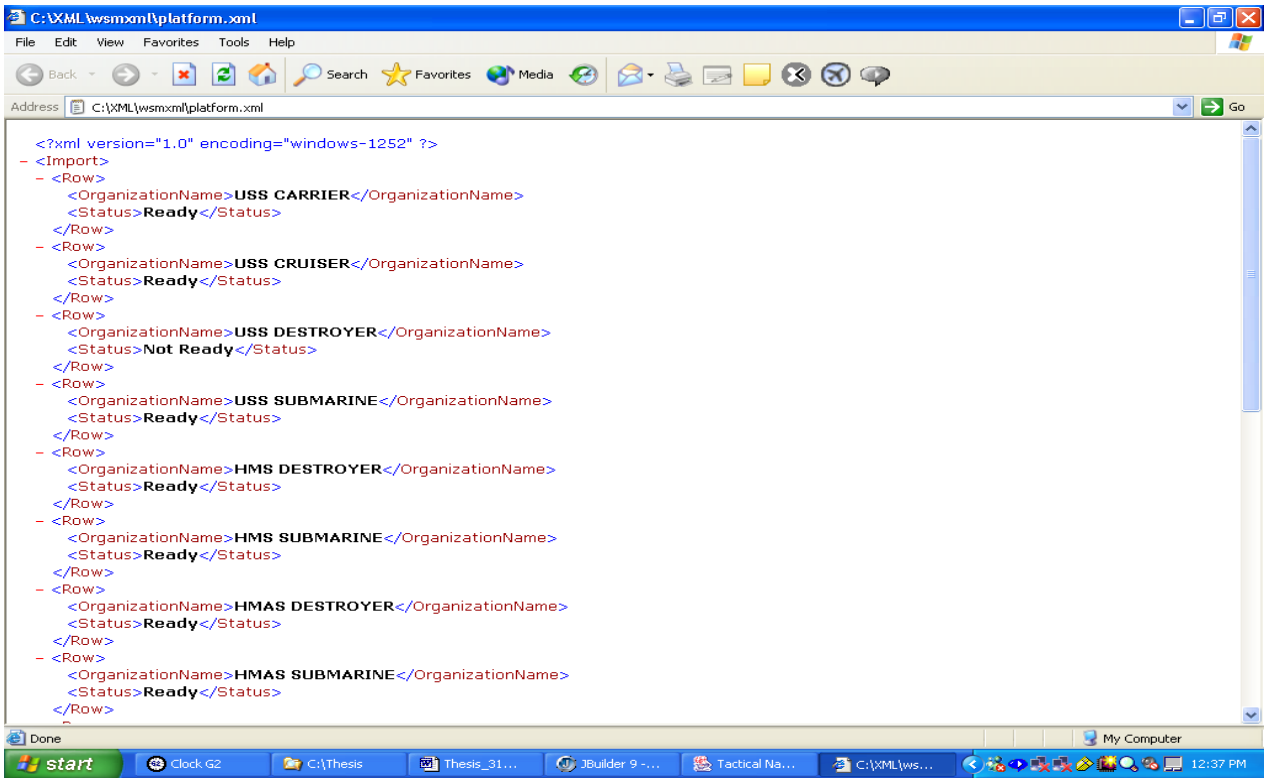
Figure 24.   Participating Forces (XML Version) GUI
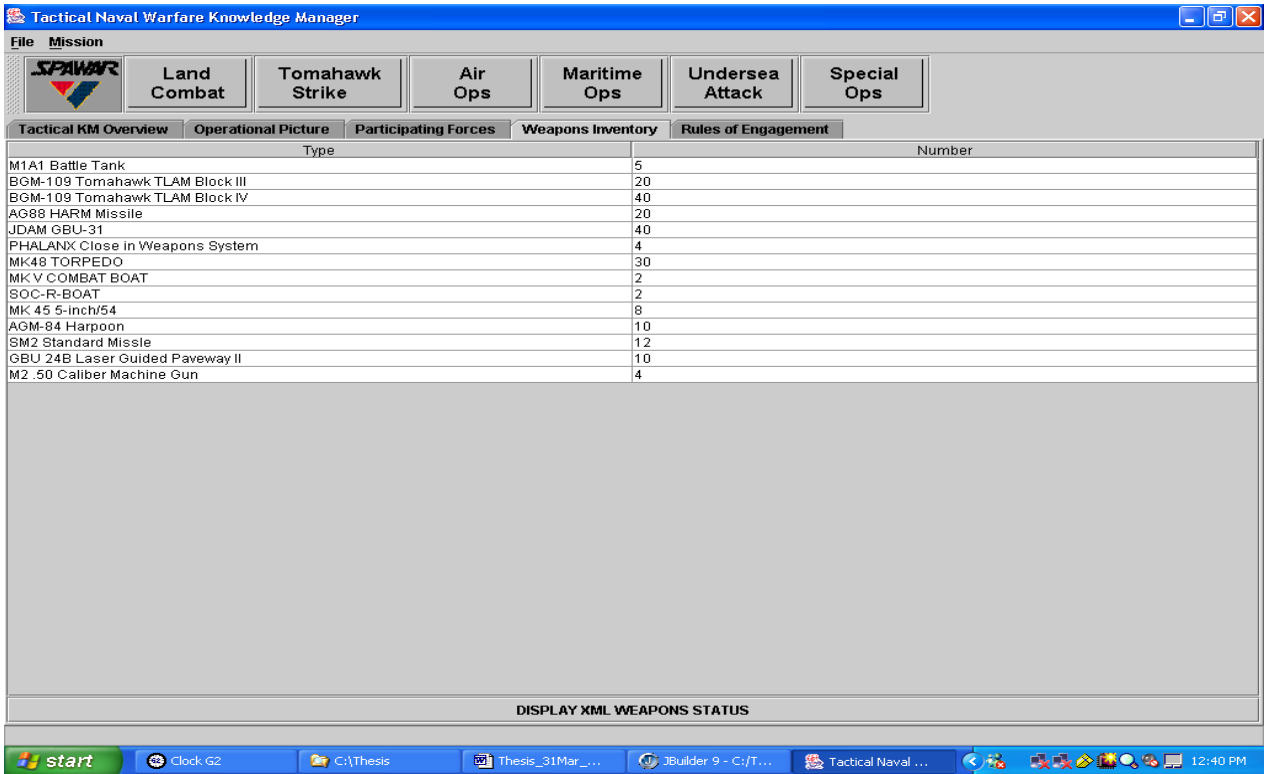
Figure 25.   Weapons Inventory GUI
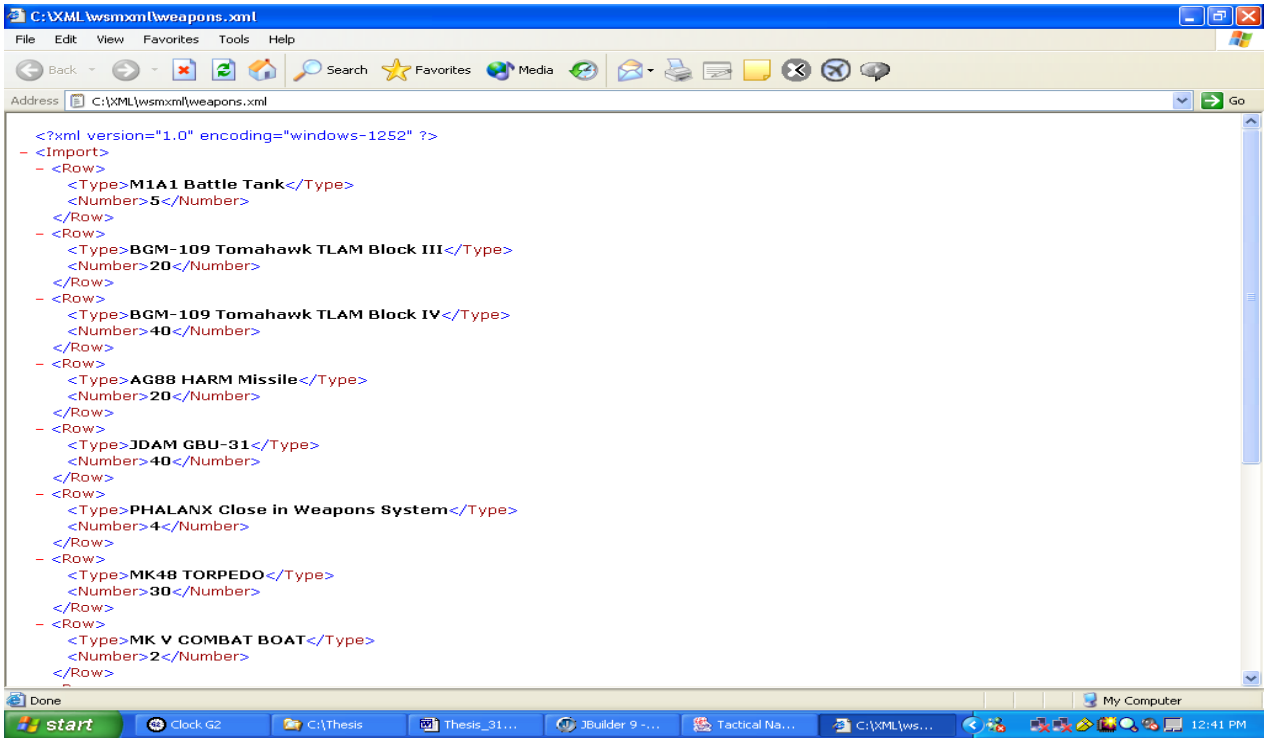
Figure 26.   Weapons Inventory (XML Version) GUI
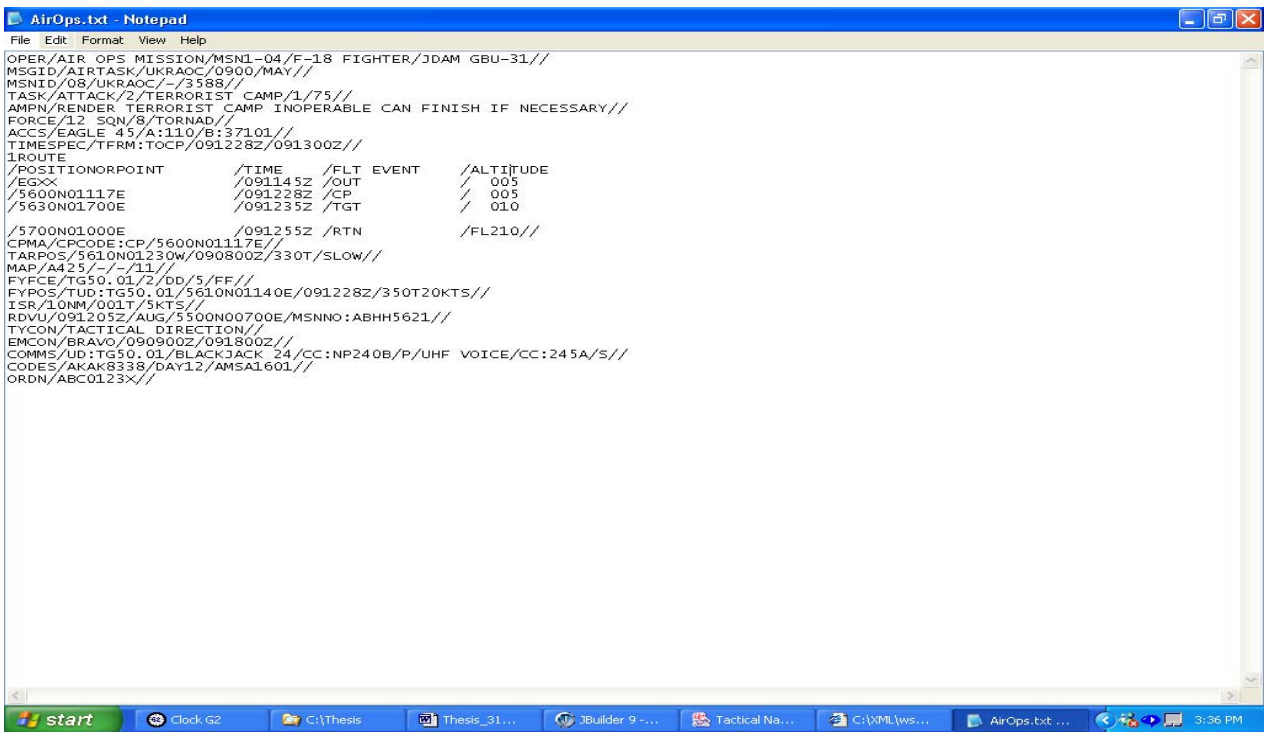
Figure 27.   USMTF Textual Formatted GUI

64
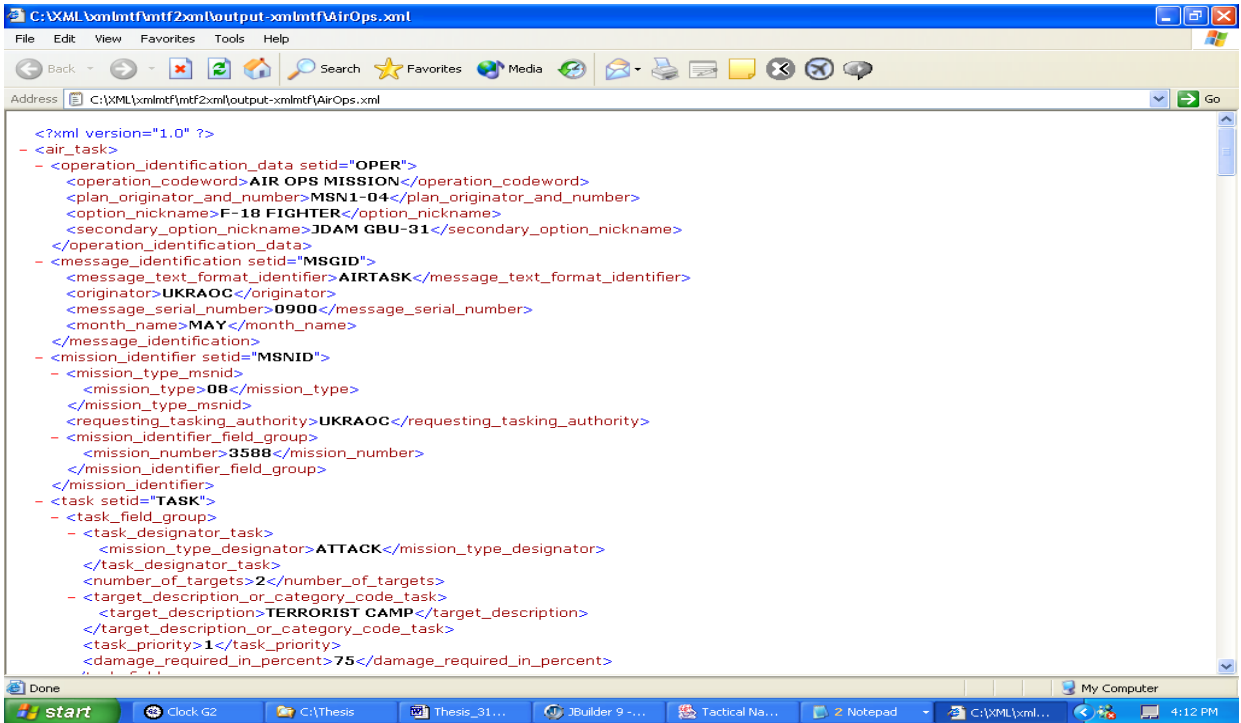
Figure 28.　　　USMTF XML Data GUI
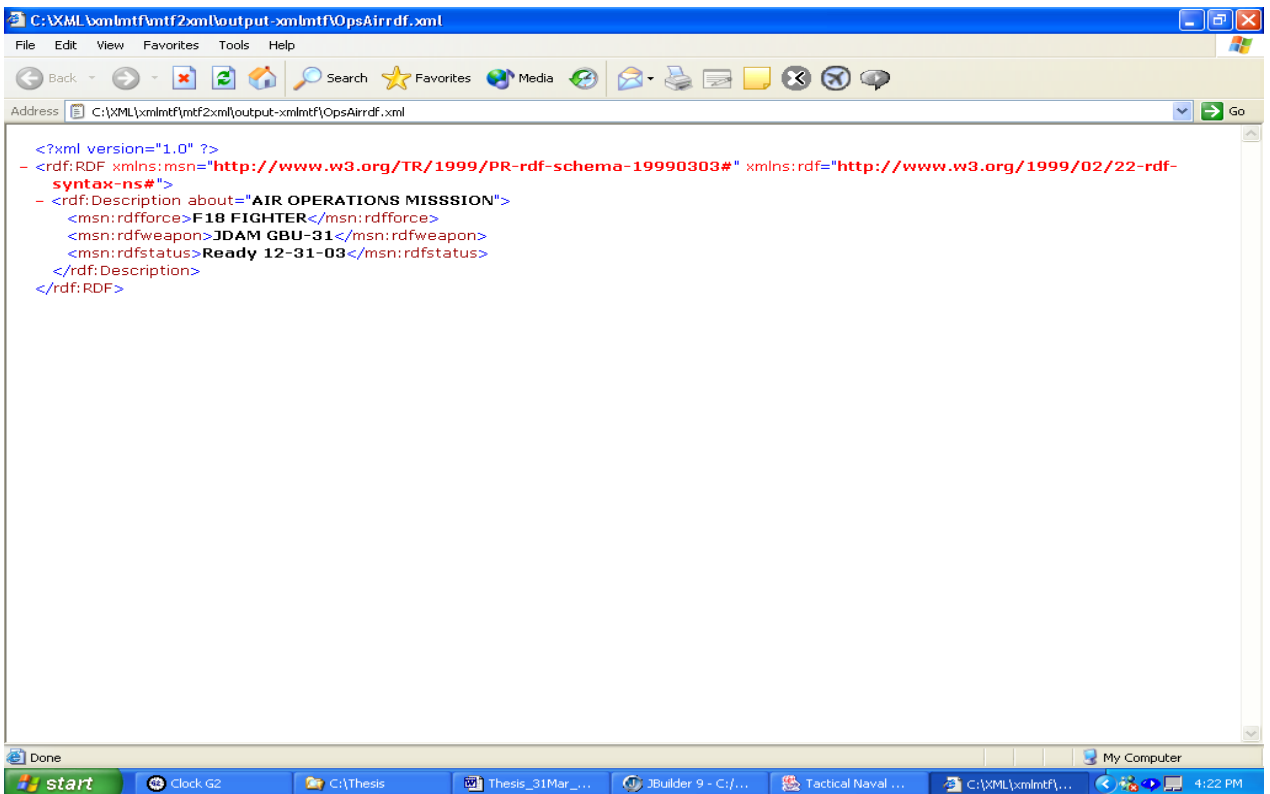
Figure 29.　　　Mission Critical RDF Data GUI

65
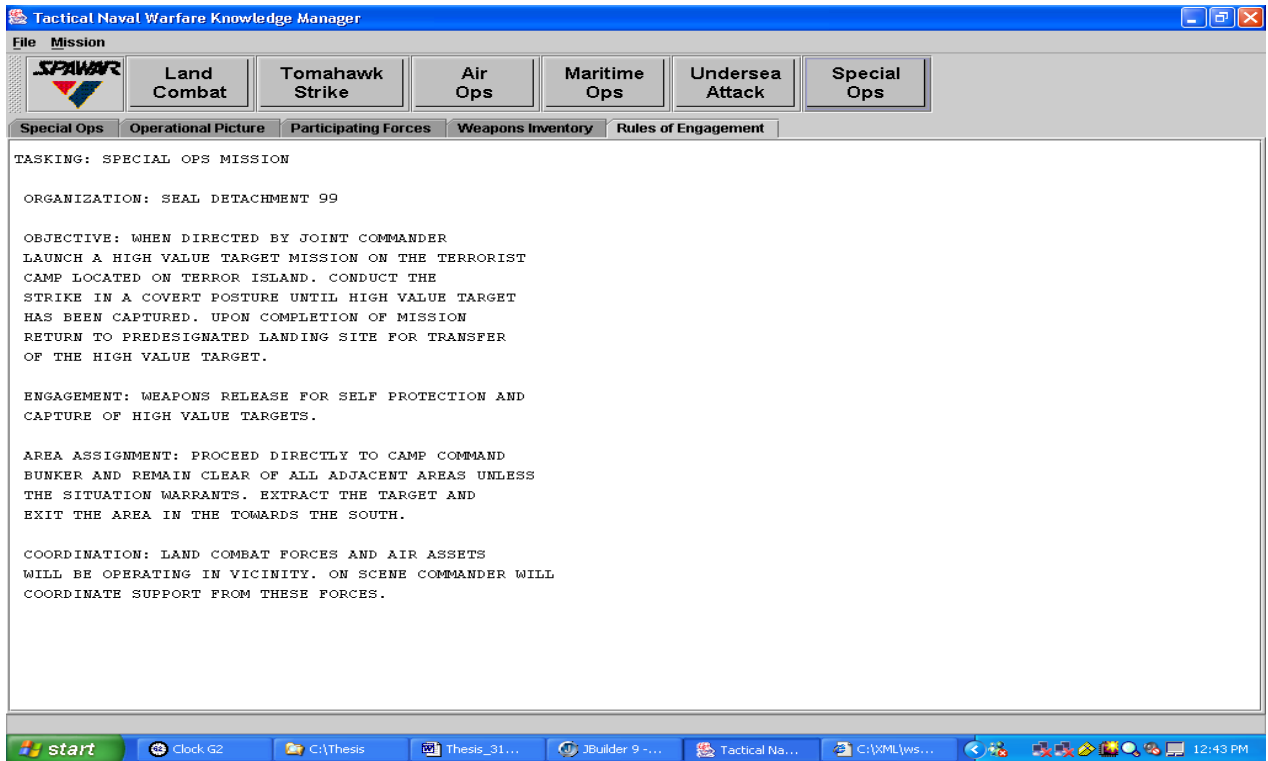
Figure 30.   Rules of Engagement GUI

Finally, the next page displays the Version 1.0 About Panel (Figure 31) and the SPAWAR logo (Figure 32) which is used as the opening Splash Panel while the program is loading.
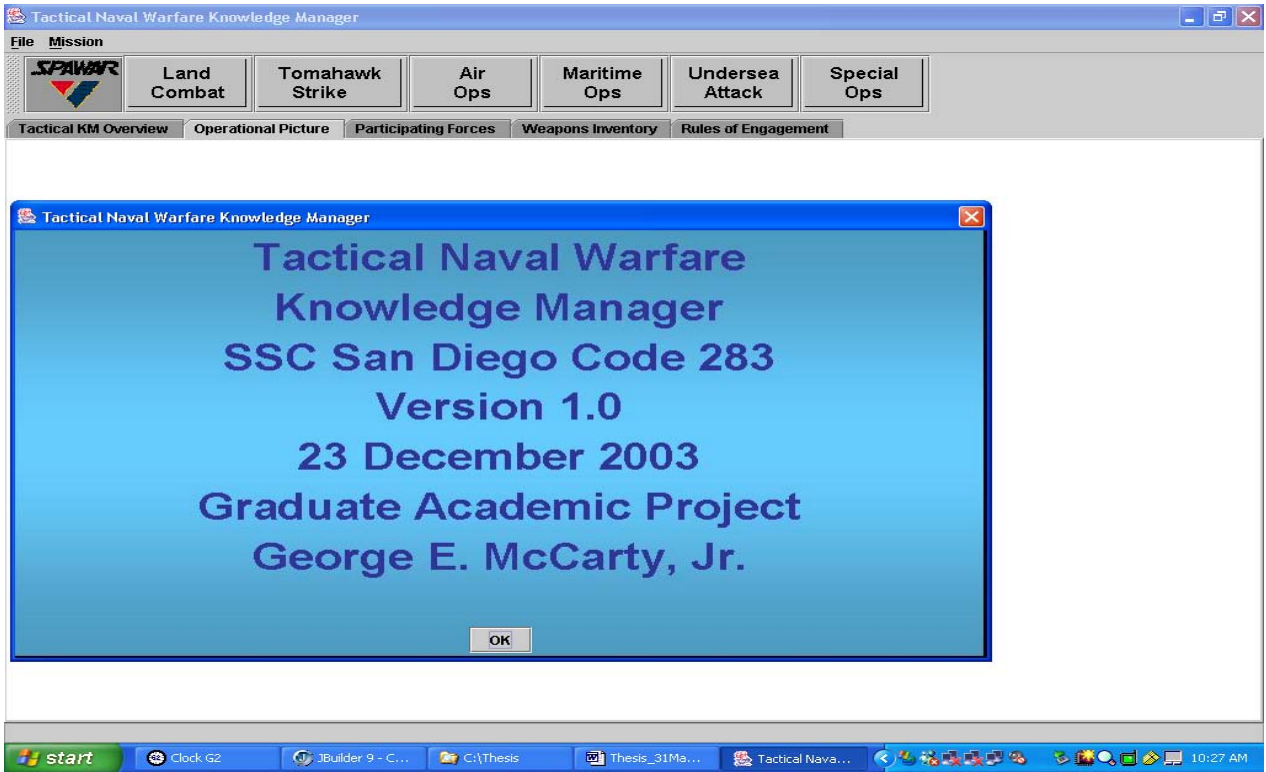
Figure 31.   About Panel GUI
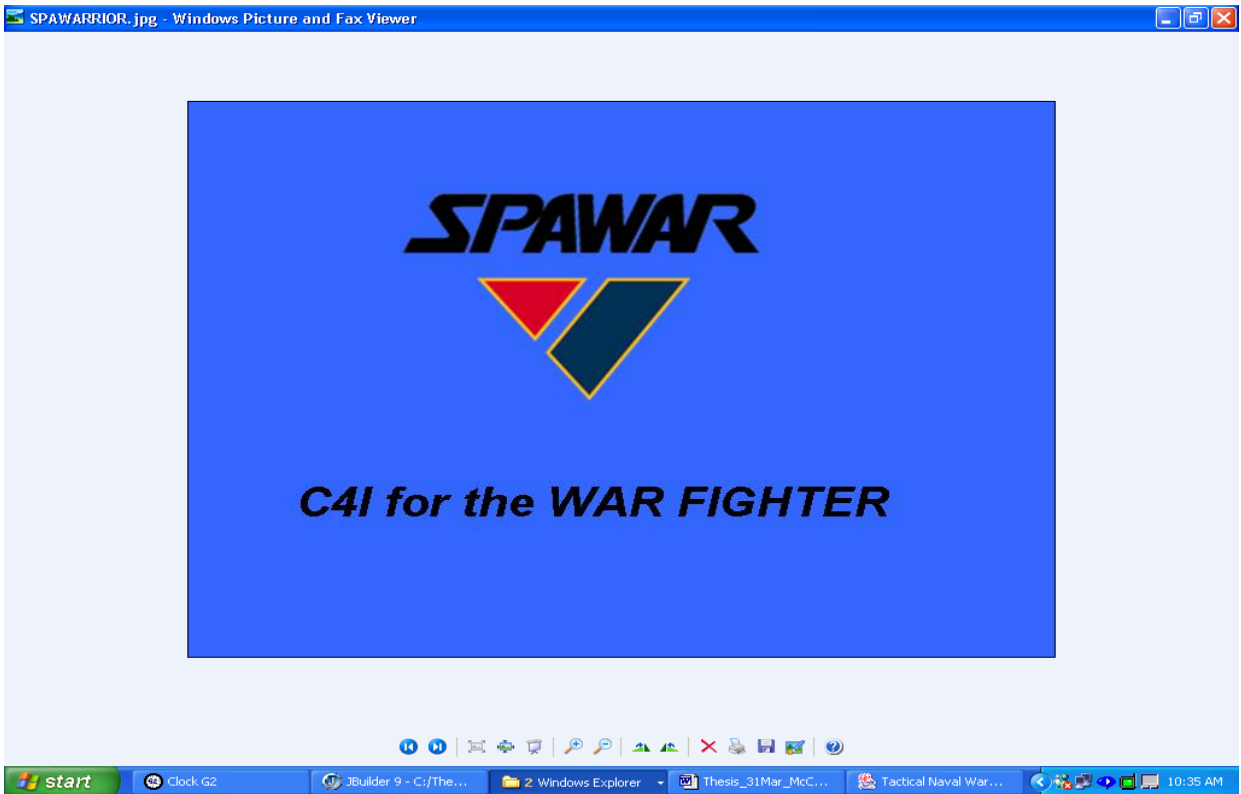
Figure 32.   Opening Splash Panel GUI

THIS PAGE INTENTIONALLY LEFT BLANK

V. IMPLEMENTATION

The Tactical Naval Warfare Knowledge Manager prototype was developed to evaluate the usefulness of employing semantic qualities such as XML and RDF in a military domain environment. A related task was to address the benefits of related toolsets including Ontology prototypes in this process. In the quest of achieving a Knowledge Management design process that progressed from high level architecture to production of usable source code, the challenge became similar to implementing a software product under current UML toolset maturity. However, as experienced under the constraints of this limited development, the basic foundation was created to explore automated development in this area. Therefore the detailed coding shown below has been based on the design provided by the higher level toolsets.

A. CODING

The detailed Tactical Naval Warfare Knowledge Manager design implemented the following classes:

- NavTacKM Class: Main class that calls all other classes.

- RDFXMLParser Class: Key class in processing XML and RDF data.

- ResourceDb Class: Resource Database access.

- KMIntroduction Class: Opening page of application and instructions for user input.

- CombatLand Class: Process data and display stoplight readiness decision for a Land Combat Mission type.

- StrikeTomahawk Class: Process data and display stoplight readiness decision for a Tomahawk Strike Mission type.

- OpsAir Class: Process data and display stoplight readiness decision for an Air Operations Mission type.

- OpsMaritime Class: Process data and display stoplight readiness decision for a Maritime Operations Mission type.

- AttackUndersea Class: Process data and display stoplight readiness decision for an Undersea Attack Mission type.

- OpsSpecial Class: Process data and display stoplight readiness decision for a Special Operations Mission type.

- DecisionLight Class: Displays current stoplight state.

- Light Class: Energizes stoplight.

Appendices A through E provide source code extracts from the above Class listing for detailed examination of unique functionalities. Due to the large number of source code lines, many routine coding functionalities have been excluded from these extracts. For a more detailed listing contact author at george.mccarty@navy.mil.

B.   **INTEGRATION**

Following initial integration testing of the design described above, it became obvious that user input was required to provide a feature that enabled the Tactical Commander to identify the critical data elements required in the decision process. Therefore a Mission Planning module was added to the application. Providing selectable criteria for decision process evaluation was envisioned as the next logical step to achieve a useful level of knowledge management. Furthermore many of the examples employed in the source code are based on hard coded attributes which could easily be changed to externally referenced values via the web or user input. A GUI was assembled to enable additional analysis of a user input design. Each Mission attribute required an option that allowed designation as a critical data element for use in the decision process. In addition, selections were required to allow text modification or entry of USMTF data, Rules of Engagement instructions, and/or recording of relevant Implicit Knowledge input. The completed total mission package may then be saved as a template for future usage and also entered as the current mission parameters for use in the decision processing. The resulting design is displayed in the following GUI:
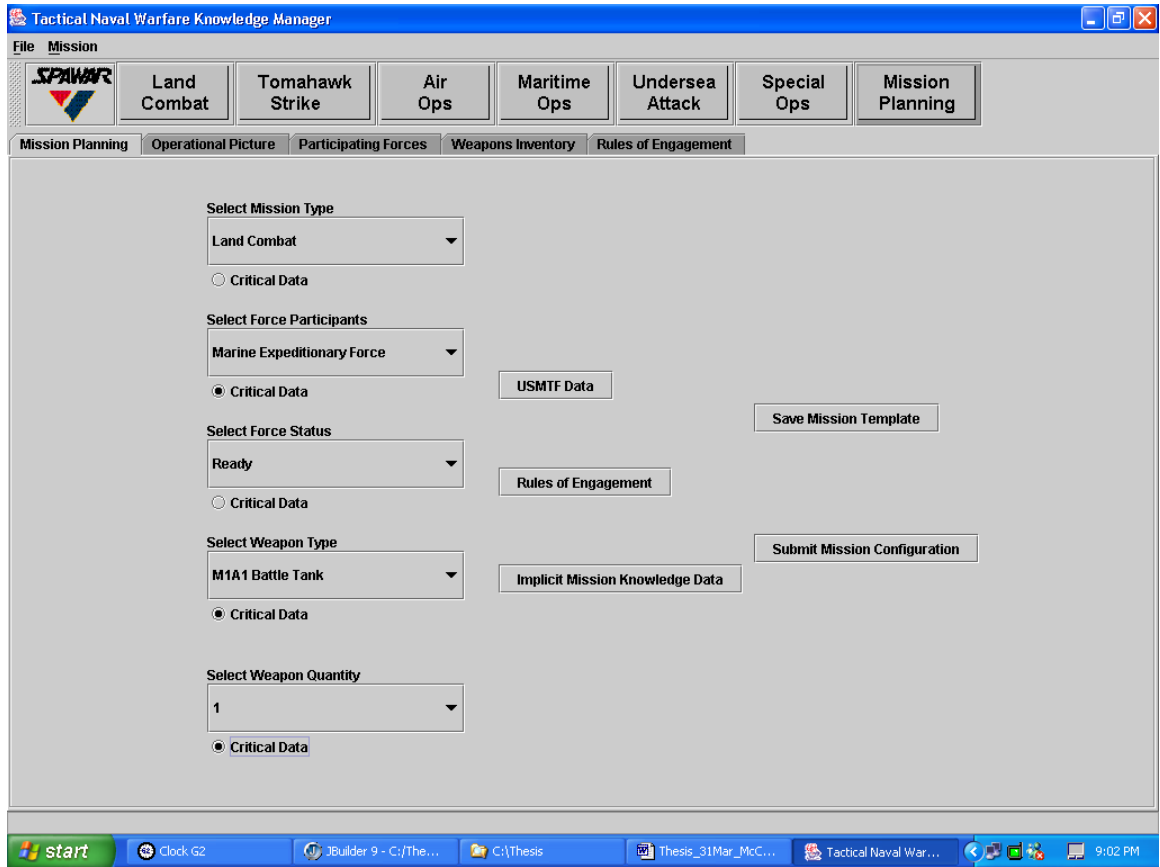
Figure 33.   Mission Planning GUI

Each mission attribute is available via a drop down box and can be selected as a critical data element. Although not incorporated in this application, the attributes could be linked to a database or a XML source. This proposed feature would allow continuous updating of dynamic data such as participating platforms, available weapons, or required quantities. The importance of evaluating each critical data element and its relationship to integration can be seen in the following figure:

71
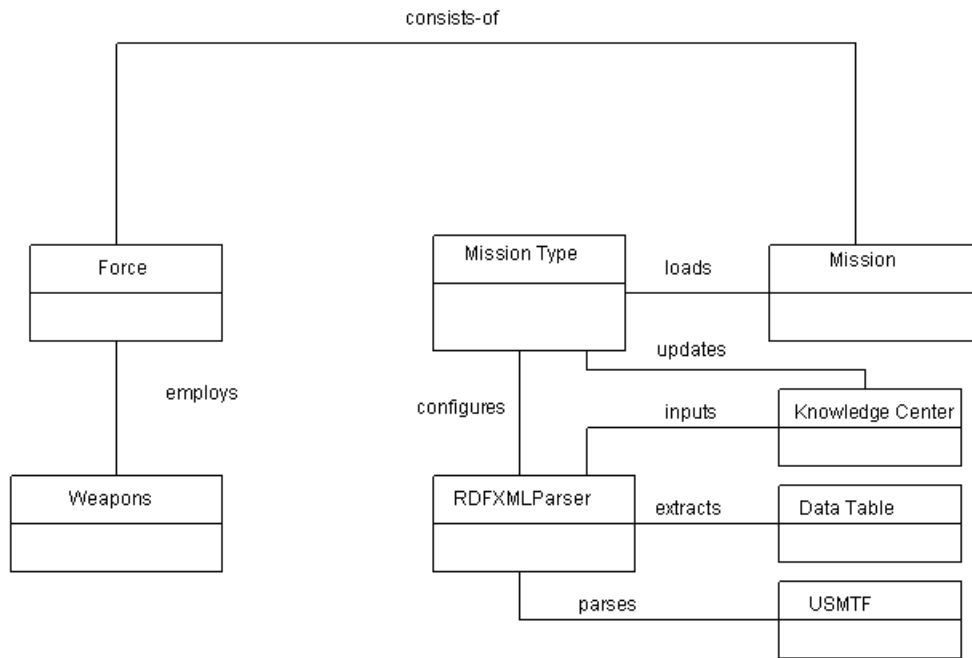
Figure 34.   Integrated NAVTACKM Relationship

Understanding the cause and effect relationship enables the Commander to make a critical data assignment that results in a decision tailored to the Mission's need. Updating the data on a real time basis will allow the decision process to be dynamically adjusted based on user input.

VI.   RESULTS, EVALUATION & METRICS ANALYSIS

As previously specified in the abstract, the Knowledge Management Model was prototyped to evaluate automation within a Tactical Naval Warfare Commander's environment through the integration of XML and RDF concepts. Although limited in functionality, the prototype provided an opportunity to evaluate KM techniques based on feedback from both naval C4I engineering and fleet representatives. It also enabled an evaluation of decision oriented software within the context of RDF under an XML structure. Model evaluation included but was not limited to the following:

- Processing required technical links in RDF/XML for feeding the KM model from multiple information sources.

- Experimentation with the visualization of Knowledge Management processing vice traditional Information Resource Display techniques.

The most significant issue identified early in the design stage became creation of the KM model itself. Due to the predominant focus on Information by most C4I professionals, the standard for a Knowledge Management tool was not readily available and therefore subject to interpretation. As a result the design process required a unique approach to achieve a model that was not another Information Product under a different name. The knowledge oriented design was achieved by emulating a traditional submarine KM device that has been in existence for many years. Although electro magnetic in design, the Submarine Diving Panel (Figure 9) provided the ideal KM device to emulate an automated tool capable of enabling a decision process. Furthermore the Diving Panel display was based on knowledge gained over many years and has survived many evolutionary trends including nuclear power. Although a static device, Diving Panel redesign would be possible by conducting extensive hydraulic, wiring and electrical panel updates.

Applying Submarine Diving Panel design to a software concept was achieved by adopting a simple stoplight display for the purpose of indicating readiness to perform the specified mission. Similar to the Diving Panel, the prototype was designed to accept inputs from multiple sources in support of achieving a readiness decision. Unlike the

73

Diving Panel, the prototype included a user input functionality to specify the critical data required for determining the desired decision. This capability advanced the KM prototype model to a dynamic tool thereby achieving a leap in technology over the static tool.

More detailed technical results and feedback will be discussed in further detail below.

A.   TECHNICAL RESULTS

From a technical aspect, the model identified the following:

- Conversion of most required data formats to XML was feasible with existing or prototype translation devices.

- The potential exists to translate unformatted text to XML however the output was not sufficiently formatted to contribute to the decision process.

- A wide variety of XML software tools were identified to process the structured data.

- RDF enabled arrangement of the XML data elements in a format that allowed establishment of a decision oriented relationship

- XML in combination with RDF comprised a powerful semantic capability that was very adaptable to a WarFighters domain.

- A converted XML document averaged an increase of approximately 70% over the original file type. See the below detailed discussion concerning timeliness.

- The use of JAVA supported all model programming requirements including Ontology, RDF and XML toolsets.

Based on experienced gained during the design, development and testing stages of the Knowledge Management prototype, all of the above items contributed to the conclusions provided in the next section. The next paragraph will review comments submitted by engineering and fleet representatives concerning the approach implemented by the Knowledge Management Model.

B.   MODEL REVIEW COMMENTS

Although the KM model served to validate several semantic qualities, overall feedback indicated a desire to expand the prototype's capability to perform additional planning qualities. Interestingly, both engineering and fleet representatives primarily focused on the Mission Planning Module and the potential of exploiting the semantic quality. Specific comments included the following:

- Difficult technical processing of common format standards has been demonstrated as feasible in a WarFighter's domain.

- Fundamental semantic operations have been achieved.

- Next step is to enhance semantic qualities.

- Several complementary efforts have been prototyped or are in use. Potentially powerful capability by combining this prototype with complementary developments into one system.

- Relevant for future design strategy.

- Mission Planning Module has potential to serve as a knowledge resource by archiving previous operations for use as templates.

- KM Model prototype has the potential to perform modeling and simulation of various missions thereby enabling the user to train and observe operations prior to actual execution.

- Model should alert the user as to what data element changed the status from a one state to another.

As stated in the prelude to this section, one of the significant benefits to the KM Model development was that the prototype focuses the observer on what a knowledge toolset should provide. While the standard remains subjective, additional prototyping in the WarFighter's domain is essential to changing the paradigm from an Information to a Knowledge Management approach. The next paragraph will focus on metrics and specifically the impact on timeliness when implementing semantic qualities in a tactical environment.

C.   MEASUREMENT CRITERIA

1.   Timeliness

Due to the extensive use of XML in this application's development, it is appropriate to evaluate the relationship between XML and the formatted textual message equivalent. In tactical terms, data transfer time can be expressed in bandwidth of the connecting path. Under limited bandwidth conditions, data size becomes a significant factor when selecting a transfer format. This data size relationship can be important as show in the following analysis.

As discussed earlier in this thesis, XML is a structured format that can be translated between other formats. In following example, a small example between the XML format and the USMTF format will be compared:

USMTF Formatted Sample Message:

OPER/TEST/FUN//

MSGID/OPREP-3/S510//

TIMELOC/261600Z/ZAKSTONIA/INIT//

GENTEXT/INCIDENT IDENTIFICATION AND DETAILS/PFC JOHN DOE//

RMKS/-- MESSAGE FOR TEST PURPOSES ONLY --UNCLAS //

EQUIVALENT XML Formatted Document:

```
<?xml version="1.0"?>
<oprep_3>
  <operation_identification_data setid = 'OPER'>
    <operation_codeword>TEST</operation_codeword>
    <plan_originator_and_number>FUN</plan_originator_and_number>
  </operation_identification_data>
  <message_identification setid = 'MSGID'>
    <message_text_format_identifier>OPREP-
3</message_text_format_identifier>
    <originator>S510</originator>
  </message_identification>
  <event_time_and_position setid = 'TIMELOC'>
    <event_time_timeloc>
      <event_day_time>
        <day>26</day>
        <hour_time>16</hour_time>
        <minute_time>00</minute_time>
```

```
        <time_zone>Z</time_zone>
      </event_day_time>
    </event_time_timeloc>
    <location_of_event_timeloc>

<location_of_event_place_name>ZAKSTONIA</location_of_event_place_name>
    </location_of_event_timeloc>
    <report_status>INIT</report_status>
  </event_time_and_position>
  <general_text_information setid = 'GENTEXT' >
    <gentext_text_indicator>INCIDENT IDENTIFICATION AND
DETAILS</gentext_text_indicator>
    <free_text xml:space = 'preserve'>PFC JOHN DOE</free_text>
  </general_text_information>
  <remarks setid = 'RMKS' >
    <free_text xml:space = 'preserve'>-- MESSAGE FOR TEST PURPOSES ONLY
--UNCLAS </free_text>
  </remarks>
</oprep_3>
```

The USMTF formatted message contained 174 data elements.

The XML equivalent document contained 1144 data elements.

On platforms where bandwidth is at a premium, an increase of data by a factor of ten may be unacceptable. Therefore additional analysis is required to determine if the above conversion truly represented the XML/USMTF relationship.

From the above sample, several observations are possible:

- Length of tag impacts XML document size

- Size of USMTF message could alter the relationship (i.e. shorter message results in greater disproportional increase as a result of the XML conversion process)

For example an analysis [20] of typical message size at a sample operational site identified the following usage:

- Short message length: 47 lines (0.75 page)
- Average message length, for most messages: 93 lines (1.5 pages)
- Long message length: 186 lines (3 pages)

The analysis [20] also compared the conversion of short, medium and large messages to XML as illustrated in the following table:

Message:	Characters (w/spaces)	Characters (no spaces)	Lines	Words	Difference % Increase (Characters w/spaces)
Text-short	1126	1080	47	87	1088
XML - short	2214	2164	74	123	97%
Text - average	2729	2656	93	160	2180
XML - average	4909	4810	163	261	80%
Text - long	6246	6143	186	283	4311
XML - long	10557	10399	336	493	69%

Table 3.   Comparison of Message Text Format versus XML equivalent

The data indicates that longer messages are more efficiently represented by XML than the shorter counterparts. The table also validates the earlier sample result of a significant increase in smaller size messages. It should be noted that the table data was acquired by a XML conversion process that was more focused on tag efficiency than the sample USMTF application. Most disturbing is that the average message most likely to be encountered is increased in size by 80% in an XML format. Extrapolation of this data found that based on 70 average sized messages, the plain text format would total 25,215,960 characters and the XML format would total 42,651,840 characters (a 69% increase in characters). The impact on timeliness is discussed in the next section under Conclusions.

VII. CONCLUSION

A. REVIEW STRATEGY

Prior to discussing the specifics of this analysis, a review of the thesis goal is necessary to ensure that key points are covered as expressed in the opening abstract. Therefore the following extract is provided for review:

> This thesis will focus on applying RDF and XML technologies to advance Naval Tactical KM as well as evaluating the integration challenges of this unique environment.

Due to the scope of the Knowledge Management and Semantic subject matter, it appears to be more logical to label this section as "Continuation" vice Conclusion. It became obvious very early in the study that narrowing the focus to a specific target would not achieve the objective described above. Therefore, the author attempted to cover a number of related Tactical Knowledge Management areas with sufficient background in order to provide the reader with an insight into the scope of the challenge. While this review will address attributes such as timeliness and flexibility, it is envisioned that more studies are required to move the military domain into a true knowledge management environment. The broad scope of the study also resulted in a multifaceted approach to evaluate the challenges. This approach resulted in the use of the following analysis techniques:

- Source code development and testing to determine the level of Knowledge Management software technology.

- Metrics study to examine the key attributes in a specific semantic area.

Utilizing several examination methods enabled the author to focus on the review from different aspects. Based on this approach at least one outcome came out differently than would have resulted with either analysis technique alone. Formatting of the following conclusions also mirror this diversion as timeliness will be examined from a metric analysis and flexibility will be reviewed from the source code effort. Finally, the study will conclude with recommendations and comments.

1. Timeliness

In comparison with a native textual USMTF format, the previous Metric analysis identified a significant overhead with conversion to a XML structured document. Although dependent on message size, the analysis identified that the overhead could cost as much as 100 percent efficiency with messages containing a small amount of data. Considering limited bandwidth in tactical situations, this data increase may be unacceptable. Therefore prior to converting all transport data to an XML format, additional research is required to determine available bandwidth, document size and alternative data transport methods.

2. Flexibility

During coding and application testing, conversion of all data to an XML format resulted in a greater flexibility for the following reasons:

- Simplified coding design requiring a minimum of classes to process a common XML data format.

- Availability of numerous toolsets allowing conversion to XML from a myriad of data formats.

- Use of XML as a semantic gateway for entry of data into a usable format.

- Provide user input into XML format allows dynamic decision processing without change in application design.

- XML has become a popular commercial standard and is increasing in popularity.

Based on the results from coding and metric analysis it would appear that XML would enhance flexibility and deteriorate timeliness. However the author suggests that an alternative is available to allow data transport efficiency and a common XML standard. The hybrid approach would be to transport data in the most bandwidth efficient format and perform data translation to and from XML at the sending and receiving sites. Due to the number of available translation devices, this approach would be feasible with many formats and shift the burden of XML overhead to the processor vice the transport system.

3. Recommendation/Comments

The following summary comments are provided:

Maximize use of XML as the standard file type for processing various format types.

Extend the use of RDF in a military domain to evaluate the use of Tactical Knowledge Management techniques.

Maintain the native serial format for transmission over IP and translate the data in the receiving platform

Continue the exploration of Ontology use for development within the military environment. Due to the prototype nature of most Ontology toolsets, it is anticipated that a higher level of maturity with increased scalability is required prior to implementation on a broad scale.

Bottom line is that the use of Tactical Knowledge Management in a military domain is feasible and can dramatically improve the ability of today's WarFighter to execute the required mission.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX A.   NAVAL TACTICAL KM CLASS

```
/**
 * Main Class for the Tactical Naval Warfare Knowledge Manager
 **/
public class NavTacKM extends JPanel {
//Array of Missions//
  String[] missions = {
      "CombatLand",
      "StrikeTomahawk",
      "OpsAir",
      "OpsMaritime",
      "AttackUndersea",
      "OpsSpecial"
    };
//Vector List of missions //
    private Vector missionsVector = new Vector();
// Resource bundle for holding text, links, etc. //
    private ResourceBundle bundle = null;
// A location for each mission //
    private AddModule currentMission = null;
    private JPanel missionPanel = null;

// NavTacKM Constructor//
    public NavTacKM() {
        initializeMission();
        }

// Main:  NavTacKM  //
    public static void main(String[] args) {
//Main Frame for application//
        frame = createFrame();
// NavTacKM Object //
        NavTacKM navtack = new NavTacKM(null);
    }
// Initialization Method //
    public void initializeMission() {
// Force Database Panel //
      JPanel force = new JPanel();
      force.setLayout(new BorderLayout());
// Force Database Table Layout //
      Container contentPane = getContentPane();
      tableNames = new JComboBox();
// Method to load each mission //
        void loadMission(String classname) {
            setStatus(getString(getString(classname + ".name")));
            AddModule mission = null;
// Try attempt to load each classname passed from mission array
//
            try {
                Class missionClass = Class.forName("navtackm."
                + classname);
```

```java
                    Constructor missionConstructor =
                    missionClass.getConstructor(new
                    Class[]{NavTacKM.class});
// Create mission object //
mission = (AddModule) missionConstructor.newInstance(new
Object[]{this});
// Add mission object to GUI //
                    addMission(mission);
                }
// Handling loading error //
catch (Exception e)
{
System.out.println("mission loading error");
                }
    }
//Method to load each Mission //
    void loadMissions() {
        for(int i = 0; i < missions.length;) {
          loadMission(missions[i]);
          i++;
        }
    }

// Resource Database Object //
      new ResourceDb(this);
// Try routine to query and display force database //
            try
            {
// Set up table and perform force database query //
                String query = "SELECT * FROM " + "Platform";
 if (rs != null) rs.close();
                rs = stmt.executeQuery(query);
                if (SCROLLABLE)

model = new ScrollingResultSetTableModel(rs);

else
model = new CachingResultSetTableModel(rs);
// Load force database onto scrollable table //
            JTable table = new JTable(model);
            scrollPane = new JScrollPane(table);
            force.add(scrollPane, BorderLayout.NORTH);
          }
// Handle database exceptions //
        catch(SQLException e)
        {  System.out.println("Error " + e);
        }
// Setup XML Table display //
        JButton xmlforce = new JButton("DISPLAY XML FORCE");
        force.add(xmlforce, BorderLayout.SOUTH);
// Action to launch XML display //
    xmlforce.addActionListener(new ActionListener(){
        public void actionPerformed(ActionEvent e) {
        try{
//Call XML file translation from XML database //
        Runtime.getRuntime().exec("cmd
        /c  \"C:\\XML\\wsmxml\\platform.xml\"");
```

```
                }
//Catch exception to XML file execution //
        catch (Exception f)
        {
                System.err.println("Failed to open xml file ");
        }
        }
    });
// Setup GUI for weapons database //
        JPanel weapons = new JPanel();
        weapons.setLayout(new BorderLayout());
// Try routine to query and display weapons database //
            try
// Set up table and perform weapons database query //
        {
         String query = "SELECT * FROM " + "Weapons";

         if (rs != null) rs.close();

         rs = stmt.executeQuery(query);

         if (SCROLLABLE) model = new
            ScrollingResultSetTableModel (rs);

         else  model = new CachingResultSetTableModel(rs);
// Load weapons database onto table //

            JTable table = new JTable(model);
            scrollPane = new JScrollPane(table);
            weapons.add(scrollPane, "Center");
        }
// Handle database exceptions //

            catch(SQLException e)
            {  System.out.println("Error " + e);
            }
// Setup XML Weapons Table display //
            JButton xmlweapons = new JButton("DISPLAY XML WEAPONS
            STATUS");
            weapons.add(xmlweapons, BorderLayout.SOUTH);
            xmlweapons.addActionListener(new ActionListener(){
// Action to launch Weapons Table XML display //
                public void actionPerformed(ActionEvent e) {
//Call XML file translation from XML database //
            try     {Runtime.getRuntime().exec("cmd
            /c   \"C:\\XML\\wsmxml\\weapons.xml\"");
                }
//Catch handling error to Weapons XML file execution //
            catch (Exception f) {
            System.err.println("Failed to open xml file ");
                }
            }
  });
```

85

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX B.   RDF & XML PARSER CLASS

```java
/**
 * Class to parse RDF and XML files [21]
**/
public class RDFXMLParser
{
// XML String and associated tag passed to parser //
      public static Vector getXMLTagValue(String xml, String tag)
// throws Exception
      {
            String xmlString = new String(xml);
            Vector v = new Vector();
            String beginTagToSearch = "<" + tag + ">";
            String endTagToSearch = "</" + tag + ">";
// First tag extracted //
            int index = xmlString.indexOf(beginTagToSearch);
            while(index != -1)
            {
// Last tag extracted //
            int lastIndex = xmlString.indexOf(endTagToSearch);
 // Tag data extracted //
            String subs = xmlString.substring((index
            beginTagToSearch.length()), lastIndex) ;

// Tag data added to Vector //
            v.addElement(subs);

 // Extract data until final tag //
            try
            {
            xmlString = xmlString.substring(lastIndex +
            endTagToSearch.length());
            }
// Handle errors //
            catch(Exception e)
            {
                  xmlString = "";
            }

// Loop until completed //

            index = xmlString.indexOf(beginTagToSearch);
            }
            return v;
      }
}
```

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX C.   DATABASE RESOURCE CLASS

```
/**
 * Class to access a specific resource database
 **/

    class ResourceDb {
// navtack object //
        NavTacKM navtack;
        public ResourceDb(NavTacKM navtack)
        {
              this.navtack = navtack;
// Attempt to open a resource database //
  try
    {
// Link to a java compatible driver //

      Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
// Declare string containing database location //
      String url = "jdbc:odbc:ODBC-Access-navtackm";

// Declare userid and password //
      String user = "";
      String password = "";
// Establish database connection //
      con = DriverManager.getConnection(url, user,
                                            password);
// Set for scrollable result //
      if (SCROLLABLE)
          stmt = con.createStatement(
          ResultSet.TYPE_SCROLL_INSENSITIVE,
          ResultSet.CONCUR_READ_ONLY);
      else
// Set for caching result //
          stmt = con.createStatement();
      DatabaseMetaData md = con.getMetaData();
      ResultSet mrs = md.getTables(null, null, null,
      new String[] { "TABLE" });
      while (mrs.next())
            tableNames.addItem(mrs.getString(3));
                              mrs.close();
      }
// Handling error for database try //
          catch(ClassNotFoundException e)
                {
                System.out.println("Error " + e);
                }
        }
```

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX D.   STOPLIGHT DISPLAY & ENERGIZE CLASSES

```
/**
 * Class to Display stoplight current state for each Mission type
 * After Ref [22]
 **/


class DecisionLight extends JFrame {

// Light object for maintaining current display across class //

Light currentdecision = new Light();

// DecisionLight Constructor //

public DecisionLight(){
Light tempdecision = new Light();
currentdecision = tempdecision;
Container contentPane = getContentPane();
contentPane.add(currentdecision);
}
}

/**
 * Class to energize stoplight current state for each Mission
type
 **/

class Light extends JPanel{
//State of stoplight//
      int nState;
//Class constructor//
public Light(){
    };

//Paint method for light object//
    public void paintComponent(Graphics g){
                g.setColor(Color.black);
                g.fillRect(100, 50, 100, 200);
                g.fillRect(115, 65, 70, 170);
                circle(g, (nState == 0) ? Color.red :
Color.red.darker().darker(), 150, 100);
                circle(g, (nState == 1) ? Color.yellow :
Color.yellow.darker().darker(), 150, 150);
                circle(g, (nState == 2) ? Color.green :
Color.green.darker().darker(), 150, 200);
                }

    //Method to build stoplight//
    private void circle(Graphics g, Color c, int nX, int nY) {
                final int nR = 20, nR2 = 2 * nR;
                g.setColor(c);
                g.fillOval(nX - nR, nY - nR, nR2, nR2);
                g.setColor(Color.black);
                g.drawOval(nX - nR, nY - nR, nR2, nR2);
                }
}
```

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX E.   LAND COMBAT CLASS

```java
  /**
   * Land Combat Mission Class
   **/

public class CombatLand extends AddModule implements
ActionListener{

// GUI declarations //
  JPanel outerPanel = new JPanel();
  JPanel innerPanel = new JPanel();
  JPanel rightPanel = new JPanel();
  JPanel northfarrightPanel = new JPanel();
  JPanel northPanel = new JPanel();
  JPanel northrightPanel = new JPanel();
  JButton luce;
  EmptyBorder border5 = new EmptyBorder(5, 5, 5, 5);
  EmptyBorder border10 = new EmptyBorder(10, 10, 10, 10);

// Class variables //
  int number;
  DecisionLight newdecision = new DecisionLight();
  static Reader in;
  static Writer out;
  LinkedList wsmList;

 // Main method for the Combat Land Class //

  public static void main(String[] args) {
    CombatLand mission = new CombatLand(null);
    mission.launchMission();
  }
// CombatLand Constructor //
  public CombatLand(NavTacKM navtack) {
    super(navtack, "CombatLand", "toolbar/JLandCombat.gif");
    launchCombatLand();
  }
// Method to initiate and display Land Combat Panel //
  public void launchCombatLand() {
      getMissionPanel().setBackground(Color.DARK_GRAY);
// Buttons to execute information and decision process //
      JButton choose = new JButton("Check Light");
      choose.setAlignmentX(RIGHT_ALIGNMENT);
      rightPanel.add(choose);
      JButton mtf = new JButton("USMTF Data");
      mtf.setAlignmentX(CENTER_ALIGNMENT);
      rightPanel.add(mtf);
      JButton xml = new JButton("XML Data");
      mtf.setAlignmentX(LEFT_ALIGNMENT);
      rightPanel.add(xml);
      JButton rdf = new JButton("RDF Data");
```

```java
            mtf.setAlignmentX(LEFT_ALIGNMENT);
            rightPanel.add(rdf);
            JButton decision = new JButton("Check Status");
            decision.setAlignmentX(RIGHT_ALIGNMENT);
            rightPanel.add(decision);
// mtf action listener //
        mtf.addActionListener(new ActionListener(){
        public void actionPerformed(ActionEvent e)
        {
// attempt to call usmtf file //
                try
                  {
                    Runtime.getRuntime().exec("cmd
                    /c \"C:\\XML\\xmlmtf\\mtf2xml\\input-
                    mtf\\LandCombat.txt\"");
                  }

// Handle errors //
                catch (Exception f)
                    {
                      System.err.println("Failed to open mtf file
");
                    }
              }
      });

// xml action listener //
xml.addActionListener(new ActionListener()
{
      public void actionPerformed(ActionEvent e)
{
// Runtime objects are the interface to system-dependent
capabilities //
             Runtime rt =Runtime.getRuntime();
// String declared for the USMTF & XML conversion bat file  //

             String callAndArgs = getString("LandCombat.xmllink");

// Attempt to execute the xml conversion routine //
             try
{
// XML bat file referenced in resource bundle is executed //
             Process child = rt.exec(callAndArgs);
                            child.waitFor();
// exit code examined for successful processing //
             System.out.println("Process exit code is:   " +
             child.exitValue());
                  }
// Handle runtime errors //
             catch (IOException f)
             {
               System.err.println("IOException starting
               process!");
             }
// Attempt to display result of XML translation //
             try
               {
```

```java
// Display XML file //
            Runtime.getRuntime().exec("cmd/c
            \"C:\\XML\\xmlmtf\\mtf2xml\\output-
            xmlmtf\\LandCombat.xml\"");
            }
// Handle runtime errors //
        catch (Exception f)
        {
            System.err.println("Failed to open xml file ");
        }
    }
  });


// rdf action listener //
  rdf.addActionListener(new ActionListener(){
      public void actionPerformed(ActionEvent e) {
      // declare local variables //
        String rdfFile = null;
        String newrdfFile = null;
        String rdf = null;
        String rdftest = null;
        String rdfstring = "Ready";
        StringBuffer buffer = new StringBuffer();
        StringBuffer rdfbuffer = new StringBuffer();
// Attempt to read xml file into buffer //
            try {
// Open LandCombat Mission xml file //
                FileInputStream fis = new FileInputStream(
                "C:\\XML\\xmlmtf\\mtf2xml\\output-
                xmlmtf\\LandCombat.xml");
                InputStreamReader isr = new
                InputStreamReader(fis,
                "UTF8");
                Reader in = new BufferedReader(isr);
                int ch;
// Read and convert file to text //
                while ( (ch = in.read()) > -1) {
                buffer.append( (char) ch);
            }
// Close all files //
            in.close();
            fis.close();
// Read buffer into string //
            rdfFile = buffer.toString();
            }
// Handle try file opening exception //
        catch (IOException g)
            {
              g.printStackTrace();
            }
// New parser object //
            RDFXMLParser parse = new RDFXMLParser();
        // Pass the string and tag name to the parser /
            Vector  box  =  parse.getXMLTagValue(rdfFile,
            "operation_codeword");
```

95

```java
// Extracted tag data returned for comparison //
                rdftest = (String) box.elementAt(0);
// Comparison of tag data and string attribute to energize
stoplight display //
                if (rdftest.compareTo(rdfstring) == 0)
                {
// nState equals stoplight condition //
                    newdecision.currentdecision.nState = 2;
                }
                else
                {
                    newdecision.currentdecision.nState = 0;
                    innerPanel.repaint();
                }
// Attempt to open converted database platform xml file //
    try {
            FileInputStream rdftext = new FileInputStream(
            "C:\\XML\\wsmxml\\Platform.xml");
// Create stream reader and buffer for platform xml file //
            InputStreamReader rtext = new
            InputStreamReader(rdftext,"UTF8");
            Reader rdfin = new BufferedReader(rtext);
            int ch;
// Read file into rdfbuffer in text format //
            while ( (ch = rdfin.read()) > -1) {
                    rdfbuffer.append( (char) ch);
                    }
// Close file //
            rdfin.close();
            rdftext.close();
// Read buffer to string //
            newrdfFile = rdfbuffer.toString();
        }
// Catch file open errors //
    catch (IOException g)
        {
            g.printStackTrace();
        }
// Parse object //
    RDFXMLParser rdfparse = new RDFXMLParser()
// Parse the data from the desired tag //
    Vector rdfnameparser = rdfparse.getXMLTagValue(newrdfFile,
    "OrganizationName");
    Vector rdfweaponparser = parse.getXMLTagValue(rdfFile,
                "secondary_option_nickname");
    Vector rdfstatusparser =   rdfparse.getXMLTagValue(
       newrdfFile,   "Status");
// Open file to hold RDF output //
    try
    {
            File rdfoutFile = new File)
            "\\XML\\xmlmtf\\mtf2xml\\output-
            xmlmtf\\LandCombatrdf.xml");
            FileWriter rdfout = new FileWriter(rdfoutputFile);
// Create rdf header //
            String rdfin = getString("CombatLand.rdfbegin") +
            getString("CombatLand.rdfmsnbegin");
```
96

```
  // Create rdf force database string //
            String rdfforce getString("CombatLand.rdfforcebegin")
            + rdfnameparser.elementAt(15).toString() +
            getString("CombatLand.rdfforceend");
// Create rdf weapons database string //
            String rdfweapon =
            getString("CombatLand.rdfweaponbegin") +
            rdfweaponparser.elementAt(0) +
            getString("CombatLand.rdfweaponend");
 // Create rdf status string //
            String rdfstatus =
            getString("CombatLand.rdfstatusbegin") +
            rdfstatusparser.elementAt(15) +
            getString("CombatLand.rdfstatusend");
// Create rdf end string //
            String rdfend = getString("CombatLand.rdfmsnend") +
            getString("CombatLand.rdfend");
// Combine Land Combat rdf string //
            String rdftotal = rdfin + rdfforce + rdfweapon +
            rdfstatus + rdfend;
// Read rdf string into file //
            rdfout.write(rdftotal);
// Close file //
            rdfout.close();
                        }
// Handle file opening errors //
      catch (IOException g)
            {
            g.printStackTrace();
            }
// Attempt to display RDF Land Combat mission file //
      try
            {
            Runtime.getRuntime().exec("cmd
            /c\"C:\\XML\\xmlmtf\\mtf2xml\\output-
            xmlmtf\\LandCombatrdf.xml\"");
                        }
// Handle file opening errors //
      catch (Exception f)
            {
            System.err.println("Failed to open xml file");
            }
          }
      });
// method to parse rdf input, compare critical elements and
energize stoplight //
      decision.addActionListener(new ActionListener(){
        public void actionPerformed(ActionEvent e) {
// declare local variables //
            String xmlFile = null;
            String rdfFile = null;
            String rdfstring = null;
            String xml = null;
            String xmltest = null;
// critical attributes //
            String status = "Ready";
```
97

```java
// This attribute is set remotely within a resource bundle //
            String partialStatus =
            getString("CombatLand.partialstatus");
            StringBuffer rdfbuffer = new StringBuffer();
// Attempt to open LandCombat RDF file for decision processing //
        try
{
                FileInputStream rdfstream = new FileInputStream(
                "C:\\XML\\xmlmtf\\mtf2xml\\output-
                xmlmtf\\LandCombatrdf.xml");
                InputStreamReader rdfisr = new
                InputStreamReader(rdfstream,
                "UTF8");
// Create buffer  & Read file into buffer in text format //
                Reader rdfreader = new BufferedReader(rdfisr);
                int ch;
                while ( (ch = rdfreader.read()) > -1) {
                rdfbuffer.append( (char) ch);
                }
// Close files //
                rdfreader.close();
                rdfstream.close();
// Read buffer into string //
                rdfFile = rdfbuffer.toString();
            }
// Catch file opening errors //
            catch (IOException g)
             {
               g.printStackTrace();
             }
// Create buffer //
         StringBuffer buffer = new StringBuffer();
// Attempt to open platform xml file //
         try {
                FileInputStream fis = new FileInputStream(
                    "C:\\XML\\wsmxml\\Platform.xml");
                    InputStreamReader isr = new
                    InputStreamReader(fis,"UTF8");
// Create buffer & and read in the xml file //
                Reader in = new BufferedReader(isr);
                int ch;
                while ( (ch = in.read()) > -1) {
                buffer.append( (char) ch);
                }
// Close all //
                in.close();
                fis.close();
// Read buffer into sting //
                xmlFile = buffer.toString();
                }
// Handle file opening errors //
             catch (IOException g)
              {
               g.printStackTrace();
              }
// Create parser object //
             RDFXMLParser rdfparser = new RDFXMLParser();
```

```
// Parse critical tag data and stow in Vector //
              Vector msnstatus =
              rdfparser.getXMLTagValue(rdfFile, "msn:rdfstatus");
// Retrieve critical data from Vector and stow in string format
//
              rdfstring = (String) msnstatus.elementAt(0);
// Compare Vector rdf string with stowed critical data //
              if (rdfstring.compareTo(status) == 0) {
                newdecision.currentdecision.nState = 2;
              }
              else if (rdfstring.compareTo(partialStatus) == 0) {
                newdecision.currentdecision.nState = 1;
              }
              else
              newdecision.currentdecision.nState = 0;
// Display decision in visual stoplight format //
              innerPanel.repaint();
              }
          }
        });
```

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

[1] J. Hahn and M. Subramani, "A Framework of Knowledge Management Systems: Issues and Challenges for Theory and Practice." ACM, pp 302-312.

[2] http://defcon.sdsu.edu./3/objects/km/defining/index.htm. Feb 2004.

[3] RizwanVirk, Chief Technology Officer CambridgeDocs, "Why Convert Documents into XML?" [http://www.cambridgedocs.com/id35.htm] Mar 2004.

[4] P. Patel-Schneider and J. Simeon, "The Ying/Yang Web: XML Syntax and RDF Semantics." ACM, pp 443-453.

[5] T. Berners-Lee. Semantic web road map. Internal note,World Wide Web Consortium, Sept. 1998. [http://www.w3.org/DesignIssues/Semantic.html], Feb 2004

[6] Emmanuel Pietriga, "IsaViz User Manual", Aug 2003..

[7] T. R. Gruber. A translation approach to portable ontology specifications. Knowledge Acquisition, 5(2), 1993.

[8] E. Grosso et al., "Knowledge Modeling at the Millennium-the Design and Evolution of Protégé-2000," Proc. 12th Int'l Workshop Knowledge Acquisition, Modeling and Management (KAW-99), 1999.

[9] Protégé-2000 Users Guide, 2002.

[10] N. Noy and D. McGuinness. "Ontology Development 101: A Guide to Creating Your First Ontology", 2001.

[11] Department of the Navy Chief Information Officer, Knowledge Management in the DON, Mar 03. [http://www.doncio.navy.mil/] Feb 2004.

[12] Naval Tactical Publication NTP-21(A) dated Jun 1997.

[13] M. Copelof, Captain, USN, OPNAV N09WD, "Expanded Technical Brief", PowerPoint Presentation, Nov 2002.

[14] M. Cokus, MITRE, Project Lead XML-MTF Development Team, "XML-MTF Mapping Third Public Working Draft Aug 2000.

[15] Gene Bellinger, The Knowledge Centered Organization, 1997.

[16] IBM Knowledge & Content Management Practice, 2000.

[17] James Herbsleb, David Zubrow, Dennis Goldenson, Will Hayes, and Mark Paulk. "Software Quality and the Capability Maturity Model", ACM Jun 1997 Vol 40 Nr 6 pp 30-40.

[18] About Submarines, "What is A "Green board"?" [http://www.queeenfish.org/noframes/mbt2.html] Feb 2004.

[19] Michael J. McQuilken, SSC SD Digital Nautical Chart Designer, Mar 2003.

[20] aSE-WSM Character Statistics Analysis, Ruth Brown, Jun 2003.

[21] RDF/XML Parser design was based on a similar effort by Anand of Veritas Software Corporation: [http://www.perfectxml.com/articles/XML/XMLParser.asp] Feb 2004.

[22] Barry W Pollack, Sierra Nevada College Stoplight.java Applet, [http://snow.sierranevada.edu/~csci/ExamplesV/StopLight/StopLight.pdf] Feb 2004.

INITIAL DISTRIBUTION LIST

1.   Defense Technical Information Center
       Ft. Belvoir, Virginia

2.   Dudley Knox Library
       Naval Postgraduate School
       Monterey, California

3.   Executive Director
       Space & Naval Warfare Systems Center San Diego
       San Diego, California

4.   Dr. Man Tak Shing
       Computer Science Department
       Naval Postgraduate School
       Monterey, California

5.   Dr. Peter Denning
       Department of Computer Science
       Naval Postgraduate School
       Monterey, California

6.   George McCarty
       Space & Naval Warfare Systems Center San Diego
       Submarine Communications & C4I Systems Division
       San Diego, California

7.   Michael Crawford
       Space & Naval Warfare Systems Center San Diego
       Submarine Communications & C4I Systems Division
       San Diego, California